Departamento de Informática
Universidad de Almería

Un Modelo Inteligente de Interacción Natural Adaptativo basado en Visión Artificial

TESIS DOCTORAL

Presentada por

Juan Jesús Ojeda Castelo

para optar al grado de
Doctor Ingeniero en Informática

Dirigida por

Dr. José Antonio Piedra Fernández
Profesor Contratado Doctor de Universidad
Departamento de Informática
Universidad de Almería

Dr. Luis Iribarne Martínez
Catedrático de Universidad
Departamento de Informática
Universidad de Almería

Almería, febrero de 2022

Escrita por: Juan Jesús Ojeda Castelo
Impresa por: XXXXXX (Almería, España)

febrero de 2022

Department of Informatics
University of Almería

An Adaptive Smart Natural Interaction Model based on Computer Vision

THESIS

Presented by

Juan Jesús Ojeda Castelo

for the degree of
PhD in Computer Science

Thesis supervised by

Dr. José Antonio Piedra Fernández
Associate Professor
Department of Informatics
University of Almería

Dr. Luis Iribarne Martínez
Full Professor
Department of Informatics
University of Almería

Almería, February 2022

Written and edited by: Juan Jesús Ojeda Castelo
Printed by: XXXXXX (Almería, Spain)

February 2022

Este documento ha sido generado usando LaTeX.

Todas las figuras y tablas de este documento son originales.

Un Modelo Inteligente de Interacción Natural Adaptativo basado en Visión Artificial

Juan Jesús Ojeda Castelo
Departamento de Informática
Grupo de investigación de Informática Aplicada (TIC-211)
Universidad de Almería
Almería, febrero de 2022

http://acg.ual.es

Con especial dedicación a mi madre

Agradecimientos

A José Antonio por su guía en este trayecto de investigación y por haberme iniciado en este mundo apasionante mucho antes de empezar esta tesis doctoral. Me enseñó un camino totalmente desconocido para mí, en el cual estoy inmerso y fascinado con él. José Antonio es una persona que no le importa compartir todo lo que sabe y está siempre dispuesto a ayudar en todo lo que puede. Me ha ayudado no solo en mi trayectoria profesional sino en las dificultades personales que han surgido a lo largo de los años. Su apoyo incondicional ha sido relevante no solo para formarme como investigador, sino también como persona y por estos motivos, además de mi mentor le considero parte de mi familia.

A Luis porque sin sus consejos y recomendaciones la tesis no estaría al mismo nivel que está ahora. Agradezco que me abriera las puertas a su grupo de investigación, creado y mantenido con esfuerzo y mimo durante todos estos años. Luis es un incesante trabajador, que se fija en esos detalles que pasan desapercibidos para la mayoría de nosotros pero que son los que marcan la diferencia. Además de aprender mucho de él en el ámbito de la investigación, estoy profundamente agradecido porque Luis es una persona cercana que se ha preocupado por mi bienestar y me ha ayudado siempre que lo he necesitado.

En definitiva, mis directores me han inspirado un enorme grado de admiración y respeto por las cualidades que poseen y no podría haber tenido unos directores mejores.

Por pertenecer al grupo de investigación Applied Computing Group (ACG) que está formado por excelentes personas de una gran calidad humana y con las que he estado colaborando todos estos años: Darwin Alulema, José Andrés Asensio, Rosa Ayala, Antonio Corral, Javier Criado, Antonio Jesús Fernández, Paco García, Manel Mena, Nicolás Padilla y Diego Rodríguez.

A mi familia, en especial a mi tía Pepi, mis tíos Rafael y José, mis primos Rafa, May y Alba y mi abuela Dolores, por tener la seguridad de que van a estar a mi lado a pesar de las circunstancias y por ayudarme cuando más lo he necesitado.

A José Aguado, José Daniel Ruiz y Sebastián Salvador porque no solo es necesario tener una familia que te apoye sino que si cuentas también con unos amigos fieles que siempre puedes contar con ellos, te puedes considerar afortunado.

A Shaz porque me enseñó lo que es el amor verdadero y me demuestra su cariño y afecto constantemente, siendo el origen de mi energía que me hace funcionar cada día. Su energía positiva, su fuerza y coraje, su determinación y seguridad en sí misma son una fuente de inspiración, mientras que su gran corazón y su bondad, empatía y comprensión infinita son la excusa perfecta para enamorarte. Shaz es una mujer única y muy especial que no solo ha alcanzado mi corazón sino que ha profundizado hasta mi alma.

A mi padre, sin el cual no hubiera podido hacer realidad la conclusión de esta tesis, gracias a su apoyo incondicional a lo largo de mi vida. Su perfeccionismo en cualquier tarea que realiza, ha hecho que me esfuerce en hacer lo mejor posible cada cosa que hago. Por cuidar y mantener unida esta familia durante todos estos años y especialmente por estar al lado de mi madre y entregarle todo tu amor hasta el último instante de su vida.

Gracias por tu ayuda, especialmente en estos momentos.

A mi querida madre, porque me enseñó la importancia de la educación y de la perseverancia para alcanzar tus objetivos. Una mujer entregada, amable y cariñosa que dedicó su vida a su hijo, a quien tengo que agradecer los valores que tengo hoy en día y la persona en la que me he convertido. Sé que tenías una espina clavada toda tu vida debido a que nunca pudiste cursar estudios universitarios. Esta tesis doctoral es tanto tuya como mía, fruto de tu dedicación, amor y compromiso durante todos estos años. Desafortunadamente, este es el último regalo que te puedo hacer con todo mi corazón pero espero que en este día te libere de esa espina. Para finalizar, solo quería decirte que me siento muy orgulloso de haber sido tu hijo y no podía haber deseado una mejor madre para guiar mi camino en esta vida.

Juan Jesús Ojeda Castelo
Departamento de Informática
Universidad de Almería
Almería, 2021

Índice

Lista de Figuras

Lista de Tablas

Resumen

En la actualidad existen diversas formas de interacción. Las más extendidas son la interacción mediante teclado y ratón en PC, gamepad en videojuegos y táctil en smartphone y tablet. Sin embargo, la interacción natural gestual sin necesidad de portar o manejar un dispositivo físico ofrecería diversas ventajas a nivel de adaptabilidad, accesibilidad y usabilidad para el usuario. Principalmente la accesibilidad beneficiaría a usuarios con diversidad funcional que, debido a sus limitaciones físicas, los modos más extendidos de interacción tradicional resultarían impracticables en algunos casos.

Esta Tesis doctoral se centra en el desarrollo de un sistema de interacción natural que se caracteriza por ser low-cost, adaptable e inteligente. En el contenido de esta, se puede apreciar tres partes claramente diferenciadas, que explican las etapas en el desarrollo del sistema y los diferentes dispositivos usados en las mismas. En primer lugar, se ha creado un sistema que tiene como dispositivo de interacción natural Microsoft Kinect v1 que permite controlar el movimiento de su cuerpo. Este sistema está compuesto por dos módulos. El primer módulo está orientado principalmente para las habilidades físicas del individuo, mientras que el segundo módulo se centra en las habilidades cognitivas. En esta parte del trabajo se ha colaborado con el Centro de Educación Especial Princesa Sofía de la provincia de Almería, lo que supuso que los propios estudiantes fueran los participantes del estudio y permitió comprobar la validez del sistema. En la evaluación se realizó una encuesta que fue cumplimentada por un conjunto de expertos valorando la usabilidad, modalidad educativa y comportamiento de los estudiantes. Además, se realizaron experimentos con usuarios para medir indicadores como el tiempo y el número de errores en la realización de una actividad. Esto facilitó la obtención de conclusiones acerca del sistema que ayudarán en su mejora.

En segundo lugar, se tiene como objetivo principal la adaptación de la interacción. El sensor utilizado fue Microsoft Kinect v2 debido a la experiencia satisfactoria proporcionada por su anterior versión. La principal aportación fue el diseño de un modelo dispositivo-interacción para poder adaptar la interacción e intentar generalizarla a un mayor número de usuarios. Las actividades propuestas para esta etapa fueron diseñadas con la colaboración de los profesores del Centro de Educación Especial Princesa Sofía. Una de las actividades desarrolladas tenía el fin de que los estudiantes asociaran conceptos respecto a una unidad didáctica. Otra actividad que fue creada tenía el objetivo de trabajar la lateralidad izquierda y derecha. Se realizaron dos tipos de evaluación: Una evaluación con expertos y una evaluación con usuarios finales. En la evaluación con expertos se aplicó el método de inspección con la combinación del recorrido cognitivo y la técnica de pensar en voz alta. En la evaluación con usuarios finales participaron

estudiantes con discapacidad física, auditiva, visual y autismo. Esta evaluación consistió en dos iteraciones donde los estudiantes realizaban las actividades y se almacenaban una serie de parámetros para obtener unas conclusiones.

En último lugar, se prescindió del dispositivo Kinect y se decidió hacer un estudio enfocado en la webcam. Esta decisión se debe principalmente a la incertidumbre con el futuro del dispositivo Microsoft Kinect, reducir el coste de adquisición y facilidad de uso. Con esta premisa se ha desarrollado un sistema de reconocimiento de gestos de la mano basado en Deep Learning y Lógica Difusa para determinar los mejores modelos de clasificación. Inicialmente se obtienen los datos que van a ser usados para el posterior entrenamiento con los modelos de Deep Learning. Para este propósito se han obtenido los vídeos de una base de datos de gestos con las manos titulada 20BN-Jester. Posteriormente, se procede a usar transferencia de aprendizaje con modalidad de fine-tuning con una serie de modelos pre-entrenados para que aprendan a clasificar los gestos con las manos. En total se han realizado 104 experimentos donde se han modificado distintos parámetros, entre ellos, el optimizador, número de gestos o la función de coste. A continuación, se han obtenido unas métricas a partir de dichos experimentos que serán las que alimenten al sistema experto difuso. Este sistema experto tiene implementado el sistema Takagi-Sugeno-Kang y está formado por 11 reglas. Estas reglas van a permitir analizar cada una de las distintas configuraciones para obtener un ranking de configuraciones ordenadas de forma descendente de acuerdo con la valoración que proporcione este sistema experto difuso.

El resultado derivado de la investigación realizada en la presente tesis ha propiciado un total de 6 contribuciones científicas, 4 en congresos internacionales con publicaciones en la serie Springer en Advances in Intelligent Systems and Computing, y otras 2 contribuciones en la revista internacional de impacto Multimedia Tools and Applications (Springer, JCR Q2, Computer Science).

Para concluir, la finalización de esta tesis ha dejado abiertas las presentes líneas de investigación: (a) el desarrollo de un sistema de interacción natural que integra Microsoft Kinect Azure como medio de interacción; (b) la creación de un sistema de interacción que sea portable y se pueda acoplar en diversos escenarios; (c) el desarrollo de un sistema que sea autoadaptativo con el objetivo de que adapte los gestos a las características de los usuarios; (d) la elaboración de un sistema híbrido de Inteligencia Artificial para ofrecer un mejor rendimiento en el reconocimiento de gestos y (e) la

creación de un sistema multimodal que incorpore diversos modos de interacción, por ejemplo, reconocimiento de gestos y reconocimiento de voz.

Agradecimientos: TIN2017-83964-R, "Estudio de un enfoque holístico para la interoperabilidad y coexistencia de sistemas dinámicos: Implicación en modelos de Smart Cities".

Abstract

Currently there are various ways of interaction. The most widespread are the interaction via keyboard and mouse on PC, gamepad in video games and touch on smartphone and tablet. However, natural gestural interaction without the need to wear or handle a physical device would offer various advantages in terms of adaptability, accessibility and usability for the user. Mainly, accessibility would benefit users with special needs who, due to their physical limitations, the more extended modes of traditional interaction would be impractical in some cases.

This dissertation consists of the development of a natural interaction system that is characterized by being low-cost, adaptable, intelligent and portable. The content has been divided into three areas; each one explaining the stages in the development of the system as different devices are used in each stage. The first phase involves creating a system as a natural interaction device by using Microsoft Kinect v1 in which the users are able to control the software by moving their body. This software is composed of a set of activities that are divided into two modules where the first module is mainly oriented towards the physical abilities of the individual while the second module focuses on cognitive abilities. We have collaborated with the Princesa Sofía Special Education Center in the province of Almería where the students were the participants of the study that was carried out to check the validity of the system. In this study, a survey was carried out to evaluate the usability, educational modality and behavior of the students. In addition, experiments were conducted with users who had to complete the activities where the time and the number of errors were measured to obtain specific information about the system.

For the next phase, the main objective is the adaptation of the interaction. The sensor used was Microsoft Kinect v2 due to the satisfactory experience provided by its previous version. The main focus was for the design of the device-interaction model to be able to adapt to the interaction of a greater number of users through the characteristics of the device. The activities proposed for this stage were again designed with the collaboration of the teachers of the Princesa Sofía Special Education Center. One of the activities developed involved the students associating concepts regarding a didactic unit. Another activity that was created assisted the students to work on their left and right laterality. Two types of evaluation were conducted at this stage involving the experts and end users. In the evaluation with experts, the inspection methods applied were the combination of the cognitive walk and the technique of thinking aloud. Students with physical, hearing, and visual disabilities and autism participated in the evaluation with end users. This evaluation consisted of two iterations of activities carried out by the students where a series of parameters during the activities were stored in obtaining conclusions.

In the last phase, the Kinect device that had been used during the work was dispensed with and a study using a webcam was conducted instead. This decision is mainly due to the belief of the uncertainty with the future of the Microsoft Kinect device, reduce cost of ownership and ease of use. With this premise, Deep Learning and Fuzzy Logic have been applied to classify the hand gestures and determine the best configurations among all those that have been tested. In the first place, the data that will be used for the subsequent training with the Deep Learning models is obtained. For this purpose, the

videos have been obtained from a database of hand gestures entitled 20BN-Jester. Once the data have been collected, we proceeded to use learning transfer with a fine-tuning modality with a series of pre-trained models so that the system learns to classify hand gestures. In total, 104 experiments have been carried out where different parameters have been modified, including the optimizer, number of gestures and the cost function. Some metrics have been obtained from the said experiments that will be the ones that feed the fuzzy expert system. This expert system has the Takagi-Sugeno-Kang system implemented and is made up of 11 rules. These rules will allow the analysis of each of the different configurations and thus obtain a classification of said configurations listed in descending order according to the assessment provided by this fuzzy expert system.

As a result derived from the research carried out in this thesis, a total of 6 scientific contributions have been obtained; 4 in international conferences with publications in the Springer series in Advances in Intelligent Systems and Computing, and another 2 contributions in the international impact journal Multimedia Tools and Applications (Springer, JCR Q2, Computer Science). In conclusion, the completion of this Thesis has left these lines of research open: (a) the development of a natural interaction system that integrates Microsoft Kinect Azure as a means of interaction; (b) the creation of an interaction system that is portable and can be used in various settings; (c) the development of a system that is self-adaptive in order

to adapt to the gestures of the characteristics of the users; (d) the development of a hybrid artificial intelligence system to offer an improved performance in gesture recognition and (e) the creation of a multimodal system that incorporates various modes of interaction, for example, gesture recognition and voice recognition.

Acknowledgments: TIN2017-83964-R, Study of a holistic approach for the interoperability and co-existence of dynamic systems: Implication in Smart Cities models.

Capítulo 1

Introducción

Capítulo 1

Introducción

Contenidos

El reconocimiento de gestos es un tema que está en auge en estos momentos donde el usuario puede interaccionar con un sistema computarizado, como por ejemplo un ordenador [Xu, 2017], una casa domótica [Zou et al., 2018b] o un automóvil [Tateno et al., 2019]. Este campo de conocimiento puede ser clasificado dependiendo de las características u objetivos (entre otros factores) que implica este tipo de reconocimiento. De acuerdo a la parte del cuerpo que es reconocido, se puede diferenciar entre mano [Li et al., 2019], cuerpo [Li et al., 2020b] y cara [Li and Zhang, 2019]. Atendiendo al tipo de gesto, si el reconocimiento implica un movimiento, entonces el gesto será dinámico, mientras que si se trata de una pose, el gesto se clasificará como estático [Zhang et al., 2020b]. Otra clasificación hace referencia al método usado para reconocer el gesto. Este reconocimiento puede ser clasificado en: reconocimiento de gestos basado en visión artificial [Sinha et al., 2019] o un reconocimiento basado en sensores [Kim et al., 2019].

En los últimos años, se han creado una serie de dispositivos para el reconocimiento de gestos tales como Microsoft Kinect [Li, 2012], Leap Motion [Lu et al., 2016], Intel RealSense [Liao et al., 2018], Myo [He et al., 2017], Camboard Pico [Zoghlami et al., 2019], entre otros. A la hora del reconocimiento de gestos corporales podemos distinguir entre gestos de motricidad fina (principalmente gestos realizados con los dedos de la mano) y gestos de motricidad gruesa (realizados con las extremidades del cuerpo). Por un lado, un ejemplo donde se integra motricidad gruesa sería Tolentino et al [Tolentino et al., 2019] que desarrollaron un sistema con el fin de detectar situaciones de emergencia mediante reconocimiento de gestos. Microsoft Kinect fue integrado para obtener la posición de las diferentes articulaciones en el cuerpo del usuario y conseguir información matemática como la distancia o el ángulo entre las distintas articulaciones para poder identificar 14 gestos. Por otro lado, un ejemplo de motricidad fina se explica en [Vaitkevičius et al., 2019], donde se reconocen gestos con las manos, a diferencia del trabajo anterior donde se reconocían gestos con el cuerpo, con el objetivo de identificar el lenguaje americano de signos. En este mismo estudio se aplica la combinación de Leap Motion y Modelos Ocultos de Markov (Hidden Markov Model, HMM) en un entorno de realidad virtual para clasificar 24 gestos. El dispositivo Leap Motion es esencial para extraer los datos 3D de las 11 articulaciones que dicho sensor es capaz de reconocer, siendo el objetivo de HMM clasificar los gestos por medio del vector de características, el cual contiene la distancia y ángulo de cada una de las articulaciones.

Sin embargo, el reconocimiento de gestos no es una técnica extraña para los usuarios debido al uso cotidiano de los smartphones. Estos dispositivos son usados para mirar el correo, navegar por internet, interactuar con las redes sociales, tomar fotografías o incluso jugar a juegos. Los smartphones se han llegado a convertir en una parte fundamental en la vida de los usuarios porque es capaz de realizar todas las acciones que podría hacer un ordenador pero con unas dimensiones muy reducidas que permite que puedan acompañar al usuario las 24 horas del día. Y la característica más destacada de estas herramientas es que se manejan mediante reconocimiento de gestos, desde el momento en el que el usuario toca el icono de una aplicación para abrirla hasta juntar los dedos para reducir el tamaño de una imagen, todos estos movimientos son gestos que las personas

realizan muchas veces al día y sin fijarse en ello. Además, estos dispositivos no son solo utilizados por adutos, sino que su inmersión en la vida diaria ha hecho que los niños sean capaces de usarlos también a una edad muy temprana y a realizar los gestos que están involucrados en su uso de una forma totalmente natural [Kılıç et al., 2019].

Por lo tanto, el reemplazo de las formas tradicionales de interacción por gestos no es una idea hipotética, aunque no ha sido implantado todavía en los sistemas informáticos al igual que en los dispositivos móviles debido a que la precisión y tiempo de respuesta no están optimizados para hacer sentir al usuario seguro y cómodo con la experiencia. Por consiguiente, existen áreas de conocimiento como la Inteligencia Artificial que han contribuido a mejorar estos parámetros con el propósito de que el reconocimiento de gestos sea el medio de interacción predominante.

Los campos de conocimiento de la Inteligencia Artificial denominados *Machine Learning* (ML) y *Deep Learning* (DL) han sido aplicados a numerosos campos científicos tales como Medicina [Ker et al., 2017], Procesamiento del Lenguaje Natural [Li, 2017], Recuperación de Información [Zhang et al., 2020a], Ciberseguridad [Apruzzese et al., 2018], Visión Artificial [Voulodimos et al., 2018], Internet de las cosas (Internet of Things, IoT) [Zantalis et al., 2019] y se han extendido a multitud de áreas. Entre ellas, se encuentra el reconocimiento de gestos para obtener mejores resultados puesto que, a pesar de que se han desarrollado modelos fiables en reconocimiento de gestos basados en visión artificial, el rendimiento necesario para su general uso todavía no se ha alcanzado.

En [Devineau et al., 2018] una nueva Red Neuronal Convolucional (Convolutional Neural Network, CNN) ha sido diseñada para reconocer gestos de las manos en 3D mediante la posición de las articulaciones de la mano sin usar la distancia de profundidad. La novedad de esta red neuronal descansa en el procesamiento paralelo y el uso de conexiones residuales para cada señal. Este enfoque obtuvo un 91.28 % de precisión en el dataset denominado Gestos de las Manos Dinámicos (Dynamic Hand Gesture,DHG)[1] que contiene una colección de gestos dinámicos. *deepGesture* [Kim et al., 2018] es un framework de *Deep Learning* que aplica unidades convolucionales y puertas recurrentes (Gated Recurrent Unit, GRU), alimentando la red neuronal recurrente (Recurrent Neural Network, RNN) con datos procedentes de giroscopios y un acelerómetros. Esta metodología consta de cuatro partes: la capa de entrada que recoge los datos de los sensores, las capas convolucionales extraen las características, las capas de GRU procesan la información secuencial y la capa totalmente conectada (fully connected) se encarga de calcular el resultado. La razón de usar este método es porque es más rápido y solo necesita una reducida cantidad de datos a diferencia de otros métodos como por ejemplo las redes de memoria largo-corto plazo (Long-short Term Memory, LSTM).

En [Li et al., 2020a] una ResNet 3D ha sido creada para extraer características espacio-temporales del conjunto de datos y junto con la combinación de un módulo de memoria ha sido posible la creación de un reconocimiento de gestos de aprendizaje de una sola vez. Asimismo, otra contribución de este trabajo es la creación de un conjunto de datos con 3045 vídeos de gestos de manos ya que a veces resulta complicado encontrar un conjunto de datos que encaje con las necesidades del problema a estudiar. En [Benalcázar et al., 2017] se describe un modelo para reconocer gestos con las manos a

[1]DHG dataset - `http://www-rech.telecom-lille.fr/DHGdataset/`

través de electromiografía (Electromyography, EMG) del antebrazo. Las señales EMG son producidas por el brazalete Myo como entrada de datos del sistema y los algoritmos K-vecinos más cercanos y la deformación dinámica del tiempo (Dynamic Time Warping, DTW) están involucradas en el proceso de clasificación. El objetivo del algoritmo de los K-vecinos más cercanos es estimar las probabilidades condicionales en la matriz de características mientras que el DTW es el encargado de calcular la función de distancia usando la distancia de Manhattan. La particularidad de este trabajo reside en el hecho de que los autores afirman que este modelo puede aprender cualquier gesto realizado con la mano mediante entrenamiento.

El hecho de que esta forma de interacción es más robusta y fiable será una necesidad en estos tiempos puesto que en la actualidad las ciudades se están convirtiendo Smart Cities, las cuales abren un mundo de posibilidades y facilidades a los usuarios con la inmensa cantidad de información que manejan. Entre esas posibilidades está que los usuarios pueden interaccionar con los sensores instalados en la propia calle, en un smart building [Al Dakheel et al., 2020] o en una smart classroom [Saini and Goel, 2019]. De hecho, los autores de [Ma et al., 2018] han desarrollado un algoritmo para reconocer los gestos de los policías cuando estos controlan el tráfico aplicando una CNN espacio-temporal. Esta propuesta ha sido creada en un entorno virtual geográfico que serviría para aplicarlo posteriormente a un ambiente real en una smart city.

Por último, la motivación de este trabajo es demostrar que se puede realizar el desarrollo de un sistema de interacción natural con la utilización de dispositivos no intrusivos como es el caso de una cámara. Las características de este sistema son: interacción natural, low-cost, adaptable y portable, para que los usuarios puedan utilizar un sistema mediante el reconocimiento de gestos, en lugar del teclado y ratón, pero sin tener que realizar una inversión económica elevada y que esta interacción se pudiera adaptar para alcanzar un mayor público, como personas de edad avanzada o con diversidad funcional entre otros colectivos.

Con esta hipótesis se han realizado diversos estudios donde se han utilizado ambas versiones de Microsoft Kinect para realizar la interacción natural debido a que este sensor tiene características convenientes para este tipo de interacción al disponer de una cámara de profunidad y una cámara RGB lo que permite reconocer las articulaciones del esqueleto y poder hacer una detección del movimiento o reconocimiento de gestos del cuerpo entero.

En primer lugar, se utilizó la primera versión de este dispositivo para probar si en realidad se podía usar para el propósito de este trabajo creando una serie de actividades para un centro de educación especial donde se hicieron experimentos sobre este desarrollo con estudiantes con necesidades especiales. Una vez comprobado que esta herramienta ofrecía diversas opciones se decidió utilizar la segunda versión para la creación de un sistema que fuera más adaptable para el usuario donde los experimentos fueron realizados también en el centro de eduación especial Princesa Sofía. En el último tramo de la investigación se optó por utilizar una webcam como dispositivo debido a que no estaba clara la continuidad de Kinect y la paralización en la distribución de los dos modelos que Microsoft había fabricado hasta el momento. En esta fase se decidió hacer uso de transferencia de aprendizaje con *Deep Learning* para probar el reconocimiento de gestos con las manos donde se probaron diferentes gestos y diferentes configuraciones para rea-

lizar un total de 104 experimentos. A este proceso se integró también un módulo basado en lógica difusa para tener conocimiento de las mejores configuraciones probadas para el reconocimiento de gestos. Los detalles de los procedimientos ejecutados se describirán en los siguientes capítulos.

1.1. JUSTIFICACIÓN

En la actualidad, la mayor parte de la población dispone de algún equipo informático (dispositivo móvil, ordenador personal o portátil) o lo utiliza en su puesto de trabajo. Según avanza el tiempo, los recursos se están digitalizando y ahora las facturas bancarias se comprueban de manera online, en vez de por correo ordinario o se lee el sitio web del periódico en lugar de comprarlo en el quiosco. El acceso a esta información se realiza fundamentalmente a través de una interfaz de usuario y las compañías no escatiman en recursos para que dicha interfaz sea usable y accesible para sus usuarios. Uno de los objetivos principales por el cual estas interfaces tienen estas características es para ofrecer una experiencia agradable al usuario y evitar la frustración. En este sentido, el tiempo de respuesta es crucial y se persigue hacer el procedimiento simple para que el usuario no pierda tiempo. Normalmente, la interacción con los distintos elementos de la interfaz de usuario se realiza mediante el ratón y teclado, o de forma táctil. Sin embargo, si esta interacción se pudiera hacer con gestos o con la voz sería más intuitivo, natural, sencillo y rápido.

En los últimos años se ha intentado incorporar otros medios de interacción a los equipos informáticos, basándose principalmente en interacción natural, y de este modo dejar de usar los tradicionales recursos, aunque este propósito todavía no se ha conseguido. Una parte considerable de estos nuevos medios de interacción se ha basado en el funcionamiento de las cámaras de tiempo de vuelo para realizar su reconocimiento de gestos. Las cámaras de tiempo de vuelo son cámaras que permiten obtener información 3D de nuestro entorno, escaneando la habitación y con ayuda de la tecnología de infrarrojos somos capaces de saber la profundidad a la que se encuentran los distintos elementos del escenario de actuación. Esta información 3D es muy útil para realizar el reconocimiento de gestos, pero la principal desventaja es que este tipo de dispositivo no es económicamente asequible para un usuario normal, puesto que el precio de una cámara modesta de este estilo es aproximadamente 2186.47€[2]. Sin embargo, existen dispositivos, como Microsoft Kinect, que simulan este tipo de tecnología incorporando un proyector y un receptor de infrarrojos. De esta forma, el proyector emite una malla de puntos infrarrojos y cuándo colisionan con algún elemento, éstos rebotan y el receptor los capta, determinando de esta forma la distancia a la que se encuentra dicho elemento. Este tipo de dispositivo está al alcance de más usuarios porque el precio de la última versión de Microsoft Kinect cuesta aproximadamente 149.99€[3].

El dispositivo Microsoft Kinect puede ser una buena opción desde el punto de vista del desarrollador que quiere incorporar reconocimiento de gestos a su sistema, porque

[2]Time of Flight Camera - https://machinevisionstore.com/Catalog/Details/1607

[3]Disponibilidad y precio del Kinect 2.0 para Windows - https://www.muycomputer.com/2014/07/07/kinect-2-0-para-windows-precio/

se puede descargar el Software Development Kit (SDK)[4] que aprovecha los sensores de infrarrojos para obtener la posición tridimensional de ciertos puntos (*joints*) que detecta a lo largo del cuerpo del usuario y serán la información fundamental para reconocer gestos de una manera más sencilla que si se utilizara técnicas de Visión Artificial. Además, el sensor de profundidad integrado en este dispositivo permite conocer la distancia a la que el usuario está situado respecto del dispositivo, así como de los elementos que se encuentran en el entorno para mejorar la interacción. Este dispositivo dispone de dos versiones, ofreciendo la segunda versión mejoras significativas como un mayor número de *joints* reconocidos, mayor resolución de las cámaras de color y profundidad y el reconocimiento de un número mayor de usuarios. Aunque este SDK solo permite hacer desarrollos en sistemas operativos Microsoft Windows, la comunidad ha desarrollado otros software como OpenNI[5] para poder crear aplicaciones en otras plataformas.

Desde el punto de vista de un usuario este dispositivo permite realizar motricidad gruesa: este concepto hace referencia a que permite detectar el movimiento en grandes grupos musculares de nuestro cuerpo, como piernas o brazos, pero no es capaz de detectar movimientos precisos que se realizan con los dedos de las manos. En ocasiones será más cómodo para el usuario un tipo de interacción más preciso para controlar la interfaz del sistema, pero ya sería necesario otro dispositivo como Ultraleap Stereo IR 170, que cuesta 256.37€[6]. La principal desventaja que tiene el uso de estos dispositivos es que para usar el proceso de reconocimiento de gestos exige al usuario tener que comprar alguno de estos dispositivos que no son comunes en el ambiente doméstico. Sin embargo, una cámara web la incluyen todos los portátiles y suele ser una herramienta habitual para hacer videoconferencias. Aunque si no se dispone alguna, el precio suele ser bastante inferior, dependiendo del modelo elegido: un ejemplo puede ser la cámara Logitech C270[7] con un precio de 26.92€. La finalidad de este trabajo de investigación es que sea un sistema de bajo coste y que pueda ser accesible para el mayor número de usuarios posibles, por esta razón se va a utilizar el dispositivo Kinect, que ofrece unas funcionalidades realmente útiles para la interacción natural a un precio muy asequible, y una cámara estándar que no supondrá un coste adicional para la mayoría de usuarios.

Por otro lado, otro de los objetivos que se establece en este trabajo es el hecho de conseguir una experiencia de usuario totalmente natural, y para alcanzar este fin, el sistema tiene que ser adaptable. Un sistema adaptable es el proceso mediante el cual un sistema interactivo adapta su comportamiento al usuario sabiendo de antemano información relevante de éste, del contexto de uso y del entorno. La información más relevante para este tipo de sistemas es el modelo de usuario que es el elemento que contiene la información relativa al usuario.

El sistema adaptable está enfocado a la interacción del usuario, ya que el objetivo principal es que esta sea la más adecuada para el usuario. Este propósito hace que solo se extraigan las características del usuario que intervienen en el proceso de interacción con el sistema. El hecho de saber las características del usuario permitirá que el sistema además de adaptable sea accesible.

[4]SDK Microsoft Kinect - `https://www.microsoft.com/en-us/download/details.aspx?id=44561`
[5]OpenNI – `http://openni.ru/index.html`
[6]Ultraleap Stereo IR 170 - `https://www.ultraleap.com/product/stereo-ir-170/`
[7]Logitech HD Webcam C270 – `https://www.pccomponentes.com/logitech-hd-webcam-c270`

El diseño de un sistema adaptable es muy útil para controlar un sistema informático, pero carece de interés si este sistema solo puede ser utilizado con ciertas restricciones (un equipo muy potente) o limitado a ciertos ambientes. Este planteamiento ha conducido a la idea de desarrollar un sistema que también sea portable. Esta portabilidad se traduce en embeber este sistema en un módulo para que pueda ser utilizado en distintos ambientes, como una casa domótica, un coche, en la Raspberry Pi, un quirófano o una clase.

La justificación de esta tesis se fundamenta en que el reconocimiento de gestos no ofrece una experiencia óptima para el individuo debido a que el tiempo de respuesta es demasiado alto, no reconoce el gesto o reconoce otro gesto distinto al que se está haciendo. Por estas razones, no se observa que los equipos informáticos vengan integrados con un sistema de reconocimiento de gestos como alternativa de interacción o que compañías importantes integren este tipo de sistemas cuando diseñen su software. La prueba más evidente es que cuando se ve a las personas trabajando o utilizando su ordenador en casa no lo controlan con gestos, sino que vemos el mismo sistema de interacción que se utilizaba en décadas anteriores. Por estos motivos es necesario el diseño de un proceso de reconocimiento de gestos que sea robusto, fluido, óptimo y preciso, junto con un sistema adaptable que tenga en consideración las características del usuario durante el proceso de interacción con el sistema. La creación de un sistema de bajo coste y portable facilitará la labor de integrar el proceso de reconocimiento de gestos en diversos ámbitos y que cada vez se vaya viendo como un proceso más natural este tipo de interacción para controlar sistemas tecnológicos.

1.2. Hipótesis

Con esta tesis doctoral se quiere demostrar que se puede desarrollar un sistema de reconocimiento de gestos natural que pueda ser utilizado por usuarios con diferentes características, sin necesidad de ningún dispositivo intrusivo, siendo una cámara el único dispositivo externo que se va a requerir en el proceso de reconocimiento. La finalidad es que este reconocimiento de gestos sea un medio de interacción accesible y natural para el usuario, mejorando los dispositivos actuales como son el teclado, el ratón o las pantallas táctiles, entre otros. Este tipo de interacción tiene la ventaja de ser personalizable para distintos tipos de usuarios, lo que permite ser utilizado por personas con discapacidad física, que están excluidas cuando los dispositivos de entrada del sistema informático se rigen por un ratón y teclado convencional.

El sistema será accesible para la mayoría de los usuarios porque se va a diseñar un sistema de bajo coste y adaptable. El sistema se define como de bajo coste porque se ha decidido hacer el reconocimiento de gestos con dispositivos ecónomicos o de uso común para el usuario, y desarrollar los algoritmos necesarios para realizar este tipo de reconocimiento. De esta forma, el usuario no tendrá que realizar una inversión sustancial para usar el medio de interacción desarrollado. El sistema será adaptable, mejorando el proceso de reconocimiento de gestos e incorporando nuevos perfiles de usuarios.

Este sistema adaptable mejorará el proceso de reconocimiento porque dependiendo de las características del usuario un gesto asociado se podrá realizar de diversas formas.

En una fase inicial del proceso será necesario definir un modelo de usuario que contenga la información útil del mismo para este proceso y que puede ir variando con el tiempo según el comportamiento del usuario con el sistema. Este sistema utilizará algoritmos de *Machine Learning* para refinar el gesto del usuario porque un simple gesto como levantar el brazo, puede ser realizado con distintas variantes dependiendo del individuo. La inclusión de estas técnicas hará que el sistema vaya aprendiendo de la interacción del usuario conforme se vaya utilizando y de esta manera los gestos se irán reconociendo con más fluidez y más precisión, adaptándose a las particularidades del usuario.

1.3. Objetivos

El objetivo principal de esta tesis doctoral consiste en la realización de un sistema de interacción natural que se caracterizará por ser de bajo coste, portable, inteligente y adaptable.

Los objetivos específicos necesarios para cumplir el propósito de esta tesis fueron:

- Realizar un sistema con interacción natural. Este sistema debería estar compuesto de algún dispositivo que permitiera realizar el reconocimiento de gestos y que fuera asequible desde el aspecto económico para el usuario, para así cubrir el aspecto de low-cost de este trabajo.

- Diseñar un sistema adaptable. Este proceso tiene que ser flexible para que pueda ser utilizado por el mayor número de usuarios. Este objetivo comprenderá la obtención de datos del usuario que sean relevantes para el proceso de interacción con el sistema y la elaboración de un modelo de usuario que se encargará de definir esa adaptabilidad a partir de las características reunidas incialmente.

- Diseñar un sistema inteligente. La inclusión de técnicas de Inteligencia Artificial, como *Machine Learning* y *Deep Learning*, proporcionará cierta autonomía al sistema debido a que son capaces de aprender a partir de un conjunto de datos de entrada. *Deep Learning* es incluso más automático que *Machine Learning* porque no es necesario alimentarlo con características manualmente ya que es capaz de obtenerlas a partir de los datos a expensas de que el volumen de datos proporcionado tiene que ser considerablemente mayor que en *Machine Learning*.

- Diseñar un sistema portable. Se propone crear un sistema que permita trasladar el reconocimiento de gestos a diferentes escenarios como por ejemplo, una casa inteligente, un coche o un ordenador de placa reducida como es el caso de Raspberry Pi o Jetson Nano. Este objetivo requiere la creación de un módulo para que sea fácil utilizar este procedimiento en cualquier ambiente que sea válido para incluir un proceso de reconocimiento gestual y un procedimiento que tenga la capacidad de asociar la interacción a una acción del propio sistema.

1.4. Etapas del trabajo de investigación

Esta tesis doctoral se descompone en cuatro fases que se describen en los siguientes capítulos de este trabajo de investigación.

(a) **Estudio preliminar**: en primer lugar, se valoró el planteamiento inicial, los objetivos que se pretendían cubrir y se consultaron las propuestas de otros estudios similares. En este caso en particular, debido a que en algunas etapas de la investigación se había utilizado el dispositivo de entrada Microsoft Kinect (independientemente de si es la primera o la segunda versión) se comprobaron manuales y referencias para tener conocimiento sobre cómo programar con este sensor. Además, se buscó información sobre las características de las técnicas que se implementaron posteriormente.

(b) **Diseño/Desarrollo**: en este apartado se pensó en la arquitectura del sistema, la funcionalidad que iban a tener los distintos módulos y los flujos de información que iban a discurrir entre ellos. Una vez que estaban claros los elementos necesarios para el desarrollo del sistema se procedió a su implementación donde se desarrollaron varios prototipos para comprobar que la funcionalidad era correcta.

(c) **Experimentos**: después de finalizar con el desarrollo del sistema correspondiente se realizaron una serie de experimentos, que dependiendo de la fase se utilizaron diferentes métodos de evaluación y valorado distintas métricas. El estudio de los métodos y métricas a evaluar y la posterior ejecución de los experimentos fue lo que forma esta tercera etapa de la metodología propuesta.

(d) **Conclusiones**: de acuerdo a los resultados obtenidos en el procedimiento anterior se expusieron las impresionas y las ideas que transmitieron este proceso de experimentación. Además, en esta etapa se verificó que la solución propuesta resolvía el problema planteado incialmente.

1.5. Contribución

El hecho de crear un sistema de interacción natural no es trivial. Existen infinidad de decisiones que son imprescindibles tomar, desde el tipo de interacción que se quiere desarrollar hasta los tipos de experimentos que se quieren llevar a cabo para que se ajuste de la mejor forma posible a la realidad. En todos los campos siempre se encuentran desafíos a superar, pero en el estudio de trabajos que se centran especialmente en el usuario tenemos el reto adicional de que se debería tener en cuenta las características de los mismos y que se podría encontrar una gran variedad de casuísticas dependiendo del usuario que lo utilice. Estas características no solo afecta a un punto determinado del estudio sino que hay que tenerlas en cuenta desde el diseño hasta la evaluación de la metodología.

En este trabajo se ha optado por integrar un tipo de interacción natural que comprende la detección del cuerpo y de las manos. Se ha considerado una de las formas más intuitivas para controlar un sistema informático. En base a esta decisión se decidió

incluir una cámara RGB con profundidad. Dentro de las cámaras de este estilo que se encuentran en el mercado, se eligió Microsoft Kinect como dispositivo de interacción para el sistema debido a su relación calidad/precio. En dicho trabajo se pueden observar tres escenarios diferentes:

(a) En el primer escenario se utilizó Microsoft Kinect v1 que permitía detectar 20 joints y de esta forma reconocer el movimiento del cuerpo de los usuarios. Se desarrollaron un conjunto de actividades donde se hacía uso del sensor para interaccionar con ellas y se realizaron una serie de experimentos con estudiantes con necesidades especiales en un centro de educación especial.

(b) En el segundo escenario se incorporó la segunda versión de Microsoft Kinect, la cual ofrecía mejoras como por ejemplo, la detección de 25 joints en todo el cuerpo. En esta fase el objetivo primordial era la adaptación de la interacción al usuario y por esta razón se diseño un modelo que optimizaba las características del dispositivo de interacción al que se denominó *modelo dispositivo-interacción*.

(c) En el último escenario se utilizaron diversos modelos de *Deep Learning* para probar distintas configuraciones y comprobar la que mejor rendimiento ofrecía en el ámbito de reconocimiento de gestos. De este modo, con ayuda de un sistema experto de lógica difusa se pueden saber las mejores configuraciones para el conjunto de datos de entrada dado y que los usuarios que pretendan utilizar reconocimiento de gestos con los modelos propuestos dispongan de un punto de partida y así no probar aleatoriamente.

Las contribuciones de esta tesis doctoral son:

(a) El desarrollo de un sistema que puede ser integrado en una clase de educación especial con estudiantes que poseen diferentes tipos de discapacidad (visual, auditiva, física y autismo) y permite al usuario controlarlo de manera autónoma sin la intervención de otros factores como compañeros de clase, profesores, entre otros.

(b) La implementación algoritmo que es capaz de detectar solamente al usuario que utiliza la aplicación excluyendo al resto de personas que se encuentran alrededor de él/ella. Para la consecución de este objetivo, el algoritmo puede detectar los joints del usuario que se encuentra más próximo y filtrar los joints que interaccionan con el dispositvo para mejorar el reconocimiento y la usabilidad.

(c) El diseño de un modelo dispositivo-interacción que se centra en el dispositivo y tiene en cuenta las características del usuario con diversidad funcional, a partir del modelo de usuario, para adaptar la interacción del individuo en el sistema.

(d) La creación de un sistema experto de lógica difusa para la selección del modelo de *Deep Learning* más apto para el reconocimiento de gestos con las manos y la realización de un número determinado de experimentos con diferentes modelos de redes neuronales convolucionales que proporcionan un punto de partida para aquellos investigadores o desarrolladores que estén interesados en aplicar el reconocimiento de gestos haciendo uso de *Deep Learning* pero no dispongan de conocimientos avanzados en la materia.

1.6. Estructura de la Tesis

Esta tesis doctoral se compone de siete capítulos, los cuales se van a describir brevemente a continuación:

El Capítulo 2 constituye el Estado del arte de este trabajo, el cual presenta los tipos más importantes de interacción natural: Reconocimiento de gestos, interacción táctil, reconocimiento de voz y la interfaz cerebro-ordenador. En estos apartados se expondrán los fundamentos de cada tipo de interacción, así como las técnicas más utilizadas e investigaciones relevantes que incluyen estos tipos de interacción en diferentes ámbitos dependiendo de dicha interacción.

El Capítulo 3 detalla un análisis bibliométrico en el cualo se han aplicado palabras clave relacionadas con el reconocimiento de gestos de diferentes partes del cuerpo y términos pertenecientes a Inteligencia Articial. Estas palabras claves han sido de utilidad para realizar una búsqueda en dos de las más destacadas bases de datos a nivel científico (Scopus y Web of Science) con el fin de obtener resultados de interés como pr ejemplo, los diez artículos más citados o los países con mayor producción científica en estos temas.

El Capítulo 4 describe el proceso seguido para la creación de un sistema que utiliza el dispositivo Microsoft Kinect v1 como medio de interacción natural. Este sistema fue evaluado en un centro de educación especial con el fin de que los estudiantes con necesidades especiales fueran capaces de mejorar sus habilidades físicas mediante la interacción realizada con el movimiento de su cuerpo.

El Capítulo 5 explica el estudio realizado con el objetivo de diseñar un modelo que ayude a adaptar la interacción de los usuarios. En este caso, también se realizó la evaluación en el centro de educación especial y se contó con la colaboración de los profesores del centro para el diseño de las actividades.

El Capítulo 6 presenta la investigación realizada con *Deep Learning* para el reconocimiento de gestos de las manos, utilizando redes neuronales convolucionales preentrenadas. Estas redes neuronales han sido entrenadas con los vídeos de una base de datos de gestos con las manos, donde se han elegido vídeos de hasta 10 diferentes tipos de gestos. Los parámetros de los entrenamientos han sido modificados para obtener diferentes configuraciones y comprobar cuáles de ellas obtenían un mejor rendimiento en el proceso de reconocimiento. Además, se ha diseñado un sistema experto de lógica difusa para obtener una clasificación de las configuraciones que habían obtenido mejores resultados y ordenarlas por este criterio mediante las reglas de inferencia implementadas por expertos.

El Capítulo 7 presenta las conclusiones que se han obtenido después de la consecución de cada una de las fases descritas en los anteriores capítulos y enuncia las líneas de investigación futuras derivadas de la investigación realizada en esta tesis doctoral.

Capítulo 2

Tecnologías

Capítulo 2

TECNOLOGÍAS

Contenidos

En el presente capítulo se van a describir los diferentes tipos de interacción natural, con especial detalle en el reconocimiento de gestos puesto que es el tipo de interacción que se desarrolla en este proyecto. En la actualidad se encuentran una serie de dispositivos que ayudan en este proceso. Estos dispositivos se han descrito junto a las técnicas más usadas en este campo de conocimiento porque son la base para constituir este tipo de reconocimiento, especialmente si se trata de un enfoque basado en visión. Además, se incluyen diversas técnicas enfocadas en Inteligencia Artificial que han sido necesarias para la finalización de este trabajo.

2.1. Reconocimiento de gestos

El reconocimiento de gestos se basa en el estudio de los movimientos que realiza el usuario para compararlo con una serie de patrones con el fin de asociarlo a un gesto. Estos gestos pueden ser reconocidos por diferentes partes del cuerpo, sin embargo,la mano es la que tiene especial relevancia. El reconocimiento de gestos con las manos se puede clasificar en reconocimiento basado en guante y basado en visión. El reconocimiento de gestos que está basado en la utilización de un guante detecta gestos mediante la trayectoria creada por los dedos, mientras que el que está basado en visión analiza las imágenes de una cámara para determinar si el usuario ha realizado un gesto.

2.1.1. Dispositivos

En esta sección se van a describir las características de los dispositivos que se aplican al reconocimiento de gestos: Microsoft Kinect, Leap Motion, Intel RealSense, Myo, Camboard Pico, ZED stereo camera, Gloveone y Tobbi Pro Glasses 2.

2.1.1.1. Kinect

Microsoft Kinect[1] es un dispositivo que se creó para reconocer el movimiento del cuerpo del usuario [Zhang, 2012] y ofrecer un modo de interacción más dinámico en el área de entretenimiento (ver Figura 2.1). Sin embargo, una vez que fue liberado su SDK, los desarrolladores e investigadores empezaron a aplicarlo a otras disciplinas como rehabilitación [Tseng et al., 2014], educación [Boutsika, 2014], medicina [Edmunds and Donovan, 2016], robótica [Velayudhan and Gireeshkumar, 2015] y realidad aumentada [Meng et al., 2013]. Este sensor está formado por los siguientes componentes: una cámara RGB, un proyector, un receptor de infrarrojos (sensores de profundidad) y una serie de micrófonos distribuidos a lo largo de este dispositivo. Los sensores de profundidad son los que detectan la distancia a la que se encuentra el usuario y junto con la característica de su SDK denominada "*Skeletal tracking*" es capaz de reconocer hasta 20 posiciones (*joints*)

[1]Microsoft Kinect Developer - https://developer.microsoft.com/en-us/windows/kinect

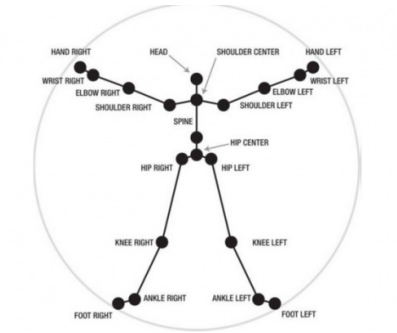
del cuerpo del usuario con Kinect v1 (ver Figura 2.2a) y 25 con Kinect v2 (ver Figura 2.2).

(a) Microsoft Kinect v1. Fuente: [kin, 2021]

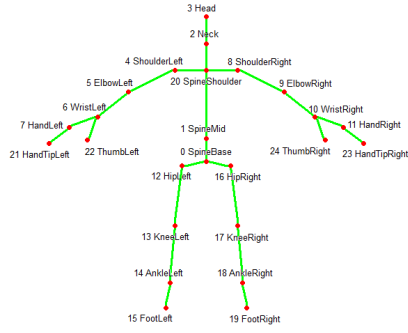
(b) Microsoft Kinect v2. Fuente: [kin, 2018]

Figura 2.1: Las dos versiones de Microsoft Kinect.

Este sensor es capaz de detectar el movimiento de grandes grupos musculares gracias a la distribución de los distintos *joints*, motivo por el que Kinect trabaja con motricidad gruesa. El hecho de conocer la posición de estos *joints* permite que se pueda realizar el reconocimiento de gestos designando uno de los *joints* como base y analizando la posición de los otros *joints* respecto a ese *joint* para detectar un determinado gesto. Además, los micrófonos ofrecen la posibilidad de realizar reconocimiento de voz.



(a) *Joints* reconocidos en Kinect v1. Fuente: [Martinovikj et al., 2013]



(b) *Joints* reconocidos en Kinect v2. Fuente: [Ousmer et al., 2019]

Figura 2.2: Los *joints* reconocidos en ambas versiones de Kinect.

Este dispositivo ha sido aplicado para la creación de un sistema de visualización sin contacto para la cirugía asistida por ordenador [Liu et al., 2017]. Este sistema está formado por dos módulos: el módulo de visualización y el de interacción. En el módulo de visualización se pueden observar los modelos anatómicos del paciente mientras que el módulo de interacción es el encargado de controlar el módulo anterior, utilizando 9

gestos con la mano. Estos gestos son reconocibles gracias a la información de profundidad y de los *joints* que es capaz de reconocer Kinect a lo largo del cuerpo. Esta información es usada para generar la imagen de profundidad de la mano, que será fundamental para extraer el histograma de grandientes orientados (Histogram of Oriented Gradients, HOG) y estas características se usarán para el reconocimiento de gestos. A continuación, se utiliza el Análisis de Componentes Principales (Principal Component Analysis, PCA) con el objetivo de reducir la dimensionalidad de las características de HOG, ya que la dimensión de este vector es demasiado grande para trabajar en un sistema a tiempo real. Por último, el clasificador de Máquinas de Soporte Vectorial (Support Vector Machine, SVM) es entrenado con las características de HOG para identificar el gesto.

En [Lee et al., 2020b] se ha diseñado un sistema interactivo basado en 8 gestos para controlar una pantalla holográfica mediante Kinect Azure. Las características de este dispositivo lo hacen idóneo para el desarrollo de este trabajo debido a que esta pantalla necesita estar en un ambiente totalmente oscuro y los datos de profundidad proporcionados por Kinect Azure permiten trabajar en estas condiciones. El proceso para el reconocimiento de gestos se basa en realizar primero un *background subtraction* para eliminar el fondo y se establece un ROI estático para eliminar el ruido que ha dejado la técnica de *background subtraction*. El siguiente paso es aplicar una arquitectura de la red de regresión de poses denominada *CrossInfoNet* [Du et al., 2019] para obtener información de la posición de 14 *joints* de la mano, incluida el centro de la palma de la mano, dato que será crucial para dibujar un *bounding box* alrededor de la misma. En último lugar, el reconocimiento de gestos se realiza a partir de la información obtenida en los procesos anteriores para calcular medidas como la distancia entre joints y de esta forma detectar los distintos gestos.

2.1.1.2. Leap Motion

El sensor Leap Motion[2] (ver Figura 2.3), es capaz de detectar las manos y reconocer sus gestos [Weichert et al., 2013]. Este dispositivo consta principalmente de dos cámaras y tres LEDs. Los LEDs se encargan de proyectar una luz infrarroja en su área de cobertura y cuando ilumina la mano se produce una reflexión que llega hasta las lentes de las cámaras. Los datos obtenidos se almacenan en la memoria del controlador USB. Este sensor sólo es capaz de detectar el movimiento de las manos, a diferencia del dispositivo Kinect, por esta razón con este sensor se trabaja la motricidad fina. Leap Motion permite realizar reconocimiento de gestos con las manos, como por ejemplo el reconocimiento del lenguaje de signos Australiano [Potter et al., 2013].

Leap Motion ha sido utilizado para el diagnóstico y seguimiento en pacientes que sufren la enfermedad de Parkinson, así como para evaluar la disfunción motora causada por esta patología [Butt et al., 2018]. El primer paso de este proceso fue obtener las características que podían ser útiles para el estudio y a continuación seleccionar las características más relevantes para los clasificadores que iban a ser utilizados posteriormente. Estos clasificadores basados en ML tienen el objetivo de diferenciar los usuarios que tenían la enfermedad de Parkinson y los que estaban sanos.

[2]Leap Motion - https://www.leapmotion.com/

En [Mittal et al., 2019]se ha diseñado un framework destinado al reconocimiento del lenguaje de signos usando Leap Motion ya que permite extraer los datos 3D tanto de los movimientos de los dedos como de la mano. La interpretación del lenguaje de signos con los gestos de la mano se realiza mediante DL, donde la CNN tiene la función de extraer las características espaciales, que alimentarán la red LSTM, la cual se encarga del reconocimiento de signos consecutivos.

Figura 2.3: El sensor Leap Motion. Fuente: [lea, 2021]

2.1.1.3. Intel RealSense

La cámara Intel RealSense[3] (ver Figura 2.4), está compuesta por una cámara RGB, dos cámaras de infrarrojos y un proyector láser de infrarrojos. Esta cámara permite realizar reconocimiento de gestos, reconocimiento de voz, reconocimiento facial y escaneo en 3D [Draelos et al., 2015].

Este dispositivo funciona de forma parecida a Microsoft Kinect ya que está compuesto por un proyector láser de infrarrojos y la cámara de profundidad. Sin embargo, Intel RealSense realiza un reconocimiento de gestos basado en motricidad fina.

Figura 2.4: La cámara Intel RealSense. Fuente: [Małecki et al., 2020]

En [Bayer and Faigl, 2019] se utiliza la cámara Intel RealSense para estimar la posición de un robot hexápodo cuando explora autónomamente su entorno. Este sistema se componen principalmente de tres módulos: módulo de mapeo, módulo de exploración y módulo de seguimiento de la trayectoria. El módulo de mapeo es el que está más relacionado con el sensor ya que se basa en los point clouds y otros datos para realizar un mapeo del entorno del robot. El módulo de exploración usa el mapa creado por el módulo anterior y la posición del robot para explorar localizaciones desconocidas en el terreno y fijar objetivos en el mapa. Por último, el módulo de trayectoria tiene la función de planificar un camino según el objetivo y de asegurar que el robot se dirige hacia el destino correcto.

En el mundo de la agricultura se ha diseñado un algoritmo que usa este sensor para el reconocimiento de frutas cítricas [Liu et al., 2018]. Este algoritmo aprovecha la diferencia

[3]Intel RealSense - https://software.intel.com/en-us/realsense/sr300

en la forma geométrica de la fruta (esfera) y la hoja (plano) en la dimensión 3D para realizar una intersección entre estos elementos y una superficie esférica, con el fin de obtener la curva de intersección que va a ser la responsable de obtener la información necesaria para hacer la distinción entre la fruta y sus hojas. Los resultados afirman que esta técnica es rápida debido a su sencillez, a la vez que fiable ya que obtuvo una tasa de acierto entre el 80 % y el 100 %.

2.1.1.4. Myo

El brazalete Myo[4] (ver Figura 2.5) interpreta los impulsos eléctricos generados por los movimientos musculares del brazo. Esta característica tiene la ventaja sobre otros dispositivos de esta índole que no necesita usar una cámara, con lo que se reduce espacio en el dispositivo y no tiene problemas con otros factores, como la cantidad de exposición de luz. Este dispositivo es capaz de distinguir los movimientos de los dedos, así como la rotación y el movimiento de la mano, midiendo los patrones de impulsos eléctricos que generan los movimientos. La información generada por los gestos se envía a un procesador, y un algoritmo la traduce en comandos que se envían mediante Bluetooth. Un ejemplo de uso de este dispositivo ha sido controlar las luces de una casa [Burmeister et al., 2016].

Figura 2.5: El brazalete Myo. Fuente: [myo, 2021]

Algunas de las aplicaciones donde se ha aplicado este brazalete ha sido en la domótica, donde Myo ha sido integrado en un sistema inteligente para reconocer una serie de gestos y poder actuar en el control de un ambiente domótico [Luna-Romero et al., 2017]. Este trabajo está orientado al control de los elementos de una casa domótica, los cuales se comunican mediante la tecnología *ZigBee*. El brazalete Myo funciona con señales de electromiografía procedentes de los movimentos del usuario con el brazo, las cuales son entrenadas en una Red Neuronal para realizar la clasificación de los gestos con una precisión del 83.33 %.

Además, este dispositivo ha sido utilizado también con niños que presentan algún tipo de discapacidad en las extremidades superiores y tienen dificultades para interaccionar con juegos [Fernandes et al., 2020]. El objetivo es desarrollar un juego de puzle para demostrar que los usuarios con discapacidad en las extremidades superiores pueden

[4]Myo - https://www.myo.com/

interactuar con aplicaciones de entretenimiento digital con el uso del dispositivo correcto. Este juego incluye el brazalete Myo para que los niños puedan completar el juego mediante el reconocimiento de gestos, situación que no sería posible con los medios de interacción convencionales. Los resultados de este estudio concluyen que este sensor es apto para ofrecer accesibilidad a las personas con discapacidad física en sus extremidades superiores en el uso de aplicaciones de entretenimiento digital y aprovechar los beneficios que ofrecen este tipo de aplicaciones como son el aprendizaje de nuevas tecnologías o la socialización con otros usuarios.

2.1.1.5. Camboard pico

Camboard pico [Camboard Pico, 2016] (ver Figura 2.6), es una cámara de tiempo de vuelo [Steich et al., 2016] fabricada por la compañía PMD Technologies[5]. Sus circuitos SBI (*Suppression of Backgound Illumination*) ofrecen la principal ventaja frente a otras cámaras de este tipo, que funcionan perfectamente tanto a plena luz del día como en la absoluta oscuridad. Los datos que se obtienen de tiempo de vuelo originan una nube de puntos que permite reconocer los gestos del usuario cuando mueve las manos alrededor de esta nube de puntos. Esta cámara permite realizar reconocimiento de gestos con motricidad fina a una distancia máxima de 50 cm del dispositivo.

Figura 2.6: La cámara Camboard pico. Fuente: [Giancola et al., 2018]

Esta cámara ha sido integrada en un dron para inspeccionar las cavidades de los árboles de una manera más segura y más eficiente [Steich et al., 2016]. Las funciones principales de este trabajo son la creación de un sistema de visión para la detección de las cavidades en los árboles y una estrategia que permita un control preciso del dron para que pueda ser controlado de manera semiautomática. Los resultados demuestran que el sistema propuesto es capaz de detectar las cavidades de los árboles con una intervención mínima por parte del usuario.

En este trabajo, en lugar de utilizar el sensor Camboard pico para tareas aéreas, se va a trasladar en sentido opuesto y se ha aplicado para detectar defectos en el sistema de alcantarillado [Haurum et al., 2021]. Este proceso aplica dos métodos de DL basado en geometría para detectar esas fallas: *PointNet* y Dynamic Graph CNN (DGCNN). El procedimiento consistió en utilizar la red PointNet como punto de referencia para después aplicar los datos de entrada con DGCNN, y comprobar si estos métodos son efectivos en el contexto de este trabajo. Los datos de entrada que se utilizaron para entrenar las técnicas mencionadas fueron *cloud points* obtenidos del dispositivo Camboard pico sobre tuberías de alcantarillado ya que normalmente los dataset que existen

[5]PMD Technologies - http://pmdtec.com/

contienen información 2D y era necesario los datos en 3D para alimentar las redes de DL aplicadas en este estudio. Finalmente, los experimentos determinaron que para esta tarea en concreto la red DGCNN obtuvo mejores resultados frente a PointNet.

2.1.1.6. ZED stereo camera

ZED stereo camera[6] (ver Figura 2.7) es una cámara con visión esteoroscópica, lo que permite obtener información 3D del entorno con ayuda de las dos cámaras que tiene integradas. Esta cámara de alta resolución permite captar vídeo en 2K y ser utilizada tanto en interior como en exterior. Una característica interesante es que tiene soporte para OpenCV, el motor de juegos Unity3D y Matlab con lo que esta herramienta puede ser muy útil para temas de Visión Artificial y videojuegos. En [Jamaluddin et al., 2016] se describe cómo ha sido utilizada este tipo de camára para crear una nueva cámara junto con dos lentes de ojo de pez. La información suministrada por esta composición híbrida permite realizar reconstrucciones en 3D, así como detectar objetos y realizar tracking. Este dispositivo es potencialmente útil en robótica y videovigilancia.

Figura 2.7: La cámara con visión esteoroscópica Zed. Fuente: [zed, 2021]

La cámara ZED ha sido usada para la detección de obstáculos en drones y de esta forma evitar daños en él mismo o sus componentes [Pérez Gutiérrez and Córdova-Cruzatty, 2020]. El objetivo principal es el desarrollo de un sistema de pilotaje inteligente que sea capaz de evitar colisiones con los elementos que puedan surgir en la trayectoria. Para este fin, la cámara ZED cumple una papel fundamental ya que al ser una cámara con visión estereoscópica funciona de manera similar a como lo hace el ojo humano y permite percibir la profundidad del entorno. Además, la funcionalidad de esta cámara está unida al software ROS para crear un algoritmo que permita al dron detectar los objetos que interfieran en su trayectoria y ser capaz de evitarlos.

Por otro lado, este dispositivo se ha integrado en un robot diseñado para pintar con el fin de detectar áreas que son aptas para ser pintadas [Tadić et al., 2021]. Este sistema de visión dispone de un algoritmo que identifica las paredes. El algoritmo requiere como entrada imágenes de profundidad de 32 bits, que una vez cargados ejecuta una operación para eliminar todos los componentes de profundidad que no son necesarios. A continuación, se aplica una operación de umbralizado junto con operaciones morfológicas para extraer las áreas de la pared que deberían ser pintadas. Finalmente, se extrae la superficie de la pared en binario que junto con los *points cloud* de la cámara ZED dotan al robot de la información necesaria para generar la trayectoria de pintado.

[6]Cámara 3D Zed - https://www.stereolabs.com/zed/specs/

2.1.1.7. Gloveone

Gloveone[7] (ver Figura 2.8) es un guante háptico que fue diseñado básicamente para interaccionar en entornos de realidad virtual y poder experimentar sensaciones como el peso de los objetos virtuales o el calor del fuego. Este guante se compone de una serie de sensores que están colocados por toda la mano, con el objetivo de que mediante las vibraciones que transmite el usuario pueda percibir diferentes sensaciones del mundo virtual. Este guante también tiene predefinido el reconocimiento de una serie de gestos que se detectan con el contacto de sus múltiples sensores.

Figura 2.8: El guante háptico Gloveone. Fuente: [Anthes et al., 2016]

Este dispositivo no ha tenido la misma repercusión que los otros sensores que han sido descritos en esta sección, y por ese motivo no se han encontrado papers con investigaciones innovadoras al respecto. A pesar de estas circunstancias, este guante es un medio de interacción útil en el ámbito de realidad virtual y se han realizado algunos estudios comparativos con otros dispositivos hápticos. Entre esos estudios destaca el trabajo realizado por Perret et al [Perret and Vander Poorten, 2018], donde se exponen las principales limitaciones técnicas en el proceso de creación de estos guantes hápticos y una revisión de estos dispositivos, comparando sus características y rendimiento. Este estudio divide estos sensores en tres categorías: guantes tradicionales, *thimble* y exoesqueleto. Los guantes tradicionales que se han analizado han sido: Gloveone, AvatarVR, Senso Glove, Cynterat y Maestro; los guantes tipo *thimble* son: GoTouchVR y Tactai Touch; y en la categoría exoesqueletos tenemos: CyberGrasp, Dexmo, HaptX, VRgluv, Sense glove DK1 y HGlove. Las características que se han analizado han sido: el tipo de guante, número de dedos, si disponen de tecnología inalámbrica, actuador, retroalimentación de fuerza, retroalimentación táctil, seguimiento de la mano, peso y precio. A partir de la comparación exhaustiva realizada en esta investigación, la principal conclusión que se ha obtenido ha sido que, en términos de principios de actuación los desarrolladores no se arriesgan, siendo el actuador comúnmente utilizado, los motores electromagnéticos tradicionales.

2.1.1.8. Tobii Pro Glasses 2

Tobii Pro Glasses 2[8] (ver Figura 2.9 es un dispositivo que permite realizar un seguimiento de los ojos del usuario. Estas gafas están compuestas por 4 cámaras para los ojos,

[7]Gloveone - https://www.neurodigital.es/gloveone/
[8]Tobii Pro Glasses 2 - https://www.tobiipro.com/product-listing/

un giroscopio y un acelerómetro. Tobii realizar este seguimiento mediante 3 elementos: cámara, iluminadores y algoritmos. Las cámaras se usan para grabar hacia dónde está mirando el usuario, mientras que los iluminadores crean un patrón de luz infrarroja en los ojos con la finalidad de que posteriormente las cámaras tomen capturas de los ojos y los patrones. Finalmente, se utilizan una serie de algoritmos para determinar la posición de los ojos y dónde está mirando el usuario.

Figura 2.9: Las gafas Tobii Pro Glasses 2. Fuente: [tob, 2021]

Este *eye tracker* es usado para un estudio que analiza el proceso de atención con estímulos sociales a través de la mirada [Vehlen et al., 2021]. La originalidad de esta investigación reside en que normalmente este tipo de estudios se hace entre una persona y una pantalla que proporciona estímulos faciales. Sin embargo, en este trabajo la interacción se realiza entre dos personas reales. Los resultados se obtienen a partir de tres experimentos diferentes, los cuales comprueban la precisión y el rendimiento del *eye tracker* en una pantalla de un ordenador convencional, la calidad de los datos procedentes del dispositivo en relación a estímulos 2D y 3D, y el patrón de comportamiento de la mirada en una conversación entre dos personas reales. Después de la finalización de estos experimentos se concluyó que el seguimiento ocular es un medio realmente útil para analizar el comportamiento de la mirada en conversaciones entre dos personas y de esta forma determinar la importancia que desempeña la atención visual en la interacción social entre las personas.

Además, este dispositivo puede resultar beneficioso para las personas con movilidad reducida como demuestra este trabajo [Quesada Elvira, 2014], donde se diseña un sistema para que este tipo de colectivo puedan controlar el cursor del ratón y por consiguiente interaccionar con el sistema informático. En este sentido se han obtenido los datos del *eye tracker* para asociar los movimientos de ambos ojos con el movimiento del cursor y la posibilidad de lanzar eventos. La aplicación ha sido diseñada de una manera intuitiva, cuyo funcionamiento consiste en situar el cursor en la localización de la pantalla donde está mirando el usuario y en pulsar o soltar un botón con la acción de cerrar o abrir un ojo. La evaluación de este trabajo muestra que se ha alcanzado el objetivo fundamental y los usuarios con discapacidad física han sido capaces de controlar el ordenador con ayuda del seguimiento de sus ojos y el movimiento de sus párpados.

2.1.2. Técnicas

En general, un sistema dedicado al reconocimiento de gestos consta de tres etapas principales [Rautaray and Agrawal, 2015]: segmentación, seguimiento y reconocimiento.

2.1.2.1. Segmentación

El primer paso, en el proceso de reconocimiento de gestos en visión artificial, consiste en discernir lo que queremos reconocer con los elementos de la imagen que no son de interés para el reconocimiento de un determinado gesto. Para este fin se van a describir varios métodos de segmentación basados en: el color, la forma, el movimiento, la información de profundidad y modelos 3D.

Segmentación basada en color

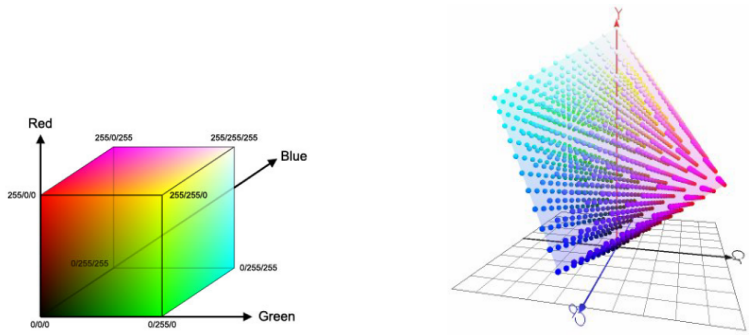
En este tipo de segmentación es de total relevancia que haya una diferencia entre el fondo y qué se quiere reconocer. En primer lugar hay que realizar una sustracción del fondo. Esta técnica consiste en obtener una imagen estática del fondo, calculándose las diferencias entre dicha imagen y el fotograma actual usando un espacio de color. Existen varios espacios de color, sin embargo los más usados [Kristensen et al., 2006] son: RGB, YCbCr o HSV.

- **RGB**: es una mezcla de los colores primarios rojo, verde y azul (ver Figura 2.10a). La principal ventaja radica en su simplicidad [Ghotkar and Kharate, 2012]. Sin embargo, no es el más utilizado por su dificultad para digitalizarlo puesto que la luminosidad y la saturación se encuentran de manera conjunta y es difícil de separar sus componentes R, G y B al estar estrechamente relacionados.
- **YCbCr**: pertenece a la familia del campo de color YUV y tiene una relación entre luminosidad y cromaticidad (ver Figura 2.10b). La Y representa a la luminosidad y los componentes Cr y Cb a la cromaticidad [Qiu-yu et al., 2015].
- **HSV**: este espacio de color está determinado por tres componentes: la tonalidad, la saturación y el brillo (ver Figura 2.11). Es un método útil para visión artificial porque la intensidad es independiente de la información de color. HSV fue creado para poder intrepetar mejor el color[Tsagaris and Manitsaris, 2013] ya que están diseñados para que se aprecien los colores de una forma más realista.

Algunos estudios afirman que usar los espacios de colores HSV o YCbCr es más eficiente [Mahmoud et al., 2008] que usar el RGB, a pesar de que es el más conocido. Esta afirmación se debe a que es preferible usar espacios de colores que separan la luminosidad de la cromaticidad porque tienen más fiabilidad frente a cambios de luminosidad [Terrillon et al., 2000].También se encuentra la técnica que utiliza una mezcla de funciones de densidad de probabilidad Gaussiana (*Mixture of Gaussians*) [Piccardi, 2004] para eliminar el fondo de la imagen.

Segmentación basada en la forma

Se puede obtener mucha información acerca del contorno de la figura que se quiere segmentar. La principal ventaja de este método es que no depende del color de la piel o de la iluminación como en otros métodos. Sin embargo, las sombras y fondos ruidosos pueden afectar a su rendimiento [Poppe, 2007].



(a) (b)

Figura 2.10: Espacios de color RGB e YCbCr. Fuente: [Molinero, 2010]

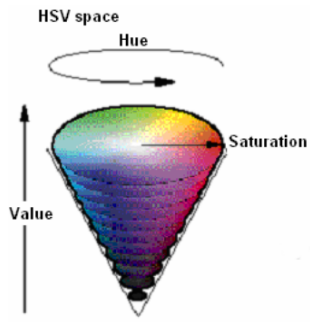


Figura 2.11: Espacio de color HSV. Fuente: [Chicala et al., 2009]

Segmentación basada en movimiento

Esta técnica se basa en comprobar si existe diferencia entre una imagen y otra consecutiva [Moeslund et al., 2006], asumiendo que esa diferencia es a causa de algún movimiento realizado por un ser humano. Para saber la dirección del movimiento se calcula el flujo óptico, colocando un rectángulo en la imagen y comparándolo con el mismo rectángulo en la imagen previa. Con el fin de determinar la dirección, se va cambiando la imagen (arriba, abajo, derecha, izquierda) dependiendo del movimiento que esté realizando el individuo.

Segmentación basada en profundidad

Este método se basa en segmentar la imagen en una serie de partes donde se conoce la información de profundidad de cada uno de los pixel que forma la imagen [Gupta et al., 2013]. Los enfoques de esta metodología se dividen tradicionalmente en: segmentación basada en regiones y basada en bordes. La diferencia principal entre ambos reside en

que mientras la segmentación de regiones trabaja con regiones continuas, la de bordes lo hace con discontinuidades. La segmentación de profundidad basada en regiones tiene el objetivo de marcar los límites de los objetos debido a que la dirección de seguimiento es detectar superficies continuas con propiedades geométricas iguales. Por otro lado, la segmentación de profundidad basada en bordes se centra en la detección de bordes donde clasifica los píxeles de la imagen como borde o ausencia de borde. Esta detección de bordes se puede realizar mediante diferentes técnicas, de las cuales destacan las siguientes: operador de Sobel, operador de Prewitt, operador Laplaciano del Gaussiano y el algoritmo de Canny.

Segmentación con modelos 3D

En el reconocimiento de gestos con las manos, este método consiste en utilizar modelos en 3D de las manos para detectar la mano en las imágenes. Se compara la cinemática del modelo 3D con la mano en 2D para ver la correspondencia respecto a las posturas en ambos modelos y obtener de este modo los parámetros de la mano [Pradipa, 2014].

2.1.2.2. Seguimiento (Tracking)

El seguimiento consiste en determinar si la posición de nuestro objeto de estudio ha sufrido alguna modificación en un intervalo de tiempo que nosotros determinaremos. La clave está en un progresivo análisis de una secuencia de frames comparando el frame actual con el inmediatamente superior y observando si se ha producido algún cambio respecto a la dirección y orientación. El objetivo es determinar en el siguiente frame, cuál será el siguiente paso qué hará el movimiento y así poder anticiparlo.

Seguimiento basado en plantillas

El seguimiento basado en plantilla es un caso particular del seguimiento visual [Kwon et al., 2014] que consiste en designar una plantilla como referencia, [Ladikos et al., 2007a] a partir de la cual se realiza el seguimiento del objeto teniendo en cuenta las diferencias entre las imágenes, para ello se utiliza el flujo óptico con el fin de determinar la translación de las imágenes, [Ladikos et al., 2007b] en el objetivo donde queremos hacer el seguimiento.

Seguimiento basado en el filtro de partículas

Esta técnica es un modelo de estimación utilizado en visión artificial para realizar seguimiento de objetos [Isard and Blake, 1998]. Esta técnica permite trabajar con múltiple hipótesis y desarrollar implementaciones simples y eficientes. Este algoritmo realiza la estimación de una variable mínimo en el tiempo. En cada estimación usa como entrada el valor de dicha variable en el instante anterior y la observación de la información.

Seguimiento Camshift

Este algoritmo es una adaptación del algoritmo *Mean Shift* [Cheng, 1995]. El desarrollo del algoritmo CamShift es debido a que el algoritmo *Mean Shift* tiene el inconveniente de que si se altera el tamaño de la ventana de seguimiento el proceso puede fallar [Collins, 2003]. Sin embargo, CamShift adapta perfectamente el tamaño de la ventana de tracking y el patrón de distribución de los objetivos, en los que se quiere realizar el seguimiento.

Los pasos a seguir en el algortimo CamShift son los siguientes [Nouar et al., 2006]:

(1) Definir el área de interés del objetivo (ROI) en la actual imagen.

(2) Selección automática de dos canales de color.

(3) En cada región se constituye el objetivo y el cálculo de la media de color a todos los píxeles correspondientes.

(4) El histograma 2D se calcula con los canales de colores del paso 2.

(5) Retroproyección del histograma con la siguiente imagen para obtener la imagen de distribución de probabilidad.

(6) Aplicación del algoritmo Mean Shift en esta imagen para determinar el nuevo centro objetivo en la imagen siguiente.

(7) Selección de las regiones que constituyen el siguiente ROI usando las mismas medias calculadas en el paso 3.

(8) Para tener en cuenta los cambios del objeto, puede que se precise volver al paso 2, en otro caso volver al paso 4.

2.1.2.3. Reconocimiento

El reconocimiento de gestos se puede clasificar en dos tipos: reconocimiento de gestos estático y reconocimiento de gestos dinámicos [Arjunlal, 2016]. La diferencia entre un gesto estático y otro dinámico reside en la permanencia del gesto en un período de tiempo. En el proceso de reconocimiento de un gesto estático, denominado postura [Weng et al., 2010], su posición no varía con respecto al tiempo, al contrario que ocurre en el reconocimiento de un gesto dinámico.

En el reconocimiento de gestos estático se usan reconocimiento de patrones, combinación de plantillas y redes neuronales [Mitra and Acharya, 2007], mientras que en el reconocimiento de gestos dinámico se usan técnicas que trabajan con el tiempo, las técnicas más usadas en este sentido son los Modelos Ocultos de Markov, Alineamiento temporal dinámico y *Time Delay Neural Network* (TDNN) [Plouffe and Cretu, 2016].

Algunas de las técnicas usadas en el reconocimiento de gestos tanto estático como dinámico son las siguientes:

K-Means

Es el algoritmo de clustering más utilizado por su fácil aplicación y eficacia. Este algoritmo representa cada uno de sus clusters por la media de sus puntos (centroide). La caracterización de cada cluster reside en su centroide que se encuentra en el medio de todos los elementos que lo forma.

Los pasos principales del algoritmo son [Jain, 2010]:

(1) Selecciona una división de partida con K clusters

(2) Se genera una nueva división donde se le asigna el patrón más cercano a su centro de cluster según una medida de distancia (euclidiana)

(3) Recalcular los centros de K cluster

(4) Repetir los pasos 2 y 3 hasta que los elementos del cluster estén estabilizados

Algoritmo de los K vecinos más cercanos

El algoritmo de los K-Vecinos Más Cercanos [Hart et al., 2000] consiste en estimar el valor de un dato desconocido a partir de las características del dato más próximo, según una medida de distancia.

El procedimiento que se sigue para calcular un dato con este método sería:

(1) Se ubica el dato a clasificar en el plano

(2) Se determina un radio, asignado por alguna heurística

(3) Se traza una circunferencia donde el centro es el dato a clasificar

(4) Se determina el valor de K, determinando si se va a comparar con el primer vecino más cercano, los dos más cercanos, etc.

(5) Asignar la clase al nuevo elemento en relación al valor de K y al número de datos que engloba la circunferencia

Máquinas de Soporte Vectorial

El objetivo de esta técnica es proporcionar una solución al dilema de la búsqueda de una función, que permita explicar el comportamiento de los datos dentro de un dominio más amplio, como es el caso de una tarea de aprendizaje con una cantidad finita de datos. El resultado tiene que ser una relación adecuada entre la precisión alcanzada con un particular conjunto de entrenamiento [Dardas and Georganas, 2011] y la capacidad para aprender con cualquier conjunto de ensayo.

Las ventajas son:

- Entrenamiento relativamente fácil.
- Se escalan relativamente bien para datos en espacios dimensionales altos.

- El compromiso entre la complejidad del clasificador y el error puede ser controlado explícitamente.
- Se pueden insertar como datos de entrada de la máquina de soporte vectorial, datos del tipo cadena de caracteres o árboles.

Modelos Ocultos de Markov

Una cadena de Markov es un autómata con un conjunto finito de estados donde en cada una de sus transiciones reside un valor de probabilidad [Rautaray and Agrawal, 2015]. Estas cadenas poseen la restricción de que por cada estado sólo puede haber una transición dada una salida. Los Modelos Ocultos de Markov es una generalización de las cadenas de Markov pero su origen no es determinístico como sucede con dichas cadenas, con lo cual estos modelos no cumplen la restricción de las cadenas de Markov y pueden tener varias transiciones con la misma salida.

Una definición formal estaría compuesta por la matriz A que es la matriz de transición de probabilidades; la matriz B es la matriz de probabilidad de salida y π es el conjunto de probabilidades de estado inicial. El Modelo Oculto de Markov se define como en la ecuación 2.1:

$$\lambda = (A, B, \pi) \tag{2.1}$$

En el ámbito de reconocimiento de gestos, cada estado podría representar un conjunto de posibles posiciones, donde cada transición representa la probabilidad de que se realice una posición dentro del gesto, así el símbolo de salida representa una postura en concreto y una secuencia de símbolos de salida definen un gesto.

Alineamiento Temporal Dinámico

El algoritmo denominado Alineamiento Temporal Dinámico (*Dynamic Time Warping*, DTW) consiste en el cálculo de las distancias entre elementos mapeados de dos secuencias [Celebi et al., 2013]. Para calcular esta distancia se utiliza como uso general la distancia euclídea, aunque dependiendo del objetivo se puede usar otro método [Bautista et al., 2013]. Estas distancias se usan para calcular una matriz de distancia acumulativa y encontrar el camino menos costoso con la ayuda de esta matriz [Müller, 2007].

En el área de reconocimiento de gestos Plouffe y Cretu [Plouffe and Cretu, 2016] usan este algoritmo para comparar qué tienen en común un gesto tomado como referencia y otro que ha sido tomado. Esta propuesta permite al usuario grabar una secuencia de gestos que serán tomados como referencia y están limitados a 40 imágenes con el fin de que no afecte al rendimiento del sistema. El proceso de reconocimiento de gestos está compuesto por 2 pasos:

(1) Buscar gestos cuya última imagen tomada y los gestos por referencia sean similares.

(2) Validar la similitud entre el gesto observado y cada gesto de referencia.

El algoritmo DTW está presente no sólo en la validación de los gestos en el paso 2, sino para elegir el mejor candidato en el paso 1.

Redes Neuronales con Retardo en el Tiempo

Las Redes Neuronales con Retardo en el Tiempo (*Time Delay Neural Network*) [Yang et al., 2002] son un tipo especial de red neuronal que gracias a la cualidad de retardo en el tiempo tiene dos características esenciales:

(1) Proporciona a cada una de las neuronas de un historial de las señales de entrada.

(2) Tiene acceso no sólo a entradas del tiempo presente si no a otros estados de tiempo como $t_1, t_2 ... t_n$.

2.1.3. Software

En esta sección se describen las librerías que se suelen utilizar para Visión Artificial y para hacer reconocimiento de gestos como OpenCV pero también se han incluido programas que son útiles cuando se desarrolla un sistema en el que se incluye el reconocimiento de gestos como MATLAB.

OpenCV

OpenCV[9] es una librería orientada al campo de la Visión Artificial [Druzhkov et al., 2011] desarrollada en C en su origen por Gary Bradski en 1999. OpenCV está desarrollada con código C++ optimizado para ejecutar sus procesos de una manera rápida y eficiente y así poder crear aplicaciones en tiempo real. Uno de los objetivos de OpenCV que justifica el hecho de que se haya escrito de esta forma, es que se pretende que esta herramienta les sirva a los desarrolladores como una infraestructura de visión artificial que les permita crear sofisticadas aplicaciones [Kaehler and Bradski, 2016] destinadas a este fin, de una manera rápida y sencilla. En la actualidad sus funciones se pueden utilizar en C++, Python, Java e incluso Android.

MATLAB

Es una herramienta desarrollada por MathWorks[10] para computación numérica [Lynch, 2014], con lo cual se relaciona con la disciplina de Matemáticas pero hay muchas otras disciplinas que hacen uso de esta potente herramienta, como por ejemplo: visión artificial, procesamiento de imágenes, robótica o aprendizaje automático.

Google Cloud Vision API

Es una API desarrollada por Google[11] para el reconocimiento de imágenes. Esta herra-

[9] OpenCV - http://opencv.org/
[10] MATLAB - https://www.mathworks.com/products/matlab.html
[11] Google Cloud Vision API - https://cloud.google.com/vision

mienta tiene la capacidad de hacer un análisis y un procesamiento en profundidad de las imágenes y además, se encuentra disponible para un considerable número de lenguajes de programación. Esta API tiene acceso a una base de datos con un gran volumen de datos, a partir de la cual analizará la imagen que se le introduzca como entrada para asignarle las etiquetas correspondientes, y de esta forma identificar los distintos elementos que componen la imagen.

iGesture

Es un framework[12] de reconocimiento de gestos basado en Java [Signer et al., 2007], que permite a los desarrolladores crear sus propios gestos y no estar limitados por un conjunto de gestos predefinidos en la librería. La principal desventaja de este framework es que la curva de aprendizaje es extensa puesto que hay que conocer en detalle las funciones propias de este software.

The Gesture Recognition Toolkit

Es una librería[13] en C++ de código abierto [Gillian and Paradiso, 2014] destinada al reconocimiento de gestos. Uno de los propósitos de esta librería es que pueda ser utilizada y sea más fácil de entender por personas que no son especialistas en este campo. Además, los usuarios que quieran desarrollar sus propios algoritmos no tendrán ningún problema porque es sencillo incluir nuevos algoritmos a esta librería. Este software contiene una gran variedad de algoritmos basados en Árboles de Decisión, Modelo Oculto de Markov, clasificadores Bayesianos, etc, donde el usuario puede elegir el que le resulte más útil dependiendo de la tarea a realizar.

2.1.4. Aplicaciones

En esta sección se describen una serie de proyectos que utilizan el reconocimiento de gestos en la interacción. Se han seleccionado las temáticas más relevantes para ofrecer una visión de la aplicabilidad que tiene este tipo de interacción en la actualidad.

2.1.4.1. Lenguaje por signos

El hecho de que la gente que no padece discapacidad auditiva no aprenda el lenguaje de signos [Adithya et al., 2013] hace que las personas con este tipo de discapacidad se vean afectadas por la marginación. El desarrollo de un sistema de esta índole es necesario para estas personas debido a que necesitan de un intérprete para comunicarse con el resto de personas, lo que genera un estado de dependencia.

Los sistemas de reconocimiento de lenguaje de signos se pueden clasificar principalmente en: sistemas basados en hardware y sistemas basados en visión artificial [Adithya et al., 2013].

[12] iGesture - http://igesture.org/
[13] Gesture Recognition Toolkit - http://www.nickgillian.com/wiki/pmwiki.php/GRT/GestureRecognitionToolkit

La diferencia más característica de estos tipos de sistemas es que el sistema basado en hardware se caracteriza por usar un dispositivo externo para obtener la información necesaria con el fin de reconocer un gesto y el sistema de visión articial se basa en técnicas de procesamiento de imágenes para conseguir el mismo fin.

Otros tipos de sistemas se basan en el uso de dispositivos que realizan reconocimiento de gestos como Microsoft Kinect, Leap Motion o Intel RealSense.

Cuando se habla en lenguajes de signos no sólo se interpreta el movimiento de las manos, sino que para entender cada uno de los signos hay que fijarse también en las expresiones faciales y movimientos de la boca [Ghotkar and Kharate, 2014]. Sin embargo, en este momento la investigación en el lenguaje de signos se está centrando principalmente en el reconocimiento de gestos únicamente con las manos.

El dispositivo que más se utiliza es una cámara web estándar, aunque también se han utilizado otros dispositivos en los sistemas de reconocimiento de gestos como son: Microsoft Kinect [Agarwal and Thakur, 2013, Chai et al., 2013], Leap Motion [Chuan et al., 2014] e Intel Real Sense.

En la fase de experimentación del sistema desarrollado con el sensor Kinect, se obtuvo una tasa de acierto del 97 % [Lang et al., 2012]. El estudio donde se incorporaba la cámara Intel RealSense comparó los resultados obtenidos [Huang et al., 2015] con este dispositivo frente a Microsoft Kinect. La tasa de acierto utilizando Intel RealSense fue de 98,9 % en comparación con Kinect que obtuvo 97,8 %. Por último, con Leap Motion se obtuvieron distintos resultados dependiendo del método de Inteligencia Artifical utilizado. Leap Motion combinado con Redes Neuronales obtuvo una tasa de acierto del 99 %, mientras que con el uso de Redes Bayesianas era de 98 % [Mohandes et al., 2014].

El perfil de usuario al que va dirigido estos sistemas, son personas que tengan discapacidad auditiva.

En los sistemas de reconocimiento de lenguaje de signos es necesario utilizar varias técnicas para cada uno de los procesos involucrados. En el proceso de segmentación se usa el modelo de color HSV [Ghotkar and Kharate, 2012], el modelo YCbCr [Adithya et al., 2013] y la combinación del espacio de color RGB y HSI [Maraqa and Abu-Zaiter, 2008]. En el proceso de seguimiento se usa el método Camshift [Ghotkar and Kharate, 2012]. En el proceso de reconocimiento de gestos son usados con frecuencia Modelos Ocultos de Markov [Lang et al., 2012, Zafrulla et al., 2011, Yang et al., 2015] y Redes Neuronales [Adithya et al., 2013, Mohandes et al., 2014, Huang et al., 2015, Al-Jarrah et al., 2006, Elons et al., 2014, Maraqa and Abu-Zaiter, 2008].

En el siguiente estudio [Ghotkar and Kharate, 2012] la imagen de la mano segmentada es representada usando ciertas características: distancia de Hausdorff y descriptor de Fourier. El reconocimiento de gestos se basa en un Algoritmo Genético que usa las características mencionadas anteriormente.

La arquitectura de un sistema tradicional de reconocimiento de signos (ver Figura 2.12) se compone de dos módulos:

- **Módulo Speech**: el propósito del primer módulo es convertir una frase del lenguaje que se quiere interpretar a lenguaje de signos. El proceso comienza con la utilización del reconocimiento de voz para capturar la frase a interpretar y realizar un procesamiento de texto. Una vez obtenida la frase en texto el motor del pro-

cesamiento de lenguaje traduce esta frase a lenguaje de signos con ayuda de una gramática sobre lenguaje de signos.

- **Módulo Gesture**: el segundo módulo reconoce los gestos que está haciendo el usuario mediante procesamiento de imágenes y le envía la información obtenida al motor del procesamiento de lenguaje. Este motor se encarga fundamentalmente de traducir los gestos a una frase del lenguaje destino utilizando como medios de salida: texto y voz.

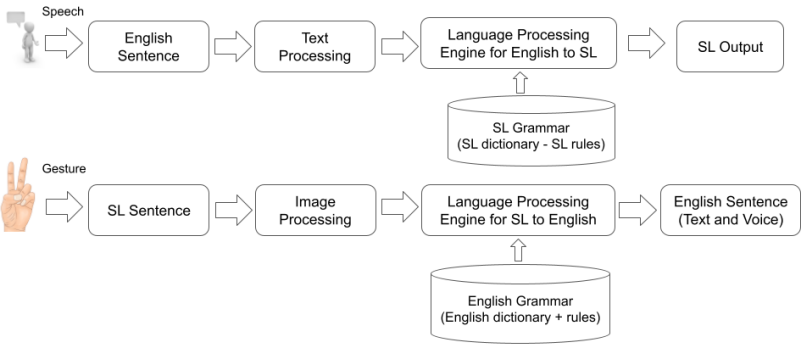


Figura 2.12: Arquitectura de un sistema de reconocimiento de signos tradicional. Fuente adaptada: [Ghotkar and Kharate, 2017].

En la literatura se pueden encontrar otras arquitecturas. Ghotkar [Ghotkar and Kharate, 2012] presenta la siguiente arquitectura (ver Figura 2.13) compuesta por cuatro módulos: segmentación, seguimiento de la mano, extracción de características y reconocimiento de gestos. El seguimiento de las manos se realiza con el algoritmo Camshift y a continuación se segmenta la mano del fondo con el modelo de color HSV. En el módulo de Extracción de características se obtiene la representación de la mano haciendo uso de estas características: distancia de Hausdorff y descriptores de Fourier. En el último paso, se realiza la contrastación de las características obtenidas con las que hay almacenadas en la base de datos y reconocer los gestos que hace el usuario mediante un algoritmo Genético.

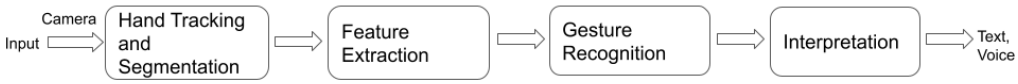


Figura 2.13: Arquitectura de un sistema de reconocimiento de signos. Adaptado de: [Yang and Zhu, 2017]

2.1.4.2. Hogares Inteligentes

En los últimos años el campo de la interacción humano-computadora ha sido de gran interés [Kim and Kim, 2006] en el ámbito de los hogares inteligentes. Este interés es suscitado por el hecho de que el reconocimiento de gestos es una manera sencilla de controlar los sistemas que contiene un hogar, como por ejemplo, el aire acondicionado o la luz. Actualmente, no se piensa en invertir en casas que tengan un reconocimiento de gestos implementado porque el reconocimiento de gestos dinámico es difícil de implementar [Shinde et al., 2016], no se ha conseguido realizar un reconocimiento que sea completamente natural. Además, los algoritmos de procesamiento requieren altas prestaciones en los equipos. Sin embargo, existen proyectos que están usando señales inalámbricas [Pu et al., 2013] para poder realizar el reconocimiento de gestos en cualquier parte de la vivienda o sensores capacitivos hechos de tela [Singh et al., 2015b], que permiten a personas con movilidad reducida hacer gestos para controlar diferentes sistemas de la casa de una forma más sencilla.

Las partes del cuerpo que se usan en el reconocimiento de gestos para estos entornos, son las manos y el cuerpo, aunque especialmente se desarrollan sistemas que reconocen gestos con las manos.

Qamar et al [Qamar et al., 2015] presentan una arquitectura basada en un hogar inteligente que tiene integrado un sistema de reconocimiento de gestos. En la Figura 2.14 se puede observar cómo el módulo denominado *Multimedia Sensors* proporcionan la información necesaria para interpretar lso gestos que está realizando el usuario en todo momento. El *Gesture to Action Map* contiene todos los gestos que interpreta el sistema y los almacena. El usuario es capaz de seleccionar la acción que quiera dependiendo de los gestos interpretados por cada uno de los dispositivos en el sistema, gracias al *Gesture Action Mapping Interface*. El *Gesture Detection and Quantification Engine* le proporciona un gesto que ha sido realizado al *Action Selector*, que se encarga de ejecutar la acción que va asociada a ese gesto en el ambiente doméstico.

En el siguiente estudio [Rahman et al., 2009] se utiliza una cámara de infrarrojos y un guante de infrarrojos para realizar el reconocimiento de gestos. El guante ha sido diseñado por los investigadores de este proyecto y se compone de un emisor de infrarrojos que permite dibujar gestos en el aire. La cámara de infrarrojos tiene un emisor de infrarrojos con seguimiento que permite captar el movimiento del guante mientras se mueve y de esta forma identificar el gesto que está realizando con la información almacenada en la base de datos.

La cámara de profundidad ha sido uno de los dispositivos elegidos por una Universidad de Korea [Dinh et al., 2014] para controlar ciertos aspectos de una casa mediante gestos con las manos. Esta cámara captura una imagen de profunidad de la mano y genera una silueta de profundidad de la mano, la cual asocia con un gesto registrado en el sistema.

Por otro lado, se han utilizado dispositivos de reconocimiento de gestos para integrarlos en una casa inteligente. En este proyecto [Qamar et al., 2015] se han utilizado tres dispositivos: Myo, Kinect v2 y Leap Motion. La justificación de usar estos dispositivos es que los usuarios llevan el dispositivo Myo en el brazo con lo cual no tienen que preocuparse de estar en una posición concreta del dispositivo, al contrario que ocurre con

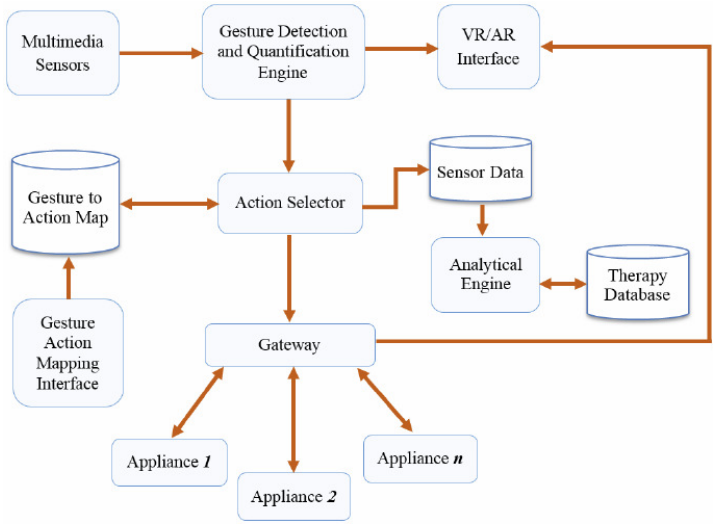


Figura 2.14: La arquitectura de una Smart Home. Fuente: [Qamar et al., 2015]

Kinect v2 y Leap Motion que se necesita mantener una distancia con estos dispositivos para su uso óptimo. Sin embargo, Myo permite reconocer un número limitado de gestos y Microsoft Kinect v2 permite no sólo reconocer más gestos, si no que se puede utilizar todo el cuerpo para este fin. El hecho de usar Leap Motion, es debido a que el sensor Kinect v2 utiliza motricidad gruesa y no permite el reconocimiento de gestos con las manos, con lo cual el dispositivo Leap Motion es usado para realizar un reconocimiento más preciso de gestos con las manos.

Además de la utilización de estos dispositivos, los guantes de infrarrojos son también un elemento común en el reconocimiento de gestos y se han incluido en este proyecto [Rahman et al., 2009] donde los usuarios tenían que dibujar los gestos con este tipo de guante. Además, en este trabajo se ha realizado una evaluación para medir la precisión del proceso de reconocimiento de gestos. La evaluación estaba dividida en cuatro fases, donde cada fase estaba compuesta por 15 usuarios que tenían que realizar 20 gestos con las manos. Aunque al principio del experimento los usuarios tenían dificultad para dibujar en el aire los gestos que se proponían, finalmente se obtuvo una tasa de acierto de 97.25 %.

A pesar de que se han utilizado diferentes dispositivos en las aplicaciones de hogares inteligentes, las cámaras estándar han sido también utilizadas para este propósito [Abid et al., 2015, Starner et al., 2000]. Algunos de los dispositivos mencionados anteriormente usan las imágenes en profundidad para realizar el reconocimiento de gestos. La investigación que usaba imágenes de profundidad en un hogar inteligente, realizó una evaluación para comprobar la validez del método de reconocimiento de posturas de las manos. Esta evaluación consistió en adquirir previamente un conjunto de imágenes de profundidad sobre siluetas de las manos, procedentes de cinco individuos diferentes. Una vez obtenidas las siluetas, cada participante realizó 40 poses con sus manos. La tasa de

acierto fue de un 98.50 % [Dinh et al., 2014].

La mayoría de estudios no especifican un usuario concreto al que va destinado dando por supuesto que puede ser utilizado por cualquier persona. Sin embargo, hay dos estudios que sí han definido un perfil concreto de usuario para sus prototipos desarrollados. Qamar [Qamar et al., 2015] afirma que su proyecto va orientado a personas con discapacidad y Abid [Abid et al., 2015] explica en su trabajo que su desarrollo puede resultar útil a personas con discapacidad auditiva y personas con problemas de comunicación.

Los resultados obtenidos en este campo son alentadores ya que la tasa de acierto obtenida de estas investigaciones ha oscilado entre un 97 % [Starner et al., 2000] y un 98.65 % [Abid et al., 2015].

En el ámbito de reconocimiento de gestos en el hogar inteligente destacan dos proyectos importantes; *WiSee* es el primer sistema *wireless* orientado al reconocimiento de gestos en un ambiente doméstico [Bedi, 2013]. Este proyecto desarrollado por investigadores de la Universidad de Washington tiene la característica de que no es necesario estar delante de ningún tipo de dispositivo de reconocimiento de gestos como Microsoft Kinect, ni requiere de una infraestructura de cámaras, puesto que las señales wifi son capaces de atravesar la estructura del edificio.

La característica principal que hace que este sistema se diferencie del resto es que no existe otro sistema actualmente que utilice las señales wifi para realizar reconocimiento de gestos.

El sistema envía un *stream* [Garber, 2013] de señales wifi. *WiSee* es capaz de reconocer tanto gestos hechos con las manos como con el cuerpo. El sistema detecta que se ha realizado un gesto porque éstos interrumpen el stream y provoca pequeños movimientos de manera frecuente (desplazamientos de Doppler). Sin embargo, estos desplazamientos son tan insignificantes, que es muy difícil detectarlos en una transmisión wifi. Los investigadores de este proyecto solucionaron este inconveniente transformando la señal recibida en un pulsos de banda estrecha con un ancho de banda de unos pocos Hercios.

Para calcular los desplazamientos Doppler de la señal transformada y obtener información necesaria para reconocer los gestos de los usuarios, es necesario hacer estos pasos [Pu et al., 2015]: construir un perfil Doppler, Segmentación y Clasificación de los gestos.

El procedimiento para construir un perfil Doppler es el siguiente:

(1) El receptor calcula una ventana de transformación de frecuencia cada 500 ms.

(2) Se genera una distribución con una resolución de 2 Hercios.

(3) A continuación, el receptor avanza 5 ms y genera otra ventana de transformación de frecuencia.

(4) Este proceso se repite hasta conseguir el perfil Doppler.

En el paso de segmentación WiSee tiene que encontrar el principio y el final de un gesto. Esto se consigue haciendo uso de los perfiles Doppler. El receptor de WiSee hace uso de los sectores positivos y negativos de los desplazamientos Doppler para detectar partes de un gesto y agrupar dichas partes en un gesto.

Por último, el proceso de clasificación de gestos consiste en encontrar los patrones que hacen únicos a cada uno de los gestos que se quieren reconocer. Para este fin, se le asigna

a cada uno de los gestos una numeración de secuencia. Los números de secuencia definen cada uno de los efectos de un desplazamiento Doppler. Por ejemplo, un desplazamiento positivo Doppler se numera como 1 y uno negativo como -1. La clasificación de los gestos consiste en comparar y asociar una secuencia de numeración recibida con las secuencias que se tienen preestablecidas.

El prototipo fue evaluado en dos ambientes: un bloque de oficinas y un piso de dos dormitorios. En este experimento participaron cinco usuarios, que fueron evaluados en seis escenarios distintos. En esta evaluación se realizó una prueba dónde participaba un solo usuario y otra donde participaban varios usuarios para ver la fiabilidad del sistema frente agentes externos que podían interferir en el reconocimiento de gestos. Los resultados de la fase donde sólo participaba un usuario y éste se encontraba en el piso de dos dormitorios, fueron una tasa de acierto del 94 % [Pu et al., 2013] en el reconocimiento de los nueve gestos implementados. En la fase donde participaban varios usuarios se obtuvo una tasa de acierto del 90 % usando una antena de cinco receptores y tres usuarios. Sin embargo, la tasa de acierto es menor del 60 % cuando se encuentran cuatro usuarios.

El otro proyecto se denomina *Inviz* y está compuesto por una interfaz que tiene la finalidad de hacer más sencilla el control de una vivienda inteligente [Singh et al., 2015a] por parte de personas con movilidad reducida. Este sistema realiza reconocimiento de gestos usando unos sensores capacitivos que pueden ser colocados en el entorno, el propio cuerpo del usuario o la ropa. El algoritmo jerárquico de procesamiento de señales convierte las señales de estos sensores en gestos fiables y de esta forma que los usuarios puedan controlar diferentes aspectos de su hogar mediante gestos corporales.

El objetivo de Inviz es proporcionar un reconocimiento de gestos en tiempo real orientado a pacientes con movilidad reducida que utilice un consumo mínimo de energía [Singh et al., 2015b].

Este proyecto tiene dos características que lo hacen innovador:

(a) Un conjunto de parches capacitivos hechos de tela que actúan como sensores de proximidad y se pueden poner en la ropa para detectar movimientos y gestos en usuarios con problemas de motricidad.

(b) Un algoritmo de procesamiento de la señal que descompone la computación en niveles de alta y baja intensidad.

Este sistema tiene dos características principales que hacen posible el reconocimiento de gestos en un hogar inteligente para personas con movilidad reducida. Estas características son: el reconocimiento de gestos y el movimiento a distancia. Estas técncias son realizadas con una serie de sensores wearables capacitivos basado en tela. Estas placas al estar basadas en tela se pueden colocar en cualquier prenda de vestir. El hecho de incluir una serie de placas es para reducir el ruido y de esta manera poder captar la velocidad y la dirección del movimiento realizado con el cuerpo.

Este proyecto presenta un algoritmo de reconocimiento de gestos jerárquico que distribuye el proceso en diferentes niveles. Las placas recogen una serie de información que es extraído por este proceso para usarla en un algoritmo de aprendizaje automático y así

determinar el gesto que tiene una mayor probabilidad de similitud. El proceso jerárquico aporta precisión y un reconocimiento de gestos continuo con una baja potencia.

En la evaluación participaron cinco personas adultas que no sufrían ninguna discapacidad física, con el objetivo de que fueran una referencia para evaluar la precisión del sistema de reconocimiento de gestos implementado. Cada sujeto realizó los gestos implementados entre 9 y 12 veces. El total de gestos incluidos en el sistema eran 16.

Se realizaron dos experimentos cuyos objetivos eran medir la precisión de reconocimiento de gestos y el consumo de energía.

En relación al reconocimiento de gestos, se propusieron en primera instancia tres algoritmos de aprendizaje automático: clasificador del algoritmo más cercano, clasificador del árbol de decisión y clasificador de redes bayesianas. El algoritmo del vecino más cercano obtuvo la tasa de precisión más alta 93 %.

En la evaluación respecto al consumo de energía se obtuvieron tres conclusiones principalmente:

(a) El consumo energético es bajo y consume alrededor de 1.7 mW, lo que se traduce en que con una batería de 1000 mAh, el sistema podría estar funcionando durante 83 días.

(b) El sistema consume cuatro veces menos energía que un sistema que no usa una arquitectura jerárquica.

(c) Los dos componentes que consumen más energía del sistema son el módulo Bluetooth y el hardware de cómputo.

2.1.4.3. Smart TV

Smart TV es una plataforma inteligente que tiene conexión a internet. Estas televisiones disponen de sistema operativo y su conexión suele ser preferentemente inalámbrica, con lo cual es prácticamente como tener un ordenador en el salón de casa pero con menos prestaciones. Los usuarios pueden navegar por internet y de esta forma consultar su correo, tener acceso a sus redes sociales o ver los vídeos en Youtube. Estas televisiones vienen con una serie de aplicaciones por defecto, lo que permitirá a los usuarios jugar a videojuegos, escuchar su música favorita o comprar películas de interés sin levantarse del sillón. Es necesario añadir que algunas Smart TV vienen con una webcam integrada y ofrecen la posibilidad de realizar una videollamada con ellas.

El reconocimiento de gestos en este ámbito está orientado a poder controlar los distintos aspectos de estas televisiones sin tener que estar cerca de ellas. Este tipo de interacción resulta muy cómoda para el usuario porque seleccionaría las canciones, las películas o cualquier otro elemento con un gesto muy sencillo y se podría desplazar entre las distintas aplicaciones con un gesto de *swipe*. Además, ofrece la posibilidad de pausar una película o bajar/subir el volumen de una forma intuitiva sin tener que usar el mando a distancia.

Los tipos de gestos elegidos han sido principalmente con las manos y los brazos aunque en un estudio también se ha usado la cara. En este estudio [Lee et al., 2013] se ha elegido la cara para hacer autentificar al usuario y el reconocimiento de gestos con las manos para controlar las funciones del Smart TV.

Sang-Heon Lee [Lee et al., 2013] propone esta arquitectura (Ver Figura 2.15) para sistemas de SmartTV con reconocimiento de gestos. Esta arquitectura se divide en dos partes: reconocimiento de interfaz de usuario e interfaz de usuario interactiva. En el Reconocimiento de Interfaz de Usuario se encuentra el núcleo del sistema compuesto por un módulo de reconocimiento de gestos y otro de reconocimiento facial. La Interfaz de Usuario Interactiva se trata de una interfaz gráfica de usuario que está controlada por el Reconocimiento de Interfaz de Usuario.

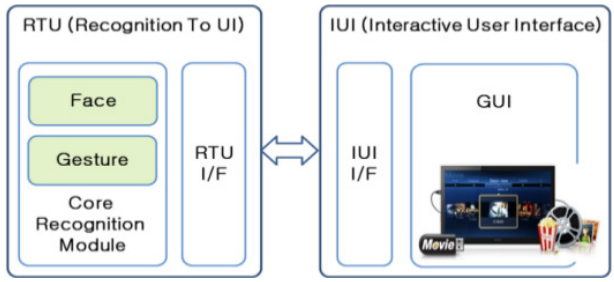


Figura 2.15: Arquitectura de un sistema de Smart TV. Fuente: [Lee et al., 2013]

La Universidad Sun Yat-sen de Taiwan ha desarrollado esta arquitectura [Lee et al., 2014] (ver Figura 2.16) para un framework orientado a proporcionar servicios para SmartTV. El framework se basa en una arquitectura cliente-servidor, que contiene tres características principales: reconocimiento de gestos realizados con el cuerpo, sistema de recomendación basado en etiquetas y una plataforma sensible al contexto para hacer mejores recomendaciones. En el módulo de reconocimiento de gestos el sensor Kinect v1 le proporciona datos acerca del usuario para entrenar el modelo de usuario mediante máquinas de soporte vectorial. El modelo entrenado es usado para realizar el reconocimiento de gestos y asociarlo con una acción para controlar el Smart TV. El sistema de recomendación primero activa el módulo encargado de asignar una prioridad a la lista de los elementos multimedia. A continuación, reordena el orden de ranking dependiendo de la información recibida del contexto.

El dispositivo por excelencia que se ha elegido para hacer el reconocimiento de gestos para Smart TV ha sido Microsoft Kinect v1, aunque se ha utilizado la cámara estándar en algunos experimentos.

El reconocimiento de gestos orientado a Smart TV va dirigido a cualquier tipo de usuario. Sin embargo, este proyecto [Hwang et al., 2015] ha sido desarrollado para que pueda ser utilizado con personas con discapacidad visual.

En el estudio [Lee et al., 2013] que utilizaban reconocimiento facial y de las manos, se ha obtenido en el reconocimiento facial usando secuencias de patrones de histogramas binarios una tasa de acierto de 97 % y en el reconocimiento de gestos con las manos el porcentaje es de un 80 % usando máquinas de soporte vectorial.

En el siguiente proyecto [Hwang et al., 2015] se hizo un estudio para medir la viabilidad de los gestos desarrollados en relación a la interfaz comercial basada en gestos para señalar. Para ello, participaron en la evaluación dieciséis usuarios que tenían discapacidad visual. Los participantes tenían que realizar gestos de cada uno de los tres tipos:

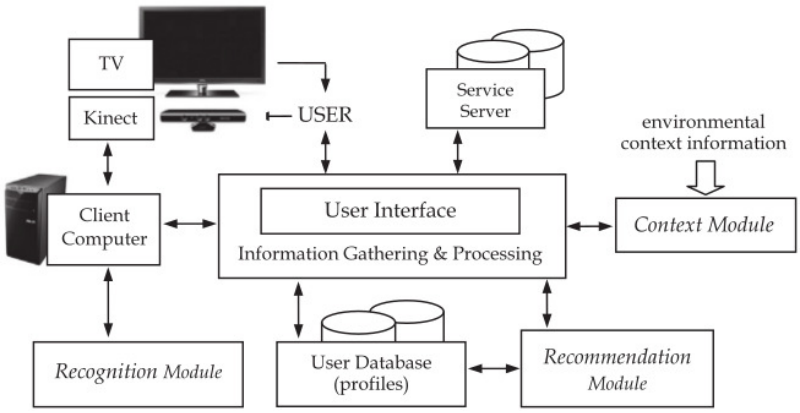


Figura 2.16: Arquitectura de un framework para un sistema Smart TV. Fuente: [Lee et al., 2014]

lineales, circulares o para señalar, de forma aleatoria. La evaluación tenía el objetivo de medir 5 factores: la exigencia de la tarea a nivel mental, a nivel físico, el éxito según el nivel de desempeño del usuario, el esfuerzo que exigía la tarea y en nivel de frustración del usuario. Los resultados que se obtuvieron en el experimento según estas medidas fueron que los tres tipos de interfaz obtuvieron puntuaciones similares en los apartados de exigencia a nivel mental, físico y esfuerzo. Las interfaces lineales y circulares obtuvieron un mayor porcentaje de tareas completadas y había menos frustración en la interfaz lineal respecto a la de señalar. Los aspectos negativos que se concluyeron de esta evaluación fueron que los participantes consideraban que la interfaz del tipo para señalar no era apropiada para usuarios que sufran de discapacidad visual y resultaba difícil localizar el cursor en un icono. En el conjunto de gestos lineales, los usuarios se quejaban de que los movimientos que tenían que realizar tenían gran envergadura. Por último. los participantes tenían dificultades para memorizar y asociar los gestos circulares a las funciones correspondientes.

En la evaluación de este sistema [Dias et al., 2013], se realizaron una serie de pruebas sobre el procedimiento de clasificación para comprobar la precisión del sistema. Según esta evaluación, habría que modificar los estados del Modelo Oculto de Markov para obtener una tasa de error de 0.3 %.

2.1.4.4. Serious Games

Los Serious Games se tratan de juegos que tienen un propósito más allá del mero entretenimiento. Los Serious Games han sido aplicados en multitud de campos como la salud, educación o militar. Este tipo de aplicación tiene la ventaja de que el usuario puede realizar acciones de situaciones reales, que sería muy costoso de hacer en otro caso, debido al tiempo, recursos que serían necesarios utilizar, etc. El reconocimiento de gestos ha sido aplicado a este ámbito porque se ha comprobado que es efectivo utilizar

este medio de interacción cuando se aplica en el aprendizaje de niños, para prevenir la demencia o en educación especial [Jain et al., 2012].

Los gestos que se han realizado en este tipo de proyectos han sido con el cuerpo, especialmente con la parte superior del mismo.

Los dispositivos que se han utilizado han sido principalmente cámaras de profundidad de diversa índole: Microsoft Kinect, PrimeSense o ASUS Xtion PRO.

El usuario al que va orientado depende del propósito del juego. Por ejemplo, en la bibliografía consultada hay juegos destinados a niños de preescolar para mejorar su aprendizaje y habilidades motoras. En otros casos, se encuentran juegos para prevenir la demencia.

En la evaluación realizada por Seong Jeong [Jeong et al., 2016] participaron adultos y niños obteniendo una precisión en el reconocimiento de gestos de 98.67 %.

En el estudio realizado por la *National Taiwan Normal University* [Hsiao and Chen, 2016] se realizó una evaluación en la cual participaron 105 estudiantes de preescolar (49 chicas y 56 chicos) de los cuales, 53 de ellos pertenecían a un grupo de control y los restantes 52 estudiantes estaban agrupados en el grupo experimental. El grupo de control usó el método tradicional de aprendizaje basado en juego, mientras que el grupo experimental hizo uso del método de aprendizaje interactivo con gestos basado en juegos. Esta evaluación está compuesta por dos pruebas, una para medir el rendimiento escolar y otra para medir las habilidades motoras. La prueba que consiste en medir habilidades motoras se centra en medir la coordinación y agilidad del individuo. Los participantes del grupo experimental obtuvieron mejores resultados que el grupo de control en ambas pruebas, lo que demuestra que esta metodología de aprendizaje basada en gestos es más efectiva que la tradicional que se usaba hasta el momento.

2.2. Otros tipos de interacción natural

En este apartado se van a describir otros tipos de interacción natural más relevantes y que podrían ser en algunos casos complementarios al reconocimiento de gestos. Estos tipos de interacción son: interacción táctil, reconocimiento de voz e interfaz cerebro-ordenador (BCI).

2.2.1. Interacción táctil

La interacción táctil implica que debe haber un contacto entre el usuario y el dispositivo que ha sido diseñado para este fin. Este tipo de interacción es familiar para los usuarios desde que los dispositivos móviles son una herramienta ordinaria en la vida cotidiana de las personas. En la actualidad, es común que los dispositivos móviles permitan una interacción multitáctil, en la cual es posible obtener una retroalimentación cuando el usuario hace contacto con varios puntos de la superficie táctil al mismo tiempo.

2.2.1.1. Dispositivos

Los dispositivos más usuales que se aplican a la interacción táctil son: la pantalla táctil y la pizarra interactiva.

La **pantalla táctil** (ver Figura 2.17a) es una superficie que reacciona al tacto del usuario y le permite interaccionar con el dispositivo. En esta clase de interacción se suelen utilizar los dedos y se puede encontrar en ordenadores tipo all-in-one, ordenadores portátiles, tablets y smartphones, entre otros.

La **pizarra interactiva** (ver Figura 2.17b) hace uso de un proyector para mostrar imágenes en su superficie. El usuario puede interaccionar con los elementos proyectados en la pizarra usando sus dedos o un bolígrafo específico como si estuviera utilizando un ratón. Este tipo de dispositivos se suelen encontrar en centros con alumnos con discapacidad porque es más fácil interaccionar con este tipo de dispositivo que con los métodos tradicionales.

(a) Pantalla táctil. Fuente: [tou, 2021]

(b) Pizarra interactiva. Fuente: [int, 2021]

Figura 2.17: Ejemplos de dispositivos de interacción táctil.

2.2.1.2. Aplicaciones

Las aplicaciones, basadas en interacción táctil, más destacadas están enfocadas a aspectos médicos y al entretenimiento. Por este motivo, en esta sección se van a describir varios trabajos aplicados a la cirugía y a la realidad virtual.

Cirugía

Este tipo de interacción es muy utilizado en cirugía minimamente invasiva ya que es sumamente importante/recomendable tener un feedback táctil cuando se practica este tipo de cirugía. Algunos de los trabajos realizados en este ámbito son:

En [Yao et al., 2005], se ha creado una herramienta cuyo objetivo principal es mejorar la experiencia táctil de la cirugía mínimamente invasiva. Este instrumento consta de un acelerómetro y un actuador que hacen posible tener una realimentación táctil y auditiva. Esta retroalimentación ha sido determinante en los experimentos ya que ha demostrado que ha mejorado notablemente el rendimiento de los usuarios que lo han utilizado. En este estudio [Okamura, 2009] se refuerza la idea de que la sensación táctil es muy significativa en la cirugía mínimamente invasiva. En este caso se expone el ejemplo de la cirugía mínimamente invasiva asistida por robot, la cual tiene como factor limitante la ausencia

de retroalimentación háptica. Sin embargo, en la actualidad existen dispositivos que se han creado para este propósito aunque tienen la limitación de que solo son capaces de proporcionar información táctil precisa durante las mediciones estáticas desde un punto porque es complicado realizar pruebas sobre sujetos vivos a menos que el dispositivo ofrezca unas garantías de seguridad muy elevadas [Konstantinova et al., 2014].

En estudios previos se ha comprobado la relevancia de combinar la retroalimentación táctil y auditiva, así que los autores de este trabajo [Lim et al., 2015], viendo los resultados positivos de combinar diferentes tipos de retroalimentación para la cirugía mínimamente invasiva han decidido desarrollar un sistema de retroalimentación háptico, el cual une la retroalimentación táctil y kinesética. La idea surge a partir de los hechos de que la retroalimentación táctil es más relevante cuando los cirujanos realizan fuerzas de menor intensidad, al contrario de la retroalimentación kinesética que es más dominante en las fuerzas de mayor intensidad. Por lo tanto, la combinación de ambas mejoraría la precisión de las acciones de este tipo de operación y además reduciría el tiempo de aprendizaje que requiere el uso de los robots destinados a cirugía.

Realidad Virtual

La mayoría de las aplicaciones de realidad virtual (Virtual Reality, VR) se basan en una experiencia visual, aunque la visión sea el más importante de nuestros sentidos [Gonzalez and Woods, 2006], este no es suficiente para que el usuario se sienta totalmente sumergido en el mundo virtual. Sin embargo, incluir una retroalimentación háptica tiene las desventajas principales de que la creación es complicada y este tipo de sistemas suelen ser caros. Por estas razones los autores de este trabajo [Pamungkas and Ward, 2016] han aplicado retroalimentación electro-táctil en la VR donde una serie de electrodos estimulan los nervios mediante impulsos eléctricos. Este sistema se compone de unas gafas de realidad virtual Oculus Rift que son las que van a proporcionar la experiencia de formar parte de un mundo virtual mediante la visión, un dispositivo Leap Motion que permite interaccionar con las manos con dicho mundo virtual a través de un seguimiento con las manos y un guante háptico que está formado por un cojunto de electrodos que transmitirán los impulsos eléctricos para integrar el sentido del tacto a la experiencia también. La hipótesis de este trabajo es que la inclusión de esta retroalimentación electrotáctil mejora la sensación de inmersión en VR y la interacción con el entorno virtual, que ha sido constatada con los experimentos realizados.

En [Scheggi et al., 2015] se propone un sistema háptico para VR cuyos componentes son el sensor Leap Motion, que al igual que en el artículo anterior, es de utilidad para realizar un seguimiento de la mano en todo momento y de esta forma tener la posibilidad de interactuar con los elementos virtuales del entorno, y cinco dispositivos táctiles pórtatiles que ofrecerán la sensación háptica. Estos sensores están conectados a una pulsera que les proporciona alimentación y conexión inalámbrica, con lo cual no es necesario el uso de cables. Esta propuesta ha sido realizada porque proporciona un método efectivo y ecónomico comparado con otras propuestas de retroalimentación háptica como son el guante CyberGrasp[Aiple and Schiele, 2013] o Rutgers Master II [Bouzit et al., 2002].

2.2.2. Reconocimiento de voz

El habla consiste básicamente en la producción de sonidos que el ser humano es capaz de realizar mediante las cuerdas vocales. Estos sonidos son ondas que se propagan en el aire y que denominamos voz. El reconocimiento de voz es un proceso que acepta como entrada un audio en el cual intenta identificar palabras del idioma que haya sido integrado en el sistema. Este proceso consta de una serie de etapas, donde en primer lugar es necesario adquirir la onda que conforma el audio para después dividirla en distintas partes que se identifican mediante los espacios y finalmente reconocer cada una de las partes divididas previamente. En este reconocimiento se trata de comparar esas ondas con los audios que se tienen de un lenguaje para identificar a qué palabra pertenece dicha onda. Este proceso de comparación se compone de diferentes tareas donde será necesario realizar una extracción de características para obtener las caracterísiticas necesarias para comparar los diferentes audios y después la creación de un modelo que contendrá los objetos matemáticos necesarios para realizar la comparación entre los distintos vectores de características [Espana-Bonet and Fonollosa, 2016].

2.2.2.1. Motores de reconocimiento de voz

Los dispositivos que son utilizados para el reconocimiento de voz principalmente son Intel RealSense 2.1.1.3 y Kinect 2.1.1.1, que han sido descritos en apartados anteriores. Sin embargo, los motores de reconocimiento de voz son el núcleo de este sistema y se van a describir algunos de ellos a continuación.

Los motores de reconocimiento de voz se suelen clasificar en: motores de reconocimiento de voz de software libre y motores de reconocimiento de voz privados [Matarneh et al., 2017]. Los motores de reconocimiento de voz de software libre más destacados en el mercado son:

- **CMU Sphinx** [Lamere et al., 2003]: este software ha sido creado por la *Carnegie Mellon University*, el cual tiene un vocabulario extenso y una base de código de reconocimiento de voz independiente del hablante. Este programa tiene varias versiones y paquetes donde la última versión ha sido desarrollada en Java que es un lenguaje multiplataforma y su modelo acústico ha sido generado a partir de un modelo de Modelos Ocultos de Markov.

- **Julius** [Lee and Kawahara, 2009]: esta aplicación ha sido desarrollada en el lenguaje C por la *Kyoto University* y es multiplataforma. Julius soporta varios idiomas y el procesamiento de archivos de audio y una transmisión de audio en vivo. Este sistema funciona con un modelo de lenguaje y un modelo acústico que ha sido definido mediante Modelos Ocultos de Markov.

- **Kaldi** [Povey et al., 2011]: es una herramienta escrita en C++ con licencia Apache License 2.0 que se puede usar en los sistemas más comunes de Unix y Microsoft Windows. Las características principales de este software son la integración con transductores de estado finito, soporte para álgebra lineal, diseño extensible que proporciona algoritmos genéricos y pruebas exhaustivas para tener rutinas de test en la mayor parte del código.

Por otro lado, los motores de reconocimiento de voz comerciales más destacados son:

- **Dragon Mobile SDK**[14]: Este software es reconocido mundialmente y está desarrollado por la prestigiosa compañía Nuance Communications. El sistema está formado por una parte cliente y otra servidor en los principales sistemas operativos para dispositivos móviles: Android, iOS, etc. Esta aplicación es capaz de convertir texto a voz y viceversa, siendo su integración con otras aplicaciones muy sencilla. Dragon Mobile SDK tiene una tasa de acierto de reconocimiento considerablemente alta, alcanzando el 99 % para el idioma inglés.
- **Google Speech Recognition API**[15]: esta librería desarrollada por Google utiliza Google Cloud Speech-to-Text para poder convertir audio a texto, donde se aplican redes neuronales convolucionales. Además es capaz de reconocer más de 120 idiomas e identificar automáticamente el idioma del audio.
- **Microsoft Speech API**[16]: es el software de reconocimiento de voz desarrollado por Microsoft que está integrado en el framework .NET. La ventaja principal de utilizar esta librería para desarrollo es que se puede integrar fácilmente con las aplicaciones desarrolladas con .NET, así como en los sistemas operativos de Windows y las herramientas de Microsoft Office.

2.2.2.2. Aplicaciones

Hoy en día, el reconocimiento de voz está presente en muchos aspectos de la vida cotidiana porque se han implementado algoritmos con una alta fiabilidad y un bajo tiempo de respuesta. Estas circunstancias han dado lugar a que se haya implantado este tipo de interacción en una gran diversidad de aplicaciones que hasta ahora han estado utilizado otros tipos de interacción, como es el caso de los dispositivos móviles, coches o casas domóticas. A continuación se presentan algunas de las aplicaciones más relevantes que hacen uso de este tipo de reconocimiento.

Domótica

En la actualidad, la población de avanzada edad es muy elevada [Shanas et al., 2017] y la domótica está orientada a ayudar a estas personas a controlar todos los elementos del hogar automatizados de una forma fácil y cómoda para el usuario. Algunos de los proyectos desarrollados en esta temática se muestran a continuación.

En [Sangeetha et al., 2015] es un sistema domótico que se basa en el reconocimiento de la voz para controlar los electrodomésticos mediante una aplicación de Android a través de una red inalámbrica. El sistema se compone de un microcontrolador en el que los electrodomésticos están conectados con dicho microcontrolador. El smartphone se

[14]Dragon Mobile SDK - `https://developer.nuance.com/public/Help/DragonMobileSDKReference_iOS_1.4.21/Introduction.html`
[15]Speech Google API - `https://cloud.google.com/speech-to-text?hl=es`
[16]Microsoft Speech API - `https://azure.microsoft.com/es-es/services/cognitive-services/speech-to-text/`

comunica con el servidor web a través de Internet y envía la señal al microcontrolador con lo que no es necesario que el usuario se sitúe en una localización concreta para hacer uso del sistema. Además, la tasa de acierto de la clasificación de los sonidos usando un clasificador máquinas de vectores soporte ha sido de un 96 %. En este trabajo [Katsamanis et al., 2014] se presenta un enfoque integrado para la localización de palabras clave y el reconocimiento de voz al combinar un modelado robusto con el objetivo de reducir el desajuste entre las condiciones de entrenamiento y prueba y un procesamiento multicanal. La selección de los canales más fiables basados en la relación señal / ruido (SNR), y luego la combinación de estos canales a través de N-best list rescoring, se realiza para obtener un sistema de reconocimiento de voz más fiable.

Los sistemas de reconocimiento de voz son también muy útiles para las personas que sufren de discapacidad física como se puede observar en este estudio [Kumar and Shimi, 2015] que presenta un sistema domótico en el cual las personas con discapacidad física como tetraplejia pueden controlar diversos electrodomésticos y accionar la elevación de la cama mediante comandos de voz. Los componentes de este sistema son: un módulo de reconocimiento de voz, un microcontrolador Arduino, un circuito de relé y una cama con reclinación ajustable. La desventaja principal de este desarrollo es que es necesario calibrar el sistema antes de usarlo para que sea capaz de reconocer los comandos por voz que diga el usuario.

Sistemas de autenticación

La ciberseguridad es muy importante porque ahora la mayoría de procesos se hacen de manera online, como por ejemplo el acceso a la cuenta bancaria. La obtención de credenciales de un cliente puede ser fatídico tanto para el propio cliente como para la empresa que se encarga de gestionar dicha información. Por lo tanto, es necesario estar seguros de que la persona que se autentica en el sistema es la correcta y los métodos tradicionales de usuario y password se están sustituyendo por otros con una menor posibilidad de riesgo. La voz tiene un papel importante en esta situación ya que es una característica particular y única de cada ser humano y por este motivo es usada para poder identificar o autenticar un individuo. Por este motivo se utiliza en sistemas biométricos para la identificación de usuarios, al igual que se hace por ejemplo con la huella dactilar o el reconocimiento facial.

En [Dovydaitis et al., 2016] se plantea la posibilidad de autenticar al usuario mediante el uso de la biometría de voz y el reconocimiento de la misma. El proceso de autenticación consiste en que el usuario diga su número de identificación y el sistema se encarga de hacer una predicción para comprobar que efectivamente la voz coincide con la del usuario identificado. El proceso de reconocimiento de la voz se realiza con la herramienta HTK donde los Modelos Ocultos de Markov creados para este caso tienen 60 estados ocultos. La tasa de acierto de este reconocimiento no es muy alta alcanzando solamente el 70,45 % pero los autores aseguran que la tasa de acierto del proceso de identificación es superior al 95 %. Boles et al [Boles and Rad, 2017] han desarrollado un sistema de autenticación mediante el reconocimiento de voz con la particularidad de que se han basado en coeficientes cepstrales en la frecuencia de Mel (MFCC). Después de extraer las características de estos coeficientes, se incluyen en la entrada de la Máquina de Vectores

de Soporte para entrenar este algoritmo y obtener como resultado la clasificación para determinar si coincide la voz y els sistema de autenticación permite el acceso al individuo. Uno de los factores principales cuando se utiliza este tipo de coeficientes es determinar el número de coeficientes que son necesarios para que el sistema funcione de manera óptima. En este caso los autores han concluido que tomando los veinte coeficientes más bajos se han obtenido los mejores resultados en los experimentos realizados con esta propuesta.

La situación actual del COVID-19 que está afectando a nivel global está haciendo que algunos procesos y trámites se cambien para garantizar la seguridad del usuario. Un ejemplo de ello es la enseñanza donde los alumnos por períodos de tiempo y sobre todo en la enseñanza universitaria se han visto obligados a quedarse en casa y realizar sus clases online. Aunque esta situación es difícil para los estudiantes también lo es para los profesores que tienen que evitar fraudes como por ejemplo que los estudiantes se copien en el examen o suplantación de identidad para realizar el mismo por otra persona distinta. Este estudio [Rudrapal et al., 2012] puede resultar útil para la problemática expuesta anteriormente, aunque el objetivo sea la identificación de usuarios que tienen una discapacidad física para acceder a un examen online y no puede hacer uso de sus extremidades para realizar la autenticación en el dicho examen. La solución que han propuesto los autores ante estas circunstancias es realizar el proceso de autenticación mediante el reconocimiento de su voz y así asegurar que es el propio estudiante quien realiza el examen. Este sistema recoge como entrada la voz del usuario que la compara con la muestra almacenada en la base de datos para a continuación comparar las ondas de ambas muestras con la transformada de Fourier y algunas funciones matemáticas con el fin de commprobar el porcentaje de similitud y concluir si los datos de entrada corresponden con el usuario.

Robótica

El reconocimiento de la voz es el medio mmás característico de comunicación del ser humano y como consecuencia despierta interés en el mundo de la robótica. Los robots tienen cada vez más precisión en la ejecución de sus tareas y ayudan y facilitan las labores de las personas en multitud de aspectos haciendo su vida más cómoda. Además, es cada vez más común fabricar robots con parecido humano, conocidos como los robots humanoides con lo cual es muy útil implementar un sistema de reconocimiento de voz en ellos para que puedan reconocer órdenes y actuar en consecuencia.

En una operación quirúrgica es muy útil que el cirujano disponga de sus manos libres y no las utilice durante la intervención para tareas de menor importancia o por el hecho de que en esos momentos ganar tiempo o perderlo puede ser crucial para la vida del paciente. Por este motivo el reconocimiento de voz puede ser un aliado significativo en estar circustancias y Zinchenko et al [Zinchenko et al., 2017] han implementado un algoritmo de reconocimiento de voz utilizando CMU Sphinx para controlar el soporte de un endoscopio robótico HIWIN. El reconocimiento de voz se realiza mediante un HMM con la finalidad de entrenar patrones de habla específicos y extraer palabras de una señal acústica para presentarlas en forma de coeficientes de cepstrales en la frecuencia de Mel.

En este trabajo [Wang et al., 2016] se presenta una interfaz de voz centrada en el

usuario diseñada específicamente para personas de avanzada edad. El objetivo es que estos usuarios puedan utilizar dicha interfaz en un entorno robótico. En este proyecto la interfaz por voz ha sido integrada en una plataforma denominada Robot-Era que cuenta con un conjunto de sistemas robóticos avanzados que son capaces de realizar ciertas tareas cotidianas como puede ser hacer la colada, limpiar o recoger la basura. Además se ha proporcionado un sistema de diálogo multilenguaje que tiene los siguientes elementos: el reconocedor y analizador de voz Nuance4speech comercial, el administrador de diálogo de código abierto Olympus5 y el sintetizador de voz Acapela6 de voz como servicio (*Voice as a Service*, VAAS).

Otro estudio dedicado a ayudar a las personas mediante el reconocimiento de voz es el realizado en la Universidad de Rostock (Alemania) [Ruzaij et al., 2016], cuyo propósito es que las personas que tienen discapacidad física y necesiten usar una silla de ruedas, puedan controlar dicha silla y un robot de rehabilitación con el uso de su voz. Para ello, se han implementado dos algoritmos: deformación dinámica del tiempo y HMM, lo que permite mejorar la precisión y reducir los errores del reconocimiento de voz.

2.2.3. Interfaz Cerebro-Ordenador

En la actualidad, la interfaz cerebro-ordenador (Brain-Computer Interface, BCI) es el modo de interacción más innovador. Esta interfaz capta las ondas cerebrales para interpretarlas y emitir una acción al sistema. La comunicación entre el dispositivo y el cerebro se realiza mediante el análisis de la actividad cerebral extraída a partir de una serie de técnicas: resonancia magnética funcional (fMRI), oximetría cerebral transcutánea (NIRS) y magnetoencefalografía (MEG). Sin embargo, el electroencefalograma (EEG) es la técnica más utilizada debido a que es menos invasivo, su relación calidad-precio y su portabilidad [Durka et al., 2012].

2.2.3.1. Dispositivos

En este apartado se van a presentar algunos de los dispositivos más usados en este tipo de interacción: OpenBCI Cyton Board, MindWave Mobile EEG Headset y EMOTIV EPOC+.

OpenBCI Cyton Board

OpenBCI Cyton Board[17] (ver Figura 2.18) es una placa diseñada para poder utilizar las características de la interfaz cerebro-ordenador de una forma económica ya que sus especificaciones y múltiples conexiones permiten medir la actividad cerebral, actividad muscular o el pulso cardíaco. Además, es compatible con la placa Arduino lo que permite controlar otro tipo de dispositivos como sensores de temperatura. En este proyecto [Alvarado-Díaz et al., 2017] la placa OpenBCI Cyton Board ha sido utilizada para que usuarios que carecen de movilidad sean capaces de controlar una silla de ruedas usando DL para la clasificación de sus ondas cerebrales.

[17]OpenBCI Cyton Board - http://openbci.com/

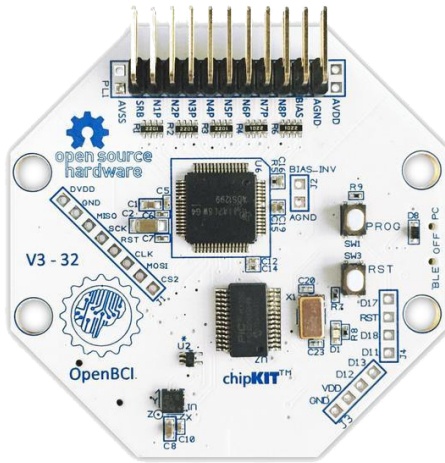


Figura 2.18: La placa OpenBCI Cyton. Fuente: [ope, 2021]

MindWave Mobile EEG Headset

MindWave Mobile EEG Headset[18] es un casco que permite monitorizar las ondas cerebrales del usuario. Este dispositivo es capaz de medir los niveles de meditación, atención, relajación. Estas medidas se obtienen a partir de un sensor localizado en la frente del usuario que tiene la capacidad de digitalizar las frecuencias de las ondas cerebrales. Además, este casco puede detectar el parpadeo de una persona. En este estudio [Navalyal and Gavas, 2014] el usuario es capaz de mover una bola 3D dependiendo de la concentración obtenida por este aparato.

EMOTIV EPOC+

EMOTIV EPOC+ [19] (Ver Figura 2.19b) es un casco formado por catorce electrodos que le permite saber la intención del usuario mediante 3 formas distintas: detección de expresiones faciales, el uso de un giroscopio para detectar el movimiento de la cabeza y detección de la actividad cerebral mediante entrenamiento previo. Aunque este dispositivo fue diseñado originalmente para el mundo de los videojuegos, se ha utilizado en diferentes ámbitos como por ejemplo para controlar un brazo robótico a través del pensamiento del usuario [Prince et al., 2015].

[18] MindWave Mobile EEG Headset - https://store.neurosky.com/
[19] EMOTIV EPOC+ - https://www.emotiv.com/epoc/

(a) MindWave Mobile EEG Headset. Fuente: [Narayana et al., 2019]

(b) EMOTIV EPOC+. Fuente: [emo, 2021]

Figura 2.19: Ejemplos de cascos diseñados para BCI.

2.2.3.2. Técnicas

En este apartado se van a describir las técnicas utilizadas para medir la actividad cerebral: Resonancia magnética funcional (fMRI), Oximetría cerebral transcutánea (NIRS), Magnetoencefalografía (MEG) y Electroencefalograma (EEG).

La **resonancia magnética funcional** obtiene datos de la actividad del cerebro cuando éste se encuentra realizando una tarea en concreto. Esta técnica se basa en utilizar un imán para averiguar la distribución de oxígeno en la sangre cuando el usuario realiza una tarea y de esta forma medir la cantidad de oxígeno que hay en una determinada zona del cerebro. Por lo tanto, se asocia un aumento de oxígeno en la sangre del cerebro con un incremento de actividad neuronal.

La principal ventaja es que es capaz de obtener información de las zonas más pronfudas del cerebro. Sin embargo, la más notable desventaja es que requiere un tiempo para procesar dicha información con lo que su velocidad de reacción es menor que la proporcionada por la EEG.

La **oximetría cerebral** (NIRS) mide el índice de saturación de oxígeno de la hemoglobina cerebral (SrO2) en una región determinada, sin ser invasivo. Esta técnica trabaja con la superficie de la corteza frontal, por este motivo se coloca un sensor en la región frontal de la cabeza del usuario que contiene un LED y dos detectores de superficie. Este sensor es capaz de monitorizar la saturación regional de oxígeno de los tejidos subyacentes a nivel capilar. La principal ventaja de esta técnica es que no es invasiva y no requiere la presencia de pulsatilidad.

La **magnetoencefalografía** (MEG) es una prueba que consiste en detectar y analizar los campos magnéticos que son generados por corrientes eléctricas en el cerebro. Estas corrientes eléctricas son el resultado del voltaje eléctrico que generan las neuronas cuando se comunican entre ellas. Para este fin, se hace uso de un dispositivo superconductor de interferencia cuántica (SQUID) y un ordenador para medir la actividad

neuromagnética del cerebro. MEG es un método muy avanzado para evaluar la actividad cerebral e identificar las áreas funcionales del cerebro y el origen de ataques epilépticos. Las principales ventajas de este método son que es una técnica no invasiva en la cual no hay exposición a radiación ionizante y es un estudio muy preciso de la actividad cerebral.

El **electroencefalograma** (EEG) es el registro de la actividad eléctrica del cerebro. En este proceso se colocan en una forma estándar los electrodos sobre el cuero cabelludo para captar las señales eléctricas y enviarlos a un ordenador donde están almacenadas. La comparación de las señales eléctricas entre diferentes electrodos hace que se obtengan una serie de trazados que ayudan a estudiar determinadas áreas del cerebro. En definitiva, este método está indicado para observar el funcionamiento eléctrico cerebral. Las principales ventajas son que es sencillo de realizar, económico y no invasivo.

2.2.3.3. Aplicaciones

Este sistema de interacción es más reciente y no hay tantos estudios realizados como en los otros tipos de interacción. Las aplicaciones de BCI más destacadas son el entretenimiento y la diversidad funcional, como se describirá a continuación.

Entretenimiento

Hoy en día se está comprobando que los juegos no son solo un mero entretenimiento para pasar el tiempo, sino que se están utilizando para otros ámbitos además del entretenimiento. Un ejemplo de esta afirmación son los serious games cuya finalidad radica en el aprendizaje de habilidades concretas y se puede aplicar desde en el colegio para enseñar matemáticas [Avila-Pesantez et al., 2018] hasta en un hospital para que los cirujanos mejoren sus habilidades en cirugía [Piedra et al., 2016]. En este ámbito es también importante considerar el modo de interacción que se utiliza en el juego para ofrecer la mejor experiencia al usuario y actualmente BCI se está considerando como una forma de controlar el juego potencial.

Esa tendencia a pensar que el hecho de controlar los elementos del juego con el cerebro puede ser atractivo para los usuarios ha desencadenado la creación de un framework BCI para estas aplicaciones. Este prototipo de framework [Gürkök et al., 2012] tiene la utilidad de que los desarrolladores de juegos pueden seguir las pautas estipuladas en el mismo para incluir esta forma de interacción a sus desarrollos. Este proyecto tiene el objetivo de combinar el punto de vista de la comunidad de videojuegos junto con la perspectiva de la comunidad de BCI, por esta razón este framework se ha creado basado en dos descriptores: un descriptor que especifica la motivación del jugador (comunidad de videojuegos) y el otro especifica el paradigma de interacción (comunidad de BCI).

Sin embargo, los dispositivos de BCI no se utilizan siempre con el propósito de controlar alguna aplicación software, sino que estos senosres pueden prorporcionar información sobre el funcionamiento del cerebro que pueden ser útiles, como describe el siguiente trabajo. BRAVO (BRAin Virtual Operator) [Marchesi and Riccò, 2013] es un sistema usado para la visualización de contenido en un contexto de e-learning móvil pero con la peculiaridad de que tiene integrada una interfaz BCI. Este módulo BCI ha sido usado para obtener información sobre la actividad cerebral como el nivel de atención o

meditación, durante el aprendizaje. Esta información extraída del dispositivo permite conocer los contenidos de una lección que le han resultado más difícil a un estudiante y de esta forma proponerlos de otra forma para que los entienda mejor usando elementos de gamificación como barras de progreso y medallas, entre otros.

La realidad virtual es uno de los modos de entretenimiento que está teniendo mucha cabida en los últimos años por su capacidad de inmersión que hace que el usuario tenga la sensación de que se encuentra en un mundo diferente a la realidad. Iidal et al [Iidal et al., 2017] han querido demostrar que BCI es compatible con la realidad virtual y para conseguir este fin han creado un sistema que tiene como componentes principales: un grabador de EEG, una tableta inteligente, un ordenador y una pantalla inmersiva montada en la cabeza. El desarrollo de un juego con el motor de juegos Unity, donde el usuario tenía que controlar un avatar a través de comandos, ha sido indispensable para validar esta hipótesis. Los experimentos fueron realizados con nueve sujetos que probaron el prototipo y después rellenaron un cuestionario sobre la experiencia. Finalmente, los autores corroboraron la hipótesis a raíz de los resultados del experimento, concluyendo que BCI es apto para ser integrado en entornos de realidad virtual.

Diversidad funcional

En [Fan et al., 2015] se ha desarrollado un simulador de conducción basado en realidad virtual dirigido a personas con trastorno del espectro autista para el entrenamiento de habilidades de conducción. El objetivo principal de este estudio es detectar el nivel de compromiso, los estados emocionales y la carga de trabajo mental durante la conducción con la ayuda de las EEG obtenidas de la interfaz BCI. Para completar dicho objetivo se adquirieron ciertas características de las EEG y se realizó una clasificación con los siguientes siete métodos: redes Bayesianas , naïve Bayes, SVM, perceptrón multicapa, K-vecinos más cercanos (KNN), random forest y J48, donde el algoritmo KNN obtuvo la mejor precisión en los experimentos que se realizaron. Este trabajo [Sreeja et al., 2016] persigue conseguir que las pesonas con discapacidad física puedan utilizar este sistema para introducir texto solo con la intención de usar la mano izquierda o derecha mediante las señales cerebrales. El proceso consta de cinco fases donde se recogerán las señales EEG para extraer 14 características de dichas señales y así realizar un entrenamiento con Máquinas de Vectores de Soporte para realizar una clasificación cuyo resultado serán los comandos necesarios para controlar la aplicación de de entrada de texto usando BCI. Esta propuesta permitirá que los usuarios que no puedan utilizar los métodos tradicionales de entrada de datos en sistemas informáticos debido a su movilidad reducida puedan hacerlo con ayuda de las ondas que transmite su cerebro.

Capítulo 3

Análisis bibliométrico

Capítulo 3

ANÁLISIS BIBLIOMÉTRICO

Contenidos

En este capítulo se expone un análisis bibliométrico basado en los campos de investigación de reconocimiento de gestos e inteligencia artificial, más específicamente enfocado a los subcampos de DL y ML. Este estudio permite que se tenga conocimiento de las palabras claves más relevantes de la temática del análisis, los autores más representativos, así como las instituciones y los artículos más importantes en relación al reconocimiento de gestos y la inteligencia artificial. A continuación se muestran los datos más significativos de este estudio.

3.1. Metodología

La identificación de los elementos claves del campo de investigación del reconocimiento gestual y la inteligencia artificial a través de herramientas de ML y DL sigue la técnica de análisis bibliométrico, que muestra información relevante sobre autores, instituciones, documentos y palabras clave [Cobo et al., 2011, Morris and Van der Veer Martens, 2008]. Este estudio bibliométrico es una consecución de cinco pasos:

(1.) Definición del campo de investigación.

(2.) Selección de la base de datos.

(3.) Ajuste de los criterios de investigación.

(4.) Codificación del material recuperado.

(5.) Examen de la información.

De esta manera, el proceso gana en claridad y podría ser reproducible (ver Figura 3.1).

El primer paso es la identificación del foco central de este estudio, con el fin de mostrar la información relativa a la producción científica (análisis de continente) y el análisis de la co-ocurrencia de palabras clave de este campo de investigación (análisis de contenido). El segundo paso es la selección de la base de datos. Teniendo en cuenta que los resultados del análisis podrían variar en función de la base de datos seleccionada, y en consonancia con [Agramunt et al., 2020], en este estudio se utilizan las dos fuentes de datos bibliométricos más utilizadas: Web of Science (WoS, producida por Clarivate Analytics) y Scopus (creada por Elsevier). Aunque el buscador de Google podría ofrecer una cobertura adicional a la WoS y Scopus, tiene ciertos problemas asociados. En primer lugar, enumera una gran cantidad de fuentes no académicas, incluida la literatura gris que no es revisada por pares [Kraus et al., 2020]. En segundo lugar, el algoritmo de búsqueda no es reproducible, ya que los resultados se muestran sobre la base de búsquedas e interacciones anteriores [Gusenbauer and Haddaway, 2020]. Tercero, es difícil de utilizar para el análisis a gran escala [Waltman and Noyons, 2018]. Por lo tanto, las limitaciones mencionadas anteriormente nos han disuadido de incluirlo en nuestro análisis.

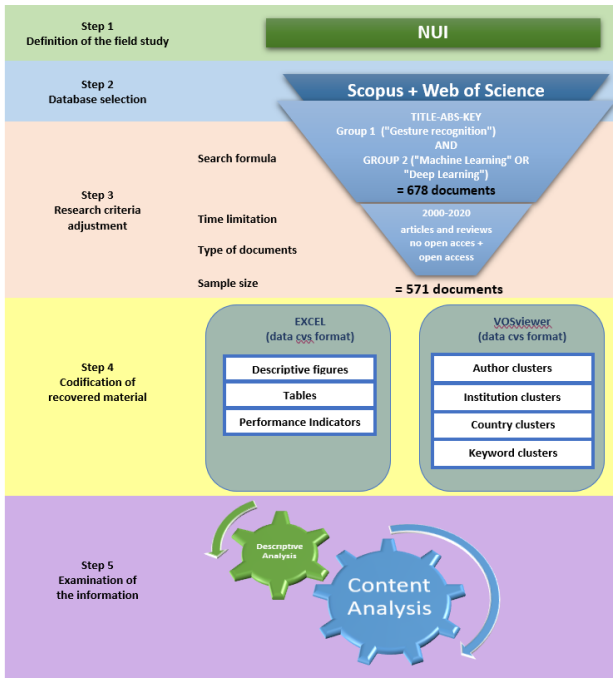


Figura 3.1: Diagrama de flujo de la metodología del bibliométrico.

Una vez seleccionadas las bases de datos, el siguiente paso es el ajuste de los criterios de investigación. En esta etapa se establecen los criterios de investigación con los operadores booleanos para obtener una búsqueda precisa y facilitar la captura de datos de gran tamaño. En consecuencia, los parámetros utilizados para recuperar la búsqueda fueron:

```
TITLE-ABS-KEY (‘‘gestur* recognition’’ OR ‘‘hand* recognition’’ OR ‘‘bod*
recognition’’ OR ‘‘fac* recognition’’ OR ‘‘hand* gestur*’’ OR ‘‘bod*
gestur*’’ OR ‘‘fac* gestur*’’ OR ‘‘gestur* interact*’’ OR ‘‘gestur* based
interact*’’ OR ‘‘hand* interact’’ OR ‘‘bod* interact*’’ OR ‘‘fac*
interact*’’ OR ‘‘gestur* detect*’’ OR ‘‘hand* detect*’’ OR ‘‘bod*
detect*’’ OR ‘‘fac* detect*’’ OR ‘‘gestur* model*’’ OR ‘‘gestur*
classif*’’) AND (‘‘machine learning’’ OR ‘‘support vector machine’’ OR
‘‘decision tree’’ OR ‘‘k-nearest neighbor’’ OR ‘‘naive bayes’’ OR
‘‘random forest’’ OR ‘‘hidden markov model’’ OR ‘‘dynamic time
warping’’ OR ‘‘bayesian network’’ OR ‘‘k-means’’ OR ‘‘artificial
neural network’’ OR ‘‘deep learning’’ OR ‘‘convolutional neural
network’’ OR ‘‘recurrent neural network’’ OR ‘‘long short-term
memory network’’ OR ‘‘deep belief Network’’)
```

La búsqueda se limitó al período 2000-2020. El primer documento sobre este tema es el

artículo de Ng C.W. y Ranganath S. titulado "*Gesture recognition via pose classification*", publicado en 2000. Este artículo describe cómo se puede entrenar un sistema de reconocimiento de gestos mediante la estimación de poses de las manos [Ng and Ranganath, 2000]. Esta estrategia que incluye las poses de las manos al entrenamiento del sistema permite reducir los tiempos de entrenamiento y reconocimiento ofreciendo la posibilidad de aplicar el sistena en tiempo real. Sin embargo, dichos autores no solamente fueron los pioneros en conjugar ambos campos de conocimiento a través de un único artículo, sino que posteriormente en 2002 publicaron "*Real-time gesture recognition system and application*" donde también se basaron en las poses de la manos para crear un sistema basado en visión que permitía controlar los objetos de una interfaz mediante gestos con una tasa de acierto del 91.9 % [Ng and Ranganath, 2002].

La búsqueda en ambas bases de datos (Scopus + Web of science) se llevó a cabo a finales de septiembre de 2020. En cuanto a los criterios de inclusión y exclusión, solo se consideraron artículos, libros y capítulos de libros, incluidos documentos de acceso abierto y de acceso no abierto [Capobianco-Uriarte et al., 2019].

En la Tabla 3.1 se resumen, se identifican y cuantifican de forma global las variables de contenido que se analizarán posteriormente más en profundidad. El número de documentos encontrados en exclusiva en WoS fue de 213 y de Scopus 221, aunque había 137 en común. Por lo tanto, la muestra final consistió en 570 documentos (ver Figura 3.2).

Datos	**Resultados**
Número de artículos	571
Número de citas	9,366
Número de palabras clave	2,897
Número de revistas	244
Númnero de autores	2,167
Número de instituciones	721
Número de países	66
Periodo de estudio	2000 - 2020
Fuentes de datos	Scopus - Web of Science

Tabla 3.1: Resumen de los datos usados en el estudio de Reconocimiento de gestos e Inteligencia Artificial.

El cuarto paso es la codificación del material recuperado, que se descargó en formato CSV y se codificó utilizando Excel y VOSviewer [Eck and Waltman, 2010]. Los datos se procesaron previamente para el análisis posterior. En primer lugar, se suprimieron los documentos duplicados que figuraban en ambas bases de datos. En segundo lugar, se examinaron el resumen y el título de cada documento para asegurarse de que cumplían los criterios de búsqueda. En tercer lugar, se corrigieron los documentos con información que faltaba.

Por último, el último paso es el examen de la información. Esta fase se lleva a cabo utilizando dos técnicas de análisis bibliométrico: el análisis de rendimiento y el mapeo científico [Cobo et al., 2011]. En primer lugar, siguiendo los estudios anteriores [Baier-Fuentes et al., 2020, Terán-Yépez et al., 2020], el análisis de rendimiento se basa en la

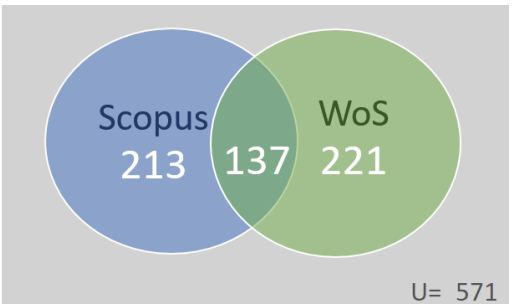


Figura 3.2: Distribución de los artículos publicados en las bases de datos usadas.

productividad, teniendo en cuenta el número de publicaciones como indicador principal. Además, el número de citas y el índice h se utilizan para enriquecer el análisis del rendimiento, a nivel de autores, revistas, instituciones y países. Su principal objetivo es ofrecer un panorama actualizado del campo de la investigación mediante la identificación de las obras que constituyen su base intelectual [Alayo et al., 2020]. En segundo lugar, el mapeo científico tiene por objeto desvelar la estructura y la dinámica de los campos científicos [Zupic and Čater, 2015]. Se trata de una representación espacial de la forma en que se relacionan entre sí las disciplinas, los campos, los autores y las obras [Alayo et al., 2020]. Este enfoque metodológico se adapta a los propósitos del presente estudio. Por lo tanto, para examinar diferentes aspectos interesantes del campo de investigación, realizamos un mapeo científico basado en análisis de co-autoría y co-escritura.

Por una parte, el análisis de coautoría permite identificar la red social de un campo de investigación a través de los vínculos entre sus autores más relevantes y los subgrupos que surgen de las colaboraciones a nivel de instituciones y países [Acedo et al., 2006]. Esta técnica capta vínculos sociales más fuertes que otras medidas de relación, lo que la hace ideal para examinar las redes sociales [Zupic and Čater, 2015]. Por otra parte, el análisis de palabras clave en conjunto o en concurrencia permite establecer la estructura intelectual de un campo científico dividiendo las palabras en diferentes grupos [Zupic and Čater, 2015]. Concretamente, este método permite relacionar y clasificar cuantitativamente el contenido conceptual de las publicaciones en función de la aparición de pares de palabras similares [Bhattacharya and Basu, 1998].

En otros términos, la co-ocurrencia de palabras clave ayuda a identificar un dominio de investigación a través de las conexiones específicas realizadas entre sus palabras clave [Callon et al., 1983, López-Fernández et al., 2016]. De hecho, el análisis de la coocurrencia de palabras clave es particularmente apropiado para este propósito en comparación con otros métodos bibliométricos, como el análisis de la cocitación y el acoplamiento bibliométrico, ya que estos últimos no son óptimos para la cartografía de los frentes de investigación debido, entre otras razones, a que las citaciones necesitan tiempo para acumularse y, por lo tanto, es más difícil conectar directamente las publicaciones más recientes a través de grupos de bases de conocimientos [Zupic and Čater, 2015]. Además, las palabras clave de un artículo reflejan su contenido principal, y la frecuencia de su aparición y coocurrencia representan los temas más importantes que abordan los trabajos

en un área de investigación y la forma en que se vinculan entre sí [Alayo et al., 2020]. La suma de todas las co-ocurrencias conjuntas entre las palabras clave permite establecer un mapa en red de la matriz de co-ocurrencia, que permite el reconocimiento de varios grupos o clusters temáticos. Por último, se elaboran mapas para diferentes períodos de tiempo a fin de trazar los cambios en el espacio conceptual del campo [Coulter et al., 1998] y las futuras vías de investigación.

A pesar de la aparición de este campo de conocimiento desde el comienzo del actual milenio, su despegue en producción científica puede aproximarse a 2015. No obstante, dentro de la producción científica se pueden contabilizar más de 22 artículos de revisión que se detallan a continuación en las Tablas 3.2 y 3.3, pero ninguno de ellos lleva a cabo un estudio bibliométrico.

Autores / Año	**Título**	**Reconocimiento de gestos**	**Inteligencia Artificial**
Arac A, 2020	A review of recent approaches for emotion classification using electrocardiography and electrodermography signals	Facial expression recognition	Support Vector Machine; K-Nearest Neighbour; Random Forest
Bulagang AF et al, 2020	Facial Expression Recognition Using Computer Vision: A Systematic Review	Facial expression recognition	Convolutional Neural Network; Support Vector Machine; K-Nearest Neighbour; Naive Bayes; Hidden Markov Model; Decision Tree; Random Forest
Canedo D & Neves AJR, 2019	Continuous authentication using biometrics: An advanced review	Face recognition	Convolutional Neural Network
Dahia G et al, 2020	Convolutional neural network: a review of models, methodologies and applications to object detection	Human activity recognition	Convolutional Neural Network
Fang YT et al, 2020	Visual Object Recognition: Do We (Finally) Know More Now Than We Did?	Face recognition	Convolutional Neural Network
Hu TH et al, 2017	Memristor devices for neural networks	None	Spiking Neural Network; Artificial Neural Network
Jeong H & Shi LP, 2019	A review of image-based automatic facial landmark identification techniques	Facial recognition	Convolutional Neural Network
Johnston B & de Chazal P, 2018	A survey on security threats and defensive techniques of machine learning: A data driven view	None	Naive Bayes; Logistic Regression; Decision Tree; Support Vector Machine; Deep Neural Network
Liu Q et al, 2018	A Review on Automated Facial Nerve Function Assessment from Visual Face Capture	Facial expression recognition	Support Vector Machine; K-Nearest Neighbour; Artificial Neural Network
Lou J et al, 2020	Artificial Intelligence in Medical Practice: The Question to the Answer?	None	Convolutional Neural Network

Tabla 3.2: Revisiones sistemáticas de la literatura más relevantes (Parte I).

Autores / Año	Título	Reconocimiento de gestos	Inteligencia Artificial
Miller DD & Brown EW, 2018	Face Space Representations in Deep Convolutional Neural Networks	Face recognition	Deep Convolutional Neural Network
O'Toole AJ eta l, 2018	Primer on machine learning: Utilization of large data set analyses to individualize pain management	Facial expression recognition	Deep Neural Network
Rashidi P et al, 2019	A multimodal approach for automatic detection of infant pain using facial expression and crying	Facial expression recognition	Convolutional Neural Network; Long Short Term Memory
Sandeep PVK & Suresh Kumar N, 2020	Artificial intelligence in radiation oncology: A specialty-wide disruptive transformation?	None	None
Thompson RF et al, 2018	Deep Learning for Computer Vision: A Brief Review	Face recognition; Activity recognition; Human pose estimation	Convolutional Neural Network; Deep Belief Network; Deep Boltzmann Machine; Stacked Denoising Autoencoders
Voulodimos A et al, 2018	Face Feature Extraction: A Complete Review	Face recognition	Convolutional Neural Network; Deep Belief Network
Wang HJ et al, 2018	Hand Gesture Recognition in Automotive Human-Machine Interaction Using Depth Cameras	Hand gesture recognition	Long Short-Term Memory; Convolutional Neural Network
Zengeler N, 2019	Review of Convolutional Neural Network		
Zhou FY eta l, 2017	Deep Learning: The Good, the Bad, and the Ugly	Face recognition; Action recognition	Convolutional Neural Network; Generative Adversarial Network

Tabla 3.3: Revisiones sistemáticas de la literatura más relevantes (Parte II).

3.2. Resultados

En los últimos años la Inteligencia Artificial ha sido un campo de conocimiento muy significativo y útil en diferentes disciplinas para las cuales ha supuesto un gran avance en sus investigaciones. La Visión Artificial ha sido una de las cuales ha resultado beneficiada de las propiedades de la Inteligencia Artificial y por ende en la Interacción Natural cuando se utilizan elementos multimedia como imágenes y vídeos aplicados a este tipo de interacción.

En la Tabla 3.4 se muestran algunos de los principales indicadores productivos de los documentos publicados por año, como el número de documentos, el promedio de citas (C/A), el número de autores, el promedio de autores por artículo (AUA), el número de revistas (RA) y los países (PA) que publicaron al menos 1 artículo en un año determinado.

En lo que respecta al número de artículos, se observa en la Figura 3.3a que a partir del año 2014 hay un interés por este área de conocimiento por parte de la comunidad científica destacando el incremento considerable de documentos en los dos últimos años donde en 2018 se redactaron menos de 50 publicaciones hasta el 2020 con 180 publi-

Año	A	C	C/A	AU	AUA	IA	RA	PA
2000	1	20	20.00	2	2	1	1	1
2001	0	-	-	-	-	-	-	-
2002	1	96	96.00	4	4	1	1	1
2003	0	-	-	-	-	-	-	-
2004	1	3	3.00	6	6	1	1	1
2005	3	65	21.67	12	12	3	3	3
2006	0	-	-	-	-	-	-	-
2007	1	73	73.00	5	5	1	1	1
2008	0	-	-	-	-	-	-	-
2009	1	5	5.00	8	8	1	1	1
2010	0	-	-	-	-	-	-	-
2011	0	-	-	-	-	-	-	-
2012	1	253	253.00	4	4	1	1	1
2013	1	2,326	2,326.00	8	8	8	2	4
2014	5	80	16.00	14	14	6	4	5
2015	6	1,844	307.33	17	17	13	6	2
2016	24	1,077	44.88	104	103	40	19	15
2017	38	1,226	32.26	151	149	60	28	14
2018	110	1,588	14.44	439	434	159	65	28
2019	178	603	3.39	706	672	289	115	48
2020*	198	107	0.54	794	727	319	95	47
2021**	1	0	0.00	2	2	1	1	1

*hasta julio incluido
**publicación temprana

Tabla 3.4: Principales características de los datos usados (A: Número de artículos; C: Citas; C/A: Citas por artículo; AU: Número de autores; AUA: Promedio de autores por artículo; IA: Instituciones; RA: Revistas por artículo; PA: Países que han publicado al menos un artículo).

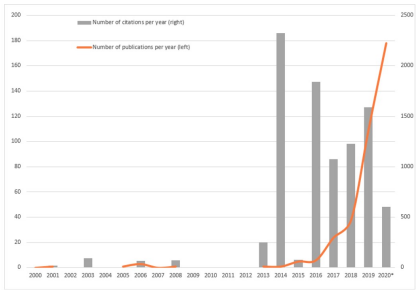
caciones. Además, el análisis del número de citas reveló que 2014 es el año con mayor número de citas (180). Respecto al número de autores se puede apreciar en la Figura 3.3b que el número de los mismos que han publicado artículos sobre el tema (AUA) ha aumentado exponencialmente a partir de 2017 con 200 autores implicados en 2020, lo que demuestra un interés creciente y un número cada vez mayor de colaboraciones entre los autores, tratando de llenar el vacío de investigación en este campo. Los datos obtenidos demuestran que este amplio campo de investigación ha ido acompañado de un crecimiento constante del número de revistas y países que publican artículos.

En la distribución de los documentos incluidos en este análisis bibliométrico, un 93,52 % corresponde a articulos, mientras que un 3,85 % son reviews y el resto esta formado por articulos publicados en ediciones especiales de ponencias, libros y capitulos de libros (ver Figura 3.3c).

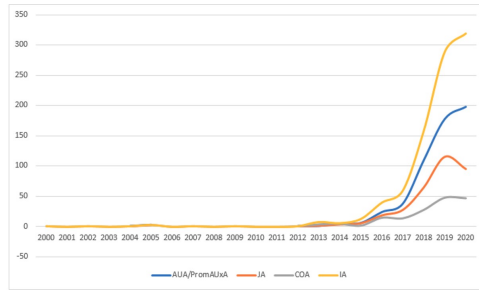
En cuanto a las revistas, en la Tabla 3.5 se presentan indicadores bibliométricos adicionales, como las citas, el promedio de citas por artículo, el año de la primera publicación, el año de la última publicación y el índice h. La revista más relevante en cuanto a número de artículos es IEEE Access con 62 artículos (A), mientras que si hacemos alusión a las citas y a las citas por artículo la revista más destacada es IEEE Transactions on Pattern Analysis and Machine Intelligence con 4.440 citas (C) y 444.00 citas

por artículo (C/A) durante el período de 2013 a 2020, correspondiente al año del primer artículo publicado y del último. En segundo lugar, se encuentra Neurocomputing con 45 artículos, seguida de Multimedia Tools and Applications con 23. Aunque, teniendo en cuenta índice h, Multimedia Tools and Applications se convierte en la revista con mayor impacto con un índice h de 43.

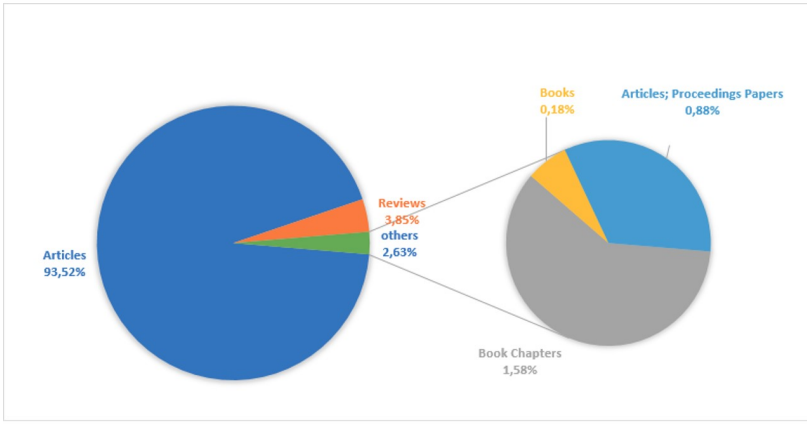
En cuanto a los autores, la Tabla 3.6 muestra los autores más productivos del área. Estos autores provienen principalmente de Corea del Sur y la institución llamada Universidad Dongguk. Es necesario mencionar que la afiliación indicada en la Tabla 3.6 pertenece al indicado en el momento de la publicación del último documento. Todos los autores han publicado el mismo número de artículos con un total de 4 artículos, con Zhan, Shu siendo el autor más citado y el que tiene el mayor número de citas por artículo entre 2016 y 2020. Este autor estaba afiliado a la Universidad Tecnológica de Hefei.



(a) Evolución de los artículos publicados y citas.



(b) Datos relacionados con la publicación de los artículos.



(c) Tipos de documentos incluidos.

Figura 3.3: Datos relevantes de los artículos publicados desde 2000 a 2020.

Orden	Revista	A	C	C/A	PA	UA	Índice-h
1	IEEE Access	62	147	2.37	2018	2020	6
2	Neurocomputing	45	819	18.20	2014	2020	14
3	Multimedia Tools and Applications	23	43	1.87	2017	2020	43
4	Sensors	18	545	30.28	2016	2020	6
5	IEEE Transactions on Image Processing	13	250	19.23	2017	2020	5
6	Computational Intelligence and Neuroscience	10	230	201.60	2016	2019	4
7	IEEE Transactions on Pattern Analysis and Machine Intelligence	10	4,440	444.0	2013	2020	9
8	IEEE Transactions on Multimedia	9	76	8.44	2016	2020	4
9	Applied Sciences-Basel	8	12	1.33	2019	2019	2
10	IEEE Transactions on Information Forensics and Security	8	34	4.25	2014	2020	2

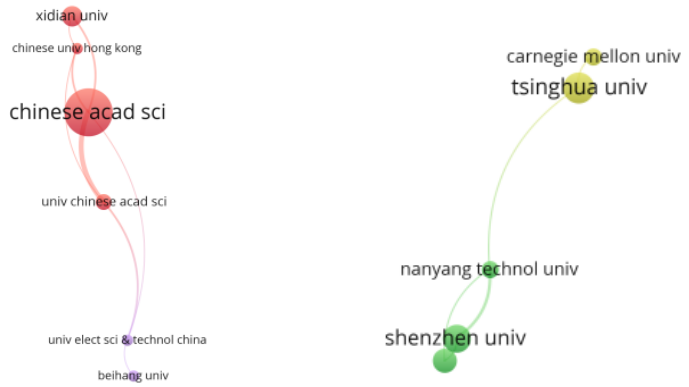
Tabla 3.5: Las diez revistas más productivas (A: Número de árticulos; C: Citas; C/A: Citas por artículos; PA: Primer artículo; UA: Último artículo).

Orden	Autores	A	C	C/A	PA	UA	Índice h	País	Afiliación
1	Zhan, Shu	4	63	15.75	2016	2020	2	China	Hefei Univ Technol
2	Park, Kang Ryoung	4	39	9.75	2017	2019	3	Corea del Sur	Dongguk Univ
3	Pham, Tuyen Danh	4	39	9.75	2017	2019	3	Corea del Sur	Dongguk Univ
4	Nguyen, Dat Tien	4	39	9.75	2017	2019	3	Corea del Sur	Dongguk Univ

Tabla 3.6: Los autores más productivos (A:Número de artículos; C: Citas; C/A: Citas por artículo; PA: Primer artículo; UA: Último artículo).

En la Figura 3.4 se muestran las redes internacionales de instituciones académicas, que tienen en común más de 5 estudios científicos entre los investigadores, con dos grupos identificados. El grupo principal (verde) es el más productivo, incluye tres universidades de procedencia china, destacando la institución académica más productiva (Tabla 3.7) Chinese Academy of Science, junto a la Xidian University y la University of Chinese Academy of Science. El otro grupo está formado por la Tsinghua University (República P. China), Shenzhen University (República P. China), la Carnegie Mellon University (EEUU) y la Nanyang Technological University (Singapur).

El análisis de las instituciones se muestra en la Tabla 3.7, en el que figuran las diez instituciones más productivas en esta área analizada desde 2000 hasta 2021. Estas instituciones están situadas todas en China, dato que no es extraño debido a que la mayoría de los autores que publican en este ámbito así como la producción científica más relevante procede de China. Según el número de artículos, la más productiva es la Chinese Academy of Science, que cuenta con 20 artículos desde 2016. La segunda institución más



(a) (b)

Figura 3.4: Red de los coautores basado en la cooperación entre instituciones

productiva es la University of Chinese Academy of Science con 10 artículos, desde 2017. La tercera institución académica es la Tsinghua University que cuenta también con 10 artículos. Según el número de citaciones por artículo (C/A), el primer lugar lo ocupa la University of Chinese Academy of Science con una proporción de 24.80 citaciones por artículo. En este sentido, la Tsinghua University ocupa el segundo lugar con 19.60 citas por artículo. Desde el punto de vista del índice h, comparten posición las universidades Tsinghua University y la Nanjing University of Science & Technology puesto que tienen un valor de 6 para este indicador. Como se puede deducir de los datos que muestra la tabla la institución que ha sido pionera en este campo de investigación ha sido la Sun Yat Sen University.

Orden	**Institución**	**País**	**A**	**C**	**C/A**	**PA**	**UA**	**Índice-h**
#1	Chinese Acad Sci	China	20	320	16.00	2016	2020	4
#2	Univ Chinese Acad Sci	China	10	248	24.80	2017	2020	3
#3	Tsinghua Univ	China	10	196	19.60	2015	2020	6
#4	Shenzhen Univ	China	10	72	7.20	2018	2020	4
#5	Nanjing Univ Sci & Technol	China	9	97	10.78	2017	2020	6
#6	Zhejiang Univ	China	9	71	7.89	2018	2020	3
#7	Xidian Univ	China	9	21	2.33	2018	2020	3
#8	Sun Yat Sen Univ	China	8	26	3.25	2014	2020	3
#9	South China Univ Technol	China	7	131	18.71	2015	2020	4
#10	Tianjin Univ	China	7	32	4.57	2018	2020	4

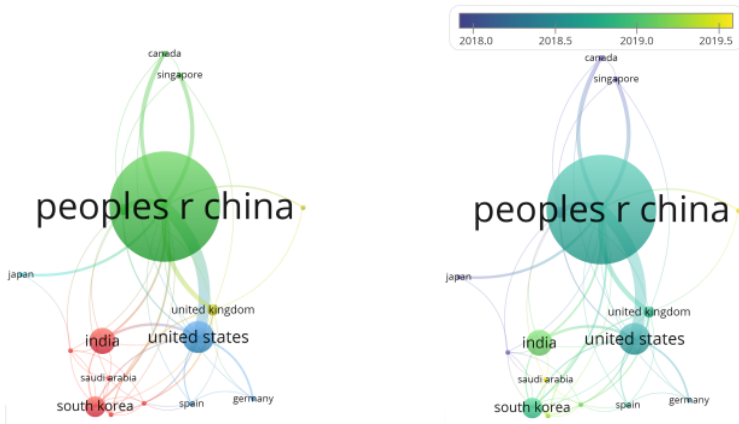
Tabla 3.7: Las diez instituciones más productivas (A: Número de artículos; C: Citas; C/A: Citas por artículo; PA: Primer articulo; UA: Último artículo).

En la tabla 3.8 se muestran los países que han publicado un mayor número de artículos sobre el reconocimiento de gestos usando técnicas del campo de Inteligencia Artificial,

donde se incluye la información relativa a las citas, los artículos, el índice h y el año de la primera y última publicación. Cabe señalar que un artículo puede representar a más de un país, ya que los países son establecidos por las instituciones afiliadas de los investigadores involucrados. Los países más influyentes en cuanto al número de artículos son China y Estados Unidos con 252 y 74 documentos respectivamente, seguidos por India (56) y Corea del Sur (47), como se puede verificar en la Figura 3.5. Además, teniendo en cuenta el número de artículos por cada millón de habitantes (AP), Australia alcanza el primer lugar con 1,04. Considerando el número de citaciones (C), destacan los documentos procedentes de China (6.325), aunque la mayor relación de citas por artículo la obtiene Estados Unidos, con 45.12 citas por documento. Si tenemos en cuenta el índice h, China lidera la clasificación con un índice de 26.

Orden	**País**	**A**	**P**	**AP**	**C**	**C/A**	**PA**	**UA**	**Índice-h**
#1	China	252	1,420,062,022	0.18	6325	25.10	2013	2020	26
#2	Estados Unidos	74	329,093,110	0.22	3339	45.12	2013	2020	17
#3	India	56	1,368,737,513	0.04	129	2.30	2016	2021	7
#4	Corea del Sur	47	51,339,238	0.92	189	4.02	2016	2020	7
#5	Australia	26	25,088,636	1.04	398	15.31	2005	2020	8
#6	Reino Unido	26	66,959,016	0.39	709	27.27	2013	2020	7
#7	Alemania	12	82,438,639	0.15	457	38.08	2016	2020	6
#8	Canadá	12	37,279,811	0.32	374	31.17	2012	2020	5
#9	Taiwán	12	23,758,247	0.51	56	4.67	2005	2020	3
#10	Pakistán	12	204,596,442	0.06	34	2.83	2018	2020	3

Tabla 3.8: Los diez países más productivos (A:Número de artículos; C: Citas; C/A: Citas por artículo; PA: Primer artículo; UA: Último artículo).



(a) (b)

Figura 3.5: Red de los coautores basado en países

En la Tabla 3.9 se muestra los diez artículos más relevantes asociados al campo de

conocimiento analizado en este trabajo. Dichos artículos han sido ordenados según el número de citas pero teniendo en cuenta el año de publicación para evitar la deficiencia que pueda suponer el hecho exclusivo de considerar las citas de un artículo. A continuación se van a presentar los datos más relevantes de cada uno de estos artículos:

(1.) En [Ji et al., 2012] se presenta una arquitectura basada en 3D-CNN para el human activity recognition. Esta arquitectura extrae las características espaciales y temporales de los vídeos que analiza y además tiene la particularidad de que genera varios canales de información de diferentes vídeos para finalmente combinar toda esa información y obtener las características.

(2.) La novedad de este trabajo [He et al., 2015] es entrenar una red neuronal con una capa espacial y piramidal de pooling, la cual muestra unos buenos resultados en tareas de detección y clasificación, así como una mayor rapidez. Este enfoque es más rápido que las R-CNN y lo hace apto para aplicaciones en el mundo real.

(3.) En [Ordóñez and Roggen, 2016] los autores han desarrollado un framework basado en DL denominado DeepConvLSTM que combina capas convolucionales y LSTM recurrentes, con la finalidad de reconocer actividades humanas procedentes de los datos extraídos por sensores portátiles. Entre las características principales de esta unión están que permite extraer las características automáticamente y modelar las dependencias temporales de la activación de sus neuronas.

(4.) En [Niu and Suen, 2012] se muestra un modelo híbrido formado por los clasificadores CNN y SVM para el reconocimiento de escritura. Este modelo ha sido creado sustituyendo la última capa de la CNN por un clasificador SVM para poder reconocer patrones desconocidos. Esta arquitectura ha presentado unos resultados prometedores debido principalmente a 3 razones: la extracción automática de las características, este modelo tiene las ventajas de las técnicas CNN y SVM y la reducción de la complejidad en el proceso de decisión respecto a otros modelos.

(5.) En [Côté-Allard et al., 2019] se presentan tres nuevas arquitecturas para CNN de clasificación de gestos basada en sEMG. Además, describe un nuevo paradigma de aprendizaje por transferencia (TL) que mejora el rendimiento de estas redes convolucionales. Este esquema se basa en el aprendizaje de dos CNN al mismo tiempo. La red a la que los autores hacen referencia como "fuente" comparte información con la segunda red a través de la suma de elementos, con la ventaja de que reduce los problemas de conectar ambas redes. Además, implementa un método que fomenta el aprendizaje residual, por lo que la segunda red solo tiene que aprender un número limitado de pesos. Se han probado dos conjuntos de datos donde se ha demostrado que este diseño de TL ha mejorado en cada una de las CNN propuestas, donde una de las redes ha logrado una precisión del 98,31 %..

(6.) En [Zhou et al., 2017] se presenta una revisión enfocada en DL ya que este área de conocimiento ha sido muy demandado a raíz de los resultados prometedores que ha tenido en tareas como reconocimiento de objetos y reconocimiento facial. Este estudio muestra la estructura y las características de los componentes de las

redes neuronales convolucionales, así como las diversas aplicaciones en las que se ha utilizado. Por último, se ha realizado unos experimentos donde se han evaluado las redes neuronales convolucionales que tenían diferentes estructuras y se han descrito las ventajas e inconvenientes de estas técnicas de DL.

(7.) En [Reichstein et al., 2019] se describe como los modelos de ML están reemplazando a los tradicionales modelos físicos en ciencias de la Tierra ya que cada vez se están extrayendo más datos de este campo de conocimiento y la ciencia de datos se van haciendo necesarias para tratar y analizar esta cantidad de volumen de datos.

(8.) En [Voulodimos et al., 2018] los autores han hecho una revisión de los principales desarrollos en arquitecturas de DL y algoritmos para Visión Artificial. Este trabajo se ha basado principalmente en 3 tipos de modelos de DL: Redes neuronales convolucionales (CNN), los modelos tipo Boltzmann y Stacked Autoencoders. Estas técnicas han sido elegidas porque han obtenido un excelente rendimiento en tareas de Visión Artificial como detección de objetos, reconocimiento facial, reconocimiento de actividades o estimación de la postura humana, entre otros.

(9.) En [Wu et al., 2016] se ha desarrollado un framework que combina redes neuronales con Modelos Ocultos de Markov (HMM) para el reconocimiento de gestos. Las redes neuronales usadas son una Gaussian-Bernoulli Deep Belief Network para extraer las características de las articulacioens del esqueleto y una red neuronal convolucional 3D para extraer características como la profundidad e imágenes RGB. La función del HMM es tener en cuenta dependencias temporales en el proceso de aprendizaje.

(10.) En [Wang and Deng, 2018] los autores se centran en analizar los métodos relacionados con deep domain adaptation (DA). En DA se están utilizando redes neuronales para mejorar el rendimiento de DA dando lugar a deep DA cuya clasificación se divide principalmente en DA homogéneo y DA heterogéneo aunque el artículo se centra especialmente en los enfoques one-step DA and multi-step DA. Además, este trabajo ofrece una perspectiva de aplicaciones para la vida real donde destacan aplicaciones centradas en Visión Artificial como por ejemplo reconocimiento facial o detección de objetos.

Después de tener conocimiento de los trabajos más relevantes sobre este tema, es necesario señalar que la TL es una técnica que ha aportado numerosas ventajas en general pero también al reconocimiento de gestos. Esta afirmación se basa en las ventajas que ofrece este método, donde las principales son el hecho de que es posible realizar una tarea con muchos menos datos que con el procedimiento tradicional, es más rápido en la ejecución del entrenamiento porque no es necesario entrenar el modelo completo sino solo una parte del mismo y su tasa de aprendizaje suele ser mayor porque ha sido previamente entrenado en una tarea similar [Olivas et al., 2009, Kocmi, 2020, Sarkar et al., 2018].

En este estudio se han determinado las ocurrencias de TL en los artículos que componen este trabajo, donde esta técnica se ha utilizado mayoritariamente en DL, especialmente en CNN, con un porcentaje del 89 % de todos los artículos que han aplicado TL y las Redes Neuronales Recurrentes llega al 4 %. Algunos de los estudios más relevantes

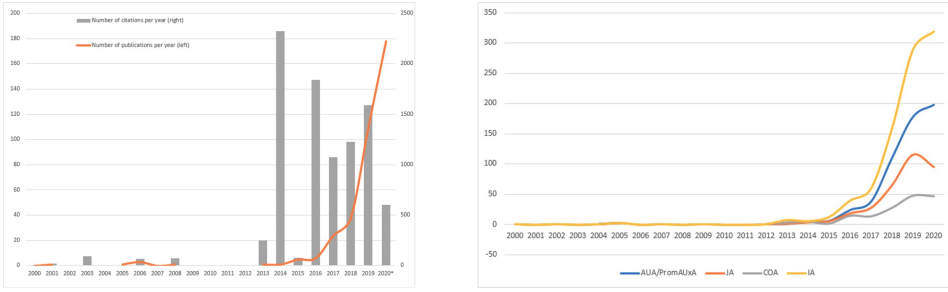
en el campo de la TL aplicada al reconocimiento de gestos serían este framework [Yang et al., 2019] que es capaz de identificar una serie de gestos a través de una arquitectura siamesa basada en Redes Convolucionales Recurrentes. En esta arquitectura, las características espaciales se aprenden primero de la CNN y luego las características temporales se aprenden con la Red Neuronal Recurrente. Los resultados de los experimentos avalan la eficacia de este método, obteniendo un 89,5 % de precisión. Otro trabajo relevante al respecto es WiADG [Zou et al., 2018a] que es un sistema de reconocimiento de gestos que funciona independientemente de cualquier dispositivo ya que utiliza la señal WiFi para realizar este reconocimiento. Este sistema tiene la particularidad de que puede identifi-

Orden	**Título**	**Autor/es**	**Revista**	**C**	**Año**	**C/A**
#1	3D Convolutional neural networks for human action recognition	Ji S, Xu W, Yang M, Yu K	IEEE Transactions on Pattern Analysis and Machine Intelligence	2,317	2013	331.0
#2	Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition	He K, Zhang X, Ren S, Sun J	IEEE Transactions on Pattern Analysis and Machine Intelligence	1,739	2015	347.8
#3	Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition	Ordonez FJ, Roggen D	Sensors	461	2016	115.3
#4	A novel hybrid CNN-SVM classifier for recognizing handwritten digits	Niu XX, Suen CY	Pattern Recognition	253	2012	31.6
#5	Deep learning for electromyographic hand gesture signal classification using transfer learning	Cote-Allard U, Fall CL, Drouin A, Campeau-Lecours A, Gosselin C, Glette K, Laviolette F, Gosselin B	IEEE Transactions on Neural Systems and Rehabilitation Engineering	241	2019	30.1
#6	Review of Convolutional Neural Network	Zhou FY, Jin LP, Dong J	Chinese Journal of Computers	174	2017	87.0
#7	Deep learning and process understanding for data-driven Earth system science	Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, Prabhat	Nature	159	2019	159.0
#8	Deep Learning for Computer Vision: A Brief Review	Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E	Computational Intelligence and Neuroscience	149	2018	74.5
#9	Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition	Wu D, Pigou L, Kindermans PJ, Le NDH, Shao L, Dambre J, Odobez JM	IEEE Transactions on Pattern Analysis and Machine Intelligence	142	2016	35.5
#10	Deep visual domain adaptation: A survey	Wang M, Deng W	Neurocomputing	119	2018	59.5

Tabla 3.9: Los diez artículos más citados que han sido publicados en la última década teniendo en cuenta el año de publicación(C: Citas; C/A: Citas por artículo).

car gestos en un entorno dinámico con el uso de una adaptación de dominio adversario supervisada, a través de los dispositivos IoT que se encuentran en él. Estos dispositivos son la fuente de información del estado del canal que es necesaria para analizar y detectar los diferentes gestos que realiza el usuario. Los resultados muestran que este sistema alcanza una precisión del 98 % en el proceso de reconocimiento de gestos.

En las Tablas 3.10 y 3.11 se pueden observar las veinte *keywords* más representativas en diferentes intervalos de tiempo. El factor común en estos diferentes periodos está caracterizado por *face recognition*, la cual se encuentra entre las posiciones más altas en cada uno de los intervalos definidos. Esta tendencia es debido a que el reconocimiento facial es un problema tradicional en Visión Artificial que sigue en auge y además que la identificación de la cara y su respectiva verificación se han hecho muy populares por temas de control de acceso y seguridad, así como aspectos de biometría [Guo and Zhang, 2019]. Sin embargo, es necesario realizar una mención especial a las keywords *Convolutional Neural Network* y DL que también están en las primeras posiciones junto con *Face recognition* puesto que DL ha realizado importantes contribuciones a Visión Artificial en los últimos años. Por último, en relación a la keyword *gesture recognition* que es una de las más representativas para este estudio, cabe destacar que ha comenzado a despertar más interés a partir de 2016 donde se encuentra entre las 10 keywords más citadas en los papers de esa época. Esta afirmación es confirmada en la Figura 3.6a donde se puede observar que el porcentaje de frecuencia de dicha keyword es mayor entre el período de 2016-2020.



(a) Las palabras clave más citadas de 2000 a 2020.

(b) Frecuencia normalizada de aparición de cada palabra clave

Figura 3.6: Datos referentes a las palabras clave de búsqueda.

Por último, los modelos de las Figuras 3.7 y 3.8 representan la relación entre las técnicas de Inteligencia Artificial y los distintos tipos de reconocimiento de gestos. Este modelo se ha creado revisando cada artículo y extrayendo el tipo de reconocimiento de gestos y las técnicas de Inteligencia Artificial que se utilizaron para identificar los gestos. Como resultado se ha obtenido el esquema, donde se puede observar cómo los nodos del centro representan los tipos de gestos mientras que los nodos periféricos muestran las diferentes técnicas de Inteligencia Artificial que se han identificado. La Figura 3.7 muestra las técnicas de DL, mientras que la Figura 3.8 representa las técnicas de ML.

Orden	2000-2020			2000-2015			2016-2018			2019-2020		
	Keywords	**A**	%	**Keywords**	**A**	%	**Keywords**	**A**	%	**Keywords**	**A**	%
1	Convolutional neural network	330	4,39 %	Face recognition (+ Facial recognition)	20	4,61 %	Convolutional neural network	130	4,56 %	Face recognition (+ Facial recognition)	270	6,37 %
2	Deep learning	282	3,75 %	Artificial neural network	10	2,30 %	Face recognition (+ Facial recognition)	103	3,62 %	Convolutional neural network	191	4,51 %
3	Face recognition (+ Facial recognition)	279	3,71 %	Convolutional neural network	9	2,07 %	Deep learning	96	3,37 %	Deep learning	179	4,22 %
4	Neural network	138	1,83 %	Deep learning	7	1,61 %	Neural network	41	1,44 %	Neural network	90	2,12 %
5	Machine learning	117	1,56 %	Neural network	7	1,61 %	Machine learning	40	1,40 %	Machine learning	73	1,72 %
6	Support vector machine	91	1,21 %	Convolution	6	1,38 %	Artificial neural network	32	1,12 %	Support vector machine	57	1,34 %
7	Feature extraction	82	1,09 %	Feature extraction	5	1,15 %	Deep neural network	31	1,09 %	Feature extraction	52	1,23 %
8	Artificial neural network	76	1,01 %	Algorithm	5	1,15 %	Support vector machine	30	1,05 %	Facial expression recognition	52	1,23 %
9	Convolution	76	1,01 %	Facial expression	5	1,15 %	Gesture recognition	29	1,02 %	Convolution	47	1,11 %
10	Facial expression recognition	71	0,94 %	Machine learning	4	0,92 %	Face	27	0,95 %	Classification	39	0,92 %

Tabla 3.10: Las veinte palabras clave más usadas (Parte I).

Orden	2000-2020			2000-2015			2016-2018			2019-2020		
	Keywords	**A**	%	**Keywords**	**A**	%	**Keywords**	**A**	%	**Keywords**	**A**	%
11	Face	64	0,85 %	Support vector machine	4	0,92 %	Feature extraction	25	0,88 %	Face	37	0,87 %
12	Gesture recognition	64	0,85 %	Facial expression recognition	4	0,92 %	Convolution	23	0,81 %	Artificial neural network	34	0,80 %
13	Deep neural network	56	0,74 %	Recognition	3	0,69 %	Algorithm	23	0,81 %	Gesture recognition	34	0,80 %
14	Classification	55	0,73 %	Recurrent neural network	2	0,46 %	Facial expression recognition	15	0,53 %	Human	33	0,78 %
15	Algorithm	51	0,68 %	Gesture recognition	1	0,23 %	Classification	15	0,53 %	Facial expression	31	0,73 %
16	Human	48	0,64 %	Classification	1	0,23 %	Human	15	0,53 %	Recognition	29	0,68 %
17	Recognition	46	0,61 %	Transfer learning	0	0,00 %	Recognition	14	0,49 %	Transfer learning	28	0,66 %
18	Facial expression	43	0,61 %	Face	0	0,00 %	Recurrent neural network	13	0,46 %	Deep neural network	25	0,59 %
19	Transfer learning	39	0,57 %	Deep neural network	0	0,00 %	Facial expression	7	0,25 %	Algorithm	23	0,54 %
20	Recurrent neural network	35	0,52 %	Human	0	0,00 %	Transfer learning	1	0,04 %	Recurrent neural network	20	0,47 %

Tabla 3.11: Las veinte palabras clave más usadas (Parte II).

Esta relación entre estos dos conceptos ha sido determinada por las aristas que forman la conexión entre los nodos central y periférico, así como el tamaño de los nodos y el grosor de dichas aristas. Por un lado, es necesario comentar que existen tres tamaños diferentes: pequeño, mediano y grande. Los nodos de gran tamaño son aquellos con un número de ocurrencias superior a 80, los de tamaño mediano están en el rango (30-80] y los pequeños están en el rango (0-30). Por otro lado, hay tres colores diferentes: verde, azul y rojo. En relación a los nodos, estos se pintaron según el tamaño, los nodos de gran tamaño tienen el color verde, los de tamaño mediano azul y los de tamaño pequeño rojo. Respecto a las aristas, su groso es el mismo para cada una de las aristas y las ocurrencias se reflejan con el color. Las aristas verdes significa que tienen un número de ocurrencias superior a 30, las aristas azules se encuentran en el rango (10-30] y las aristas de color rojo están en el rango (0-10]. Sin embargo, estas aristas no solo nos dicen el número de ocurrencias sino que también nos muestran qué técnicas de Inteligencia Artificial se han utilizado para cada tipo de reconocimiento ya que si no hay ocurrencias entonces las aristas no se pintan en la figura.

La creación de este modelo dio lugar al estudio de los casos más relevantes en relación a propuestas y fiabilidad. En cuanto a las propuestas, el 24 % corresponde a modelos híbridos, los cuales son ampliamente utilizados porque el objetivo es aprovechar las ventajas de los métodos involucrados y de esta manera lograr un mayor rendimiento y precisión, además de que son más flexibles y robustos que los modelos que no son híbridos [Aziz et al., 2017]. La propuesta híbrida más desarrollada ha sido la combinación de Máquinas de Soporte Vectorial y CNN, que representa el 48 % del total de estos modelos híbridos. Un ejemplo de estas metodologías es HandSense [Zhang et al., 2018], que consiste en un sistema que tiene el objetivo de reconocer gestos dinámicos con las manos en varias CNN 3D que se dedican a extraer características espacio-temporales para que la técnica de SVM pueda reconocer los diferentes gestos a través de dichas características. Sin embargo, CNN es la técnica popular más utilizada, con un porcentaje del 56 % y se ha aplicado principalmente para reconocer expresiones faciales, como en este estudio [Lopes et al., 2017], donde las imágenes se preprocesan antes de ser insertadas como datos de entrada en la CNN para aprender y detectar expresiones faciales. Aunque en el proceso hubo un problema que era la ausencia de datos suficientes para el aprendizaje con CNN pero los autores lo resolvieron mediante la técnica de *Data Augmentation*. Las Redes Neuronales Recurrentes es otra de las técnicas más utilizadas, aunque su porcentaje es menor que CNN, con un 11 %. En el siguiente trabajo [Ofodile et al., 2019], su finalidad consiste en el reconocimiento de determinadas acciones mediante la aplicación de Redes Neuronales Recurrentes. El proceso de reconocimiento se basa en una Red Neuronal Recurrente que recibe una serie de medidas de una cámara de tiempo de vuelo de las que obtiene una predicción analizando la secuencia temporal completa. Este sistema es capaz de reconocer con gran precisión las siguientes acciones: caminar hacia adelante, caminar hacia atrás, sentarse, levantarse y saludar con la mano.

A pesar de que estos son los enfoques más populares, también debemos destacar nuevas propuestas como el 3DCNN piramidal [Zhu et al., 2016] o la red de cápsulas [Lee et al., 2020a]. La arquitectura piramidal 3DCNN se compone de una entrada piramidal, fusión piramidal, fusión multimodal y dos 3DCNN. La entrada piramidal tiene la función de segmentar cada uno de los videos que recibe como entrada porque tienen diferentes

longitudes y utiliza un muestreo uniforme con *jitter* temporal para obtener la entrada piramidal. A continuación, esta entrada alimenta la fase de fusión piramidal para unir las características de la entrada piramidal y, finalmente, debido a que las redes habían entrenado los datos con RGB y profundidad de forma independiente, la fusión multimodal debe fusionar ambas modalidades para permitir el reconocimiento de gestos. La red de cápsulas está formada por 5 capas convolucionales, las capas de cápsulas primarias que están formadas por 3872 cápsulas de 8 dimensiones y la capa llamada *GestureCaps* que contiene 5 cápsulas de 16 dimensiones. Cada cápsula es un conjunto de neuronas y tiene una probabilidad de activación y una *pose matrix*. El funcionamiento de esta arquitectura consiste en que las 5 capas convolucionales se utilizan para reducir la dimensión y luego las capas de la cápsula tienen una función de aplastamiento para obtener la salida de la cápsula que es multidimensional. De esta manera, la salida alimenta la capa *GestureCaps* cuya salida afectará la decisión de la capa de salida aplicada por softmax para obtener la etiqueta correspondiente en la clasificación de los gestos.

Aunque es importante aprender acerca de las propuestas más novedosas, no son útiles si no son fiables. Esa es la razón por la que se describirán algunos de los casos más fiables. En este trabajo [Sun et al., 2018] las expresiones faciales y los gestos corporales se reconocen mediante un modelo que combina CNN, memoria a corto plazo y PCA para extraer características espacio-temporales. Esta propuesta ha alcanzado una precisión del 99,57 % en un conjunto de datos de la cara y cuerpo (FABO) [Gunes and Piccardi, 2006]. En este estudio [Sun and Lv, 2019] se ha creado un modelo híbrido que combina las características extraídas de la transformación de características invariantes de escala (SIFT) con las características de CNN para reconocer la expresión facial. Esta combinación ha logrado una precisión del 94,82 % en la base de datos Cohn-Kanade (CK +) [Lucey et al., 2010]. En [Czuszyński et al., 2018] se han registrado un total de 27 gestos con un sensor óptico lineal para detectar los gestos de las manos. Se han registrado tres tipos de representación de gestos: sin procesar, características simples y características de alto nivel; que han sido los datos de entrada de la Red Neuronal Recurrente para clasificar los gestos con las manos. De estos tres tipos de representación, los datos sin procesar han sido los que han obtenido mayor tasa de aciertos y la precisión media ha sido del 96,89 %. Este trabajo [Swarnkar and Ambhaikar, 2019] está destinado a aplicar el reconocimiento de gestos estáticos al lenguaje de signos. Se ha desarrollado una Red Neuronal Convolucional mejorada (ICNN) y se ha aplicado *dropout* y la regularización L2 para evitar el sobreajuste. La eficacia de este método para el reconocimiento de la lengua de signos se refleja en la precisión de clasificación obtenida del 99,96 %.

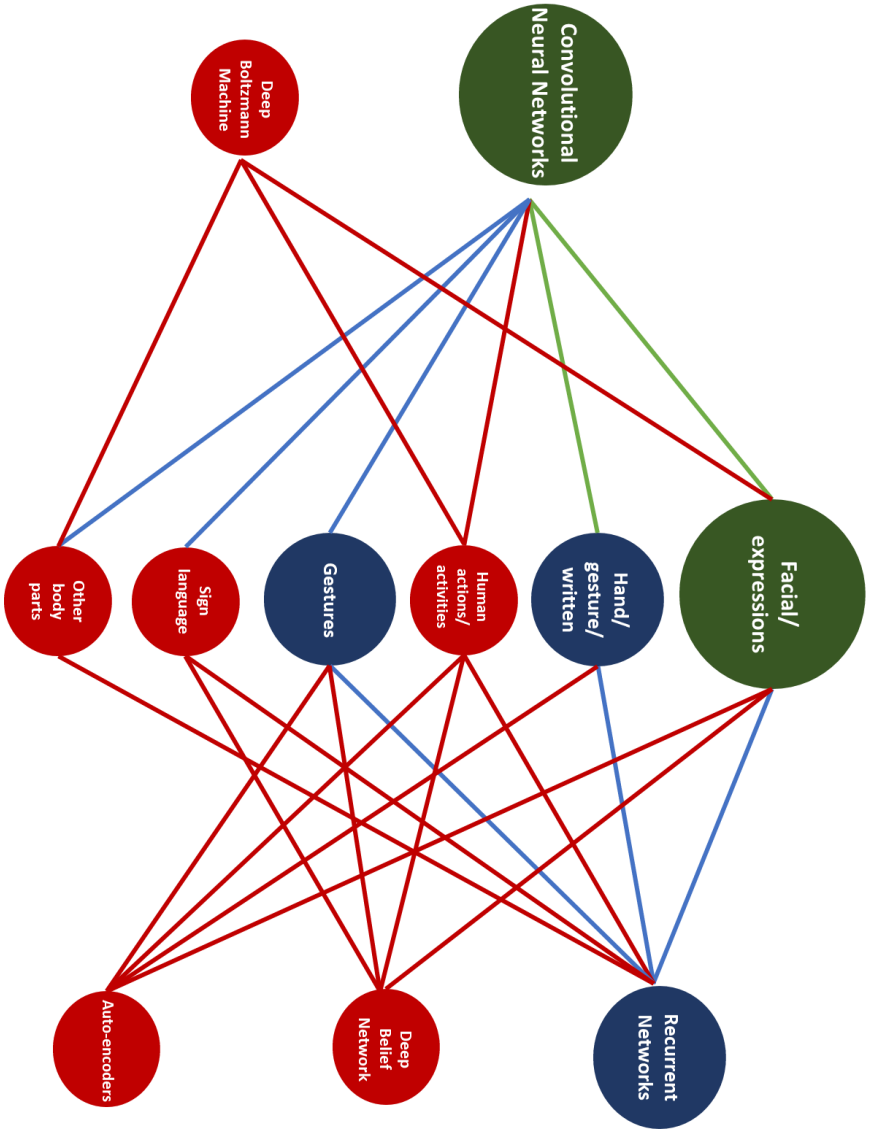


Figura 3.7: Relación entre las técnicas de *Deep Learning* y el reconocimiento de gestos.

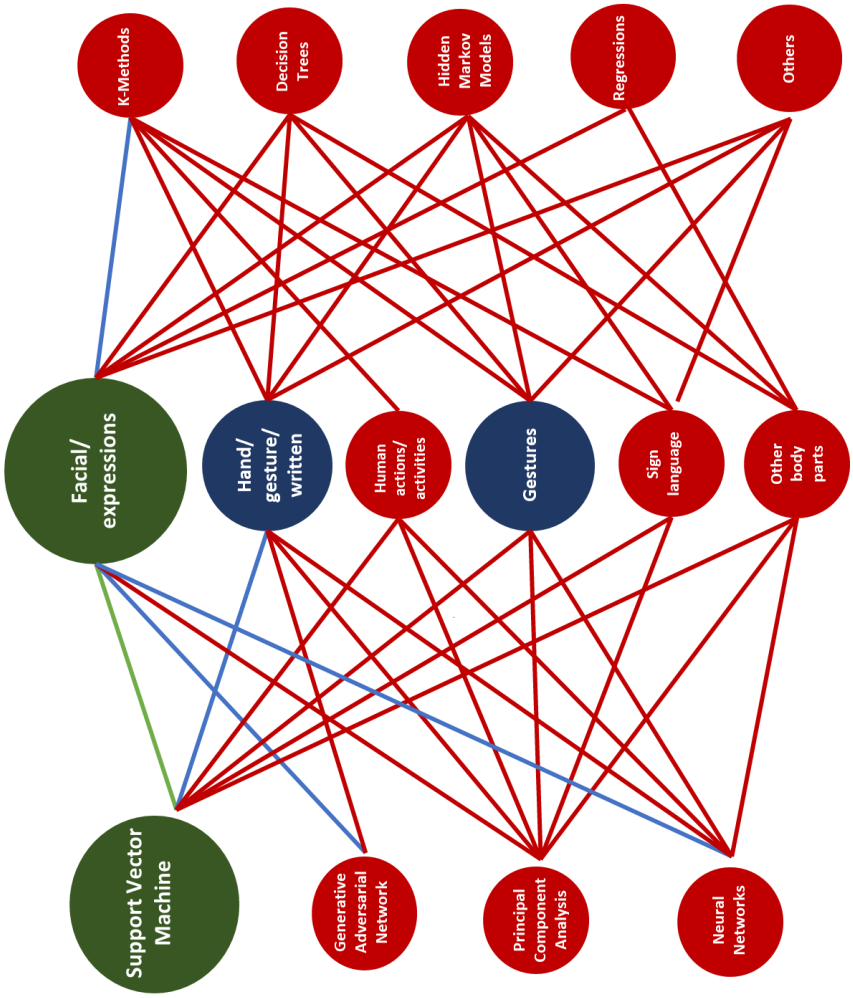


Figura 3.8: Relación entre las técnicas de *Machine Learning* y el reconocimiento de gestos.

3.3. Conclusiones

En este estudio se han recogido los indicadores más relevantes respecto a investigación en el tema de las técnicas de Inteligencia Artificial aplicadas a gesture recognition que abarcan los últimos veinte años, del año 2000 a 2020.

De este estudio se ha averiguado que ha habido un incremento sustancial del número de artículos elaborados y de las citas referentes al tema de este análisis bibliométrico a partir del año 2016, mientras que anteriormente pasaba desaprecibido. Un dato a destacar es que las instituciones académicas que han mostrado un mayor interés por el tema y han contribuido mayormente a la investigación en el reconcimiento de gestos e Inteligencia Artificial proceden principalmente de China. Probablemente este hecho viene incitado porque los métodos de DL se están aplicando a numerosas áreas de conocimiento entre las que se encuentra Visión Artificial. Esta combinación está produciendo grandes avances en el campo de Visión Artificial, el cual está estrechamente relacionado con el reconocimiento de gestos debido a que en la clasificación de este tipo de reconocimiento existe el reconocimiento de gestos basado en visión. Por lo tanto, no es de extrañar que haya existido un reciente interés por este tema en los últimos años para obtener resultados prometedores en este campo con la utilización de técnicas de Inteligencia Artificial.

En este análisis bibliométrico se ha trabajado con aproximadamente 571 artículos extraídos de las bases de datos científicas WoS and Scopus. A partir de esta información se han descrito los aspectos más relevantes relacionados con la producción científica en el tema de este estudio, donde en primer lugar se ha explicado el proceso que se ha seguido para elaborar este estudio bibliométrico en la sección 3.1 y a continuación se han compartido los resultados obtenidos de dicho proceso en la sección 3.2. En esta sección se han presentado los indicadores que son relevantes para este estudio, entre ellos se ha mostrado la evolución de los artículos publicados durante los años que comprende este estudio y se ha descrito las revistas con mayor cantidad de artículos, los autores que han sido más citados, los países donde se han producido el mayor número de artículos y con mayor h-index, una descripción breve de cada uno de los diez artículos con más citas, así como las keywords que son más significativas cuando se realiza una búsqueda sobre los términos concernientes a este trabajo.

A pesar de las investigaciones que se están realizando, el interés por incluir algoritmos de Inteligencia Artificial en el reconocimiento de gestos es reciente y todavía existen bastantes limitaciones que superar. Sin embargo, los avances son prometedores como es posible apreciar en la descripción de los diez artículos más citados que se han incluido en este análisis.

Capítulo 4

Sistema interactivo basado en Kinect

Capítulo 4

SISTEMA INTERACTIVO BASADO EN KINECT

Contenidos

Al comienzo de este trabajo se decidió utilizar Microsoft Kinect para el proceso de interacción natural. Esta decisión se tomó debido a algunos de los objetivos de este estudio, ya que el sistema tenía que ser de bajo coste, portátil y basarse en interacción natural, entre otros. Kinect era el mejor dispositivo RGB-D del mercado porque era económico y se podía transportar a cualquier parte sin mucho esfuerzo. Además, tenía algunas funcionalidades muy útiles como el seguimiento del esqueleto donde se reconocían ciertas articulaciones de los usuarios y se podía trabajar con su posición para detectar gestos tanto estáticos como dinámicos.

El primer enfoque fue la creación de un sistema que detecta el movimiento y reconoce los gestos estáticos. El objetivo principal de este sistema era ayudar a los estudiantes con necesidades especiales en su aprendizaje cognitivo y mejorar su condición física por medio de una interacción sencilla y cómoda. Además, las actividades están diseñadas como juegos para motivar a los estudiantes y mantener su atención.

Las principales aportaciones de esta parte del trabajo son desarrollar un sistema que pueda integrarse en un aula de educación especial donde se encuentran alumnos con diferentes discapacidades (visual, auditiva, física y autismo) y un algoritmo que detecta solo al usuario que opera la aplicación. y filtra las articulaciones del usuario que interactúa con el dispositivo para un mejor reconocimiento y usabilidad.

La arquitectura del sistema se basa en los siguientes módulos: módulo de entrada, módulo HCI, sistema inteligente, el ciclo de vida del juego y el módulo de salida (ver Figura 4.1).

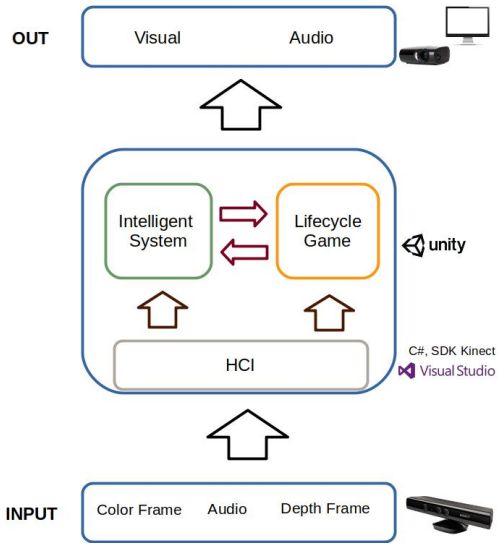


Figura 4.1: Arquitectura del sistema.

El módulo de entrada contiene datos del dispositivo Kinect que incluyen los flujos de vídeo, audio y profundidad. Este sistema maneja los datos recibidos por el módulo de entrada mientras que la interacción del usuario es responsable del uso del flujo de datos. La unidad HCI envía la información de la interacción entre el usuario y el dispositivo Kinect, por ejemplo, la posición de cada articulación identificada en el seguimiento del esqueleto en cada momento. Esta información se envía luego al módulo de juego del ciclo de vida y al módulo del sistema inteligente que utiliza esta información para hacer la interacción más adecuada para los usuarios.

4.1. Sistema Inteligente

Este módulo va a determinar las acciones que realiza el sistema mediante el uso de un sistema basado en reglas. Este componente de la arquitectura funciona de la siguiente forma (ver Figura 4.2) y este pseudocódigo (ver Algoritmo 1) muestra el flujo principal de la propuesta. En primer lugar, el sistema comprueba si Microsoft Kinect está en uso y detecta el esqueleto del usuario en su rango de rendimiento funcional. Sin embargo, el dispositivo no puede detectar el esqueleto más cercano en movimiento. El método `quitarEsqueletos (línea 9)` fue desarrollado para resolver este problema. Esta función necesita una matriz de esqueletos como parámetro para verificar la distancia entre las articulaciones de cada esqueleto y la posición relativa al sensor Kinect. Después de elegir el mejor esqueleto, se eliminan el resto. El siguiente paso es detectar si hay articulaciones superpuestas para que se filtren con el fin de hacer un seguimiento al esqueleto del usuario de manera eficiente. El método `filtrarArticulacionesSuperpuestas (línea 10)` comprueba la distancia entre las diferentes articulaciones de cada esqueleto para asignar la articulación al esqueleto correcto. Esto se aplica a los casos en que el tutor se sienta al lado del alumno y afecta al sensor de detección del dispositivo. El método mencionado propone solucionar este problema.

Siguiendo este paso, el usuario seleccionará la actividad deseada. Dependiendo de la actividad seleccionada, el sistema inteligente realizará la acción. En las actividades de números y formas, el sistema inteligente puede adaptarse para aumentar el nivel de dificultad de la actividad en función de la articulación elegida. Los elementos se colocarán dentro de un área cercana en el nivel más fácil. Como articulación elegida, los objetos tendrán que reajustar su posición y se colocarán en una zona donde el usuario no tenga que hacer ningún movimiento incómodo o crear una situación frustrante para él o ella. Además, el método tiene un proceso de retroalimentación. Si detecta que la tasa de fallos es muy alta o el usuario comete muchos errores de forma continua, cambia algunas características como la velocidad o la posición de los elementos para que el usuario pueda realizar la acción con precisión. La actividad de coordinación tiene la ventaja de que los tutores pueden seleccionar la posición de los nodos, y el orden en el que se tienen que unir los mismos para personalizar el movimiento que el usuario tiene que realizar. En esta actividad, este sistema tiene la función de supervisar que el usuario realiza la secuencia en el orden correcto, penalizando la acción en caso contrario. Finalmente, en la actividad de grafomotricidad, el sistema activará el reconocimiento de gestos para la realización del ejercicio y analizará si los gestos realizados por el usuario son correctos

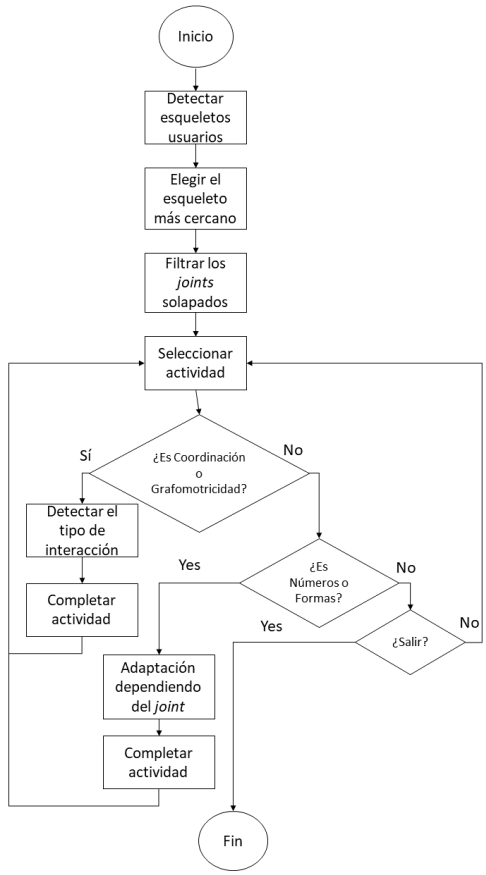


Figura 4.2: Diagrama de flujo del sistema.

para la completitud de la actividad de acuerdo con la modalidad elegida previamente.

El sistema inteligente permite que el software realice tareas como reconocimiento de gestos o seguimiento de los *joints* para la detección del movimiento. Esta unidad se divide en dos procesos principales: el reconocimiento y la interpretación. En el apartado de reconocimiento, se identifican diferentes gestos implementados con la ayuda de un conjunto de reglas sobre gestos. Las reglas de gestos predefinidos aseguran que los gestos verifiquen las condiciones en cada fase del proceso y evalúen la posición de las articulaciones involucradas en las actividades. Dependiendo de las reglas que se hayan recopilado, se confirma un gesto que se ha reconocido y se notifica al módulo de interpretación al respecto. El apartado de interpretación no realiza ninguna acción hasta que la unidad de reconocimiento de gestos haya obtenido el resultado correcto. El resultado luego se comunica al ciclo de vida del juego, que proporciona una respuesta de audio y visual a cambio. Este módulo es capaz de evaluar y ejecutar la acción deseada en función

Algorithm 1 Procedimiento de interacción.

```
1: procedure FlujoPrincipal
2:     inicializar el dispositivo Microsoft Kinect
3:     cargar reglas
4:     sistemaReglas ← inicializlar el sistema basado en reglas
5:     while usuario no detectado do
6:         usuarios ← comprobar si hay usarios para usar Kinect
7:     end while
8:     if usuarios > 1 then
9:         quitarEsqueletos()
10:        articulaciones ← filtrarArticulacinesSuperpuestas()
11:    end if
12:    actividad ← elegirActividad()
13:    configuracion ← configurarActividad()
14:    sistemaReglas.cargarConfiguracion()
15:    sistemaReglas.aplicarConfiguracion()
16:    tipoInteraccion ← sistemaReglas.obtenerInteraccion()
17:    if tipoInteraccion = gestos then
18:        activar el Proceso de Reconocimiento de Gestos
19:    else
20:        activar el Seguimiento de los Joints
21:    end if
22:    actividadFinalizada ← false
23:    while actividadFinalizada = false do
24:        actividadFinalizada ← comprobar el estado de la actividad
25:    end while
26:    mostrarFeedback()
27: end procedure
```

del gesto identificado.

Las ventajas de introducir nuevas reglas son que permite a los usuarios tener un sistema de control más preciso y la posibilidad de personalizarlo a sus necesidades. Estas reglas integran nuevas funcionalidades al sistema, incluir nuevos perfiles, desarrollar nuevas habilidades, añadir nuevos gestos, adaptar las actividades, introducir nuevos aspectos en el apartado de configuración y mejorar el reconocimiento de gestos y voz. Estas reglas se almacenan en un archivo XML que se puede editar para modificar el comportamiento del sistema.

El uso de estas reglas ha sido el factor determinante para toda la adaptación del sistema relacionada con la configuración de las actividades y la interacción. En la configuración podemos seleccionar la velocidad de movimiento y tamaño del elemento de interacción, la activación de la cámara RGB desde el dispositivo Kinect durante la realización de la actividad, la articulación elegida para trabajar y los niveles de dificultad.

4.2. Módulos del sistema

El sistema está compuesto por dos módulos; un módulo de coordinación para el desarrollo de habilidades físicas y un módulo de actividades que tiene como objetivo mejorar las habilidades cognitivas. Para desarrollar estos módulos se utilizó Microsoft Visual Studio, junto con el lenguaje de programación C#. Se utilizaron librerías de terceros, Kinect Library SDK y Unity 3D como framework para desarrollar las diferentes actividades como juegos, utilizando diferentes componentes característicos como Game Loop [Qu et al., 2013, Joselli et al., 2012]. El módulo de coordinación tiene una actividad donde el usuario tiene que unir un conjunto de nodos, con la parte del cuerpo previamente seleccionada en la configuración (ver Figura 4.3a). Al inicio de esta actividad se muestra en pantalla un conjunto de nodos con un color diferente en función de su estado, que son los siguientes:

- Naranja: es el color por defecto que significa que no es el nodo objetivo.
- Azul: es el nodo objetivo.
- Verde: el nodo ha sido tocado por el usuario satisfactoriamente.

La actividad comienza con todos los nodos en color naranja, excepto uno que es de color azul (el nodo objetivo). El usuario debe tocar el nodo objetivo. Una vez que el nodo objetivo se toca con la parte correcta del cuerpo, su color cambiará a verde (ver Figura 4.3b). Esto le mostrará al usuario que la acción se completó con éxito. Este proceso continuará hasta que todos los nodos hayan cambiado sus colores a verde 4.3d. Luego, se cargará la pantalla de opciones (ver Figura 4.3c).

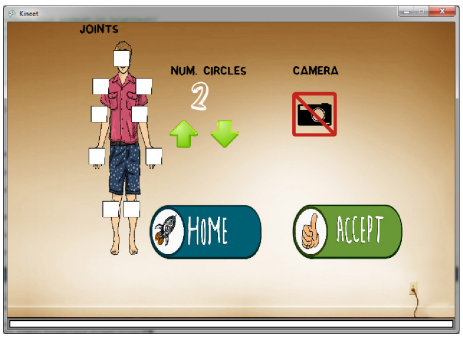
Hay tres actividades en el módulo de actividades:

#1 **Números**: en esta actividad, los números caen con diferentes velocidades según la configuración predeterminada. El estudiante puede tocar fácilmente los números con la mano derecha o izquierda. El objetivo de esta actividad es tocar los números en orden ascendente 4.4a. La secuencia cambiará según el nivel de dificultad.

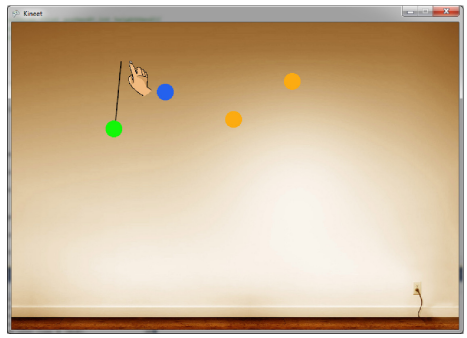
#2 **Formas**: es similar al ejercicio anterior. Las diferencias de este ejercicio respecto al anterior son que caerán formas 4.4b en lugar de números y que el estudiante tiene que tocar la forma correcta en respuesta a las instrucciones dadas en la pantalla sin seguir ningún orden.

#3 **Grafomotricidad**: el ejercicio comienza con dos imágenes; una de ellas se asigna como inicial mientras que la otra como final. En esta actividad se activará el reconocimiento de gestos o la detección de movimiento según la modalidad elegida. Si el modo es Horizontal (ver Figura 4.5a) o Vertical (ver Figura 4.5b), entonces se realizará el reconocimiento de gestos, mientras que si es Estilo libre (ver Figura 4.5c), la detección de movimiento será iniciada. En el modo Horizontal, el usuario tendrá que realizar un movimiento con el brazo de izquierda-derecha (similar al gesto *swipe left*) o de derecha-izquierda (similar al gesto *swipe right*). Sin embargo, el modo Vertical requiere que el usuario haga un gesto de arriba-abajo (similar al

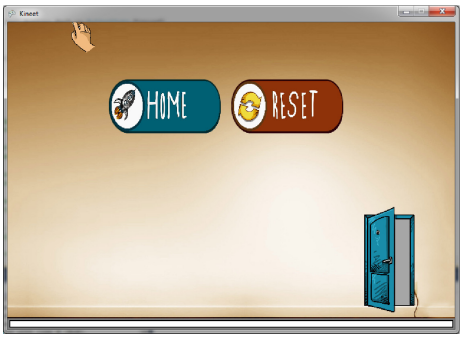
gesto *swipe down*) o de abajo-arriba (similar a *swipe up*) para alcanzar el objetivo. En la versión de Estilo libre, el individuo demostrará su precisión con la detección del movimiento puesto que tendrá que realizar un trazo a mano alzada, evitando una serie de obstáculos que aparecerán en la pantalla entre el punto inicial y final. La retroalimentación de esta actividad es una animación que está relacionada con el tema de la actividad (ver Figura 4.5d).



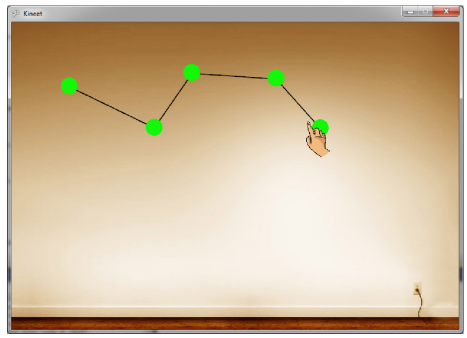
(a) Pantalla de configuración.



(b) Acción correcta durante la realización de la actividad.

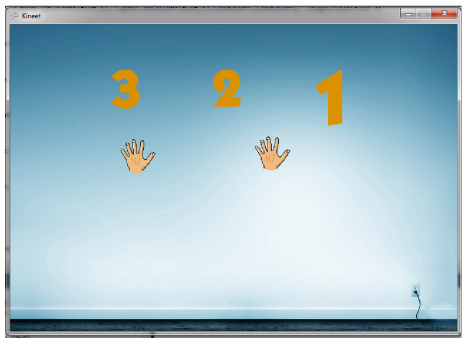


(c) Pantalla de opciones.

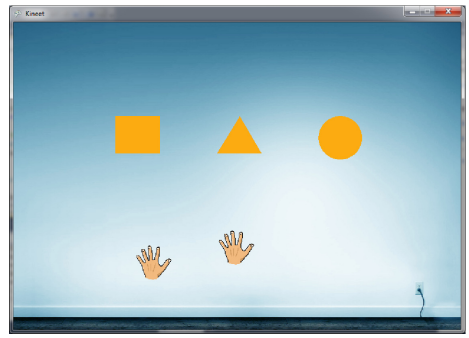


(d) Finalización de la actividad.

Figura 4.3: Pantallas relacionadas con el ejercicio de coordinación.

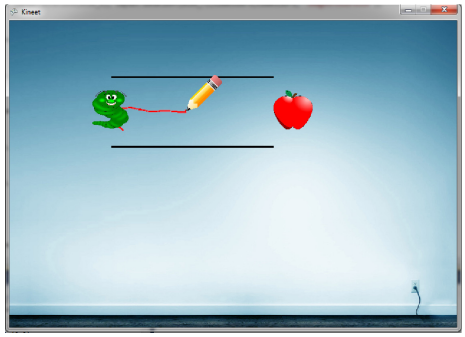


(a) Actividad de los números.

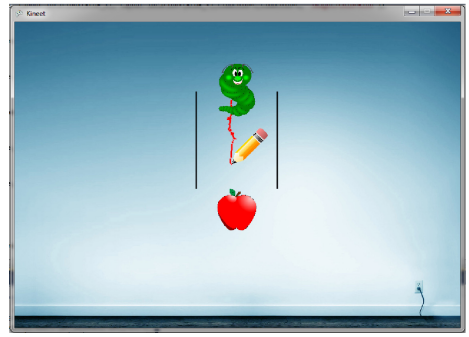


(b) Actividad de las formas.

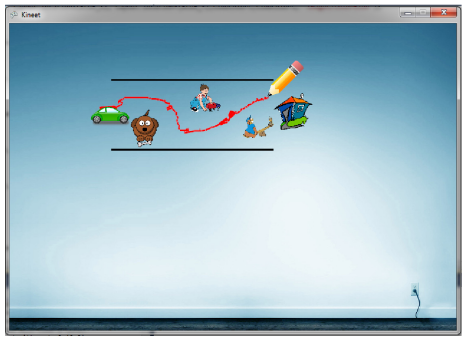
Figura 4.4: Pantallas de las actividades de los números y las formas.



(a) Actividad de grafomotrocidad modo Horizontal.



(b) Actividad de grafomotrocidad modo Vertical.



(c) Actividad de grafomotricidad modo Estilo libre.

(d) *Feedback* de la actividad de grafomotricidad.

Figura 4.5: Actividades de grafomotricidad.

4.3. Resultados

Los experimentos de este estudio se han realizado en el centro de educación especial Princesa Sofía que es una institución de ámbito público de la Junta de Andalucía que se encuentra situado en la provincia de Almería. Este colegio está integrado por alumnos que tienen características muy variadas ya que presentan discapacidad física, sensorial, intelectual o una combinación de las anteriores, que es el caso predominante. El personal de este centro se compone de veintidós monitores, tres fisioterapeutas, siete logopedas, un orientador, una enfermera y un médico. En este estudio exploratorio, los profesores del centro de educación especial Princesa Sofía han estado colaborando durante la duración de esta parte del proyecto donde se desarrolló una aplicación útil para sus alumnos de acuerdo con los problemas encontrados en sus prácticas docentes en el aula. La herramienta es innovadora porque utiliza la interacción natural con la ayuda del sensor Kinect. El estudio implicó implantar la nueva aplicación en el aula durante varias sesiones para determinar su efectividad y utilidad para los estudiantes.

Se llevaron a cabo dos tipos de evaluaciones; una de ellas fue realizada por los expertos, y la otra con los usuarios finales. La evaluación con expertos consistió en rellenar una encuesta con sus valoraciones en tres aspectos: usabilidad, modalidad educativa y comportamiento de los estudiantes. Las encuestas se realizaron con la escala Likert[Joshi et al., 2015] con valores que van del 1 al 5, donde 5 significa que el encuestado está totalmente de acuerdo con la afirmación planteada, y por el contrario, 1 quiere decir que el encuestado no está de acuerdo con dicha afirmación. Estos expertos procedían de varios campos, donde además de los tres tutores incluidos en este proyecto, en este proceso de evaluación también colaboraron un experto en técnicas educativas y un experto en informática.

Los experimentos con usuarios finales se probaron con cuatro estudiantes. Cada uno de ellos tiene características diferentes: discapacidad visual, discapacidad auditiva, discapacidad física y autismo. Cada participante había probado el sistema durante treinta días con sus respectivos tutores. En cada sesión, los estudiantes realizaron todas las actividades (ver Figura 4.6) mientras la aplicación recopilaba el tiempo y los errores automáticamente.

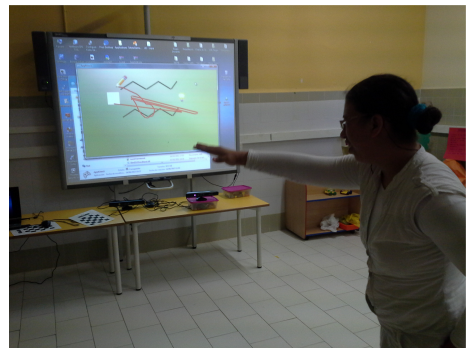
4.3.1. Evaluación de usabilidad

El objetivo de esta evaluación es conocer la facilidad de uso de este sistema y las dificultades que pueden encontrar los usuarios al utilizarlo.

Los resultados obtenidos en esta evaluación se muestran en la Tabla 4.1 y el cuestionario utilizado para esta evaluación en el Anexo B Tabla B.1. De los resultados se puede apreciar que la característica con mayor puntuación es el acceso a las actividades. Esto se debe a que cada módulo está asociado con diferentes iconos que repiten el propósito de cada módulo y también diferentes colores: el color naranja se utiliza para la sección de coordinación y el color azul para la sección de actividades.

La función que ha recibido la puntuación más baja es la configuración del software. La razón es que los pasos para instalar este programa en otro ordenador podrían ser complicados ya que el usuario no solo tiene que configurar la aplicación, sino también

(a) Tutora asistiendo al estudiante en la realización del ejercicio.



(b) Estudiante realizando la actividad de manera autónoma.

Figura 4.6: Experimentos con usuarios finales.

Características	EDES1	EDES2	F	ETE	EI	Media
Facilidad de uso	3	4	3	5	3	3.6
Actividades fácilmente accesibles	5	4	4	5	2	4
Configuración del software	2	2	1	4	2	2.2
Personalización de las Actividades	4	4	3	4	3	3.6
Período de Adaptación Bajo	3	4	3	5	3	3.6

Tabla 4.1: Resultados de la evaluación de usabilidad. EDES1: Experto en Educación Especial / EDES2: Experto en Educación Especial / F:Fisioterapeuta / ETE: Experto en Técnicas Educativas / EI: Experto en Informática

instalar los controladores del dispositivo Kinect y ciertas utilidades como la plataforma de reconocimiento de voz de Microsoft. No obstante, se ha creado una guía de configuración para realizar correctamente cada uno de los pasos durante el proceso de instalación. En relación al resto de características, otra afirmación destacada sería la usabilidad del programa que posee una calificación baja. Esto podría deberse a que el usuario puede encontrar dificultades al utilizar el software si nunca antes ha utilizado el dispositivo, ya que requiere un período de adaptación.

4.3.2. Evaluación Educativa

Este experimento consistió en evaluar las condiciones necesarias para decidir si este sistema era adecuado para el proceso de aprendizaje de los estudiantes. Los resultados de esta evaluación se resumen en la Tabla 4.2 y el cuestionario utilizado para esta evaluación en el Anexo B Tabla B.2. Estos resultados muestran que los profesores creen que esta herramienta es capaz de ayudar a sus alumnos en el aula, especialmente si están motivados para realizar las tareas involucradas en el programa. Sus opiniones generales sobre la adaptación de este sistema fueron positivas, aunque es necesario mejorar la accesibilidad de los ejercicios ya que los usuarios estaban teniendo problemas.

Características	EDES1	EDES2	F	ETE	EI	Media
Adaptación a las demandas del profesorado	4	4	3	5	3	3.8
Beneficio del sistema para los estudiantes	4	3	4	5	4	4
Adaptación de las actividades para los estudiantes	4	3	3	5	3	3.6
Motivación de las actividades para los alumnos	5	5	5	5	3	4.6
Accesibilidad de las actividades	3	2	2	4	4	3
Personalización de las actividades	3	3	3	4	3	3.2

Tabla 4.2: Resultados de la evaluación de la parte educativa. EDES1: Experto en Educación Especial / EDES2: Experto en Educación Especial / F:Fisioterapeuta / ETE: Experto en Técnicas Educativas / EI: Experto en Informática

4.3.3. Encuesta a los estudiantes

Esta evaluación pone de manifiesto que los estudiantes se motivaron (ver Tabla 4.3) cuando usaron el software y también mostraron más atención con la retroalimentación visual y auditiva, con lo cual pudieron completar las actividades. El comportamiento de los estudiantes fue supervisado mientras realizaban las tareas.

Características	EDES1	EDES2	F	ETE	EI	Media
Se reconocen a ellos mismos	4	3	5	3	4	3.8
Identifican sus movimientos en la pantalla	4	3	5	4	4	4
Movimientos de lateralidad correctos	4	2	3	4	3	3.2
Entiende las instrucciones dadas por el profesor	4	2	5	4	4	3.8
Imita el movimiento de la instrucción	3	3	5	4	4	3.8
Realiza la intrucción dada independientemente	2	2	5	3	4	3.2
Realiza la instrucción con modelación	4	3	2	4	3	3.2
Realiza la instrucción con moldeamiento	4	3	2	4	3	3.2
Realiza la instrucción con encadenamiento	3	3	3	4	3	3.2
Repite las directrices verbales	1	3	2	2	4	2.4
Se regulan autónomamente durante la actividad	2	2	2	3	3	2.4
Los estudiantes se sienten motivados con la actividad	4	5	5	5	4	4.6
Demuestra un aumento del interés con incentivos externos	5	5	5	5	5	5
Expresa su frustración con las dificultades	4	3	3	2	1	2.6
Grado de éxito realizando las actividades	4	4	4	4	4	4

Tabla 4.3: Evaluación de los resultados de los estudiantes. EDES1: Experto en Educación Especial / EDES2: Experto en Educación Especial / F:Fisioterapeuta / ETE: Experto en Técnicas Educativas / EI: Experto en Informática

Los resultados de la encuesta son muy similares a la anterior y el cuestionario utilizado para esta evaluación se puede consultar en el Anexo B, Tabla B.3. De estos resultados

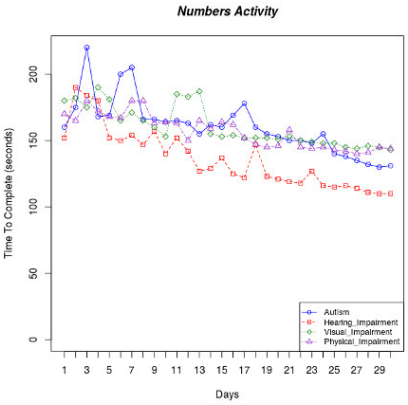
se puede concluir que algunos de los estudiantes pudieron realizar las tareas de forma independiente, mientras que otros aún necesitaban ayuda para realizar dichas tareas. Otro resultado positivo es que los alumnos no mostraron frustración cuando encontraron dificultades al interactuar con el sistema. Por tanto, este tipo de interacción puede ser ideal para alumnos con características similares en el aula.

4.3.4. Encuesta de las actividades

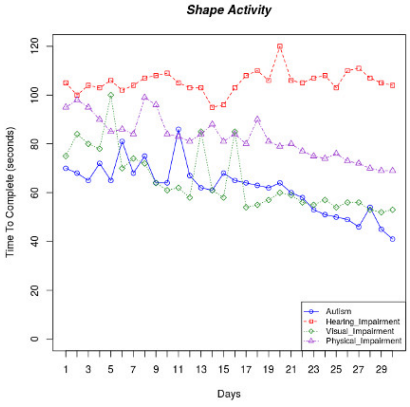
En esta sección se analizan los resultados de las evaluaciones con respecto al tiempo que un estudiante tardó en realizar la actividad, así como los errores cometidos durante este tiempo. Se han evaluado las siguientes actividades: Coordinación, Números, Formas y Grafomotricidad.

En general, los estudiantes han reducido el tiempo de ejecución en las actividades y el número de errores cada día. El gráfico muestra que los valores más altos provienen de los primeros diez días del período de evaluación (ver Figura 4.7 y 4.8). Esto se debe a que los alumnos no tenían experiencia previa con la nueva aplicación y tuvieron que adaptarse a ella.

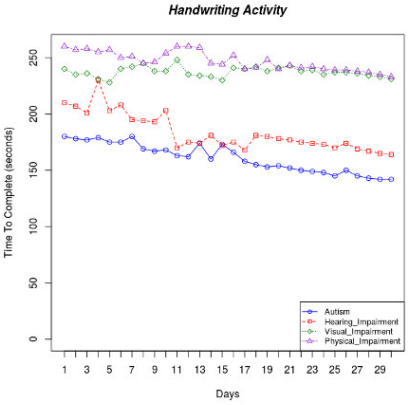
Es más difícil para el alumno con discapacidad física interactuar con el ejercicio de números en comparación con el ejercicio de formas porque en estos ejercicios las ubicaciones de las formas están más cerca entre sí que la ubicación de los números. Como consecuencia, el alumno debe hacer un gran esfuerzo para lograr el objetivo final en el ejercicio de los números (ver Figuras 4.7a y 4.8a) y en el ejercicio de las formas (ver Figuras 4.7b y 4.8b). Estos estudiantes han tenido un ratio bajo en la actividad de coordinación que ha sido diseñada para alumnos con discapacidad física (ver Figuras 4.7d y 4.8d). Sin embargo, los estudiantes pudieron hacer los movimientos de manera rápida y fluida durante los últimos días de los experimentos.



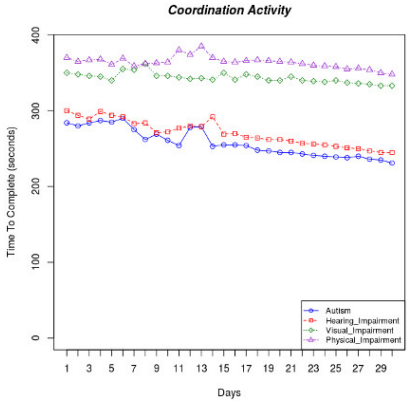
(a) Gráfica de tiempos de la actividad de números.



(b) Gráfica de tiempos de la actividad de formas.

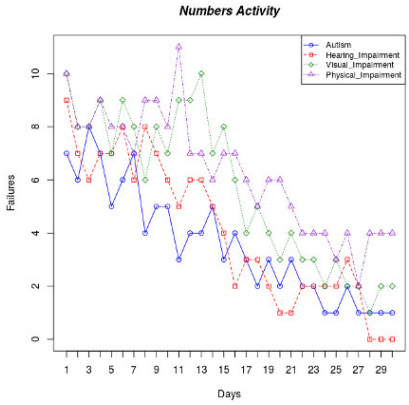


(c) Gráfica de tiempos de la actividad de grafomotricidad.

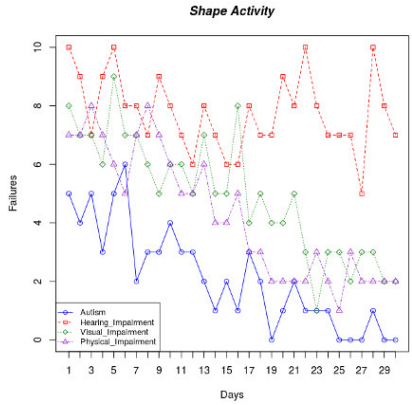


(d) Gráfica de tiempos de la actividad de coordinación.

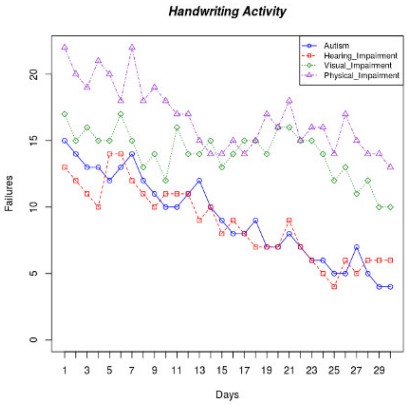
Figura 4.7: Resultados del tiempo de ejecución.



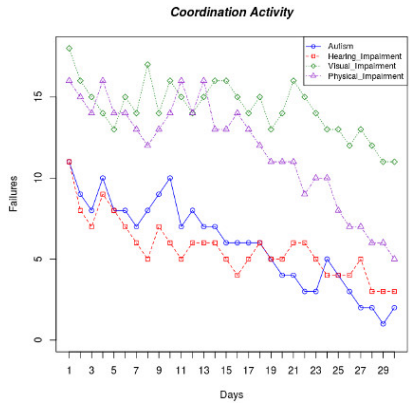
(a) Gráfica de errores de la actividad de números.



(b) Gráfica de errores de la actividad de formas.



(c) Gráfica de errores de la actividad de grafomotricidad.



(d) Gráfica de errores de la actividad de coordinación.

Figura 4.8: Resultados de la tasa de errores.

4.4. Resumen

En este capítulo se ha mostrado el sistema desarrollado inicialmente con Microsoft Kinect v1 como medio de interacción para que los usuarios sean capaces de controlar una aplicación software, que se podría extrapolar a un sistema informático, sin la necesidad de usar teclado y ratón sino solo con el movimiento de su cuerpo.

La arquitectura es explicada, así como cada uno de sus componentes:

- El **módulo de entrada** se encarga de obtener los datos del dispositivo para que puedan ser utilizados por el sistema.
- El elemento **HCI** se va a encargar de tratar la información referente a la interacción como por ejemplo, la posición de las articulaciones del usuario en todo momento.
- El **sistema inteligente** tiene el objetivo de comprobar ciertos aspectos del sistema para mejorar la experiencia de usuario durante la realizacíon de las actividades.
- El **ciclo de vida de juego** se implementó debido a que las actividades se diseñaron como juegos porque los usuarios que lo iban a probar eran estudiantes de un centro de educación especial.
- El módulo de salida se encarga de transmitir los aspectos visuales y de sonido del sistema al usuario.

A continuación, se describen las actividades que se han creado para que los usuarios puedan probar el sistema. Estas actividades se han dividido en dos módulos. El primer módulo está orientado especialmente a las habilidades físicas del individuo mientras que el segundo módulo tenía el objetivo adicional de que el usuario utilizara las habilidades cognitivas también. Este primer módulo estaba compuesto por una actividad mientras que el segundo estaba formado por cuatro actividades. La actividad del primer módulo consistía en unir un conjunto de nodos, que eran predefinidos previamente, con una parte del cuerpo determinada que se seleccionaba antes de empezar. Las actividades del segundo módulo son:

- **Números**: En esta actividad el usuario tendrá que tocar los números en orden ascendente.
- **Formas**: El usuario tiene que tocar la forma correcta de entre las que aparecen según las instrucciones indicadas al inicio del ejercicio.
- **Grafomotricidad**: En este ejercicio se tienen que t razar líneas horizontales, verticales o arbitrarias dependiendo de la distribución de los elementos en la pantalla.

En esta parte del trabajo se colaboró con el Centro de Eduación Especial Princesa Sofía de la provincia de Almería y por lo tanto las actividades descritas anteriormente se desarrollaron con las sugerencias y conocimientos que los profesores aportaron en toda la fase de realización. Esta colaboración supuso que los estudiantes de dicho centro fueran los participantes del estudio que se realizó para comprobar la validez del sistema.

En esta evaluación se realizó una encuesta que fue rellenada por un conjunto de expertos valorando la usabilidad, modalidad educativa y comportamiento de los estudiantes. Además, se realizaron los experimentos con usuarios que tenían que completar las actividades donde se midió el tiempo y el número de errores para obtener conclusiones acerca del sistema.

Los resultados demuestran que incluir un sistema con otro tipo de interacción hace difícil para los estudiantes con diversidad funcional seguir el ritmo de las dinámicas de la actividad y las instrucciones del profesor, pero con la estimulación apropiada (en este caso el *feedback*), los estudiantes se pueden sentir cómodos con este tipo de interacción. Otro aspecto a destacar es el hecho de que los resultados fueron mejorando hasta alcanzar su valor óptimo al final del proceso de evaluación, lo que indica que los usuarios necesitan un período de adaptación para usar este nuevo tipo de interacción pero que son capaces de utilizarlo e interaccionar con las diferentes actividades.

Capítulo 5

Sistema interactivo adaptativo

Capítulo 5

Sistema interactivo adaptativo

Contenidos

Una vez concluido el estudio previo con el sensor Microsoft Kinect v1 se verificó que este dispositivo era útil para realizar interacción natural debido a la posibilidad de detectar el movimiento del cuerpo del usuario y el reconocimiento de gestos. Los resultados de esa fase del trabajo determinaron no solo que esta herramienta era apta para la interacción natural, sino que incluso las personas que presentan algún tipo de discapacidad podían beneficiarse de sus características. Sin embargo, se detectó que una de las limitaciones era la adaptabilidad, especialmente al hacer los ensayos con personas con necesidades especiales. Aunque no era posible ni recomendable que el objetivo del diseño fuera que el sistema se adaptara a cada usuario particular [Cooper et al., 2003], sí que se propuso diseñarlo para que la interacción se adaptara a ciertos usuarios con unas características específicas.

La característica principal del sistema propuesto es su adaptabilidad en relación con la interacción, siendo este un aspecto fundamental de cualquier sistema informático. Esta parte del trabajo ha sido diseñada para usuarios con discapacidad física o sensorial debido a que los participantes de la evaluación tenían estas características. Además, son unos usuarios adecuados para este estudio porque el objetivo consiste en mejorar la interacción según las características del usuario para de este modo asegurar una experiencia óptima. Se organizaron dos experimentos para validar este estudio. El primer experimento fue una evaluación de expertos donde se aplicacron las técnicas de Recorrido Cognitivo con usuarios (Cognitive Walkthrough) y Pensar en Voz Alta (Thinking Aloud). Los expertos que participaron en el experimento fueron un experto en sistemas interactivos, un especialista en educación especial y dos profesores del centro de educación especial.

Después, se realizó un experimento con doce estudiantes del centro de educación especial. En dicho experimento participaron estudiantes con autismo, discapacidad física, auditiva y visual. Los participantes tenían que completar las actividades desarrolladas en este trabajo donde se medían el tiempo y los errores que realizaron durante su actuación. Había dos tipos de actividades; una actividad para asociar conceptos sobre un tema conceptual y otra actividad para trabajar los problemas de lateralidad. La actividad para asociar conceptos tenía dos versiones: Conceptos sobre animales y conceptos sobre vehículos. En esta propuesta, la interacción natural fue integrada porque era más conveniente para los estudiantes con diversidad funcional puesto que el único requerimiento para la interacción con el dispositivo es que el usuario tenía que posicionarse a una distancia determinada del dispositivo. El sensor Microsoft Kinect v2 fue incluido principalmente debido a que es capaz de reconocer el movimiento del cuerpo pero también porque tiene una resolución mayor tanto en su cámara de color como de su cámara de profundidad, es capaz de reconocer hasta 25 joints del cuerpo, el rango de distancia con el que se puede trabajar es más amplio y permite reconocer 6 personas al mismo tiempo [Cai et al., 2017]. Además, este dispositivo es más preciso, tiene mejor capacidad de respuesta y una capacidad intuitiva que lo diferencia de su predecesor [Samir et al., 2015]. Los elementos necesarios para realizar una interacción adaptable en este proyecto son básicamente: el modelo de usuario, el modelo dispositivo-interacción y las reglas de adaptación.

5.1. Modelo de usuario

El modelo de usuario elegido para este trabajo es un modelo basado en características [Brusilovsky and Millán, 2007]. Los tutores registran a sus estudiantes en el sistema, rellenando los datos necesarios para la creación del modelo de usuario del estudiante. Este modelo almacena las siguientes características del usuario: nombre completo, edad, sexo, problemas de lateralidad y discapacidad.

En este trabajo las características más relevantes son: problemas de lateralidad y discapacidad. Cuando hablamos de discapacidad, es recomendable mencionar la Clasificación internacional del funcionamiento, de la discapacidad y de la salud creado por la Organización Mundial de la Salud[1], cuyo objetivo principal es establecer un lenguaje estándar para la descripción de la salud y sus estados donde la discapacidad es uno de ellos, entre otros [Cieza and Stucki, 2008]. Las discapacidades contenidas en el modelo de usuario están incluidas en esta clasificación y son: personas con autismo, discapacidad física, discapacidad auditiva y discapacidad visual. Estas características han sido incluidas por el tipo de usuario que participó en esta investigación. La característica referente a los problemas de lateralidad describe cuando los usuarios no son capaces de distinguir perfectamente entre el lado izquierdo y el lado derecho de su cuerpo. De este modelo cabe destacar que las características de discapacidad y problemas de lateralidad tienen una relevancia destacada en el proceso de adaptación puesto que afectarán directamente al modelo dispositivo-interacción.

5.2. Modelo dispositivo-interacción

El modelo dispositivo-interacción tiene en consideración las características del usuario, aunque este modelo se centra principalmente en el dispositivo. Esto es así debido a que el dispositivo es el medio por el cual los estudiantes son capaces de usar el sistema y esta es la razón por la que tiene un papel importante en este modelo. El objetivo de este modelo es optimizar la interacción del usuario con el sistema gracias a las características que posee el dispositivo. Las características del modelo dispositivo-interacción son:

- **Detección de la posición bípeda**: Esta propiedad determina si el usuario está sentado o de pie. Este sistema será usado por los estudiantes que usan silla de ruedas y detectar esta situación es imprescindible para la adaptación de la actividad. Cuando se detecta que el usuario está sentado, un seguimiento más meticuloso de la parte superior del cuerpo es llevado a cabo mientras que la parte inferior se ignora. Por el contrario, si se detecta que el usuario se encuentra de pie, todos los joints que Microsoft Kinect v2 es capaz de reconocer serán tenidos en cuenta.
- **Activación de la cámara RGB**: Algunos de los estudiantes tienen un nivel cognitivo muy bajo y no son capaces de asociar sus movimientos con los elementos de la pantalla a menos que ellos puedan verse en espejo en la pantalla. Esta característica hace posible que se pueda activar la cámara RGB del dispositivo para que

[1] Clasificación internacional del funcionamiento, de la discapacidad y de la salud - `https://www.who.int/classifications/icf/en/`

de esta forma los estudiantes puedan identificarse en la pantalla e interactuar con el entorno.

- **La distancia de profundidad**: Este atributo almacena la distancia óptima a la que el usuario tiene que situarse del dispositivo Kinect para interaccionar adecuadamente. Sin embargo, existe una distancia recomendada para este sensor (1.2 - 3.5m) dependiendo de la altura del usuario.
- **El movimiento del brazo**: Esta característica identifica si el usuario puede mover ambos brazos o solamente uno de ellos. La interacción con el sistema requiere que el usuario pueda mover uno de los brazos como mínimo. Este atributo está especialmente diseñado para los estudiantes que tengan algún tipo de discapacidad física y solo puedan mover uno de sus brazos. En el caso de que el usuario pueda usar ambos brazos, el sistema determinará el brazo dominante para facilitar la interacción dependiendo de si el usuario es diestro o zurdo.

5.2.1. Reglas de adaptación

Las reglas de adaptación son básicas para adaptar las actividades a las características del usuario. En un principio las reglas en el proyecto estaban basadas en Accessible Games Standard v1.0[2] y Game Accessibility Guidelines[3]. Esto se debe a que las actividades del proyecto incluían mecánicas basadas en el juego y Kinect es una herramienta que fue diseñada para interaccionar con juegos. Sin embargo, se hicieron mejoras en las reglas de adaptación después de recibir el *feedback* de los expertos en educación especial que participaron en el estudio. Estas reglas fueron definidas previamente y son activadas dependiendo del modelo de usuario y las características del modelo dispositivo-interacción. Las reglas más importantes son descritas posteriormente. La Tabla 5.1 resume las reglas creadas para este estudio y muestra las acciones o cambios que el sistema hace cuando detecta que el usuario tiene discapacidad física, auditiva, visual, autismo, limitación en el movimiento de sus extremidades o si se encuentra en silla de ruedas.

La regla #1 aplica la acción cuando el usuario posee una discapaciad visual. En este caso, todas las instrucciones de la actividad se transmiten por audio, el color de fondo es negro y los objetos 3D con los que interacciona el usuario son amarillos para contrastar con el fondo y hacer más sencilla su identificación. El modo interactivo está basado en seguimiento del brazo que el usuario pueda mover. En el apartado del modo de interacción, cuando se señala que es a través de colisión, significa que cuando el usuario tiene que interaccionar con los elementos en la escena, la colisión entre el cursor controlado por el usuario y los elementos 3D será detectada. El *feedback* de la actividad es mediante audio para que el usuario sepa cuando se ha equivocado.

La regla #2 es para usuarios con discapacidad auditiva. En esta situación, todas las instrucciones se muestran de forma visual y el modo de interacción es basado en reconocimiento de gestos. El modo sin retraso en la sección de reconocimiento de gestos significa que hay un tiempo variable cuando se mide el tiempo desde la posición inicial

[2]Accessible Games Standard v1.0 - `https://bbc.in/31Wx2Hi`
[3]Game accessibility guideliness - `http://gameaccessibilityguidelines.com/`

#	**Discapacidad**	**I**	**CF**	**C3D**	**MI**	**Fed**	**G**	**MIV**	**D**	**DM**
1	Visual	Audio	Negro	Amarillo	Colisión	Audio	No	No	Estándar	Brazo dominante
2	Auditiva	Visual	Imagen	Normal	Gestos	Visual	Yes	No	Estándar	Brazo dominante
3	Physical	Audio	Imagen	Normal	Colisión	Visual&Audio	No	No	Estándar	-
4	Autismo	Audio	Imagen	Normal	Drag&Drop	Visual&Audio	No	Yes	Estándar	Brazo dominante
5	Física (silla de ruedas)	Audio	Imagen	Normal	Colisión	Visual&Audio	No	No	Reducida	Brazo dominante
6	Física (mov. brazo derecho)	Audio	Imagen	Normal	Colisión	Visual&Audio	No	No	Estándar	Brazo derecho
7	Física (mov. brazo izquierdo)	Audio	Imagen	Normal	Colisión	Visual&Audio	No	No	Estándar	Brazo izquierdo
8	Física (mov. ambos brazos)	Audio	Imagen	Normal	Colisión	Visual&Audio	No	No	Estándar	Brazo dominante

Tabla 5.1: Reglas de adaptación. (I: Instrucciones / CF: Color de fondo / C3D: Color objetos 3D / MI: Modo de interacción / Fed: Feedback / G: Gestos / MIV: Mostrar iconos visuales / D: Distancia entre elementos / DM: Detección del movimiento).

a la final. En este caso, esta variable es asignada al valor mínimo en el cual el gesto es identificado inmediatamente, de esta manera el usuario no se siente frustrado al tener que esperar por la respuesta del sistema a su acción. Por último, el *feedback* es visual.

La regla #3 hace referencia a las acciones que se ejecutan para los usuarios con discapacidad física. En esta situación, todas las instrucciones de las actividades son audio. Las instrucciones podrían haber sido de forma visual porque según el perfil de este usuario no padece ninguna discapacidad sensorial pero este modo fue elegido porque es más rápido para los estudiantes el hecho de escuchar las instrucciones que leerlas. El modo interactivo está basado en una detección de movimiento por colisión y el *feedback* de la actividad combina audio y visual.

La regla #4 está relacionada con los usuarios con autismo. En este caso, todas las instrucciones de la actividad son en audio por las mismas razones que la regla anterior. El modo de interacción se basa en la detección del movimiento *drag & drop*. En este tipo de interacción, cuando la colisión entre el cursor y un elemento es detectado, este elemento es arrastrado hasta que alcanza el área donde el usuario puede soltarlo. Algunos iconos visuales son mostrados junto a los objetos 3D para ayudar a la comprensión de los estudiantes ya que los estudiantes con autismo están acostumbrados a trabajar con pictogramas. El *feedback* que se utiliza es tanto audio como visual.

La regla #5 se aplica a los usuarios que utilizan silla de ruedas. En este caso, la acción consiste en reducir la distancia entre los elementos porque en esta posición el movimiento es más limitado que cuando el usuario está de pie. En este sentido, el usuario es capaz de interactuar con los diferentes elementos de la interfaz fácilmente.

La regla #6 es para los usuarios que tienen una discapacidad física que afecta al movimiento de su brazo izquierdo mientras que la regla #7 está diseñada para aquellos que su discapacidad física afecta al movimiento de su brazo derecho. En esta situación, los *joints* detectados par el seguimiento y la detección del movimiento están relacionados con el brazo izquierdo para el primer caso y con el brazo derecho para el segundo.

La regla #8 está creada específicamente para usuarios con discapacidad física que pueden mover ambos brazos. Esta regla les permite usar cualquiera de los dos brazos (normalmente el brazo dominante) para su mayor comodidad durante el proceso de interacción.

5.2.2. Sistema adaptado con un modelo dispositivo-interacción

El diseño soporta diferente tipos de dispositivos de entrada que interaccionan en el sistema (ver Figura 5.1). Estos dispositivos son el ratón, el teclado y Microsoft Kinect v2. Esta arquitectura está formada por distintos subsistemas que hacen este sistema adaptable de acuerdo a las características de los usuarios, especialmente en relación con la interacción. A continuación, la funcionalidad de cada componente del sistema es descrito.

El sistema de interacción se compone de tres componentes: *Reconocimiento E/S*, *Reconocimiento de gestos* y *Detección del movimiento*. El uso de cada componente depende del dispositivo que el usuario está usando para la interacción. Por ejemplo, cuando el tutor está organizando perfiles o registrando a un nuevo usuario, los dispositivos de entrada son el ratón y el teclado puesto que esta tarea es más difícil de completar si se

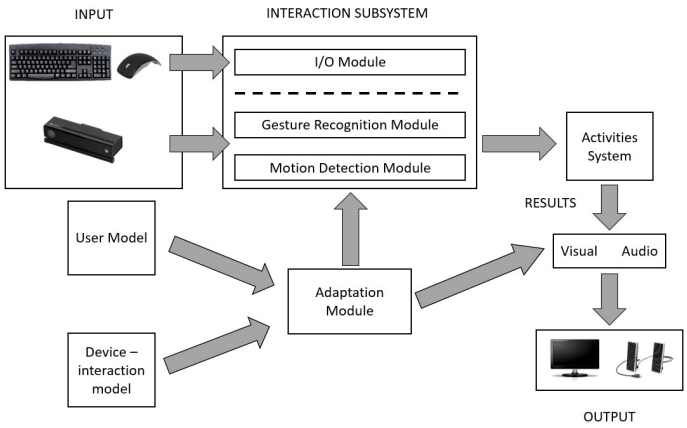


Figura 5.1: Arquitectura del sistema.

usara el dispositivo Kinect v2. Sin embargo, cuando un estudiante está realizando una actividad, utiliza el sensor Kinect v2 porque es más apropiado para completar esta tarea. El componente de *Reconocimiento E/S* se activa cuando se usa el ratón y el teclado y el dispositivo Kinect v2 es excluido de este proceso para que la interacción sea más fluida. Los componentes denominados *Reconocimiento de gestos* y *Detección del movimiento* solo pueden ser activados cuando Microsoft Kinect v2 es usado para la interacción. Sin embargo, el *Módulo de adaptación* es quien toma la decisión de activar cada componente. Este *Módulo de adaptación* se compone de un conjunto de reglas basadas en el modelo de usuario para identificar las características del usuario y ofrecer un modo de interacción adecuado para el usuario.

El módulo de *Reconocimiento de Gestos* identifica cuatro gestos; levantar el brazo izquierdo, levantar el brazo derechos, mover el brazo horizontalmente hacia la izquierda y hacia la derecha. En la fase inicial se decidió integrar estos gestos básicos para hacer la ejecución más fácil para ciertos estudiantes. El proceso de reconocimiento de gestos se realiza con una máquina de estados finitos donde cada estado compruebe si la posición actual del usuario es idéntica a la que es almacenada en ese estado. Los estados tendrán un estado inicial y otro final, donde se comprobará la posición inicial y final del gesto para el reconocimiento y un conjunto de estados intermedios que evaluarán el gesto para verificar que este gesto coincide con el que el usuario quiere que sea reconocido. La evaluación de la posición inicial y final no es suficiente porque el usuario puede realizar el gesto inicial (un movimiento circular) y posicionarse en la posición final cuando el sistema lo que quiere reconocer es un movimiento ascendente.

El objetivo de estos estados intermedios es verificar que la trayectoria del movimiento es correcta mediante el proceso de validación del gesto. En estos gestos la mano derecha, la mano izquierda y el hombro izquierdo son considerados. Para validar la posición inicial, la posición de la mano hecha por el hombro izquierdo tiene que ser comprobada. La validación de la posición final tiene en cuenta dos aspectos: la posición de la mano

con respecto al cuerpo es verificado y segundo, que el gesto se ha realizado dentro de un tiempo determinado, por ejemplo, si al usuario le lleva mucho tiempo ir desde la posición inicial hasta la final, el gesto no será considerado como válido.

Tan pronto como se activa el módulo de reconocimiento de gestos, el *listener* de gestos comenzará también. Su función comprueba constantemente si el usuario cumple con los criterios del estado inicial de algunos de los gestos para completar el proceso de reconocimiento de gestos e informar al sistema si el gesto se ha realizado correctamente.

El módulo de *Detección de movimiento* utiliza el software de kit de desarrollo (SDK)[4] de Microsoft Kinect v2: un kit de desarrollo para crear aplicaciones que soporta la tecnología Kinect para asignar uno de los 25 *joints* que reconoce y, por lo tanto, solo realizará el seguimiento de esos *joints* específicos. El *joint* será seleccionado dependiendo del *Módulo de adaptación*, que es responsable de informar al *Módulo de Detección de Movimiento* sobre qué *joint* seguir. Este módulo trabaja con la mano izquierda y derecha porque su objetivo es mover un objeto 3D con forma de mano como si se tratara de un cursor y de esta manera interactuar con otros objetos 3D que se mostrarán en la pantalla. Por lo tanto, para que la mano se comportara como un cursor y hacer que el objeto 3D se mueva al unísono con el movimiento de la mano del usuario, las coordenadas X, Y y Z de ambos tienen que coincidir.

Además del *Subsistema de Interacción*, el entorno propuesto tiene un *Subsistema de actividades*, para organizar las actividades que realiza el usuario según su perfil y estas se adaptan a sus características. Este sistema tiene en cuenta dos características del modelo de usuario: la discapacidad y el problema de lateralidad. Si el usuario no tiene ningún problema de lateralidad, el sistema no proporciona esta opción para que el usuario la seleccione en la interfaz. El *Módulo de adaptación* obtiene la información del modelo de usuario, el modelo dispositivo-interacción y las reglas asociadas que se envían al *módulo de Definición de Actividades*, el cual es el encargado de crear actividades que permitan que las instrucciones, el *feedback*, los componentes, la lógica y la interacción tengan características específicas dependiendo de la discapacidad del usuario. Además, este módulo también permite que el *Modelo de interacción* pueda hacer uso de las características del dispositivo Kinect v2 según las preferencias que se han guardado durante el proceso de verificación automática. De esta forma, además de organizar la actividad según el tipo de discapacidad del usuario, estas tres características también se consideran: la postura habitual del usuario (silla de ruedas o de pie), la activación de la cámara RGB o el brazo utilizado para la interacción.

Los resultados del componente del entorno muestran un *feedback* a los usuarios dependiendo de sus acciones en el sistema. El *feedback* puede ser visual o auditivo. El *feedback* visual se representa con una cara sonriente o triste y el audio con un sonido particular del objeto que ha sido seleccionado o un sonido asociado con un error. En este proceso, también se comprueba si la selección del usuario corresponde con la petición que realiza la actividad. Un tipo diferente de *feedback* sería dado al usuario dependiendo de si la acción es correcta o errónea. El *Módulo de Adaptación* también forma parte de este componente ya que, dependiendo del tipo de discapacidad, el *feedback* se muestra de una manera diferente. Por ejemplo, si el estudiante tiene discapacidad auditiva, el

[4]Kinect Windows SDK - `https://msdn.microsoft.com/en-us/library/dn799271.aspx`

feedback será visual, mientras que si el el alumno tiene discapacidad visual, el *feedback* se presentará en forma de audio.

5.2.3. Actividades interactivas propuestas

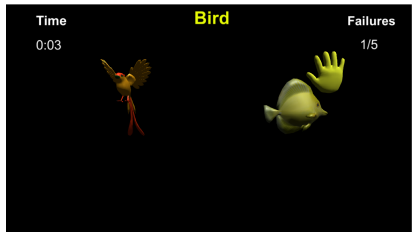
En esta sección, las actividades implementadas (ver Tabla 5.2) en el prototipo son descritas y cada especificación relativa a las características del usuario se explica en detalle. Se desarrollaron dos tipos de actividades. Los profesores propusieron actividades que fueran útiles para sus estudiantes en ese momento y nos guiaron durante la fase de desarrollo. A sus estudiantes les resultó difícil diferenciar entre algunos conceptos y su dominio de las manos, ojos, pies u oídos del lado derecho o izquierdo no fue consistente (lateralidad cruzada). Esta lateralidad cruzada fue descubierta hace unos cuarenta años y afecta a la organización de las funciones superiores en nuestro sistema, este trastorno afecta al aprendizaje del lenguaje y las matemáticas, análisis, lógica, de comprensión y concentración, percepción tiempo-espacio y equilibrio, entre otros [Ferrero et al., 2017]. Por tanto, el objetivo de una actividad es que el alumno asocie conceptos y de la otra es mejorar el problema de lateralidad. El usuario puede elegir entre dos temas en la actividad de asociación de conceptos: animales o vehículos. A partir de ahí, el tutor seleccionará uno de ellos dependiendo de lo que quiera enseñar en cada ocasión.

Modo	**Asociación de conceptos**	**Actividad de lateralidad**
Visual	Instrucciones: Audio Color de fondo: Negro Color objetos 3D: Amarillo Interacción: Detección del movimiento Feedback: Audio	Instrucciones: Audio Color de fondo: Negro Color objetos 3D: Amarillo Interacción: Detección del movimiento Traslación del objeto
Auditivo	Instrucciones: Visual Interacción: Reconocimiento de gestos Feedback: Visual	Instrucciones: Visual Interacción: Reconocimiento de gestos Feedback: Visual Desplazamiento del objeto (izquierda / derecha)
Físico	Interacción: Adaptación según la movilidad del usuario Posición de los elementos según el movimiento Feedback: Visual y audio	Instrucciones: Audio Interacción: Detección del movimiento Intervalo de desplazamiento según el movimiento Localización de bordes en la pantalla Feedback: Visual and audio
Autismo	Instrucciones: Visual y audio Interacción: Drag & drop Identificación de elementos mediante pictogramas Feedback: Visual y audio	Instrucciones: Audio Mayor número de elementos de interacción Desplazamiento de objetos Identificación mediante pictogramas Interacción: Drag & drop Feedback: Visual and audio

Tabla 5.2: Resumen de las características principales de las actividades.

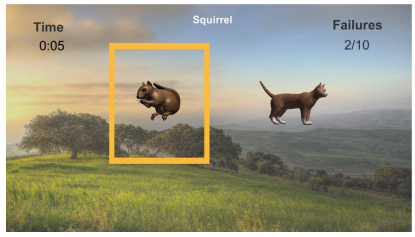
5.2.3.1. Actividad sobre la asociación de conceptos

El objetivo de esta actividad es que los alumnos aprendan conceptos dentro de un tema (ver Figura 5.2) y el sistema proporciona la información esencial para el estudiante con el propósito de identificar el modelo que debe seleccionar. Esta información se muestra de forma visual o transmitida en audio según la discapacidad del usuario. Cuando el alumno ha seleccionado una de las opciones, le permitirá al usuario modificarlas dentro de un cierto período de tiempo. Entonces, el sistema dará un *feedback* positivo o negativo para el usuario dependiendo de la acción del usuario. Esta secuencia puede repetirse el número de veces que el tutor considere oportuno. Los medios de interacción son diferentes según el modelo de usuario y el modelo dispositivo-interacción. Las diferentes versiones según la discapacidad del usuario se describen a continuación.

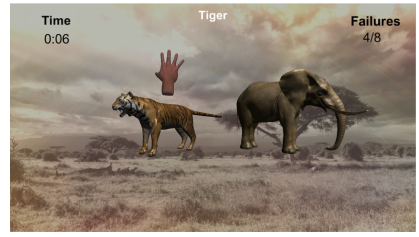


(a) Contraste entre los elementos y el fondo para los estudiantes con discapacidad visual.

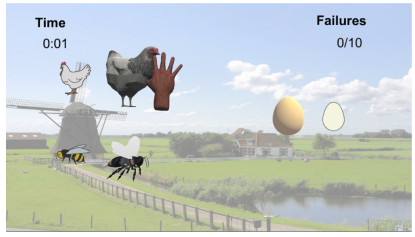
(b) La pantalla anterior antes de iniciar la actividad para estudiantes con discapacidad auditiva.



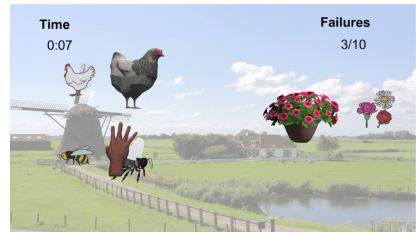
(c) Se selecciona la opción elegida.



(d) Los objetos están más cerca para los casos de discapacidad física.



(e) Actividad para usuarios con autismo.



(f) Actividad para usuarios con autismo.

Figura 5.2: Pantallas relacionadas con la actividad de asociación de conceptos.

- **Discapacidad visual**: en este caso, se cambian los colores de la interfaz gráfica: el color de fondo es negro y el color del objeto 3D es amarillo para crear contraste (ver Figura 5.2a). Las instrucciones mostradas al inicio de la actividad son audibles. La interacción es a través de la detección de movimiento, donde los usuarios pueden controlar un cursor con forma de mano según su brazo dominante. Cuando se selecciona uno de los elementos, aparecerá un rectángulo amarillo alrededor del elemento seleccionado para saber qué elemento ha sido seleccionado por el usuario. El *feedback* es exclusivamente de audio.

- **Discapacidad auditiva**: en esta versión, los elementos y los colores de la interfaz se muestran sin ninguna modificación (ver Figura 5.2c). Las instrucciones se muestran de forma visual (ver Figura 5.2b) y la interacción es a través de reconocimiento de gestos. Los gestos son levantar el brazo derecho y el brazo izquierdo. Si el brazo izquierdo está levantado, el elemento ubicado en el lado izquierdo de la pantalla será seleccionado, pero si se levanta el brazo derecho, el elemento situado en el lado derecho se seleccionará. El *feedback* en esta versión es totalmente visual. Si la respuesta es correcta, aparecerá una cara sonriente en la pantalla; en caso contrario, se muestra una cara triste.

- **Discapacidad física**: cuando el usuario tiene una discapacidad física, los cambios en relación con la interacción dependerá de los valores en el modelo dispositivo-interacción: si la persona está usando su brazo izquierdo o derecho. Por ejemplo, si una persona solo puede mover su brazo derecho, el sistema seleccionará ese brazo para interaccionar con los diferentes elementos de la interfaz. Otro aspecto relevante es que si el usuario tiene que utilizar una silla de ruedas, los elementos estarán más cerca y centrados en la pantalla (ver Figura 5.2d) para que el alcance del movimiento sea más reducido, haciéndolo más accesible para el usuario. El *feedback* es visual y auditivo, a diferencia de la situación anterior.

- **Autismo**: las instrucciones se muestran de forma visual y auditiva. El modo de interacción con los elementos es distinta porque sigue una metodología que es más complicada como *drag & drop*. El objetivo es asociar correctamente los elementos que se encuentran en el lado izquierdo de la pantalla con los del lado derecho (ver Figuras 5.2e y 5.2f). Por esta razón, cuando el cursor colisiona con un elemento ubicado en el lado izquierdo, este elemento es arrastrado por el cursor mediante una técnica de seguimiento y cuando hace contacto con el elemento de la derecha, el proceso de seguimiento se detiene. Hay algunos pictogramas junto a los diferentes objetos para que sea más fácil para ellos identificar cada elemento. El *feedback* es visual y auditivo.

5.2.3.2. Actividad de lateralidad

El objetivo de esta actividad es trabajar la lateralidad izquierda y derecha. En el comienzo de la actividad, aparece una pelota en el medio de la pantalla y el usuario tiene que tocarlo con el cursor o realizar el gesto correspondiente. Cuando el cursor y el objeto 3D hacen contacto entre sí o se realiza el gesto correcto, esta pelota se moverá en

la pantalla. Esta traducción depende del problema de lateralidad de cada alumno: Si el usuario no reconoce el lado derecho, el objeto se moverá unos centímetros hacia la derecha, pero si el usuario no puede reconocer el lado izquierdo, entonces el objeto se moverá unos centímetros a la izquierda. Esta secuencia se repetirá varias veces hasta que la pelota alcance una posición predeterminada en la pantalla. La forma de interacción con el sistema depende de las características del usuario:

- **Discapacidad visual**: las instrucciones están en formato de audio. Esta actividad proporciona contraste de color como la actividad anterior en la que se pinta el fondo de negro mientras que la pelota es amarilla. La interacción de la actividad es reconocimiento de gestos, donde el usuario tiene que realizar un movimiento con el brazo tipo *swipe* con el brazo dominante en la dirección correspondiente, dependiendo de si el usuario tiene problemas con la lateralidad izquierda o derecha.

- **Discapacidad auditiva**: las instrucciones son visuales. El usuario interactúa con el sistema a través del reconocimiento de gestos. Cuando el gesto de levantar el brazo derecho es identificado, el balón se mueve hacia la derecha si la opción lateralidad derecha se seleccionó. De lo contrario, si se reconoce la elevación del brazo izquierdo, el balón se mueve hacia la izquierda si la opción lateralidad izquierda fue seleccionada.

- **Discapacidad física**: la interacción es la misma que en la versión de discapacidad visual, con la excepción de que el fondo no es negro y la pelota no es de color amarillo para proporcionar contraste entre los elementos. Cuando el usuario está en silla de ruedas, el modelo 3D no se mueve al otro extremo de la pantalla y el desplazamiento es menor que en otras versiones para que el usuario no tenga que hacer ningún movimiento incómodo para completar la actividad.

- **Autismo**: Al principio, las instrucciones de la actividad se comunican a través de audio. En esta versión, la pelota siempre comienza en el centro de la pantalla y se mueve una canasta de baloncesto 3D en lugar de la pelota. Cuando el usuario realiza un movimiento con el brazo tipo *swipe* en la dirección correcta, el balon se moverá hacia la canasta con una trayectoria en forma de parábola como si el usuario la hubiera lanzado. La canasta se mueve en la dirección correspondiente en función de la versión seleccionada.

5.3. Resultados

Esta sección describe el procedimiento para evaluar el sistema. Hay dos tipos de evaluación involucrada: una evaluación por expertos y una evaluación con usuarios finales. En la evaluación por expertos, una técnica denominada recorrido cognitivo con usuarios (Cognitive Walkthrough with Users) se utiliza en combinación con el método pensar en voz alta(Thinking Aloud), en el que participaron dos expertos y dos profesores. En la evaluación con los usuarios finales, estudiantes con diferentes tipos de discapacidad del centro de educación especial Princesa Sofía probaron el prototipo. Ciertos factores que

fueron considerados relevantes para este estudio se midieron con el fin de obtener una validación para este experimento.

5.3.1. Evaluación de expertos

Las técnicas de evaluación se clasifican en diferentes categorías según las características de los métodos. La clasificación general es: indagación, inspección y prueba [Granollers and Lorés, 2006]. El método de inspección es el más interesante para esta parte del experimento ya que los evaluadores (expertos) dan su juicio con respecto a la usabilidad y accesibilidad de los sistemas [Kushniruk et al., 2015]. El recorrido cognitivo es el método de inspección que se incluye en esta evaluación porque es adecuado para sistemas adaptativos interactivos [Dhouib et al., 2016] y por sus varias ventajas: es económico, se puede llevar a cabo en una etapa temprana de desarrollo, requiere poco esfuerzo y un experto suele ser suficiente. Por otro lado, se requiere un análisis muy detallado de las tareas y no hay calificación por parte de usuarios reales [Dhouib et al., 2016, Lewis and Wharton, 1997].

En esta evaluación, la colaboración del usuario es fundamental y por esta razón, dos evaluadores expertos en usabilidad de sistemas interactivos y necesidades especiales participaron en este experimento junto con dos profesores del centro de educación especial. En esta fase de la evaluación, el recorrido cognitivo se utilizó junto con un método llamado pensar en voz alta. La inclusión de esta técnica ha hecho que este proceso de evaluación sea más completo puesto que los usuarios tienen que expresar sus pensamientos mientras interactúan simultáneamente con el sistema. Además, se puede preguntar a los participantes por qué están realizando una determinada acción o si consideran algunos aspectos del prototipo confuso. De esta evaluación, se obtuvo información importante con la que mejorar la interfaz gráfica y la funcionalidad del prototipo con el fin de facilitar su implementación y hacerla más intuitiva para el usuario final.

Para realizar los experimentos utilizando el recorrido cognitivo con usuarios, primero deben realizar el recorrido cognitivo de la forma tradicional y una vez terminado, los usuarios se incorporan al estudio [Granollers and Lorés, 2006]. Con esta premisa, este método fue aplicado primero por expertos en usabilidad de sistemas interactivos y diversidad funcional. Posteriormente, los dos profesores realizaron el recorrido cognitivo en combinación con pensar en voz alta ya que esta técnica se suele aplicar a los usuarios finales del sistema. En este punto es necesario aclarar que la sesión con cada participante fue individual. En general se utiliza un prototipo de baja fidelidad en el recorrido cognitivo y para el método de pensar en voz alta, un prototipo de alta fidelidad [Zaini et al., 2019]. Sin embargo, se decidió utilizar un prototipo de alta fidelidad en esta evaluación porque ambas técnicas se combinaron y el objetivo era que todos los usuarios tuvieran el mismo prototipo para realizar dicha evaluación. Durante el recorrido cognitivo, los evaluadores tomaron notas sobre los aspectos negativos o los que necesitaban ser mejorados mientras realizaban las distintas tareas. Sin embargo, el método para recoger la información de los profesores fue distinta ya que se grabó el audio de los profesores durante su sesión. Esta forma de actuación se hizo como consecuencia de la técnica de pensar en voz alta, la cual consiste en decir lo que piensan sobre el sistema en ese momento, no obstante también se les permitió tomar notas en caso de que consideraran oportuno

escribir los aspectos que debemos tener en cuenta a la hora de mejorar el prototipo final.

Los expertos completaron una serie de tareas relacionadas con el sistema. Este sistema permite a cada tutor gestionar la información del alumno que se almacena en una base de datos (ver Figura 5.3). Se les pidió que completaran varias tareas que podrían dividirse en dos categorías: Gestión y Actividades. La categoría de *Gestión* englobaría las siguientes tareas:

- Registrar un docente
- Iniciar sesión en el sistema
- Añadir un estudiante
- Acceder al menú denominado Mi perfil
- Editar los datos del perfil
- Seleccionar un estudiante de la lista
- Editar los datos del estudiante
- Eliminar el perfil de un alumno
- Volver a la lista de alumnos
- Ir a la pantalla de selección de actividades
- Seleccionar una actividad
- Cerrar sesión

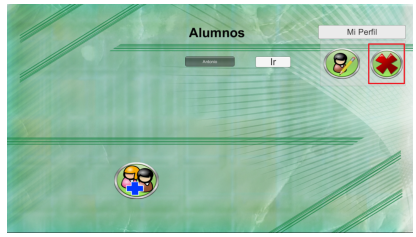
Por otro lado, en la categoría de *Actividades* los participantes tenían que completar las actividades de una manera correcta y equivocándose para comprobar los distintos *feedbacks*. Estas tareas serían:

- Completar la actividad para personas con autismo
- Completar la actividad para personas con discapacidad auditiva
- Completar la actividad para personas con discapacidad física
- Completar la actividad para personas con discapacidad visual

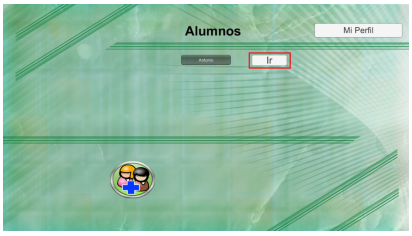
A partir de esta evaluación, algunos resultados se analizaron para mejorar el experimento, considerándose lo siguiente:

- Modificaciones en la interfaz de usuario, por ejemplo, añadir iconos en algunos botones en vez de texto o cambiar la localización de algunos elementos en la interfaz gráfica.
- Implantar el sistema de reconocimiento de gestos para los usuarios con discapacidad auditiva porque están acostumbrados a comunicarse con este tipo de interación, en lugar de la detección de movimiento.

(a) Añadir usuario.



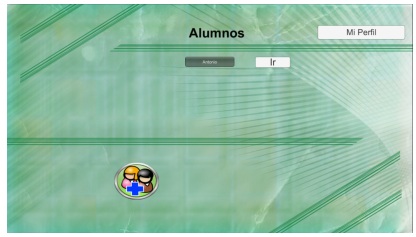
(b) Cerrar sesión.



(c) Botón de ir.

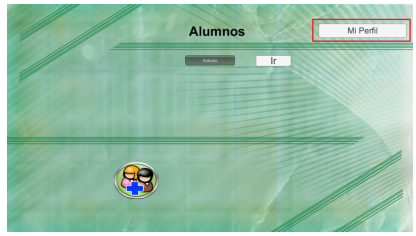
(d) Selección de alumnos.

(e) Selección de alumnos (*combobox* desplegado).



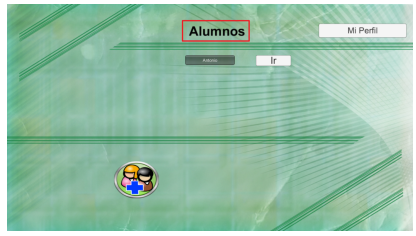
(f) Gestión de alumnos.

(g) Gestión de alumnos pulsado.



(h) Botón Mi perfil.

(i) Botón Mi perfil pulsado.



(j) Título de la pantalla.

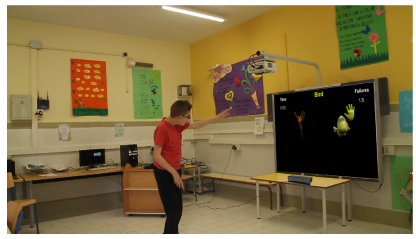
Figura 5.3: Interfaz de la parte de gestión de alumnos.

- Sugerencia sobre el tamaño de los elementos 3D para los alumnos con discapacidad visual.
- El uso de pictogramas para estudiantes con autismo con el objetivo de que puedan comprender mejor la tarea.
- La creación de la actividad de lateralidad.
- En la actividad de asociación de conceptos es necesario que se le dibuje un borde a la opción seleccionada.
- El sistema no debería reproducir música durante la tarea para evitar distraer al estudiante.
- El objetivo de la actividad tiene que estar resaltado para que la atención de los estudiantes no se distraiga con el fondo.

5.3.2. Evaluación de usuarios

En esta evaluación participaron doce alumnos del centro de educación especial Princesa Sofía. Estos estudiantes tenían diferentes tipos de discapacidad: tres estudiantes con discapacidad física, tres estudiantes con discapacidad auditiva, tres estudiantes con discapacidad visual y tres estudiantes con autismo. La evaluación consistió en dos iteraciones. En la Figura 5.4, se muestran los experimentos realizados en este proceso de evaluación.

(a) Experimento con estudiante con discapacidad auditiva.



(b) Experimento con estudiante con discapacidad visual.

(c) Experimento con estudiante con discapacidad física.

(d) Experimento con estudiante con autismo.

Figura 5.4: Experimentos con los diferentes casos.

En la primera iteración, se utilizó la misma metodología para cada estudiante que participó en la evaluación. El proceso consistió en tres sesiones, cada una realizada en un día diferente. La prueba se realizó en una habitación donde los estudiantes estaban acostumbrados a realizar diferentes tipos de actividades porque esta sala contiene una pizarra interactiva, el proyector, recursos educativos como hojas, libros y un par de ordenadores. Además, esta sala es el lugar ideal para usar Kinect porque el ordenador está conectado a la pizarra que es más grande que un monitor convencional y la habitación es muy espaciosa. Por lo tanto, los alumnos pueden moverse libremente y estar lo más lejos posible de la pantalla como se recomienda para una experiencia óptima con este dispositivo. Los alumnos encontraban individualmente en la sala con su tutor, salvo en el caso de que varios alumnos tuvieran el mismo tutor.

En este caso, todos los alumnos del mismo tutor estaban en la sala, pero el prototipo solo fue probado por uno de los estudiantes mientras que los demás esperaban. La primera sesión duró más que otras sesiones con el fin de crear el modelo de dispositivo-interacción para cada participante. Se realizó la misma actividad en cada sesión. No había límite de tiempo para completar la actividad pero la actividad se repitió diez veces. La recopilación de datos era automática porque el sistema estaba conectado a una base de datos y cada alumno tiene un perfil en el sistema. Los datos recopilados en esta iteración fueron el tiempo y el número de errores.

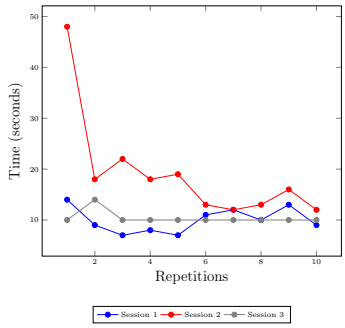
Los siguientes gráficos muestran los datos del usuario, agrupados según los diferentes tipos de discapacidad. La Figura 5.5 muestra los datos relacionados con los usuarios con autismo y la Figura 5.6 muestra los datos relacionados con usuarios con discapacidad auditiva. Los datos correspondientes a los usuarios con discapacidad física se muestran en la Figura 5.7 y para usuarios con discapacidad visual, los datos se pueden ver en la Figura 5.8.

En la primera iteración, una reducción en el tiempo y el número de errores cometidos por los usuarios se pueden ver en los diferentes grupos de usuarios que han participado en el experimento. Estos resultados son alentadores porque la reducción de los errores significa que la actividad se llevó a cabo concienzudamente y la mejora en el tiempo no es una consecuencia de elecciones aleatorias hechas para terminar el ejercicio antes.

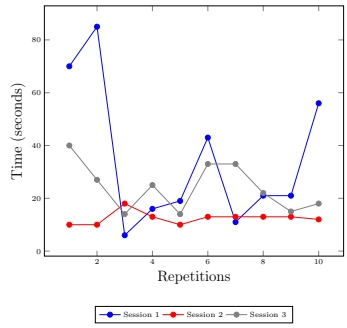
Los usuarios se sintieron cómodos y más seguros con respecto a la interacción cuanto más participaron en las sesiones. Un factor importante en esta iteración fue que los usuarios de los diferentes grupos pudieran completar la actividad ya que cada usuario tenía características muy diferentes, como por ejemplo, el uso de silla de ruedas. A pesar de los resultados prometedores durante estas sesiones, es necesario destacar:

(a) En las pruebas realizadas con los alumnos con discapacidad auditiva, se observa que aunque se muestran las instrucciones de texto, hay casos en los que el tutor tiene que indicar cómo hacer el ejercicio. Por lo tanto, se recomienda que estas instrucciones se muestren de forma gráfica, incluso con una ilustración.

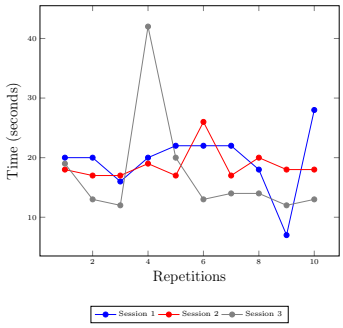
(b) En el caso de que el usuario tenga una discapacidad física, vale la pena enfocarse sobre la irregularidad del segundo participante (ver Figura 5.7b) en el gráfico. Esta situación se debe a que el estudiante pierde interés rápidamente y como consecuencia, algunas repeticiones se completaron en un tiempo razonable mientras otras repeticiones, el tiempo aumentó considerablemente.



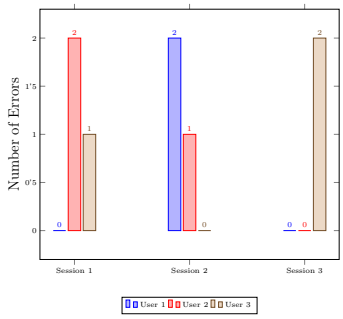
(a) Gráfico de tiempos del participante 1.



(b) Gráfico de tiempos del participante 2.



(c) Gráfico de tiempos del participante 3.

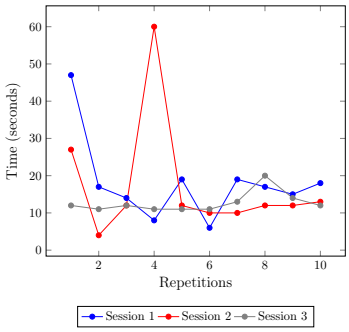


(d) Gráfica de errores.

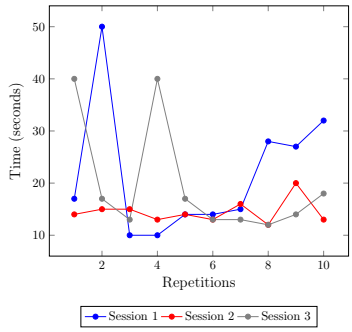
Figura 5.5: Participantes con autismo. Resultados en la primera iteración.

(c) A los estudiantes con autismo no les resultó particularmente difícil comprender y adaptarse al ejercicio como se puede ver en la Figura 5.5, donde los participantes 1 y 2 completaron la última sesión sin errores.

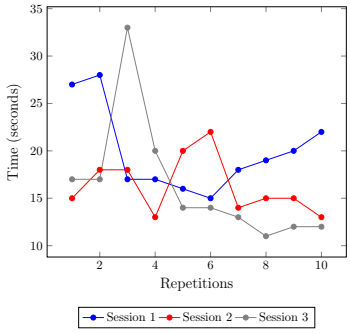
En la segunda iteración, la sala de evaluación se cambió y las pruebas se realizaron en una sala específica que fue designada por el director del centro para realizar las actividades con Kinect. Esta decisión se tomó porque en los gráficos de tiempo de la iteración anterior hay una intervalo muy grande entre los valores en la misma sesión de cada grupo. Estos intervalos fueron causados por diferentes elementos (juguetes, carteles, etc.) que estaban en la sala y distraían a los alumnos. Por consiguiente, esta nueva habitación fue pintada de blanco y solo contenía el equipo esencial para realizar los ejercicios con este dispositivo: un ordenador de sobremesa con unos altavoces, un monitor de 42 pulgadas y el dispositivo Kinect. Al igual que en la iteración anterior, todos los estudiantes estaban en la sala con sus tutores. La única diferencia fue que los estudiantes tuvieron que esperar fuera su turno. Así, todas las sesiones se llevaron a cabo de forma individual. A diferencia de la iteración anterior, se realizaron dos actividades en cada sesión: la actividad de asociación de conceptos (cambio de tema entre animales y transporte) y



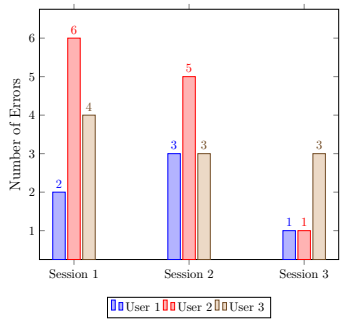
(a) Gráfico de tiempos del participante 1.



(b) Gráfico de tiempos del participante 2.



(c) Gráfico de tiempos del participante 3.



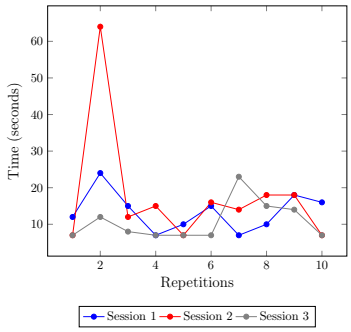
(d) Gráfica de errores.

Figura 5.6: Participantes con discapacidad auditiva. Resultados en la primera iteración.

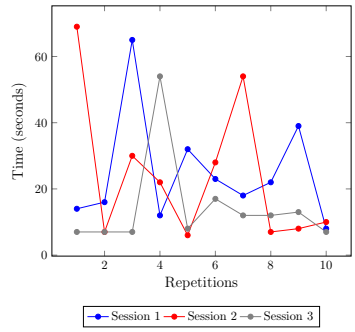
la actividad de lateralidad.

En las Figuras 5.9, 5.10, 5.11 and 5.12 se muestran los datos obtenidos de este segundo experimento con los estudiantes.

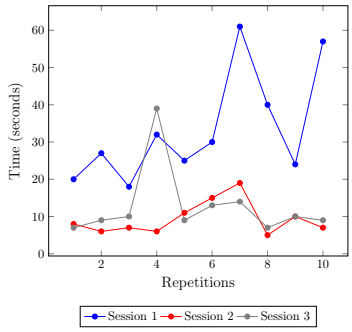
En esta segunda iteración, es importante comparar los resultados con los de la primera iteración con el fin de comprobar el progreso de los estudiantes y ver si el modelo dispositivo-interacción está funcionando. La característica más importante en esta segunda fase es que el tiempo y el número de errores han disminuido con respecto a la primera iteración. Los gráficos relacionados con el rendimiento del tiempo (ver Figuras 5.9, 5.10, 5.11 y 5.12) muestran la reducción en el tiempo. Con respecto a la disminución del número de errores en esta segunda iteración (ver Figura 5.9d, 5.10d, 5.11d y 5.12d) se deduce que los usuarios han entendido el propósito de las tareas, y la interacción con el sistema ya no era un obstáculo para lograr los objetivos. Es necesario decir que ninguno de los usuarios tenía experiencia previa con el dispositivo de interacción y esto explica por qué los resultados fueron muy lentos e irregulares al comienzo de la primera sesión en la iteración anterior. Esto no es evidente en la presente iteración. Los valores más destacados son los mostrados a continuación:



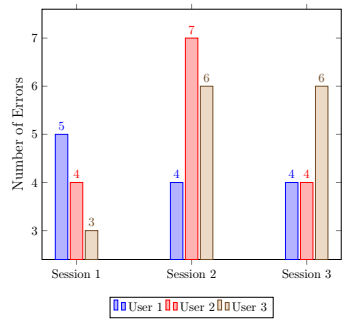
(a) Gráfico de tiempos del participante 1.



(b) Gráfico de tiempos del participante 2.



(c) Gráfico de tiempos del participante 3.



(d) Gráfica de errores.

Figura 5.7: Participantes con discapacidad física. Resultados en la primera iteración.

- Los usuarios con discapacidad auditiva disminuyeron el tiempo y el número de errores y los resultados fueron uniformes. Para esta evaluación, se agregaron imágenes en lugar de texto y las instrucciones de los tutores no fueron necesarias.

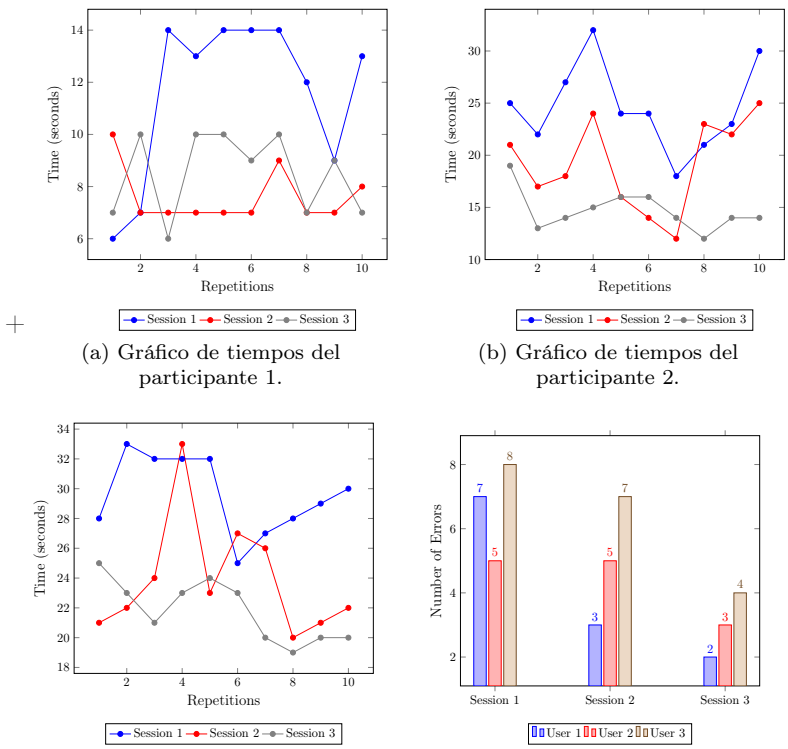
- Se observó que los estudiantes con discapacidad física parecían más indecisos cuando interactuaban con el sistema. Además, sus movimientos fueron más limitados y, como resultado, su tiempo de ejecución fue más lento que el resto de participantes. Esto demostró que, aunque la distancia entre los elementos se ha reducido para facilitar la interacción, tendría que reducirse aún más, especialmente en los casos en que el usuario utiliza una silla de ruedas.

- En la evaluación con los estudiantes con autismo se puede concluir que les resulta más fácil interactuar con el sistema que el resto de los participantes. El motivo principal es que estos alumnos no tienen ninguna limitación en sus movimientos o en cualquier aspecto sensorial. El único inconveniente es que pueden perder la atención y el interés más fácilmente. Por eso la metodología de las actividades en estos estudiantes es tan diferente con respecto a otros casos.

Después de explicar los resultados individuales según las características de los participantes, los datos se agrupan para tener una visión general de la evaluación de los usuarios. Para hacer esto, los tiempos medios de los estudiantes con autismo, discapacidad física, discapacidad auditiva y dicapacidad visual se calcularon, y se obtuvieron los siguientes gráficos con los datos respecto a cada una de las iteraciones (ver Figura 5.13). Además, se calculó el promedio de los errores en ambas iteraciones (ver Figura 5.14).

La Figura 5.13 (a) muestra los resultados de los tiempos de ejecución de los estudiantes, independientemente de sus características, donde se puede apreciar que los tiempos se reducen en las sesiones consecutivas, siendo la última sesión la que presenta los valores más bajos. Por tanto, desde un punto de vista general, los participantes se han acostumbrado a interactuar con el sistema y han podido completar el ejercicios en un menor tiempo.

Por otro lado, si miramos las líneas del gráfico en la Figura 5.13 (b), se pueden obtener las mismas conclusiones que en el gráfico anterior ya que el tiempo de ejecución disminuye a medida que avanzan las sesiones. Sin embargo, el aspecto significativo de este gráfico radica en el eje y porque la escala es más pequeña que en el gráfico anterior, el



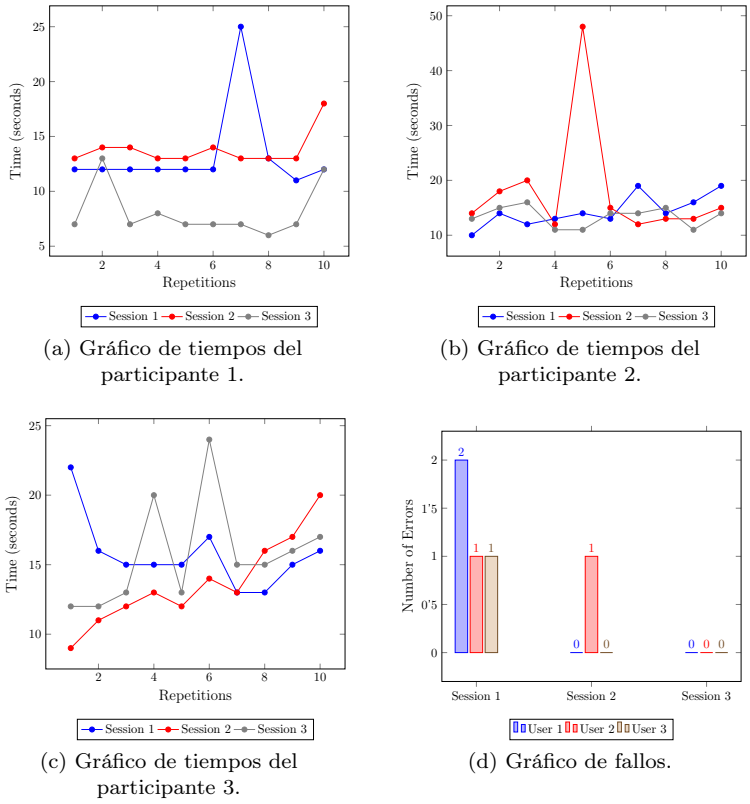
(a) Gráfico de tiempos del participante 1.

(b) Gráfico de tiempos del participante 2.

(c) Gráfico de tiempos del participante 3.

(d) Gráfica de errores.

Figura 5.8: Participantes con discapacidad visual. Resultados en la primera iteración.



(a) Gráfico de tiempos del participante 1.

(b) Gráfico de tiempos del participante 2.

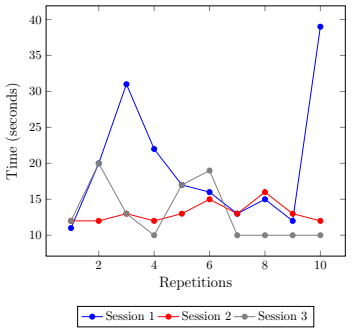
(c) Gráfico de tiempos del participante 3.

(d) Gráfico de fallos.

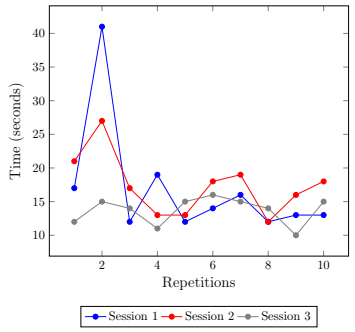
Figura 5.9: Participantes con autismo. Resultados en la segunda iteración.

cual muestra que los participantes están haciendo los ejercicios en menos tiempo. A pesar de que en la primera iteración el uso de Kinect junto con un nuevo modo de interacción podría representar un obstáculo mayor, los estudiantes con el tiempo pueden interactuar con el sistema y sentirse cómodos con la ayuda del modelo dispositivo-interacción. De hecho, se ajusta a las necesidades de cada individuo para facilitar la interacción con las diferentes actividades propuestas. Además, la tabla de errores (ver Figura 5.14) muestra que el número de fallos es menor en la segunda iteración que en la primera. Esta situación es similar a los resultados donde los datos están separados por discapacidades.

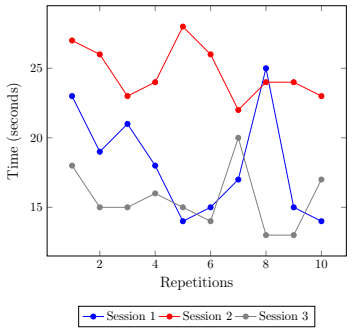
Finalmente, la Tabla 5.3 muestra el promedio, la desviación estándar y el coeficiente de variación de los tiempos obtenidos de las diferentes sesiones que se han organizado en las dos iteraciones. El promedio por sí solo no es un factor del todo relevante ya que puede haber una divergencia en los valores, lo que significaría que algunos de los parámetros de la actividad como el modo interactivo o que el alumno no entiende las instrucciones. Estas circunstancias harían que los estudiantes perdieran confianza y se sintieran incómodos con las actividades y, en consecuencia, provocar estas drásticas variaciones en el tiempo. Por consiguiente, el coeficiente de variación se calculó y conociendo la homogeneidad



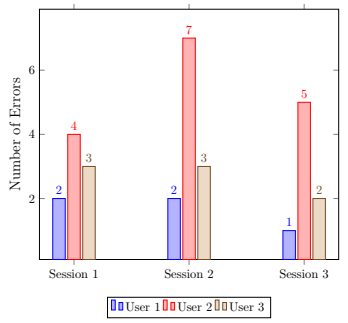
(a) Gráfico de tiempos del participante 1.



(b) Gráfico de tiempos del participante 2.



(c) Gráfico de tiempos del participante 3.



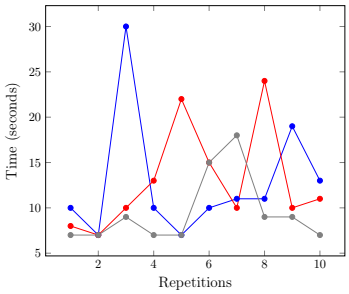
(d) Gráfico de fallos.

Figura 5.10: Participantes con discapacidad auditiva. Resultados en la segunda iteración.

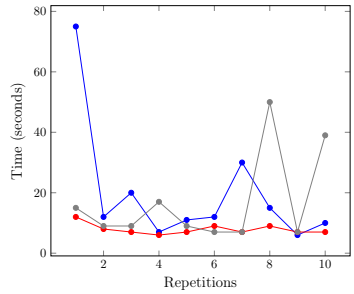
de los valores, cuanto más cerca estuvieran estos valores a cero, mayor homogeneidad habrá. Este coeficiente está más cerca a cero en la segunda iteración, por lo tanto, podemos asumir que los usuarios entienden la dinámica de la actividad y son capaces de realizarla con el modelo propuesto. Además, esto sirve para apoyar la decisión del cambio de habitación para realizar las actividades debido a las distracciones en la ubicación anterior puesto que el coeficiente de variación es menor en la segunda iteración y los valores son más lineales con una minoría de valores altos respecto a la primera sesión.

5.3.3. Cuestionario de experiencia de usuario

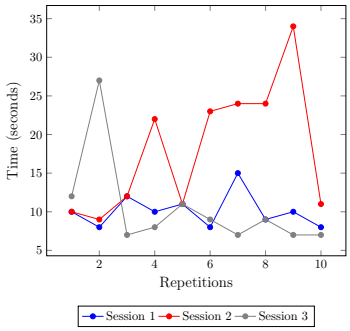
Por último, se realizó una valoración cualitativa con los cinco tutores de los estudiantes que participaron en la evaluación anterior. Los tutores tuvieron que llenar el Cuestionario de Experiencia de Usuario, del inglés User Experience Questionnaire (UEQ) [Laugwitz et al., 2008] de acuerdo con su experiencia con el sistema. El UEQ es un método eficaz para recopilar las opiniones de los usuarios con respecto a su experiencia con un producto



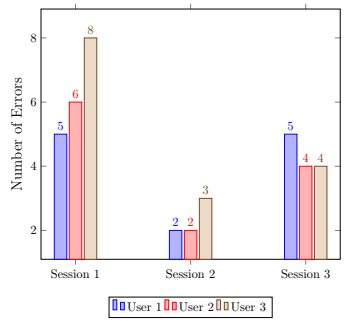
(a) Gráfico de tiempos del participante 1.



(b) Gráfico de tiempos del participante 2.

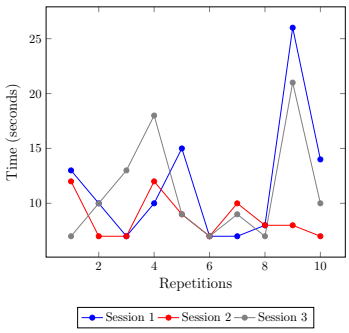


(c) Gráfico de tiempos del participante 3.

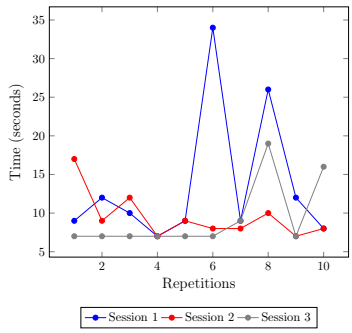


(d) Gráfico de fallos.

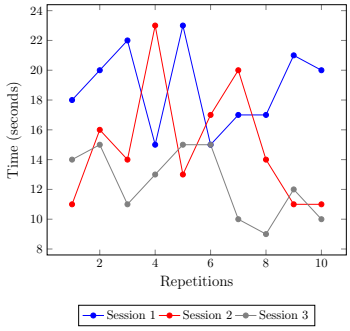
Figura 5.11: Participantes con discapacidad física. Resultados en la segunda iteración.



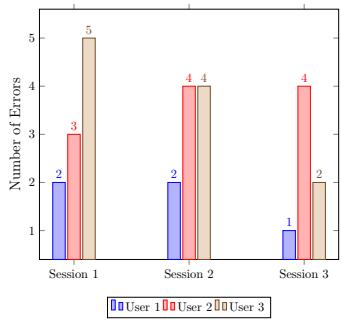
(a) Gráfico de tiempos del participante 1.



(b) Gráfico de tiempos del participante 2.

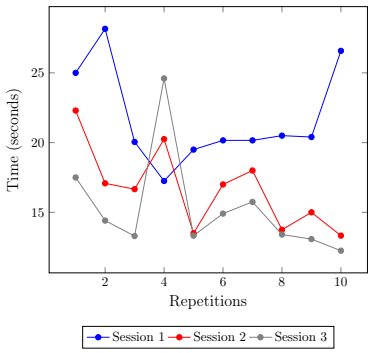


(c) Gráfico de tiempos del participante 3.

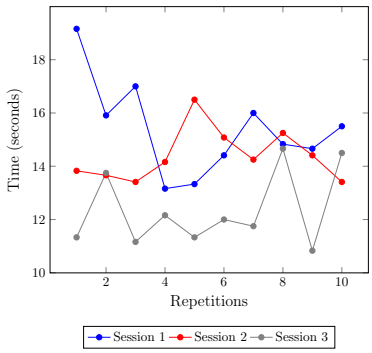


(d) Gráfico de fallos.

Figura 5.12: Participantes con discapacidad visual. Resultados en la segunda iteración.



(a) Resultados de los tiempos de la primera iteración.



(b) Resultados de los tiempos de la segunda iteración.

Figura 5.13: Resultados de los tiempos generales de la primera y la segunda iteración.

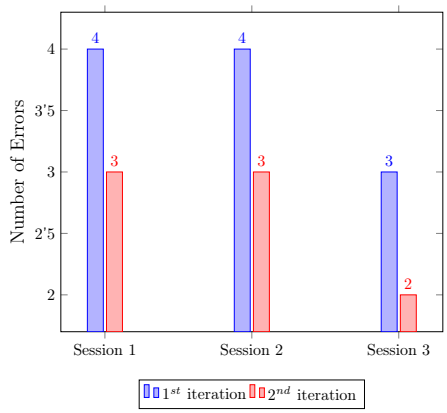


Figura 5.14: Resultados de los errores generales.

Discapacidad	Usuario	#iterac.	Promedio	DE	CV
Autismo	1	1	13,16	7,49	0,56
		2	11,73	3,83	0,32
	2	1	25,7	18,51	0,72
		2	15,26	6,64	0,43
	3	1	18,46	6,19	0,33
		2	14,7	2,79	0,19
Discapacidad auditiva	1	1	15,96	11,26	0,70
		2	15,26	6,36	0,41
	2	1	18,63	9,86	0,52
		2	16	5,84	0,36
	3	1	17,5	5,02	0,28
		2	14,7	4,75	0,19
Discapacidad física	1	1	13,96	10,68	0,76
		2	11,76	5,69	0,48
	2	1	21,13	18,03	0,85
		2	14,86	15,18	1
	3	1	18,5	14,82	0,80
		2	12,83	6,34	0,49
Discapacidad visual	1	1	9,23	2,67	0,28
		2	10,5	4,56	0,43
	2	1	19,5	5,44	0,27
		2	10,8	6,16	0,57
	3	1	25,1	4,40	0,17
		2	15,4	4,04	0,26

Tabla 5.3: Parámetros estadísticos sobre el tiempo de todos los participantes (DE: Desviación Estándar; CV: Coeficiente de Variación).

[Laugwitz et al., 2008]. El objetivo principal de la UEQ es ofrecer una forma rápida de medir la experiencia del usuario en un producto. Además, La fiabilidad y validez del UEQ ha sido probada con 144 participantes y una encuesta en línea con 722 participantes, que proporciona una herramienta de evaluación de calidad [Schrepp et al., 2014]. Este cuestionario contiene 26 elementos (ver Apéndice B Figura B.1) que se dividen en 6 escalas que se presentan a continuación:

- Atractivo: identifica si al usuario le gusta o no le gusta el producto.
- Perspicuidad: verifica si es fácil aprender a usar el producto.
- Eficacia: muestra si el usuario resuelve las tareas sin mucho esfuerzo.
- Fiabilidad: valida si el usuario se siente cómodo con la interacción.
- Estímulo: identifica si el usuario se siente motivado mientras usa el producto.
- Innovación: manifiesta si el producto es innovador.

Aunque existía una versión corta del UEQ que contenía solo 8 elementos [Schrepp et al., 2017b], se decidió utilizar la versión estándar ya que todos los elementos encajaban con el sistema y podía ayudar a mejorar los resultados del estudio. Las principales ventajas de aplicar este método son:

(a) Tiene en cuenta estos tres criterios en relación con la experiencia de usuario:
   — Sus sentimientos acerca de la interacción con el producto según el estándar ISO 9241-10 [DIN, 1996].
   — La eficacia o rendimiento de acuerdo el ISO 9241-11 [of human-system interaction (Subcommittee), 1998].
   — La satisfación del usuario relacionado con la cualidad hedonista [of human-system interaction (Subcommittee), 1998].

(b) Es sencillo y rápido.

(c) Un benchmark fue creado para mejorar la precisión.

(d) Ha sido traducido a muchos idiomas y de este modo facilitando su uso y haciendo los resultados más fiables.

Los cinco tutores contestaron al cuestionario UEQ (ver Tabla 5.4) de forma anónima según la experiencia que habían tenido en el experimento anterior con los estudiantes y de usar el sistema ellos mismos. Luego, se analizaron los resultados con las herramientas de análisis de datos que los autores proporcionan para saber la puntuación del sistema en términos de experiencia de usuario. Los resultados se muestran en la Tabla 5.5 y Figura 5.15.

Los autores del UEQ establecieron un punto de referencia para este cuestionario porque pensaron que sería útil, especialmente cuando se evalúa el producto por primera

Elemento	#P1	#P2	#P3	#P4	#P5
1	6	7	6	5	6
2	5	5	7	5	6
3	3	5	3	3	1
4	5	3	1	2	1
5	2	4	3	2	2
6	4	6	5	5	5
7	6	5	6	5	6
8	7	6	6	7	7
9	3	5	4	2	3
10	1	2	1	3	1
11	6	7	6	5	6
12	1	1	1	2	1
13	5	6	6	6	7
14	6	6	7	5	6
15	7	6	6	5	7
16	4	5	6	6	6
17	2	2	3	1	1
18	3	4	2	2	1
19	2	2	3	2	2
20	5	6	5	5	6
21	3	2	2	6	1
22	7	6	6	6	6
23	2	1	2	1	1
24	1	1	1	2	1
25	1	2	2	2	2
26	7	6	6	5	6

Tabla 5.4: Respuestas del cuestionario de experiencia de usuario. (Participantes #1 a #5).

Escala	Media	Varianza
Atractivo	2.200	0.10
Perspicuidad	1.600	1.02
Eficacia	1.700	0.11
Fiabilidad	2.150	0.14
Estímulo	1.400	0.21
Innovación	1.900	0.58

Tabla 5.5: Escalas para el UEQ.

vez, donde no es posible hacer comparaciones con evaluaciones previas [Schrepp et al., 2017a]. Por lo tanto, lo usamos para tener una línea de base fiable para nuestro análisis. Según el índice de referencia (ver Tabla 5.6), los resultados que obtuvimos (ver Tabla 5.5 y Figura 5.15) del cuestionario sería: Atractivo Excelente, Perspicuidad Buena, Eficiencia Buena, Fiabilidad Excelente, Estímulo Bueno e Innovación Excelente.

Como se puede observar, las escalas de atractivo, fiabilidad e innovación obtuvieron los mejores resultados en la prueba. Sin embargo, a pesar de que la perspicuidad, la eficiencia y el estímulo obtuvieron resultados más bajos, estos todavía eran lo suficientemente buenos. De este hecho, se deduce que los participantes calificaron la novedad

como muy alta debido al uso de Kinect para interactuar mediante gestos o detección de movimiento con la aplicación, sobre todo porque no hay otra actividad similar en el centro. En cuanto a la puntuación de atractivo, a los participantes les gustó la aplicación, por lo que parece que la evaluación con expertos fue útil, ya que después de esto cambiamos algunos aspectos en la interfaz y el diseño. Para este estudio, el factor más importante fue la fiabilidad ya que está relacionado con la interacción y nuestro principal objetivo en este estudio es demostrar que el modelo de dispositivo-interacción que hemos diseñado es útil y práctico. Por tanto, obtener una calificación tan alta en esa escala es una excelente indicación, ya que está claro que los usuarios pensaron que el sistema fue adecuadamente interactivo. Por otro lado, la perspicuidad tiene una puntuación más baja porque los estudiantes no están acostumbrados a interactuar con Kinect y tuvieron que hacer un proceso de formación antes de los experimentos para acostumbrarse. La puntuación más baja correspondía a la escala de estimulación a pesar de que diseñamos las actividades como juegos para involucrar a los estudiantes y garantizar que no se aburrieran al completar las tareas. Debemos comprobar el diseño de las actividades para motivar a los estudiantes y quizás agregar más actividades para ampliar la variedad.

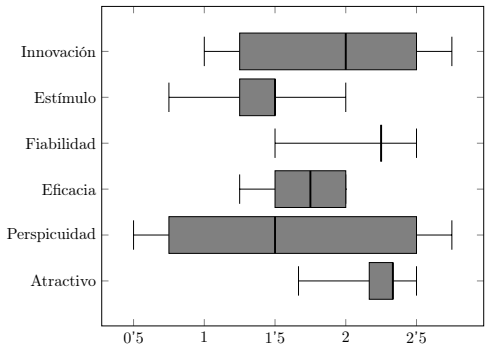


Figura 5.15: Gráficas de escalas UEQ.

Categoría	**A**	**P**	**E**	**F**	**ES**	**I**
Excelente	≥ 1.75	≥ 1.78	≥ 1.9	≥ 1.65	≥ 1.55	≥ 1.4
Bueno	≥ 1.52	≥ 1.47	≥ 1.56	≥ 1.48	≥ 1.31	≥ 1.05
	< 1.75	< 1.78	< 1.9	< 1.65	< 1.55	< 1.4
Encima de la media	≥ 1.17	≥ 0.98	≥ 1.08	≥ 1.14	≥ 0.99	≥ 0.71
	< 1.52	< 1.47	< 1.56	< 1.48	< 1.31	< 1.05
Debajo de la media	≥ 0.7	≥ 0.54	≥ 0.64	≥ 0.78	≥ 0.5	≥ 0.3
	< 1.17	< 0.98	< 1.08	< 1.14	< 0.99	< 0.71
Malo	< 0.7	< 0.54	< 0.64	< 0.78	< 0.5	< 0.3

Tabla 5.6: Límites del Benchmark para UEQ [Schrepp et al., 2017a] (A: Atractivo, P: Perspicuidad, E: Eficacia, F: Fiabilidad, ES: Estímulo, I: Innovación).

5.4. Resumen

En este capítulo se presenta la siguiente fase del trabajo, la cual se centra en la adaptación de la interacción para el usuario. Después de finalizar el estudio del capítulo anterior y comprobar que el sensor Kinect podría ser una herramienta realmente útil en términos de interacción gracias a su sensor de profundidad, se decidió continuar con el uso de dicho sensor pero adquiriendo la siguiente versión del dispositivo: Microsoft Kinect v2.

En la anterior fase del trabajo se evaluó el sistema con estudiantes con diferentes tipos de discapacidad, lo que provocó que la interacción fuera más adecuada en términos de usabilidad para ciertos estudiantes. Sin embargo, no sería un objetivo viable ni realista hacer una adaptación para cada uno de los usuarios que utilizaran el sistema pero sí lo sería, hacer una adaptación para un número mayor de usuarios. Por este motivo, se propuso la creación de un modelo enfocado en la adaptación de la interacción del usuario.

Este modelo se denominó *modelo dispositivo-interacción*, cuyo objetivo era adaptar la interacción entre el dispositivo y el usuario. Al igual que en el capítulo anterior se presenta la arquitectura del sistema y sus elementos que consisten en:

- **Subsistema de interacción**: Este subsistema activará el medio de interacción más adecuado dependiendo de lo que le indique el módulo de adaptación.
- **Modelo de usuario**: Es un modelo de usuario basado en características, el cual contiene las siguientes características del usuario: nombre, edad, sexo, problemas de lateralidad y discapacidad.
- **Modelo dispositivo-interacción**: El objetivo de este modelo es optimizar la interacción del usuario teniendo en cuenta las características del dispositivo.
- **Módulo de adaptación**: Este módulo hace uso del modelo de usuario, el modelo dispositivo-interacción y las reglas asociadas para enviar información a los subsistemas de interacción y actividades y que estos actúen en consecuencia.
- **Subsistema de actividades**: Este componente tiene la responsabilidad de definir cada una de las actividades que forman el sistema en base a la información que recibe de los otros elementos de la arquitectura.

A continuación se describen las actividades propuestas diseñadas también con la colaboración de los profesores del centro de eduación especial Princesa Sofía, donde se han desarrollado una actividad con el fin de que los estudiantes asocien conceptos respecto a una unidad didáctica. Por otro lado, se diseñó una actividad que tenía el objetivo de trabajar la lateralidad izquierda y derecha. Para estas actividades se han modificado determinados aspectos, como es el medio de interacción, la forma de realizar la actividad o diferentes propiedades de los elementos, dependiendo de las características de los usuario que está interactuando con el sistema y del *modelo dispositivo-interacción*.

Por último, se realizaron dos tipos de evaluación: Una evaluación con expertos y una evaluación con usuarios finales. En la evaluación con expertos se aplicó el método de inspección con la combinación del recorrido cognitivo y la técnica de pensar en voz alta. En la evaluación con usuarios finales se realizó una evaluación cuantitativa y otra cualitativa. En la evaluación de orden cuantitativo participaron estudiantes con discapacidad

física, auditiva, visual y autismo que utilizaron el sistema. Esta evaluación consistió en dos iteraciones donde los estudiantes realizaban las actividades un número de repeticiones determinado y no había límite de tiempo para completarlas. Además, se realizó una evaluación cualitativa, en la cual participaron los tutores de los estudiantes para agrupar más información sobre la experiencia de usuario. Esta evaluación consistió en rellenar un cuestionario denominado *User Experience Questionnaire* que tiene el objetivo de medir cómo ha sido la experiencia de un usuario respecto a un producto. En esta metodología se miden seis escalas diferentes: atractivo, perspicuidad, eficacia, fiabilidad, estímulo e innovación. La escala de fiabilidad era la que resultaba más interesante para este estudio porque estaba relacionada con la interacción, y donde se obtuvo la máxima puntuación.

En la evaluación de tipo cuantitativo se observó que el tiempo de ejecución y el número de errores disminuyeron, lo que demuestra que todos los usuarios, a pesar de sus diversas características, lograron completar las actividades propuestas sin ningún problema. Además, se llevó a cabo una evaluación cualitativa y los resultados mostraron que a los usuarios les gustó la aplicación, que no tuvieron muchos problemas con la interacción a pesar de utilizar Kinect y que pensaron que el sistema era original.

Capítulo 6

Sistema de reconocimiento de gestos con webcam

Capítulo 6

SISTEMA DE RECONOCIMIENTO DE GESTOS CON WEBCAM

Contenidos

Esta última fase del proyecto se vio motivada por la incertidumbre de la continuidad de Kinect por parte de Microsoft, la falta de distribución de los modelos v1 y v2 utilizados en este trabajo y el hecho de que aunque estos dispositivos no eran excesivamente caros, el usuario tenía que realizar una inversión solo para hacer uso de los sistemas descritos en los apartados anteriores. Por lo tanto, se decidió que en esta fase del proyecto se iba a prescindir de este dispositivo y realizar una investigación con una serie de vídeos e imágenes que permitiera el uso de una webcam estándar (ver Tabla 6.1) para el reconocimiento de gestos con las manos mediante el uso de métodos de Inteligencia Artificial.

Tipos	**Ventajas**	**Desventajas**
Kinect	Tiene sensor de profundidad, cámara RGB y array de micrófonos. Permite realizar reconocimiento de gestos y de voz. Reconoce hasta 25 joints del cuerpo humano. Reconocimiento del usuario hata una distancia de 4,5m. Reconocimiento de múltiples personas. Permite trabajar en el espectro infrarrojo. Extracción de modelos 3D.	Soportado solo en Windows [Cai et al., 2017]. Es necesario un intermediario para extraer los flujos de datos.
Webcam	Soporta Windows, Linux y Macintosh. Está integrado en la mayoría de portátiles y dispositivos móviles. Las librerías de visión artificial pueden extraer los datos directamente.	Se ve afectado por la luminosidad. El usuario tiene que estar a una distancia próxima al sensor.

Tabla 6.1: Tabla comparativa de los dispositivos utilizados en esta Tesis doctoral.

6.1. Reconocimiento de gestos con Deep Learning y Fuzzy Logic

En esta etapa se dispuso la utilización de ciertas técnicas de DL porque esta disciplina no requiere de características hechas a mano. En cambio, se basan en estructuras predefinidas y datos etiquetados y pueden aprender características por sí mismas [Bonanno et al., 2017]. La clasificación de los gestos procedentes de la entrada de datos del sistema se realizó mediante una serie de pruebas con distintos modelos que tenían diversas configuraciones, las cuales dieron lugar a la ejecución de 104 experimentos. Dichos experimentos fueron la base para poder realizar una clasificación dependiendo de las diferentes características de estos, mediante la integración de un sistema experto basado en Lógica Difusa. La contribución de esta fase del proyecto consiste en el hecho de que esta metodología de clasificación va a ser útil para aquellos desarrolladores, investigado-

res o entusiastas que quieran realizar tareas basadas en DL y reconocimiento de gestos porque les va a ahorrar mucho tiempo y esfuerzo. Cuando se va a realizar cualquier tarea en DL es normal que surjan dudas sobre cuántas capas debería tener la red neuronal [Brownlee, 2018] o qué función de coste es mejor para un determinado problema. La cuestión es que no hay establecidas una serie de reglas o métodos matemáticos que determinen con exactitud cómo hay que configurar una red neuronal para abordar un problema en concreto ya que para cada problema lo más probable es que la configuración necesaria sea totalmente diferente. Cuando se quiere realizar una tarea específica como el reconocimiento de gestos utilizando DL lo que se tiene que hacer es probar diferentes configuraciones y comparar los resultados hasta que se obtenga el resultado esperado. Esta metodología (ver Figura 6.1) puede ayudar a tener un punto de partida para elegir una configuración satisfactoria cuando se quieren reconocer determinados tipos de gestos. En las pruebas previas que se programaron para obtener dicha metodología se entrenaron hasta 10 gestos con distintos parámetros de configuración.

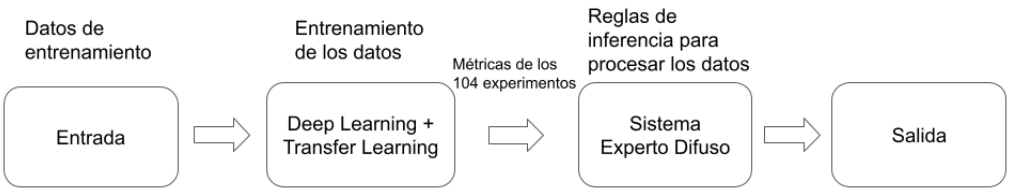


Figura 6.1: Metodología del sistema de *Deep Learning* y *Fuzzy Logic*.

6.1.1. Metodología del sistema

Esta metodología comienza con el módulo denominado *Deep Learning + Transfer Learning*, el cual aplica la transferencia de aprendizaje (Transfer Learning). Esta técnica se basa en usar un modelo obtenido en la solución de un problema para dar solución a otro problema que aunque son distintos tienen características similares. Si trasladamos esta definición a un tema más específico como DL se puede decir que la transferencia de aprendizaje se basa en utilizar una red neuronal convolucional que ha sido pre-entrenada en un conjunto de datos de envergadura y sus pesos para encontrar solución a otro problema. La ventaja principal de esta técnica reside en que se ahorra mucho tiempo en el entrenamiento del modelo puesto que no hay que entrenar dicho modelo desde cero, el cual requiere altos recursos computacionales.

El proceso de transferencia de aprendizaje se puede realizar a través de dos medios: extracción de caracterísicas o fine-tuning [Al Hadhrami et al., 2018].

En la modalidad de extracción de características en lugar de dejar que la imagen recorra cada una de las capas de la red neuronal convolucional se para en una de las capas convolucionales o de pooling. Lo usual es que esta parada se haga antes de que alcance las capas totalmente conectadas (*fully connected*). A continuación, se extraen las características en forma de vector de características para usar éstas en el entrenamiento del modelo de ML que se haya decidido utilizar en la propuesta.

Por otro lado, a diferencia de la extracción de características, *fine-tuning* requiere que se actualice la arquitectura del modelo sustituyendo las capas totalmente conectadas por unas nuevas y entrenando las nuevas capas para predecir las distintas clases. Para este trabajo se ha elegido la técnica de *fine-tuning* porque posee la ventaja principal de que permite optimizar ciertos parámetros del modelo como el optimizador o la función de coste, para obtener los mejores resultados.

En definitiva, el uso de extracción de características no requiere reentrenar la red neuronal convolucional original mientras que con fine-tuning es necesario actualizar la arquitectura de la red neuronal convolucional y además reentrenarla para aprender otras clases nuevas.

En la implementación de esta tarea se ha usado el framework denominado Keras[1] que soporta el lenguaje Python. Keras es un API de código abierto enfocado en DL, que se encuentra entre los frameworks más utilizados en DL y permite crear nuevos experimentos de una manera sencilla y rápida. Este framework desarrollado por Google está basado en Tensorflow 2.0[2] lo que permite aprovechar las características de esta librería y poder ser usada en diferentes entornos como iOS o Android. Su reconocimiento y utilidad para la comunidad científica ha hecho que Keras se haya aplicado en diversas organizaciones de prestigio como la NASA[3]. Por estas razones, se ha elegido Keras para realizar la clasificación de los gestos con las manos mediante transferencia de aprendizaje con *fine-tuning* a partir de la base de datos 20BN-Jester[4]. Esta base de datos contiene vídeos de personas que realizan diferentes tipos de gestos con las manos (ver Anexo **??** Figuras A.14 y A.23), la cual se compone de 148.092 vídeos y 27 etiquetas.

En la implementación de *fine-tuning* se han realizado los siguientes pasos:

(1.) Primero se debe cortar/separar el conjunto final de las capas totalmente conectadas donde se devuelven las predicciones de las clases de etiquetas de una red entrenada previamente.

(2.) A continuación se reemplaza la sección cortada de las capas totalmente conectadas del paso anterior con inicializaciones aleatorias.

(3.) Por último, se entrena la red neuronal convolucional con una tasa de aprendizaje reducida para que las nuevas capas puedan aprender de las capas convolucionales previas de la red neuronal convolucional.

Keras dispone de una serie de modelos pre-entrenados con la base de datos de imágenes ImageNet[5] para que puedan ser utilizados en *fine-tuning*[6].

A continuación, se van a describir en detalle los algoritmos que han proporcionado los mejores resultados, para después exponer las características más relevantes del resto de modelos. Además, se han elaborado las Tablas 6.2 y 6.3 que contienen las ventajas y desventajas de cada uno de ellos.

[1]Keras - `https://keras.io/`
[2]Tensorflow 2.0 - `https://www.tensorflow.org/guide/effective_tf2`
[3]NASA - `https://www.nasa.gov/`
[4]20BN-Jester - `https://20bn.com/datasets/jester`
[5]ImageNet - `http://www.image-net.org/`
[6]Modelos Keras - `https://keras.io/api/applications/`

6.1.1.1. Modelos

En este apartado se va a describir, en primera instancia, los mejores modelos candidatos que se han probado. Estos han sido las redes residuales (ResNet) y las redes convolucionales muy profundas (VGG). Por último, se va a describir brevemente el resto de modelos que han sido aplicados en esta etapa.

Redes residuales

Las redes residuales se caracterizan porque todas tienen estructuras similares, pero se diferencian en el nivel de profundidad que alcanzan [Nguyen et al., 2018] (ver Figura 6.2). Estas redes se crearon a raíz del problema que existe cuando las redes neuronales denominadas planas (como por ejemplo Alexnet) crean demasiadas capas de profundidad y se genera el problema de los gradientes de desaparición/explosión (*vanishing/exploding*). Este problema se produce durante la propagación hacia atrás (*backpropagation*), cuando se realiza la derivada parcial de la función de coste con respecto al peso actual en cada iteración del entrenamiento, puesto que tiene el efecto de multiplicar N de estos números (que pueden ser pequeños o grandes) para calcular los gradientes de las capas frontales en una red de N capas. En este caso se pueden dar dos circunstancias, una de ellas sería cuando la red es profunda y se multiplica N de estos pequeños números lo que el gradiente será cero (desaparecerá) y por otro lado, cuando se multiplica por n números grandes el gradiente será demasiado grande (explotará) y de ahí reside el nombre de este problema.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figura 6.2: Diferentes configuraciones a nivel de arquitectura para las redes residuales[7]

Para evitar esta situación las redes residuales tienen un mecanismo denominado *conexión de salto/acceso directo* (ver Figura 6.3), el cual agrega la entrada x a la salida después de algunas capas de peso y por lo tanto, la salida tiene la forma H(x) = F(x) + x [He et al., 2016]. Las capas de peso en realidad es aprender una especie de mapeo

[7]Fuente: https://supervise.ly/explore/models/res-net-152-image-net-4228/overview

residual: F(x) = H(x) -x. Además, no hay que temer si hay un gradiente de desaparición para las capas de peso, puesto que la identidad x está ahí para transferir de nuevo a las capas anteriores.

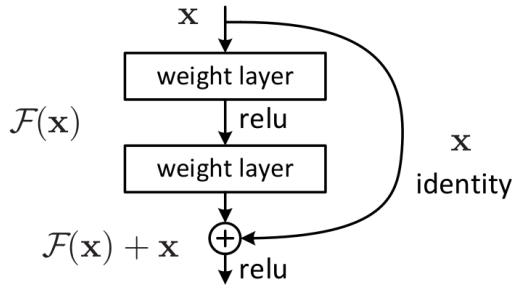


Figura 6.3: Conexión de salto/acceso directo de la redes residuales. Fuente: [He et al., 2016].

Redes convolucionales muy profundas

Estas redes son muy simples ya que se forman utilizando solo capas convolucionales de 3x3 que se van apilando para ir aumentando su profundidad, donde el *max pooling* tiene la función de reducir el tamaño del volumen. Luego, dos capas completamente conectadas, cada una con 4096 nodos, son seguidas por un clasificador softmax. Estas redes se denominan comúnmente como VGGXX, donde ese XX representa un número que suele ser 16 ó 19, estos números representan el número de capas de peso en la red. A pesar de esto estas redes cuentan con las desventajas de que los pesos de la arquitectura de red son bastante grandes y que el proceso de entrenamiento es muy lento. Las arquitecturas de las redes VGG16 y VGG19 se pueden observar en las Figuras 6.4 y 6.5.

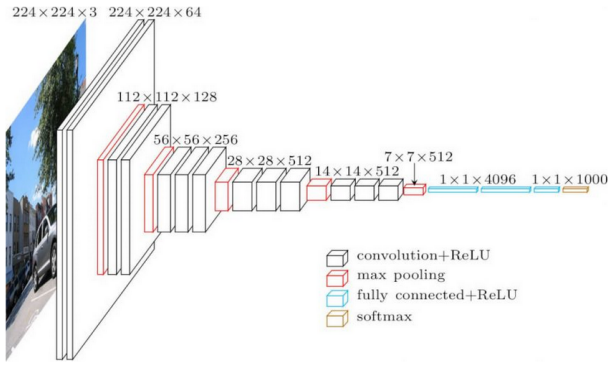


Figura 6.4: Arquitectura del modelo VGG16. Fuente: `https://neurohive.io/en/popular-networks/vgg16/`

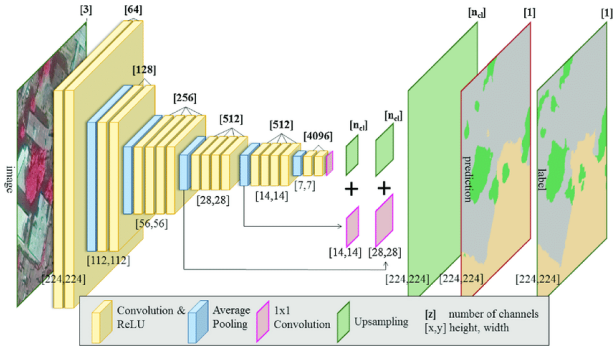


Figura 6.5: Arquitectura del modelo VGG19. Fuente: [Wurm et al., 2019]

6.1.1.2. Optimizadores

En este apartado se van a describir los optimizadores que han sido utilizados para realizar el entrenamiento de los diferentes modelos de este estudio. Estos optimizadores han sido: Adam y Stochastic Gradient Descent (SGD).

Adam

Es un método para la optimización estocástica que se caracteriza porque requiere gradientes de primer orden y la reducida cantidad de memoria necesaria para su ejecución. Este método se encarga de realizar el cálculo de las tasas de aprendizaje adaptativo individuales para diferentes parámetros a partir de estimaciones del primer y segundo momento de los gradientes [Kingma and Ba, 2014]. Adam está basado en la combinación de otros dos optimizadores: AdaGrad [Duchi et al., 2011] y RMSProp [Hinton et al., 2012]. Este optimizador combina la ventaja de AdaGrad para lidiar con gradientes dispersos y la de RMSProp para tratar con objetivos no estacionarios. La ecuación 6.1 define a este optimizador:

$$\theta_{t+1,i} = \theta_{t,1} - \frac{\eta}{\sqrt{\epsilon + \sum_{\tau=1}^{t}(\nabla J(\theta_{\tau,i}))^2}} \nabla J(\theta_{t,i}) \tag{6.1}$$

Stochastic Gradient Descent (SGD)

Este optimizador elimina la redundancia que provocaba el descenso de gradientes por lotes en grandes conjuntos de datos ya que SGD realiza actualizaciones de gradiente para cada muestra cada vez que se actualiza [Ruder, 2016]. Por lo tanto, este método suele ser más rápido y realiza actualizaciones frecuentes con una alta variación que hacen que la función objetivo fluctúe mucho, lo que permite saltar a mínimos locales nuevos y potencialmente mejores. Las ventajas que destacan de este optimizador son su eficiencia y facilidad de implementación, mientras que entre sus desventajas se encuentran que este

método requiere de varios hiperparámetros y es sensible a la escala de características. La ecuación 6.2 define a este método de optimización:

$$\theta = \theta - \eta * \nabla_\theta J(\theta) \tag{6.2}$$

El resto de modelos que se han aplicado en este estudio han sido:

- **ResNet50, ResNet101 y ResNet152**: esta arquitectura se basa en el principio de aprender de residuos en vez de aprender características. Para realizar ese aprendizaje residual ResNet usa conexiones de acceso rápido (*shortcut connections*) que realizan el mapeo de identidades, y sus salidas se suman a las de las capas apiladas. Este mecanismo permite que estas capas se ajusten a un mapeo residual en lugar de ajustarse a un mapeo subyacente deseado [He et al., 2016]. La diferencia entre estos modelos es la cantidad de capas que poseen.

- **ResNet50V2, ResNet101V2 y ResNet152V2**: estos modelos están basados en el principio de Redes Residuales explicado en la entrada anterior pero con algunas diferencias. En esta versión la mejora principal es la disposición de las capas en los bloques residuales, donde la normalización del *batch* y la función de activación ReLU, van antes de la convolución 2D [Atienza, 2018].

- **VGG16 y VGG19**: están basados en una Red Neuronal Convolucional que tiene la particularidad de que no contiene una gran cantidad de hiperparámetros, sino que se compone de capas de convolución de 3x3 filtros de *stride* 1 y el mismo *padding*. Además, tiene una capa *max pool* de 2x2 filtros de *stride* 2, que suele ser inferior si lo comparamos con otras redes del mismo tipo [Simonyan and Zisserman, 2014]. El número al final del nombre indica el número de capas que tiene, de esta forma, VGG16 está formado por 16 capas mientras que VGG19 por 19 capas.

- **InceptionV3**: la premisa sobre la que se creó esta arquitectura fue usar filtros de diferentes tamaños al mismo nivel. Por esta razón, el módulo denominado *inception* realiza una convolución con tres filtros de tamaños diferentes (1x1, 3x3, 5x5). A continuación, las salidas se concatenan y se envían al siguiente módulo de *inception*. En la tercera versión se introdujeron algunas mejoras para reducir tanto la complejidad computacional como los cuellos de botella. Estas mejores fueron básicamente: usar el optimizador *RMSprop*, convoluciones factorizadas de 7x7, normalización del *batch* en los clasificadores auxiliares e incorporar el *Label Smoothing*, cuyo objetivo es prevenir el sobreajuste, añadiendo un componente de regularización a la función de coste [Szegedy et al., 2016].

- **InceptionResNetV2**: este modelo está compuesto por 164 capas y está basado en una combinación de la estructura de *Inception* y la conexión residual. Esta situación se traduce en que una diversidad de filtros convolucionales se combinan con conexiones residuales y en consecuencia se produce una reducción del tiempo de entrenamiento y evita el problema de degradación [Szegedy et al., 2017].

- **MobileNetV2**: Es un modelo que es usado en entornos móviles debido a su precisión y rapidez. Está basado en su predecesor, la red MobileNetV1, pero tiene

algunas características adicionales. Estas características son que incluye cuellos de botella lineales entre las capas y conexiones de acceso rápido entre los cuellos de botella. Estos cuellos de botella son los encargados de codificar las entradas y salidas intermedias del modelo, mientras que la capa interna tiene asignada la función de transformar conceptos de nivel inferior a descriptores de nivel superior [Sandler et al., 2018].

- **DenseNet121, DenseNet169 y DenseNet201**: los principales componentes de esta arquitectura son los bloques densos y las capas de transición. Los bloques densos definen cómo se concatenan las entradas y las salidas, mientras que las capas de transdición controlan el número de canales con la finalidad de que no crezca demasiado [Huang et al., 2017]. La diferencia entre estas tres versiones radica en el número de capas que contengan estos modelos.

- **NasNetLarge**: Este modelo se engloba en el paradigma de *Automated Machine Learning* que tiene el objetivo de automatizar procesos lentos y repetitivas en el desarrollo de modelos de ML, como es el caso del entrenamiento de estos modelos. *NasNetLarge* está basado en algoritmos de optimización como Reinforcement Learning. El propósito de esta acción es evitar una inserción repetitiva de capas, con la característica de que esta Red Neuronal Convolucional puede ser entrenada por sus parámetros pero también por su propia arquitectura, añadiendo y cambiando capas, funciones de activación y conexiones [Zoph et al., 2018].

A continuación, se va a explicar la implementación que se ha seguido para realizar la clasificación de los gestos de la base de datos 20BN-Jester usando transferencia de aprendizaje con el framework Keras.

Un aspecto muy importante es definir las clases que se quieren clasificar en la tarea, para ello se define un set que contiene las etiquetas de cada una de las clases. En este estudio se han realizado experimentos con 3, 5, 8 y 10 clases, con lo cual en el Listado 6.1 se puede apreciar los set creados para cada uno de los casos. Se entrenaron los modelos con un número diferente de gestos para comprobar si se identificaban mejor los gestos cuando había un número mayor o menor de ellos.

```
1 ETIQUETAS_3g = set(["swipe_left", "thumb_up", "stop_sign"])
2
3 ETIQUETAS_5g = set(["swipe_left", "thumb_up", "stop_sign", "swipe_down"
      , "slide_2_fingers_right"])
4
5 ETIQUETAS_8g = set(["swipe_left", "thumb_up", "stop_sign", "swipe_down"
      , "slide_2_fingers_right", "slide_2_fingers_up", "
     zoom_in_with_2_fingers", "pull_hand_in"])
6
7 ETIQUETAS_10g = set(["swipe_left", "thumb_up", "stop_sign", "swipe_down
     ", "slide_2_fingers_right", "slide_2_fingers_up", "
     zoom_in_with_2_fingers", "pull_hand_in", "zoom_in_with_full_hand",
     "zoom_out_with_2_fingers"])
```

Listado 6.1: Etiquetas para la clasificación de los gestos.

Modelo	Ventajas	Desventajas
ResNet50; ResNet101; ResNet152	• Mejora del rendimiento de entrenamiento evitando aprender características repetidas. • Evita el problema de *exploding gradient*. • Reducción del tiempo de entrenamiento. • Evita el problema de *vanishing gradient*.	• El *shortcut* de identidad limita su capacidad de representación [Zhang et al., 2021].
ResNet50V2; ResNet101V2; ResNet152V2	• La velocidad de convergencia es más rápida que en su primera versión. • Evita el problema de *exploding gradient*. • Reducción del tiempo de entrenamiento. • Mejora del rendimiento de entrenamiento evitando aprender características repetidas. • El uso de la preactivación fortalece la regularización del modelo. • Evita el problema de *vanishing gradient*.	• El *shortcut* de identidad limita su capacidad de representación.
VGG16; VGG19	• Mejora de la profundidad. • Mejora de la velocidad. • Incremento de la no linealidad debido al número de capas con kernels más pequeños.	• El problema del *vanishing gradient*. • Es más lenta que ResNet. • Los pesos de esta arquitectura son muy grandes.
InceptionV3	• Mayor facilidad para procesar representaciones de mayor dimensión. • Evitar cuellos de botella. • Equilibrio entre la anchura y la profundidad de la red.	• El tiempo de entrenamiento requiere mucho tiempo.

Tabla 6.2: Tabla comparativa de los modelos de *Deep Learning* (Parte I).

Después de definir las etiquetas de cada una de las clases que se quiere detectar en el conjunto de datos, es necesario cargar las imágenes y realizar el preprocesamiento de cada una de las imágenes, por consiguiente se utilizará la librería OpenCV para este cometido. Para cargar la imagen se hará uso del método *imread()* (ver línea 1 Listado 6.2) y a continuación se realizará el preprocesamiento que consistirá en dos pasos:

1. Cambiar el canal de color a RGB (ver línea 2 Listado 6.2).

2. Redimensionar el tamaño de la imagen a 224x224 (ver línea 3 Listado 6.2).

```
1 imagen = cv2.imread(rutaImagen)
2 imagen = cv2.cvtColor(imagen, cv2.COLOR_BGR2RGB)
3 imagen = cv2.resize(imagen, (224, 224))
4 (trainX, testX, trainY, testY) = train_test_split(datos, etiquetas,
      test_size=0.25, stratify=etiquetas, random_state=30)
```

Listado 6.2: Preprocesamiento de las imágenes y asignación del conjunto de datos.

Ambas medidas son para compatibilizar las imágenes para usar *fine-tuning* con Keras debido a que este framework requiere que las imágenes estén en el espacio de color RGB,

Modelo	Ventajas	Desventajas
InceptionResNetV2	• Disminuye el tiempo de entrenamiento de la red Inception significativamente [Szegedy et al., 2017]. • Evita el problema de *exploding gradient*. • Alcanza una mayor precisión en un tiempo reducido • Evita el problema de *vanishing gradient*.	• No se puede asignar un número de filtros muy alto porque entonces el entrenamiento no se completa.
MobileNetV2	• Modelo de tamaño reducido. • Es rápido en rendimiento que lo convierte en apropiado para aplicaciones móviles. • Número de parámetros reducidos. • Red Neuronal Convolucional de baja latencia. • Evita el overfitting.	• Menor precisión. • Los cuellos de botella durante el *downsampling* impiden el flujo de datos.
DenseNet121; DenseNet169; DenseNet201	• Mejora la propagación de las características tanto hacia delante como hacia atrás. • Disminuye el problema de *vanishing gradient*. • Fomenta la reutilización de características. • Reduce el número de parámetros.	• Tiene una gran cantidad de memoria redundante que dificulta la convergencia del modelo. • Requiere memoria cuadrática respecto a su profundidad. • Muchas conexiones residuales incrementa la posibilidad de overfitting. • La concatenación densa provoca que este modelo requiera mucha memoria de la GPU y más tiempo de entrenamiento [Lodhi and Kang, 2019]. • El uso repetido de la información durante el entrenamiento hace que no aprenda características de alto nivel para tareas complejas.
NasNetLarge	• Capacidad de aprender por sí mismo, sin la intervención humana. • Alta precisión.	• Tiempo de entrenamiento muy lento.

Tabla 6.3: Tabla comparativa de los modelos de *Deep Learning* (Parte II).

y respecto al tamaño se podría incluir un tamaño diferente, pero siempre hay que tener en cuenta las dimensiones de entrada con las que se ha entrenado los distintos modelos que se van a utilizar, ya que incluir unas dimensiones muy alejadas de ese valor afectará negativamente al rendimiento.

A continuación, se va a separar el conjunto de datos para de esta forma construir un modelo fiable. Esta separación se hará con ayuda de la librería *Scikit-learn* y se divide en tres partes: entrenamiento, validación y test. *Scikit-learn* es una librería de código abierto basada en el análisis de datos, la cual contiene numerosos algoritmos de ML [Kramer, 2016]. Esta librería es muy útil para esta tarea porque tiene implementada la función *train_test_split()* que va a separar el conjunto de datos con el fin de que pueda ser utilizado para el entrenamiento y la validación (ver línea 4 Listado 6.2). Los parámetros más relevantes de esta función son:

- *test_size*: En este caso se ha definido a 0.1 para que asigne el 10 % de los datos al conjunto de validación y el 80 % al de entrenamiento, dejando el restante 10 % del conjunto de datos para el test.

- *random_state*: A este parámetro se le asigna un número para que cada vez que se ejecuta el código los resultados sean los mismos ya que si se omite el método proporcionará diferentes resultados en cada ejecución.

Uno de los inconvenientes de usar *fine-tuning* es el *overfitting* y como consecuencia se utiliza *Data Augmentation* para paliar este problema. Esta técnica que se aplica entre el preprocesamiento y el entrenamiento, permite modificar ligeramente los datos de entrada para así crear nuevos datos y proporcionar ciertos beneficios al entrenamiento del modelo. Las técnicas más comunes que se suelen aplicar cuando se realiza *Data Augmentation* en un conjunto de datos formados por imágenes son: voltear horizontal o verticalmente, rotar, recortar, hacer zoom o modificar el brillo o el contraste. Este método se puede clasificar principalmente en dos categorías: basado en datos y basado en caracterísiticas. En esta Tesis se ha integrado *Data Augmentation* basado en los datos porque ofrecen un mejor rendimiento y provocan una mayor reducción del *overfitting* que en el modo basado en características [Wong et al., 2016]. Al utilizar el framework Keras se ha realizado data augmentation en los datos usando la clase *ImageDataGenerator* (ver Listado 6.3).

```
augmentation = ImageDataGenerator(
    rotation_range=90,
    zoom_range=0.15,
    width_shift_range=[-150,150],
    height_shift_range=0.5,
    shear_range=0.15,
    horizontal_flip=True,
    britghtness_range=[0.3,1.0],
    fill_mode="nearest")
```

Listado 6.3: Data augmentation con la clase ImageDataGenerator.

El siguiente paso es cargar el modelo que se quiere entrenar de todos los que dispone Keras. En el Listado 6.4 se muestran los modelos que se han utilizado en este desarrollo.

Después de obtener el modelo que se va a utilizar para la clasificación, se tiene que crear una nueva capa totalmente conectada y reemplazarla por la original, situándola a la cabeza del modelo que se ha cargado previamente (ver Listado 6.5).

Es necesario clarificar que si se trabaja con los modelos InceptionV3 e InceptionResNetV2 hay que incluir el hiperparámetro padding con el valor "same" para asegurar que la longitud en cada una de las dimensiones tanto de la entrada como de la salida son iguales o de lo contrario dará un error (ver línea 7 Listado 6.5).

Para terminar este proceso de modificación del modelo, hay que "interrumpir" el proceso de *backpropagation* en la red neuronal convolucional, con el objetivo de que no se actualicen los pesos de las capas y de esta manera no perder las características que serán útiles para la tarea de reconocimiento de gestos. Para hacer esto, se debe de poner a *False* el atributo *trainable* de cada una de las capas (ver línea 8 Listado 6.5).

```
modelo = ResNet50(weights="imagenet", include_top=False, input_tensor=
    Input(shape=(224, 224, 3)))
modelo = ResNet101(weights="imagenet", include_top=False, input_tensor=
    Input(shape=(224, 224, 3)))
modelo = ResNet152(weights="imagenet", include_top=False, input_tensor=
    Input(shape=(224, 224, 3)))
modelo = ResNet50V2(weights="imagenet", include_top=False, input_tensor
    =Input(shape=(224, 224, 3)))
modelo = ResNet101V2(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))
modelo = ResNet152V2(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))
modelo = vgg16.VGG16(include_top=False, weights='imagenet',
    input_tensor=Input(shape=(224, 224, 3)))
modelo = vgg19.VGG19(include_top=False, weights='imagenet',
    input_tensor=Input(shape=(224, 224, 3)))modelo = inception_v3.
    InceptionV3(include_top=False, weights='imagenet', input_tensor=
    Input(shape=(224, 224, 3)))
modelo = inception_resnet_v2.InceptionResNetV2(include_top=False,
    weights='imagenet', input_tensor=Input(shape=(224, 224, 3)))
modelo = mobilenet_v2.MobileNetV2(input_shape=None, alpha=1.0,
    include_top=False, weights='imagenet', input_tensor=Input(shape
    =(224, 224, 3)))
aseModel = densenet.DenseNet121(include_top=False, weights='imagenet',
    input_tensor=Input(shape=(224, 224, 3)))
modelo = densenet.DenseNet169(include_top=False, weights='imagenet',
    input_tensor=Input(shape=(224, 224, 3)))
modelo = densenet.DenseNet201(include_top=False, weights='imagenet',
    input_tensor=Input(shape=(224, 224, 3)))
modelo = nasnet.NASNetLarge(input_shape=None, include_top=False,
    weights='imagenet', input_tensor=Input(shape=(224, 224, 3)))
```

Listado 6.4: Modelos de Keras.

```
nueva_fclayer = modelo.output
nueva_fclayer = AveragePooling2D(pool_size=(7, 7))(nueva_fclayer)
nueva_fclayer = Flatten(name="flatten")(nueva_fclayer)
nueva_fclayer = Dense(512, activation="relu")(nueva_fclayer)
nueva_fclayer = Dropout(0.5)(nueva_fclayer)
nueva_fclayer = Dense(len(lb.classes_), activation="softmax")(
    nueva_fclayer)
nueva_fclayer = AveragePooling2D(pool_size=(7, 7), padding='same')(
    nueva_fclayer)
for capa in modelo.layers:
    capa.trainable = False
```

Listado 6.5: Creación de las capas para hacer *fine-tuning*.

Posteriormente, se tiene que compilar el modelo, para lo cual es necesario elegir el optimizador adecuado que en este caso serían Adam y SGD (ver líneas 1 y 2 del Listado 6.6). Por otro lado, hay que designar también la función de coste donde en este trabajo se ha optado por *mean squared error* y *categorical crossentropy* que son los más utilizados

para los problemas de multiclasificación (ver líneas 3 y 4 del Listado 6.6), aunque se podrían utilizar otros como *Hinge loss*, *Hamming loss* o *Huber loss*, entre otros.

```
1 optimizador = Adam(lr=0.001, beta_1=0.9, beta_2=0.009, epsilon=0.1)
2 optimizador = SGD(lr=1e-4, momentum=0.9, decay=1e-4 / epochs)
3 modelo.compile(loss="categorical_crossentropy", optimizer=optimizador,
      metrics=["accuracy"])
4 modelo.compile(loss="mean_squared_error", optimizer=optimizador,
      metrics=["accuracy"])
```

Listado 6.6: Compilación del modelo.

Después de compilar el modelo, se entrenan las capas que han sido reemplazadas anteriormente, ya que el resto de ellas han sido interrumpidas para que no se actualicen sus valores durante esta fase (ver Listado 6.7).

```
1 modelo = model.fit_generator(
2     trainAug.flow(trainX, trainY, batch_size=32),
3     steps_per_epoch=len(trainX) // 32,
4     validation_data=augmentation.flow(testX, testY),
5     validation_steps=len(testX) // 32,
6     epochs=epochs)
```

Listado 6.7: Entrenamiento del modelo.

El modelo necesita pasar una evaluación para comprobar que efectivamente este funciona correctamente y ofrece unos resultados satisfactorios en la clasificación. Para alcanzar este fin, primero se obtuvieron las predicciones del modelo (ver línea 1 del Listado 6.8) con el propósito de comprobar si ha sido capaz de identificar las clases correctamente y acto seguido se obtuvo la precisión del modelo mediante el método *evaluate()*, cuyos resultados para cada uno de los modelos se pueden consultar en el Apéndice A, en la sección A.2.1. Después de obtener estos datos, se integraron las predicciones en el método *classification_report* procedente de la API que se mencionó anteriormente *scikit-learn* (ver línea 2 del Listado 6.8). Este método permite obtener las métricas de *precision*, *recall*, *f1_score* y *support* a través de las predicciones extraídas para estimar el rendimiento del modelo aplicado en la tarea.

```
1 predicciones = modelo.predict(testX, batch_size=32)
2 evaluaciones = model.evaluate(testX, testY)
3 informe = classification_report(testY.argmax(axis=1), predicciones.
      argmax(axis=1), target_names=lb.classes_)
4 num_gestures = 10
5 classification_report_csv(informe, filename_class, num_gestures)
```

Listado 6.8: Evaluación del modelo.

El método *classification_report_csv()* (ver Listado 6.9) tiene el objetivo de almacenar los valores de cada una de las métricas que se obtienen del método *classification_report* en un fichero para que pueda alimentar la entrada del sistema experto con Lógica Difusa.

```
def classification_report_csv(report, filename_class, num_gestures):
    report_data = []
    lines = report.split('\n')
    end = 2+num_gestures
    for line in lines[2:end]:
        row = {}
        row_data = line.split()
        row_data = list(filter(None, row_data))
        row['class'] = row_data[0]
        row['precision'] = float(row_data[1])
        row['recall'] = float(row_data[2])
        row['f1_score'] = float(row_data[3])
        row['support'] = float(row_data[4])
        report_data.append(row)
    dataframe = pd.DataFrame.from_dict(report_data)
    dataframe.to_csv(filename_class, index = False)
```

Listado 6.9: Método para almacenar los resultados de las métricas en un archivo csv.

A continuación, se crean unas gráficas con la ayuda de la librería matplotlib[8] donde se representan los valores de la precisión y el error obtenidos del conjunto de entrenamiento y validación (ver Listado 6.10).

```
N = epochs
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, N), modelo.history["loss"], label="train_loss")
plt.plot(np.arange(0, N), model.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, N), model.history["accuracy"], label="train_acc")
plt.plot(np.arange(0, N), model.history["val_accuracy"], label="val_acc
    ")
plt.ylim((-0.5, 1.5))
plt.title("Precision y Error del Entrenamiento")
plt.xlabel("Num Epoch")
plt.ylabel("Precision/Error")
plt.legend(loc="lower left")
plt.savefig(ruta_figuras + "figura.png")
plt.show()
```

Listado 6.10: Gráficas de la precisión y error del conjunto de datos.

Por último, se serializa tanto el modelo como el *label binarizer* para poder usarlo posteriormente (ver Listado 6.11).

```
modelo.save(path_model)
# Serialize the label binarizer to disk
f = open(label_bin, "wb")
f.write(pickle.dumps(lb))
f.close()
```

Listado 6.11: Serialización del modelo y las etiquetas.

De este proceso se obtienen unas gráficas que reflejan el valor de la precisión y la

[8]Matplotlib - `https://matplotlib.org/`

pérdida para los conjuntos de entrenamiento y validación, y una serie de métricas. Estas métricas calculadas por el framework *scikit-learn* [Pedregosa et al., 2011] son:

- Precision: Es la métrica que cuantifica el número de predicciones positivas realizadas correctamente. Esta métrica se puede expresar como:

$$precision = true\_positive/(true\_positive + false\_positive)$$

- Recall: Es una métrica que cuantifica el número de predicciones positivas correctas realizadas a partir de todas las predicciones positivas. A diferencia de la precisión, esta métrica proporciona una indicación de predicciones positivas fallidas. Recall se representa con la siguiente expresión:

$$recall = true\_positive/(true\_positive + false\_negative)$$

- F1-score: Esta métrica proporciona una forma de combinar *precision* y *recall* en una sola medida que engloba ambas propiedades. El cálculo de esta métrica se realiza de la siguiente manera:

$$f1 - score = (2 * Precision * Recall)/(Precision + Recall)$$

- Support: Es el número de apariciones de cada clase en y_true. Es el número de ocurrencias de la clase dada en su conjunto de datos.

Para entender mejor las ecuaciones anteriores es necesario definir algunos conceptos [Wang and Zheng, 2013]:

- True_positives: Son las muestras correctamente clasificadas como positivas.
- False_negatives: Son las muestras clasificadas incorrectamente como negativas.
- False_positives: Son las muestras incorrectamente clasificadas como positivas.
- True_negatives: Son las muestras correctamente clasificadas como negativas.

Este proceso que se corresponde con el módulo *Deep Learning + Transfer Learning* obtiene los modelos y las etiquetas de los entrenamientos realizados con los diferentes algoritmos así como las métricas que serán la entrada del siguiente módulo denominado Sistema Experto Difuso, el cual ha sido implementado utilizando la API *fuzzylite* [Rada-Vilela, 2018] y se va a describir a continuación.

6.1.1.3. Sistema experto difuso

En primer lugar, es necesario definir las funciones de pertenencia. La definición de un función de pertenencia en Lógica Difusa sería: "una función de pertenencia para un conjunto difuso A en el universo del discurso X se define como $\mu_A : X \rightarrow [0, 1]$, donde cada elemento de X se asigna a un valor entre 0 y 1". Este valor cuantifica el grado de pertenencia del elemento en X al conjunto difuso A. Las funciones miembro que se han utilizado en este trabajo son: triangular, Gaussiana, campana generalizada y rampa.

La función triangular viene definida por tres parámetros: a corresponde con el límite más bajo del intervalo, b con el límite más alto y m define el centro de esta función. En la ecuación 6.3 se puede observar las relaciones entre los diferentes parámetros.

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{m-a}, & a < x \leq m \\ \frac{b-x}{b-m}, & m < x < b \\ 0, & x \geq b \end{cases} \tag{6.3}$$

La función Gaussiana está representada en la ecuación 6.4, donde los principales parámetros que definen esta función son m que representa el valor central y la desviación estándar k que tiene que ser mayor que 0.

$$\mu_A(x) = e^{-\frac{(x-m)^2}{2k^2}} \tag{6.4}$$

La función de campana generalizada está definida por tres parámetros: a representa la anchura, b la pendiente y m es el centro. La ecuación 6.5 representa esta función.

$$\mu_A(x) = \frac{1}{1+ \mid \frac{x-m}{a} \mid^2 b} \tag{6.5}$$

La función de rampa en *Fuzzylite* figura en la imagen 6.6 está definida por dos funciones:

- La función-R que tiene como parámetros $a = b = -\infty$ que se muestran en la ecuación 6.6

$$\mu_A(x) = \begin{cases} 0, & x > b \\ \frac{b-x}{b-a}, & a \leq x \leq b \\ 1, & x < a \end{cases} \tag{6.6}$$

- La función-L que tiene como parámetros $c = d = +\infty$, representadas en la ecuación 6.7

$$\mu_A(x) = \begin{cases} 0, & x < c \\ \frac{x-c}{d-c}, & c \leq x \leq d \\ 1, & x > d \end{cases} \tag{6.7}$$

Las funciones de pertenencia triangular y gaussiana son las más usadas. En este trabajo se han probado otras funciones además de esas para ver su rendimiento en esta situación. Finalmente se ha optado por utilizar la función gaussiana debido a que ha obtenido los mejores resultados en la tarea de clasificación de los experimentos. Esta decisión se ha basado en los resultados obtenidos después de aplicar las distintas funciones al sistema de Takagi-Sugeno-Kang, donde las matrices de confusión asociadas a cada una de estas funciones de pertenencia muestran de una manera visual la diferencia de rendimiento entre las funciones triangular, gaussiana, campana generalizada y rampa (ver Figura 6.7). Estas matrices han ayudado para determinar qué función se debería elegir para obtener los mejores resultados en el sistema. Del estudio se concluyó que la

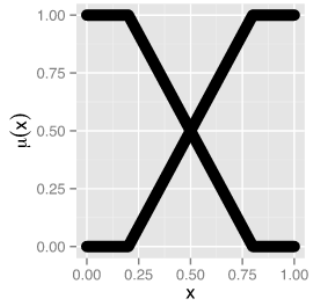


Figura 6.6: Gráfica de la función de pertenencia de rampa de *Fuzzylite*. Fuente: [Rada-Vilela, 2018]

función gaussiana sería la mejor opción porque clasifica correctamente la mayoría de los casos de las cuatro clases que componen el modelo.

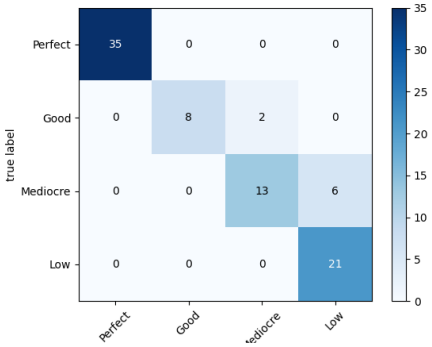
En *Fuzzylite* primero es necesario iniciar el *engine* (ver Listado 6.12) y después crear las distintas variables de entrada para el sistema difuso. Las variables que se han creado para este sistema han sido las métricas de *precision*, *recall* y *f1-score* que han sido obtenidas de los entrenamientos realizados en la fase anterior con DL. En los Listados 6.13, 6.14, 6.15, 6.16 se pueden observar algunas de estas variables para las distintas funciones de pertenencia con las que se ha experimentado en este estudio.

```
def __init__(self):
        self.engine = fl.Engine(
        name="Takagi-Sugeno-Kang_Fuzzy_System",
        description="")
```

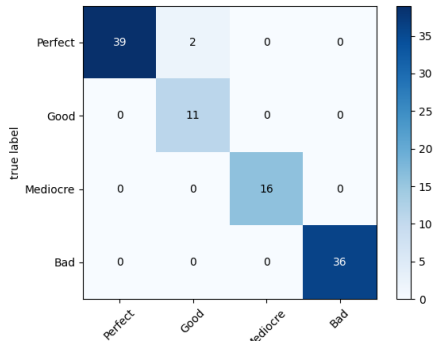
Listado 6.12: Inicialización del engine en Fuzzylite.

```
self.engine.input_variables = [
        fl.InputVariable(
            name="Precision",
            description="",
            enabled=True,
            minimum=0.000,
            maximum=1.000,
            lock_range=False,
            terms=[
                fl.Triangle("VERY_HIGH",0.9,1.0,1.0),
                fl.Triangle("HIGH",0.7,0.9,1.0),
                fl.Triangle("MEDIUM",0.35,0.7,0.9),
                fl.Triangle("LOW",0,0.35,0.7)
                ]
            ),
```

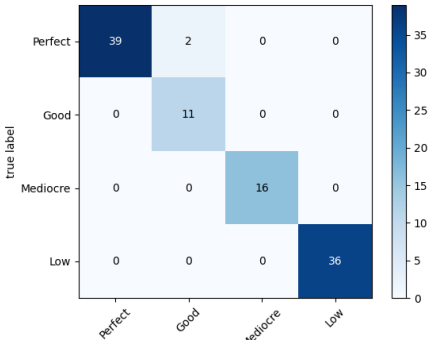
Listado 6.13: Ejemplo de variable de entrada triangular.



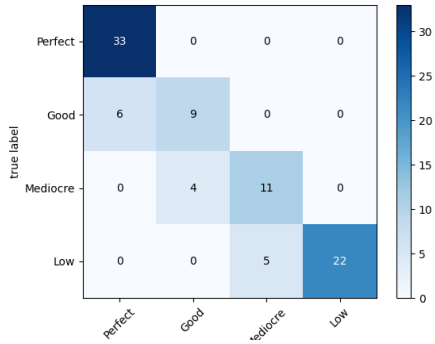
(a) Matriz de confusión de la función triangular.



(b) Matriz de confusión de la función gaussiana.



(c) Matriz de confusión de la función campana generalizada.



(d) Matriz de confusión de la función rampa.

Figura 6.7: Matrices de confusión de distintas funciones de pertenencia en el modelo Takagi-Sugeno-Kang.

```
def creating_input(self):
        x = np.arange(0,1,0.1)
        self.engine.input_variables = [
        fl.InputVariable(
            name="Precision",
            description="",
            enabled=True,
            minimum=0.000,
            maximum=1.000,
            lock_range=False,
            terms=[
                fl.Gaussian("VERY_HIGH",1.0,0.5),
                fl.Gaussian("HIGH",0.9,0.5),
                fl.Gaussian("MEDIUM",0.7,0.5),
                fl.Gaussian("LOW",0.35,0.5)
                 ]
            ),
```

Listado 6.14: Ejemplo de variable de entrada gaussiana.

```
self.engine.input_variables = [
        fl.InputVariable(
            name="Precision",
            description="",
            enabled=True,
            minimum=0.000,
            maximum=1.000,
            lock_range=False,
            terms=[
                fl.Bell("VERY_HIGH", 1.0, 8, 4), # Bell(center, width, slope)
                fl.Bell("HIGH", 0.9, 8, 4),
                fl.Bell("MEDIUM", 0.7, 8, 4),
                fl.Bell("LOW", 0.35, 8, 4)
                ]
            ),
```

Listado 6.15: Ejemplo de variable de entrada campana generalizada.

```
self.engine.input_variables = [
        fl.InputVariable(
            name="Precision",
            description="",
            enabled=True,
            minimum=0.000,
            maximum=1.000,
            lock_range=False,
            terms=[
                fl.Ramp("VERY_HIGH", 0.95, 1.0), # Ramp (start, end)
                fl.Ramp("HIGH", 0.8, 0.95),
                fl.Ramp("MEDIUM", 0.55, 0.8),
                fl.Ramp("LOW", 0.0, 0.55)
                ]
            ),
```

Listado 6.16: Ejemplo de variable de entrada rampa.

Otro elemento fundamental de este tipo de sistemas es el método que se va a utilizar. Los métodos usados en Lógica Difusa son [Singh and Lone, 2020]:

(a.) El modelo de **Mamdani** es el sistema de inferencia difusa más utilizado debido a su estructura simple, se utiliza para resolver todos los problemas generales de toma de decisiones.

(b.) Los sistemas de inferencia difusa **Takagi-Sugeno-Kang (TKS)** se utilizan para modelar sistemas complejos no lineales. El proceso completo de aplicar un operador difuso y luego aplicar un operador difuso a las entradas es el mismo que con el enfoque Mamdani. El único cambio se produce en la función de pertenencia de salida, que es lineal o constante.

(c). El modelo de **Tsukamoto**, en lugar de tener una función de pertenencia difusa de salida constante o lineal, tiene una función de pertenencia monótona, que *de-fuzzifies* utilizando el enfoque de promedio ponderado. Dado que es un enfoque de

promedio ponderado, el proceso se vuelve muy rápido y, por lo tanto, no se pierde tiempo durante el proceso detallado de *defuzzification*. La salida del método de Tsukamoto siempre es nítida, sin importar los tipos de entrada.

El modelo que se ha utilizado es TKS, las razones de esta elección son:

(a.) Este modelo es adaptable a otros algoritmos.

(b.) El proceso de defuzzificación para este sistema es más eficiente computacionalmente porque utiliza un promedio ponderado o una suma ponderada de algunos puntos de datos.

(c.) Takagi-Sugeno-Kang tiene menos parámetros que los otros modelos, con lo cual lo convierte en una metodología más sencilla.

La implementación del modelo TKS causa que la variable de salida se cree con el método de promedio ponderado y los términos sean definidos como constantes, tal y como se puede apreciar en el Listado 6.17.

```
def creating_output(self):
        self.engine.output_variables = [
        fl.OutputVariable(
            name="Result",
            description="",
            enabled=True,
            minimum=0.000,
            maximum=1.000,
            lock_range=False,
            aggregation=None,
            defuzzifier=fl.WeightedAverage("TakagiSugeno"),
            lock_previous=False,
            terms=[
                fl.Constant("PERFECT", 1),
                fl.Constant("GOOD", 0.8),
                fl.Constant("MEDIOCRE", 0.6),
                fl.Constant("BAD", 0.35),
                ]
            )
        ]
```

Listado 6.17: Variable de salida en el sistema TKS.

En este tipo de sistemas el núcleo está compuesto por una serie de reglas que definen el comportamiento del sistema experto. En este caso, se han definido 11 reglas, donde los valores de los parámetros que la integran, se pueden consultar en la Tabla 6.4.

Estas reglas se pueden ver implementadas en el Listado 6.18 y están representadas por un antecedente y un consecuente. Los antecedentes están condicionados por las variables de entrada, siendo sus posibles valores: Very High, High, Medium y Low. El consecuente de estas reglas viene definido por los valores Perfect, Good, Mediocre y Bad que corresponden a los valores de las variables de entrada y la consecución de uno de estos resultados será determinado por los antecedentes. Además, el consecuente está determinado por una sola variable denominada resultado, y el antecedente depende de los valores de las métricas de *precision*, *recall* y *f1-score* asociadas al experimento. Las

Id. Regla	Precision	Recall	F1_score	Result
#1	Very High	Very High	Very High	Perfect
#2	High	High	High	Good
#3	High	Medium	High	Good
#4	Medium	High	High	Good
#5	High	Medium	Medium	Mediocre
#6	Medium	High	Medium	Mediocre
#7	Medium	Medium	Medium	Mediocre
#8	High	Low	Medium	Mediocre
#9	Medium	Low	Medium	Mediocre
#10	Any	Any	Low	Bad
#11	Low	Any	Any	Bad

Tabla 6.4: Valores de las reglas del sistema experto difuso.

reglas han sido definidas por expertos en la materia, teniendo en cuenta que *f1-score* ha tenido el mayor peso porque involucra tanto a la precision como el *recall*, y de esta forma no obtener un mejor resultado porque una de estas métricas tiene un valor más alto. El siguiente parámetro relevante ha sido la métrica *precision*, debido a que en este caso concreto eran más aceptable los *false negatives* que los *false positives* y por ende da más prioridad a la confianza en los *true positives*. De acuerdo a estos criterios han sido creadas las reglas del sistema difuso (ver Listado 6.18).

```
1  def creating_fuzzy_rules(self):
2          self.engine.rule_blocks = [
3              fl.RuleBlock(
4                  name="Rules",
5                  description="",
6                  enabled=True,
7                  conjunction=fl.Minimum(),
8                  disjunction=fl.Maximum(),
9                  implication=None,
10                 activation=fl.Highest(),
11                 rules=[
12                     fl.Rule.create("if Precision is VERY_HIGH and Recall
    is VERY_HIGH and F1_score is VERY_HIGH then Result is PERFECT", self.
    engine),
13                     fl.Rule.create("if Precision is HIGH and Recall is
    HIGH and F1_score is HIGH then Result is GOOD", self.engine),
14                     fl.Rule.create("if Precision is HIGH and Recall is
    MEDIUM and F1_score is HIGH then Result is GOOD", self.engine),
15                     fl.Rule.create("if Precision is MEDIUM and Recall is
    HIGH and F1_score is HIGH then Result is GOOD", self.engine),
16                     fl.Rule.create("if Precision is HIGH and Recall is
    MEDIUM and F1_score is MEDIUM then Result is MEDIOCRE", self.engine),
17                     fl.Rule.create("if Precision is MEDIUM and Recall is
    HIGH and F1_score is MEDIUM then Result is MEDIOCRE", self.engine),
18                     fl.Rule.create("if Precision is MEDIUM and Recall is
    MEDIUM and F1_score is MEDIUM then Result is MEDIOCRE", self.engine),
19                     fl.Rule.create("if Precision is HIGH and Recall is LOW
     and F1_score is MEDIUM then Result is MEDIOCRE", self.engine),
20                     fl.Rule.create("if Precision is MEDIUM and Recall is
    LOW and F1_score is MEDIUM then Result is MEDIOCRE", self.engine),
21                     fl.Rule.create("if F1_score is LOW then Result is BAD"
```

```
    , self.engine),
                    fl.Rule.create("if Precision is LOW then Result is BAD
    ", self.engine)
                ]
            )
        ]
```

Listado 6.18: Reglas del sistema experto difuso.

En definitiva, los pasos que se han seguido para realizar el sistema experto difuso han sido:

(1.) Obtener los valores de las métricas de los 104 experimentos.

(2.) Crear el sistema difuso: creación de las variables de entrada, variables de salida y conjunto de reglas.

(3.) Introducir como entrada las métricas del paso 1.

(4.) Analizar cada uno de los experimentos según las reglas definidas en el sistema difuso.

(5.) Obtener una clasificación de los experimentos ordenados de mejor a peor.

(6.) Por último, evaluar el sistema.

El resultado final de este proceso será una tabla de clasificación que reflejará los experimentos que han sido más aptos en la tarea de reconocer los gestos con las manos. Es necesario aclarar que para generar esta tabla de clasificación se tiene en cuenta principalmente el resultado que devuelve el conjunto de reglas. Sin embargo, hay situaciones en las que el resultado es el mismo para varios experimentos, en este caso se comprobará el valor de las métricas de *f1-score*, *precision* y *recall*, en este orden. Esta afirmación significa que primero se revisará el valor de *f1-score* de los experimentos en conflicto, siendo designado para alcanzar una posición más elevada en la clasificación aquel con el valor más alto de *f1-score*. Si los valores de la métrica *f1-score* de los ensayos involucrados fueran idénticos, entonces se comprobaría el valor de la variable *precision*, y así sucesivamente.

La evaluación se ha realizado comparando las decisiones que ha tomado el sistema difuso con las que tomaría un experto en la temática. En este proceso se han comprobado dos aspectos: la asignación que hace el sistema experto del experimento y el orden de la tabla de clasificación. El primero de estos aspectos hace referencia a la asignación por parte del sistema de una de las etiquetas que se ha comentado anteriormente, (*Perfect*, *Good*, *Mediocre* y *Bad*) cuando se analiza los valores de las métricas asociadas al experimento. El otro elemento está relacionado con el orden que establece el sistema experto a cada uno de los ensayos en la tabla de clasificación teniendo en cuenta los resultados obtenidos de los mismos.

6.2. Resultados

Inicialmente se realizaron unas pruebas con 150 imágenes por gesto, donde se probaron 3 gestos (*swipe left*, *thumb up* y *stop sign*) realizando aprendizaje por transferencia con *fine-tuning* con el modelo ResNet50. Después de estas pruebas, se duplicaron el número de imágenes por gesto y se probaron varios modelos que han sido descritos anteriormente en este capítulo. Además de ir variando los modelos también se fueron cambiando varios de los parámetros como el optimizador, la función de coste o el número de epochs, y de esta forma probar distintas configuraciones para observar su comportamiento. Finalmente, se hicieron las pruebas con los modelos que mostraron mejores resultados durante el proceso, donde el incremento del número de imágenes por gesto y la modificación del número de epochs fueron los cambios más notorios en esta última fase.

Los parámetros, que se han mencionado anteriormente, han sido:

- **Modelo**: son los modelos pre-entrenados que Keras dispone en su API para poder aplicar transferencia de aprendizaje.
- **Optimizador**: son los algoritmos que se han usado para ir actualizando los pesos e ir minimizando la función de pérdida durante el proceso de entrenamiento. Elegir el optimizador más adecuado no es una decisión trivial y es necesario dedicarle algún tiempo porque dicha decisión puede ser la diferencia entre que la duración del entrenamiento sea más o menos extensa. Los optimizadores que se han aplicado han sido: gradiente descendente estocástico (SGD) [Ketkar, 2017] y Adam [Kingma and Ba, 2014]. Estos optimizadores son los más usados debido a que SGD es rápido y fácil de entender y calcular y Adam es apto para grandes conjuntos de datos, es computacionalmente eficiente y consume pocos recursos de memoria [Ruder, 2016].
- **Tasa de aprendizaje (learning rate)**: este parámetro controla cuánto es necesario cambiar el modelo para ajustar los pesos en relación al error estimado. Los valores que puede tener este parámetro son entre 0.0 y 1.0.
- **Epsilon**: su función consiste en evitar cualquier división por cero en la implementación. Su valor es ínfimo.
- **Número epochs**: cada *epoch* representa un ciclo donde se entrena al conjunto de datos entero una sola vez. Los valores que se le han dado han sido: 30, 50, 70, 100, 150 y 200.
- **Función de pérdida**: esta función tiene como objetivo evaluar una solución candidata donde se busca minimizar el error. Esta función está íntimamente relacionada con el optimizador.
- **Número de gestos**: es el número de gestos que se han entrenado, variando este valor entre 3, 5, 8 y 10.
- **Número de imágenes**: el número de imágenes que se han usado para entrenar al modelo. En este caso, esta cantidad variará dependiendo del número de gestos que se ha procesado.

Se van a presentar 5 de los experimentos que han obtenido mejores resultados, 5 con resultados aceptables y 5 que están entre los peores resultados de todas las pruebas que se han realizado.

Las gráficas de la Figura 6.8 y las Tablas 6.5 y 6.6 presentan la información sobre los 5 experimentos que han obtenido de los mejores resultados del total. Estos resultados han sido obtenidos por las redes convolucionales muy profundas (VGG), tanto en su versión de 16 capas como en la versión de 19 capas, con el optimizador Adam principalmente, la función de coste *Entropía cruzada categórica* y con el número de epochs entre 70 y 100, aunque predomina 100 epochs.

Por otro lado, las gráficas de la Figura 6.9 y las Tablas 6.7 y 6.8 presentan la información sobre los 5 experimentos que han obtenido unos resultados normales. Estos resultados han sido conseguidos principalmente por las redes residuales destacando la red ResNet152, con el optimizador Adam, la función de coste *Error cuadrático medio* y el número de epochs variando entre 30-200, siendo el predominante 100 epochs.

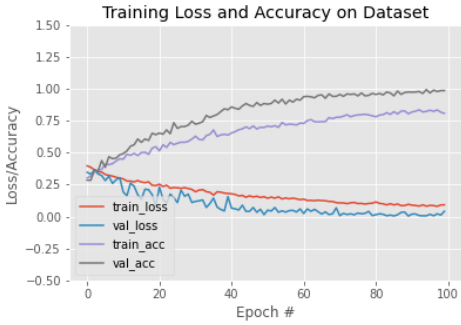
Finalmente, las gráficas de la Figura 6.10 y las Tablas 6.9 y 6.10 presentan la información sobre los 5 experimentos que han obtenido de los peores resultados del total. En este caso no ha prevalecido ningún modelo en particular habiendo una variedad de arquitecturas como redes residuales ResNet50, MobileNetV2 o DenseNet121, destacando que no aparecen redes convolucionales muy profundas (VGG16 y VGG19), el optimizador usado es SGD, la función de coste es *Error cuadrático medio* y el número de epochs sería 100.

Id	API	M	O	LR	E	NE	F	NG	NIT
27	Keras	VGG19	SGD	-	-	100	mse	3	1055
55	Keras	VGG19	Adam	0.001	0.1	100	categorical_crossentropy	5	1757
69	Keras	VGG16	Adam	0.001	0.1	100	categorical_crossentropy	8	2832
84	Keras	VGG16	Adam	0.001	0.1	100	categorical_crossentropy	10	3563
104	Keras	VGG19	Adam	0.001	0.1	70	categorical_crossentropy	10	6047

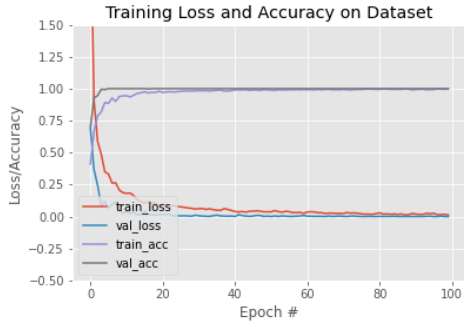
Tabla 6.5: Configuración de los 5 experimentos que han conseguido muy buenos resultados. M: Modelo / O: Optimizador / LR: Learning rate / E: Epsilon / NE: Nº de epochs / F: Función de coste / NG: Nº de gestos / NIT: Nº de imágenes totales.

Id	Gesto	Precision	Recall	F1_score	Support
27	stop_sign	0.99	0.97	0.98	90.0
	swipe_left	0.98	0.99	0.98	123.0
	thumb_up	1.0	1.0	1.0	86.0
55	slide_2_fingers_right	1.0	1.0	1.0	89.0
	stop_sign	1.0	1.0	1.0	90.0
	swipe_down	1.0	1.0	1.0	87.0
	swipe_left	1.0	1.0	1.0	88.0
	thumb_up	1.0	1.0	1.0	86.0
69	pull_hand_in	1.0	1.0	1.0	90.0
	slide_2_fingers_right	1.0	1.0	1.0	89.0
	slide_2_fingers_up	1.0	1.0	1.0	88.0
	stop_sign	1.0	1.0	1.0	90.0
	swipe_down	1.0	1.0	1.0	86.0
	swipe_left	1.0	1.0	1.0	88.0
	thumb_up	1.0	1.0	1.0	86.0
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0
84	pull_hand_in	1.0	1.0	1.0	90.0
	slide_2_fingers_right	1.0	1.0	1.0	89.0
	slide_2_fingers_up	1.0	1.0	1.0	88.0
	stop_sign	1.0	1.0	1.0	90.0
	swipe_down	1.0	1.0	1.0	86.0
	swipe_left	1.0	1.0	1.0	88.0
	thumb_up	1.0	1.0	1.0	86.0
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0
	zoom_in_with_full_hand	1.0	1.0	1.0	87.0
	zoom_out_with_2_fingers	1.0	1.0	1.0	89.0
104	pull_hand_in	1.0	1.0	1.0	154.0
	slide_2_fingers_right	1.0	1.0	1.0	152.0
	slide_2_fingers_up	1.0	1.0	1.0	159.0
	stop_sign	1.0	1.0	1.0	155.0
	swipe_down	1.0	0.99	1.0	150.0
	swipe_left	1.0	1.0	1.0	150.0
	thumb_up	1.0	0.97	0.99	147.0
	zoom_in_with_2_fingers	1.0	1.0	1.0	152.0
	zoom_in_with_full_hand	0.97	1.0	0.98	145.0
	zoom_out_with_2_fingers	1.0	1.0	1.0	145.0

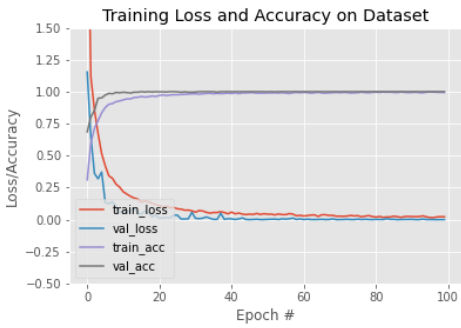
Tabla 6.6: Métricas de los 5 experimentos que han conseguido muy buenos resultados.



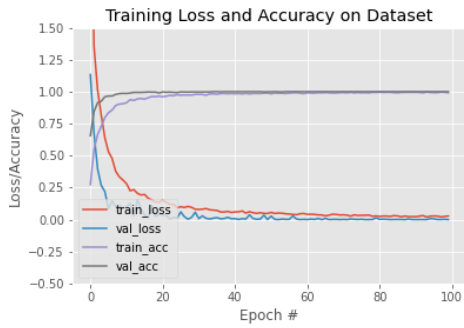
(a) Gráfica de resultados de entrenamiento del experimento con id. 27.



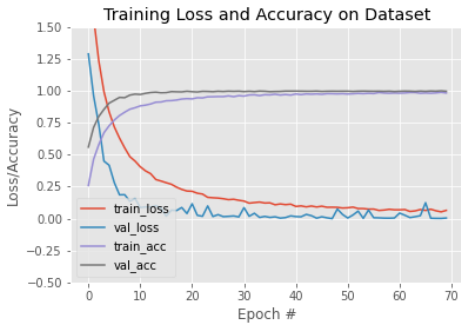
(b) Gráfica de resultados de entrenamiento del experimento con id. 55.



(c) Gráfica de resultados de entrenamiento del experimento con id. 69.



(d) Gráfica de resultados de entrenamiento del experimento con id. 84.



(e) Gráfica de resultados de entrenamiento del experimento con id. 104.

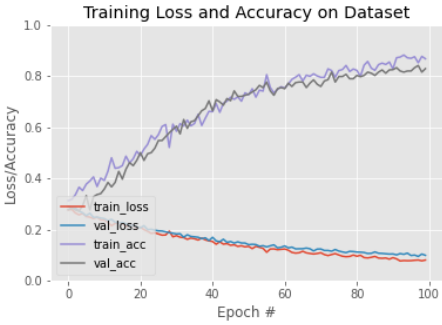
Figura 6.8: Gráficas de los 5 experimentos que han conseguido muy buenos resultados.

Id	API	M	O	LR	E	NE	F	NG	NIT
6	Keras	ResNet152	SGD	-	-	100	mse	3	1055
64	Keras	ResNet50	Adam	0.001	0.1	100	mse	8	2832
74	Keras	ResNet152	Adam	0.001	0.1	200	mse	8	2832
81	Keras	ResNet152	Adam	0.001	0.1	100	mse	10	3563
100	Keras	VGG16	Adam	0.001	0.1	30	mse	10	6047

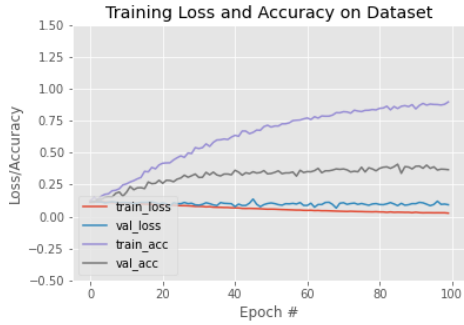
Tabla 6.7: Configuración de los 5 experimentos que han obtenido resultados normales. M: Modelo / O: Optimizador / LR: Learning rate / E: Epsilon / NE: Nº de epochs / F: Función de coste / NG: Nº de gestos / NIT: Nº de imágenes totales.

Id	Gesto	Precision	Recall	F1_score	Support
6	stop_sign	0.96	0.56	0.7	90.0
	swipe_left	0.78	0.81	0.79	88.0
	thumb_up	0.64	0.91	0.75	86.0
64	pull_hand_in	0.79	0.34	0.48	90.0
	slide_2_fingers_right	1.0	0.04	0.09	89.0
	slide_2_fingers_up	0.44	0.83	0.57	88.0
	stop_sign	0.87	0.29	0.43	90.0
	swipe_down	1.0	0.09	0.17	86.0
	swipe_left	0.91	0.35	0.51	88.0
	thumb_up	0.21	1.0	0.34	86.0
	zoom_in_with_2_fingers	1.0	0.09	0.16	91.0
74	pull_hand_in	0.67	0.83	0.74	90.0
	slide_2_fingers_right	1.0	0.38	0.55	89.0
	slide_2_fingers_up	0.77	0.98	0.86	88.0
	stop_sign	0.87	0.74	0.8	90.0
	swipe_down	1.0	0.17	0.3	86.0
	swipe_left	0.78	0.84	0.81	88.0
	thumb_up	0.41	1.0	0.59	86.0
	zoom_in_with_2_fingers	1.0	0.6	0.75	91.0
81	pull_hand_in	0.42	0.72	0.53	90.0
	slide_2_fingers_right	1.0	0.44	0.61	89.0
	slide_2_fingers_up	0.79	0.22	0.34	88.0
	stop_sign	0.75	0.56	0.64	90.0
	swipe_down	1.0	0.07	0.13	86.0
	swipe_left	0.37	0.95	0.53	88.0
	thumb_up	0.46	0.72	0.56	86.0
	zoom_in_with_2_fingers	1.0	0.52	0.68	91.0
	zoom_in_with_full_hand	1.0	0.18	0.31	87.0
	zoom_out_with_2_fingers	0.45	0.83	0.58	89.0
100	pull_hand_in	0.62	0.65	0.63	154.0
	slide_2_fingers_right	0.62	0.86	0.72	152.0
	slide_2_fingers_up	0.5	0.53	0.52	159.0
	stop_sign	0.67	0.65	0.66	155.0
	swipe_down	0.7	0.5	0.58	150.0
	swipe_left	0.67	0.41	0.51	150.0
	thumb_up	0.73	0.46	0.56	147.0
	zoom_in_with_2_fingers	0.6	0.9	0.72	152.0
	zoom_in_with_full_hand	0.59	0.74	0.66	145.0
	zoom_out_with_2_fingers	0.56	0.45	0.5	145.0

Tabla 6.8: Métricas de los 5 experimentos que han obtenido resultados normales.

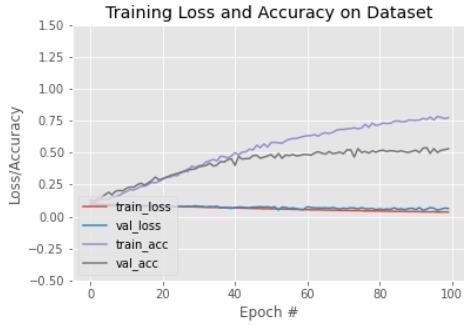


(a) Gráfica de resultados de entrenamiento del experimento con id. 6.

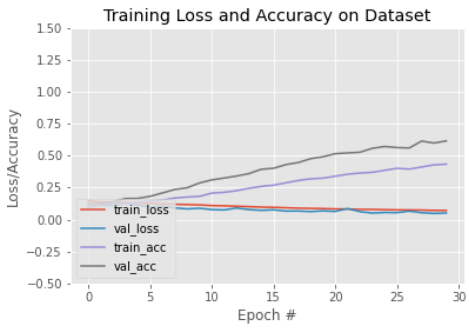


(b) Gráfica de resultados de entrenamiento del experimento con id. 64.

(c) Gráfica de resultados de entrenamiento del experimento con id. 74.



(d) Gráfica de resultados de entrenamiento del experimento con id. 81.



(e) Gráfica de resultados de entrenamiento del experimento con id. 100.

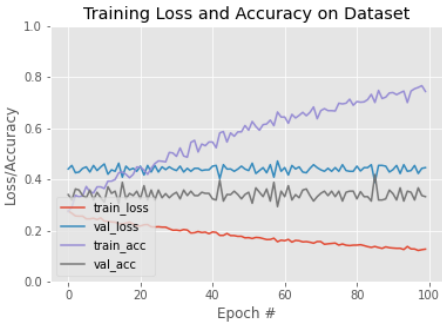
Figura 6.9: Gráficas de los 5 experimentos que han han obtenido resultados normales.

Id	API	M	O	LR	E	NE	F	NG	NIT
7	Keras	ResNet50V2	SGD	-	-	100	mse	3	1055
30	Keras	InceptionV3	SGD	-	-	100	mse	3	1055
35	Keras	MobileNetV2	Adam	0.001	0.1	100	mse	3	1055
36	Keras	DenseNet121	SGD	-	-	100	mse	3	1055
56	Keras	ResNet50	SGD	-	-	100	mse	5	1757

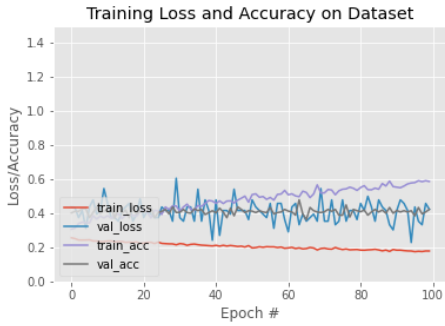
Tabla 6.9: Configuración de los 5 experimentos que han obtenido deficientes resultados. M: Modelo / O: Optimizador / LR: Learning rate / E: Epsilon / NE: Nº de epochs / F: Función de coste / NG: Nº de gestos / NIT: Nº de imágenes totales.

Id	Gesto	Precision	Recall	F1_score	Support
7	stop_sign	0.06	0.04	0.05	90.0
	swipe_left	0.0	0.0	0.0	123.0
	thumb_up	0.26	0.72	0.39	86.0
30	stop_sign	0.02	0.01	0.02	90.0
	swiping_left	0.32	0.81	0.46	88.0
	thumb_up	0.0	0.0	0.0	86.0
35	stop_sign	0.2	0.11	0.14	90.0
	swiping_left	0.4	0.11	0.18	88.0
	thumb_up	0.29	0.63	0.39	86.0
36	stop_sign	0.0	0.0	0.0	90.0
	swiping_left	0.33	1.0	0.5	88.0
	thumb_up	0.0	0.0	0.0	86.0
56	slide_2_fingers_right	0.0	0.0	0.0	89.0
	stop_sign	0.24	0.16	0.19	90.0
	swipe_down	0.0	0.0	0.0	87.0
	swipe_left	0.44	0.29	0.35	87.0
	thumb_up	0.23	0.84	0.36	86.0

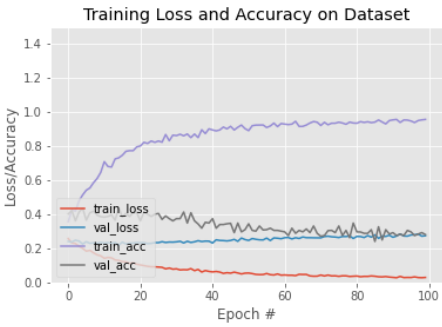
Tabla 6.10: Métricas de los 5 experimentos que han obtenido deficientes resultados.



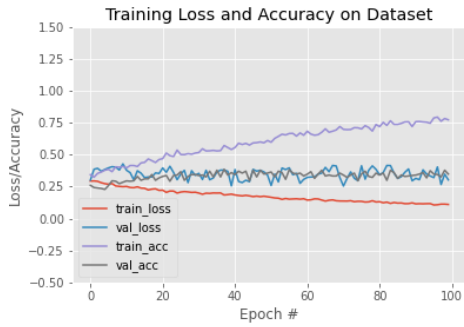
(a) Gráfica de resultados de entrenamiento del experimento con id. 7.



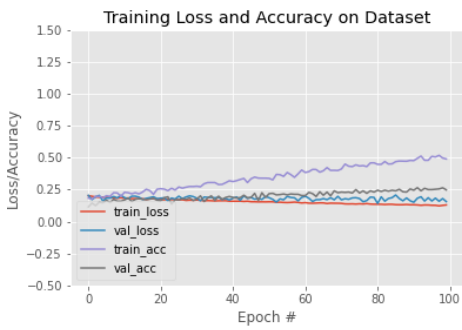
(b) Gráfica de resultados de entrenamiento del experimento con id. 30.



(c) Gráfica de resultados de entrenamiento del experimento con id. 35.



(d) Gráfica de resultados de entrenamiento del experimento con id. 36.



(e) Gráfica de resultados de entrenamiento del experimento con id. 56.

Figura 6.10: Gráficas de los 5 experimentos que han obtenido deficientes resultados.

En un principio se hicieron los ensayos con solo 3 gestos para entrenar los modelos durante 100 *epochs*, donde en cada *epoch* el conjunto de datos es pasado completamente tanto *forward* como *backward* a la red neuronal convolucional una sola vez. En este caso, han sido usadas las funciones de coste *Error cuadrático medio* y *Entropía cruzada categórica*, mientras que si las pruebas se hubieran realizado solo con 2 gestos, habría sido más apropiado utilizar una función de coste *Entropía cruzada binaria* [Janocha and Czarnecki, 2017]. Sin embargo, al tener como objetivo aumentar el número de gestos que se pretende reconocer, se consideró mejor empezar con 3 gestos para usar estas funciones de multiclasificación y tener unos resultados iniciales que nos afirmen si estamos en la dirección correcta.

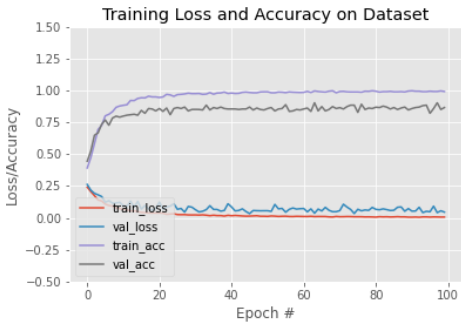
Un inconveniente que tiene DL es que para entrenar los modelos consume mucho tiempo y va a depender de factores como el número de *epochs*, la cantidad de imágenes que contenga el conjunto de entrenamiento y validación, entre otros. Por este motivo, en esta fase donde se hicieron las pruebas con solo 3 gestos, se decidió utilizar la mayoría de los modelos de los que dispone Keras para elegir solo los más prometedores en relación a los resultados obtenidos y así incrementar el número de gestos exclusivamente con estos modelos, por la razón que se ha comentado anteriormente sobre el tiempo que es necesario invertir en este proceso. El conjunto de datos utilizado cuando se realiza la transferencia de aprendizaje depende del número de gestos:

- 3 gestos → 1055 imágenes.
- 5 gestos → 1757 imágenes.
- 8 gestos → 2832 imágenes.
- 10 gestos → 3563 imágenes.

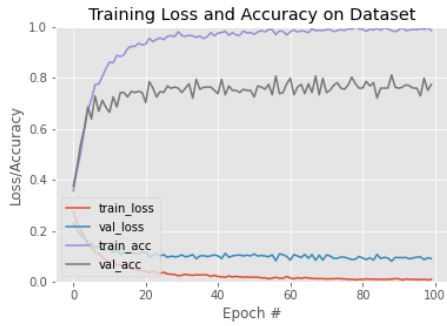
En las Figuras 6.11 y 6.12 se muestran los resultados de los diferentes modelos ejecutados donde se puede observar la diferencia entre ellos. En estas gráficas están representados los datos de entrenamiento y validación conjuntamente, tanto de la precisión como del coste, de los cuales del conjunto de datos utilizado en cada caso, se destina el 25 % al conjunto de validacíon y el restante 75 % al conjunto de entrenamiento. Los gráficos referentes a la precisión tienen que ser logarítmicos que tienden a 1.0 para obtener los resultados óptimos, en el caso del coste tienen que ser exponenciales decrecientes donde la base es inferior a 1.0 para que de esta forma los valores tiendan a 0 progresivamente. De acuerdo a los resultados obtenidos, los algoritmos denominados ResNet50, ResNet101, ResNet152, VGG16 y VGG19 han sido elegidos para realizar los siguientes experimentos. Además, se han variado la tasa de aprendizaje adquiriendo los valores 0.0001; 0.001 y 0.01, y el epsilon entre 0.1 y 1.0 que son los valores más comunes.

A continuación se han utilizado los modelos comentados en el principio de la subsección 6.1.1 pero incrementando el número de gestos a 5, 8 y 10, y el número de epochs al doble experimentando también con 200 *epochs* para comprobar los resultados obtenidos.

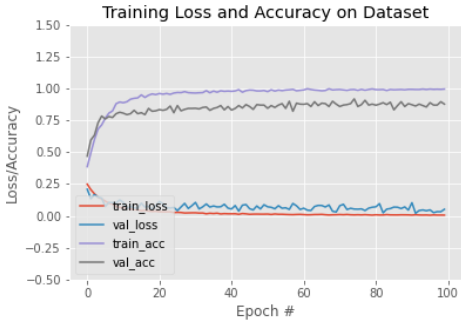
En una última fase, se han hecho los experimentos con 10 gestos, pero se ha incrementado el conjunto de datos usado pasando de 3563 a 6047 y se ha ido cambiando el número de *epochs* entre 30, 50, 70 y 100. Además, los modelos usados fueron el VGG16



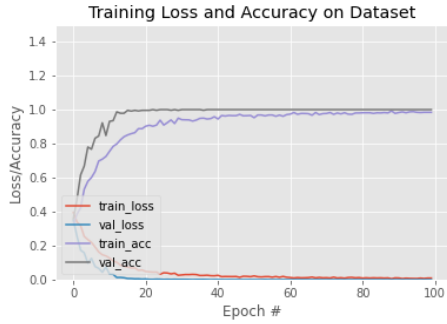
(a) ResNet50.



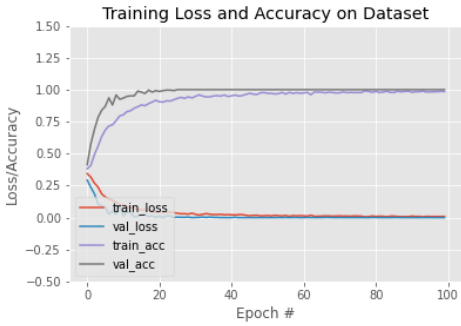
(b) ResNet101.



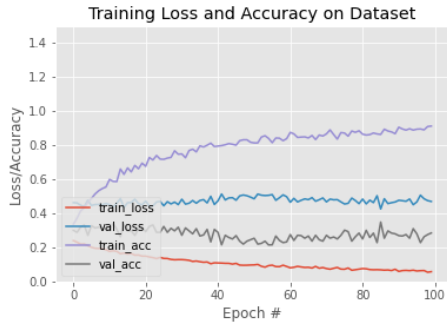
(c) ResNet152.



(d) VGG16.

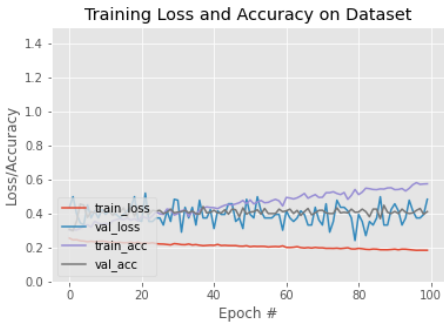


(e) VGG19.

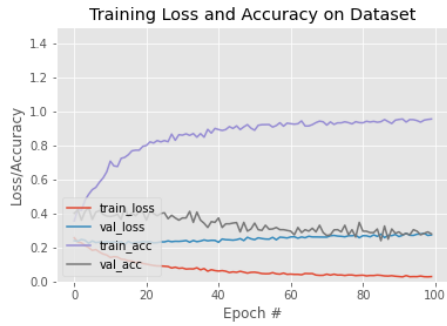


(f) InceptionV3.

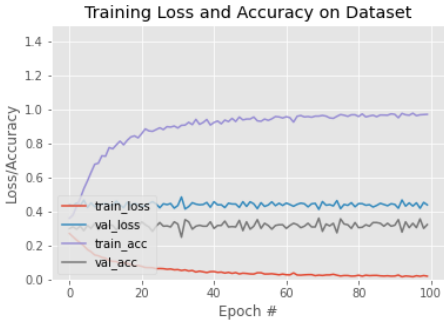
Figura 6.11: Gráficas de rendimiento de los modelos (Parte I).



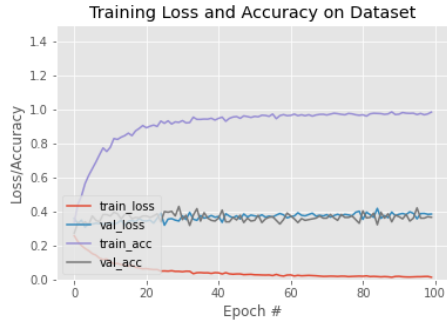
(a) InceptionResNetV2.



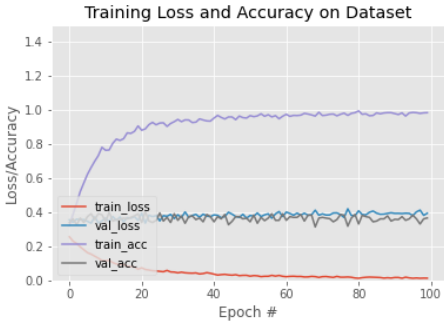
(b) MobileNetV2.



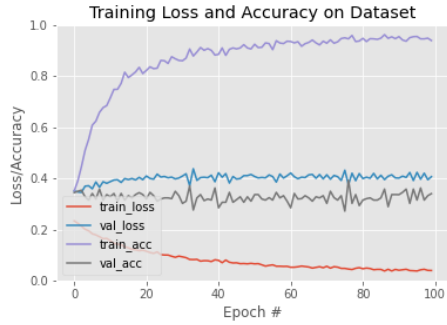
(c) DenseNet121.



(d) DenseNet169.



(e) DenseNet201.



(f) NasNetLarge.

Figura 6.12: Gráficas de rendimiento de los modelos (Parte II).

y el VGG19 porque fueron los que presentaron mejor rendimiento en los procesos anteriores. El aumento del conjunto de datos no ha supuesto mejora significativa en las métricas, por el contrario el aumento del número de epochs sí ha mejorado las métricas donde se puede ver claramente la diferencia entre los experimentos realizados con 30 y 100 *epochs*.

Después de la realización de todos los experimentos se puede observar el aumento de gestos no ha suspuesto cambios relevantes en los resultados, el optimizador Adam ha obtenido mejores resultados y también se ha observado que la función de pérdida *entropía cruzada categórica* ha obtenido mejores resultados que la cuadrática especialmente cuando el número de epochs es más reducido. Por último, cabe destacar que los algoritmos VGG16 y VGG19 cuando se utilizan con la función de coste *entropía cruzada categórica* obtuvieron los mejores resultados.

Los resultados de estos experimentos van a ser útiles para elegir los mejores algoritmos basándonos en los datos del conjunto de entrenamiento y el conjunto de validación.

6.3. Resumen

En los capítulos previos se ha utilizado la cámara Microsoft Kinect para realizar la interacción con el sistema. Sin embargo, para esta etapa se tómo la decisión de usar una webcam estándar debido principalmente a dos motivos: la incertidumbre con el futuro del dispositivo Microsoft Kinect y el hecho de que el usuario tenía que realizar una inversión adicional adquiriendo el producto para el uso del sistema.

En esta última fase se ha utilizado DL y Lógica Difusa para realizar una clasificación de los gestos con la mano y realizar una clasificación para determinar las mejores configuraciones entre todas las que se han probado. El proceso que se ha realizado ha sido el siguiente:

(1.) **Entrada de datos de entrenamiento**: En primer lugar se obtienen los datos que van a ser usados para el posterior entrenamiento con los modelos de DL. Para este propósito se han obtenido los vídeos de una base de datos de gestos con las manos titulada 20BN-Jester.

(2.) **Deep Learning**: La transferencia de aprendizaje con modalidad de fine-tuning ha sido elegida como método para la clasificación de los gestos. En este caso se utilizan una serie de modelos preentrenados del framework Keras. Estos modelos aunque estén preentrenados, es necesario actualizar la arquitectura de la red neuronal convolucional y además reentrenarla para aprender otras clases nuevas. En total se han realizado 104 experimentos donde se han modificado distintos parámetros, como por ejemplo, el optimizador, número de gestos, la función de coste, entre otros. Además de la librería Keras se ha utilizado la API scikit-learn para obtener determinadas métricas para estimar el rendimiento del modelo aplicado en la tarea. Estas métricas servirán para alimentar el sistema experto difuso.

(3.) **Sistema Experto Difuso**: Este sistema se ha implementado con la API *fuzzylite* [Rada-Vilela, 2018]. El sistema que se ha usado ha sido Takagi-Sugeno-Kang con

la función de pertenencia gaussiana y un total de 11 reglas que compondrían el sistema experto. Estas reglas tienen como antecedente la valoración de las métricas obtenidas en el apartado anterior. En definitiva, se crearía el sistema difuso con los componentes ya mencionados para posteriormente introducir como entrada las métricas y de este modo analizar cada uno de los experimentos según las reglas definidas en el sistema difuso. Por último, se obtendría una clasificación de los experimentos ordenados de mejor a peor de acuerdo a la valoración que proporciona este sistema en el proceso de evaluación de las reglas.

En relación a los resultados obtenidos se puede afirmar que las redes residuales (ResNet50, ResNet101 y ResNet152), junto con las redes convolucionales muy profundas (VGG16 y VGG19) son los modelos más adecuados para reconocer los gestos con las manos presentados en este trabajo. Además, se recomienda la utilización del optimizador *Adam* y la función de coste *entropía cruzada categórica* ya que han obtenido unos resultados óptimos. Por otro lado, los modelos MobileNetV2, InceptionV3 y DenseNet121, no se recomendarían para aplicarlo en este tipo de tarea debido a su falta de precisión en el reconocimiento de los gestos que se han probado.

Capítulo 7

Conclusiones

Capítulo 7

CONCLUSIONES

Contenidos

Este capítulo describe las conclusiones que se han obtenido del proceso de investigación realizado en esta tesis doctoral, donde también se analiza la consecución de los objetivos propuestos inicialmente en este trabajo. Esta tesis doctoral ha culminado en una serie de líneas de investigación que pueden ser continuadas en un futuro y se pueden consultar en la sección 7.2. Por último, en la sección 7.3 se enumeran las publicaciones derivadas de este trabajo de investigación.

7.1. Conclusiones

En el comienzo de este trabajo se utilizó Microsoft Kinect para interactuar de forma natural. Esta herramienta resultó muy útil ya que es capaz de hacer el seguimiento de los *joints* (20 o 26 según la versión) a lo largo del cuerpo con el apoyo de su Software Development Kit (SDK). En el estudio realizado con la primera versión de este dispositivo (ver capítulo 4) el objetivo era verificar que este sensor era apto para poder utilizarlo en entornos de interacción natural. Para conseguir este objetivo se desarrollaron diferentes métodos que permiten detectar el usuario que está usando la aplicación mediante la distancia de profundidad y los *joints* de los usuarios, así como filtrar los *joints* superpuestos para garantizar una mejor interacción y disminución de errores. Este sistema tiene implementado un sistema basado en reglas que, entre otras funciones, elegirá el tipo de interacción más adecuado: reconocimiento de gestos o seguimiento del movimiento.

La colaboración con el Centro de Educación Especial Princesa Sofía tuvo como resultado que se desarrollara un prototipo que pudiera ser usado por estudiantes con diversas capacidades mediante interacción natural y les ayudara en el desarrollo de ciertas habilidades, especialmente de índole físico. En los experimentos, se observó que en los primeros días de las pruebas, se obtuvieron peores resultados, independientemente de las características del usuario. La razón de esto es que los estudiantes tuvieron que adaptarse al uso de Kinect ya que es una forma de interacción diferente de lo que solían hacer. A pesar de estas circunstancias, todos los estudiantes fueron capaces de completar los diferentes ejercicios y llevar a cabo la ejecución de tareas en menos tiempo, y con una disminución del número de errores al final del período de evaluación. La contribución de esta parte del trabajo fue el desarrollo de un sistema que ha sido utilizado por estudiantes con diferentes tipos de discapacidad usando como medio de interacción el movimiento de su cuerpo con ayuda del sensor Microsoft Kinect v1. Además, este sistema tiene la característica específica de que permite al usuario interaccionar de manera autónoma y se ha desarrollado un algoritmo que detecta al usuario que está usando la aplicación e ignora el resto de elementos en su entorno para que no interfiera en la interacción.

Los resultados de este estudio muestran también que incluir un sistema con un tipo de interacción novedoso, dificulta que los estudiantes con necesidades especiales sigan el ritmo de la dinámica de la actividad y las instrucciones del profesor, pero si obtienen la estimulación adecuada (en este caso, la retroalimentación), los estudiantes pueden sentirse cómodos con este tipo de interacción. Otro aspecto que se ha observado es que es

recomendable que el alumno con diversidad funcional realice una actividad durante un período de tiempo hasta que se sientan cómodos con ella. Esto se debe a que la variación de ejercicios en la sesión puede convertirse en un problema para estos estudiantes, especialmente si no se les dio pautas adecuadas al principio de la actividad. Al proporcionar repetición, podrán desarrollar su confianza con el sistema y en ellos mismos con el fin de pasar a la siguiente etapa de la actividad o realizar diferentes tipos de tareas.

Después de estos experimentos, se decidió que una manera de mejorar el sistema sería que la interacción pudiera adaptarse a las características del individuo y por esta razón se creó el *modelo dispositivo-interacción* (ver capítulo 5, sección 5.2). En este estudio se puede interaccionar con el dispositivo Microsoft Kinect v2 o con el teclado y ratón, dependiendo de la acción que se vaya a realizar en el sistema. La interacción que se ha implementado con Microsoft Kinect v2 ha sido la detección del movimiento y el reconocimiento de gestos. Estos gestos son identificados mediante una máquina de estados finitos donde se detectan las diferentes posiciones que tendrá un gesto analizando la localización de los *joints* en cada momento. Los elementos principales que intervendrán en el proceso de adaptación son: el *modelo dispositivo-interacción*, un modelo de usuario basado en características y las reglas de adaptación. Este modelo de usuario, que se basa en el valor de la característica, mejora la comprensión e interpretación del sistema, ya que se ha integrado fácilmente con las reglas de adaptación. El modelo dispositivo-interacción aprovecha las características del sensor Microsoft Kinect v2 para detectar si está sentado o de pie o los brazos que puede utilizar para la interacción, entre otros. Este entorno cuenta con un sistema de actividades que se adapta a la actividad seleccionada en función de la información almacenada en el modelo de usuario y el *modelo dispositivo-interacción*. Este modelo denominado *modelo dispositivo-interacción* es la principal contribución de esta etapa donde tiene en consideración las caracterísiticas del dispositivo para adaptar la interacción del usuario cuando hace uso del sistema.

La evaluación de este sistema estaba integrada por varias etapas. En primer lugar, se realizó una evaluación de expertos y profesores, seguida de otra evaluación con 12 participantes, la cual fue segmentada según los diferentes tipos de discapacidad. En la segunda evaluación, se señaló que el tiempo de rendimiento y la cantidad de errores disminuyeron, lo que muestra que cada usuario, a pesar de sus discapacidades, había logrado completar las actividades propuestas sin ningún tipo de problema. Por último, se realizó una evaluación cualitativa mediante el Cuestionario de Experiencia de Usuario en el que participaron los tutores con el fin de determinar su experiencia con el sistema. Este método evalúa el sistema de acuerdo con seis escalas: atractivo, perspicuidad, eficiencia, confiabilidad, estimulación y novedad. En definitiva, los resultados mostraron que a los usuarios les gustó la aplicación (atractivo), que no tuvieron muchos problemas con la interacción a pesar de usar Kinect (confiabilidad) y pensaron que el sistema era original. Aunque el resto de las categorías obtuvieron puntuaciones más bajas, también fueron satisfactorias.

El inconveniente principal de las metodologías descritas anteriormente en relación a los objetivos de este estudio es que, aunque el dispositivo utilizado no es realmente caro, se obliga a los usuarios a comprar este sensor específico solo para interaccionar con el sistema. Además, este equipo (en el momento de escribir esta memoria) no está disponible para la venta y no hay soporte para las versiones Kinect v1 y Kinect v2

a pesar de que Microsoft ha lanzado Azure Kinect DK cuando estoy escribiendo esta tesis se vende por un precio más alto que los anteriores. Por consiguiente, se desarrolló un proceso de reconocimiento de gestos, donde la entrada fueron vídeos de la base de datos denominada *20BN-JESTER*, con el objetivo de que la mayoría de los usuarios de computadoras no tengan que comprar ningún gadget adicional ya que podrían utilizar una cámara web que es una herramienta muy común y está integrada en la mayoría de ordenadores portátiles.

Por último, se ha utilizado transferencia de aprendizaje con Deep Learning para el reconocimiento de gestos con las manos mediante una webcam (ver capítulo 6, sección 6.1). Los modelos utilizados para realizar esta transferencia de aprendizaje son una tecnología basada en la librería Keras, que ofrece la ventaja de que el tiempo de entrenamiento es menor que si se tuvieran que entrenar los modelos desde cero. Se ejecutaron diversas pruebas donde se modificaron ciertos parámetros como el modelo empleado o el optimizador hasta probar un total de 104 configuraciones distintas. De estas pruebas se ha extraído el siguiente conocimiento:

- Los modelos VGG16 y VGG19 eran los modelos que ofrecieron mejor rendimiento para el conjunto de datos de entrada.
- El optimizador Adam es más adecuado que el optimizador SGD.
- No se apreció cambio significativo al incrementar el conjunto de datos.
- El aumento del número de epochs mejoró el rendimiento.
- El hecho de aumentar el número de gestos no supuso variaciones considerables en los resultados.
- La función de pérdida de la entropía cruzada categórica obtuvo mejores resultados cuando el número de epochs era más reducido frente al error cuadrático medio.

Las métricas obtenidas de estos experimentos sirvieron para alimentar el sistema de lógica difusa mediante sus reglas de inferencia, lo que permitió hacer una clasificación de estos 104 experimentos para saber las configuraciones más apropiadas en el reconocimiento de gestos con las manos. Esta metodología es muy útil especialmente porque no hay un método matemático que ofrezca una orientación sobre cuáles son los parámetros adecuados (número de epochs, optimizador, función de pérdida, etcétera) para afrontar un problema. La única manera de resolver un problema determinado mediante estas técnicas es con el método de prueba y error. Este desarrollo contribuye a que si algún miembro de la comunidad científica está interesado en hacer un reconocimiento de gestos usando estos algoritmos tenga un punto de partida en lugar de asignar aleatoriamente la configuración necesaria para utilizar transferencia de aprendizaje con DL en este paradigma.

En relación a la consecución de los objetivos de esta tesis doctoral, se puede afirmar que:

- La realización de un sistema con interacción natural ha sido abordado en cada una de las fases de este trabajo, mediante el uso del sensor Microsoft Kinect como

medio de interacción para detectar el movimiento del usuario y el reconocimiento de gestos. De este modo, en la última fase también se ha realizado reconocimiento de gestos aplicando técnicas de Deep Learning donde los datos proceden de vídeos tomados por una webcam estándar. El uso de estos dispositivos tiene como consecuencia que el sistema se pueda utilizar con una inversión reducida, cubriendo así el objetivo que se perseguía de que el sistema fuera de bajo coste.

- El diseño de un sistema adaptable ha sido contemplado en la segunda fase de este proyecto con la creación del modelo dispositivo-interacción que permite adaptar la interacción del individuo con el sistema a partir de las características del dispositivo, el diseño de un módulo de usuario y un módulo de adaptación basado en un sistema basado en reglas.

- El diseño de un sistema inteligente ha sido abordado en cada una de las fases donde en las dos primeras se ha integrado un sistema basado en reglas para la toma de decisiones y en la última fase se ha aplicado la técnica de *Deep Learning*, transferencia de aprendizaje y lógica difusa para la consecución de la propuesta.

- El diseño de un sistema portable no ha sido posible completarlo aunque se tenía pensado utilizar la placa Nvidia Jetson Nano para el desarrollo de esta parte del trabajo. Por este motivo, se ha incluido en la sección siguiente como trabajo futuro.

7.2. Líneas de investigación abiertas

El trabajo realizado en esta tesis doctoral deja abierta una serie de líneas de investigación que se enuncian a continuación:

- El desarrollo de un sistema de interacción natural que integra Microsoft Kinect Azure como medio de interacción. Este dispositivo fue adquirido por el Departamento de Informática de la Universidad de Almería en marzo de 2021 y presenta algunas ventajas sobre las versiones previas: más ligera, mejor resolución angular, menor ruido y mejor precisión [Tölgyessy et al., 2021].

- La creación de un sistema de interacción que sea portable y se pueda acoplar en diversos escenarios. Para este fin se pueden utilizar placas de tamaño reducido como Nvidia Jetson Nano[1], Google Coral[2] o la popular Raspberry Pi[3]. El hecho de que estas placas tengan recursos limitados tiene como consecuencia que es necesario desarrollar los algoritmos adaptados a estas placas e incluso paralelizar para obtener una mejora del rendimiento.

- El desarrollo de un sistema que sea autoadaptativo con el objetivo de que adapte los gestos a las características de los usuarios. Un mismo gesto puede ser realizado de forma diferente por los usuario que utilizan el sistema, como por ejemplo, una

[1]Nvidia Jetson Nano - `https://developer.nvidia.com/embedded/jetson-nano-developer-kit`
[2]Google Coral - `https://coral.ai/`
[3]Raspberry Pi - `https://www.raspberrypi.org/`

mayor amplitud o un distinto ángulo al implementado por defecto. Para evitar estos inconvenientes se debería desarrollar un módulo de calibración donde los usuarios realicen el gesto implementado en el sistema para su control una serie de repeticiones para almacenar los parámetros requeridos con el fin de realizar esa adaptación de la interacción. Además, sería recomendable añadir otro módulo cuyo propósito sea ir aprendiendo la forma en la que el usuario realiza los gestos durante la interacción con el sistema.

- La elaboración de un sistema híbrido de Inteligencia Artificial para ofrecer un mejor rendimiento en el reconocimiento de gestos, el cual integre un algoritmo genético para mejorar el rendimiento de los algoritmos de Inteligencia Artificial implementados.
- La creación de un sistema multimodal que incorpore diversos modos de interacción, por ejemplo, reconocimiento de gestos y reconocimiento de voz. Por lo tanto, habrá que investigar las técnicas y algoritmos que existen para estos tipos de interacción adicionales y elegir los más adecuados para implementar según los objetivos establecidos. BCI es un tipo de interacción que resulta muy interesante con prometedoras expectativas pero que actualmente no está obteniendo buenos resultados en términos de interacción, no obstante, no se descarte su inclusión como parte del módulo de interacción del sistema multimodal.

7.3. Publicaciones

En esta sección se presentan las publicaciones que se han elaborado a partir de la investigación realizada en esta tesis doctoral, de las cuales cuatro son artículos en revistas de impacto, tres son aportaciones a congresos internacionales y tres presentaciones en las Jornadas de Doctorado en Informática de la Universidad de Almería. A continuación se muestran dichos trabajos ordenados por la fecha de publicación:

- Ojeda-Castelo, J. J., Piedra-Fernandez, J. A., Bernal-Bravo, C., & Iribarne-Martinez, L. (2016, September). Sign Communication for People with Disabilities Using Kinect Technology at Home. In 2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES) (pp. 1-2). IEEE.
- Piedra, J. A., Ojeda-Castelo, J. J., Quero-Valenzuela, F., & Piedra-Fdez, I. (2016, September). Virtual environment for the training of the hands in minimally invasive thoracic surgery. In 2016 8th International Conference on games and virtual worlds for serious applications (VS-Games) (pp. 1-4). IEEE.
- Osimani, C., Piedra-Fernandez, J. A., Ojeda-Castelo, J. J., & Iribarne, L. (2017, April). Hand posture recognition with standard webcam for natural interaction. In World Conference on Information Systems and Technologies (pp. 157-166). Springer, Cham.
- Castelo, J. J. O. (2018). Un Modelo Inteligente de Interacción Natural Autoadaptativo basado en Visión Artificial. I Jornadas de Doctorado en Informática, 69.

- Ojeda-Castelo, J. J., Piedra-Fernandez, J. A., Iribarne, L., & Bernal-Bravo, C. (2018). KiNEEt: application for learning and rehabilitation in special educational needs. Multimedia Tools and Applications, 77(18), 24013-24039.
- Castelo, J. J. O. (2019). El Modelo Dispositivo-Interacción y Machine Learning en Interacción Natural. II Jornadas de Doctorado en Informática, 72.
- Castelo, J. J. O. (2020). Interacción Natural: Últimas Publicaciones y el inicio con Deep Learning. III Jornadas de Doctorado en Informática, 48.
- Ojeda-Castelo, J. J., Piedra-Fernandez, J. A., & Iribarne, L. (2021). A device-interaction model for users with special needs. Multimedia Tools and Applications, 80(5), 6675-6710.
- Tejedor, A., Piedra-Fernandez, J. A., Ojeda-Castelo, J. J. & Iribarne, L. (2021). Ithaca. A tool for integrating fuzzy logic in Unity. In 10th International Congress on Advanced Applied Informatics.
- Ojeda-Castelo, J. J., María de las Mercedes Capobianco-Uriarte., Piedra-Fernandez, J. A., & Ayala, R. (2021). A Survey on Intelligent GestureRecognition Techniques. **Enviado pendiente de aceptación**.
- Ojeda-Castelo, J. J. & Piedra-Fernandez, J. A. (2021). Evaluation of Transfer Learning by means of Fuzzy Logic for Hand Gesture Recognition. **Enviado pendiente de aceptación**.

Anexo A

Experimentos de Deep Learning con Keras

Anexo A

EXPERIMIENTOS DE DEEP LEARNING CON KERAS

Contenidos

Este anexo contiene todos los metamodelos desarrollados en la metodología y que describen los distintos lenguajes específicos de dominio (DSL) que se utilizan.

A.1. Configuraciones de experimentos

En esta sección se van a reflejar los datos relevantes de los 104 experimentos realizados en Deep Learning mediante distintas tablas y gráficas.

A.2. Características de los experimentos

En este apartado se muestra la configuración de los experimentos con el valor de los parámetros que se han utilizado en cada uno de ellos.

Id	API	M	O	LR	E	NE	F	NG	NV
1	Keras	ResNet50	Adam	0.001	1.0	100	categorical_crossentropy	3	10
2	Keras	ResNet50	Adam	0.0001	1.0	100	categorical_crossentropy	3	10
3	Keras	ResNet50	SGD	-	-	100	categorical_crossentropy	3	10
4	Keras	ResNet50	SGD	-	-	100	mse	3	10
5	Keras	ResNet101	SGD	-	-	100	mse	3	10
6	Keras	ResNet152	SGD	-	-	100	mse	3	10
7	Keras	ResNet50V2	SGD	-	-	100	mse	3	10
8	Keras	ResNet101V2	SGD	-	-	100	mse	3	10
9	Keras	ResNet152V2	SGD	-	-	100	mse	3	10
10	Keras	ResNet50V2	SGD	-	-	100	categorical_crossentropy	3	10
11	Keras	ResNet50	Adam	0.001	1.0	100	mse	3	10
12	Keras	ResNet101	Adam	0.001	1.0	100	mse	3	10
13	Keras	ResNet152	Adam	0.001	1.0	100	mse	3	10
14	Keras	ResNet50V2	Adam	0.001	1.0	100	mse	3	10
15	Keras	ResNet101V2	Adam	0.001	1.0	100	mse	3	10
16	Keras	ResNet50	Adam	0.0001	1.0	100	mse	3	10
17	Keras	ResNet50	Adam	0.001	0.1	100	mse	3	10
18	Keras	ResNet101	Adam	0.001	0.1	100	mse	3	10
19	Keras	ResNet152	Adam	0.001	0.1	100	mse	3	10
20	Keras	ResNet50	Adam	0.01	0.1	100	mse	3	10
21	Keras	ResNet101	Adam	0.01	0.1	100	mse	3	10
22	Keras	VGG16	SGD	-	-	100	mse	3	10
23	Keras	VGG16	SGD	-	-	100	categorical_crossentropy	3	10
24	Keras	VGG16	Adam	0.001	0.1	100	mse	3	10
25	Keras	VGG16	Adam	0.001	0.1	100	categorical_crossentropy	3	10
26	Keras	VGG19	Adam	0.001	0.1	100	categorical_crossentropy	3	10
27	Keras	VGG19	SGD	-	-	100	mse	3	10
28	Keras	VGG19	Adam	0.001	0.1	100	mse	3	10
29	Keras	VGG19	SGD	-	-	100	categorical_crossentropy	3	10
30	Keras	InceptionV3	SGD	-	-	100	mse	3	10
31	Keras	InceptionV3	Adam	0.001	0.1	100	mse	3	10
32	Keras	InceptionResNetV2	SGD	-	-	100	mse	3	10
33	Keras	InceptionResNetV2	Adam	0.001	0.1	100	mse	3	10
34	Keras	MobileNetV2	SGD	-	-	100	mse	3	10
35	Keras	MobileNetV2	Adam	0.001	0.1	100	mse	3	10
36	Keras	DenseNet121	SGD	-	-	100	mse	3	10
37	Keras	DenseNet121	Adam	0.001	0.1	100	mse	3	10
38	Keras	DenseNet169	SGD	-	-	100	mse	3	10
39	Keras	DenseNet169	Adam	0.001	0.1	100	mse	3	10
40	Keras	DenseNet201	SGD	-	-	100	mse	3	10

Tabla A.1: Configuración de experimentos (Parte I). M: Modelo / O: Optimizador / LR: Learning rate / E: Epsilon / NE: Nº epochs / F: Función de coste / NG: Nº gestos / NV: Nº vídeos

Id	API	M	O	LR	E	NE	F	NG	NV
41	Keras	DenseNet201	Adam	0.001	0.1	100	mse	3	10
42	Keras	NasNetLarge	SGD	-	-	100	mse	3	10
43	Keras	NasNetLarge	Adam	0.001	0.1	100	mse	3	10
44	Keras	VGG16	Adam	0.001	0.1	150	categorical_crossentropy	3	10
45	Keras	VGG16	Adam	0.001	0.1	150	mse	3	10
46	Keras	ResNet152	SGD	-	-	150	mse	3	10
47	Keras	VGG16	Adam	0.001	0.1	200	categorical_crossentropy	3	10
48	Keras	VGG19	Adam	0.001	0.1	200	categorical_crossentropy	3	10
49	Keras	ResNet50	Adam	0.001	0.1	100	mse	5	10
50	Keras	ResNet101	Adam	0.001	0.1	100	mse	5	10
51	Keras	ResNet152	Adam	0.001	0.1	100	mse	5	10
52	Keras	VGG16	Adam	0.001	0.1	100	mse	5	10
53	Keras	VGG19	Adam	0.001	0.1	100	mse	5	10
54	Keras	VGG16	Adam	0.001	0.1	100	categorical_crossentropy	5	10
55	Keras	VGG19	Adam	0.001	0.1	100	categorical_crossentropy	5	10
56	Keras	ResNet50	SGD	-	-	100	mse	5	10
57	Keras	ResNet50	Adam	0.001	0.1	200	mse	5	10
58	Keras	ResNet101	Adam	0.001	0.1	200	mse	5	10
59	Keras	ResNet152	Adam	0.001	0.1	200	mse	5	10
60	Keras	VGG16	Adam	0.001	0.1	200	mse	5	10
61	Keras	VGG19	Adam	0.001	0.1	200	mse	5	10
62	Keras	VGG16	Adam	0.001	0.1	200	categorical_crossentropy	5	10
63	Keras	VGG19	Adam	0.001	0.1	200	categorical_crossentropy	5	10
64	Keras	ResNet50	Adam	0.001	0.1	100	mse	8	10
65	Keras	ResNet101	Adam	0.001	0.1	100	mse	8	10
66	Keras	ResNet152	Adam	0.001	0.1	100	mse	8	10
67	Keras	VGG16	Adam	0.001	0.1	100	mse	8	10
68	Keras	VGG19	Adam	0.001	0.1	100	mse	8	10
69	Keras	VGG16	Adam	0.001	0.1	100	categorical_crossentropy	8	10
70	Keras	VGG19	Adam	0.001	0.1	100	categorical_crossenropy	8	10
71	Keras	ResNet50	SGD	-	-	100	mse	8	10
72	Keras	ResNet50	Adam	0.001	0.1	200	mse	8	10
73	Keras	ResNet101	Adam	0.001	0.1	200	mse	8	10
74	Keras	ResNet152	Adam	0.001	0.1	200	mse	8	10
75	Keras	VGG16	Adam	0.001	0.1	200	mse	8	10
76	Keras	VGG19	Adam	0.001	0.1	200	mse	8	10
77	Keras	VGG16	Adam	0.001	0.1	200	categorical_crossentropy	8	10
78	Keras	VGG19	Adam	0.001	0.1	200	categorical_crossentropy	8	10
79	Keras	ResNet50	Adam	0.001	0.1	100	mse	10	10
80	Keras	ResNet101	Adam	0.001	0.1	100	mse	10	10

Tabla A.2: Configuración de experimentos (Parte II). M: Modelo / O: Optimizador / LR: Learning rate / E: Epsilon / NE: Nº epochs / F: Función de coste / NG: Nº gestos / NV: Nº vídeos

Id	API	M	O	LR	E	NE	F	NG	NV
81	Keras	ResNet152	Adam	0.001	0.1	100	mse	10	10
82	Keras	VGG16	Adam	0.001	0.1	100	mse	10	10
83	Keras	VGG19	Adam	0.001	0.1	100	mse	10	10
84	Keras	VGG16	Adam	0.001	0.1	100	categorical_crossentropy	10	10
85	Keras	VGG19	Adam	0.001	0.1	100	categorical_crossentropy	10	10
86	Keras	ResNet50	Adam	0.001	0.1	200	mse	10	10
87	Keras	ResNet101	Adam	0.001	0.1	200	mse	10	10
88	Keras	ResNet152	Adam	0.001	0.1	200	mse	10	10
89	Keras	VGG16	Adam	0.001	0.1	200	mse	10	10
90	Keras	VGG19	Adam	0.001	0.1	200	mse	10	10
91	Keras	VGG16	Adam	0.001	0.1	200	categorical_crossentropy	10	10
92	Keras	VGG19	Adam	0.001	0.1	200	categorical_crossentropy	10	10
93	Keras	VGG16	Adam	0.001	0.1	30	categorical_crossentropy	10	17
94	Keras	VGG16	Adam	0.001	0.1	50	categorical_crossentropy	10	17
95	Keras	VGG16	Adam	0.001	0.1	50	mse	10	17
96	Keras	VGG19	Adam	0.001	0.1	50	categorical_crossentropy	10	17
97	Keras	VGG19	Adam	0.001	0.1	100	categorical_crossentropy	10	17
98	Keras	VGG16	Adam	0.001	0.1	70	categorical_crossentropy	10	17
99	Keras	VGG16	Adam	0.001	0.1	100	categorical_crossentropy	10	17
100	Keras	VGG16	Adam	0.001	0.1	30	mse	10	17
101	Keras	VGG16	Adam	0.001	0.1	70	mse	10	17
102	Keras	VGG16	Adam	0.001	0.1	100	mse	10	17
103	Keras	VGG19	Adam	0.001	0.1	30	categorical_crossentropy	10	17
104	Keras	VGG19	Adam	0.001	0.1	70	categorical_crossentropy	10	17

Tabla A.3: Configuración de experimentos (Parte III). M: Modelo / O: Optimizador / LR: Learning rate / E: Epsilon / NE: Nº epochs / F: Función de coste / NG: Nº gestos / NV: Nº vídeos

A.2.1. Métricas de los experimentos

En esta sección se muestra los valores de las métricas que se han obtenido en los experimentos para cada gesto que integra cada uno de los experimentos.

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
	stop_sign	1.0	0.3	0.46	90.0	
1	swipe_left	0.92	0.89	0.9	123.0	0.738
	thumb_up	0.55	0.99	0.71	86.0	
	stop_sign	0.86	0.53	0.66	90.0	
2	swipe_left	0.79	0.91	0.85	123.0	0.723
	thumb_up	0.61	0.72	0.66	86.0	
	stop_sign	0.85	0.58	0.69	90.0	
3	swipe_left	0.83	0.93	0.88	123.0	0.791
	thumb_up	0.79	0.92	0.85	86.0	
	stop_sign	0.82	0.72	0.77	90.0	
4	swipe_left	0.82	0.95	0.88	123.0	0.757
	thumb_up	0.88	0.79	0.83	86.0	
	stop_sign	0.82	0.51	0.63	90.0	
5	swipe_left	0.93	0.93	0.93	123.0	0.678
	thumb_up	0.63	0.88	0.74	86.0	
	stop_sign	0.96	0.56	0.7	90.0	
6	swipe_left	0.78	0.81	0.79	88.0	0.818
	thumb_up	0.64	0.91	0.75	86.0	
	stop_sign	0.06	0.04	0.05	90.0	
7	swipe_left	0.0	0.0	0.0	123.0	0.378
	thumb_up	0.26	0.72	0.39	86.0	
	stop_sign	0.0	0.0	0.0	90.0	
8	swipe_left	0.35	0.45	0.39	123.0	0.310
	thumb_up	0.31	0.51	0.38	86.0	
	stop_sign	0.0	0.0	0.0	90.0	
9	swipe_left	0.41	1.0	0.58	123.0	0.325
	thumb_up	0.0	0.0	0.0	86.0	
	stop_sign	0.31	1.0	0.48	90.0	
10	swipe_left	0.0	0.0	0.0	123.0	0.333
	thumb_up	0.0	0.0	0.0	86.0	
	stop_sign	1.0	0.22	0.36	90.0	
11	swipe_left	0.68	0.87	0.76	123.0	0.734
	thumb_up	0.61	0.86	0.71	86.0	
	stop_sign	0.88	0.48	0.62	90.0	
12	swipe_left	0.75	0.89	0.81	123.0	0.598
	thumb_up	0.6	0.72	0.66	86.0	
	stop_sign	0.91	0.48	0.63	90.0	
13	swipe_left	0.69	0.99	0.81	123.0	0.852
	thumb_up	0.81	0.7	0.75	86.0	
	stop_sign	0.31	0.97	0.46	90.0	
14	swipe_left	0.29	0.03	0.06	123.0	0.337
	thumb_up	0.0	0.0	0.0	86.0	

Tabla A.4: Métricas de los experimentos (Parte I).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
	stop_sign	0.36	0.97	0.52	90.0	
15	swiping_left	0.23	0.03	0.06	88.0	0.303
	thumb_up	0.0	0.0	0.0	86.0	
	stop_sign	1.0	0.03	0.06	90.0	
16	swipe_left	0.46	0.54	0.49	123.0	0.329
	thumb_up	0.44	0.78	0.56	86.0	
	stop_sign	0.95	0.84	0.89	90.0	
17	swipe_left	1.0	0.81	0.9	123.0	0.863
	thumb_up	0.7	0.97	0.81	86.0	
	stop_sign	1.0	0.46	0.63	90.0	
18	swipe_left	0.94	0.97	0.95	123.0	0.742
	thumb_up	0.66	1.0	0.79	86.0	
	stop_sign	1.0	0.77	0.87	90.0	
19	swipe_left	0.96	0.91	0.93	123.0	0.867
	thumb_up	0.73	0.97	0.83	86.0	
	stop_sign	0.98	0.56	0.71	90.0	
20	swiping_left	0.88	0.72	0.79	88.0	0.950
	thumb_up	0.6	0.99	0.75	86.0	
	stop_sign	1.0	0.59	0.74	90.0	
21	swipe_left	0.84	1.0	0.91	123.0	0.768
	thumb_up	0.86	1.0	0.92	86.0	
	stop_sign	0.93	0.97	0.95	90.0	
22	swiping_left	0.88	0.98	0.92	88.0	0.984
	thumb_up	0.96	0.8	0.87	86.0	
	stop_sign	1.0	1.0	1.0	90.0	
23	swiping_left	1.0	1.0	1.0	123.0	1.0
	thumb_up	1.0	1.0	1.0	86.0	
	stop_sign	1.0	1.0	1.0	90.0	
24	swipe_left	1.0	1.0	1.0	123.0	1.0
	thumb_up	1.0	1.0	1.0	86.0	
	stop_sign 1.0	1.0	1.0	90.0		
25	swipe_left	1.0	1.0	1.0	123.0	1.0
	thumb_up	1.0	1.0	1.0	86.0	
	stop_sign	1.0	1.0	1.0	90.0	
26	swipe_left	1.0	1.0	1.0	123.0	1.0
	thumb_up	1.0	1.0	1.0	86.0	
	stop_sign	0.99	0.97	0.98	90.0	
27	swipe_left	0.98	0.99	0.98	123.0	0.928
	thumb_up	1.0	1.0	1.0	86.0	
	stop_sign	1.0	1.0	1.0	90.0	
28	swipe_left	1.0	1.0	1.0	123.0	1.0
	thumb_up	1.0	1.0	1.0	86.0	

Tabla A.5: Métricas de los experimentos (Parte II).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
	stop_sign	1.0	1.0	1.0	90.0	
29	swiping_left	1.0	1.0	1.0	88.0	1.0
	thumb_up	1.0	1.0	1.0	86.0	
	stop_sign	0.02	0.01	0.02	90.0	
30	swiping_left	0.32	0.81	0.46	88.0	0.284
	thumb_up	0.0	0.0	0.0	86.0	
	stop_sign	0.32	0.51	0.39	90.0	
31	swiping_left	0.26	0.28	0.27	88.0	0.227
	thumb_up	0.24	0.06	0.09	86.0	
	stop_sign	0.0	0.0	0.0	90.0	
32	swiping_left	0.3	1.0	0.5	88.0	0.340
	thumb_up	0.0	0.0	0.0	86.0	
	stop_sign	0.34	1.0	0.51	90.0	
33	swiping_left	0.0	0.0	0.0	88.0	0.363
	thumb_up	0.0	0.0	0.0	86.0	
	stop_sign	0.36	0.39	0.38	90.0	
34	swiping_left	1.0	0.08	0.15	88.0	0.303
	thumb_up	0.34	0.63	0.44	86.0	
	stop_sign	0.2	0.11	0.14	90.0	
35	swiping_left	0.4	0.11	0.18	88.0	0.268
	thumb_up	0.29	0.63	0.39	86.0	
	stop_sign	0.0	0.0	0.0	90.0	
36	swiping_left	0.33	1.0	0.5	88.0	0.337
	thumb_up	0.0	0.0	0.0	86.0	
	stop_sign	0.38	0.03	0.06	90.0	
37	swiping_left	1.0	0.01	0.02	88.0	0.231
	thumb_up	0.32	0.94	0.48	86.0	
	stop_sign	0.0	0.0	0.0	90.0	
38	swipe_left	0.4	0.92	0.56	123.0	0.333
	thumb_up	0.37	0.08	0.13	86.0	
	stop_sign	0.33	0.64	0.43	90.0	
39	swipe_left	1.0	0.07	0.14	123.0	0.310
	thumb_up	0.29	0.37	0.32	86.0	
	stop_sign	0.35	0.99	0.52	90.0	
40	swiping_left	1.0	0.03	0.07	88.0	0.287
	thumb_up	0.2	0.02	0.04	86.0	
	stop_sign	0.38	0.06	0.1	90.0	
41	swiping_left	1.0	0.11	0.2	88.0	0.310
	thumb_up	0.34	0.94	0.5	86.0	
	stop_sign	0.0	0.0	0.0	90.0	
42	swiping_left	0.18	0.05	0.07	88.0	0.287
	thumb_up	0.32	0.87	0.47	86.0	

Tabla A.6: Métricas de los experimentos (Parte III).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
	stop_sign	0.34	1.0	0.51	90.0	
43	swiping_left	0.0	0.0	0.0	88.0	0.321
	thumb_up	0.0	0.0	0.0	86.0	
	stop_sign	1.0	1.0	1.0	90.0	
44	swiping_left	1.0	1.0	1.0	88.0	1.0
	thumb_up	1.0	1.0	1.0	86.0	
	stop_sign	1.0	1.0	1.0	90.0	
45	swiping_left	1.0	1.0	1.0	88.0	1.0
	thumb_up	1.0	1.0	1.0	86.0	
	stop_sign	0.95	0.6	0.73	90.0	
46	swiping_left	0.7	0.91	0.79	88.0	0.780
	thumb_up	0.81	0.87	0.84	86.0	
	stop_sign	1.0	1.0	1.0	90.0	
47	swiping_left	1.0	1.0	1.0	88.0	1.0
	thumb_up	1.0	1.0	1.0	86.0	
	stop_sign	1.0	1.0	1.0	90.0	
48	swiping_left	1.0	1.0	1.0	88.0	1.0
	thumb_up	1.0	1.0	1.0	86.0	
	slide_2_fingers_right	1.0	0.17	0.29	89.0	
	stop_sign	1.0	0.5	0.67	90.0	
49	swipe_down	1.0	0.01	0.02	87.0	0.530
	swipe_left	0.85	0.67	0.75	87.0	
	thumb_up	0.28	1.0	0.43	86.0	
	slide_2_fingers_right	1.0	0.12	0.22	89.0	
	stop_sign	1.0	0.32	0.49	90.0	
50	swipe_down	0.67	0.02	0.04	87.0	0.514
	swipe_left	0.69	0.7	0.7	88.0	
	thumb_up	0.28	1.0	0.44	86.0	
	slide_2_fingers_right	1.0	0.28	0.44	89.0	
	stop_sign	0.88	0.84	0.86	90.0	
51	swipe_down	1.0	0.03	0.07	87.0	0.674
	swipe_left	0.75	0.85	0.8	88.0	
	thumb_up	0.38	1.0	0.55	86.0	
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	stop_sign	1.0	1.0	1.0	90.0	
52	swipe_down	1.0	1.0	1.0	87.0	0.995
	swipe_left	1.0	1.0	1.0	87.0	
	thumb_up	1.0	1.0	1.0	86.0	

Tabla A.7: Métricas de los experimentos (Parte IV).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
	slide_2_fingers_right	1.0	0.99	0.99	89.0	
	stop_sign	0.99	1.0	0.99	90.0	
53	swipe_down	1.0	1.0	1.0	87.0	0.995
	swipe_left	1.0	0.99	0.99	88.0	
	thumb_up	0.99	1.0	0.99	86.0	
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	stop_sign	1.0	1.0	1.0	90.0	
54	swipe_down	1.0	1.0	1.0	87.0	1.0
	swipe_left	1.0	1.0	1.0	87.0	
	thumb_up	1.0	1.0	1.0	86.0	
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	stop_sign	1.0	1.0	1.0	90.0	
55	swipe_down	1.0	1.0	1.0	87.0	1.0
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	
	slide_2_fingers_right	0.0	0.0	0.0	89.0	
	stop_sign	0.24	0.16	0.19	90.0	
56	swipe_down	0.0	0.0	0.0	87.0	0.298
	swipe_left	0.44	0.29	0.35	87.0	
	thumb_up	0.23	0.84	0.36	86.0	
	slide_2_fingers_right	1.0	0.26	0.41	89.0	
	stop_sign	1.0	0.44	0.62	90.0	
57	swipe_down	1.0	0.1	0.19	87.0	0.526
	swipe_left	0.83	0.69	0.75	87.0	
	thumb_up	0.29	1.0	0.45	86.0	
	slide_2_fingers_right	1.0	0.25	0.4	89.0	
	stop_sign	0.93	0.46	0.61	90.0	
58	swipe_down	1.0	0.07	0.13	87.0	0.466
	swipe_left	0.55	0.99	0.71	88.0	
	thumb_up	0.4	0.98	0.57	86.0	
	slide_2_fingers_right	0.96	0.53	0.68	89.0	
	stop_sign	0.95	0.89	0.92	90.0	
59	swipe_down	1.0	0.21	0.34	87.0	0.740
	swipe_left	0.69	0.84	0.76	87.0	
	thumb_up	0.47	0.99	0.63	86.0	
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	stop_sign	1.0	1.0	1.0	90.0	
60	swipe_down	1.0	1.0	1.0	87.0	1.0
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	

Tabla A.8: Métricas de los experimentos (Parte V).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
61	slide_2_fingers_right	1.0	0.99	0.99	89.0	1.0
	stop_sign	0.99	1.0	0.99	90.0	
	swipe_down	1.0	1.0	1.0	87.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	
62	slide_2_fingers_right	1.0	1.0	1.0	89.0	1.0
	stop_sign	1.0	1.0	1.0	90.0	
	swipe_down	1.0	1.0	1.0	87.0	
	swipe_left	1.0	1.0	1.0	87.0	
	thumb_up	1.0	1.0	1.0	86.0	
63	slide_2_fingers_right	1.0	1.0	1.0	89.0	1.0
	stop_sign	1.0	1.0	1.0	90.0	
	swipe_down	1.0	1.0	1.0	87.0	
	swipe_left	1.0	1.0	1.0	87.0	
	thumb_up	1.0	1.0	1.0	86.0	
64	pull_hand_in	0.79	0.34	0.48	90.0	0.514
	slide_2_fingers_right	1.0	0.04	0.09	89.0	
	slide_2_fingers_up	0.44	0.83	0.57	88.0	
	stop_sign	0.87	0.29	0.43	90.0	
	swipe_down	1.0	0.09	0.17	86.0	
	swipe_left	0.91	0.35	0.51	88.0	
	thumb_up	0.21	1.0	0.34	86.0	
	zoom_in_with_2_fingers	1.0	0.09	0.16	91.0	
65	pull_hand_in	0.69	0.61	0.65	90.0	0.426
	slide_2_fingers_right	0.0	0.0	0.0	89.0	
	slide_2_fingers_up	0.62	0.62	0.62	88.0	
	stop_sign	0.74	0.22	0.34	90.0	
	swipe_down	1.0	0.06	0.11	86.0	
	swipe_left	0.55	0.59	0.57	88.0	
	thumb_up	0.21	0.97	0.35	86.0	
	zoom_in_with_2_fingers	1.0	0.22	0.36	91.0	

Tabla A.9: Métricas de los experimentos (Parte VI).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
66	pull_hand_in	0.43	0.86	0.57	90.0	0.644
	slide_2_fingers_right	0.91	0.34	0.49	89.0	
	slide_2_fingers_up	0.75	0.43	0.55	88.0	
	stop_sign	1.0	0.44	0.62	90.0	
	swipe_down	1.0	0.06	0.11	86.0	
	swipe_left	0.62	0.77	0.69	88.0	
	thumb_up	0.35	0.93	0.51	86.0	
	zoom_in_with_2_fingers	0.98	0.63	0.77	91.0	
67	pull_hand_in	0.89	0.93	0.91	90.0	0.984
	slide_2_fingers_right	0.94	1.0	0.97	89.0	
	slide_2_fingers_up	0.93	0.92	0.93	88.0	
	stop_sign	0.89	1.0	0.94	90.0	
	swipe_down	1.0	0.85	0.92	86.0	
	swipe_left	0.98	1.0	0.99	88.0	
	thumb_up	1.0	0.88	0.94	86.0	
	zoom_in_with_2_fingers	0.99	1.0	0.99	91.0	
68	pull_hand_in	0.99	0.87	0.92	90.0	0.957
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	slide_2_fingers_up	0.94	0.94	0.94	88.0	
	stop_sign	0.85	0.99	0.91	90.0	
	swipe_down	1.0	0.99	0.99	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	0.97	0.98	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
69	pull_hand_in	1.0	1.0	1.0	90.0	1.0
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	slide_2_fingers_up	1.0	1.0	1.0	88.0	
	stop_sign	1.0	1.0	1.0	90.0	
	swipe_down	1.0	1.0	1.0	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
70	pull_hand_in	1.0	1.0	1.0	90.0	1.0
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	slide_2_fingers_up	1.0	1.0	1.0	88.0	
	stop_sign	1.0	1.0	1.0	90.0	
	swipe_down	1.0	1.0	1.0	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	

Tabla A.10: Métricas de los experimentos (Parte VII).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
71	pull_hand_in	0.25	0.44	0.32	90.0	0.180
	slide_2_fingers_right	0.22	0.21	0.21	89.0	
	slide_2_fingers_up	0.33	0.22	0.26	88.0	
	stop_sign	0.33	0.29	0.31	90.0	
	swipe_down	0.55	0.13	0.21	86.0	
	swipe_left	0.22	0.52	0.31	88.0	
	thumb_up	0.0	0.0	0.0	86.0	
	zoom_in_with_2_fingers	0.4	0.42	0.41	91.0	
72	0.81	0.47	0.59	90.0		0.490
	slide_2_fingers_right	1.0	0.13	0.24	89.0	
	slide_2_fingers_up	0.49	0.97	0.65	88.0	
	stop_sign	0.91	0.68	0.78	90.0	
	swipe_down	1.0	0.01	0.02	86.0	
	swipe_left	0.77	0.69	0.73	88.0	
	thumb_up	0.27	1.0	0.43	86.0	
	zoom_in_with_2_fingers	1.0	0.12	0.22	91.0	
73	pull_hand_in	0.97	0.39	0.56	90.0	0.488
	slide_2_fingers_right	1.0	0.11	0.2	89.0	
	slide_2_fingers_up	0.67	0.66	0.66	88.0	
	stop_sign	0.91	0.43	0.59	90.0	
	swipe_down	1.0	0.02	0.05	86.0	
	swipe_left	0.67	0.91	0.77	88.0	
	thumb_up	0.22	1.0	0.36	86.0	
	zoom_in_with_2_fingers	1.0	0.23	0.38	91.0	
74	pull_hand_in	0.67	0.83	0.74	90.0	0.672
	slide_2_fingers_right	1.0	0.38	0.55	89.0	
	slide_2_fingers_up	0.77	0.98	0.86	88.0	
	stop_sign	0.87	0.74	0.8	90.0	
	swipe_down	1.0	0.17	0.3	86.0	
	swipe_left	0.78	0.84	0.81	88.0	
	thumb_up	0.41	1.0	0.59	86.0	
	zoom_in_with_2_fingers	1.0	0.6	0.75	91.0	
75	pull_hand_in	1.0	0.99	0.99	90.0	0.992
	slide_2_fingers_right	0.99	1.0	0.99	89.0	
	slide_2_fingers_up	0.98	0.99	0.98	88.0	
	stop_sign	0.99	0.99	0.99	90.0	
	swipe_down	1.0	1.0	1.0	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	0.99	0.99	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	

Tabla A.11: Métricas de los experimentos (Parte VIII).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
76	pull_hand_in	0.99	0.96	0.97	90.0	0.975
	slide_2_fingers_right	0.99	0.99	0.99	89.0	
	slide_2_fingers_up	1.0	0.94	0.97	88.0	
	stop_sign	0.91	0.99	0.95	90.0	
	swipe_down	1.0	1.0	1.0	86.0	
	swipe_left	0.95	1.0	0.97	88.0	
	thumb_up	1.0	0.93	0.96	86.0	
	zoom_in_with_2_fingers	0.99	1.0	0.99	91.0	
77	pull_hand_in	1.0	1.0	1.0	90.0	1.0
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	slide_2_fingers_up	1.0	1.0	1.0	88.0	
	stop_sign	1.0	1.0	1.0	90.0	
	swipe_down	1.0	1.0	1.0	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
78	pull_hand_in	1.0	1.0	1.0	90.0	1.0
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	slide_2_fingers_up	1.0	1.0	1.0	88.0	
	stop_sign	1.0	1.0	1.0	90.0	
	swipe_down	1.0	1.0	1.0	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
79	pull_hand_in	0.68	0.44	0.54	90.0	0.403
	slide_2_fingers_right	0.0	0.0	0.0	89.0	
	slide_2_fingers_up	0.41	0.8	0.54	88.0	
	stop_sign	0.77	0.62	0.69	90.0	
	swipe_down	1.0	0.12	0.21	86.0	
	swipe_left	0.69	0.6	0.64	88.0	
	thumb_up	0.19	0.85	0.31	86.0	
	zoom_in_with_2_fingers	1.0	0.05	0.1	91.0	
	zoom_in_with_full_hand	1.0	0.06	0.11	87.0	
	zoom_out_with_2_fingers	0.58	0.66	0.62	89.0	

Tabla A.12: Métricas de los experimentos (Parte IX).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
80	pull_hand_in	0.49	0.19	0.27	90.0	0.410
	slide_2_fingers_right	1.0	0.01	0.02	89.0	
	slide_2_fingers_up	0.93	0.15	0.25	88.0	
	stop_sign	0.61	0.16	0.25	90.0	
	swipe_down	1.0	0.09	0.17	86.0	
	swipe_left	0.61	0.7	0.66	88.0	
	thumb_up	0.21	0.94	0.35	86.0	
	zoom_in_with_2_fingers	1.0	0.13	0.23	91.0	
	zoom_in_with_full_hand	1.0	0.09	0.16	90.0	
	zoom_out_with_2_fingers	0.25	0.85	0.39	88.0	
81	pull_hand_in	0.42	0.72	0.53	90.0	0.554
	slide_2_fingers_right	1.0	0.44	0.61	89.0	
	slide_2_fingers_up	0.79	0.22	0.34	88.0	
	stop_sign	0.75	0.56	0.64	90.0	
	swipe_down	1.0	0.07	0.13	86.0	
	swipe_left	0.37	0.95	0.53	88.0	
	thumb_up	0.46	0.72	0.56	86.0	
	zoom_in_with_2_fingers	1.0	0.52	0.68	91.0	
	zoom_in_with_full_hand	1.0	0.18	0.31	87.0	
	zoom_out_with_2_fingers	0.45	0.83	0.58	89.0	
82	pull_hand_in	0.82	0.97	0.89	90.0	0.892
	slide_2_fingers_right	0.99	1.0	0.99	89.0	
	slide_2_fingers_up	0.95	0.95	0.95	88.0	
	stop_sign	0.7	0.97	0.81	90.0	
	swipe_down	0.98	0.64	0.77	86.0	
	swipe_left	0.92	0.88	0.9	88.0	
	thumb_up	0.97	0.9	0.93	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
	zoom_in_with_full_hand	0.94	0.88	0.91	90.0	
	zoom_out_with_2_fingers	0.89	0.85	0.87	88.0	
83	pull_hand_in	0.92	0.81	0.86	90.0	0.894
	slide_2_fingers_right	0.91	1.0	0.95	89.0	
	slide_2_fingers_up	0.99	0.81	0.89	88.0	
	stop_sign	0.73	0.99	0.84	90.0	
	swipe_down	0.96	0.78	0.86	86.0	
	swipe_left	0.98	0.97	0.97	88.0	
	thumb_up	0.98	0.94	0.96	86.0	
	zoom_in_with_2_fingers	0.98	1.0	0.99	91.0	
	zoom_in_with_full_hand	1.0	0.84	0.92	90.0	
	zoom_out_with_2_fingers	0.78	0.94	0.86	88.0	

Tabla A.13: Métricas de los experimentos (Parte X).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
84	pull_hand_in	1.0	1.0	1.0	90.0	1.0
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	slide_2_fingers_up	1.0	1.0	1.0	88.0	
	stop_sign	1.0	1.0	1.0	90.0	
	swipe_down	1.0	1.0	1.0	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
	zoom_in_with_full_hand	1.0	1.0	1.0	87.0	
	zoom_out_with_2_fingers	1.0	1.0	1.0	89.0	
85	pull_hand_in	0.99	1.0	0.99	90.0	1.0
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	slide_2_fingers_up	1.0	1.0	1.0	88.0	
	stop_sign	1.0	1.0	1.0	90.0	
	swipe_down	1.0	1.0	1.0	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
	zoom_in_with_full_hand	1.0	1.0	1.0	90.0	
	zoom_out_with_2_fingers	1.0	0.99	0.99	88.0	
86	pull_hand_in	0.67	0.46	0.54	90.0	0.476
	slide_2_fingers_right	1.0	0.03	0.07	89.0	
	slide_2_fingers_up	0.48	0.9	0.62	88.0	
	stop_sign	0.9	0.73	0.81	90.0	
	swipe_down	1.0	0.12	0.21	86.0	
	swipe_left	0.71	0.64	0.67	88.0	
	thumb_up	0.2	0.94	0.33	86.0	
	zoom_in_with_2_fingers	1.0	0.1	0.18	91.0	
	zoom_in_with_full_hand	1.0	0.01	0.02	87.0	
	zoom_out_with_2_fingers	0.73	0.64	0.68	89.0	
87	pull_hand_in	0.79	0.33	0.47	90.0	0.436
	slide_2_fingers_right	1.0	0.04	0.09	89.0	
	slide_2_fingers_up	0.45	0.67	0.54	88.0	
	stop_sign	0.84	0.18	0.29	90.0	
	swipe_down	0.0	0.0	0.0	86.0	
	swipe_left	0.84	0.49	0.62	88.0	
	thumb_up	0.22	0.95	0.35	86.0	
	zoom_in_with_2_fingers	0.97	0.33	0.49	91.0	
	zoom_in_with_full_hand	1.0	0.04	0.09	90.0	
	zoom_out_with_2_fingers	0.34	0.89	0.5	88.0	

Tabla A.14: Métricas de los experimentos (Parte XI).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
88	pull_hand_in	0.62	0.81	0.7	90.0	0.516
	slide_2_fingers_right	1.0	0.37	0.54	89.0	
	slide_2_fingers_up	0.65	0.68	0.67	88.0	
	stop_sign	0.89	0.53	0.67	90.0	
	swipe_down	1.0	0.1	0.19	86.0	
	swipe_left	0.76	0.77	0.77	88.0	
	thumb_up	0.47	0.87	0.61	86.0	
	zoom_in_with_2_fingers	1.0	0.46	0.63	91.0	
	zoom_in_with_full_hand	1.0	0.3	0.46	87.0	
	zoom_out_with_2_fingers	0.32	0.94	0.48	89.0	
89	pull_hand_in	0.98	0.98	0.98	90.0	0.959
	slide_2_fingers_right	0.99	1.0	0.99	89.0	
	slide_2_fingers_up	0.99	1.0	0.99	88.0	
	stop_sign	0.97	0.99	0.98	90.0	
	swipe_down	0.99	0.98	0.98	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
	zoom_in_with_full_hand	1.0	1.0	1.0	90.0	
	zoom_out_with_2_fingers	0.98	0.94	0.96	88.0	
90	pull_hand_in	1.0	0.99	0.99	90.0	0.934
	slide_2_fingers_right	0.99	1.0	0.99	89.0	
	slide_2_fingers_up	0.99	1.0	0.99	88.0	
	stop_sign	0.98	0.99	0.98	90.0	
	swipe_down	1.0	0.99	0.99	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	0.95	0.98	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
	zoom_in_with_full_hand	1.0	1.0	1.0	90.0	
	zoom_out_with_2_fingers	0.95	0.98	0.96	88.0	
91	pull_hand_in	1.0	1.0	1.0	90.0	1.0
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	slide_2_fingers_up	1.0	1.0	1.0	88.0	
	stop_sign	1.0	1.0	1.0	90.0	
	swipe_down	1.0	1.0	1.0	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	1.0	1.0	1.0	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
	zoom_in_with_full_hand	1.0	1.0	1.0	90.0	
	zoom_out_with_2_fingers	1.0	1.0	1.0	88.0	

Tabla A.15: Métricas de los experimentos (Parte XII).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
92	pull_hand_in	1.0	0.99	0.99	90.0	0.998
	slide_2_fingers_right	1.0	1.0	1.0	89.0	
	slide_2_fingers_up	1.0	1.0	1.0	88.0	
	stop_sign	1.0	1.0	1.0	90.0	
	swipe_down	1.0	1.0	1.0	86.0	
	swipe_left	1.0	1.0	1.0	88.0	
	thumb_up	0.99	1.0	0.99	86.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	91.0	
	zoom_in_with_full_hand	1.0	1.0	1.0	87.0	
	zoom_out_with_2_fingers	1.0	1.0	1.0	89.0	
93	pull_hand_in	1.0	1.0	1.0	154.0	0.998
	slide_2_fingers_right	0.99	1.0	1.0	152.0	
	slide_2_fingers_up	1.0	1.0	1.0	159.0	
	stop_sign	1.0	1.0	1.0	155.0	
	swipe_down	1.0	1.0	1.0	150.0	
	swipe_left	1.0	1.0	1.0	150.0	
	thumb_up	0.99	0.99	0.99	147.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	152.0	
	zoom_in_with_full_hand	0.99	0.98	0.99	145.0	
	zoom_out_with_2_fingers	1.0	1.0	1.0	145.0	
94	pull_hand_in	0.99	1.0	1.0	154.0	0.996
	slide_2_fingers_right	1.0	1.0	1.0	152.0	
	slide_2_fingers_up	1.0	1.0	1.0	159.0	
	stop_sign	1.0	0.99	1.0	155.0	
	swipe_down	0.99	1.0	1.0	150.0	
	swipe_left	1.0	1.0	1.0	150.0	
	thumb_up	1.0	0.98	0.99	147.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	152.0	
	zoom_in_with_full_hand	0.98	0.99	0.99	145.0	
	zoom_out_with_2_fingers	1.0	1.0	1.0	145.0	
95	pull_hand_in	0.73	0.7	0.72	154.0	0.726
	slide_2_fingers_right	0.61	0.91	0.73	152.0	
	slide_2_fingers_up	0.69	0.81	0.74	159.0	
	stop_sign	0.64	0.8	0.71	155.0	
	swipe_down	0.75	0.57	0.65	150.0	
	swipe_left	0.78	0.72	0.75	150.0	
	thumb_up	0.88	0.65	0.75	147.0	
	zoom_in_with_2_fingers	0.78	0.85	0.81	152.0	
	zoom_in_with_full_hand	0.83	0.63	0.72	145.0	
	zoom_out_with_2_fingers	0.81	0.66	0.73	145.0	

Tabla A.16: Métricas de los experimentos (Parte XIII).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
96	pull_hand_in	1.0	1.0	1.0	154.0	0.997
	slide_2_fingers_right	1.0	1.0	1.0	152.0	
	slide_2_fingers_up	1.0	1.0	1.0	159.0	
	stop_sign	1.0	1.0	1.0	155.0	
	swipe_down	1.0	0.99	0.99	150.0	
	swipe_left	1.0	1.0	1.0	150.0	
	thumb_up	0.99	0.99	0.99	147.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	152.0	
	zoom_in_with_full_hand	0.98	0.99	0.99	145.0	
	zoom_out_with_2_fingers	0.99	1.0	1.0	145.0	
97	pull_hand_in	1.0	1.0	1.0	154.0	0.998
	slide_2_fingers_right	1.0	1.0	1.0	152.0	
	slide_2_fingers_up	1.0	1.0	1.0	159.0	
	stop_sign	1.0	1.0	1.0	155.0	
	swipe_down	1.0	1.0	1.0	150.0	
	swipe_left	1.0	1.0	1.0	150.0	
	thumb_up	0.96	1.0	0.98	147.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	152.0	
	zoom_in_with_full_hand	1.0	0.96	0.98	145.0	
	zoom_out_with_2_fingers	1.0	1.0	1.0	145.0	
98	pull_hand_in	0.99	1.0	1.0	154.0	0.996
	slide_2_fingers_right	1.0	1.0	1.0	152.0	
	slide_2_fingers_up	1.0	1.0	1.0	159.0	
	stop_sign	1.0	0.99	1.0	155.0	
	swipe_down	1.0	1.0	1.0	150.0	
	swipe_left	1.0	1.0	1.0	150.0	
	thumb_up	0.97	1.0	0.99	147.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	152.0	
	zoom_in_with_full_hand	1.0	0.97	0.99	145.0	
	zoom_out_with_2_fingers	1.0	1.0	1.0	145.0	
99	pull_hand_in	1.0	1.0	1.0	154.0	1.0
	slide_2_fingers_right	1.0	1.0	1.0	152.0	
	slide_2_fingers_up	1.0	1.0	1.0	159.0	
	stop_sign	1.0	1.0	1.0	155.0	
	swipe_down	1.0	1.0	1.0	150.0	
	swipe_left	1.0	1.0	1.0	150.0	
	thumb_up	1.0	1.0	1.0	147.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	152.0	
	zoom_in_with_full_hand	1.0	1.0	1.0	145.0	
	zoom_out_with_2_fingers	1.0	1.0	1.0	145.0	

Tabla A.17: Métricas de los experimentos (Parte XIV).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
100	pull_hand_in	0.62	0.65	0.63	154.0	0.569
	slide_2_fingers_right	0.62	0.86	0.72	152.0	
	slide_2_fingers_up	0.5	0.53	0.52	159.0	
	stop_sign	0.67	0.65	0.66	155.0	
	swipe_down	0.7	0.5	0.58	150.0	
	swipe_left	0.67	0.41	0.51	150.0	
	thumb_up	0.73	0.46	0.56	147.0	
	zoom_in_with_2_fingers	0.6	0.9	0.72	152.0	
	zoom_in_with_full_hand	0.59	0.74	0.66	145.0	
	zoom_out_with_2_fingers	0.56	0.45	0.5	145.0	
101	pull_hand_in	0.87	0.77	0.82	154.0	0.785
	slide_2_fingers_right	0.75	0.88	0.81	152.0	
	slide_2_fingers_up	0.84	0.75	0.79	159.0	
	stop_sign	0.74	0.83	0.78	155.0	
	swipe_down	0.83	0.82	0.82	150.0	
	swipe_left	0.82	0.84	0.83	150.0	
	thumb_up	0.93	0.69	0.79	147.0	
	zoom_in_with_2_fingers	0.84	0.91	0.87	152.0	
	zoom_in_with_full_hand	0.88	0.85	0.86	145.0	
	zoom_out_with_2_fingers	0.79	0.86	0.82	145.0	
102	pull_hand_in	0.91	0.82	0.86	154.0	0.866
	slide_2_fingers_right	0.81	0.96	0.88	152.0	
	slide_2_fingers_up	0.87	0.84	0.86	159.0	
	stop_sign	0.76	0.87	0.81	155.0	
	swipe_down	0.86	0.87	0.86	150.0	
	swipe_left	0.88	0.89	0.88	150.0	
	thumb_up	0.91	0.76	0.83	147.0	
	zoom_in_with_2_fingers	0.83	0.94	0.88	152.0	
	zoom_in_with_full_hand	0.85	0.81	0.83	145.0	
	zoom_out_with_2_fingers	0.97	0.81	0.88	145.0	
103	pull_hand_in	1.0	0.99	1.0	154.0	0.942
	slide_2_fingers_right	1.0	1.0	1.0	152.0	
	slide_2_fingers_up	1.0	1.0	1.0	159.0	
	stop_sign	1.0	0.99	0.99	155.0	
	swipe_down	1.0	0.99	1.0	150.0	
	swipe_left	1.0	1.0	1.0	150.0	
	thumb_up	0.99	1.0	1.0	147.0	
	zoom_in_with_2_fingers	0.99	1.0	0.99	152.0	
	zoom_in_with_full_hand	1.0	1.0	1.0	145.0	
	zoom_out_with_2_fingers	0.99	1.0	1.0	145.0	

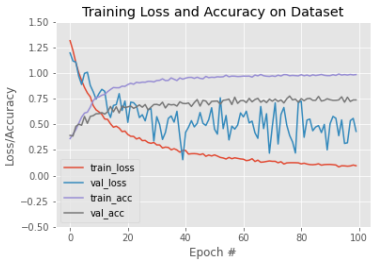
Tabla A.18: Métricas de los experimentos (Parte XV).

Id	Gesto	Precision	Recall	F1_score	Support	Accuracy
104	pull_hand_in	1.0	1.0	1.0	154.0	0.998
	slide_2_fingers_right	1.0	1.0	1.0	152.0	
	slide_2_fingers_up	1.0	1.0	1.0	159.0	
	stop_sign	1.0	1.0	1.0	155.0	
	swipe_down	1.0	0.99	1.0	150.0	
	swipe_left	1.0	1.0	1.0	150.0	
	thumb_up	1.0	0.97	0.99	147.0	
	zoom_in_with_2_fingers	1.0	1.0	1.0	152.0	
	zoom_in_with_full_hand	0.97	1.0	0.98	145.0	
	zoom_out_with_2_fingers	1.0	1.0	1.0	145.0	

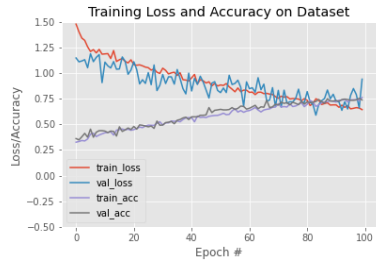
Tabla A.19: Métricas de los experimentos (Parte XVI).

A.2.2. Gráficas de los experimentos

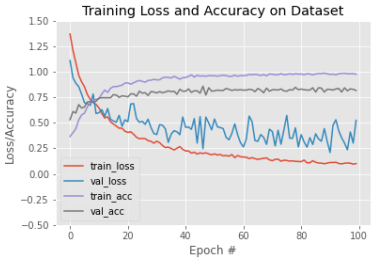
En esta sección se ilustran las gráficas que han sido obtenidas en el entrenamiento de los experimentos, donde se muestra la tendencia para la precisión y el error tanto en el proceso de entrenamiento como de validación del modelo.



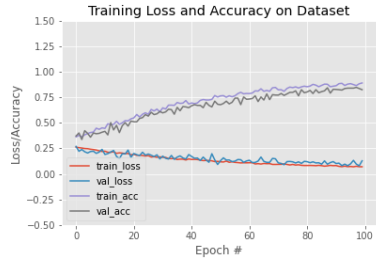
(a)



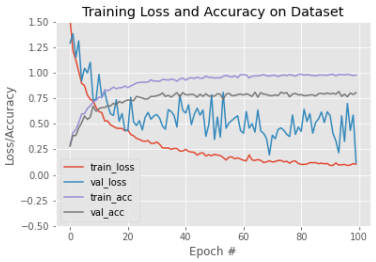
(b)



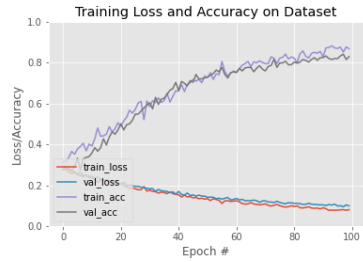
(c)



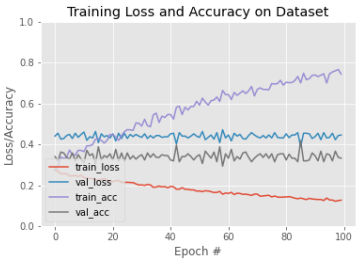
(d)



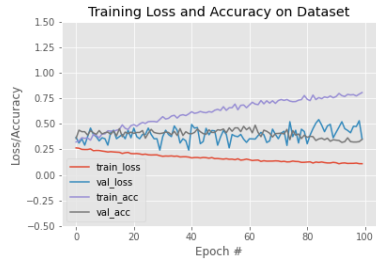
(e)



(f)

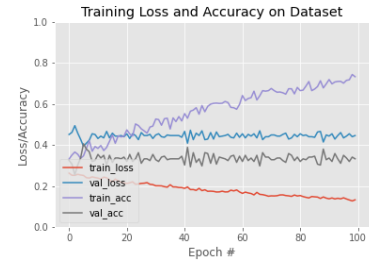


(g)

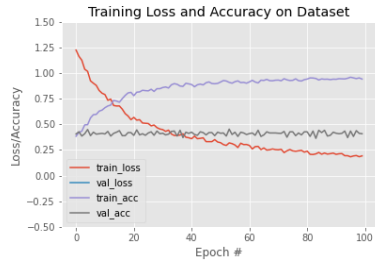


(h)

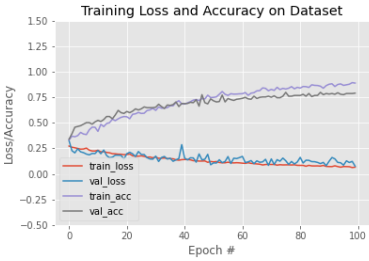
Figura A.1: Gráficas de los experimentos (Parte I).



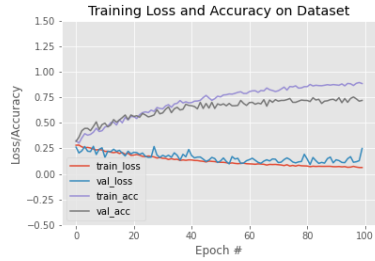
(a)



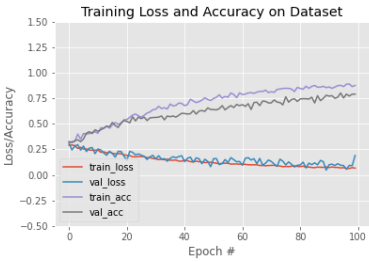
(b)



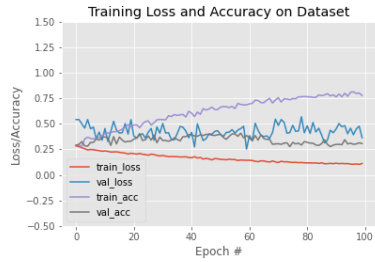
(c)



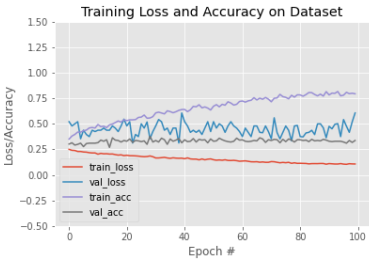
(d)



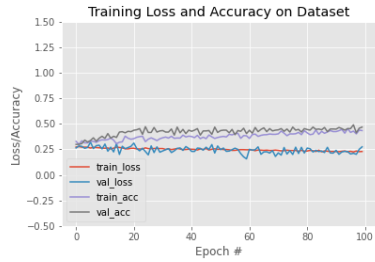
(e)



(f)

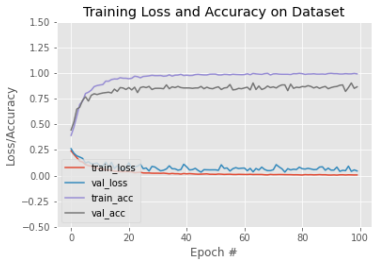


(g)

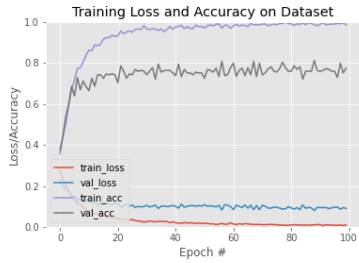


(h)

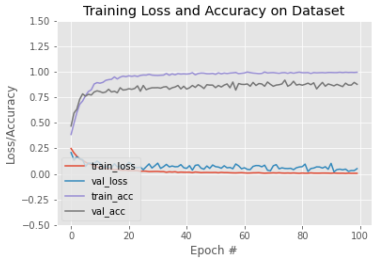
Figura A.2: Gráficas de los experimentos (Parte II).



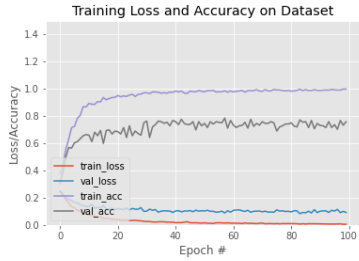
(a)



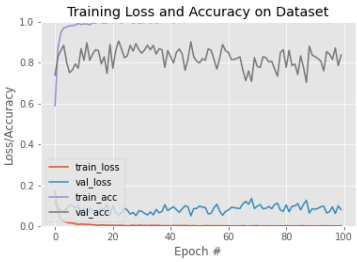
(b)



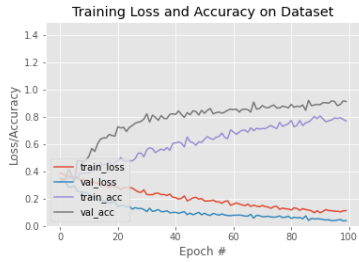
(c)



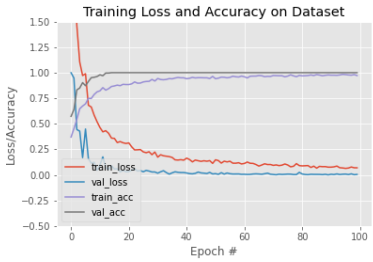
(d)



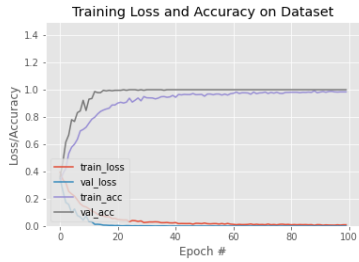
(e)



(f)

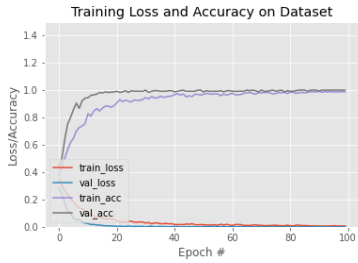


(g)

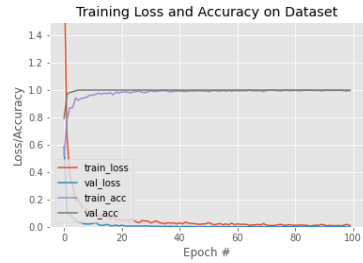


(h)

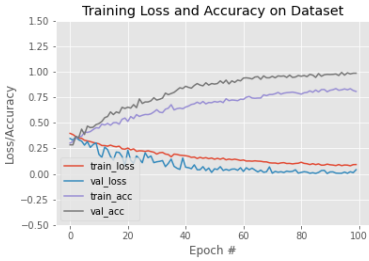
Figura A.3: Gráficas de los experimentos (Parte III).



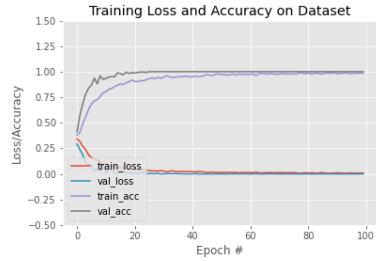
(a)



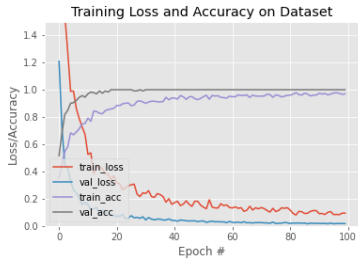
(b)



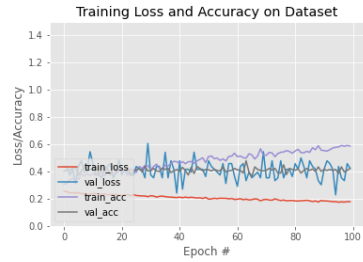
(c)



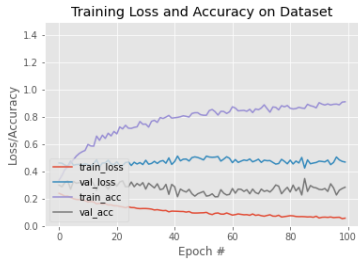
(d)



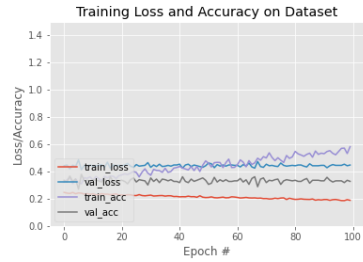
(e)



(f)

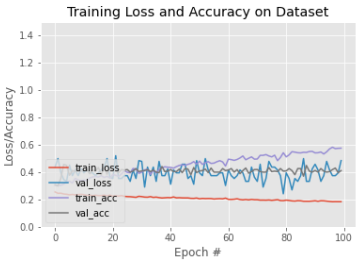


(g)

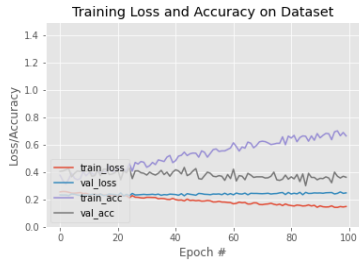


(h)

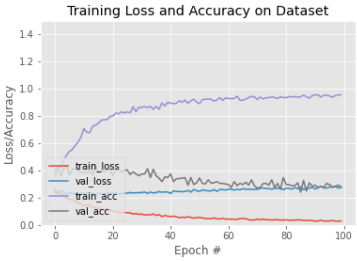
Figura A.4: Gráficas de los experimentos (Parte IV).



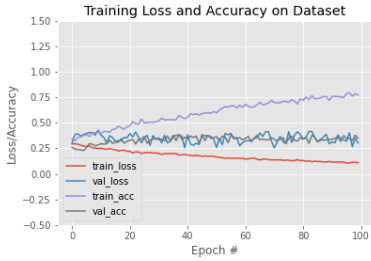
(a)



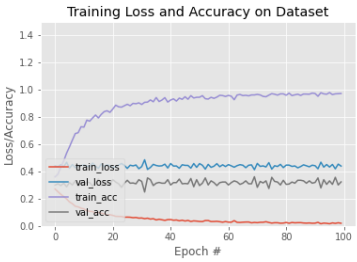
(b)



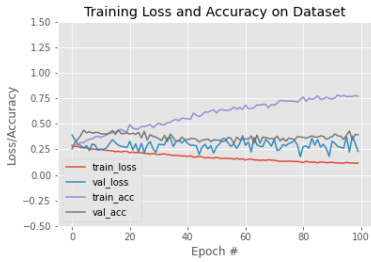
(c)



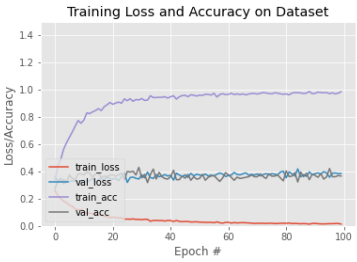
(d)



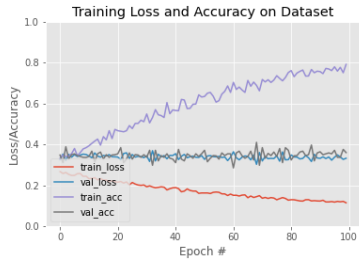
(e)



(f)



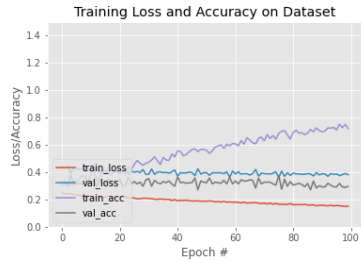
(g)



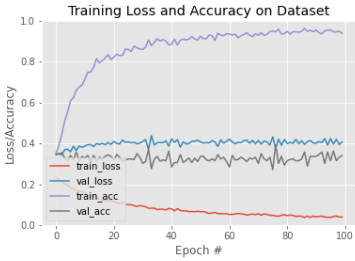
(h)

Figura A.5: Gráficas de los experimentos (Parte V).

(a)

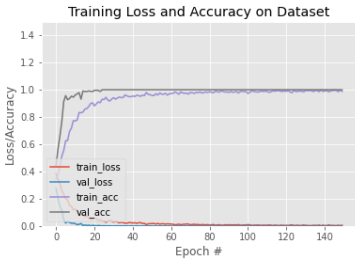


(b)

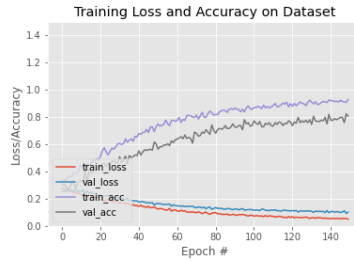


(c)

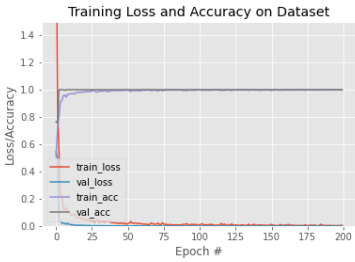
(d)



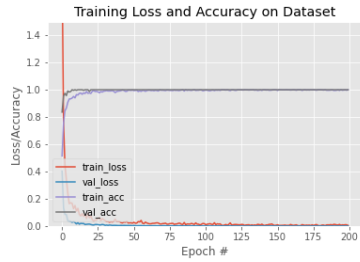
(e)



(f)



(g)



(h)

Figura A.6: Gráficas de los experimentos (Parte VI).

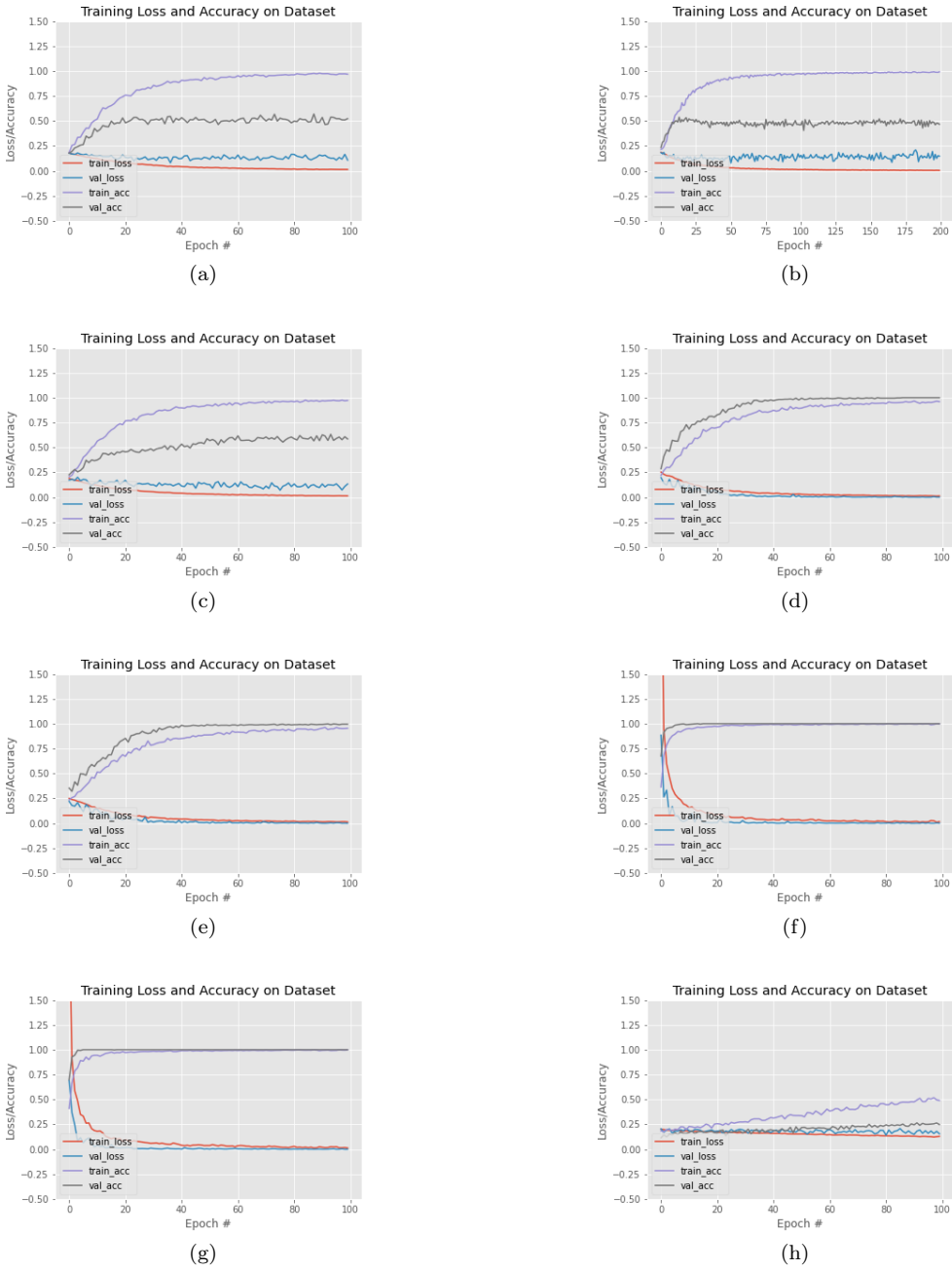
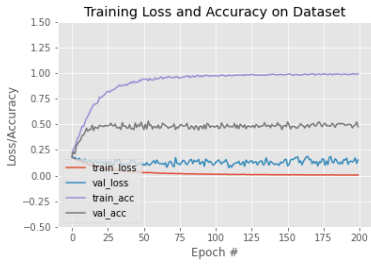
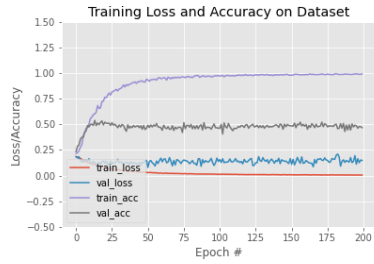


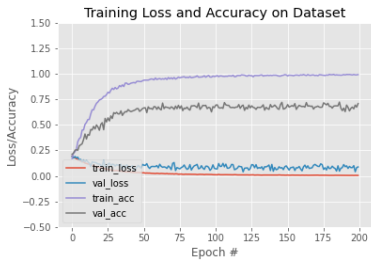
Figura A.7: Gráficas de los experimentos (Parte VII).



(a)

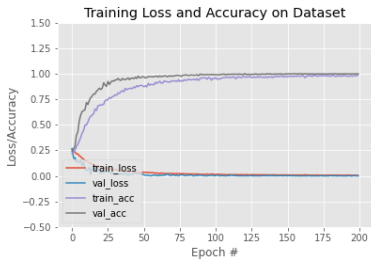


(b)

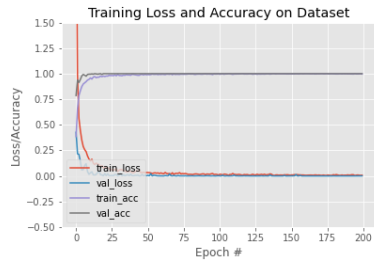


(c)

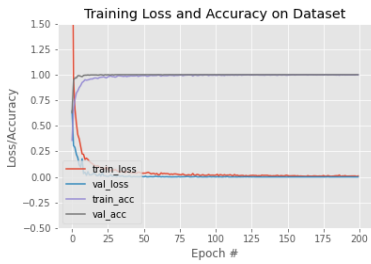
(d)



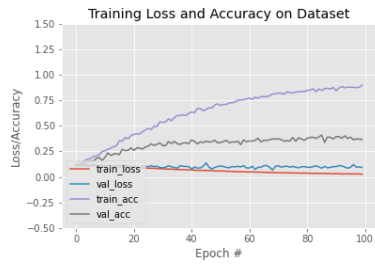
(e)



(f)

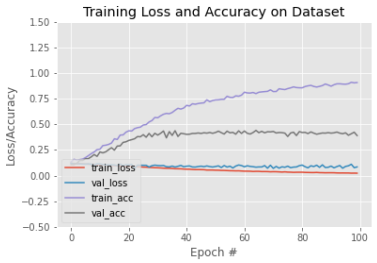


(g)

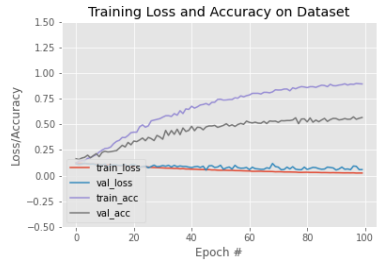


(h)

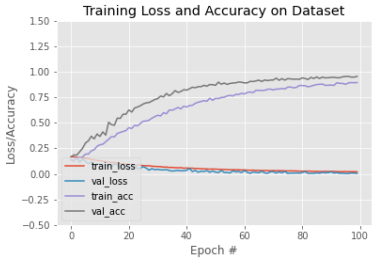
Figura A.8: Gráficas de los experimentos (Parte VIII).



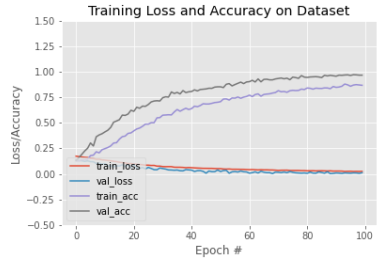
(a)



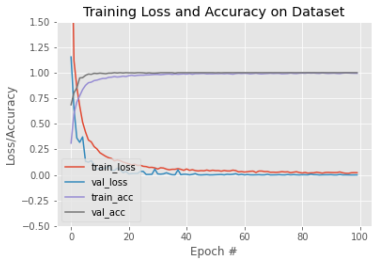
(b)



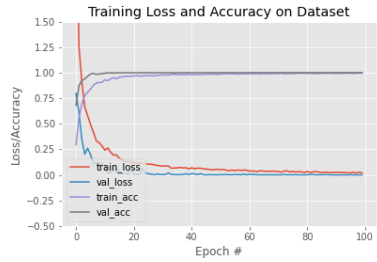
(c)



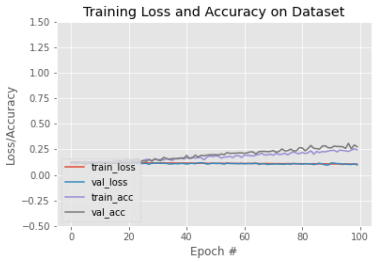
(d)



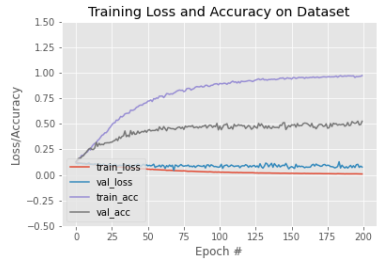
(e)



(f)

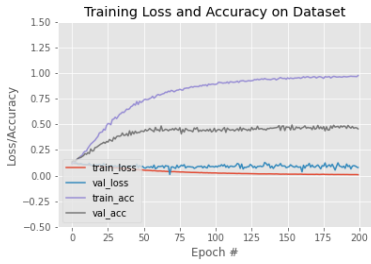


(g)

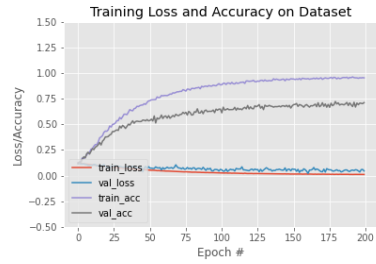


(h)

Figura A.9: Gráficas de los experimentos (Parte IX).



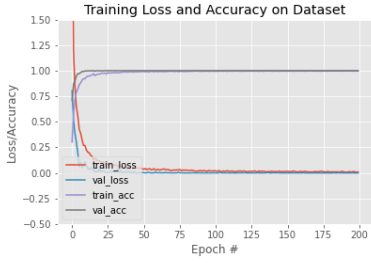
(a)



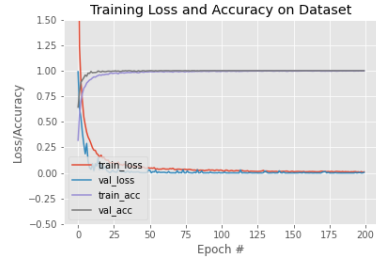
(b)

(c)

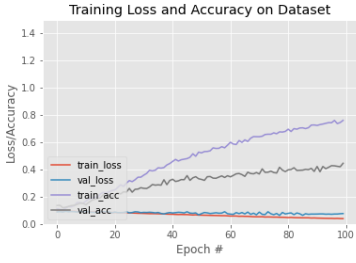
(d)



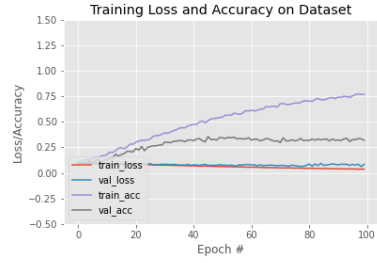
(e)



(f)



(g)



(h)

Figura A.10: Gráficas de los experimentos (Parte X).

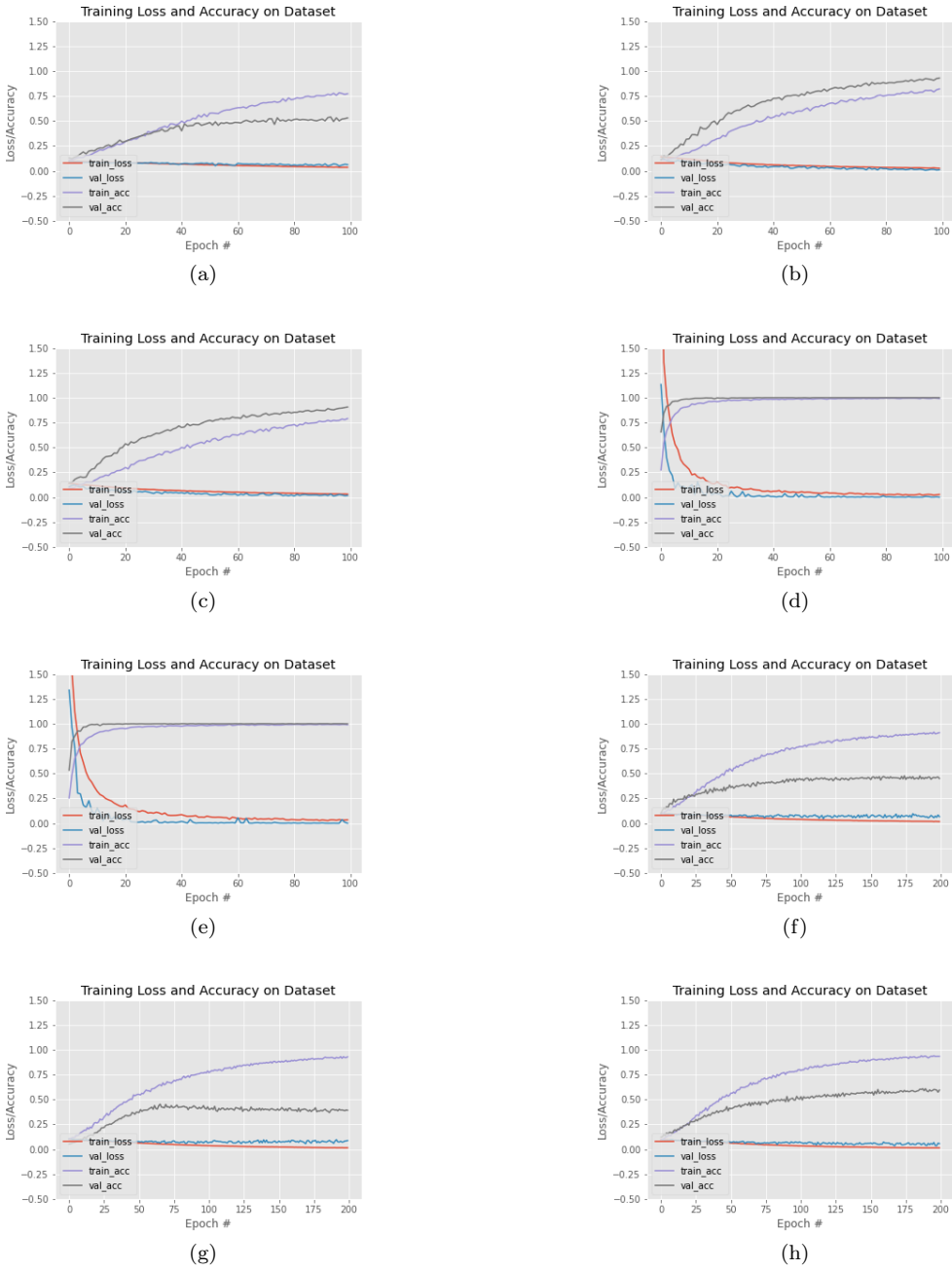
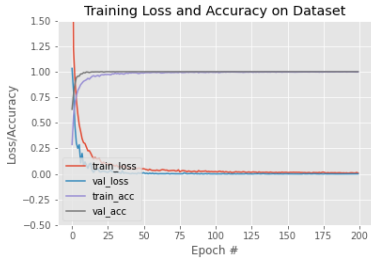


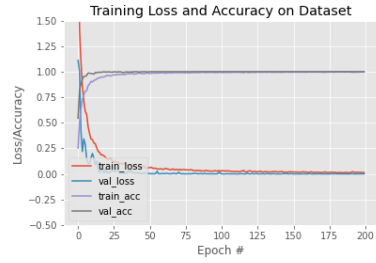
Figura A.11: Gráficas de los experimentos (Parte XI).

(a)

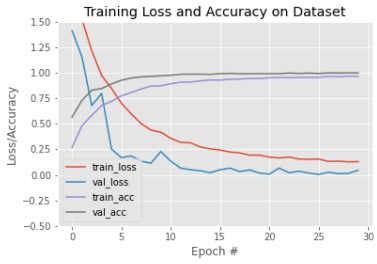
(b)



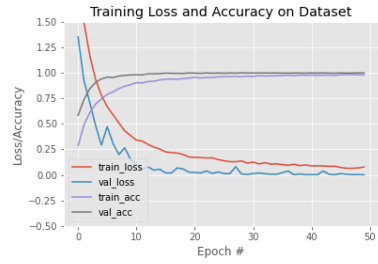
(c)



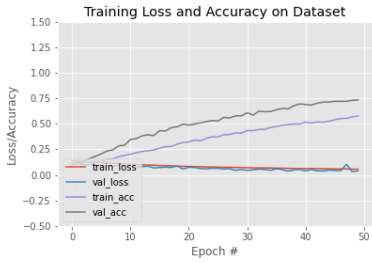
(d)



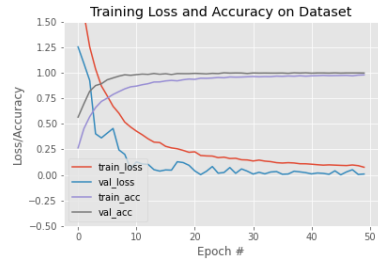
(e)



(f)

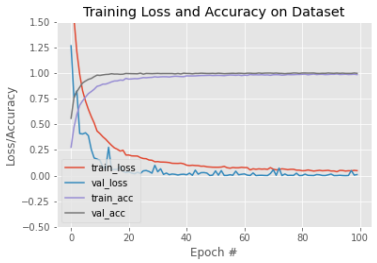


(g)

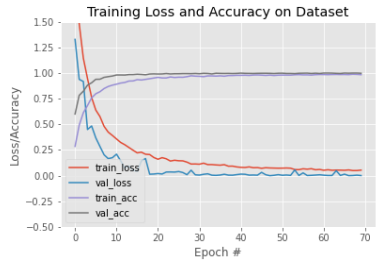


(h)

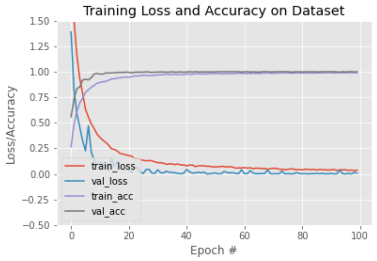
Figura A.12: Gráficas de los experimentos (Parte XII).



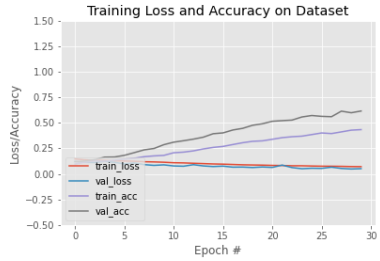
(a)



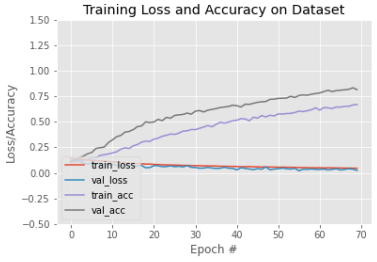
(b)



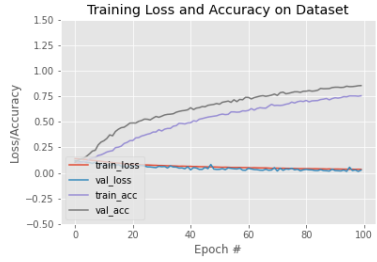
(c)



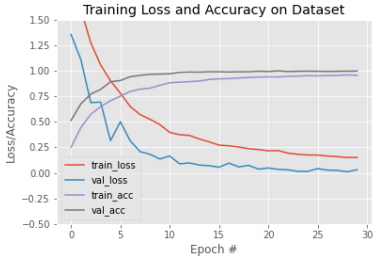
(d)



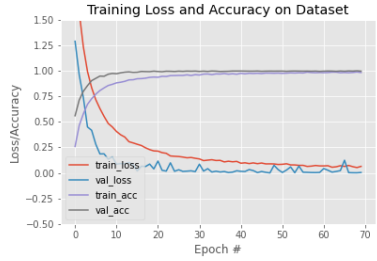
(e)



(f)



(g)

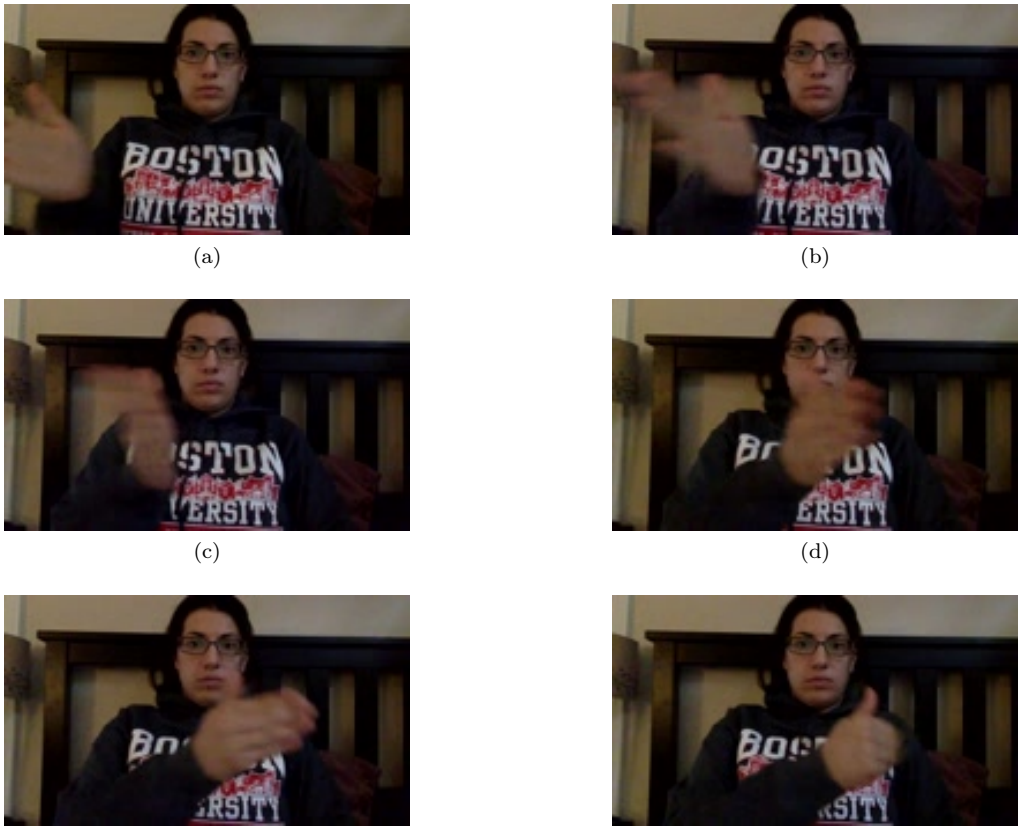


(h)

Figura A.13: Gráficas de los experimentos (Parte XIII).

A.3. Ejemplos de gestos

En esta Tesis doctoral se han utilizado los gestos de las manos que contiene la base de datos 20BN-jester. En esta sección se muestran algunos ejemplos de los 10 gestos que se han utilizado para este trabajo procedentes de dicha base de datos.



(a) (b) (c) (d) (e) (f)

Figura A.14: Ejemplo del gesto deslizamiento a la izquierda (swipe left) de la base de datos 20BN-Jester.

(a) (b)

(c) (d)

(e) (f)

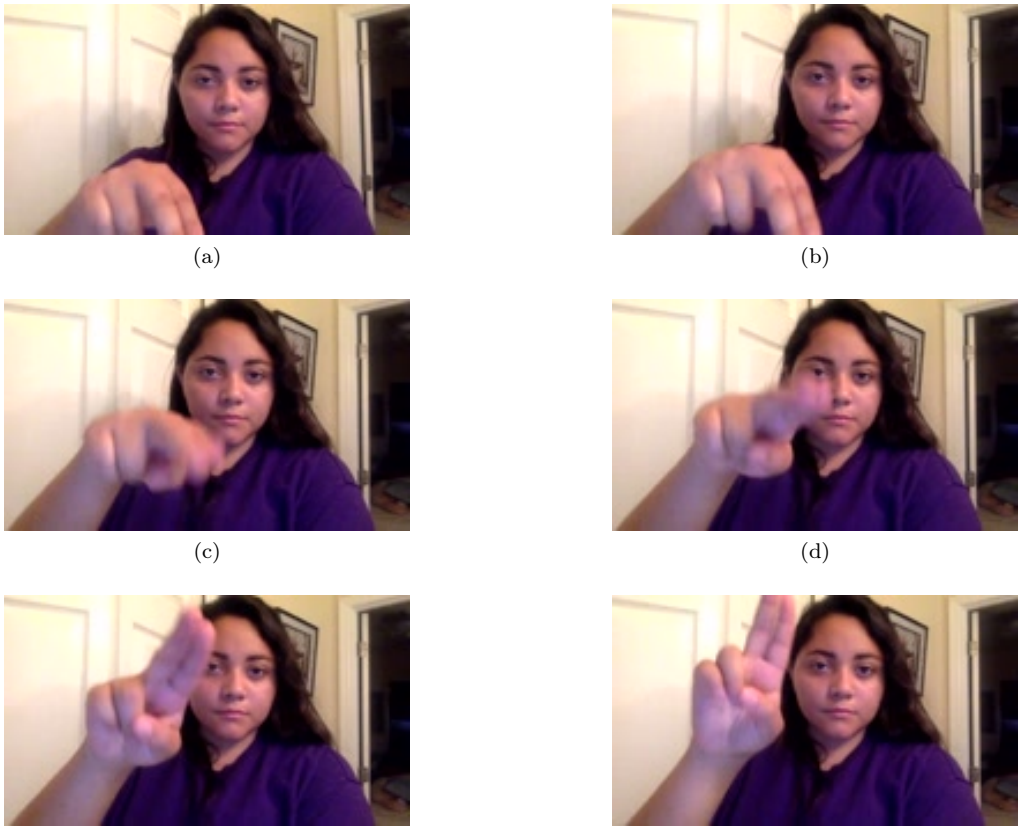
Figura A.15: Ejemplo del gesto deslizamiento hacia abajo (swipe down) de la base de datos 20BN-Jester.

(a) (b) (c) (d) (e) (f)

Figura A.16: Ejemplo del gesto tirar de la mano (pull hand in) de la base de datos 20BN-Jester.

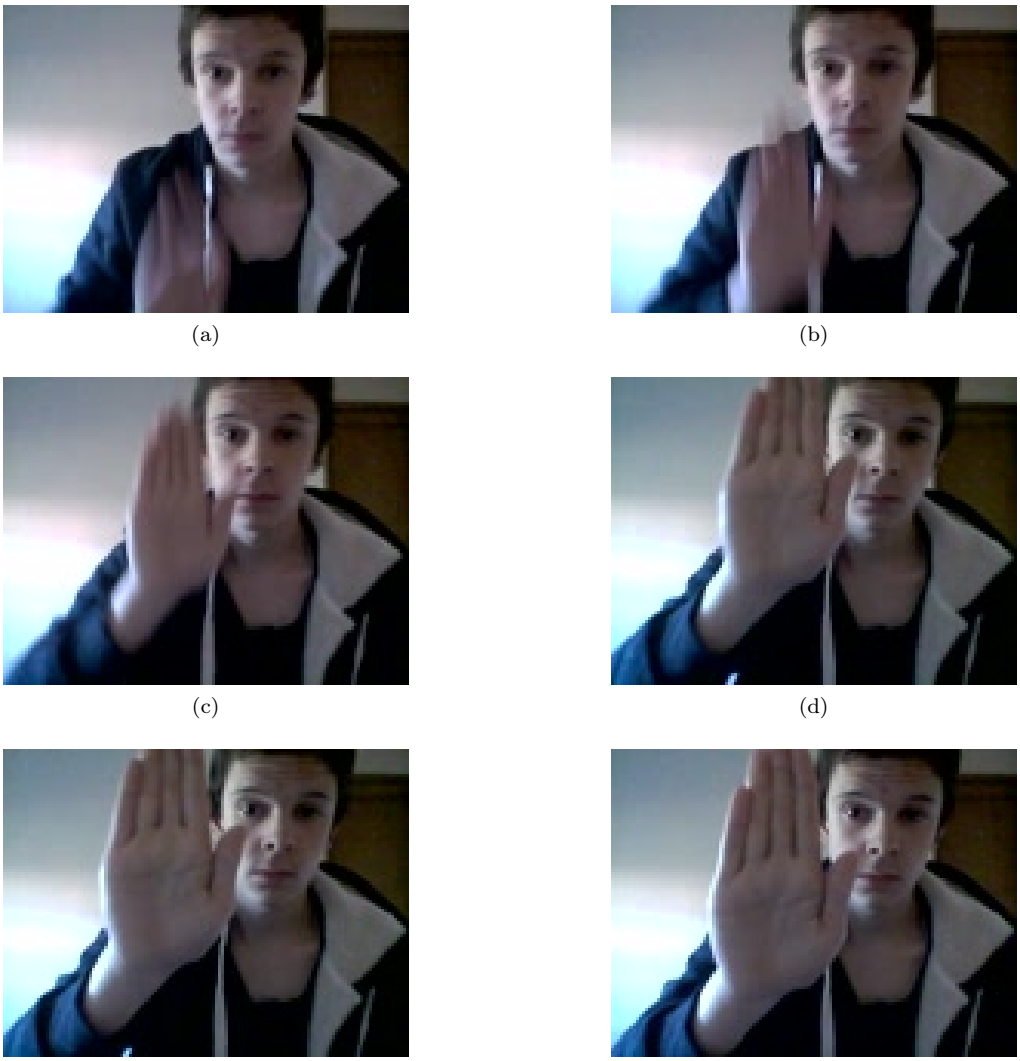
(a) (b) (c) (d) (e) (f)

Figura A.17: Ejemplo del gesto deslizar 2 dedos a la derecha (slide 2 fingers right) de la base de datos 20BN-Jester.



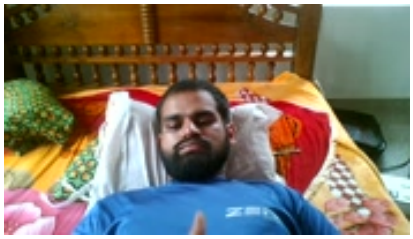
(a) (b) (c) (d) (e) (f)

Figura A.18: Ejemplo del gesto deslizar 2 dedos hacia arriba (slide 2 fingers up) de la base de datos 20BN-Jester.

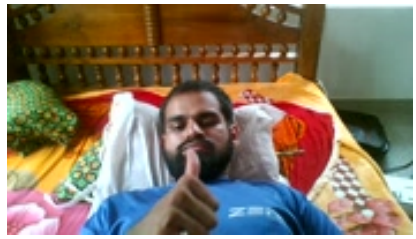


(a) (b) (c) (d) (e) (f)

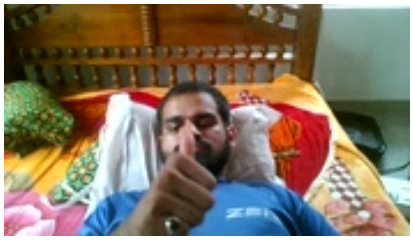
Figura A.19: Ejemplo del gesto de parar (stop sign) de la base de datos 20BN-Jester.



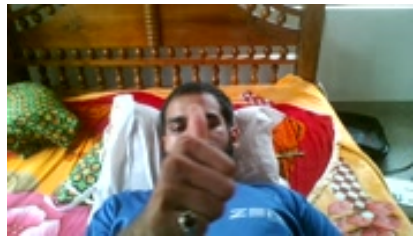
(a)



(b)

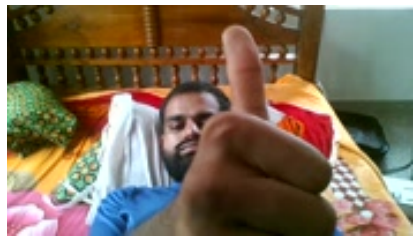


(c)



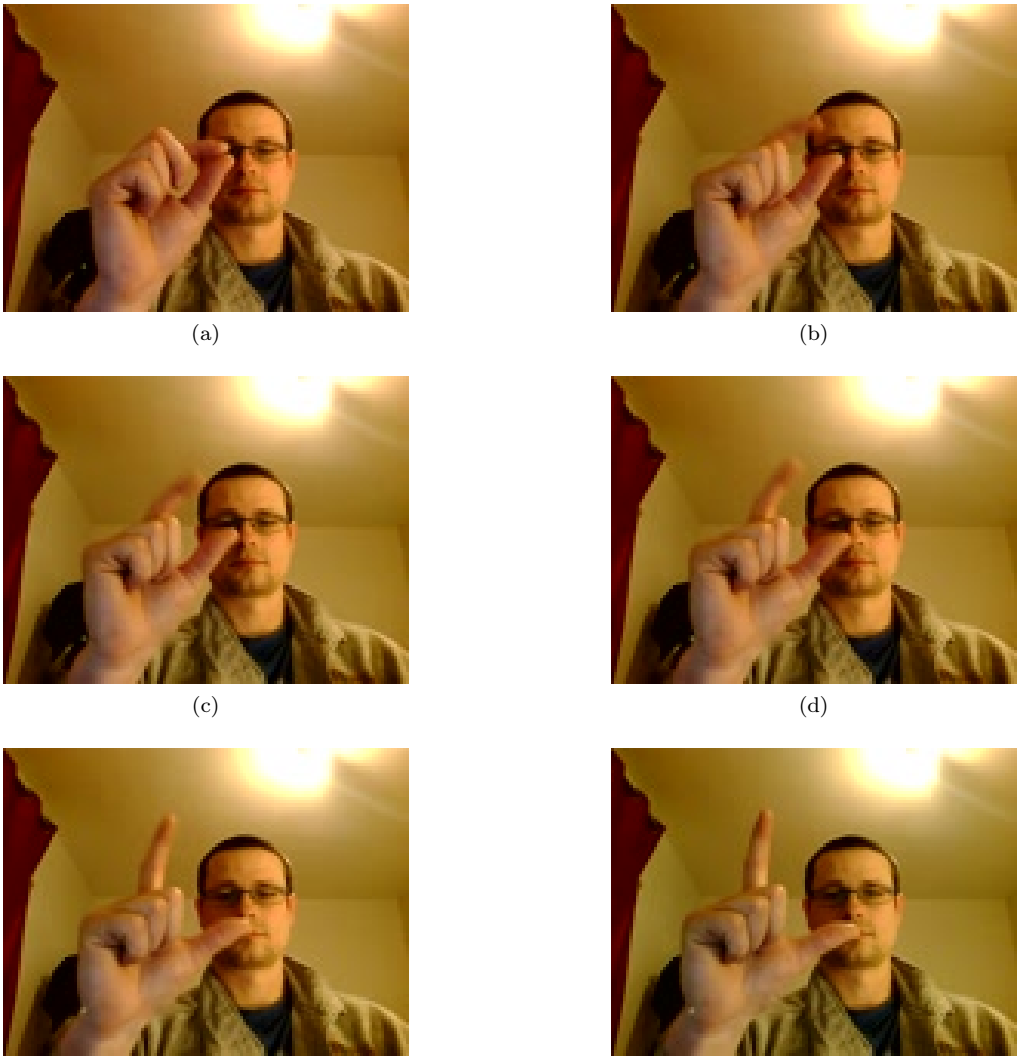
(d)

(e)



(f)

Figura A.20: Ejemplo del gesto de pulgar arriba (thumb up) de la base de datos 20BN-Jester.



(a) (b)

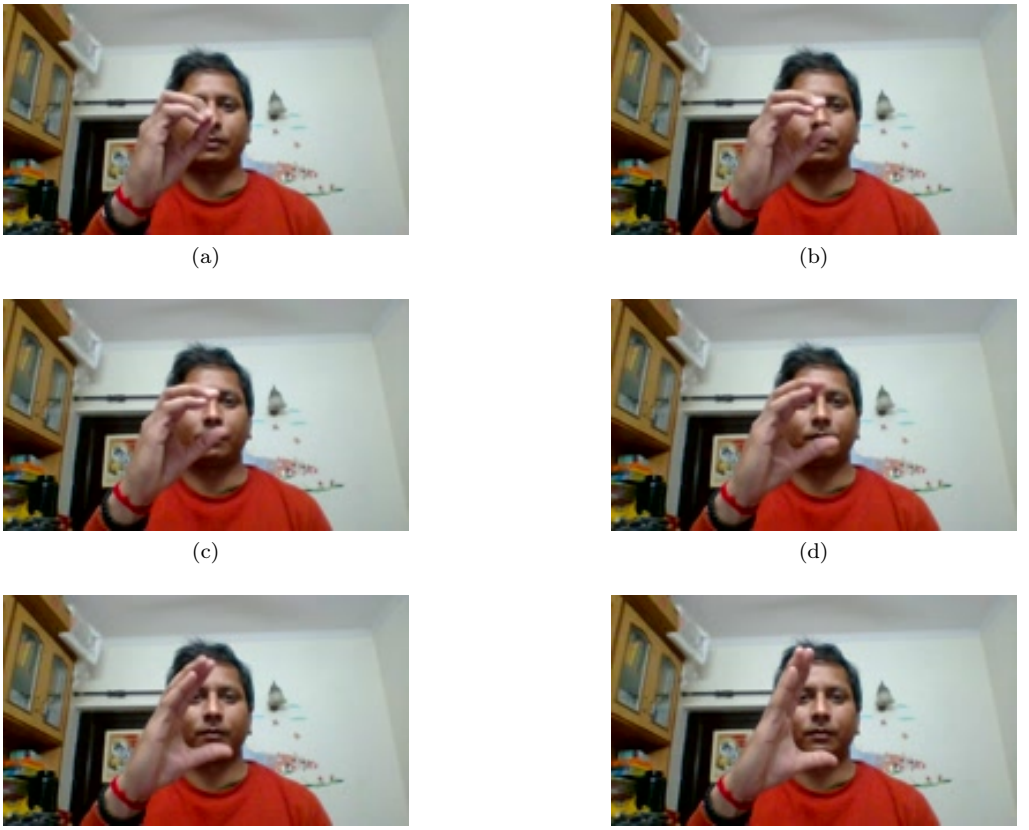
(c) (d)

(e) (f)

Figura A.21: Ejemplo del gesto de hacer zoom con 2 dedos (zoom in with 2 fingers) de la base de datos 20BN-Jester.

(a) (b) (c) (d) (e) (f)

Figura A.22: Ejemplo del gesto de disminuir zoom con 2 dedos (zoom out with 2 fingers) de la base de datos 20BN-Jester.



(a) (b) (c) (d) (e) (f)

Figura A.23: Ejemplo del gesto de hacer zoom (zoom in) con la mano de la base de datos 20BN-Jester.

Anexo B

Cuestionarios

Anexo B

CUESTIONARIOS

En este anexo se van a incluir los instrumentos que han sido utilizados en los procesos de evaluación para extraer la opinión de los usuarios finales respecto a los sistemas desarrollados en esta tesis doctoral.

Sentencia	**1**	**2**	**3**	**4**	**5**
La aplicación es fácil de usar					
Se accede fácilmente al tipo de actividad que quiero hacer					
La instalación del dispositiov sé hacerla sin problemas					
La instalación del dispositivo sé hacerla sin problemas					
La personalización de las actividades sé hacerlas fácilmente					
He tardado menos de un par de horas en usar la aplicación					

Tabla B.1: Cuestionario de usabilidad.

Sentencia	**1**	**2**	**3**	**4**	**5**
Las actividades se adaptan a mis demandas como docente					
Las actividades son útiles para mis estudiantes					
Las actividades se adaptan a las necesidades de mis estudiantes					
Las actividades son un recurso que utilizo por ser accesible (características físicas)					
Las actividades son un recurso que utilizao por se personalizable					

Tabla B.2: Cuestionario del apartado educativo.

Sentencia	**1**	**2**	**3**	**4**	**5**
Se reconoce a sí mismo en la actividad					
Identifica los movimientos de la pantalla con sus propios movimientos					
Utiliza de forma correcta la lateralidad					
Comprende la instrucción dada					
Imita el movimiento de la instrucción					
Realiza la instrucción dada de forma autónoma					
Realiza la instrucción dada con algún tipo de ayuda					
Repite la consigna verbal					
Se autorregula durante la actividad					
Presenta motivación ante la actividad					
Muestra mayor interés ante las actividades con sonidos, estímulos visuales, feedback, etc					
Presenta frustración ante dificultades en el uso de la aplicación					
Grado de éxito en la realización de la tarea					
Tiempo de latencia de respuesta					

Tabla B.3: Cuestionario de las capacidades del alumnado.

	1	2	3	4	5	6	7		
desagradable	○	○	○	○	○	○	○	agradable	1
no entendible	○	○	○	○	○	○	○	entendible	2
creativo	○	○	○	○	○	○	○	sin imaginación	3
fácil de aprender	○	○	○	○	○	○	○	difícil de aprender	4
valioso	○	○	○	○	○	○	○	de poco valor	5
aburrido	○	○	○	○	○	○	○	emocionante	6
no interesante	○	○	○	○	○	○	○	interesante	7
impredecible	○	○	○	○	○	○	○	predecible	8
rápido	○	○	○	○	○	○	○	lento	9
original	○	○	○	○	○	○	○	convencional	10
obstructivo	○	○	○	○	○	○	○	impulsor de apoyo	11
bueno	○	○	○	○	○	○	○	malo	12
complicado	○	○	○	○	○	○	○	fácil	13
repeler	○	○	○	○	○	○	○	atraer	14
convencional	○	○	○	○	○	○	○	novedoso	15
incómodo	○	○	○	○	○	○	○	cómodo	16
seguro	○	○	○	○	○	○	○	inseguro	17
activante	○	○	○	○	○	○	○	adormecedor	18
cubre expectativas	○	○	○	○	○	○	○	no cubre expectativas	19
ineficiente	○	○	○	○	○	○	○	eficiente	20
claro	○	○	○	○	○	○	○	confuso	21
no pragmático	○	○	○	○	○	○	○	pragmático	22
ordenado	○	○	○	○	○	○	○	sobrecargado	23
atractivo	○	○	○	○	○	○	○	feo	24
simpático	○	○	○	○	○	○	○	antipático	25
conservador	○	○	○	○	○	○	○	innovador	26

Figura B.1: Cuestionario UEQ (Los valores son asignados de 1 a 7).

Acrónimos

Acrónimos

AI	Artificial Intelligence
API	Application Programming Interface
BCI	Brain Computer-Interaction
CPU	Central Processing Unit
CSV	Comma Separated Values
CNN	Convolutional Neural Network
DA	Deep Domain Adaptation
DGCNN	Dynamic Graph Convolutional Neural Network
DL	Deep Learning
DTW	Dynamic Time Warping
EMG	Electromyography
FL	Fuzzy Logic
GB	Gigabytes
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
KNN	K-Nearest Neighbor
LSTM	Long-short Term Memory
MB	Megabytes
ML	Machine Learning

MINECO	Ministerio de Economía y Competitividad
NUI	Natural User Interfaces
PCA	Principal Component Analysis
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TDNN	Time Delay Neural Network
TL	Transfer Learning
UEQ	User Experience Questionnaire
UI	User Interface
URL	Uniform Resource Locator
XML	eXtensible Markup Language
VAAS	Voice as a Service
VR	Virtual Reality

Bibliografía

Bibliografía

[kin, 2018] (2018). Microsoft kinect v2. https://www.profesionalreview.com/2018/01/04/microsoft-mata-kinect-forma-oficial/. Acceso: 14-12-2021.

[emo, 2021] (2021). Emotiv epoc+. https://www.emotiv.com/setup/. Acceso: 15-12-2021.

[lea, 2021] (2021). Leap motion. https://www.freepng.es/png-7ami71/. Acceso: 14-12-2021.

[kin, 2021] (2021). Microsoft kinect v1. https://es.wikipedia.org/wiki/Kinect. Acceso: 14-12-2021.

[myo, 2021] (2021). Myo. https://www.pinterest.es/pin/204843483026827070/. Acceso: 15-12-2021.

[ope, 2021] (2021). Openbci cyton board. https://docs.openbci.com/docs/02Cyton/CytonLanding. Acceso: 15-12-2021.

[tou, 2021] (2021). Pantalla táctil. https://pixabay.com/es/photos/smartphone-tel\%C3\%A9fono-celular-1894723/. Acceso: 15-12-2021.

[int, 2021] (2021). Pizarra interactiva. https://pixabay.com/es/photos/profesor-clase-de-educaci\%C3\%B3n-general-3765909/. Acceso: 15-12-2021.

[tob, 2021] (2021). Tobii pro glasses 2. https://www.tobiipro.com/es/products/tobii-pro-glasses-2/. Acceso: 15-12-2021.

[zed, 2021] (2021). Zed stereo camera. https://www.stereolabs.com/zed/. Acceso: 15-12-2021.

[Abid et al., 2015] Abid, M. R., Petriu, E. M., and Amjadian, E. (2015). Dynamic sign language recognition for smart home interactive application using stochastic linear formal grammar. *IEEE Transactions on Instrumentation and Measurement*, 64(3):596–605.

[Acedo et al., 2006] Acedo, F. J., Barroso, C., Casanueva, C., and Galán, J. L. (2006). Co-authorship in management and organizational studies: An empirical and network analysis. *Journal of Management Studies*, 43(5):957–983.

[Adithya et al., 2013] Adithya, V., Vinod, P., and Gopalakrishnan, U. (2013). Artificial neural network based method for indian sign language recognition. In *IEEE Conference on Information & Communication Technologies (ICT), 2013*, pages 1080–1085. IEEE.

[Agarwal and Thakur, 2013] Agarwal, A. and Thakur, M. K. (2013). Sign language recognition using microsoft kinect. In *Sixth International Conference on Contemporary Computing (IC3), 2013*, pages 181–185. IEEE.

[Agramunt et al., 2020] Agramunt, L., Berbel-Pineda, J., Capobianco-Uriarte, M., and Casado-Belmonte, M. (2020). Review on the relationship of absorptive capacity with interorganizational networks and the internationalization process. *Complexity*, 2020.

[Aiple and Schiele, 2013] Aiple, M. and Schiele, A. (2013). Pushing the limits of the cybergrasp^{TM} for haptic rendering. In *2013 IEEE International Conference on Robotics and Automation*, pages 3541–3546. IEEE.

[Al Dakheel et al., 2020] Al Dakheel, J., Del Pero, C., Aste, N., and Leonforte, F. (2020). Smart buildings features and key performance indicators: A review. *Sustainable Cities and Society*, page 102328.

[Al Hadhrami et al., 2018] Al Hadhrami, E., Al Mufti, M., Taha, B., and Werghi, N. (2018). Transfer learning with convolutional neural networks for moving target classification with micro-doppler radar spectrograms. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 148–154. IEEE.

[Al-Jarrah et al., 2006] Al-Jarrah, O. M., Shatnawi, A., and Halawani, A. (2006). Recognition of gestures in arabic sign language using neural networks. In *Artificial Intelligence and Soft Computing*, pages 132–137.

[Alayo et al., 2020] Alayo, M., Iturralde, T., Maseda, A., and Aparicio, G. (2020). Mapping family firm internationalization research: bibliometric and literature review. *Review of Managerial Science*, pages 1–44.

[Alvarado-Díaz et al., 2017] Alvarado-Díaz, W., Roman-Gonzalez, A., and Meneses, B. (2017). Implementation of a brain-machine interface for controlling a wheelchair. In *IEEE CHILECON 2017*.

[Anthes et al., 2016] Anthes, C., García-Hernández, R. J., Wiedemann, M., and Kranzlmüller, D. (2016). State of the art of virtual reality technology. In *2016 IEEE Aerospace Conference*, pages 1–19.

[Apruzzese et al., 2018] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., and Marchetti, M. (2018). On the effectiveness of machine and deep learning for cyber security. In *2018 10th International Conference on Cyber Conflict (CyCon)*, pages 371–390. IEEE.

[Arjunlal, 2016] Arjunlal, M. (2016). A survey on hand gesture recognition and hand tracking. *International Journal of Scientific Engineering and Applied Science*, 2(1):514–518.

[Atienza, 2018] Atienza, R. (2018). *Advanced Deep Learning with Keras: Apply deep learning techniques, autoencoders, GANs, variational autoencoders, deep reinforcement learning, policy gradients, and more*. Packt Publishing Ltd.

[Avila-Pesantez et al., 2018] Avila-Pesantez, D. F., Vaca-Cardenas, L. A., Avila, R. D., Padilla, N. P., and Rivera, L. A. (2018). Design of an augmented reality serious game for children with dyscalculia: a case study. In *International Conference on Technology Trends*, pages 165–175.

[Aziz et al., 2017] Aziz, R., Verma, C., and Srivastava, N. (2017). Dimension reduction methods for microarray data: a review. *AIMS Bioengineering*, 4(2):179–197.

[Baier-Fuentes et al., 2020] Baier-Fuentes, H., González-Serrano, M. H., Santos, A.-D., Inzunza-Mendoza, W., Pozo-Estrada, V., et al. (2020). Emotions and sport management: a bibliometric overview. *Frontiers in Psychology*, 11:1512.

[Bautista et al., 2013] Bautista, M. A., Hernández-Vela, A., Ponce, V., Perez-Sala, X., Baró, X., Pujol, O., Angulo, C., and Escalera, S. (2013). Probability-based dynamic time warping for gesture recognition on rgb-d data. In *Advances in Depth Image Analysis and Applications*, pages 126–135.

[Bayer and Faigl, 2019] Bayer, J. and Faigl, J. (2019). On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense t265. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE.

[Bedi, 2013] Bedi, A. (2013). Gesture technologies & elderly-overview. *Internation Journal of Information System and Engineering (IJISE)*, 1(1):33–37.

[Benalcázar et al., 2017] Benalcázar, M. E., Jaramillo, A. G., Zea, A., Páez, A., Andaluz, V. H., et al. (2017). Hand gesture recognition using machine learning and the myo armband. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1040–1044. IEEE.

[Bhattacharya and Basu, 1998] Bhattacharya, S. and Basu, P. (1998). Mapping a research area at the micro level using co-word analysis. *Scientometrics*, 43(3):359–372.

[Boles and Rad, 2017] Boles, A. and Rad, P. (2017). Voice biometrics: Deep learning-based voiceprint authentication system. In *2017 12th System of Systems Engineering Conference (SoSE)*, pages 1–6.

[Bonanno et al., 2017] Bonanno, D., Nock, K., Smith, L., Elmore, P., and Petry, F. (2017). An approach to explainable deep learning using fuzzy inference. In *Next-Generation Analyst V*, volume 10207, page 102070D.

[Boutsika, 2014] Boutsika, E. (2014). Kinect in education: A proposal for children with autism. *Procedia Computer Science*, 27:123–129.

[Bouzit et al., 2002] Bouzit, M., Burdea, G., Popescu, G., and Boian, R. (2002). The rutgers master ii-new design force-feedback glove. *IEEE/ASME Transactions on mechatronics*, 7(2):256–263.

[Brownlee, 2018] Brownlee, J. (2018). How to configure the number of layers and nodes in a neural network. https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/. Consultado 9-6-2021.

[Brusilovsky and Millán, 2007] Brusilovsky, P. and Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In *The Adaptive Web*, pages 3–53.

[Burmeister et al., 2016] Burmeister, D., Schrader, A., and Altakrouri, B. (2016). Reflective interaction capabilities by use of ambient manuals for an ambient light-control. In *International Conference on Human-Computer Interaction*, pages 409–415. Springer.

[Butt et al., 2018] Butt, A. H., Rovini, E., Dolciotti, C., De Petris, G., Bongioanni, P., Carboncini, M., and Cavallo, F. (2018). Objective and automatic classification of parkinson disease with leap motion controller. *Biomedical Engineering Online*, 17(1):1–21.

[Cai et al., 2017] Cai, Z., Han, J., Liu, L., and Shao, L. (2017). Rgb-d datasets using microsoft kinect or similar sensors: a survey. *Multimedia Tools and Applications*, 76(3):4313–4355.

[Callon et al., 1983] Callon, M., Courtial, J.-P., Turner, W. A., and Bauin, S. (1983). From translations to problematic networks: An introduction to co-word analysis. *Social Science Information*, 22(2):191–235.

[Camboard Pico, 2016] Camboard Pico (2016). Reference design brief camboard pico. https://pmdtec.com/picofamily/wp-content/uploads/2018/03/PMD_DevKit_Brief_CB_pico_flexx_CE_V0218-1.pdf. Consultado 9-6-2021.

[Capobianco-Uriarte et al., 2019] Capobianco-Uriarte, M. d. l. M., Casado-Belmonte, M. d. P., Marín-Carrillo, G. M., and Terán-Yépez, E. (2019). A bibliometric analysis of international competitiveness (1983–2017). *Sustainability*, 11(7):1877.

[Celebi et al., 2013] Celebi, S., Aydin, A. S., Temiz, T. T., and Arici, T. (2013). Gesture recognition using skeleton data with weighted dynamic time warping. In *International Conference on Computer Vision Theory and Applications*, pages 620–625.

[Chai et al., 2013] Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X., and Zhou, M. (2013). Sign language recognition and translation with kinect. In *IEEE International Conference on Automatic Face Gesture Recognition*.

[Cheng, 1995] Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799.

[Chicala et al., 2009] Chicala, J., Jacho, R., Atiencia Alarcon, L., and Vintimilla, B. (2009). *Reconocimiento y seguimiento de objetos móviles en un sistema de fútbol robótico*. Escuela Superior Politécnica del Litoral.

[Chuan et al., 2014] Chuan, C.-H., Regina, E., and Guardino, C. (2014). American sign language recognition using leap motion sensor. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 541–544. IEEE.

[Cieza and Stucki, 2008] Cieza, A. and Stucki, G. (2008). The international classification of functioning disability and health: its development process and content validity. *European Journal of Physical and Rehabilitation Medicine*, 44(3):303–313.

[Cobo et al., 2011] Cobo, M. J., López-Herrera, A. G., Herrera-Viedma, E., and Herrera, F. (2011). Science mapping software tools: Review, analysis, and cooperative study among tools. *Journal of the American Society for information Science and Technology*, 62(7):1382–1402.

[Collins, 2003] Collins, R. T. (2003). Mean-shift blob tracking through scale space. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003.*, volume 2, pages II–234. IEEE.

[Cooper et al., 2003] Cooper, A., Reimann, R., et al. (2003). *About face 2.0: The essentials of interaction design*, volume 17. Wiley Indianapolis.

[Côté-Allard et al., 2019] Côté-Allard, U., Fall, C. L., Drouin, A., Campeau-Lecours, A., Gosselin, C., Glette, K., Laviolette, F., and Gosselin, B. (2019). Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):760–771.

[Coulter et al., 1998] Coulter, N., Monarch, I., and Konda, S. (1998). Software engineering as seen through its research literature: A study in co-word analysis. *Journal of the American Society for Information Science*, 49(13):1206–1223.

[Czuszyński et al., 2018] Czuszyński, K., Rumiński, J., and Kwaśniewska, A. (2018). Gesture recognition with the linear optical sensor and recurrent neural networks. *IEEE Sensors Journal*, 18(13):5429–5438.

[Dardas and Georganas, 2011] Dardas, N. H. and Georganas, N. D. (2011). Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *IEEE Transactions on Instrumentation and Measurement*, 60(11):3592–3607.

[Devineau et al., 2018] Devineau, G., Moutarde, F., Xi, W., and Yang, J. (2018). Deep learning for hand gesture recognition on skeletal data. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 106–113. IEEE.

[Dhouib et al., 2016] Dhouib, A., Trabelsi, A., Kolski, C., and Neji, M. (2016). A classification and comparison of usability evaluation methods for interactive adaptive systems. In *2016 9th International Conference on Human System Interactions (HSI)*, pages 246–251. IEEE.

[Dias et al., 2013] Dias, T., Variz, M., Jorge, P., and Jesus, R. (2013). Gesture interaction system for social web applications on smart tvs. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, pages 225–226.

[DIN, 1996] DIN, E. (1996). 9241-10: Ergonomic requirements for office work with visual display terminals (vdts)–part 10: Dialogue principles. *International Organization for Standardization, Geneva, Switzerland*.

[Dinh et al., 2014] Dinh, D.-L., Kim, J. T., and Kim, T.-S. (2014). Hand gesture recognition and interface via a depth imaging sensor for smart home appliances. *Energy Procedia*, 62:576–582.

[Dovydaitis et al., 2016] Dovydaitis, L., Rasymas, T., and Rudžionis, V. (2016). Speaker authentication system based on voice biometrics and speech recognition. In *International Conference on Business Information Systems*, pages 79–84. Springer.

[Draelos et al., 2015] Draelos, M., Qiu, Q., Bronstein, A., and Sapiro, G. (2015). Intel realsense= real low cost gaze. In *IEEE International Conference on Image Processing (ICIP), 2015*, pages 2520–2524. IEEE.

[Druzhkov et al., 2011] Druzhkov, P., Erukhimov, V., Zolotykh, N. Y., Kozinov, E., Kustikova, V., Meerov, I., and Polovinkin, A. (2011). New object detection features in the opencv library. *Pattern Recognition and Image Analysis*, 21(3):384–386.

[Du et al., 2019] Du, K., Lin, X., Sun, Y., and Ma, X. (2019). Crossinfonet: Multi-task information sharing based hand pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9896–9905.

[Duchi et al., 2011] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7).

[Durka et al., 2012] Durka, P., Kuś, R., Żygierewicz, J., Michalska, M., Milanowski, P., Łabęcki, M., Spustek, T., Laszuk, D., Duszyk, A., and Kruszyński, M. (2012). User-centered design of brain-computer interfaces: OpenBCI.pl and BCI appliance. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 60(3):427–431.

[Eck and Waltman, 2010] Eck, N. J. and Waltman, L. (2010). Software survey: Vosviewer, a computer program for bibliometric mapping. *Scientometrics*, 2(84):523–538.

[Edmunds and Donovan, 2016] Edmunds, D. and Donovan, E. (2016). Su-g-jep4-01: An assessment of a microsoft kinect v2 sensor for voluntary breath-hold monitoring in radiotherapy. *Medical Physics*, 43(6):3681–3681.

[Elons et al., 2014] Elons, A., Ahmed, M., Shedid, H., and Tolba, M. (2014). Arabic sign language recognition using leap motion sensor. In *9th International Conference on Computer Engineering & Systems (ICCES), 2014*, pages 368–373. IEEE.

[Espana-Bonet and Fonollosa, 2016] Espana-Bonet, C. and Fonollosa, J. A. (2016). Automatic speech recognition with deep neural networks for impaired speech. In *International Conference on Advances in Speech and Language Technologies for Iberian Languages*, pages 97–107. Springer.

[Fan et al., 2015] Fan, J., Wade, J. W., Bian, D., Key, A. P., Warren, Z. E., Mion, L. C., and Sarkar, N. (2015). A step towards eeg-based brain computer interface for autism intervention. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3767–3770.

[Fernandes et al., 2020] Fernandes, F. G., Cardoso, A., and Lopes, R. d. A. (2020). Games applied to children with motor impairment using the myo wearable device. *Anais da Academia Brasileira de Ciências*, 92.

[Ferrero et al., 2017] Ferrero, M., West, G., and Vadillo, M. A. (2017). Is crossed laterality associated with academic achievement and intelligence? a systematic review and meta-analysis. *PLOS ONE*, 12(8):1–18.

[Garber, 2013] Garber, L. (2013). Gestural technology: Moving interfaces in a new direction [technology news]. *Computer*, 46(10):22–25.

[Ghotkar and Kharate, 2012] Ghotkar, A. S. and Kharate, G. K. (2012). Hand segmentation techniques to hand gesture recognition for natural human computer interaction. *International Journal of Human Computer Interaction (IJHCI)*, 3(1):15.

[Ghotkar and Kharate, 2014] Ghotkar, A. S. and Kharate, G. K. (2014). Study of vision based hand gesture recognition using indian sign language. *Computer*, 55:56.

[Ghotkar and Kharate, 2017] Ghotkar, A. S. and Kharate, G. K. (2017). Study of vision based hand gesture recognition using indian sign language. *International Journal on Smart Sensing and Intelligent Systems*, 7(1).

[Giancola et al., 2018] Giancola, S., Valenti, M., and Sala, R. (2018). *A survey on 3D cameras: Metrological comparison of time-of-flight, structured-light and active stereoscopy technologies*. Springer.

[Gillian and Paradiso, 2014] Gillian, N. E. and Paradiso, J. A. (2014). The gesture recognition toolkit. *Journal of Machine Learning Research*, 15(1):3483–3487.

[Gonzalez and Woods, 2006] Gonzalez, R. C. and Woods, R. E. (2006). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., USA.

[Granollers and Lorés, 2006] Granollers, T. and Lorés, J. (2006). Incorporation of users in the evaluation of usability by cognitive walkthrough. In *HCI related papers of Interacción 2004*, pages 243–255.

[Gunes and Piccardi, 2006] Gunes, H. and Piccardi, M. (2006). A bimodal face and body gesture database for automatic analysis of human nonverbal affective behavior. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 1148–1153.

[Guo and Zhang, 2019] Guo, G. and Zhang, N. (2019). A survey on deep learning based face recognition. *Computer Vision and Image Understanding*, 189:102805.

[Gupta et al., 2013] Gupta, N., Xu, W., and Kamboj, D. (2013). Depth-based segmentation—a review. In *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*, pages 1–9.

[Gürkök et al., 2012] Gürkök, H., Nijholt, A., and Poel, M. (2012). Brain-computer interface games: Towards a framework. In *International Conference on Entertainment Computing*, pages 373–380.

[Gusenbauer and Haddaway, 2020] Gusenbauer, M. and Haddaway, N. R. (2020). Which academic search systems are suitable for systematic reviews or meta-analyses? evaluating retrieval qualities of google scholar, pubmed, and 26 other resources. *Research Synthesis Methods*, 11(2):181–217.

[Hart et al., 2000] Hart, P. E., Stork, D. G., and Duda, R. O. (2000). *Pattern classification*. Wiley Hoboken.

[Haurum et al., 2021] Haurum, J. B., Allahham, M. M., Lynge, M. S., Henriksen, K. S., Nikolov, I. A., and Moeslund, T. B. (2021). Sewer defect classification using synthetic point clouds. In *VISIGRAPP (5: VISAPP)*, pages 891–900.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

[He et al., 2017] He, S., Yang, C., Wang, M., Cheng, L., and Hu, Z. (2017). Hand gesture recognition using myo armband. In *2017 Chinese Automation Congress (CAC)*, pages 4850–4855. IEEE.

[Hinton et al., 2012] Hinton, G., Srivastava, N., Swersky, K., Tieleman, T., and Mohamed, A. (2012). Coursera: Neural networks for machine learning. *Lecture 9c: Using Noise as a Regularizer*.

[Hsiao and Chen, 2016] Hsiao, H.-S. and Chen, J.-C. (2016). Using a gesture interactive game-based learning approach to improve preschool children's learning performance and motor skills. *Computers & Education*, 95:151–162.

[Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708.

[Huang et al., 2015] Huang, J., Zhou, W., Li, H., and Li, W. (2015). Sign language recognition using real-sense. In *Signal and Information Processing (ChinaSIP), 2015 IEEE China Summit and International Conference on*, pages 166–170. IEEE.

[Hwang et al., 2015] Hwang, I., Kim, H.-C., Cha, J., Ahn, C., Kim, K., and Park, J.-I. (2015). A gesture based tv control interface for visually impaired: Initial design and user study. In *21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV), 2015*, pages 1–5. IEEE.

[Iidal et al., 2017] Iidal, Y., Tsutsumi, D., Saeki, S., Ootsuka, Y., Hashimoto, T., and Horie, R. (2017). The effect of immersive head mounted display on a brain computer interface game. In *Advances in Affective and Pleasurable Design*, pages 211–219.

[Isard and Blake, 1998] Isard, M. and Blake, A. (1998). Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28.

[Jain, 2010] Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.

[Jain et al., 2012] Jain, S., Tamersoy, B., Zhang, Y., Aggarwal, J., and Orvalho, V. (2012). An interactive game for teaching facial expressions to children with autism spectrum disorders. In *5th International Symposium on Communications Control and Signal Processing (ISCCSP), 2012*, pages 1–4. IEEE.

[Jamaluddin et al., 2016] Jamaluddin, A. Z., Mazhar, O., Morel, O., Seulin, R., and Fofi, D. (2016). Design and calibration of an omni-rgb+ d camera. In *13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2016*, pages 386–387. IEEE.

[Janocha and Czarnecki, 2017] Janocha, K. and Czarnecki, W. M. (2017). On loss functions for deep neural networks in classification. *arXiv preprint arXiv:1702.05659*.

[Jeong et al., 2016] Jeong, S., Ju, H., Choi, H.-R., and Kim, T. (2016). Customized gesture recognition for educational games. *TECHART: Journal of Arts and Imaging Science*, 3(2):1–5.

[Ji et al., 2012] Ji, S., Xu, W., Yang, M., and Yu, K. (2012). 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231.

[Joselli et al., 2012] Joselli, M., da Silva, J. R., Zamith, M., Clua, E., Pelegrino, M., Mendonça, E., and Soluri, E. (2012). An architecture for game interaction using mobile. In *IEEE International Games Innovation Conference (IGIC), 2012*, pages 1–5. IEEE.

[Joshi et al., 2015] Joshi, A., Kale, S., Chandel, S., and Pal, D. K. (2015). Likert scale: Explored and explained. *British Journal of Applied Science & Technology*, 7(4):396.

[Kaehler and Bradski, 2016] Kaehler, A. and Bradski, G. (2016). *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc.

[Katsamanis et al., 2014] Katsamanis, A., Rodomagoulakis, I., Potamianos, G., Maragos, P., and Tsiami, A. (2014). Robust far-field spoken command recognition for home automation combining adaptation and multichannel processing. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5547–5551. IEEE.

[Ker et al., 2017] Ker, J., Wang, L., Rao, J., and Lim, T. (2017). Deep learning applications in medical image analysis. *IEEE Access*, 6:9375–9389.

[Ketkar, 2017] Ketkar, N. (2017). Stochastic gradient descent. In *Deep learning with Python*, pages 113–132.

[Kılıç et al., 2019] Kılıç, A. O., Sari, E., Yucel, H., Oğuz, M. M., Polat, E., Acoglu, E. A., and Senel, S. (2019). Exposure to and use of mobile devices in children aged 1–60 months. *European Journal of Pediatrics*, 178(2):221–227.

[Kim and Kim, 2006] Kim, D. and Kim, D. (2006). An intelligent smart home control using body gestures. In *2006 International Conference on Hybrid Information Technology*, volume 2, pages 439–446. IEEE.

[Kim et al., 2018] Kim, J.-H., Hong, G.-S., Kim, B.-G., and Dogra, D. P. (2018). deepgesture: Deep learning-based gesture recognition scheme using motion sensors. *Displays*, 55:38–45.

[Kim et al., 2019] Kim, M., Cho, J., Lee, S., and Jung, Y. (2019). Imu sensor-based hand gesture recognition for human-machine interfaces. *Sensors*, 19(18):3827.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kocmi, 2020] Kocmi, T. (2020). Exploring benefits of transfer learning in neural machine translation. *arXiv preprint arXiv:2001.01622*.

[Konstantinova et al., 2014] Konstantinova, J., Jiang, A., Althoefer, K., Dasgupta, P., and Nanayakkara, T. (2014). Implementation of tactile sensing for palpation in robot-assisted minimally invasive surgery: A review. *IEEE Sensors Journal*, 14(8):2490–2501.

[Kramer, 2016] Kramer, O. (2016). Scikit-learn. In *Machine learning for evolution strategies*, pages 45–53.

[Kraus et al., 2020] Kraus, S., Li, H., Kang, Q., Westhead, P., and Tiberius, V. (2020). The sharing economy: A bibliometric analysis of the state-of-the-art. *International Journal of Entrepreneurial Behavior & Research*.

[Kristensen et al., 2006] Kristensen, F., Nilsson, P., and Öwall, V. (2006). Background segmentation beyond rgb. In *Asian Conference on Computer Vision*, pages 602–612. Springer.

[Kumar and Shimi, 2015] Kumar, M. and Shimi, S. (2015). Voice recognition based home automation system for paralyzed people. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, 4(10).

[Kushniruk et al., 2015] Kushniruk, A. W., Monkman, H., Tuden, D., Bellwood, P., Borycki, E. M., et al. (2015). Integrating heuristic evaluation with cognitive walkthrough: development of a hybrid usability inspection method. In *Driving Quality in Informatics: Fulfilling the Promise*, pages 221–225.

[Kwon et al., 2014] Kwon, J., Lee, H. S., Park, F. C., and Lee, K. M. (2014). A geometric particle filter for template-based visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):625–643.

[Ladikos et al., 2007a] Ladikos, A., Benhimane, S., and Navab, N. (2007a). High performance model-based object detection and tracking. In *International Conference on Computer Vision and Computer Graphics*, pages 191–204. Springer.

[Ladikos et al., 2007b] Ladikos, A., Benhimane, S., and Navab, N. (2007b). A real-time tracking system combining template-based and feature-based approaches. In *Proceedings of the Second International Conference on Computer Vision Theory and Applications*, pages 325–332. Citeseer.

[Lamere et al., 2003] Lamere, P., Kwok, P., Gouvea, E., Raj, B., Singh, R., Walker, W., Warmuth, M., and Wolf, P. (2003). The cmu sphinx-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, volume 1, pages 2–5.

[Lang et al., 2012] Lang, S., Block, M., and Rojas, R. (2012). Sign language recognition using kinect. In *International Conference on Artificial Intelligence and Soft Computing*, pages 394–402. Springer.

[Laugwitz et al., 2008] Laugwitz, B., Held, T., and Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In *Symposium of the Austrian HCI and Usability Engineering Group*, pages 63–76. Springer.

[Lee and Kawahara, 2009] Lee, A. and Kawahara, T. (2009). Recent development of open-source speech recognition engine julius. In *Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference*, pages 131–137.

[Lee et al., 2020a] Lee, A.-r., Cho, Y., Jin, S., and Kim, N. (2020a). Enhancement of surgical hand gesture recognition using a capsule network for a contactless interface in the operating room. *Computer Methods and Programs in Biomedicine*, 190:105385.

[Lee et al., 2020b] Lee, C., Kim, J., Cho, S., Kim, J., Yoo, J., and Kwon, S. (2020b). Development of real-time hand gesture recognition for tabletop holographic display interaction using azure kinect. *Sensors*, 20(16):4566.

[Lee et al., 2013] Lee, S.-H., Sohn, M.-K., Kim, D.-J., Kim, B., and Kim, H. (2013). Smart tv interaction system using face and hand gesture recognition. In *2013 IEEE International Conference on Consumer Electronics (ICCE)*, pages 173–174. IEEE.

[Lee et al., 2014] Lee, W.-P., Kaoli, C., and Huang, J.-Y. (2014). A smart tv system with body-gesture control, tag-based rating and context-aware recommendation. *Knowledge-Based Systems*, 56:167–178.

[Lewis and Wharton, 1997] Lewis, C. and Wharton, C. (1997). Cognitive walkthroughs. In *Handbook of Human-Computer Interaction*, pages 717–732.

[Li et al., 2019] Li, G., Tang, H., Sun, Y., Kong, J., Jiang, G., Jiang, D., Tao, B., Xu, S., and Liu, H. (2019). Hand gesture recognition based on convolution neural network. *Cluster Computing*, 22(2):2719–2729.

[Li, 2017] Li, H. (2017). Deep learning for natural language processing: advantages and challenges. *National Science Review*.

[Li and Zhang, 2019] Li, J. and Zhang, D. (2019). Face gesture recognition based on clustering algorithm. In *2019 Chinese Control And Decision Conference (CCDC)*, pages 2008–2012. IEEE.

[Li et al., 2020a] Li, L., Qin, S., Lu, Z., Xu, K., and Hu, Z. (2020a). One-shot learning gesture recognition based on joint training of 3d resnet and memory module. *Multimedia Tools and Applications*, 79(9):6727–6757.

[Li et al., 2020b] Li, R., Zou, K., and Wang, W. (2020b). Application of human body gesture recognition algorithm based on deep learning in non-contact human body measurement. *Journal of Ambient Intelligence and Humanized Computing*.

[Li, 2012] Li, Y. (2012). Hand gesture recognition using kinect. In *2012 IEEE International Conference on Computer Science and Automation Engineering*, pages 196–199. IEEE.

[Liao et al., 2018] Liao, B., Li, J., Ju, Z., and Ouyang, G. (2018). Hand gesture recognition with generalized hough transform and dc-cnn using realsense. In *2018 Eighth International Conference on Information Science and Technology (ICIST)*, pages 84–90. IEEE.

[Lim et al., 2015] Lim, S.-C., Lee, H.-K., and Park, J. (2015). Role of combined tactile and kinesthetic feedback in minimally invasive surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 11(3):360–374.

[Liu et al., 2017] Liu, J., Fujii, R., Tateyama, T., Iwamoto, Y., and Chen, Y. (2017). Kinect-based gesture recognition for touchless visualization of medical images. *International Journal of Computer and Electrical Engineering*, 9(2):421–429.

[Liu et al., 2018] Liu, J., Yuan, Y., Zhou, Y., Zhu, X., and Syed, T. N. (2018). Experiments and analysis of close-shot identification of on-branch citrus fruit with realsense. *Sensors*, 18(5):1510.

[Lodhi and Kang, 2019] Lodhi, B. and Kang, J. (2019). Multipath-densenet: A supervised ensemble architecture of densely connected convolutional networks. *Information Sciences*, 482:63–72.

[Lopes et al., 2017] Lopes, A. T., de Aguiar, E., De Souza, A. F., and Oliveira-Santos, T. (2017). Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. *Pattern Recognition*, 61:610–628.

[López-Fernández et al., 2016] López-Fernández, M. C., Serrano-Bedia, A. M., and Pérez-Pérez, M. (2016). Entrepreneurship and family firm research: A bibliometric analysis of an emerging field. *Journal of Small Business Management*, 54(2):622–639.

[Lu et al., 2016] Lu, W., Tong, Z., and Chu, J. (2016). Dynamic hand gesture recognition with leap motion controller. *IEEE Signal Processing Letters*, 23(9):1188–1192.

[Lucey et al., 2010] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 94–101.

[Luna-Romero et al., 2017] Luna-Romero, S., Delgado-Espinoza, P., Rivera-Calle, F., and Serpa-Andrade, L. (2017). A domotics control tool based on myo devices and neural networks. In *International Conference on Applied Human Factors and Ergonomics*, pages 540–548.

[Lynch, 2014] Lynch, S. (2014). A tutorial introduction to matlab. In *Dynamical Systems with Applications using MATLAB®*, pages 1–14.

[Ma et al., 2018] Ma, C., Zhang, Y., Wang, A., Wang, Y., and Chen, G. (2018). Traffic command gesture recognition for virtual urban scenes based on a spatiotemporal convolution neural network. *ISPRS International Journal of Geo-Information*, 7(1):37.

[Mahmoud et al., 2008] Mahmoud, T. M. et al. (2008). A new fast skin color detection technique. *World Academy of Science, Engineering and Technology*, 43:501–505.

[Małecki et al., 2020] Małecki, K., Nowosielski, A., and Kowalicki, M. (2020). Gesture-based user interface for vehicle on-board system: A questionnaire and research approach. *Applied Sciences*, 10(18):6620.

[Maraqa and Abu-Zaiter, 2008] Maraqa, M. and Abu-Zaiter, R. (2008). Recognition of arabic sign language (arsl) using recurrent neural networks. In *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the*, pages 478–481. IEEE.

[Marchesi and Riccò, 2013] Marchesi, M. and Riccò, B. (2013). Bravo: a brain virtual operator for education exploiting brain-computer interfaces. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 3091–3094.

[Martinovikj et al., 2013] Martinovikj, D., Mihova, M., and Jovanov, M. (2013). Analysis of the problem of macedonian folk dance recognition. In *III International Conference on Computational Intelligence and Information Technology*, pages 165–168.

[Matarneh et al., 2017] Matarneh, R., Maksymova, S., Lyashenko, V., and Belova, N. (2017). Speech recognition systems: A comparative review. *IOSR Journal of Computer Engineering*, 19(5):71–79.

[Meng et al., 2013] Meng, M., Fallavollita, P., Blum, T., Eck, U., Sandor, C., Weidert, S., Waschke, J., and Navab, N. (2013). Kinect for interactive ar anatomy learning. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2013*, pages 277–278.

[Mitra and Acharya, 2007] Mitra, S. and Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324.

[Mittal et al., 2019] Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., and Chaudhuri, B. B. (2019). A modified lstm model for continuous sign language recognition using leap motion. *IEEE Sensors Journal*, 19(16):7056–7063.

[Moeslund et al., 2006] Moeslund, T. B., Hilton, A., and Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126.

[Mohandes et al., 2014] Mohandes, M., Aliyu, S., and Deriche, M. (2014). Arabic sign language recognition using the leap motion controller. In *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, pages 960–965. IEEE.

[Molinero, 2010] Molinero, G. (2010). Segmentacion de imagenes en color basada en el crecimiento de regiones. `http://bibing.us.es/proyectos/abreproy/11875/`. Consultado 9-6-2021.

[Morris and Van der Veer Martens, 2008] Morris, S. A. and Van der Veer Martens, B. (2008). Mapping research specialties. *Annual Review of Information Science and Technology*, 42(1):213–295.

[Müller, 2007] Müller, M. (2007). Dynamic time warping. *Information Retrieval for Music and Motion*, pages 69–84.

[Narayana et al., 2019] Narayana, S., Prasad, R. V., and Warmerdam, K. (2019). Mind your thoughts: Bci using single eeg electrode. *IET Cyber-Physical Systems: Theory & Applications*, 4(2):164–172.

[Navalyal and Gavas, 2014] Navalyal, G. U. and Gavas, R. D. (2014). A dynamic attention assessment and enhancement tool using computer graphics. *Human-centric Computing and Information Sciences*, 4(1):11.

[Ng and Ranganath, 2000] Ng, C. W. and Ranganath, S. (2000). Gesture recognition via pose classification. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 3, pages 699–704.

[Ng and Ranganath, 2002] Ng, C. W. and Ranganath, S. (2002). Real-time gesture recognition system and application. *Image and Vision computing*, 20(13-14):993–1007.

[Nguyen et al., 2018] Nguyen, L. D., Lin, D., Lin, Z., and Cao, J. (2018). Deep cnns for microscopic image classification by exploiting transfer learning and feature concatenation. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5.

[Niu and Suen, 2012] Niu, X.-X. and Suen, C. Y. (2012). A novel hybrid cnn–svm classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325.

[Nouar et al., 2006] Nouar, O.-D., Ali, G., and Raphael, C. (2006). Improved object tracking with camshift algorithm. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 2, pages II–II. IEEE.

[of human-system interaction (Subcommittee), 1998] of human-system interaction (Subcommittee), I. S. . E. (1998). *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs): Guidance on Usability*. International Organization for Standardization.

[Ofodile et al., 2019] Ofodile, I., Helmi, A., Clapés, A., Avots, E., Peensoo, K. M., Valdma, S.-M., Valdmann, A., Valtna-Lukner, H., Omelkov, S., Escalera, S., et al. (2019). Action recognition using single-pixel time-of-flight detection. *Entropy*, 21(4):414.

[Okamura, 2009] Okamura, A. M. (2009). Haptic feedback in robot-assisted minimally invasive surgery. *Current Opinion in Urology*, 19(1):102.

[Olivas et al., 2009] Olivas, E. S., Guerrero, J. D. M., Martinez-Sober, M., Magdalena-Benedito, J. R., Serrano, L., et al. (2009). *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques*. IGI Global.

[Ordóñez and Roggen, 2016] Ordóñez, F. J. and Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115.

[Ousmer et al., 2019] Ousmer, M., Vanderdonckt, J., and Buraga, S. (2019). An ontology for reasoning on body-based gestures. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pages 1–6.

[Pamungkas and Ward, 2016] Pamungkas, D. S. and Ward, K. (2016). Electro-tactile feedback system to enhance virtual reality experience. *International Journal of Computer Theory and Engineering*, 8(6):465–470.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pérez Gutiérrez and Córdova-Cruzatty, 2020] Pérez Gutiérrez, M. F. and Córdova-Cruzatty, A. C. (2020). Obstacle detection algorithm by means of images with a zed camera using ros software in a drone. In *International Conference on Applied Technologies*, pages 458–466.

[Perret and Vander Poorten, 2018] Perret, J. and Vander Poorten, E. (2018). Touching virtual reality: a review of haptic gloves. In *ACTUATOR 2018; 16th International Conference on New Actuators*, pages 1–5.

[Piccardi, 2004] Piccardi, M. (2004). Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics, 2004*, volume 4, pages 3099–3104.

[Piedra et al., 2016] Piedra, J. A., Ojeda-Castelo, J. J., Quero-Valenzuela, F., and Piedra-Fdez, I. (2016). Virtual environment for the training of the hands in minimally invasive thoracic surgery. In *2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, pages 1–4.

[Plouffe and Cretu, 2016] Plouffe, G. and Cretu, A.-M. (2016). Static and dynamic hand gesture recognition in depth data using dynamic time warping. *IEEE Transactions on Instrumentation and Measurement*, 65(2):305–316.

[Poppe, 2007] Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer vision and image understanding*, 108(1):4–18.

[Potter et al., 2013] Potter, L. E., Araullo, J., and Carter, L. (2013). The leap motion controller: a view on sign language. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, pages 175–178.

[Povey et al., 2011] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.

[Pradipa, 2014] Pradipa, K. (2014). Hand gesture recognition - analysis of various techniques, methods and their algorithms. *International Jorunal of Innovative Research in Science, Engineering and Technology*, 3:2003–2010.

[Prince et al., 2015] Prince, D., Edmonds, M., Sutter, A., Cusumano, M., Lu, W., and Asari, V. (2015). Brain machine interface using emotiv epoc to control robai cyton robotic arm. In *Aerospace and Electronics Conference (NAECON), 2015 National*, pages 263–266. IEEE.

[Pu et al., 2013] Pu, Q., Gupta, S., Gollakota, S., and Patel, S. (2013). Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, pages 27–38.

[Pu et al., 2015] Pu, Q., Gupta, S., Gollakota, S., and Patel, S. (2015). Gesture recognition using wireless signals. *GetMobile: Mobile Computing and Communications*, 18(4):15–18.

[Qamar et al., 2015] Qamar, A. M., Khan, A. R., Husain, S. O., Rahman, M. A., and Baslamah, S. (2015). A multi-sensory gesture-based occupational therapy environment for controlling home appliances. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 671–674.

[Qiu-yu et al., 2015] Qiu-yu, Z., Jun-chi, L., Mo-yi, Z., Hong-xiang, D., and Lu, L. (2015). Hand gesture segmentation method based on ycbcr color space and k-means clustering. *Interaction*, 8:106–116.

[Qu et al., 2013] Qu, J., Song, Y., and Wei, Y. (2013). Applying design patterns in game programming. In *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*, page 1.

[Quesada Elvira, 2014] Quesada Elvira, M. (2014). Interacción persona-ordenador mediante la captura del movimiento ocular utilizando la herramienta de eye-tracking tobii. Master's thesis, Universidad de Castilla-La Mancha.

[Rada-Vilela, 2018] Rada-Vilela, J. (2018). The fuzzylite libraries for fuzzy logic control.

[Rahman et al., 2009] Rahman, A., Hossain, M. A., Parra, J., and El Saddik, A. (2009). Motion-path based gesture interaction with smart home services. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 761–764.

[Rautaray and Agrawal, 2015] Rautaray, S. S. and Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54.

[Reichstein et al., 2019] Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204.

[Ruder, 2016] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[Rudrapal et al., 2012] Rudrapal, D., Das, S., Debbarma, S., Kar, N., and Debbarma, N. (2012). Voice recognition and authentication as a proficient biometric tool and its application in online exam for ph people. *International Journal of Computer Applications*, 39(12):6–12.

[Ruzaij et al., 2016] Ruzaij, M. F., Neubert, S., Stoll, N., and Thurow, K. (2016). Hybrid voice controller for intelligent wheelchair and rehabilitation robot using voice recognition and embedded technologies. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 20(4):615–622.

[Saini and Goel, 2019] Saini, M. K. and Goel, N. (2019). How smart are smart classrooms? a review of smart classroom technologies. *ACM Computing Surveys (CSUR)*, 52(6):1–28.

[Samir et al., 2015] Samir, M., Golkar, E., and Rahni, A. A. A. (2015). Comparison between the kinect™ v1 and kinect™ v2 for respiratory motion tracking. In *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 150–155.

[Sandler et al., 2018] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

[Sangeetha et al., 2015] Sangeetha, S. B. et al. (2015). Intelligent interface based speech recognition for home automation using android application. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–11. IEEE.

[Sarkar et al., 2018] Sarkar, D., Bali, R., and Ghosh, T. (2018). *Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras*. Packt Publishing Ltd.

[Scheggi et al., 2015] Scheggi, S., Meli, L., Pacchierotti, C., and Prattichizzo, D. (2015). Touch the virtual reality: using the leap motion controller for hand tracking and wearable tactile devices for immersive haptic rendering. In *ACM SIGGRAPH 2015 Posters*.

[Schrepp et al., 2014] Schrepp, M., Hinderks, A., and Thomaschewski, J. (2014). Applying the user experience questionnaire (ueq) in different evaluation scenarios. In *International Conference of Design, User Experience, and Usability*, pages 383–392.

[Schrepp et al., 2017a] Schrepp, M., Hinderks, A., and Thomaschewski, J. (2017a). Construction of a benchmark for the user experience questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(4):40–44.

[Schrepp et al., 2017b] Schrepp, M., Hinderks, A., and Thomaschewski, J. (2017b). Design and evaluation of a short version of the user experience questionnaire (UEQ-S). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(6):103–108.

[Shanas et al., 2017] Shanas, E., Townsend, P., Wedderburn, D., Friis, H. K., Milhoj, P., and Stehouwer, J. (2017). *Old people in three industrial societies*.

[Shinde et al., 2016] Shinde, R. V., Shimpi, S. M., Lanjewar, P. S., Nivangune, P. A., Mundkar, D. S., and Sonawane, A. R. (2016). Vision based hand gesture recognition for real time home automation application. *International Journal of Engineering Science*, 3176.

[Signer et al., 2007] Signer, B., Kurmann, U., and Norrie, M. (2007). igesture: a general gesture recognition framework. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 954–958. IEEE.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Singh et al., 2015a] Singh, G., Nelson, A., Robucci, R., Patel, C., and Banerjee, N. (2015a). Demo abstract: Inviz: Low-power personalized gesture recognition using wearable textile capacitive sensor arrays. In *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), 2015*, pages 223–224. IEEE.

[Singh et al., 2015b] Singh, G., Nelson, A., Robucci, R., Patel, C., and Banerjee, N. (2015b). Inviz: Low-power personalized gesture recognition using wearable textile capacitive sensor arrays. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pages 198–206. IEEE.

[Singh and Lone, 2020] Singh, H. and Lone, Y. A. (2020). *Deep Neuro-Fuzzy Systems with Python*. Springer.

[Sinha et al., 2019] Sinha, K., Kumari, R., Priya, A., and Paul, P. (2019). A computer vision-based gesture recognition using hidden markov model. In *Innovations in Soft Computing and Information Technology*, pages 55–67.

[Sreeja et al., 2016] Sreeja, S., Joshi, V., Samima, S., Saha, A., Rabha, J., Cheema, B. S., Samanta, D., and Mitra, P. (2016). BCI augmented text entry mechanism for people with special needs. In *International Conference on Intelligent Human Computer Interaction*, pages 81–93.

[Starner et al., 2000] Starner, T., Auxier, J., Ashbrook, D., and Gandy, M. (2000). The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In *The Fourth International Symposium on Wearable Computers*, pages 87–94.

[Steich et al., 2016] Steich, K., Kamel, M., Beardsley, P., Obrist, M. K., Siegwart, R., and Lachat, T. (2016). Tree cavity inspection using aerial robots. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4856–4862. IEEE.

[Sun et al., 2018] Sun, B., Cao, S., He, J., and Yu, L. (2018). Affect recognition from facial movements and body gestures by hierarchical deep spatio-temporal features and fusion strategy. *Neural Networks*, 105:36–51.

[Sun and Lv, 2019] Sun, X. and Lv, M. (2019). Facial expression recognition based on a hybrid model combining deep and shallow features. *Cognitive Computation*, 11(4):587–597.

[Swarnkar and Ambhaikar, 2019] Swarnkar, S. K. and Ambhaikar, A. (2019). Improved convolutional neural network based sign language recognition. *International Journal of Advanced Science and Technology*, 27:302 – 317.

[Szegedy et al., 2017] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI Conference on Artificial Intelligence*.

[Szegedy et al., 2016] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

[Tadić et al., 2021] Tadić, V., Odry, Á., Burkus, E., Kecskés, I., Király, Z., Vízvári, Z., Tóth, A., and Odry, P. (2021). Application of the zed depth sensor for painting robot vision system development. *IEEE Access*, 9:117845–117859.

[Tateno et al., 2019] Tateno, S., Zhu, Y., and Meng, F. (2019). Hand gesture recognition system for in-car device control based on infrared array sensor. In *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 701–706. IEEE.

[Terán-Yépez et al., 2020] Terán-Yépez, E., Marín-Carrillo, G. M., del Pilar Casado-Belmonte, M., and de las Mercedes Capobianco-Uriarte, M. (2020). Sustainable entrepreneurship: Review of its evolution and new trends. *Journal of Cleaner Production*, 252:119742.

[Terrillon et al., 2000] Terrillon, J.-C., Shirazi, M. N., Fukamachi, H., and Akamatsu, S. (2000). Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000*, pages 54–61. IEEE.

[Tolentino et al., 2019] Tolentino, R. E., Guinto, P. M. F., and Maypa, D. Y. B. (2019). Recognition of different emergency situation through body gesture using microsoft kinect sensor. In *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pages 1–5.

[Tölgyessy et al., 2021] Tölgyessy, M., Dekan, M., Chovanec, L., and Hubinskỳ, P. (2021). Evaluation of the azure kinect and its comparison to kinect v1 and kinect v2. *Sensors*, 21(2):413.

[Tsagaris and Manitsaris, 2013] Tsagaris, A. and Manitsaris, S. (2013). Colour space comparison for skin detection in finger gesture recognition. *International Journal of Advances in Engineering & Technology*, 6(4):1431.

[Tseng et al., 2014] Tseng, C. M., Lai, C. L., Erdenetsogt, D., and Chen, Y. F. (2014). A microsoft kinect based virtual rehabilitation system. In *Computer, Consumer and Control (IS3C), 2014 International Symposium on*, pages 934–937. IEEE.

[Vaitkevičius et al., 2019] Vaitkevičius, A., Taroza, M., Blažauskas, T., Damaševičius, R., Maskeliūnas, R., and Woźniak, M. (2019). Recognition of american sign language gestures in a virtual reality using leap motion. *Applied Sciences*, 9(3):445.

[Vehlen et al., 2021] Vehlen, A., Spenthof, I., Tönsing, D., Heinrichs, M., and Domes, G. (2021). Evaluation of an eye tracking setup for studying visual attention in face-to-face conversations. *Scientific Reports*, 11(1):1–16.

[Velayudhan and Gireeshkumar, 2015] Velayudhan, A. and Gireeshkumar, T. (2015). An autonomous obstacle avoiding and target recognition robotic system using kinect. In *Intelligent Computing, Communication and Devices*, pages 643–649.

[Voulodimos et al., 2018] Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*.

[Waltman and Noyons, 2018] Waltman, L. and Noyons, E. (2018). *Bibliometrics for research management and research evaluation.* Centre for Science and Technology Studies (University of Leiden).

[Wang and Zheng, 2013] Wang, H. and Zheng, H. (2013). *True Positive Rate*, pages 2302–2303. Springer New York, New York, NY.

[Wang and Deng, 2018] Wang, M. and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153.

[Wang et al., 2016] Wang, N., Broz, F., Di Nuovo, A., Belpaeme, T., and Cangelosi, A. (2016). A user-centric design of service robots speech interface for the elderly. In *Recent Advances in Nonlinear Speech Processing*, pages 275–283.

[Weichert et al., 2013] Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D. (2013). Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393.

[Weng et al., 2010] Weng, C., Li, Y., Zhang, M., Guo, K., Tang, X., and Pan, Z. (2010). Robust hand posture recognition integrating multi-cue hand tracking. In *International Conference on Technologies for E-Learning and Digital Entertainment*, pages 497–508. Springer.

[Wong et al., 2016] Wong, S. C., Gatt, A., Stamatescu, V., and McDonnell, M. D. (2016). Understanding data augmentation for classification: when to warp? In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6.

[Wu et al., 2016] Wu, D., Pigou, L., Kindermans, P. J., Le, N. D. H., Shao, L., Dambre, J., and Odobez, J. M. (2016). Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1583–1597.

[Wurm et al., 2019] Wurm, M., Stark, T., Zhu, X. X., Weigand, M., and Taubenböck, H. (2019). Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, 150:59–69.

[Xu, 2017] Xu, P. (2017). A real-time hand gesture recognition and human-computer interaction system. *arXiv preprint arXiv:1704.07296*.

[Yang et al., 2019] Yang, J., Zou, H., Zhou, Y., and Xie, L. (2019). Learning gestures from wifi: A siamese recurrent convolutional architecture. *IEEE Internet of Things Journal*, 6(6):10763–10772.

[Yang et al., 2002] Yang, M.-H., Ahuja, N., and Tabb, M. (2002). Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1061–1074.

[Yang and Zhu, 2017] Yang, S. and Zhu, Q. (2017). Continuous chinese sign language recognition with CNN-LSTM. In *Ninth International Conference on Digital Image Processing (ICDIP 2017)*, volume 10420, page 104200F. International Society for Optics and Photonics.

[Yang et al., 2015] Yang, W., Tao, J., Xi, C., and Ye, Z. (2015). Sign language recognition system based on weighted hidden markov model. In *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pages 449–452. IEEE.

[Yao et al., 2005] Yao, H.-Y., Hayward, V., and Ellis, R. E. (2005). A tactile enhancement instrument for minimally invasive surgery. *Computer Aided Surgery*, 10(4):233–239.

[Zafrulla et al., 2011] Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., and Presti, P. (2011). American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 279–286. ACM.

[Zaini et al., 2019] Zaini, N. A., Noor, S. F. M., and Wook, T. (2019). Evaluation of api interface design by applying cognitive walkthrough. *Evaluation*, 10(2).

[Zantalis et al., 2019] Zantalis, F., Koulouras, G., Karabetsos, S., and Kandris, D. (2019). A review of machine learning and iot in smart transportation. *Future Internet*, 11(4):94.

[Zhang et al., 2021] Zhang, C., Benz, P., Argaw, D. M., Lee, S., Kim, J., Rameau, F., Bazin, J.-C., and Kweon, I. S. (2021). Resnet or densenet? introducing dense shortcuts to resnet. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3550–3559.

[Zhang et al., 2020a] Zhang, W., Zhao, X., Zhao, L., Yin, D., Yang, G. H., and Beutel, A. (2020a). Deep reinforcement learning for information retrieval: Fundamentals and advances. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2468–2471.

[Zhang et al., 2020b] Zhang, Y., Huang, Y., Sun, X., Zhao, Y., Guo, X., Liu, P., Liu, C., and Zhang, Y. (2020b). Static and dynamic human arm/hand gesture capturing and recognition via multiinformation fusion of flexible strain sensors. *IEEE Sensors Journal*, 20(12):6450–6459.

[Zhang, 2012] Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10.

[Zhang et al., 2018] Zhang, Z., Tian, Z., and Zhou, M. (2018). Handsense: smart multimodal hand gesture recognition based on deep neural networks. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–16.

[Zhou et al., 2017] Zhou, F. Y., Jin, L., and Dong, J. (2017). Review of convolutional neural network. *Chinese Journal of Computers*, 40(6):1229–1251.

[Zhu et al., 2016] Zhu, G., Zhang, L., Mei, L., Shao, J., Song, J., and Shen, P. (2016). Large-scale isolated gesture recognition using pyramidal 3d convolutional networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 19–24.

[Zinchenko et al., 2017] Zinchenko, K., Wu, C., and Song, K. (2017). A study on speech recognition control for a surgical robot. *IEEE Transactions on Industrial Informatics*, 13(2):607–615.

[Zoghlami et al., 2019] Zoghlami, F., Heinrich, H., Schneider, G., and Hamdi, M. A. (2019). Tracking body motions in order to guide a robot using the time of flight technology. In *IPIN (Short Papers/Work-in-Progress Papers)*, pages 48–55.

[Zoph et al., 2018] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8697–8710.

[Zou et al., 2018a] Zou, H., Yang, J., Zhou, Y., Xie, L., and Spanos, C. J. (2018a). Robust wifi-enabled device-free gesture recognition via unsupervised adversarial domain adaptation. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8.

[Zou et al., 2018b] Zou, H., Zhou, Y., Yang, J., Jiang, H., Xie, L., and Spanos, C. J. (2018b). Wifi-enabled device-free gesture recognition for smart home automation. In *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, pages 476–481. IEEE.

[Zupic and Čater, 2015] Zupic, I. and Čater, T. (2015). Bibliometric methods in management and organization. *Organizational Research Methods*, 18(3):429–472.

Este documento ha sido generado usando LaTeX.

Todas las figuras y tablas de este documento son originales.

Un Modelo Inteligente de Interacción Natural Adaptativo basado en Visión Artificial

Juan Jesús Ojeda Castelo
Departamento de Informática
Grupo de Investigación de Informática Aplicada (TIC-211)
Universidad de Almería
Almería, febrero de 2022

http://acg.ual.es

FE DE ERRATAS

1. En la página 129, donde dice ver Anexo ??, debe decir ver Anexo A.