

UNIVERSIDAD DE ALMERIA

ESCUELA SUPERIOR DE INGENIERÍA

2D VISION SYSTEMS FOR AUTOMATED
MICRO-MANIPULATION:
COMPONENTS CLASSIFICATION METHODS

Curso 2017/2018

Alumno/a:

Marco Di Salvatore

Director/es:

Dr.D. Giovanni Legnani

Dr.D. Francisco Rodríguez Díaz





**UNIVERSITÀ
DEGLI STUDI
DI BRESCIA**

DIPARTIMENTO DI INGEGNERIA MECCANICA E INDUSTRIALE
CORSO DI LAUREA MAGISTRALE IN
INGEGNERIA DELL'AUTOMAZIONE INDUSTRIALE

**SISTEMI DI VISIONE 2D PER
MICROMANIPOLAZIONE AUTOMATIZZATA:
METODI DI CLASSIFICAZIONE
DI COMPONENTI**

Relatore:

Dott. Legnani Giovanni

Corelatori:

Dott.ssa Fassi Irene

Ing. Ruggeri Serena

Ing. Fontana Gianmauro

Dott. Francisco Rodríguez Díaz

Laureando:

Di Salvatore Marco

matr. **85447**

Anno Accademico 2016/2017

A Ciccio

Indice

Introduzione	ix
1 Sistemi di visione 2D in ambito micro	1
1.1 Tecnologie Micro	1
1.2 Sistemi di visione	3
1.3 Esempi di machine vision in ambito micro	7
1.3.1 Esempi di applicazione del metodo della classificazione	15
1.4 Software e librerie di machine vision	20
1.4.1 Open CV	20
1.4.2 Halcon	22
1.4.3 MATLAB	23
1.4.4 LabVIEW	25
2 Image processing e algoritmi di machine vision (MV)	26
2.1 Threshold	27
2.1.1 Impostazione manuale del threshold	28
2.1.2 Impostazione automatica del threshold	29
2.2 Operazioni morfologiche binarie	31
2.2.1 Erosione (Erosion)	32
2.2.2 Dilatazione (Dilation)	32
2.2.3 Apertura (Opening)	34
2.2.4 Chiusura (Closing)	34
2.2.5 Eliminazione particolari che toccano il bordo (Border function)	34
2.3 Particle Analysis	37
2.4 Edge detection	39
2.4.1 Misurazione	41
2.4.2 Rilevazione	42
2.4.3 Allineamento	42
2.5 Pattern Matching	43
2.5.1 Correlazione Incrociata Normalizzata	46
2.6 Geometric matching	48
2.7 Confronto dei metodi	54
3 Classificazione	55
3.1 Fasi della classificazione	56
3.1.1 Training	56
3.1.2 Classificazione	59

3.2	Parametri del classificatore	61
3.2.1	Feature: estrazione e tipi	61
3.2.2	Distance metrics	68
3.2.3	Classifier method	69
3.3	Valutazione del classificatore	72
3.3.1	Valutazione dei dati delle feature raccolte	72
3.3.2	Determinazione della qualità di un classificatore	74
3.3.3	Valutazione della classificazione	75
3.4	Requisiti fondamentali per la classificazione degli oggetti	79
3.4.1	Gli illuminatori	79
3.4.2	Tipo di oggetto	83
4	Sviluppo del Software di classificazione	85
4.1	Training nel caso del software Particle Classifier NI	87
4.1.1	Selezione dei parametri di training	87
4.1.2	Generazione delle classi ed associazione dei campioni	89
4.1.3	Gestione delle classi e dei campioni associati	91
4.2	Training nel caso del software Custom Particle Classifier PFV	93
4.2.1	Selezione dei parametri di training	93
4.2.2	Generazione delle classi ed associazione dei campioni	93
4.2.3	Gestione delle classi e dei campioni associati	94
4.3	Test del classificatore	95
4.4	Applicazione della classificazione	97
4.4.1	Scelta dell'immagine sorgente e del classificatore	97
4.4.2	Scelta degli oggetti da identificare	97
4.4.3	Cambiamento del method e del distance metric	99
5	Test del software	101
5.1	Esempio di classificatori Scale Dependent ed Independent	102
5.1.1	Classificatori Multiclasse	103
5.1.2	Classificatori Monoclasse	107
5.2	Esempio di classificatore Multiclasse Specifico	109
5.2.1	Campioni non estranei	111
5.2.2	Campioni estranei	112
5.3	Esempio di classificatore Multiclasse Generale	114
5.3.1	Campioni non estranei	116
5.3.2	Campioni estranei	117
5.4	Conclusioni sui classificatori Multiclasse	120
5.5	Esempio di classificatore Monoclasse	120
5.6	Esempio di classificatore Multiclasse Mirror Dependent	126
5.7	Esempio di classificazione On-line	129
	Conclusioni	135
	Bibliografia	138

Elenco delle figure

1.1 Esempi di componenti micro.	2
1.2 Confronto tra forze superficiali e la forza gravitazionale al variare del raggio dell'oggetto [25].	2
1.3 Parametri dei sistemi di visione [25].	5
1.4 Confronto tra tecnologia tradizionale ed embedding [14].	7
1.5 Immagine che rappresenta esempi di cave nel substrato e di un componente (0402) inserito all'interno di una di queste [14].	8
1.6 Modello CAD componente MEMS [10].	8
1.7 Cella di micro-manipolazione [10].	9
1.8 Illustrazione task di manipolazione [10].	10
1.9 Applicazione dell'indicatore di posizionamento sulla microparte. I punti (p) sono usati per stimare la proiezione del modello CAD (PM) come mostrato nell'immagine [10].	11
1.10 Assemblaggio riuscito dei due micro-componenti A e B : (a) inizializzazione della movimentazione con rilevazione del componente e proiezione del modello CAD; (b)-(d) posizioni intermedie di A ; (f) inizializzazione della movimentazione con rilevazione del componente e proiezione del modello CAD di B ; (g) e (h) posizioni intermedie di B ; (i) una vista ravvicinata della micro-struttura 3D [10].	11
1.11 Assi degli embrioni di Zebrafish (EP1, EP2 e EP3) e micropipetta (MP) [39].	12
1.12 Rilevazione della deformazione degli embrioni di zebrafish durante la micro-manipolazione [27].	13
1.13 Rilevazione dei DSM e della localizzazione dell'obbiettivo [22].	14
1.14 Rilevazione della micropipetta durante l'occlusione [22].	15
1.15 Cella di micro-manipolazione di [41].	16
1.16 Errore assoluto di rilevazione della vista laterale del chip SU-8 al variare della distanza dal piano focale [41].	17
1.17 Rilevazione con algoritmo di ML del chip da vista laterale. (a)Il rilevatore localizza la faccia giusta del chip (rettangolo rosa) nonostante contenga difetti e sia parzialmente occluso e ruotato. (b)Il rilevatore localizza il chip a una distanza di $750\mu\text{m}$ dal piano focale [41].	17
1.18 Illustrazione degli eventi da classificare. (a)Evento di prelievo mentre il chip viene manipolato sopra il piano di lavoro. (b)Evento di incollaggio mentre il chip rimane attaccato a una delle punte della pinza durante il rilascio [40].	18
1.19 (a)Evento di presa ripreso con la vista dall'alto.(b)Evento di presa dalla vista laterale. (c)Evento di incollaggio ripreso con la vista dall'alto. (d)Evento di incollaggio dalla vista laterale [40].	19

1.20	Caratteristiche da 1 a 4. (a)La distanza minima d_{upper} e d_{lower} tra la punta alta e bassa della pinza dall'oggetto manipolato; $d_{opening}$ grandezza di apertura della pinza, tutte misurate grazie alla vista dall'alto. (b)Le differenti regioni che determinano il segno e il valore della feature d [40].	19
1.21	Logo di OpenCV.	20
1.22	Schema rappresentativo dell'organizzazione di OpenCV.	21
1.23	Logo di Halcon.	22
1.24	Logo di Matlab.	24
1.25	Logo di LabVIEW.	25
2.1	(a) Sensore di tipo CMOS. (b) Illustrazione del Filtro di Bayer. (c) Esempio di un'immagine in scala di grigio con il relativo istogramma. (d) Esempio di un'immagine a colori con il relativo istogramma RGB.	27
2.2	Illustrazione di un istogramma su cui si applica il Threshold.	28
2.3	Esempio di applicazione dell'operazione di Threshold su un'immagine.	29
2.4	Illustrazione dei dati utilizzati dai metodi di Auto Threshold: i è l'intensità di grigio, k è l'intensità di grigio scelta per il threshold, $h(i)$ è la funzione che descrive il numero di pixel ad ogni valore di grigio contenuti nell'immagine, N è il numero totale di livelli di scala di grigio (256 nel caso di un'immagine a 8 bit) e n è il numero totale di pixel in una immagine [38].	30
2.5	Esempio di Structuring Element 3x3.	32
2.6	Esempi di Structuring Element di differenti dimensioni.	33
2.7	Esempi di Structuring Element: valori strutturali differenti.	33
2.8	Esempi di forme del Structuring Element con differenti dimensioni; i pixel colorati sono quelli considerati durante l'elaborazione.	33
2.9	Esempio di applicazione dell'erosione.	35
2.10	Esempio di applicazione della dilatazione.	35
2.11	Esempio di applicazione dell'apertura.	35
2.12	Esempio di applicazione della chiusura.	36
2.13	Esempio di applicazione del reject border.	36
2.14	Esempio di applicazione di Particle Analysis.	38
2.15	Esempio di edge detection: (1)linea di ricerca e (2)bordo rilevato.	39
2.16	Modello utilizzato per la rilevazione di un edge. (1)Profilo della scala di grigi, (2)lunghezza dell'edge, (3)forza dell'edge e (4)localizzazione dell'edge.	40
2.17	Illustrazione del cambiamento dell' edge constant lungo il bordo dell'oggetto relativamente a differenti condizioni di illuminamento.	40
2.18	Esempi di polarità di bordo	41
2.19	Esempio di utilizzo dell'edge detection per la rilevazione dello spazio necessario al funzionamento della candela di innesco.	41
2.20	Esempio di utilizzo dell'edge detection per il controllo di conformità di un connettore.	42
2.21	Esempio di utilizzo dell'edge detection per la determinazione dell'orientamento di una brugola.	43
2.22	Esempio di utilizzo del pettern matching per il riconoscimento dei fori centraggio di una guarnizione.	44

2.23 Esempi di funzionamento del pattern matching con rilevazione di istanze multiple ruotate e scalate	45
2.24 Esempio di funzionamento del pattern matching con cambiamenti di illuminazione uniforme	45
2.25 Esempio di funzionamento del pattern matching nel caso in cui l'immagine sia disturbata o offuscata	45
2.26 Esempi di limitazioni del pattern matching	46
2.27 Illustrazione del calcolo della correlazione	47
2.28 Esempi di applicazione del geometric matching.	49
2.29 Immagini di oggetti che possono essere identificati attraverso il geometric matching.	49
2.30 Tipo di caratteristiche che il geometric matching può estrarre da un template con illustrazione della differenza tra edge-based e feature-based. (1) Curve, (2) caratteristica circolare, (3) caratteristica rettangolare, (4) caratteristica lineare, (5) angoli o spigoli.	50
2.31 Esempio di un riconoscimento di un oggetto ruotato e scalato (b) rispetto al template di riferimento (a).	52
2.32 Esempio di riconoscimento di un oggetto con diverse condizioni d'illuminamento (b)-(e) rispetto al template (a).	53
2.33 Esempio di inversione di contrasto (b) rispetto al template (a).	53
2.34 Esempio di un riconoscimento di un oggetto occluso.	53
2.35 Esempio di un riconoscimento dell'oggetto nonostante lo sfondo sia differente rispetto al template.	54
3.1 Campioni della classe "vite".	56
3.2 Campioni della classe "sfera".	56
3.3 Distribuzione dei campioni delle sfere (○) e delle viti (□) nel piano delle feature (circolarità in ascissa e allungamento in ordinata).	58
3.4 Illustrazione del processo di training e di test del classifier.	59
3.5 Campione sconosciuto da classificare.	60
3.6 Inserimento campione da classificare (X) nel piano delle feature.	60
3.7 Illustrazione del classificatore parallelo di NI Particle Classifier.	62
3.8 Illustrazione del Massimo Diametro di Feret (D).	64
3.9 Illustrazione della Convex Hull Area. In nero è rappresentato l'oggetto, mentre la linea rossa indica il più piccolo poligono convesso che contiene tutti i punti dell'oggetto.	64
3.10 Illustrazione dei valori B_R , B_L , B_T , B_B , W e H rispetto al profilo di un oggetto.	65
3.11 Esempio di applicazione del Convex's Perimeter Ratio applicato su una ruota dentata.	66
3.12 Esempio di applicazione del Convex's Area Ratio applicato su una ruota dentata.	66
3.13 Esempio in cui il Number of Hole permette di discriminare la ruota alleggerita da quella non alleggerita.	67
3.14 Esempio in cui l'Hole Area Ratio permette di discriminare le due ruote dentate in base alla parte di area totale dell'oggetto costituita da fori, tutto ciò nonostante il rumore che può essere percepito come tanti fori.	67

3.15 Esempio in cui l'unica caratteristica che contraddistingue le due sfere è l'area occupata da queste nell'immagine. In questo caso, lo Scale Area Ratio permette l'identificazione delle due classi.	68
3.16 Esempio di distribuzione dei campioni nel piano delle feature in cui il NN classifier funziona bene. \circ = Classe 1 e \square = Classe 2.	71
3.17 Esempio di distribuzione dei campioni nel piano delle feature in cui il KNN classifier funziona bene. \triangle = Classe 1, \circ = Classe 2 e \square = Classe 3.	71
3.18 Esempio di distribuzione dei campioni nel piano delle feature in cui il MMD classifier funziona bene. \circ = Classe 1, \square = Classe 2 e \triangle = Classe 3.	72
3.19 Esempio di distribuzione di probabilità che un certo numero di campioni, della classe A o B, assumano un certo valore della caratteristica.	77
3.20 Distribuzione della classification confidence: probabilità con cui un certo numero di campioni gli venga assegnato un certo classification score. Questa distribuzione è stata generata con un set di campioni di test, differenti dai campioni di training, di componenti meccanici.	78
3.21 Distribuzione della classification confidence: probabilità con cui un certo numero di campioni gli venga assegnato un certo classification score. Questa distribuzione è stata generata con un set di campioni di test, differenti dai campioni di training, di componenti elettronici.	78
3.22 Tipi di illuminatori [18].	80
3.23 Acquisizioni di immagini con un illuminatore Dark-field. (a) Il numero di serie inciso su una CPU, notare come i graffi sulla CPU vengono risaltati; (b) la stampa di caratteri Braile su una scatola di farmaci messi in evidenza.	80
3.24 Esempio di un blister illuminato con un sistema Bright-field front light. Si noti che le pastiglie all'interno della copertura sono visibili.	81
3.25 Bright-field Back Light: caso di riflessione che può accadere per oggetti con grandi spessori. (A) Telecamera; (B) oggetto; (C) Sorgente luminosa Bright-field back light; (D) raggio luminoso corretto che risalta i contorni dell'oggetto; (E) raggio luminoso riflesso dal bordo dell'oggetto.	81
3.26 Esempi di oggetti illuminati con un illuminatore Bright-field Back Light.	82
3.27 Esempi di oggetti da classificare illuminati con un Bright-field Back light.	82
3.28 (a-c)Esempi di oggetti non classificabili a causa della presenza di particelle interne. (d-f)Esempi di oggetti classificabili perché formati da una particella unica.	83
3.29 Esempio in cui non è possibile generare un classificatore in quanto le monete si distinguono per le caratteristiche interne legate alla scala di grigi.	84
3.30 Esempio in cui è possibile generare un classificatore, nonostante le caratteristiche interne, perché ciascuna moneta ha una dimensione diversa.	84
4.1 Front panel dei due software sviluppati.	86
4.2 Pagina del software NI dedicata alla selezione dei parametri del classificatore.	87
4.3 Pagina della funzione, dedicata al metodo NI, che permette l'acquisizione di nuovi campioni da un sistema di visione collegato al PC.	89
4.4 Pagina del programma NI in cui si effettua la creazione e il popolamento delle classi.	90
4.5 Pagina della funzione dedicata alla generazione dell'elenco delle classi.	90

4.6	Pagina del programma NI in cui si effettua la gestione dei campioni associati alle classi.	92
4.7	Pagina del software PFV dedicata alla selezione dei parametri del classificatore.	94
4.8	Pagina del programma PFV in cui si effettua la creazione e il popolamento delle classi.	95
4.9	Pagina dei programmi NI e PFV in cui si effettua il test del classificatore.	96
4.10	Pagina iniziale del modulo di classificazione.	98
4.11	Pagina di selezione degli oggetti da identificare del software di classificazione.	99
4.12	Pagina per il cambiamento del method e del distance metric del software di classificazione.	100
5.1	Oggetti utilizzati nella prova.	102
5.2	Immagine contenente differenti campioni su cui si è applicato il classificatore monoclasse.	108
5.3	Oggetti utilizzati per la fase di training.	109
5.4	Oggetti utilizzati come campioni estranei.	113
5.5	Immagine su cui è stato applicato il classificatore Multiclasse Specifico.	113
5.6	Oggetti della classe Vite.	114
5.7	Oggetti della classe Ruota Dentata.	115
5.8	Oggetti estranei.	118
5.9	Immagine utilizzata per la classificazione.	118
5.10	Oggetti utilizzati per la classificazione (estranei).	120
5.11	Oggetti utilizzati per creare i classificatori (non estranei).	121
5.12	Immagine contenente tutti gli oggetti a cui è stata applicata la classificazione monoclasse.	121
5.13	Oggetti rappresentanti le classi generate.	126
5.14	Oggetti utilizzati per la classificazione illuminati con back light.	129
5.15	Frammenti di fibra ottica illuminato con front light.	130
5.16	Sfera illuminata con front light.	130
5.17	Esempio con classificatore Multiclasse Specifico PFV (illuminatore front light).	132
5.18	Esempio con classificatore Monoclasse NI (illuminatore front light).	132
5.19	Esempio con classificatore Monoclasse NI (illuminatore back light).	133
5.20	Esempio con classificatore Multiclasse Generale PFV (illuminatore back light).	133
5.21	Esempio con classificatore Multiclasse Specifico NI (illuminatore back light).	134

Introduzione

Al giorno d'oggi l'utilizzo dei sistemi di visione a fini industriali permette: il riconoscimento di difetti e il rispetto delle tolleranze, l'orientamento, il posizionamento e la guida robot, la misura non a contatto, la verifica su nastri in continuo, la classificazione e la scelta, la lettura di caratteri e codici, la sicurezza degli spazi di lavoro condivisi con la macchina; ma l'utilizzo non si limita al settore industriale, si estende alla ricerca scientifica, pubblica sicurezza, rilevazione di infrazioni stradali, sistemi di diagnostica medica. Il sistema di visione è ormai necessario nel settore industriale perché permette di cogliere informazioni approfondite sulla zona di lavoro e/o poter rendere automatiche alcune operazioni che necessitano di una supervisione e, in alcuni casi, sollevare da incarichi pericolosi o delicati l'operatore, basti pensare alle operazioni di saldatura, verniciatura o controllo di conformità; rendendo la macchine più flessibili e autonome, oltre che a diminuire drasticamente i tempi di set-up. A fronte di questi vantaggi vi è uno sviluppo continuo di algoritmi per l'estrazione, l'elaborazione e l'ottimizzazione delle informazioni ottenibili dalle immagini acquisite da un sistema di visione.

In questa tesi è stato analizzato e implementato un algoritmo utilizzato nei sistemi di visione 2D, che permette l'identificazione di classi di oggetti nell'ambito del micro-assemblaggio automatico. Nella manipolazione automatizzata gli algoritmi di riconoscimento, che rientrano nella categoria degli algoritmi di *machine vision*, sono fondamentali per identificare, distinguere e localizzare i componenti con forme e finalità differenti presenti all'interno dell'area di lavoro del manipolatore. Ciò permette di rendere il manipolatore ancor più autonomo e versatile per innumerevoli applicazioni.

Il metodo di identificazione studiato è quello della **classificazione delle particelle** ed è stato implementato per l'utilizzo in una cella di micro-manipolazione presente presso i laboratori del gruppo Micro-Enabled Devices and Systems (MEDIS) dell'Istituto di Tecnologie Industriali e Automazione del Consiglio Nazionale delle Ricerche (ITIA-CNR) di Milano. L'algoritmo verrà sfruttato all'interno del programma di gestione della cella per lo sviluppo di task di manipolazione.

Capitolo 1

Sistemi di visione 2D in ambito micro

1.1 Tecnologie Micro

La tendenza alla miniaturizzazione dei dispositivi, in corso ormai da un decennio, sta diventando sempre più rilevante nella vita di tutti i giorni. Molte applicazioni integrano componenti e dispositivi miniaturizzati, infatti il campo di applicazione non si limita all'elettronica (smartphone, computer portatili, ecc.), ma si estende ai settori automotive (sensori per veicoli), medicale (sonde biologiche, sistemi medici di diagnosi, ricerca genetica), meccanico (microattuatori, microcomponenti), ecc. [25], [16], [14].

Oltre alle tecniche di lavorazione nel campo micro, vi è un interesse crescente rivolto alla manipolazione e assemblaggio di diversi micro-componenti, caratterizzati da alte prestazioni e funzionalità complesse; costituiti da parti da dover assemblate tra loro, composte da diversi materiali (metallici, polimerici o ceramici) ed ottenute con metodi di produzione appropriati. Alcuni esempi di questi prodotti sono micro-pompe, micro-motori, micro-telecamere o altro come mostrato in Figura 1.1 [25]. Nonostante le alte prestazioni e funzionalità, questi micro-prodotti tridimensionali ibridi non hanno ancora consolidato il loro mercato, principalmente a causa dei limiti dei processi di micro-lavorazione e di assemblaggio dei micro-componenti.

Il micro-assemblaggio è definito come le operazioni di posizionamento, orientamento e fixturing di componenti alla micro-scala in micro-sistemi complessi [21].

In molti casi di micro-assemblaggio viene scelto l'assemblaggio manuale perchè i sistemi tradizionali macro non sono idonei a tali operazioni; ma questo richiede operatori altamente specializzati che devono manipolare micro-parti (come resistori, capacitori, ingranaggi o attuatori) con micro-pinze molto delicate e microscopi ad alta magnificazione. La scala ridotta, inoltre, introduce nuove problematiche e sfide connesse alla predominanza delle forze superficiali rispetto a quelle di volume (Figura 1.2), rendendo la manipolazione significativamente diversa da quella alla macro-scala [25]. Questa fase manuale del processo necessita di molto tempo, il che comporta un aumento dei costi. Da ciò si intende che la micro-manipolazione influisce



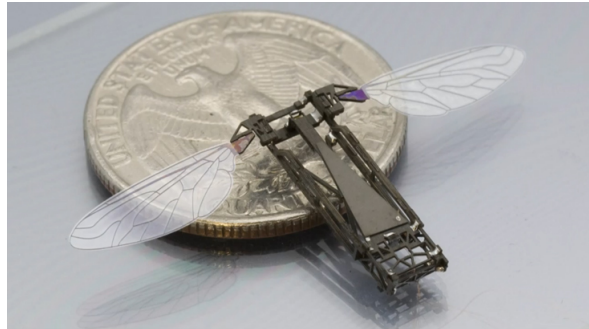
(a) Micro-riduttore planetari con servomotore DC brushless: diametro 8 mm (Cortesia di Faulhaber, www.faulhaber.com).



(b) Micro-ScoutCam™ LEDprobe: endoscopio con diametro 1.2mm per usi medicali dotata di LED di illuminazione (Cortesia di MicroScoutcam, www.microscoutcam.com).



(c) Micro vite senza fine con servomotore DC brushless: passo 0.25 mm e diametro 1.2 mm (Cortesia di Faulhaber, www.faulhaber.com).



(d) MoBee (Micro Robot Ape) (Harvard Microbotics Laboratory, <http://micro.seas.harvard.edu>).

Figura 1.1: Esempi di componenti micro.

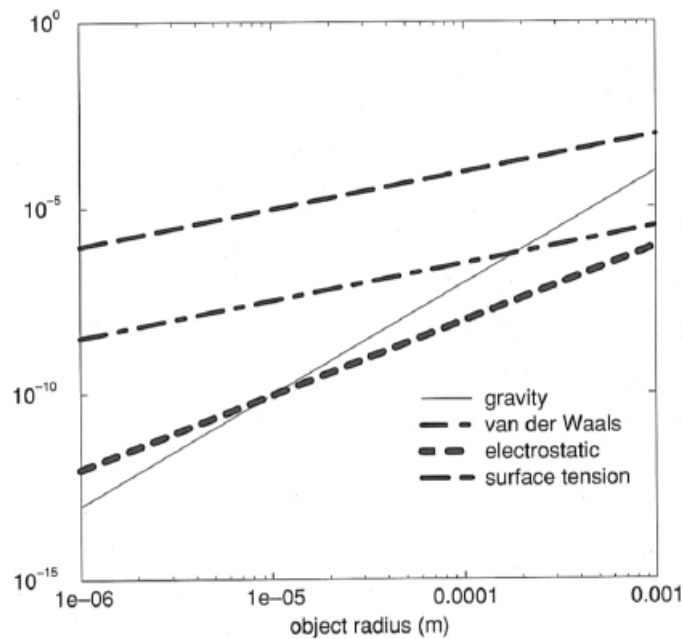


Figura 1.2: Confronto tra forze superficiali e la forza gravitazionale al variare del raggio dell'oggetto [25].

significatamente sul costo del prodotto con valori che possono oscillare dal 60 al 90% del costo finale [25]. Per questa ragione, lo sviluppo di metodi di micro assemblaggio automatici dovrebbe aumentare l'efficienza e l'affidibilità dei componenti e ridurre i tempi e quindi i costi di produzione [26]. L'obiettivo che si pone il settore del micro-assemblaggio è di fornire un mezzo per ottenere dispositivi ibridi alla micro-scala di elevata complessità, mantenendo un alto rendimento e un basso costo [21].

1.2 Sistemi di visione

L'integrazione di sistemi di visione in celle di manipolazione risulta particolarmente utile per svolgere compiti di ispezione, controllo qualità, supervisione o controllo di robot. In questo ultimo caso, le informazioni visive estratte dalle immagini vengono utilizzate per controllare e guidare il robot, aumentando di conseguenza la sua flessibilità e autonomia in un ambiente poco strutturato o dalle difficili caratteristiche operative. Questo è particolarmente vero nella micro-manipolazione, dove le dimensioni fisiche dei componenti richiedono obbligatoriamente l'uso di strumenti di visione dell'area di lavoro.

Le tecniche di visione utilizzate per la micro-manipolazione sono:

- la microscopia a scansione (SEM);
- la microscopia ottica convenzionale e la visione su obiettivi standard.

I dispositivi SEM vengono utilizzati solo per specifiche applicazioni, a causa dell'atmosfera confinata della camera a vuoto e del costo elevato; risentono inoltre anche del problema del limitato campo visivo (*field of view*) che condiziona la dimensione e del tipo di materiale del campione, che deve essere conduttivo.

Al contrario, la visione basata su obiettivi standard e la microscopia ottica convenzionale vengono impiegate in un'ampia gamma di applicazioni di micro-manipolazione e micro-assemblaggio. Queste tecniche offrono un buon compromesso tra risoluzione e facilità d'utilizzo, in più possono operare in diversi ambienti: convenzionale (aria), umido, in liquidi e sotto vuoto [30].

Tali sistemi (da ora in poi verranno chiamati genericamente sistemi di visione) sono costituiti da una lente o una serie di lenti che focalizzano i fotoni sul sensore che fornisce le immagini. Le telecamere possono essere lineari o matriciali e si suddividono in base al tipo di sensore utilizzato Charge Coupled Device (CCD) o Complementary metal oxide Semiconductor (CMOS); si caratterizzano in base allo spettro di luce che è in grado di rilevare (visibile, IR). Per quanto riguarda le telecamere nel visibile possono essere a scala di grigi o a colori (RGB). In base al tipo di oggetto inquadrato ed alle caratteristiche che si vogliono evidenziare devono essere utilizzati sistemi di illuminazione adeguati; in commercio ne sono presenti numerosi e progettati in base

al tipo di applicazione.

Una telecamera fornisce immagini bidimensionali, possono quindi essere estratte facilmente informazioni bidimensionali. Al fine di ottenere informazioni sulla terza dimensione, quali la profondità o il rilievo, è possibile adottare una strategia motion stereo che utilizza una telecamera montata su un robot, la quale acquisisce immagini in due o più posizioni che, dopo opportune elaborazioni, forniscono la posizione di un componente nello spazio; oppure con la visione 3D basata sulla variazione della messa a fuoco/working distance (WD). Nel caso in cui si possano usare più dispositivi si può sfruttare la stereovisione utilizzando una o più sorgenti d'immagine che, combinate tramite opportuni algoritmi, consentono di ricavare il profilo tridimensionale dell'oggetto. In alternativa, si possono utilizzare tecnologie che attraverso la misura del tempo di volo di un emettitore laser permettono di misurare la profondità della scena rispetto al sensore [31]. Un'altra opzione è l'utilizzo di un sistema di scansione laser 3D costituito da una sorgente laser (puntale, a lama o reticolo) e una telecamera; con il quale è possibile ottenere una nuvola di punti in tre dimensioni della zona di lavoro. Oppure si possono ottenere gli stessi risultati utilizzando la visione 3D con luce strutturata, questa tecnologia utilizza un proiettore, che proietta un pattern sull'oggetto da digitalizzare, e una telecamera.

Diversi parametri caratterizzano un sistema di visione: la risoluzione, l'apertura ottica, la magnificazione, la lunghezza focale, il contrasto, la profondità di campo, il campo visivo e la distanza di lavoro. In Figura 1.3 si illustrano le relazioni esistenti tra questi parametri [25].

La risoluzione del sistema di visione deve essere compatibile con la dimensione degli oggetti che si vogliono visualizzare e in base alle caratteristiche da rilevare. Per ottenere un elevato livello di risoluzione nel campo micro, al fine di distinguere il maggior numero di dettagli nelle immagini, l'ottica del sistema di visione deve essere capace di fornire un alto grado d'ingrandimento. Questo requisito comporta un capo visivo ridotto, che nel caso dei micromanipolatori comporta la riduzione dell'area di micro-assemblaggio visibile. La lunghezza focale e il campo visivo sono inversamente proporzionali e ciò si evince dalla relazione che li lega. È quindi necessario trovare un compromesso tra risoluzione e grandezza del campo visivo, a tale proposito sono state individuate diverse soluzioni: la prima è rappresentata dalla combinazione di sorgenti multiple con differenti risoluzioni e campi visivi. Per poter applicare questo metodo ci devono essere almeno due sistemi di visione: il primo ad alta risoluzione con un piccolo campo visivo (in modo da localizzare le immagini all'interno della scena) e il secondo a bassa risoluzione con un ampio campo visivo (per favorire immagini globali della scena); naturalmente ogni sistema avrà il proprio setting di calibrazione (la calibrazione verrà spiegata in seguito) [9]. Un'alternativa alla sorgente multipla è l'utilizzo di una sorgente singola che però utilizza un sistema ottico regolabile meccanicamente con diversi settaggi d'ingrandimento (*zoom*); bassi livelli di zoom producono immagini a bassa risoluzione su un ampio campo visivo, mentre elevati livelli di zoom forniscono immagini ad alta risoluzione su un piccolo campo visivo (anche in questo

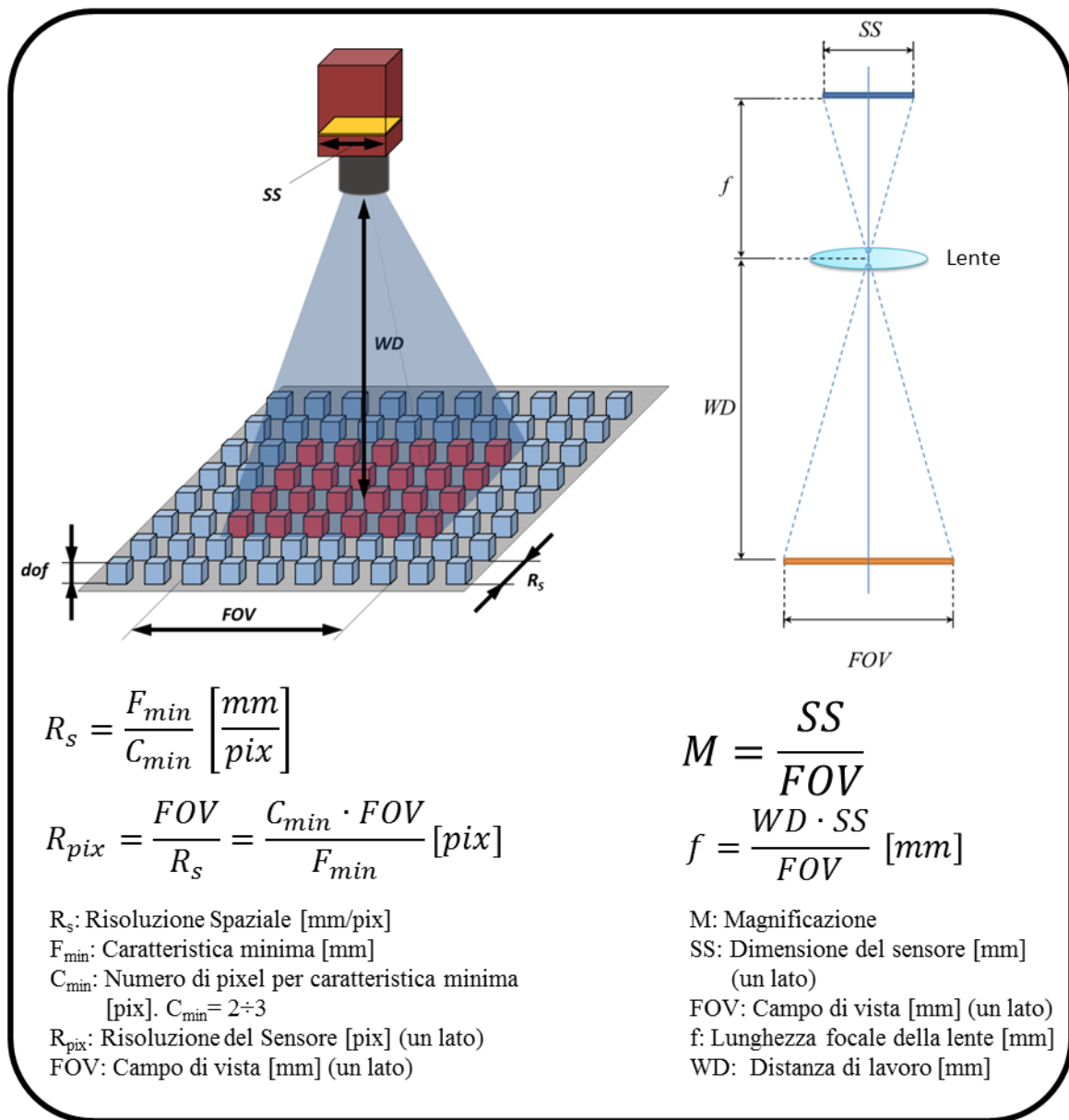


Figura 1.3: Parametri dei sistemi di visione [25]

caso si prevede un setting di calibrazione per ogni ingrandimento del sistema ottico utilizzato) [20, 35]. Bisogna evidenziare che l'elevato grado d'ingrandimento porta ad avvicinare molto il sistema di visione alla zona di lavoro per poterla mettere a fuoco; questa distanza deve tuttavia essere abbastanza grande affinché i componenti possano essere accessibili e manipolabili, a meno che il sistema di visione non sia posto nella zona sottostante a quella di lavoro, che sarà trasparente, rendendola completamente accessibile con qualsiasi strumento. Con quest'ultima configurazione del sistema di visione se lo sfondo dell'immagine non è uniforme, rendendo difficile la discriminazione degli oggetti, si utilizza un pannello di contrasto che uniforma lo sfondo dietro gli oggetti inquadrati.

Un ulteriore problema, quando si effettuano ingrandimenti elevati, è la riduzione della profondità di campo che potrebbe generare immagini sfocate nel caso in cui gli oggetti inquadrati abbiano una profondità maggiore rispetto alla profondità di fuoco del sistema di visione. Per incrementare la profondità di campo possono essere adottate tecniche di focalizzazione automatica (autofocus), che permettono di ridurre tali problemi [28, 17].

Una fase fondamentale di inizializzazione del sistema di visione è la calibrazione, effettuata per calcolare la trasformazione dei pixel dell'immagine nell'unità di misura reale (metrica) e per compensare gli errori prospettici, di distorsione ottica (generalmente, la lente della telecamera introduce distorsione radiale, il che significa che le informazioni dell'immagine sono spostate rispetto al centro ottico della lente [33]) e di referenziazione spaziale del sistema stesso [23]. Il sistema di visione è una sorgente di immagini ottiche che può essere modellato attraverso il modello *pin-hole* della telecamera e il modello di distorsione della lente. La calibrazione delle telecamere è solitamente effettuata prendendo l'immagine di un oggetto di forma e dimensione conosciuta (target di calibrazione) e confrontando la posizione di alcune sue caratteristiche nell'immagine, misurate in pixel, con la posizione effettiva conosciuta, espressa con un'unità di misura metrica. Questo procedimento di calibrazione, comunemente utilizzato alla scala macro, è molto oneroso alla scala micro; nel dominio micro, pertanto, un metodo applicabile prevede la costruzione di un modello di test di calibrazione virtuale in cui si sfrutta l'elevata risoluzione dei sistemi di posizionamento (ad esempio robot). Un target viene posizionato all'interno del campo visivo in zone diverse e tutte queste posizioni vengono utilizzate per la calibrazione [18, 6].

Dal momento che un sistema di visione è di fondamentale aiuto nei sistemi di micro-assemblaggio, l'uso del controllo basato sulla visione risulta buona scelta che permette di migliorare le operazioni di assemblaggio e di sviluppare task automatici. In letteratura sono spesso utilizzate le tecniche di *visual-servoing* ("asservimento visuale") dell'immagine o di *look-and-move* [30].

Questi metodi sono a volte combinati con altri metodi di controllo alternativi, basati per esempio sull'uso di sensori di forza [39, 20, 15] o sensori di posizione ottici [13].

La tecnica di *look-and-move* è una tecnica di *pseudo-visual servoing* in anello aperto: si effettua l'intero processo di estrazione delle informazioni dall'immagine e di calcolo dei comandi per il controllo di movimentazione in passi temporali differenti dal passo di campionamento del controllore di movimento (ad esempio quando un componente deve essere preso o rilasciato). Si tratta di un approccio empirico che non richiede alcuna modellazione [20, 32], il cui vantaggio principale è la facilità d'implementazione.

Il controllo basato sulla visione denominato *visual servoing* utilizza le informazioni delle immagini nell'anello di controllo di movimento dei dispositivi (ad esempio i robot); è una tecnica di controllo ad anello chiuso e, durante il suo funzionamento, i comandi sono espressi nello spazio dell'immagine sotto forma di indicatori visuali come punti, linee e regioni. Per questo motivo

il visual servoing richiede una relazione cinematica da stabilire tra la scena e l'immagine [24].

1.3 Esempi di machine vision in ambito micro

In un'industria particolarmente guidata dal mercato della telefonia mobile, i prodotti elettronici stanno rapidamente diminuendo la loro dimensione mentre aumenta il numero delle loro funzionalità. Ciò comporta un aumento del numero di componenti sulle schede e una riduzione dello spazio a loro disposizione sulla scheda. Per non fermare il continuo sviluppo si ha una continua miniaturizzazione dei componenti, insieme a una maggiore integrazione all'interno del substrato della scheda. L'inserimento dei componenti all'interno del substrato (tecnologia embedding) libera la superficie dal 30% al 50% aumentando così la flessibilità di progettazione (Figura 1.4) [14].

In aggiunta, dato che le connessioni sono più corte, si ha un miglioramento delle prestazioni elettriche e termiche. Questo nuovo tipo di progettazione aiuta inoltre la riduzione del consumo di energia, che permette un maggiore risparmio nell'utilizzo della batteria nelle applicazioni di telefonia mobile.

Il posizionamento dei Surface Mounting Devices (SMD) su schede tradizionali avviene posizionando il componente su una pasta saldante. L'accuratezza richiesta nel pick and place può non essere eccessivamente spinta perché avviene una compensazione dovuta all'effetto di self-alignment dovuto al reflow della pasta saldante. Per quanto riguarda il caso embedding invece, gli SMD vengono posizionati all'interno di cave su uno strato adesivo e, quando il substrato successivo viene aggiunto, le connessioni in rame fanno contatto con il componente; in questo caso l'effetto di self-alignment non è presente, il posizionamento richiede quindi un basso valore di accuratezza che deve essere al di sotto del $50 \mu m$, considerando che la tolleranza delle cave è di circa $20 \mu m$ e in futuro tenderà a diminuire (Figura 1.5) [14].

I requisiti necessari per questa applicazione portano alla scelta di un micro-manipolatore che lavora con l'ausilio di un sistema di visione. Quest'ultimo è usato per l'identificazione dei



(a) Tecnologia convezionale.

(b) Tecnologia embedding.

Figura 1.4: Confronto tra tecnologia tradizionale ed embedding [14].

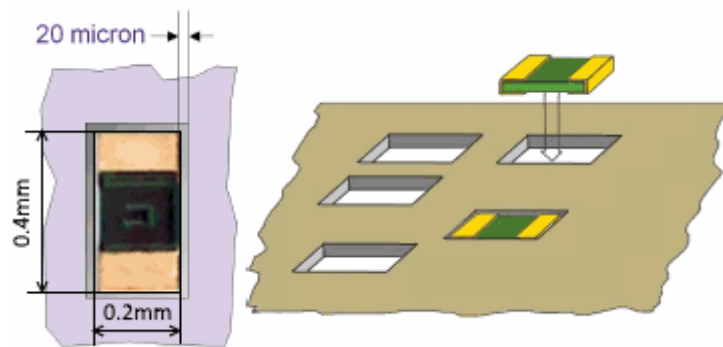


Figura 1.5: Immagine che rappresenta esempi di cave nel substrato e di un componente (0402) inserito all'interno di una di queste [14].

componenti da inserire e delle rispettive cave in cui verranno inseriti. Il sistema di visione incrementa le prestazioni del posizionamento, anche se persiste il problema del ridotto campo visivo rispetto alla grandezza della scheda [34].

Un ultimo fattore da considerare durante l'operazione di inserimento del componente è la forza d'inserimento, d'impatto e statica, da limitare a meno di $2N$ (o anche meno in base al componente) per non danneggiare il componente al momento del contatto con il substrato. Per controllare il contatto in forza si può scegliere tra due modalità: controllo passivo con molla o controllo ad anello chiuso.

In [10] il sistema di visione viene utilizzato per applicazioni di visual servoing di un manipolatore utilizzato per le operazioni di micro-assemblaggio di Micro-Electro-Mechanical Systems (MEMS) (Figura 1.6). La cella di micro-manipolazione è composta da un micro-manipolatore e un sistema di visione; dal punto di vista cinematico la cella ha cinque gradi di libertà, tre lineari (xyz) e due rotazionali ($\alpha\phi$), che sono distribuiti in due sistemi robotizzati $xy\alpha$, che effettuano la movimentazione del piano di lavoro, e $z\phi$, che effettuano la movimentazione della pinza, come mostrato in Figura 1.7. Per quanto riguarda il sistema di visione, è installata una telecamera CCD, con una serie di lenti con lunghezza focale 11.452 mm e un tubo ottico di 140 mm, il tutto posizionato a 45° rispetto al piano orizzontale di manipolazione in modo da

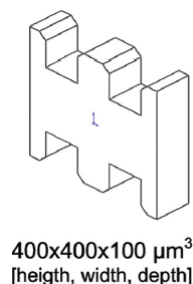


Figura 1.6: Modello CAD componente MEMS [10].

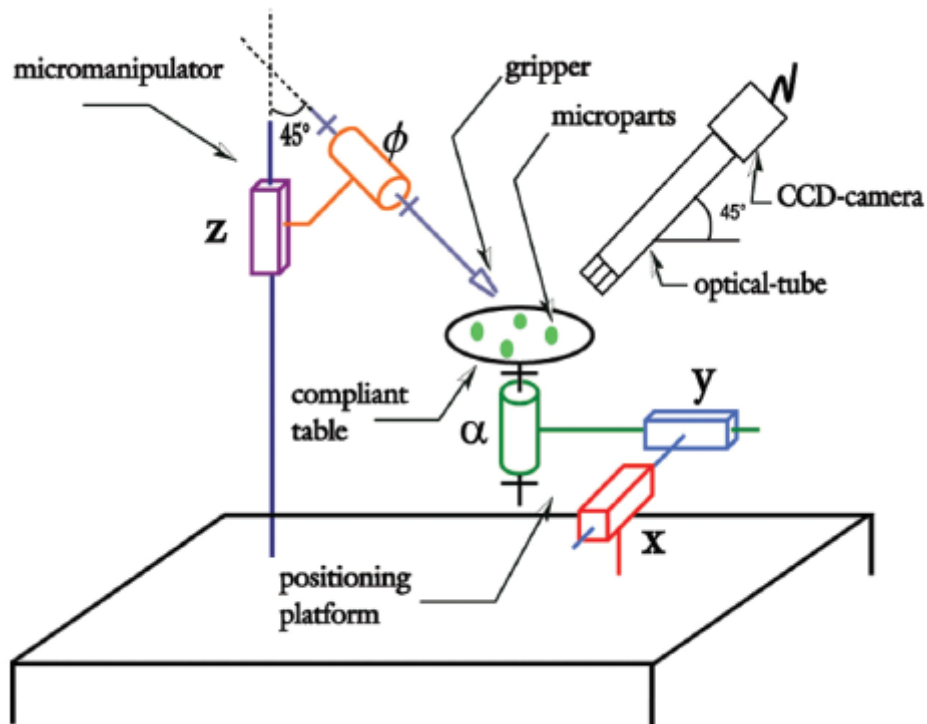


Figura 1.7: Cella di micro-manipolazione [10].

assicurare la migliore vista in prospettiva durante l'assemblaggio.

La manipolazione dei singoli componenti è divisa nelle seguenti fasi: rilevazione e localizzazione del componente, posizionamento dell'end-effector, presa, trasferimento e rilascio del componente. Sono stati assemblati tra loro due o più componenti di dimensione $400 \mu m \times 400 \mu m \times 100 \mu m$, caratterizzati da cavità squadrate di dimensione $97 \mu m \times 97 \mu m \times 100 \mu m$ che ne permettono l'assemblaggio. I ricercatori hanno studiato, implementato e analizzato un algoritmo 3D model-based tracking che, con l'ausilio del modello CAD del componente (Figura 1.6), permette la sua identificazione e posizionamento relativo rispetto al sistema di visione, consentendo di prevedere i movimenti di parti nascoste dell'oggetto.

Siano A e B due componenti da assemblare; le fasi in cui si divide l'assemblaggio sono: posizionamento della parte A (Task # 1), posizionamento della parte B (Task # 2) ed inserimento di B in A (Task #3) (Figura 1.8). Ciascun task inizialmente effettua la localizzazione del componente e successivamente ne effettua la movimentazione verso una nuova posizione. La movimentazione, come detto in precedenza, avviene in visual servoing grazie all'algoritmo di 3D model-based tracking nel seguente modo: una volta localizzato il componente, questo viene movimentato in modo tale da minimizzare la distanza tra la posizione del pezzo e la posizione da raggiungere identificata dal suo modello CAD proiettato nell'immagine; quando la distanza tra i due assume un valore inferiore a una certa soglia, il task di movimentazione è terminato (Figura 1.9). Il controllo degli assi del manipolatore è deciso in base alla posizione visiva rela-

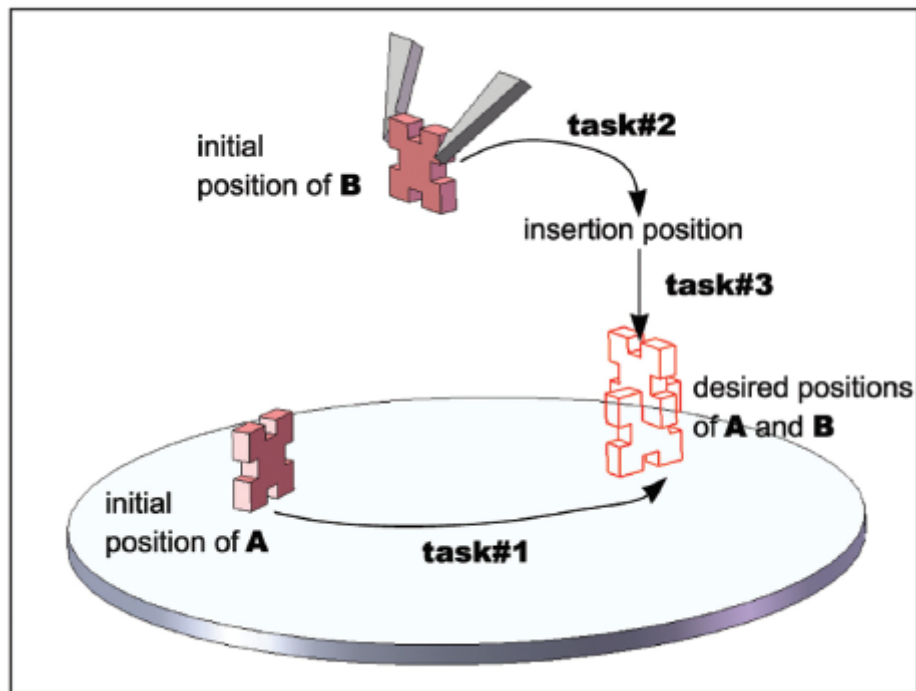


Figura 1.8: Illustrazione task di manipolazione [10].

tiva dell'oggetto e alla posizione finale del modello CAD proiettato (Figura 1.10); la posizione finale che l'oggetto deve assumere viene registrata nella fase di setting dei task di manipolazione dall'operatore che manipola l'oggetto in modalità teleguidata.

La combinazione del CAD model based visual tracking e del controllo visuale ha permesso di ottenere buona ripetibilità e precisione durante numerosi assemblaggi. Il tempo medio di ciclo necessario per assemblare due micro-parti è di 40.8 s, di cui 28.8 s necessari per posizionare i componenti nelle rispettive posizioni di inserzione e 12.0 s per l'inserzione; questo rende la cella 15 volte più rapida e precisa dell'operatore umano aprendo la via per applicazioni industriali di assemblaggio più preciso di MEMS ibridi prodotti in piccole e medie serie.

Oltre al settore industriale, l'ambito micro è di interesse, come detto in precedenza, anche nel campo medico dove in molte situazioni si ha a che fare con campioni biologici di dimensione micrometrica. Numerose ricerche, che sono state effettuate e sono tutt'ora in corso, hanno lo scopo di automatizzare operazioni che ora sono svolte manualmente, in modo da renderle più veloci, precise e ripetibili. Un esempio di micro-manipolazione è la micro-iniezione di uova di zebrafish, effettuate nel campo dell'ingegneria genetica. Le capacità e l'esperienza dell'operatore giocano un ruolo cruciale per il successo di un'iniezione. L'operatore per poter raggiungere tali capacità impiega solitamente alcuni mesi di pratica. L'esecuzione dei vari passaggi della micro-iniezione richiedono un controllo molto fine sia nella posizione che nella forza, difficilmente gestibile contemporaneamente da parte dell'operatore [39].

I tre punti chiave per ottenere una buona iniezione sono scegliere: il punto di iniezione op-

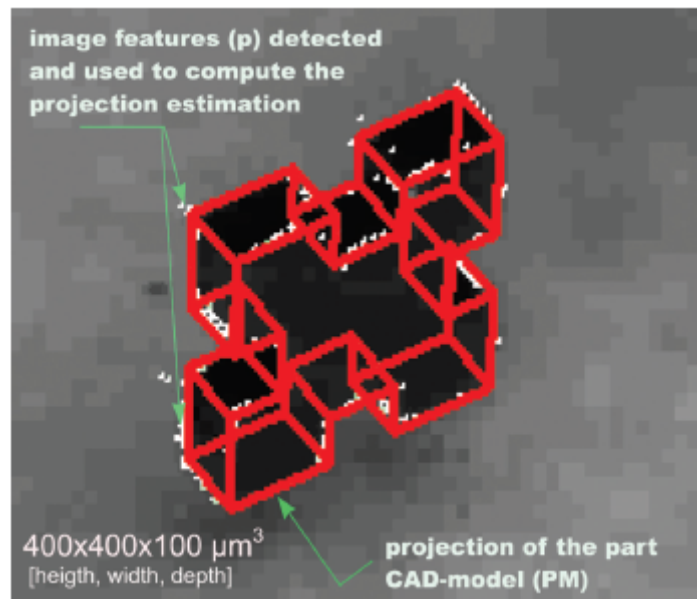


Figura 1.9: Applicazione dell'indicatore di posizionamento sulla microparte. I punti (p) sono usati per stimare la proiezione del modello CAD (PM) come mostrato nell'immagine [10].

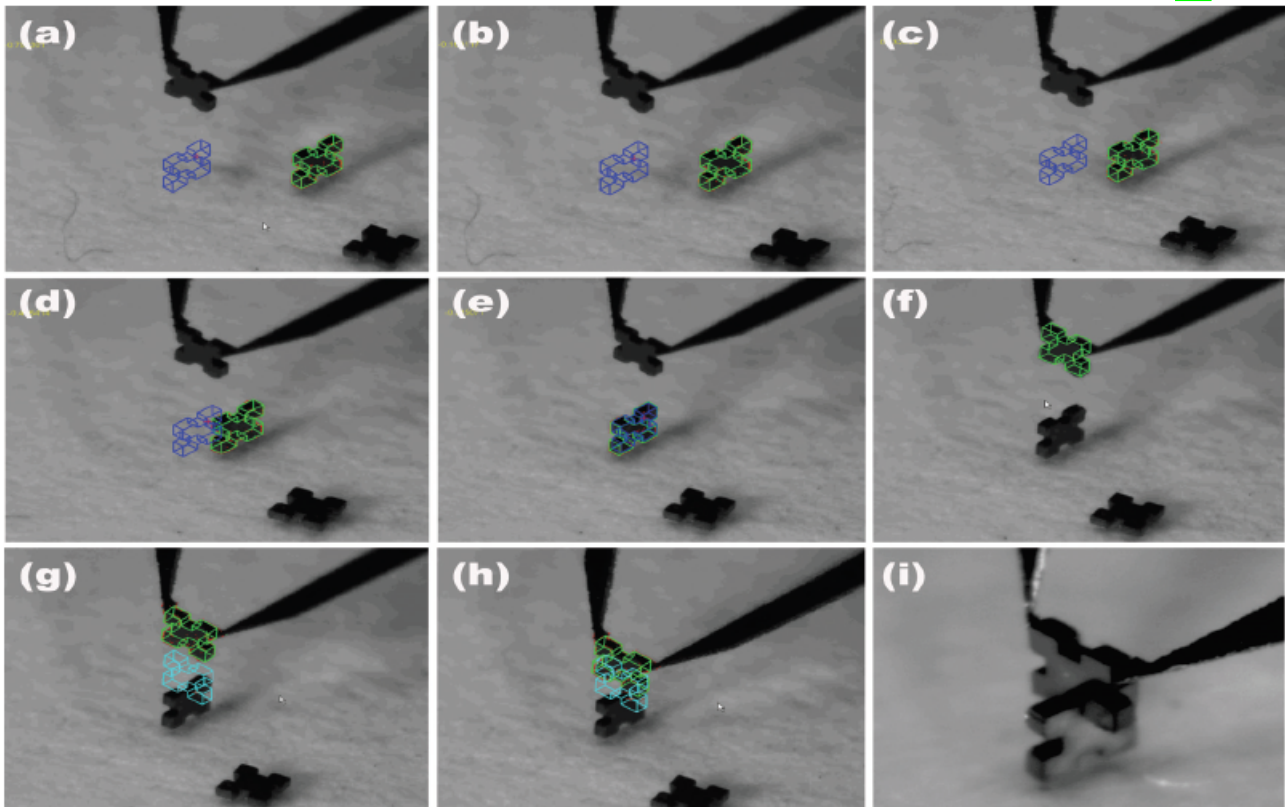


Figura 1.10: Assemblaggio riuscito dei due micro-componenti **A** e **B**: (a) inizializzazione della movimentazione con rilevazione del componente e proiezione del modello CAD; (b)-(d) posizioni intermedie di **A**; (f) inizializzazione della movimentazione con rilevazione del componente e proiezione del modello CAD di **B**; (g) e (h) posizioni intermedie di **B**; (i) una vista ravvicinata della micro-struttura 3D [10].

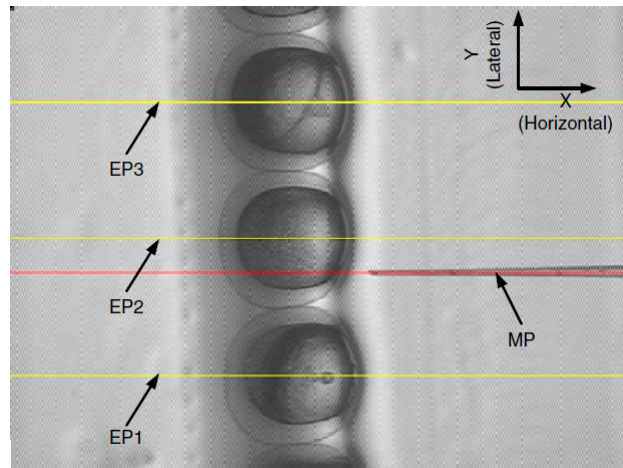


Figura 1.11: Assi degli embrioni di Zebrafish (EP1, EP2 e EP3) e micropipetta (MP) [39].

portuna, una idonea velocità di penetrazione e una traiettoria di penetrazione appropriata. L'utilizzo di micromanipolatori con elevata accuratezza e ripetibilità porterebbe all'aumento del numero di successi di iniezione ed alla diminuzione dell'influenza del fattore umano [39]. In [39] il sistema di visione gioca un ruolo fondamentale per l'esecuzione di task automatici di micro-iniezione. L'algoritmo di machine vision determina il numero di embrioni e ne traccia a video l'asse di ognuno in modo che sia parallelo all'asse della micro-pipetta, anch'esso tracciato a video (Figura 1.11).

La micro-pipetta viene movimentata lateralmente in modo che questa si possa allineare con gli assi degli embrioni presenti su una holder lineare. Una volta allineati embrione e micro-pipetta comincia un movimento orizzontale in avanti di quest'ultima verso l'embrione a velocità costante per effettuare la fase di avvicinamento e la successiva iniezione. Per rilevare la perforazione della membrana si utilizza un sensore di forza piezo-resistivo incollato dietro la pipetta. Per effettuare l'identificazione e la localizzazione dell'embrione si utilizza un algoritmo di machine vision che cerca il suo tuorlo, dato che è la caratteristica dominante, all'interno della zona di lavoro del micromanipolatore inquadrata dal sistema di visione. Viene effettuata anche l'identificazione e localizzazione della micro-pipetta. Come algoritmo di machine vision si utilizza un metodo chiamato *pattern matching*, che verrà spiegato nei prossimi capitoli. Gli assi riportati a video permettono di identificare il punto di inizio iniezione di ciascun embrione.

In questa applicazione la parte di machine vision permette di effettuare l'iniezione nel punto ottimale in maniera autonoma, e quindi abilita il sistema ad effettuare task automatici di iniezione. L'unico problema che rimane in [39] è che dopo la penetrazione non è possibile effettuare la iniezione perché la cavità della micro-pipetta è tappata dalla colla utilizzata per fissare il sensore di forza.

In un'altra applicazione, che riguarda sempre il campo della micro-iniezione di uova di zebrafish, il sistema di visione è stato abbinato a una rete neurale; così facendo è stato possibile

modellare una precisa cella di carico software che misura la forza applicata a un'embrione in funzione della sua deformazione (misura indiretta) [27]. Tutto ciò senza l'utilizzo di modelli matematici di deformazione, molto complessi e soggetti a molte variabili. Il primo passo è acquisire una sequenza di immagini di un embrione in diverse condizioni di carico, che, grazie a una serie di elaborazioni, permettono l'estrazione delle caratteristiche geometriche, come deformazione o dimensione, dell'embrione; queste acquisizioni ed elaborazioni sono effettuate per più embrioni di diverse dimensioni in modo da poterne considerare la dimensione nel modello di deformazione. Tutte le informazioni raccolte vengono utilizzate nella fase di training della rete neurale in modo da generare una correlazione che lega la deformazione dell'embrione con la forza applicata a esso (Figura 1.12). Le informazioni visuali estratte da nuove immagini, attraverso la rete neurale, possono quindi essere utilizzate come una cella di carico per stimare la forza applicata all'embrione presente nell'immagine. Questo utilizzo del sistema di visione è molto utile nel caso in cui si necessiti di una retroazione in forza, ma l'uso di un sensore non sia pratico (come nel caso precedente dove si verificava un chiusura della cavità della micro-pipetta [39]), o sia troppo costoso.

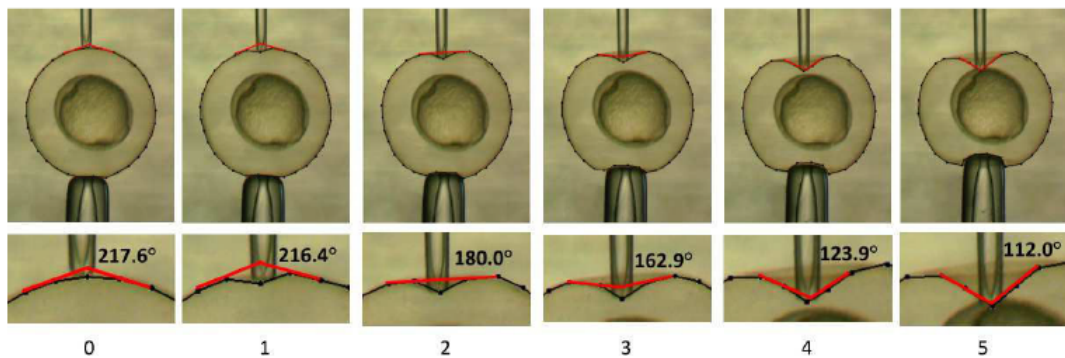


Figura 1.12: Rilevazione della deformazione degli embrioni di zebrafish durante la micro-manipolazione [27].

Un secondo esempio di micromanipolazione nel campo bio-ingegneristico è l'assemblaggio di vasi artificiali per la vascolarizzazione di tessuti [22].

Tali micro-tubi sono costituiti da piccoli anelli, detti *donutshaped micromodules* (DSM), che vengono impilati l'uno sull'altro in modo da creare un condotto. Il task automatico è un ciclo di assemblaggio definito come "*repetitive single-step assembly*" (RSA). Per fornire informazioni in diretta per il task RSA automatico i DSM vengono identificati, selezionati e localizzati automaticamente attraverso il sistema di visione. Come mostrato in Figura 1.13 dopo una sogliatura (o binarizzazione) dell'immagine per mettere in evidenza ciò che è presente nell'area di lavoro, vengono rilevati i contorni di tutte le DSM e della micro-pipetta; nel frattempo tutti i DSM non utilizzabili sono scartati in base al *minimum enclosing rectangle method* [22]. Con specifici criteri di area la micro-pipetta viene discriminata dai DSM e ne vengono evidenziati i

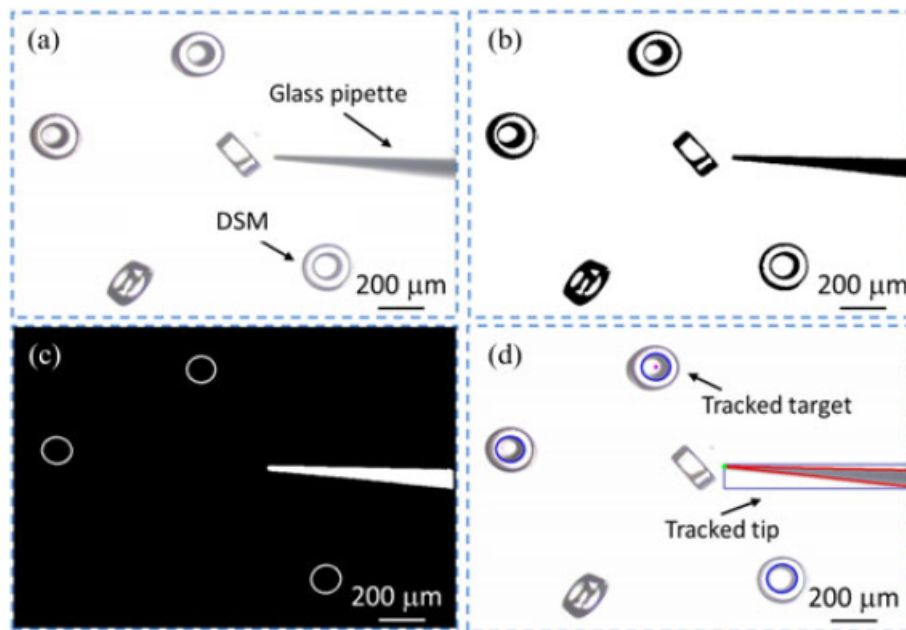


Figura 1.13: Rilevazione dei DSM e della localizzazione dell'obiettivo [22].

contorni (Figura 1.13). Per rilevare la quota Z del componente si utilizza il metodo di contatto che non richiede sensori addizionali. La quota si identifica facendo muovere la micro-pipetta in verticale a velocità costante fino alla rilevazione di una brusca variazione della forma del DSM vista dal sistema di visione, tale variazione è generata dal contatto della micro-pipetta con il DSM. Una volta rilevate le posizioni e la quota verticale del DSM e localizzata la micro-pipetta si procede con le operazioni per impilare i DSM.

L'occlusione tra la micropipetta e i DSM da prelevare e impilati è inevitabile, ciò causa problemi a livello di localizzazione della punta, causando errati avvicinamenti nel punto di pick. Nel caso in cui la micro-pipetta è parzialmente occlusa da un DSM (Figura 1.14(a)), si approssima la sua forma a un triangolo effettuando un line fitting dei suoi punti estremi come si può vedere in Figura 1.14(b). Nel caso in cui si stia effettuando l'avvicinamento alla zona di presa, la punta viene occlusa impedendo la rilevazione della micro-pipetta con il modello triangolare, considerando che la dimensione della micro-pipetta e dei DSM è molto differente, l'area occlusa è sempre caratterizzata da angoli di forma che possono essere considerati come nuove caratteristiche utili per rilevare la micro pipetta (Figura 1.14(c)). Questo tipo di caratteristiche è rilevabile con l'Harris Corner Detection (HCD) [8]. Con tali punti si ricalcola il line fitting e si rileva la punta della micro-pipetta come si può vedere da Figura 1.14(d) [22].

Il sistema di visione in questa applicazione permette di ottenere un task di assemblaggio completamente autonomo e veloce.

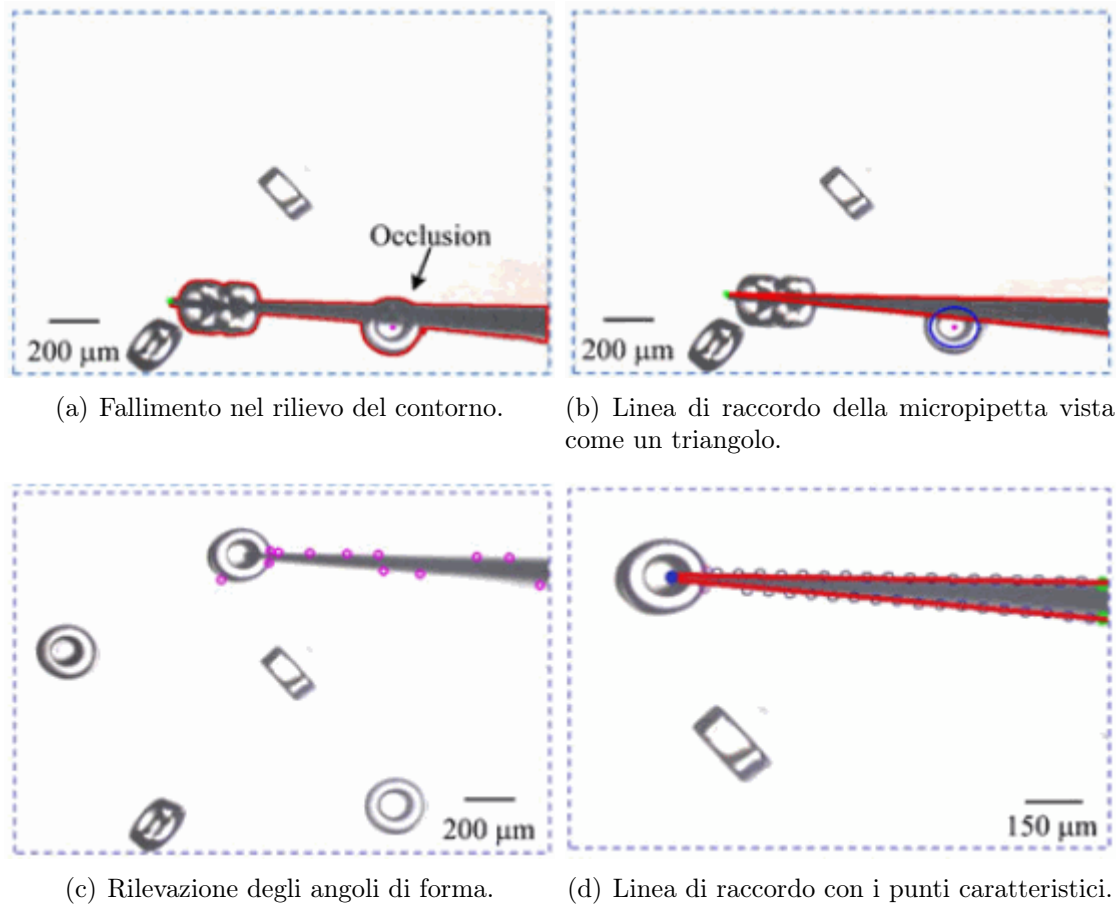


Figura 1.14: Rilevazione della micropipetta durante l'occlusione [22].

1.3.1 Esempi di applicazione del metodo della classificazione

Gli esempi illustrati in seguito danno un'idea dell'utilizzo della classificazione nel caso in cui si voglia identificare un oggetto o un evento.

In [41] i ricercatori si sono posti l'obiettivo di localizzare le coordinate cartesiane 3D di un microchip trasparente SU-8, di dimensione $250\mu m \times 250\mu m \times 30\mu m$, durante la sua manipolazione. Per raggiungere lo scopo, nella cella di micro-manipolazione (Figura 1.15) sono stati installati due sistemi di visione: uno che riprende la zona di lavoro dall'alto, mentre il secondo che la riprende lateralmente. Con il primo sistema di visione per la rilevazione è stato utilizzato semplicemente un algoritmo di pattern matching, perché il microchip non è mai occluso e la sua immagine è sempre a fuoco dato che si trova sempre nel piano focale. Per quanto riguarda invece la vista laterale si hanno i seguenti problemi:

1. l'area di lavoro del micro-manipolatore lungo l'asse ottico è di $1500\mu m$ e il microchip è lungo $250\mu m$, mentre la profondità di campo del sistema di visione utilizzato è di circa $100\mu m$ con un contrasto relativo maggiore di 0.8 (contrasto relativo: è la più piccola differenza di luminosità di un'area osservata necessaria perché questa risulti visibile rispetto

- alla luminanza del fondo). Ciò comporta che spesso il microchip apparirà parzialmente o completamente sfocato.
2. I bordi del chip sono molto meno visibili nella vista laterale rispetto alla vista dall'alto, a causa della configurazione degli illuminatori.
 3. Il microchip è frequentemente occluso dalle punte della pinza, dato che viene preso dai lati.
 4. La pinza può ruotare lungo l'asse Z causando cambiamenti di vista laterale del chip.

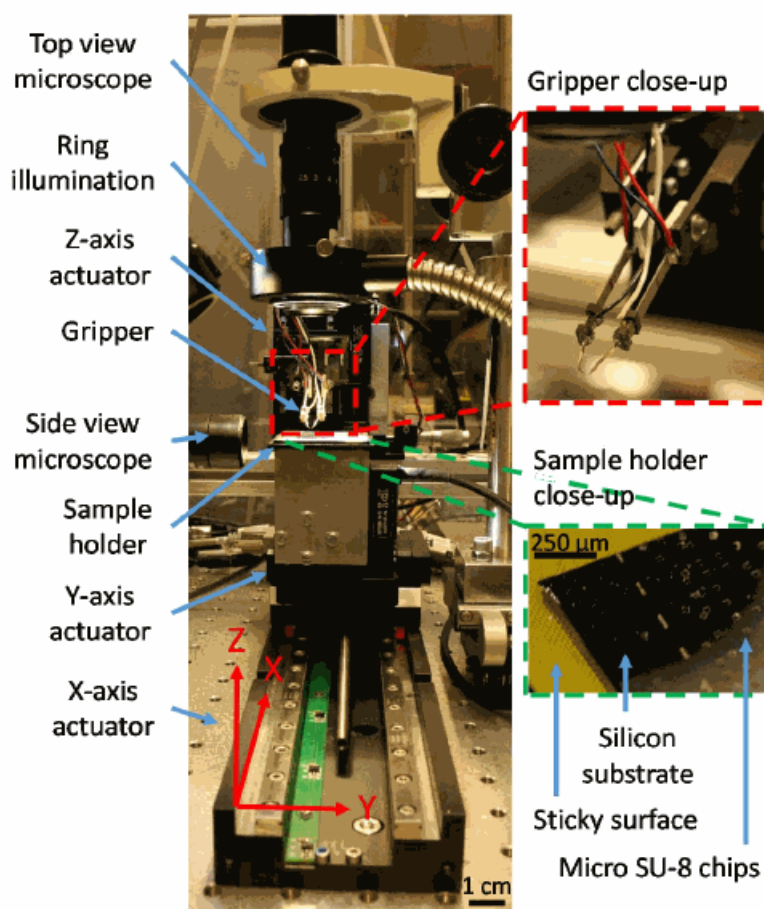


Figura 1.15: Cella di micro-manipolazione di [41].

Le tecniche convenzionali di machine vision che sono state testate non permettono la rilevazione del chip dalla vista laterale; per questo è stata applicata la Classificazione utilizzando come feature il Local Binary Pattern (LBP) [5] applicato alle sub-immagini dell'area inquadrata. Utilizzando il ML, che è stato istruito con varie immagini dell'oggetto in diverse posizioni nel piano di lavoro, il massimo errore nella zona compresa tra $+350\mu\text{m}$ a $-350\mu\text{m}$ dal piano focale è di $30\mu\text{m}$ (pari al 12% della lunghezza del chip) (Figura 1.16).

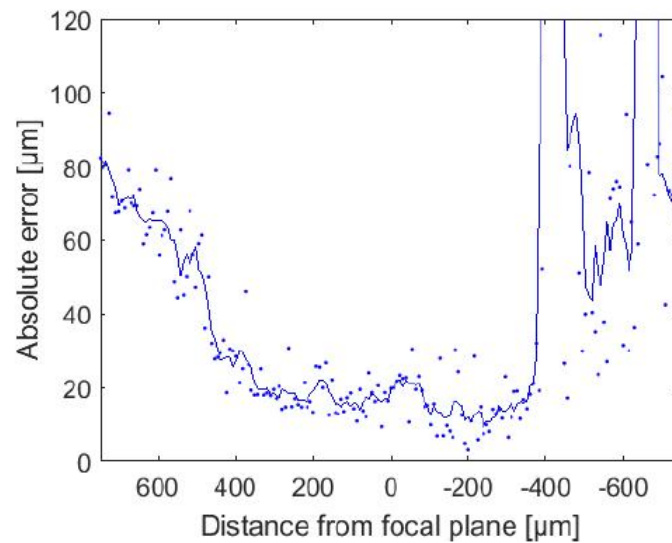


Figura 1.16: Errore assoluto di rilevazione della vista laterale del chip SU-8 al variare della distanza dal piano focale [41].

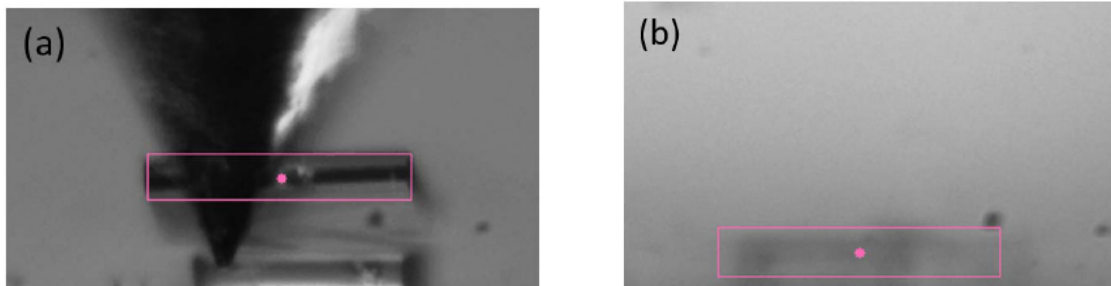


Figura 1.17: Rilevazione con algoritmo di ML del chip da vista laterale. (a) Il rilevatore localizza la faccia giusta del chip (rettangolo rosa) nonostante contenga difetti e sia parzialmente occluso e ruotato. (b) Il rilevatore localizza il chip a una distanza di $750\mu m$ dal piano focale [41].

L'oggetto risulta quindi accuratamente rilevato e localizzato in situazioni dove i metodi convenzionali non possono essere usati, ma anche in immagini sfocate a tal punto da non essere riconoscibili dall'occhio umano (Figura 1.17). La localizzazione è possibile grazie alla divisione dell'immagine in sub-immagini che permette di identificare la zona in cui avviene il riconoscimento del micro-chip.

In [40], che è uno sviluppo ulteriore della [41], viene utilizzata la classificazione per rilevare un evento che descrive lo stato di manipolazione. L'oggetto manipolato è sempre il microchip SU-8 e nella ricerca si sono voluti rilevare in particolare due eventi (Figura 1.18):

- evento in cui il microchip è in presa;
- evento in cui il microchip si incolla alla punta della pinza nella fase di rilascio.

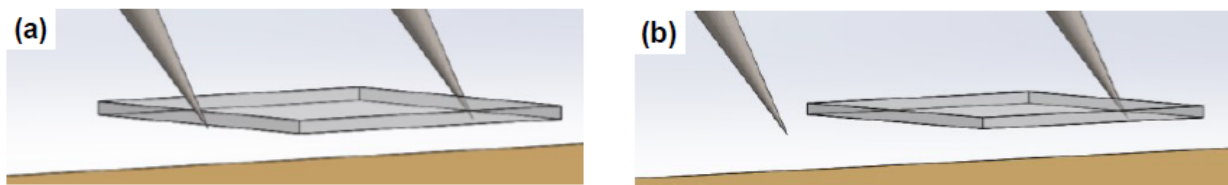


Figura 1.18: Illustrazione degli eventi da classificare. (a)Evento di prelievo mentre il chip viene manipolato sopra il piano di lavoro. (b)Evento di incollaggio mentre il chip rimane attaccato a una delle punte della pinza durante il rilascio [40].

Per rilevare l'evento si analizzano le caratteristiche che legano la pinza e il chip, misurabili con i due sistemi di visione menzionati in precedenza (Figura 1.19). Le feature che il classificatore utilizza non descrivono caratteristiche di forma, bensì sono delle misurazioni che permettono l'identificazione dell'evento:

- **Caratteristiche 1 e 2: minima distanza tra pinza e chip, misurata con vista dall'alto** (Figura 1.20(a));
- **Caratteristica 3: apertura della pinza, misurata con vista dall'alto** (Figura 1.20(a));
- **Caratteristica 4 : allineamento tra pinza e chip, misurato con vista laterale** (Figura 1.20(b)), è la distanza d tra le punte della pinza e il chip. Se la pinza si trova tra la parte alta e la parte bassa del chip d è zero, se si trova sopra il chip d è positivo, mentre se si trova sotto è negativo;
- **Caratteristiche 5-11: velocità dell'oggetto interessato** (Tabella 1.1) le velocità della pinza e del chip sono richieste per rilevare lo slittamento del chip quando viene preso e per la classificazione dell'evento d'incollaggio nella fase di rilascio.

Tabella 1.1: Tabella delle feature di velocità [40].

Feature	Descrizione feature
5	velocità in x del chip manipolato nella vista dall'alto
6	velocità in y del chip manipolato nella vista dall'alto
7	velocità angolare del chip nella vista dall'alto
8	velocità in y della punta alta della pinza nella vista dall'alto
9	velocità in y della punta bassa della pinza nella vista dall'alto
10	velocità in y del chip nella vista laterale
11	velocità in y delle punte della pinza nella vista laterale

In [40] si è affermato e dimostrato che la classificazione basata sul ML è un metodo adeguato per identificare eventi durante la micro-manipolazione. Questo esempio permette di capire la versatilità dell'algoritmo della classificazione.

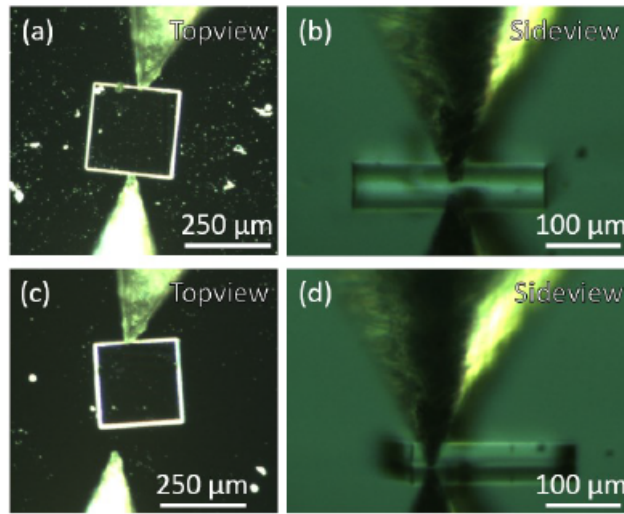


Figura 1.19: (a)Evento di presa ripreso con la vista dall'alto.(b)Evento di presa dalla vista laterale. (c)Evento di incollaggio ripreso con la vista dall'alto. (d)Evento di incollaggio dalla vista laterale [40].

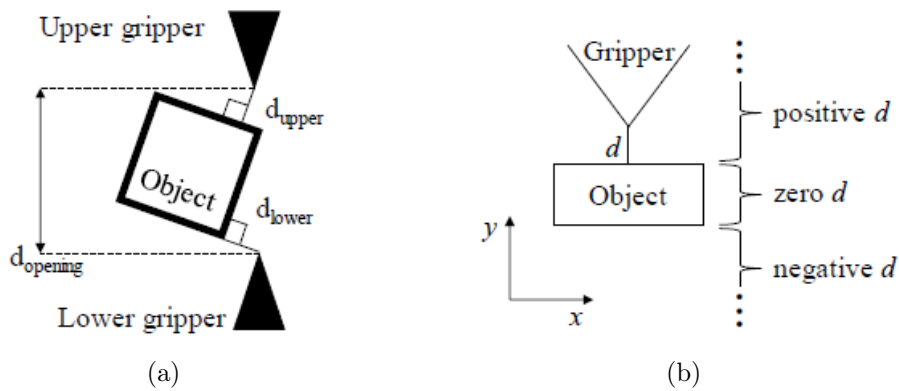


Figura 1.20: Caratteristiche da 1 a 4. (a)La distanza minima d_{upper} e d_{lower} tra la punta alta e bassa della pinza dall'oggetto manipolato; $d_{opening}$ grandezza di apertura della pinza, tutte misurate grazie alla vista dall'alto. (b)Le differenti regioni che determinano il segno e il valore della feature d [40].

1.4 Software e librerie di machine vision

Nel campo dell'immagine processing e della machine vision esistono librerie e software appositi come: Open CV, Halcon, Computer Vision System Toolbox di Matlab e NI Vision di LabView. Esistono anche altre librerie ma la maggior parte si basano su Open Cv.

1.4.1 Open CV



Figura 1.21: Logo di OpenCV.

Open CV(**Open Source Computer Vision Library**) è una libreria *open-source*, scritta in C/C++, per lo *streaming video real-time*, ovvero la computer vision. Si interfaccia con linguaggi C++, C, Python e Java ed è supportato Windows, Linux, Mac OS, iOS e Android.

Open CV nasce da una iniziativa dei tecnici dell'Intel, mentre lavoravano sul miglioramento delle CPU Intel usata per applicazione intensive, come *ray-tracing* [6] in tempo reale e *proiezione 3D*. Uno degli addetti della Intel aveva notato come in molte Università, tra cui il MIT Media Lab, erano state create diverse infrastrutture per la computer vision, il cui codice era passato da studente a studente. A tal proposito si decise di iniziare a progettare, a partire da questo codice, un'architettura logica di supporto per la computer vision per i processori Intel. Il primo avvio di tale progetto, con la collaborazione di un team Intel russo, fu nel 1999. La prima release ufficiale di Open CV risale, invece, al 2006. Nel 2011 è stata rilasciata la versione 2.2.0, che supporta Android. Ora è disponibile la release 3.4. La struttura di Open CV consta delle seguenti 6 parti (Figura 1.22):

- **CXCORE**: contiene le definizioni di tutte le strutture dati e le funzioni per la gestione di immagini e video;
- **CV**: contiene tutte le funzioni per l'elaborazione e l'analisi delle immagini, la calibrazione e il tracking;
- **ML (Machine Learning)**: contiene molte funzioni sul Machine Learning e il pattern recognition, quali il clustering e la classificazione;
- **HighGUI**: contiene le definizioni delle interfacce utenti (GUI);

- **CVCAM**: contiene interfacce webcam;
- **CVAUX**: contiene algoritmi sperimentali per scopi diversi, ad esempio: segmentazione, sottrazione del background, modelli HMM, ecc.

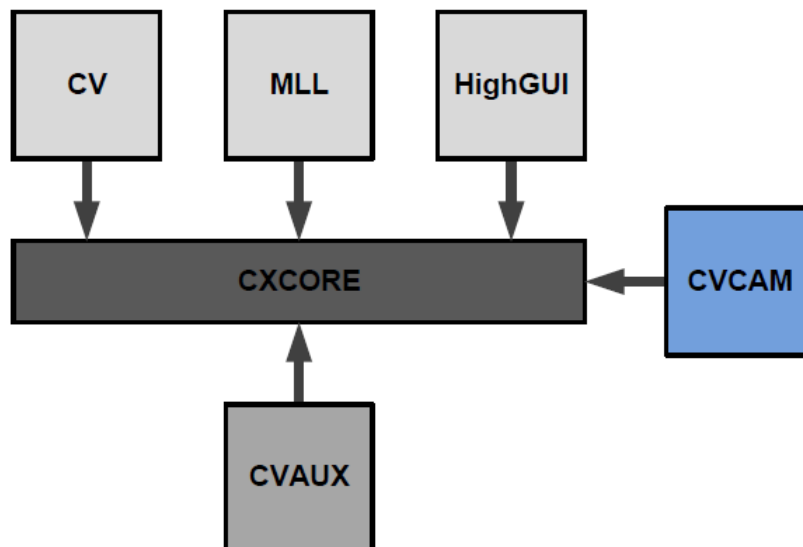


Figura 1.22: Schema rappresentativo dell'organizzazione di OpenCV.

Per velocizzare OpenCV è possibile utilizzare la libreria **IPP** (*Intel Performance Primitives*). Sono delle librerie altamente ottimizzate, che agiscono direttamente a livello delle micro-istruzioni del processore, evitando qualsiasi perdita di tempo [36].

Il vantaggio delle librerie di OpenCV è costituito dalla loro natura Open Source, che le rende libere sia per scopi scolastici che industriali; possono inoltre subire continui miglioramenti da parte degli utenti. Il lato negativo è costituito dall'incertezza riguardo alle funzioni fornite dagli utenti, che potrebbero non essere interamente testate o ottimizzate. Bisogna quindi prestare elevata attenzione quando queste vengono utilizzate. Inoltre, essendo una libreria open source, non è disponibile un servizio di assistenza, bisogna quindi affidarsi ai vari forum specializzati in cui si può ricevere aiuto da altri utenti. Di seguito si riportano alcune funzioni fornite da OpenCV:

- **Threshold**: l'operazione di Threshold consiste nell'associazione del valore 0 o 1 a un pixel di una immagine nel caso in cui questo abbia un valore inferiore o superiore a un certo valore di soglia, tale operazione viene applicata su tutti i pixel di una immagine. La funzione *cvThreshold* riceve in ingresso un array (*src*), rappresentante l'immagine a cui applicare l'elaborazione, con una soglia (*threshold*) e restituisce un array (*dst*), rappresentate l'immagine elaborata, con i valori associati a ciascun pixel;

- **Template Matching:** la funzione ricerca un frammento di immagine facendolo scorrere all'interno di un'altra immagine da ispezionare. Il frammento viene confrontato con la zona ricoperta ad ogni spostamento verificandone la somiglianza. La funzione *cvMatchTemplate* riceve in ingresso il frammento d'immagine di riferimento (*templ*) e l'immagine in cui cercare (*image*); in uscita la funzione restituisce l'immagine che ha trovato con la relativa posizione all'interno dell'immagine (*result*).

1.4.2 Halcon



Figura 1.23: Logo di Halcon.

Halcon è una libreria proprietaria di algoritmi per l'elaborazione delle immagini e la visione artificiale. Il software è sviluppato dalla società di Monaco **MVTec**.

La libreria Halcon fornisce circa 2100 operatori diversi e permette di interfacciarsi con molte fotocamere e frame grabber. È progettata per il funzionamento su sistemi multicore e per GPGPU (*General Purpose Computation on Graphics Processing Unit*). Ci sono versioni anche per vari sistemi embedded [\[1\]](#).

Le aree di applicazione degli algoritmi sono:

- elaborazione delle immagini 3D;
- OCR(*Optical Character Recognition*): programmi dedicati al rilevamento dei caratteri contenuti in un documento e al loro trasferimento in testo digitale leggibile da una macchina;
- Blob analisi: rilevamento di punti e/o regioni in una immagine che differiscono in proprietà come luminosità o colore rispetto all'ambiente;
- Calibrazione della camera;
- Misurazione accurata dei subpixel;
- Trasformazioni geometriche;
- Identificazione di codice;

- Elaborazione video;
- Machine learning;
- Deep Learning.

Halcon è utilizzato prevalentemente in applicazioni industriali e mediche.

Infine, essendo Halcon un software proprietario, permette di avere la garanzia di test approfonditi e di ottimizzazione delle funzioni fornite. Naturalmente tutto ciò ha un costo che corrisponde al pagamento della licenza di uso del software.

Di seguito sono riportate alcune funzioni:

- **Blob Analysis:** in un'immagine un oggetto può essere identificato in funzione dei valori di grigio che assumono i pixel che lo rappresentano. Si rileva quindi l'oggetto selezionando solo i pixel dell'immagine che assumono un certo valore o appartengono a un certo intervallo di scala di grigio. Ciò permette di evidenziare oggetti diversi, in una immagine, in base al suo valore di grigio. La Blob Analysis consta di tre fase: acquisizione dell'immagine, segmentazione dell'immagine (selezione dei pixel rappresentanti l'oggetto con un certo valore di grigio) e infine estrazione delle caratteristiche (come area, perimetro, ecc.). Questa funzione è possibile concatenando quattro funzioni: `read_image`, `threshold`, `connection` e `area_center`.
- **Matching:** questa funzione effettua il search and match di una immagine di riferimento all'interno di un'altra immagine. Per prima cosa si genera un'immagine di training chiamata `template`. La funzione deriva un modello del `template` con cui si ricerca all'interno dell'immagine da ispezionare una copia del `template`.

1.4.3 MATLAB

MATLAB (abbreviazione per **MA**Lrix **LAB**oratory) è un ambiente per il calcolo numerico e l'analisi statistica scritto in C, che comprende anche l'omonimo linguaggio di programmazione creato dalla *Math Works*. MATLAB consente di manipolare matrici, visualizzare funzioni e dati, implementare algoritmi, creare HMI e interfacciarsi con altri programmi.

MATLAB è usato da milioni di persone nell'industria per via dei suoi numerosi strumenti a supporto dei più disparati campi di studi applicati; funziona inoltre su diversi sistemi operativi, tra cui Windows, Mac OS, GNU/Linux e Unix [3].

MATLAB ha un modulo chiamato **Vision System Toolbox** che mette a disposizione algoritmi, funzioni e applicazioni per la progettazione e la simulazione di sistemi di computer vision e elaborazione video. Con questo modulo si possono inoltre effettuare rilevazioni, estrazione e matching di caratteristiche, oltre alla rilevazione e inseguimento di oggetti. Per le applicazioni

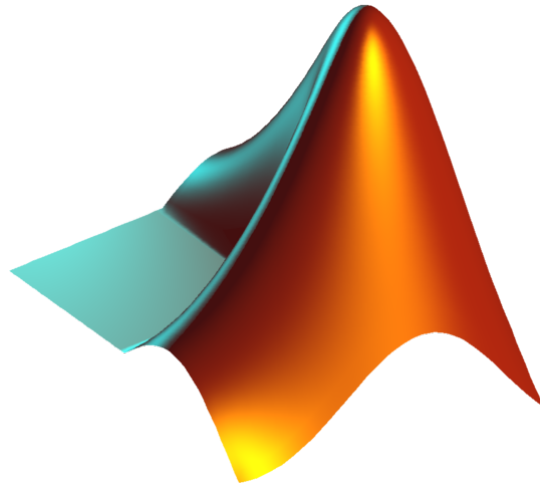


Figura 1.24: Logo di Matlab.

3D di computer vision, il system toolbox supporta camere singole, stereo e *fisheye*; inoltre permette la calibrazione di telecamere, la ricostruzione 3D e l'elaborazione di nuvole di punti 3D.

Gli algoritmi per il deep learning e machine vision permettono di identificare visi, pedoni o altri oggetti comuni usando un identificatore che ha subito una operazione di pretraining.

Sono presenti numerosi programmi interattivi di MATLAB creati dagli utenti, utilizzabili senza dover scrivere alcuna riga di codice. Tali applicazioni possono essere scaricate gratuitamente da MATLAB File Exchange, ed essendo create dagli utenti sono messe in condivisione permettendo di risparmiare tempo agli sviluppatori [2]. MATLAB è un software proprietario come Halcon, ciò comporta che queste due piattaforme hanno le stesse caratteristiche in quanto ad assistenza e ad affidabilità delle funzioni fornite dalle loro librerie.

Di seguito sono riportati alcuni esempi di funzioni:

- **edge(I)**: questa funzione permette di identificare i profili in una immagine. L'immagine in ingresso I alla funzione deve essere binarizzata e la risposta che fornisce è una immagine in cui i pixel a 1 indicano la rilevazione di un bordo in quel punto, mentre i pixel a 0 indicano che non è stato rilevato niente.
- **bwtraceboundary(BW,P,fstep)**: traccia le linee di contorno di un oggetto presente in una immagine binarizzata. L'immagine fornita dalla funzione avrà i pixel appartenenti agli oggetti non a zero, mentre i pixel appartenenti allo sfondo saranno a zero. La variabile P è un vettore di due elementi che indica il pixel da cui si vuole cominciare la ricerca; $fstep$ è una stringa di caratteri in cui si specifica la direzione in cui cercare il prossimo pixel dell'oggetto connesso a P .

1.4.4 LabVIEW



Figura 1.25: Logo di LabVIEW.

LabVIEW (abbreviazione di **L**aboratory **V**irtual **I**strumentation **E**ngineering **W**orkbench) è un software di tipo proprietario ed è l'ambiente di sviluppo integrato per il linguaggio di programmazione virtuale della National Instrument. Tale linguaggio grafico viene chiamato *Linguaggio G*. Realizzato inizialmente per Apple Macintosh nel 1986, LabVIEW viene utilizzato principalmente per programmi di acquisizione e analisi dati, controllo di processi, generazione di rapporti, o più generalmente per tutto ciò che concerne l'automazione industriale. Oggi funziona su diverse piattaforme come Windows, Linux, Mac OS e controllori National Instrument [7]. LabVIEW ha una libreria dedicata al machine vision chiamata **NI Vision**, la quale è suddivisa in categorie in base alle funzionalità offerte [38]:

- **Vision Utilities:** le cui funzioni permettono di manipolare e visualizzare un'immagine;
- **Image Processing:** le cui funzioni permettono di effettuare operazioni di analisi, filtraggio e elaborazione dell'immagine;
- **Machine vision:** funzioni di alto livello che semplificano i task comuni di machine vision;
- **Vision Express** permette di sviluppare velocemente comuni applicazioni di acquisizione e elaborazione.

LabVIEW, come Halcon e MATLAB, presenta tutte le caratteristiche dei software proprietari. Questo software è stato utilizzato per sviluppare questa tesi, dato che nella cella di micro-manipolazione è stato utilizzato LabVIEW per la gestione dei dispositivi e dei task. Bisogna evidenziare che, per quanto riguarda l'assistenza di NI, non sempre vengono forniti chiarimenti sulle logiche interne delle funzioni o delle librerie.

Nei capitoli successivi si esporranno alcune funzioni messe a disposizione nella libreria **NI Vision**.

Capitolo 2

Image processing e algoritmi di machine vision (MV)

Il sistema di visione è in grado di fornire immagini in formato digitale grazie all'utilizzo di un sensore sul quale viene proiettata l'immagine tramite un sistema di lenti. Il sensore è un rettangolo di silicio su cui sono stati impiantati una serie di elementi fotosensibili (fotodiodi), noti come pixel, dalla dimensione molto ridotta; il sensore può essere di tipo CMOS (Figura [2.1\(a\)](#)) o CCD.

Ogni elemento fotosensibile genera un segnale elettrico proporzionale alla luce che lo colpisce, tale segnale viene convertito da un convertitore analogico digitale (A/D) in un valore chiamato scala di grigio, definito come un valore compreso tra 0 e 255, tale fondo scala deriva dal fatto che il convertitore A/D è a 8 bit; 0 sta per luce assente (nero) e 255 sta per luce massima (bianco); raccogliendo tutti i valori dagli elementi fotosensibili è possibile ottenere l'immagine in bianco e nero di ciò che è proiettato sul sensore. L'immagine è rappresentata da una matrice che avrà un numero di elementi e una disposizione pari a quella dei fotodiodi del sensore. Per ottenere una immagine a colori bisogna applicare una matrice di filtri (rosso, verde e blu) che segue uno schema ben preciso chiamato Filtro di Bayer (Figura [2.1\(b\)](#)). I tre filtri (Red, Green e Blu) eliminano le componenti di colore differenti da quello del filtro, il che permette al pixel di ricevere una quantità di luce direttamente proporzionale al colore del filtro stesso. Con l'utilizzo dei filtri di colore si ottengono immagini RGB (Red, Green e Blu). Nelle applicazioni industriali solitamente con l'utilizzo di telecamere a scala di grigi è possibile estrarre la maggior parte delle informazioni necessarie, ma in alcune applicazioni dove è necessaria l'informazione del colore si utilizzano telecamere a colori; in questo elaborato l'utilizzo di una telecamera a scala di grigi permette l'estrazione di tutte le informazioni necessarie.

Da un'immagine digitale in scala di grigio è possibile ricavare la distribuzione dei valori di scala di grigio, questa viene rappresentata con un istogramma che in ascissa riporta l'intensità in scala di grigio e in ordinata il numero di pixel che assumono un certo valore di intensità (Figura

2.1(c)). Nel caso d'immagini a colori è possibile ricavare la distribuzione per ciascun colore, quindi si otterrà un istogramma per R, uno per G e uno per B (Figura 2.1(d)). L'istogramma associato a un'immagine è utile per l'applicazione di numerose funzioni di image processing (Figura 2.1(c)).

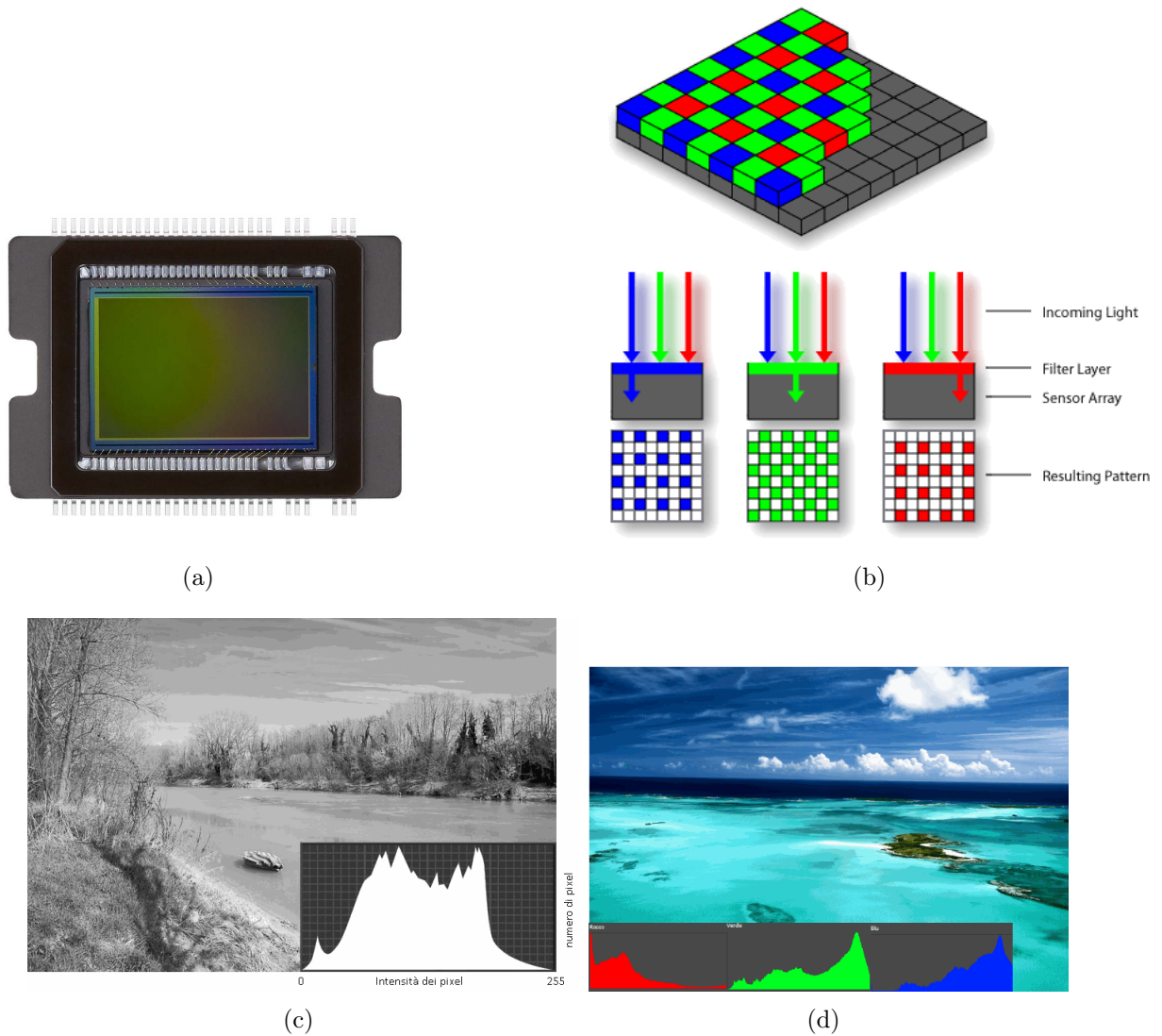


Figura 2.1: (a) Sensore di tipo CMOS. (b) Illustrazione del Filtro di Bayer. (c) Esempio di un'immagine in scala di grigio con il relativo istogramma. (d) Esempio di un'immagine a colori con il relativo istogramma RGB.

2.1 Threshold

L'operazione di threshold permette di evidenziare aree significative di un'immagine in scala di grigi, rendendola adatta per una serie di analisi o di elaborazioni; il metodo è applicabile a immagini che rappresentano oggetti o parti i cui pixel hanno un'intensità appartenente ad

un certo intervallo di scala di grigi definito come **range di threshold**; tutti i restanti pixel, che non appartengono all'oggetto o alla parte, devono avere un valore di intensità fuori dal range e sono considerati come sfondo dell'immagine. L'operazione di threshold imposta tutti i pixel che appartengono al range di threshold a 1 (o 255), o a un valore definito dall'utente, mentre i restanti a 0 (Figura 2.2); facendo ciò si mette in evidenza l'oggetto o il particolare (Figura 2.3) e si genera un'immagine definita **binarizzata**. Il valore di threshold può essere scelto manualmente o in modo automatico con l'ausilio di appositi algoritmi che analizzano l'istogramma dell'immagine [37].

2.1.1 Impostazione manuale del threshold

L'intervallo di threshold è impostato manualmente dall'utente definendo:

- Valore inferiore di threshold (lower threshold, LTH);
- Valore superiore di threshold (upper threshold, UTH).

Tutti i pixel con intensità pari o superiore al lower value e uguale o inferiore all'upper value appartengono al particolare dell'immagine, mentre i pixel che hanno intensità inferiore al lower value e superiore all'upper value appartengono allo sfondo:

$$P(i, j) = \begin{cases} 1 & \text{se } p(i, j) \in [LTH, UTH] \\ 0 & \text{se } p(i, j) \in]0, LTH[\cup]UTH, 255[\end{cases} \quad (2.1)$$

dove $p(i, j)$ è l'intensità del pixel nella posizione (i, j) dell'immagine e $P(i, j)$ è il valore che assumerà il pixel nella stessa posizione dopo l'operazione di threshold.

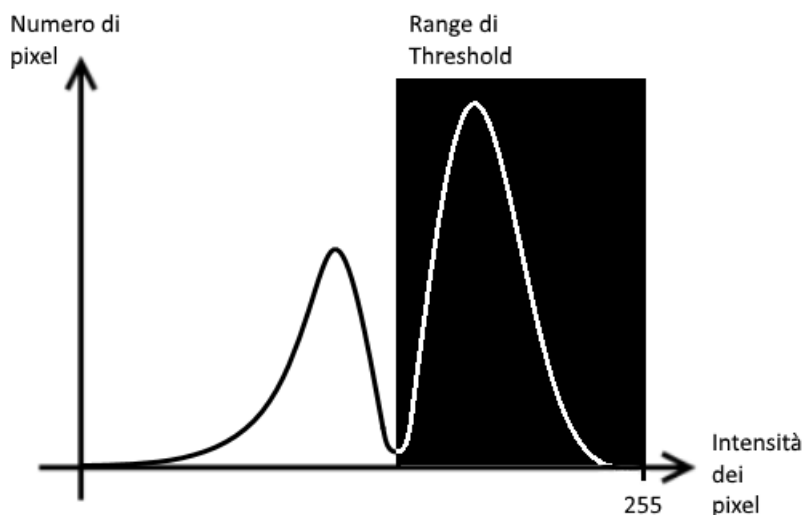
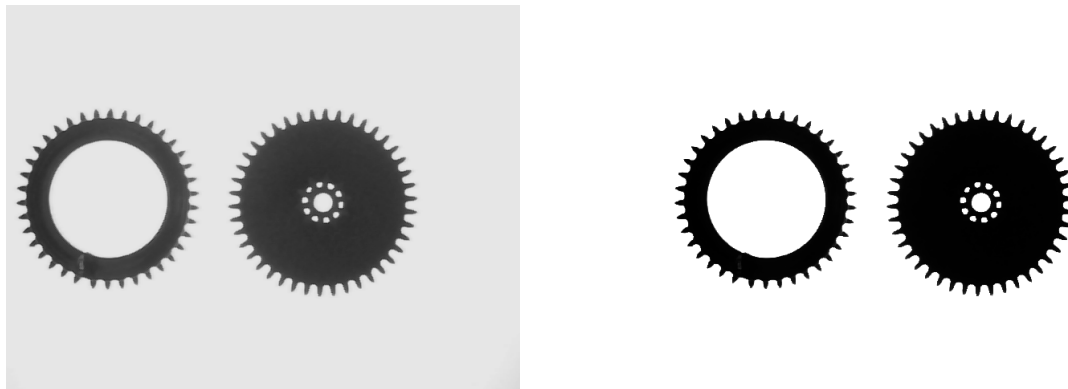


Figura 2.2: Illustrazione di un istogramma su cui si applica il Threshold.



(a) Immagine originale.

(b) Immagine binarizzata.

Figura 2.3: Esempio di applicazione dell'operazione di Threshold su un'immagine.

2.1.2 Impostazione automatica del threshold

Esistono diverse tecniche di threshold automatico che, al contrario del threshold manuale, non richiedono all'utente la definizione dell'intervallo di threshold, che viene invece calcolato in autonomia dalla funzione, ma la scelta del metodo da applicare; è utile nel caso in cui le condizioni di illuminazione cambino da immagine a immagine, poiché si renderebbe necessario calcolare per ciascuna immagine un nuovo valore di soglia per poter binarizzare l'immagine al meglio. Tutte le tecniche di threshold automatico si basano sullo studio dell'istogramma dell'immagine per poter calcolare i limiti di threshold (Figura 2.4).

Di seguito vengono riportati alcuni metodi di threshold automatico [38]:

- **Clustering:** è la tecnica più usata ed è indicata nel caso in cui nell'immagine siano presenti più classi (zone in cui l'intensità dei pixel è circa omogenea); il primo passo è suddividere l'istogramma dell'immagine in un numero di classi corrispondente al numero di gruppi di pixel omogenei percepito nell'immagine, successivamente si ipotizza un primo valore di soglia, si calcolano i baricentri di ciascuna classe e si definisce il nuovo valore di soglia dato dalla media dei baricentri delle classi. Si ripetono quindi le ultime due operazioni finché il nuovo valore di soglia non cambia rispetto al precedente. Il valore di threshold (k) è definito come:

$$k = \frac{\mu_1 + \mu_2}{2} \quad (2.2)$$

dove μ_1 è la media di tutti i pixel che hanno un valore di grigio tra 0 e k , e μ_2 è la media di tutti i pixel che hanno valore di grigio tra $k + 1$ e 255.

- **Entropia:** si basa su una tecnica di analisi di immagini classica ed è la migliore per identificare particolari che sono presenti in piccole proporzioni nell'immagine (come difetti superficiali). Si definisca la curva di probabilità di presenza nell'immagine di ciascun

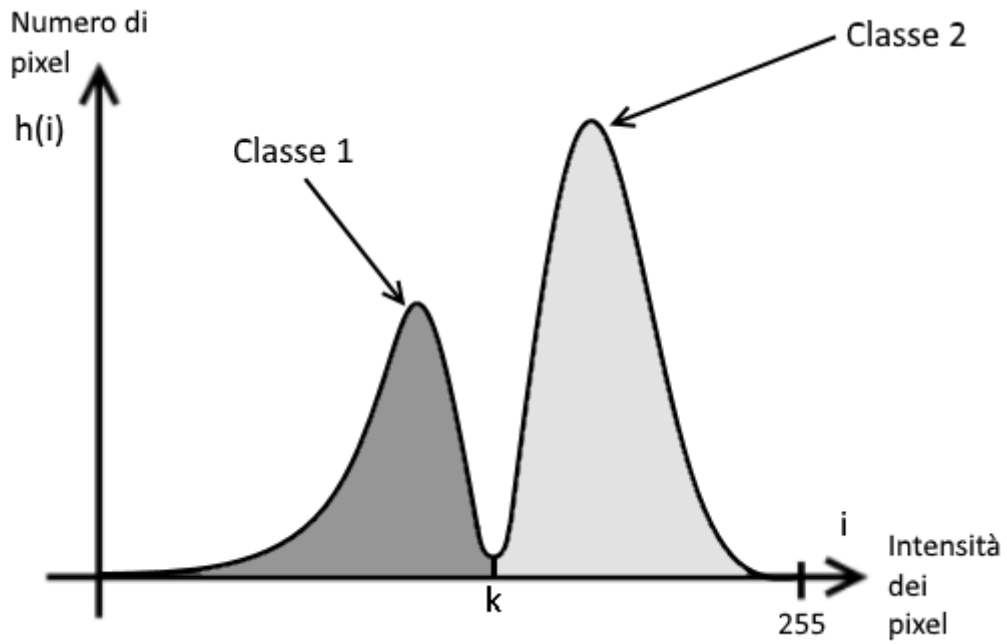


Figura 2.4: Illustrazione dei dati utilizzati dai metodi di Auto Threshold: i è l'intensità di grigio, k è l'intensità di grigio scelta per il threshold, $h(i)$ è la funzione che descrive il numero di pixel ad ogni valore di grigio contenuti nell'immagine, N è il numero totale di livelli di scala di grigio (256 nel caso di un'immagine a 8 bit) e n è il numero totale di pixel in una immagine [38].

valore d'intensità i :

$$p(i) = \frac{h(i)}{\sum_{i=0}^{N-1} h(i)} \tag{2.3}$$

L'entropia dell'istogramma di un'immagine nell'intervallo di intensità $[0, N - 1]$ è definito come:

$$H = \sum_{i=0}^{N-1} p(i) \log_2 p(i) \tag{2.4}$$

Siano $p_b(i)$ e $p_w(i)$ rispettivamente le distribuzioni di probabilità dei pixel dello sfondo (neri) e dell'oggetto (bianchi). Si definiscono:

$$H_b = \sum_{i=0}^k p_b(i) \log_2 p_b(i) \tag{2.5}$$

$$H_w = \sum_{i=k+1}^{N-1} p_w(i) \log_2 p_w(i) \tag{2.6}$$

le entropie relative ai pixel neri (H_b) e bianchi (H_w) dopo l'operazione di threshold. Il valore ottimale di threshold si ottiene massimizzando l'entropia dell'immagine binarizzata

calcolata come:

$$H_b + H_w \quad (2.7)$$

- **Metric:** il valore di threshold k corrisponde al valore che minimizza la funzione che segue:

$$\sum_{i=0}^k h(i) |i - \mu_1| + \sum_{i=k+1}^{N-1} h(i) |i - \mu_2| \quad (2.8)$$

dove μ_1 e μ_2 sono le medie dei pixel rispettivamente con intensità compresa tra 0 e k nel primo caso, tra $k + 1$ e 255 nel secondo.

- **Moment:** tecnica utile nel caso di immagini con basso contrasto; si basa sull'ipotesi che l'immagine osservata sia una versione offuscata di un'immagine originalmente binarizzata. Tale offuscamento si è generato durante il processo di acquisizione a causa di disturbi elettronici e da leggera defocalizzazione. Si definisce il j -esimo momento m di una immagine come:

$$m_j = \frac{1}{n} \sum_{i=0}^{N-1} i^j h(i) \quad (2.9)$$

L'immagine è elaborata come se i momenti statistici delle medie e delle varianze fossero gli stessi sia per l'immagine sfocata che per l'immagine originale, permettendo quindi di ricalcolare l'immagine binarizzata originale. Il valore di threshold è calcolato in modo tale che il momento dell'immagine originale rimanga invariato nell'immagine binarizzata.

2.2 Operazioni morfologiche binarie

Le **Operazioni morfologiche binarie (Binary morphological operations)** effettuano l'estrazione o l'alterazione della struttura di un'immagine **binaria** (Capitolo 2.1); vengono solitamente applicate prima delle operazioni di ispezione dell'immagine per migliorarne la qualità delle informazioni utilizzate per estrarre le caratteristiche geometriche (come area, perimetro, ecc.). Questi miglioramenti sono necessari perché l'immagine, dopo l'operazione di binarizzazione, può contenere caratteristiche indesiderate (ad esempio disturbi, parti che toccano il bordo dell'immagine o con bordo irregolare) che possono influenzare in modo negativo le informazioni estratte. Le funzioni morfologiche, modificando la forma del particolare, rimuovono i difetti migliorandone le informazioni estraibili [37].

Le funzioni agiscono su ciascun pixel in base al valore dei pixel vicini, quindi utilizzano una maschera quadrata binaria 2D (Structuring Element) la cui dimensione e il cui contenuto specificano quali pixel vicini tenere in considerazione per la trasformazione morfologica e per determinare il nuovo valore che deve assumere il pixel elaborato. Solitamente è di dimensione 3x3, e tutti i suoi elementi possono avere valore 0 o 1 (Figura 2.5).

1	1	1
1	1	1
1	1	1

Figura 2.5: Esempio di Structuring Element 3x3.

Ci sono tre fattori che riguardano la maschera che permettono la definizione di quali pixel considerare durante la trasformazione morfologica:

- Dimensione della struttura (DxD): deve essere sempre dispari e determina il numero di pixel, che circondano il pixel elaborato, contenuti nella maschera (Figura 2.6). Usando gli structuring elements si deve aggiungere un bordo attorno all'immagini pari a $\frac{D-1}{2}$ pixel di larghezza. La dimensione determina anche la velocità di elaborazione che diminuisce all'aumentare di D;
- Valore della struttura: serve a definire quali pixel contenuti vengono considerati durante la trasformazione: se il valore di una cella è 1 allora il pixel influenzerà l'elaborazione, viceversa se è zero non la influenzerà (Figura 2.7). In base a quali pixel influenzano l'elaborazione si ottengono risultati differenti.
- Forma del frame: i pixel di un'immagine sono posizionati in una griglia rettangolare, ma le trasformazioni morfologiche permettono di elaborare pixel con una maschera a disposizione quadrata (allineate) o esagonale (sfalsata) (Figura 2.8). Questo parametro è importante per le funzioni che devono alterare l'intensità del pixel in accordo con quella dei pixel vicini (si noti come i pixel dell'immagine non siano fisicamente sfalsati nel frame orizzontale).

Di seguito vengono illustrate alcune alcune operazioni morfologiche.

2.2.1 Erosione (Erosion)

L'erosione elimina i pixel isolati sullo sfondo ed erode i contorni dei particolari in funzione della forma e della definizione della maschera (Figura 2.9). Per un dato pixel P_0 su cui è centrata una maschera, i cui componenti a 1 sono riferiti ai pixel P_i che circondano P_0 , si ha che:

$$P_0 = \begin{cases} 0 & \text{se un pixel } P_i = 0 \\ 1 & \text{se tutti pixel } P_i = 1 \end{cases} \quad (2.10)$$

2.2.2 Dilatazione (Dilation)

La dilatazione elimina piccoli buchi isolati presenti nel particolare e ne espande i contorni in funzione della forma e definizione del Structuring Element; in pratica ha l'effetto contrario

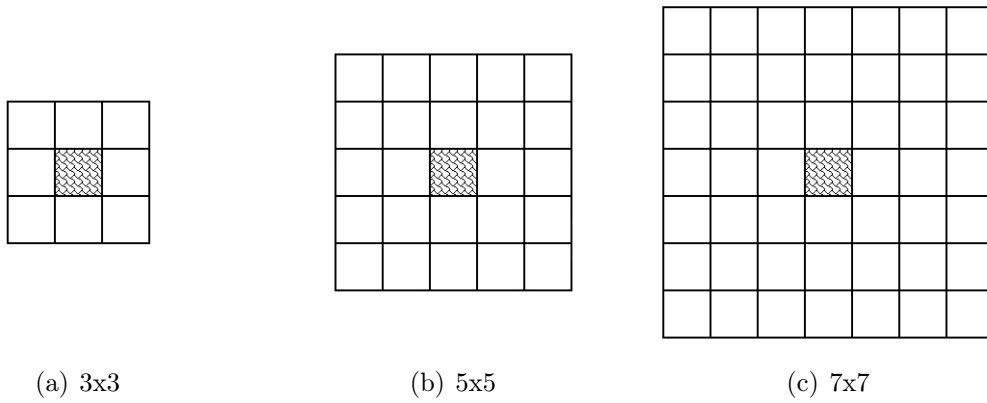


Figura 2.6: Esempi di Structuring Element di differenti dimensioni.

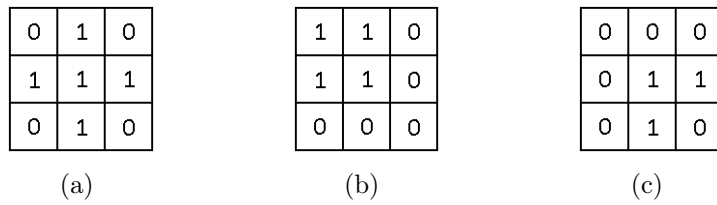


Figura 2.7: Esempi di Structuring Element: valori strutturali differenti.

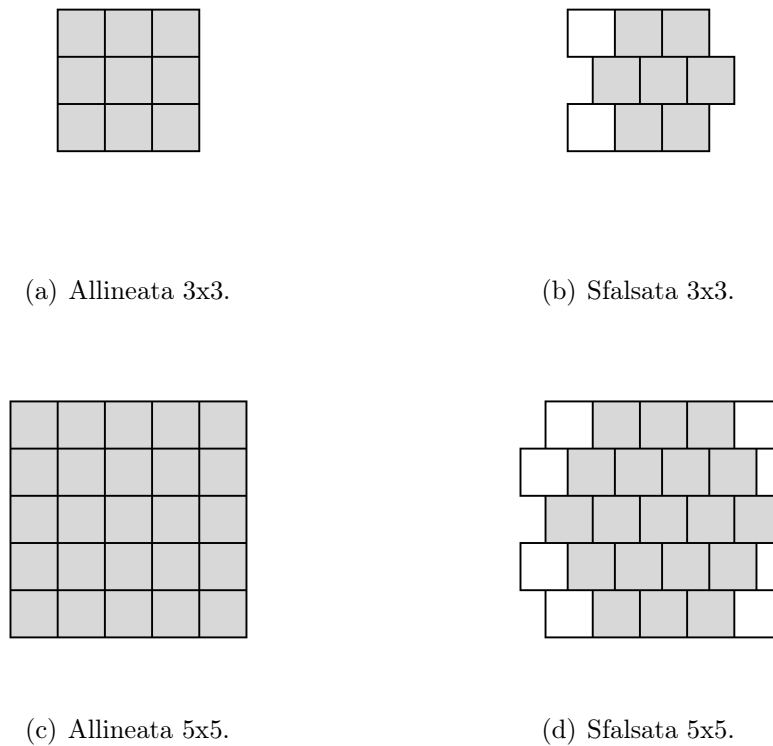


Figura 2.8: Esempi di forme del Structuring Element con differenti dimensioni; i pixel colorati sono quelli considerati durante l'elaborazione.

dell'erosione (Figura 2.10). Per un dato pixel P_0 su cui è stata centrata una maschera i cui componenti a 1 sono riferiti ai pixel P_i che circondano P_0 , si ha che:

$$P_0 = \begin{cases} 1 & \text{se un pixel } P_i = 1 \\ 0 & \text{se tutti pixel } P_i = 0 \end{cases} \quad (2.11)$$

2.2.3 Apertura (Opening)

La funzione di apertura consta in una erosione seguita da una dilatazione; permette di rimuovere piccole particelle e arrotondare bordi senza alterare significativamente l'area e la forma del particolare. Dato che l'erosione e la dilatazione sono trasformazioni duali, i bordi rimossi dall'erosione sono ripristinati dalla dilatazione, tuttavia i piccoli particolari rimossi durante l'erosione non vengono ripristinati dalla dilatazione (Figura 2.11). Definendo con I un'immagine binarizzata la funzione di apertura è definita come:

$$\text{opening}(I) = \text{dilation}(\text{erosion}(I)) \quad (2.12)$$

2.2.4 Chiusura (Closing)

La funzione di chiusura si compone in una dilatazione seguita da una erosione; permette di riempire piccoli buchi e arrotondare bordi senza alterare significativamente l'area e la forma del particolare. Dato che l'erosione e la dilatazione sono morfologicamente complementari, i bordi espansi dalla dilatazione sono rimossi dall'erosione, tuttavia i piccoli fori riempiti durante l'espansione non vengono ripristinati dall'erosione (Figura 2.12). Definendo con I un'immagine binarizzata la funzione di chiusura è definita come:

$$\text{closing}(I) = \text{erosion}(\text{dilation}(I)) \quad (2.13)$$

2.2.5 Eliminazione particolari che toccano il bordo (Border function)

La Border function rimuove i particolari che toccano il bordo dell'immagine binarizzata, che possono essere stati troncati durante l'acquisizione dell'immagine (Figura 2.13). L'eliminazione di tali parti permette di evitare misurazioni e statistiche errate.

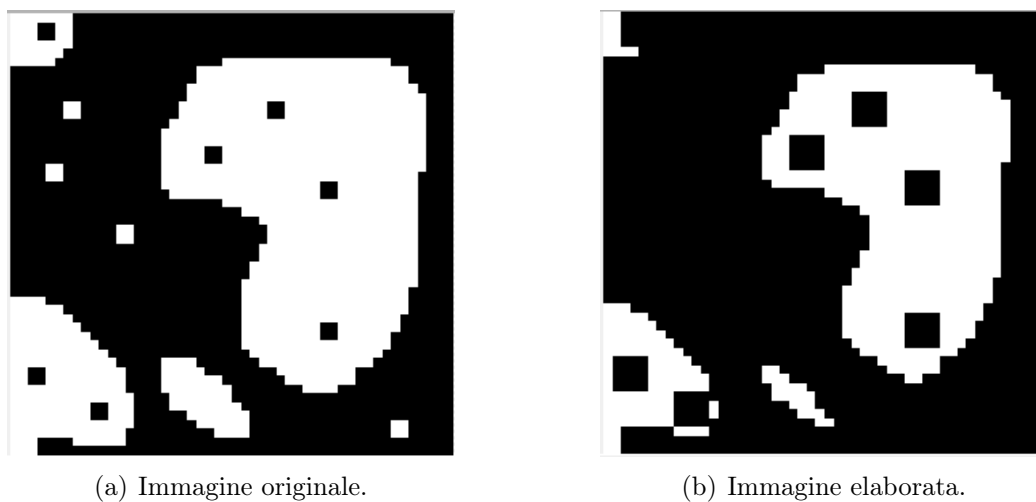


Figura 2.9: Esempio di applicazione dell'erosione.

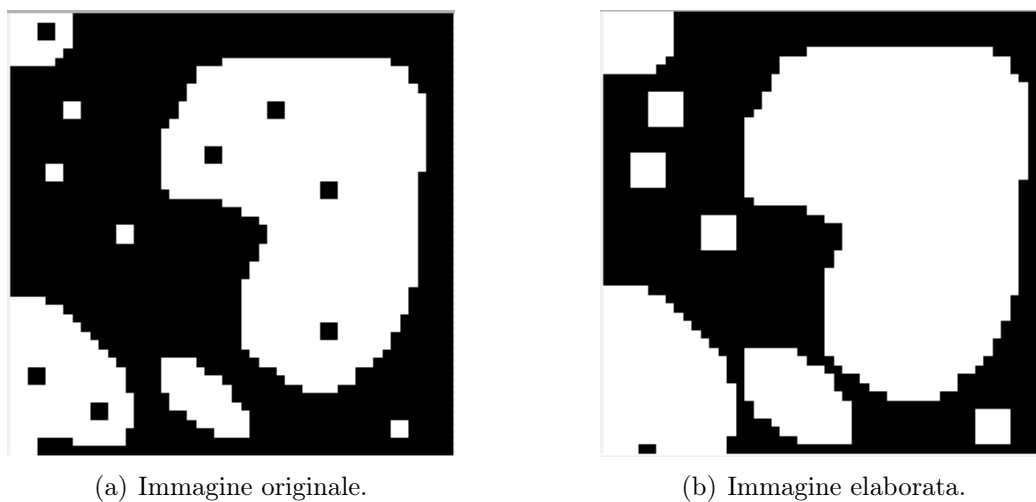


Figura 2.10: Esempio di applicazione della dilatazione.

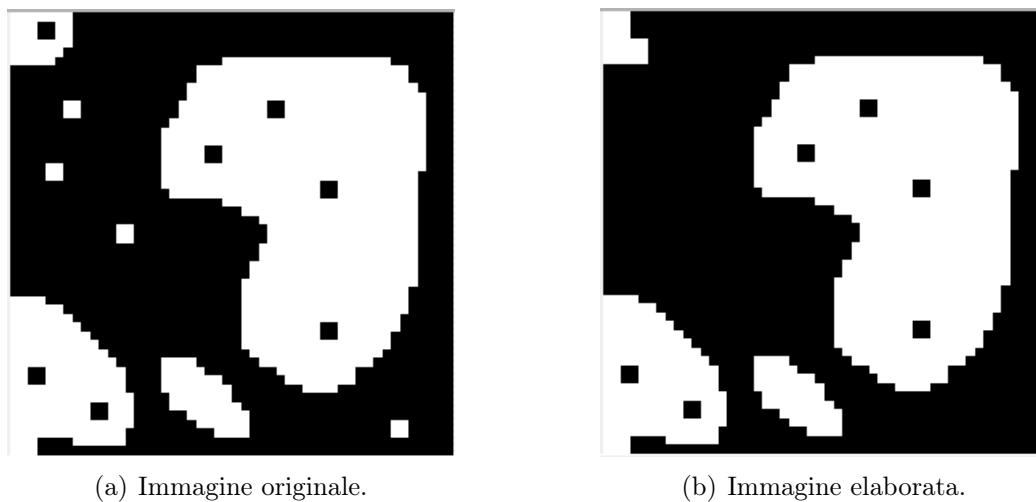


Figura 2.11: Esempio di applicazione dell'apertura.

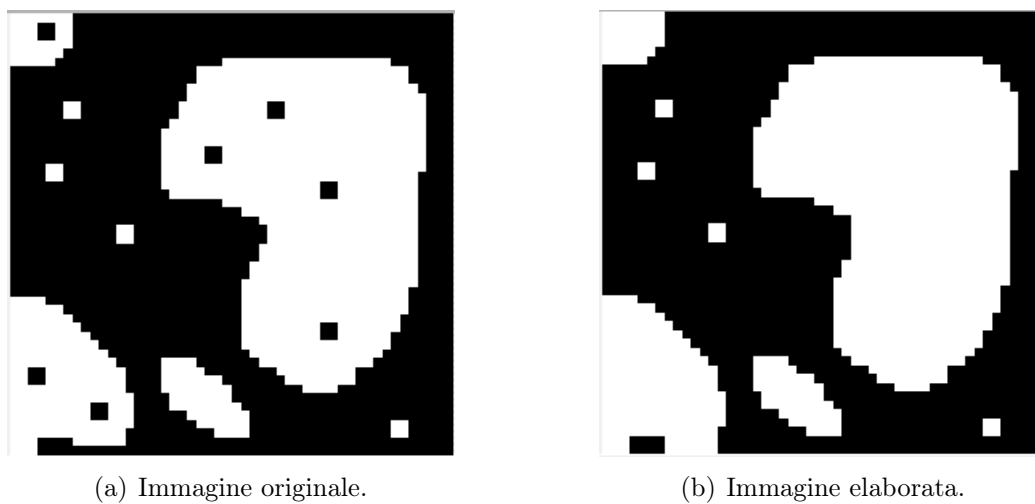


Figura 2.12: Esempio di applicazione della chiusura.

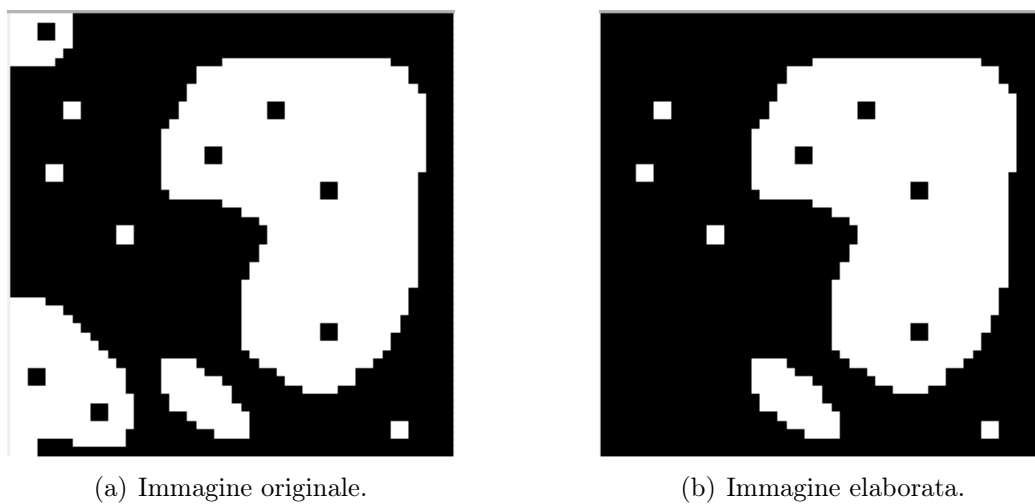


Figura 2.13: Esempio di applicazione del reject border.

2.3 Particle Analysis

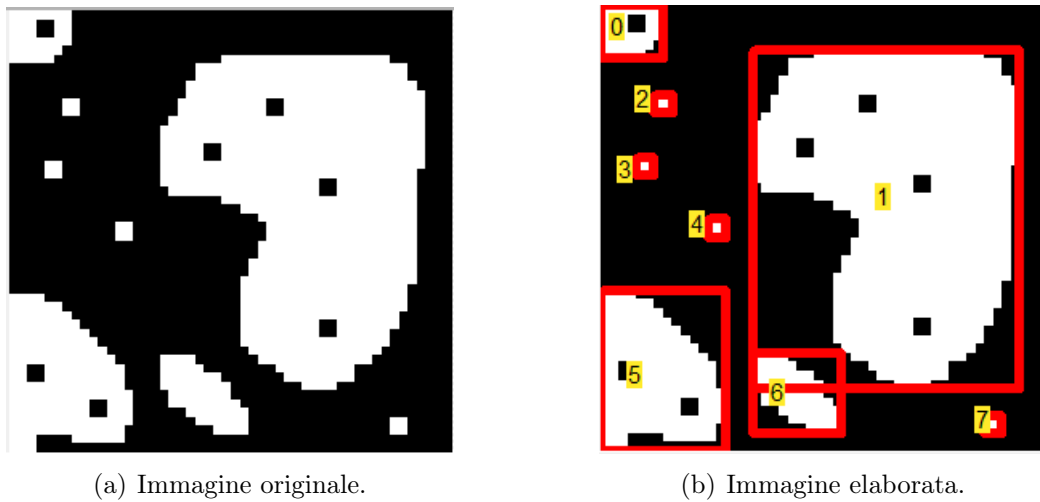
Si utilizza la particle analysis per rilevare regioni connesse (o gruppi di pixel) in un'immagine binarizzata ed effettuare delle misurazioni su queste [37]. Un particolare (o particella) è definito come una regione contigua di pixel con la stessa intensità diversa da zero. Si possono estrarre particelle da un'immagine in scala di grigi effettuando un'operazione di threshold (Capitolo 2.1) in modo da dividere l'immagine in due parti: primo piano, insieme dei pixel con valore pari a 1 (o 255), o al valore definito dall'utente, e sfondo, insieme dei pixel con valore zero; si possono quindi effettuare eventuali elaborazioni morfologiche nel caso in cui ci sia bisogno di migliorare l'immagine (Capitolo 2.2).

La Particle Analysis effettua una scansione all'interno di un'intera immagine, rileva tutte le particelle e per ciascuna di essa compila un rapporto in cui sono presenti differenti sue caratteristiche come: posizione, perimetro, area, centro di massa, ecc.; tali informazioni consentono di creare numerose applicazioni di ispezione per la rilevazione di imperfezioni strutturali o qualitative, o applicazioni di rilevazione di oggetti per il motion control.

Le caratteristiche possono essere inoltre utilizzate singolarmente o in gruppo per poter classificare le particelle o, nel caso di applicazioni di analisi con tempo computazionale elevato, per effettuare una prima selezione dei particolari a cui applicare tale analisi.

Le caratteristiche possono quindi essere utilizzate sia per identificare una particella che per effettuare una prima selezione per poi poter applicare un'analisi differente. Per questo i valori delle caratteristiche devono rientrare all'interno di determinate tolleranze che decretano la conformità o meno di una particella all'applicazione di una successiva elaborazione o alla sua identificazione.

Tutto ciò è possibile nel caso in cui le particelle rilevate presentino almeno una caratteristica distintiva e misurabile dalla Particle Analysis. Di seguito vengono illustrati alcuni metodi di machine vision che possono essere applicati con una qualsiasi libreria di quelle illustrate in precedenza; in questo caso si è preferito illustrare tali metodi riferendosi alla libreria di LabVIEW dato che la National Instrument fornisce un modulo in grado di illustrarli tutti [37].



Particella		0	1	2	3	4	5	6	7
Area [$pixel^2$]		35	789	4	4	4	179	56	4
Numero di buchi		1	4	0	0	0	2	0	0
Orientamento [$^\circ$]		19.59	101.65	0	0	0	116.76	139.68	0
Dimensioni [$pixel^2$]	Larghezza	7	30	2	2	2	14	10	0
	Altezza	6	38	2	2	2	18	9	2
Centro di massa [$pixel^2$]	X	2.71	33.49	6.50	4.50	12.50	5.38	21.29	43.50
	Y	2.43	20.76	10.50	17.50	24.50	41.15	43.05	46.50
Rettangolo esterno [$pixel^2$]	Sinistra	0	17	6	4	12	0	17	43
	Alto	0	5	10	17	24	32	39	46
	Destra	7	47	8	6	14	14	27	45
	Basso	6	43	12	19	26	50	48	48

Figura 2.14: Esempio di applicazione di Particle Analysis.

2.4 Edge detection

L'**Edge detection** è un rilevatore di margine; questo metodo è in grado di rilevare i bordi lungo la linea di pixel di un'immagine (area 1D), ossia un *significativo* cambiamento nella scala di grigi tra due pixel adiacenti di un'immagine, come mostrato in Figura 2.15. L'area di ricerca 1D può essere: una linea, il perimetro di un cerchio, di un'ellisse, di un rettangolo, di un poligono o di una regione disegnata a mano. Il software analizza i valori dei pixel lungo il profilo e rileva significativi cambiamenti di intensità, e permette di specificare le caratteristiche di cambiamento di intensità che costituiscono un bordo. La Figura 2.16 illustra un modello di

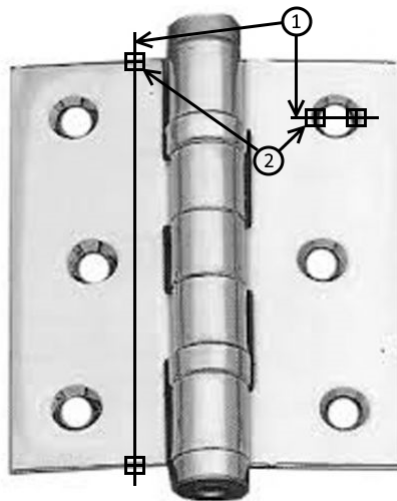


Figura 2.15: Esempio di edge detection: (1) linea di ricerca e (2) bordo rilevato.

caratterizzazione comunemente utilizzato i cui parametri principale sono:

- Forza dell'edge: detta anche *edge contrast*, definisce la minima differenza di scala di grigi tra lo sfondo e l'oggetto. Nella Figura 2.17 vengono mostrate immagini dello stesso oggetto ma che richiedono edge costant differenti. Questo parametro può variare per le seguenti ragioni:
 - condizioni di illuminazione: se l'illuminazione generale della scena è bassa, i profili dell'immagine saranno poco pronunciati;
 - oggetti con caratteristiche di scala di grigi differenti: la presenza di un oggetto molto brillante fa sì che gli oggetti nell'immagine con un'intensità totale più bassa abbiano bordi meno pronunciati.
- Lunghezza di edge: definisce la distanza entro cui la differenza desiderata di intensità tra lo sfondo e il bordo deve avvenire. La lunghezza caratterizza la rapidità di transizione

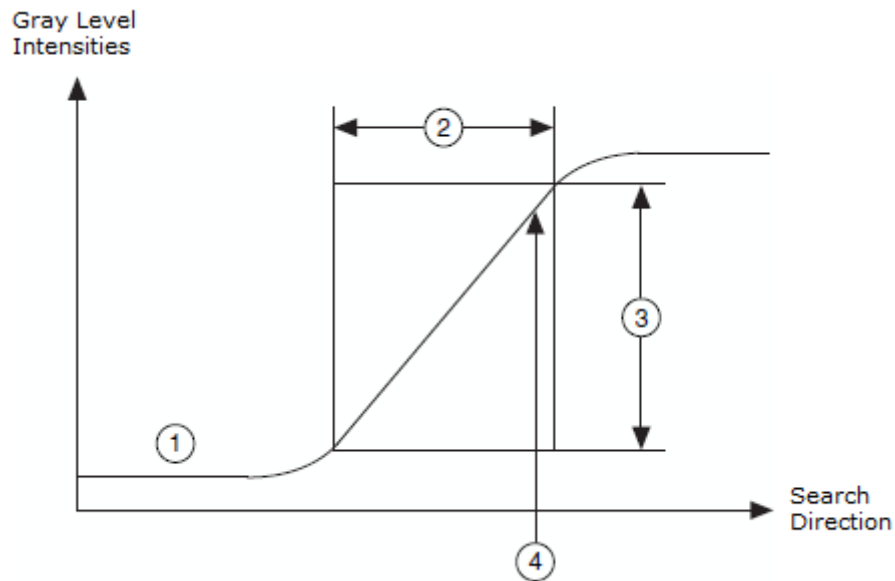


Figura 2.16: Modello utilizzato per la rilevazione di un edge. (1) Profilo della scala di grigi, (2) lunghezza dell'edge, (3) forza dell'edge e (4) localizzazione dell'edge.

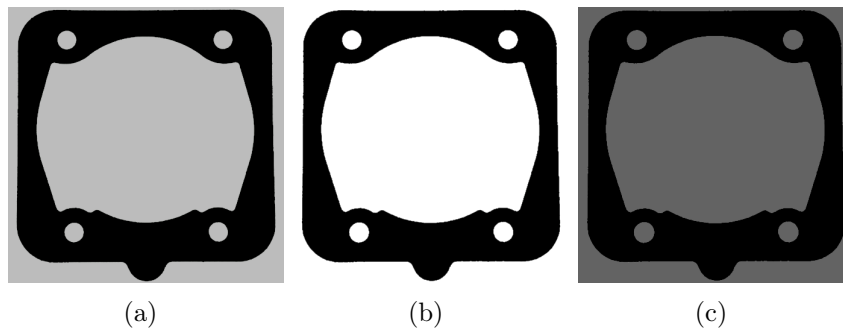


Figura 2.17: Illustrazione del cambiamento dell' edge costante lungo il bordo dell'oggetto relativamente a differenti condizioni di illuminamento.

dell'intensità del bordo. Si usa una lunghezza di bordo più elevata per rilevare bordi con una transizione graduale tra sfondo e bordo.

- Localizzazione dell'edge: è la quota a cui viene rilevato il bordo.
- Polarità dell'edge: un bordo di risalita è caratterizzato da un aumento nel valore di scala di grigi quando si attraversa il bordo. Un bordo di discesa è caratterizzato da un valore di scala di grigi che decresce quando si attraversa il bordo. La polarità del bordo è in grado di stabilire se questo è rilevato in salita o in discesa. Tale caratteristica è legata alla direzione di ricerca. Nella Figura 2.18 vengono illustrati alcuni esempi di polarità di bordo.

L'Edge detection può essere utilizzato in tre ambiti: misurazione, rilevazione e allineamento.

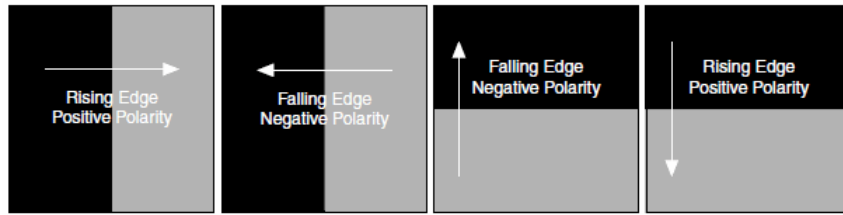


Figura 2.18: Esempi di polarità di bordo

2.4.1 Misurazione

Permette di effettuare misurazioni dimensionali critiche (come lunghezze, distanze, diametri, angoli e qualità) in grado di determinare se il prodotto sotto ispezione è stato lavorato correttamente. Questo permette di verificare se le caratteristiche proprie dell’oggetto rispettano le tolleranze di limite definite dall’utente, così da poter effettuare una selezione dei componenti sotto osservazione.

La misurazione può essere impiegata sia nella produzione inline che offline. Per quanto riguarda il controllo inline, ogni componente è ispezionato per verificare la lavorazione che ha subito (come ad esempio la verifica di assemblaggio nel settore meccanico o l’ispezione in quelli dell’imballaggio elettronico, di container, di fiale in vetro e connettori).

In riferimento alla misurazione offline, vengono valutate le distanze caratteristiche di un oggetto, estratto dalla linea di produzione, verificando se queste sono comprese nel range di tolleranze definite dall’utente. In Figura 2.19 viene riportata l’immagine in cui si mostra un’applicazione di controllo della lunghezza dell’intervallo di una candela.

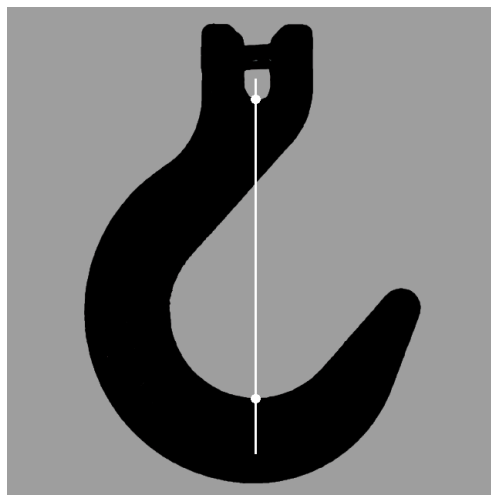


Figura 2.19: Esempio di utilizzo dell’edge detection per la rilevazione dello spazio necessario al funzionamento della candela di innesco.

2.4.2 Rilevazione

Per rilevazione si intende la verifica della presenza di un oggetto ed è tipicamente usata nell'assemblaggio meccanico o di connettori elettronici. Si definisce il numero di bordi che il software deve rilevare, lungo una linea, affinché la rilevazione dia esito positivo.

La Figura 2.20 mostra una semplice applicazione di rilevamento per la verifica di assemblaggio di un connettore. Se vengono conteggiati 8 bordi allora si conferma la presenza di 4 cavi (Figura 2.20(a)), al contrario se si ottiene un conteggio diverso significa che non sono presenti tutti i 4 cavi: in questo caso il pezzo viene scartato (Figura 2.20(b)). Questo metodo può essere utilizzato per rilevare difetti strutturali dell'oggetto analizzato (come cricche, difetti estetici o graffi), e ciò è possibile solo quando il colore dell'oggetto è uniforme, poiché questi difetti si mostrano come un brusco cambiamento chiaro nel profilo di intensità che può essere scambiato per un bordo.

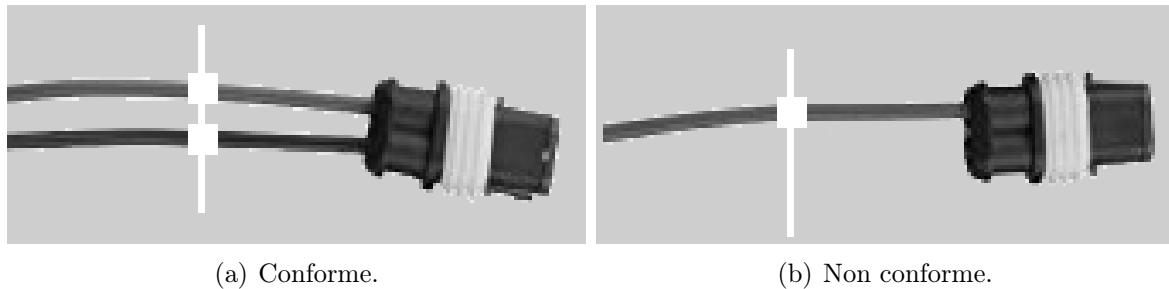


Figura 2.20: Esempio di utilizzo dell'edge detection per il controllo di conformità di un connettore.

2.4.3 Allineamento

Per allineamento si intende la determinazione della posizione e dell'orientamento di un oggetto. Considerando che nelle applicazioni di machine vision gli oggetti che si vogliono ispezionare e/o analizzare possono essere in diverse posizioni ed orientamenti nell'immagine, è quindi utile rilevare questi parametri per ridurre la zona di analisi alla sola area di interesse in modo da ridurre il dispendio di capacità computazionale del computer, ovvero le stesse informazioni possono essere usate per fornire un feedback ad un dispositivo di posizionamento. In Figura 2.21 si mostra un esempio di rilevazione del profilo sinistro di un disco al fine di determinare la posizione e l'orientamento di quest'ultimo (Figura 2.21).

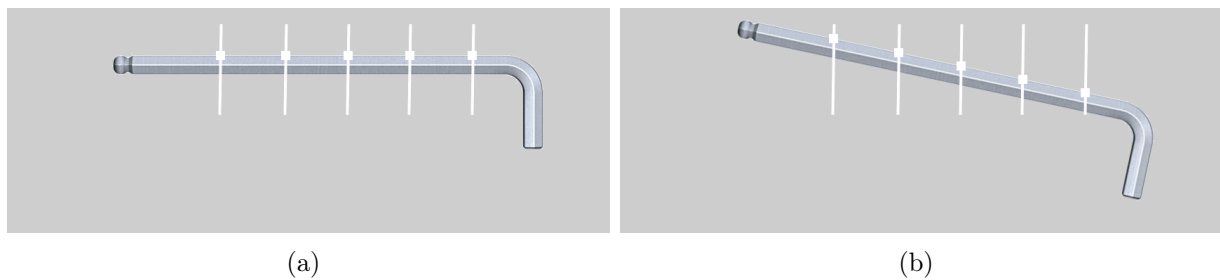


Figura 2.21: Esempio di utilizzo dell'edge detection per la determinazione dell'orientamento di una brugola.

2.5 Pattern Matching

Il **Pattern Matching** fa parte degli algoritmi di *search and match* e localizza regioni in un'immagine che coincidono con una di campione (template) usandone l'intensità dei pixel come caratteristica primaria per l'identificazione. Ci sono diversi metodi di pattern matching, ma tutti si fondano sulla **correlazione incrociata normalizzata (normalized cross-correlation 2.5.1)** e si dividono in due fasi: istruzione e matching.

La cross-correlation è un calcolo che permette di valutare se un'immagine è contenuta in un'altra più grande e, se sì, ne permette la localizzazione.

Per quanto riguarda le fasi di lavoro dell'algoritmo, nella prima avviene l'estrazione dei valori di grigio e/o dei gradienti di bordo dal template, le informazioni vengono poi memorizzate ed associate al template in modo tale da facilitare la velocità di ricerca nell'immagine. Durante la fase di matching l'algoritmo estrae dall'immagine, in cui viene effettuata la ricerca, gli stessi tipi di informazioni estratti nella fase precedente e poi calcola la cross-correlazione tra i due set di dati per ricercarne il matching; per localizzare il matching si prendono in considerazione le regioni in cui il valore di cross-correlazione è più alto. Per ogni istanza identificata viene calcolato uno score che indica quanto questa è rappresentativa del template. Un esempio di applicazione del pattern matching è la ricerca all'interno dell'immagine di un circuito stampato (PCB) di uno o più marker utilizzabili per l'allineamento del dispositivo di montaggio dei chip. Nella Figura 2.22 è mostrata una parte di una PCB (2.22(a)) e il template del marker utilizzato per la ricerca (2.22(b)).

Questo algoritmo è uno dei più importanti tra quelli per il machine vision a causa delle sue numerose aree di applicazione tra cui:

- allineamento: determina la posizione e l'orientamento di un oggetto attraverso la localizzazione di sue caratteristiche;
- misurazione: viene utilizzato per localizzare l'oggetto su cui si vogliono effettuare delle misurazioni;

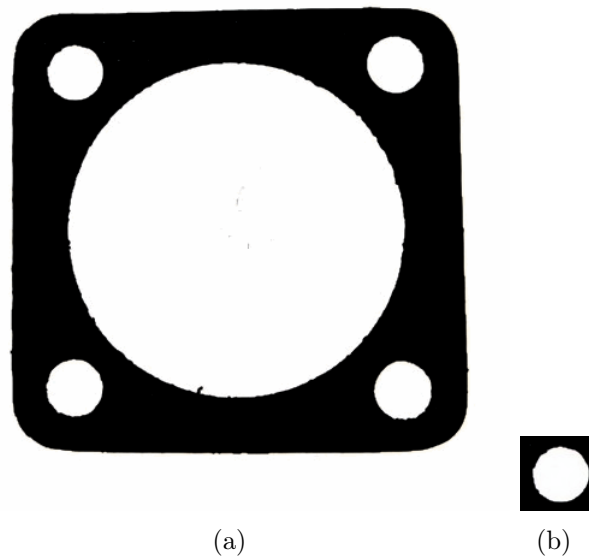


Figura 2.22: Esempio di utilizzo del pattern matching per il riconoscimento dei fori centraggio di una guarnizione.

- ispezione: rileva semplici imperfezioni, come parti mancanti o stampe indecifrabili.

Il pattern matching è il primo passo di analisi per molte applicazioni di machine vision, quindi deve funzionare correttamente nelle condizioni di lavoro più svariate, poiché nelle applicazioni automatiche di machine vision, la visione del materiale o del componente ispezionato può variare a causa di molti fattori (come il cambio di orientamento dell'oggetto, di scala o di illuminazione) rispetto alle condizioni di acquisizione del template. L'elenco che segue riporta i casi in cui il metodo funziona correttamente:

- Cambiamento dell'orientamento del pattern e istanze multiple: il metodo localizza il pattern nell'immagine anche se il pattern nell'immagine è ruotato o scalato (Figura [2.23](#)). Quando un pattern è ruotato o scalato nell'immagine, l'algoritmo può rilevare:
 - il pattern dell'immagine;
 - la posizione del pattern nell'immagine;
 - l'orientamento del pattern;
 - istanze multiple del pattern nell'immagine, se è possibile averne più di una.
- Cambiamento delle condizioni di illuminazione ambientale: il metodo ha la capacità di trovare il pattern di riferimento nell'immagine in condizioni di cambiamenti di illuminazione uniforme in tutta l'immagine (Figura [2.24](#)).
- Cambiamento delle condizioni di offuscamento e in presenza di disturbi: un'ulteriore capacità è quella di trovare il pattern che ha subito alcune trasformazioni a causa di

offuscamenti o rumori nell'immagine (Figura 2.25). L'offuscamento solitamente accade a causa della messa a fuoco errata o di cambiamenti della profondità di campo.

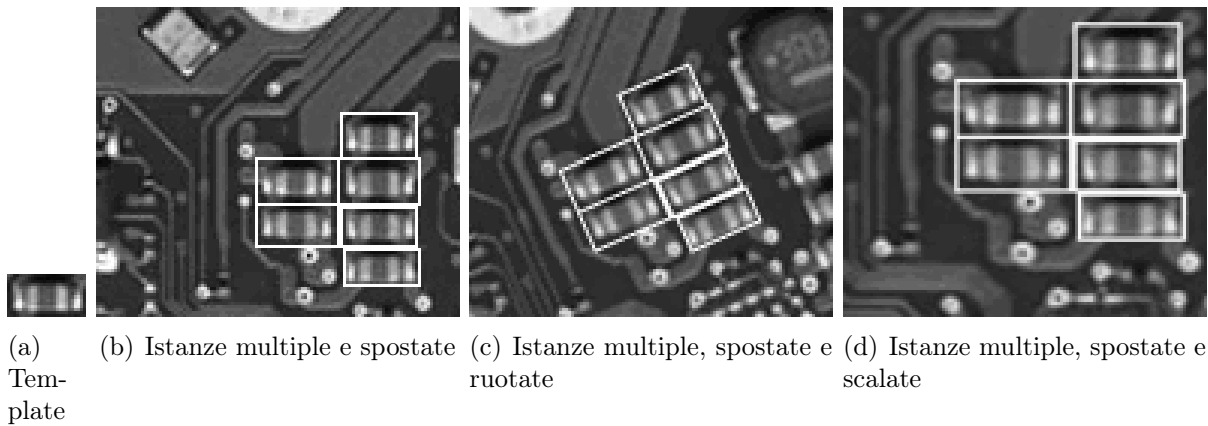


Figura 2.23: Esempi di funzionamento del pattern matching con rilevazione di istanze multiple ruotate e scalate

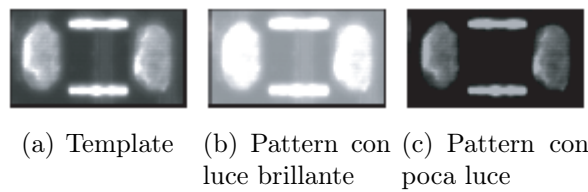


Figura 2.24: Esempio di funzionamento del pattern matching con cambiamenti di illuminazione uniforme

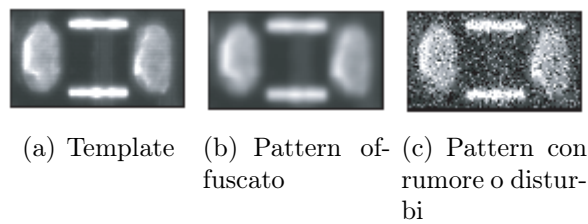


Figura 2.25: Esempio di funzionamento del pattern matching nel caso in cui l'immagine sia disturbata o offuscata

Ci sono alcune limitazioni in cui il pattern matching non funziona correttamente:

- quando l'oggetto è occluso per più del 10% (2.26(a));
- l'illuminazione dell'immagine non è uniforme (2.26(b)).

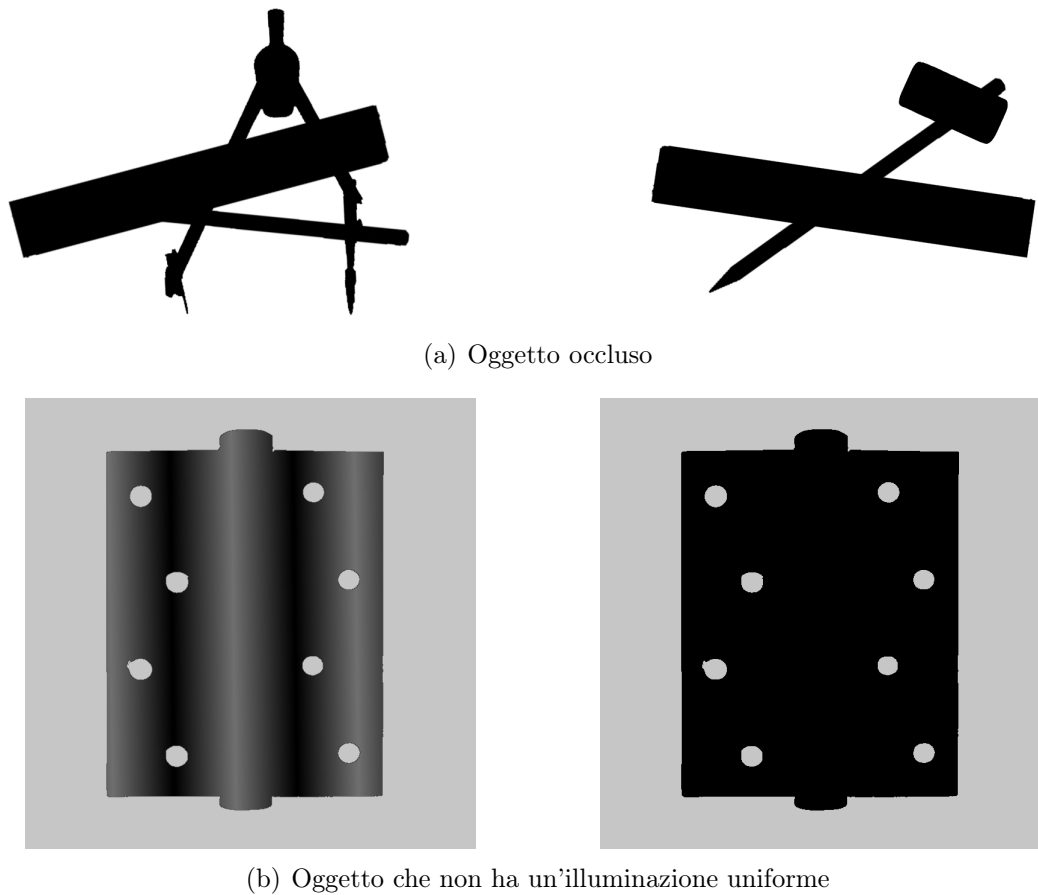


Figura 2.26: Esempi di limitazioni del pattern matching

2.5.1 Correlazione Incrociata Normalizzata

Si consideri una sub-immagine $w(x, y)$ di dimensioni $K \times L$ all'interno di una immagine $f(x, y)$ di dimensione $M \times N$, dove $K \leq M$ e $L \leq N$. La correlazione tra $w(x, y)$ e $f(x, y)$ nel punto (i, j) è data da:

$$C(i, j) = \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} w(x, y) f(x + i, y + j) \tag{2.14}$$

dove $i = 0, 1, \dots, M - 1$ e $j = 0, 1, \dots, N - 1$. La sommatoria è eseguita su tutta l'immagine dove w e f si sovrappongono. Nella Figura 2.27 è illustrato il calcolo della cross-correlation nel punto (i, j) . Si assuma che l'origine dell'immagine f sia nell'angolo in alto a sinistra; la correlazione avviene muovendo il template w in f e calcolando il valore di C per ogni punto di quest'ultima. Questo implica moltiplicare ogni pixel del template per il rispettivo pixel di f ricoperto e sommare i risultati di ogni moltiplicazione. Il massimo valore di C indica la posizione dove w combacia al meglio con f . È però opportuno ricordare che i valori di correlazione non sono accurati sui bordi dell'immagine; inoltre il calcolo è molto sensibile ai cambiamenti di intensità sia nell'immagine che nel template (ad esempio, se l'intensità dell'immagine f è doppia

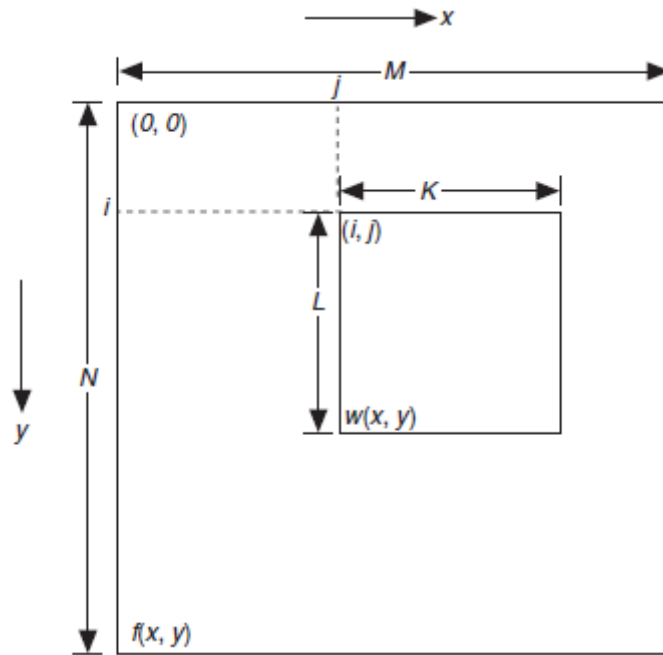


Figura 2.27: Illustrazione del calcolo della correlazione

così lo sarà anche il valore di c). Per sopperire a questo inconveniente si effettua il calcolo della **Normalized Cross-Correlation**:

$$t(i, j) = \frac{\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (w(x, y) - \bar{w})(f(x + i, y + j) - \bar{f}(i, j))}{\sqrt{\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (w(x, y) - \bar{w})^2 \cdot \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (f(x + i, y + j) - \bar{f}(i, j))^2}} \quad (2.15)$$

dove \bar{w} , calcolato solo una volta, è l'intensità media dei pixel del template. Mentre \bar{f} è il valore medio dell'immagine f nella regione coincidente con quella della posizione corrente di w . Il valore di t appartiene all'intervallo tra -1 e 1 ed è indipendente da variazioni di scala d'intensità di f e w .

Come si può notare dalla definizione, la cross-correlation si basa su una numerosa serie di operazioni di moltiplicazione che comportano un elevato dispendio di tempo che non si può trascurare nelle applicazioni real-time. [37] Per incrementare la velocità del processo si può ridurre la dimensione o la regione di immagine in cui ricercare, ovvero applicare il **Piramidal Matching** o l'**Image Understanding** che sono stati sviluppati per incrementare la velocità del processo di riconoscimento.

La cross-correlazione normalizzata è una buona tecnica per la ricerca ma questo solo quando le immagini non sono scalate o ruotate rispetto al template. Tipicamente la cross-correlation ha una tolleranza sulla rotazione del pattern che va da 5° ai 10°, senza però avere cambiamenti di scala. Rendere il metodo di riconoscimento indipendente sia da cambiamenti di scala che di rotazione è molto complesso, basti pensare che per rendere il metodo invariante ai cambiamenti

di scala bisogna:

- scalare il template in base alla variazione di scala che può avere l'oggetto;
- effettuare il calcolo della cross-correlation per ogni template scalato.

Tutto ciò aggiunge un'enorme mole di lavoro al processo di calcolo rendendo così il metodo ancor più lento.

Se si utilizzano il pyramidal matching o l'immagine undestanding non si possono riconoscere oggetti che hanno una scala diversa rispetto al template di riferimento.

In merito al metodo invariante alla rotazione:

- nel caso in cui si conosca la probabile rotazione del pattern: si ruota il template della stessa quantità e si effettua il calcolo della cross-correlation;
- in caso contrario: si devono effettuare diverse rotazioni del template per ottenere il miglior matching possibile.

Si può anche spostare la correlazione nel dominio della frequenza usando le trasformate di Fourier (FFT). Se l'immagine e il template sono della stessa dimensione, questo approccio è molto più efficiente rispetto alla correlazione nel dominio spaziale. Nel dominio della frequenza, la correlazione è ottenuta moltiplicando la FFT dell'immagine per il complesso coniugato della FFT del template, ma la cross-correlazione in questo dominio è molto più difficile da implementare.

2.6 Geometric matching

Il **Geometric Matching** fa parte degli algoritmi di *search and match* e localizza regioni nell'immagine che corrispondono a un'immagine di riferimento (template) usando le sue informazioni geometriche come caratteristiche primarie. In Figura [2.28](#) sono riportati alcuni esempi di applicazione del geometry matching.

Il processo di geometric matching consiste in due fasi: learning e matching. Durante la fase di learning, l'algoritmo estrae le informazioni geometriche dal template di riferimento, le organizza e immagazzina insieme alle relazioni spaziali in modo da facilitare la ricerca nell'immagine da ispezionare. Durante la fase di matching, l'algoritmo estrae le informazioni geometriche dall'immagine da ispezionare, successivamente cerca corrispondenze attraverso la localizzazione di regioni dove le caratteristiche si allineano nel pattern spaziale similmente al template, per ogni istanza trovata calcola uno score che sta ad indicare quando questa assomiglia al template.

Il Geometric matching aiuta a localizzare velocemente oggetti con buone caratteristiche geometriche. In Figura [2.29](#) sono illustrati alcuni esempi di oggetti con buone informazioni geometriche o di forma.

Le caratteristiche geometriche possono essere di due tipi:

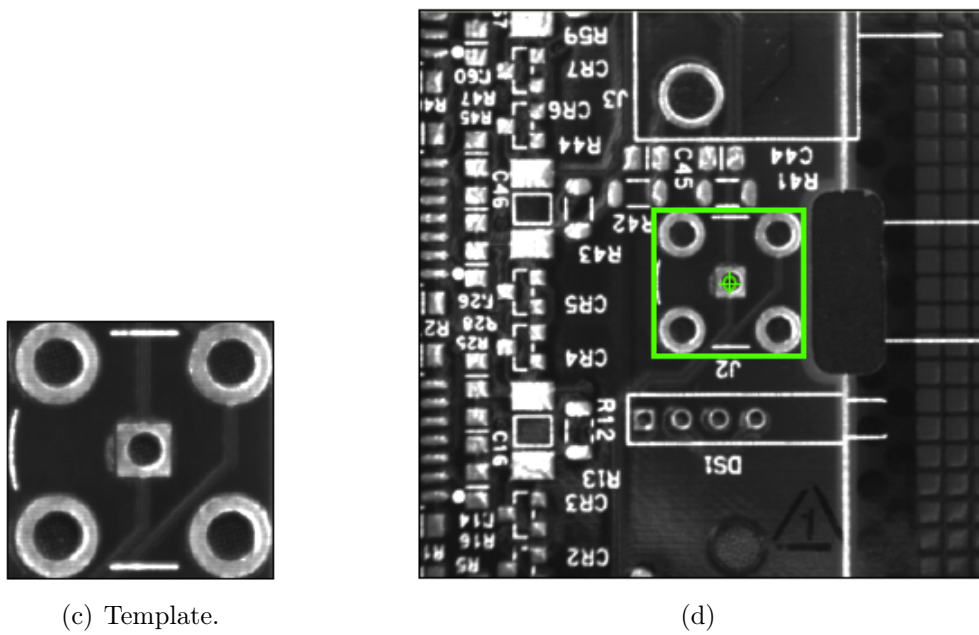
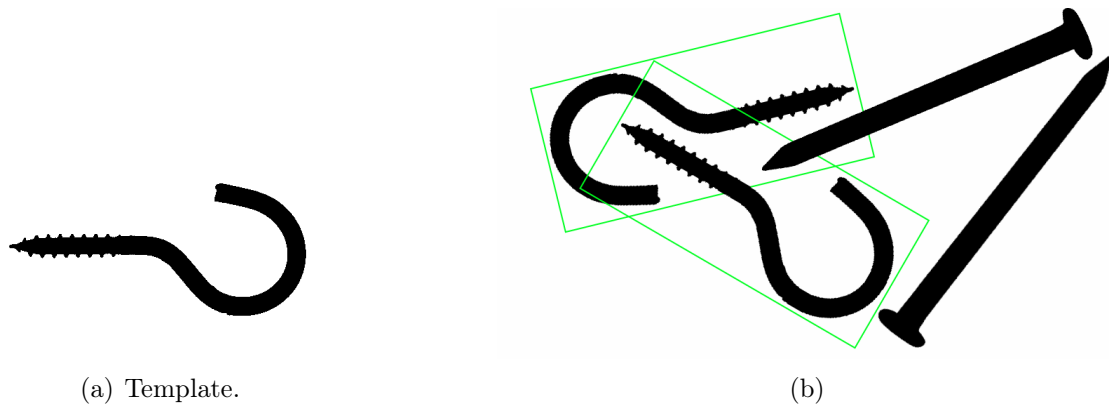


Figura 2.28: Esempi di applicazione del geometric matching.

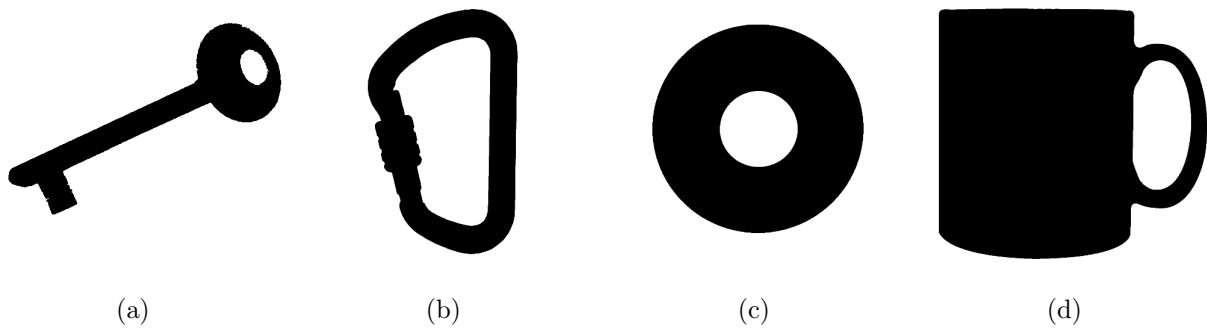


Figura 2.29: Immagini di oggetti che possono essere identificati attraverso il geometric matching.

- di basso libello: come bordi o curve;
- di alto livello: come forme geometriche composte da curve.

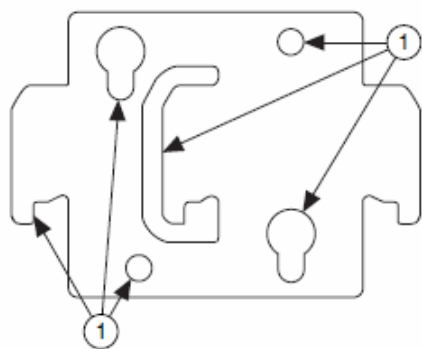
Per questo motivo NI ha implementato due metodi di geometric matching che differiscono per il modo con cui le informazioni delle curve vengono usate per effettuare il matching:

- **Edge-Based Matching Technique** (tecnica basata sul profilo) identifica la curva nell'immagine (basso livello) ed ad ogni punto associa la posizione rispetto al centro dell'immagine; questo insieme di punti rappresenta il template utilizzato per effettuare il matching.
- **Feature-Based Matching Method** (matching basato sulle caratteristiche geometriche) identifica i profili, li approssima con forme geometriche (alto livello) e memorizza le loro posizioni relative, queste caratteristiche insieme vengono usate per effettuare il matching.

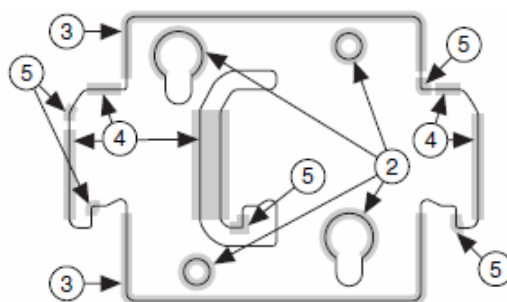
La Figura 2.30(a) mostra le curve che corrispondono ai bordi dell'immagine di template, queste sono alla base delle informazioni che sono usate nell'edge-based technique. La Figura 2.30(b) mostra caratteristiche di forma di alto livello che l'algoritmo feature-based matching method utilizza.

Il metodo va scelto in base al tipo di oggetto che si deve ricercare e alle sue condizioni di visualizzazione:

- per quanto riguarda l'edge-based technique funziona con ogni forma arbitraria e garantisce il ritrovamento di oggetti nell'immagine ispezionata finché una significativa porzione della forma rimane simile alla forma del template. Finché le curve rilevate attorno all'oggetto nell'immagine ispezionata duplicano le curve del template il metodo riuscirà ad effettuare il riscontro.



(a) Edge-Based (caratteristiche di basso livello)



(b) Feature-Based (caratteristiche di alto livello)

Figura 2.30: Tipo di caratteristiche che il geometric matching può estrarre da un template con illustrazione della differenza tra edge-based e feature-based. (1) Curve, (2) caratteristica circolare, (3) caratteristica rettangolare, (4) caratteristica lineare, (5) angoli o spigoli.

- mentre il feature-based matching technique si basa sull'assunzione che la forma nel template può essere rappresentata in maniera affidabile da un insieme di caratteristiche geometriche, quindi dovrebbe essere impiegato solo quando il template e l'immagine ispezionata possono essere rappresentate in maniera regolare e affidabile da forma geometriche come circonferenze, rettangoli e linee.

Infine la memoria e le performance necessarie per l'applicazione possono influenzare la decisione su quale tecnica utilizzare, dato che il template edge-based utilizza più memoria del template feature-based e tale differenza aumenta rendendo il metodo indipendente alla scala (maggiore è la variazione di scala che si vuole rilevare maggiore sarà la dimensione del template edge-based), per la stessa ragione il metodo edge-based è anche il più lento.

Tornando al geometric matchig in generale, si può utilizzare nelle seguenti aree di applicazione:

- misurazione: per localizzare oggetti, o aree di oggetti che si vogliono valutare. Si può utilizzare l'informazione sulla dimensione dell'oggetto per precludere il geometric matching nel caso di oggetti con dimensioni troppo grandi o troppo piccole;
- ispezione: rilevare piccoli difetti, come ad esempio graffi, oggetti mancanti (utilizzando lo score di occlusione fornito dalla funzione), o scritte illeggibili;
- allineamento: determina la posizione e l'orientamento di un oggetto conosciuto attraverso la localizzazione di suoi punti di riferimento o sue caratteristiche peculiari;
- smistamento: smistamento di oggetti basato su forma e/o dimensione. Il geometric matching restituisce la localizzazione, l'orientamento e la dimensione di ciascun oggetto anche quando gli oggetti si occludono a vicenda.

Il geometric matching è un importante strumento per le applicazioni di machine vision, questo funziona in modo affidabile sotto varie, a volte dure, condizioni. Nelle applicazioni automatiche di machine vision, specialmente quelle inserite in un processo manifatturiero, la visione del materiale o del componente sotto ispezione può cambiare rispetto alle condizioni di learning a causa di variazioni dell'orientamento, della scala o dell'illuminazione dell'oggetto. Il geometric matching mantiene la sua abilità di localizzazione nonostante questi cambiamenti. Di seguito vengono descritte le situazioni in cui funziona correttamente:

- Quantità, orientamento e dimensione del componente variabili: l'algoritmo di geometric matching può rilevare i seguenti punti in una immagine analizzata:
 - uno o più template che combaciano;
 - posizione del template identificato;
 - l'orientamento del template identificato;

- cambiamenti di dimensione del template identificato comparato con il template dell'immagine.

Con il geometric matching si possono localizzare template che sono ruotati o scalati di una certa quota. La Figura 2.31 mostra un template di riferimento (Figura 2.31(a)) e un template identificato che è ruotato e scalato rispetto a quello di riferimento (Figura 2.31(b));

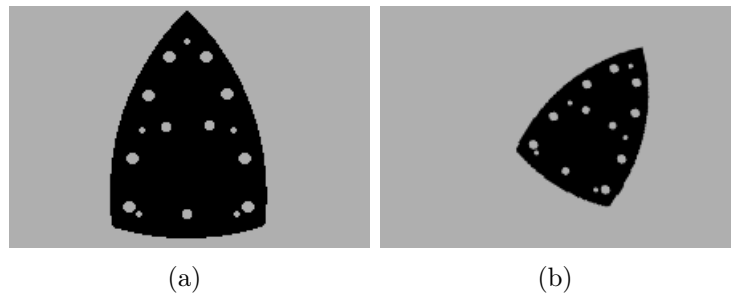


Figura 2.31: Esempio di un riconoscimento di un oggetto ruotato e scalato (b) rispetto al template di riferimento (a).

- Condizioni di illuminazione non lineari o non uniformi: il geometric matching può trovare un template in una immagine in condizioni di cambiamenti di illuminazione dell'immagine non lineari e non uniformi. Questi cambiamenti includono deviazioni luminose, abbagliamenti e ombre. La Figura 2.32(a) mostra un template, mentre in Figura 2.32(b) sono mostrate le tipiche condizioni in cui il metodo riesce a funzionare correttamente.
- Inversione di contrasti: l'algoritmo può trovare template in una immagine anche se il contrasto del template riconosciuto è invertito. La Figura 2.33 illustra un caso tipico di inversione di contrasto. L'algoritmo identifica la Figura 2.33(b) con la stessa accuratezza della parte nella Figura 2.33(a).
- Parziale occlusione: l'algoritmo può trovare un template in una immagine ispezionata anche quando il template è parzialmente occluso a causa del fatto che alcune parti sono coperte o perché alcune parti non sono contenute nei bordi dell'immagine. In aggiunta alla localizzazione di oggetti occlusi, l'algoritmo restituisce la percentuale di occlusione di ogni riconoscimento.

In molte applicazioni di machine vision, la parte sotto ispezione può essere parzialmente occlusa da altre parti che la toccano o la coprono. Inoltre, la parte può sembrare parzialmente occlusa a causa del degradamento del processo manifatturiero. In Figura 2.34 vengono mostrati differenti scenari di occlusione in cui il geometric matching può trovare un template.

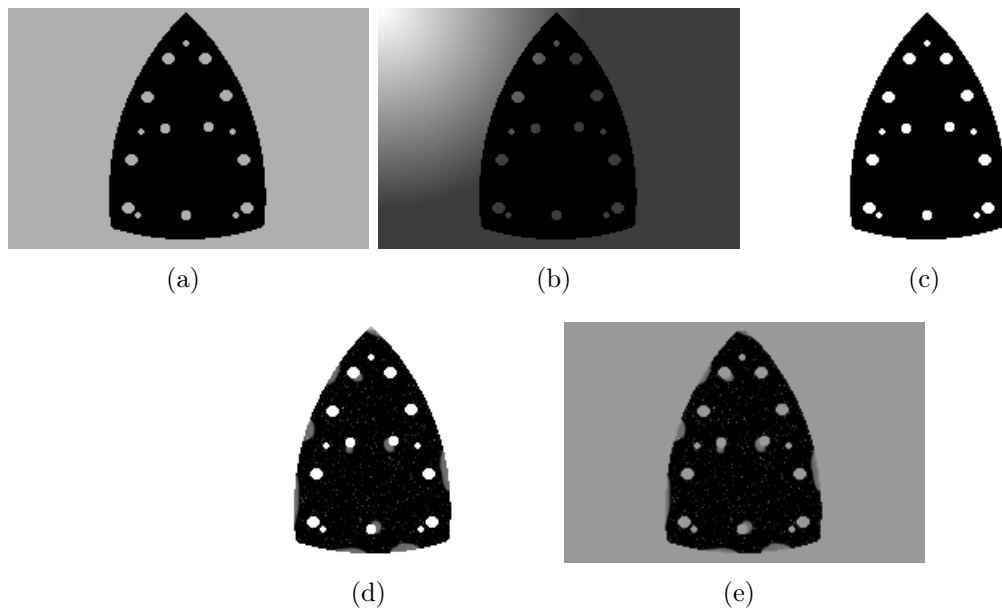


Figura 2.32: Esempio di riconoscimento di un oggetto con diverse condizioni d'illuminamento (b)-(e) rispetto al template (a).

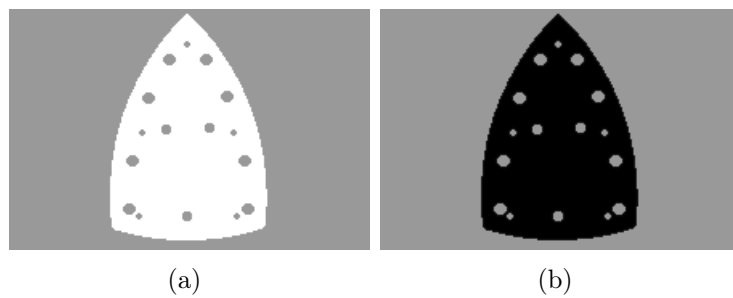


Figura 2.33: Esempio di inversione di contrasto (b) rispetto al template (a).

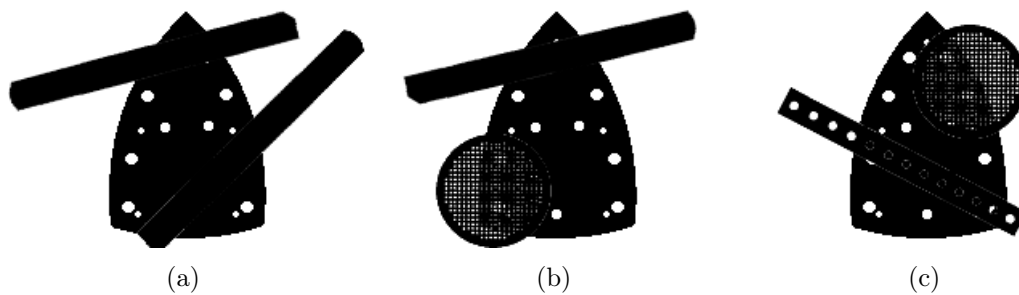


Figura 2.34: Esempio di un riconoscimento di un oggetto occluso.

- Differenti sfondi dell'immagine: l'algoritmo può trovare il template anche se l'immagine ispezionata ha uno sfondo differente rispetto al template di riferimento. Le figure seguenti mostrano esempi di geometric matching in una immagine con diverso sfondo rispetto al template.

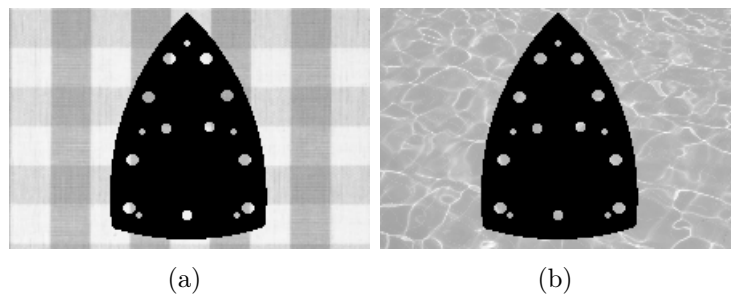


Figura 2.35: Esempio di un riconoscimento dell'oggetto nonostante lo sfondo sia differente rispetto al template.

2.7 Confronto dei metodi

I tre metodi possono essere confrontati e in Tabella 2.1 vengono confrontati i metodi illustrati in precedenza.

Tabella 2.1: Tabella di confronto dei metodi di ricerca [19]

	Metodi di ricerca		
	Edge Detection	Pattern Matching	Geometric Matching
Algoritmo	Edge detection	Cross-correlation normalizzata	Trasformata di Hough generalizzata
Outputs	Numero di bordi rilevati e la posizione lungo il profilo	Numero di istanze trovate e posizione planare	Numero di istanze trovate e posizione planare
Condizioni di ricerca del template		Spostato, ruotato e scalato (fino al $\pm 5\%$)	Spostato, ruotato e scalato
Condizioni ambientali	Nessuna variazione di illuminazione	Variazione di illuminazione uniforme ed immagini sfocate o con disturbi	Illuminazione non uniforme e non lineare, ed immagini sfocate o con disturbi
Tipo di caratteristica analizzata	Sconosciuta o altamente variabile	Intensità del pixel	Caratteristiche geometriche
Tempo computazionale di riconoscimento	Veloce	Medio	Alto
Precisione di riconoscimento	Bassa	Media	Alta

Capitolo 3

Classificazione

La **Classificazione** è un algoritmo di **Machine vision** basato sul **ML (Machine learning)**, che permette di **identificare la classe di appartenenza** di un campione sconosciuto comparando una serie di sue **caratteristiche significative** con quelle di un insieme di campioni **conosciuti** raggruppati in **classi**.

Una **Classe** indica un gruppo di elementi che presentano la stessa modalità di un carattere qualitativo, oppure il valore numerico di un carattere quantitativo uguale o appartenente a un range [4]. Ad esempio se si pensa al settore degli autotrasporti si possono avere le classi: ciclomotore, motociclo, autoveicolo, autocarro, ecc. Oppure nel settore meccanico si possono avere la classi: vite, bullone, rondella, spina di bloccaggio, ecc.

Il **Particle Classifier** è un modulo software che utilizza le caratteristiche geometriche per identificare la classe di un campione; tali caratteristiche possono essere il numero di buchi, la circolarità o l'elongazione e sono estratte effettuando l'analisi dell'immagine binarizzata del campione. Tutte le caratteristiche di un campione vengono inserite all'interno di un vettore chiamato **Feature Vector**.

La classificazione consta di due fasi:

- **Training:** è un'operazione iniziale che ha lo scopo di misurare le caratteristiche peculiari degli oggetti di ogni classe. Durante questa fase infatti si effettua il training del software di machine vision con una serie di campioni associati alle relative classi che si vogliono classificare. Tali campioni saranno utilizzati nella fase successiva per la classificazione. Tutti i gruppi delle classi, contenenti i campioni con le rispettive caratteristiche misurate, vengono immagazzinati in un file denominato **classifier**. Questa fase dovrebbe essere svolta una volta sola, ma è possibile associare nuovi campioni alle classi esistenti o aggiungere nuove classi a un classifier esistente. Questo è un vantaggio che si ha rispetto agli altri metodi di identificazione che non permettono una modifica o un'aggiunta immediata di nuovi campioni da identificare.

- **Classifying:** durante questa fase il campione sconosciuto è confrontato con i campioni di training e gli viene associata una classe. Il campione viene associato alla classe i cui campioni presentano caratteristiche simili al campione da classificare.

La classificazione può essere utilizzata solitamente per due tipi di applicazione:

- *Selezione* di campioni con caratteristiche diverse. Un esempio è la selezione di differenti articoli che transitano su un nastro trasportatore per lo smistamento, oppure l'identificazione di parti meccaniche presenti in una zona di lavoro per effettuare task automatici di assemblaggio.
- *Ispezione* per l'eliminazione di campioni che presentano caratteristiche di non conformità rispetto a quelle dei campioni associati nella fase di training; l'applicazione di machine vision fornisce per ciascun campione in input un output di "conforme" o "non conforme".

3.1 Fasi della classificazione

3.1.1 Training

In base alle classi che si vogliono identificare con il classificatore si deve raccogliere una serie di immagini campione rappresentative delle stesse. Ad ogni classe possono essere associati un numero illimitato di campioni; si noti che, per avere un classificatore robusto, è necessario che ad ogni classe siano associati molti campioni. Ad esempio, se si vuole creare un classificatore che riconosca le classi "vite" e "sfera" occorre raccogliere un certo numero di immagini campione di viti (Figura 3.1) e sfere (Figura 3.2). Dopo aver raccolto un gruppo di campioni per ciascuna



Figura 3.1: Campioni della classe "vite".



Figura 3.2: Campioni della classe "sfera".

classe si determinano l'insieme dei tipi di caratteristiche geometriche da inserire nei feature vectors, che sono necessari per la classificazione. Le feature devono descrivere inconfondibilmente le caratteristiche peculiari delle classi. Dopodiché si effettua l'estrazione delle caratteristiche per ciascun campione e si inseriscono i valori ottenuti nel feature vector di quest'ultimo. Terminata l'estrazione si associano i campioni con i relativi feature vector alle classi. Ciò che si otterrà è un'insieme di classi, ciascuna delle quali sarà composta da un insieme di campioni accoppiati con il proprio feature vector, quest'ultimo conterrà lo stesso tipo di informazioni geometriche in tutti i campioni associati.

Fase fondamentale per l'estrazione delle caratteristiche geometriche del Particle Classifier è l'operazione di sogliatura, che permette di discriminare l'oggetto rispetto allo sfondo. Possono essere applicate ulteriori elaborazioni come:

- rimozione di piccole particelle (erosion, Capitolo [2.2.1](#));
- rimozione di oggetti che toccano i bordi (borde function, Capitolo [2.2.5](#));
- operazioni di chiusura o apertura (closing e opening, Capitolo [2.2.4](#)[2.2.3](#)).

Tutti i parametri di queste elaborazioni devono essere scelti in modo tale da permettere l'estrapolazione delle caratteristiche di tutti i campioni in modo univoco senza alterazioni.

Le feature che descrivono un campione e che permettono d'identificare la sua classe di appartenenza, nel *Particle Classifier*, sono associate ai *descrittori di forma* (*shape descriptor*), i quali a loro volta si basano su *misurazioni di forma* del campione (*shape measurement*); tali misurazioni si estraggono attraverso la particle analysis applicata dopo le operazioni di image processing elencate in precedenza.

Facendo nuovamente riferimento alle classi "vite" e "sfera", le feature che possono descrivere le due classi sono la circolarità e la forma allungata dell'oggetto (elongazione). I feature vector dei campioni delle viti e delle sfere permettono la loro rappresentazione in un piano cartesiano come illustrato in Figura [3.3](#). Dalla rappresentazione cartesiana dei campioni di esempio si deduce che le feature devono essere selezionate in modo tale che i campioni di una classe siano vicini tra loro, ma distanti dai campioni di un'altra classe.

Una volta che si hanno tutte le classi con i rispettivi campioni, rappresentati con i feature vector, si configura il classificatore selezionando il **Classification Method** (Capitolo [3.2.3](#)) e **Distance Metric** (Capitolo [3.2.2](#)) più appropriati alle caratteristiche delle classi. Il classificatore può essere configurato in modo tale che si possano distinguere classi con:

- piccole differenze indipendentemente da variazioni di scala e da riflessioni speculari dell'oggetto rispetto ai campioni di training;
- forme che differiscono solo per fattori di scala;

- forme che differiscono sole per la riflessione speculare;
- forme che presentano tutte le caratteristiche evidenziate nei punti precedenti.

Infine si può generare il file classifier.

Dopo la fase di training e la generazione del classifier, si effettua una fase di test con dei campioni, differenti da quelli usati nella fase di training, di cui si conosce la classe di appartenenza in modo da poter verificare il comportamento del classificatore (Capitolo 3.3). Se dalla valutazione si riscontra un malfunzionamento del classificatore o i risultati ottenuti non sono soddisfacenti si può:

- effettuare da capo la fase di training usando nuovi campioni più rappresentativi;
- aggiungere nuovi campioni a quelli già utilizzati in precedenza;
- effettuare un nuovo training cambiando le feature selezionate, o i parametri di distance metric o classification method.

Le operazioni da eseguire per la fase di training sono riassunte in Figura 3.4.

In alcune applicazioni di machine vision può capitare che nuovi oggetti debbano essere aggiunti a un classificatore esistente, utilizzando il metodo della classificazione questo può essere fatto immediatamente aggiungendo al classificatore esistente la nuova classe con i rispettivi campioni.

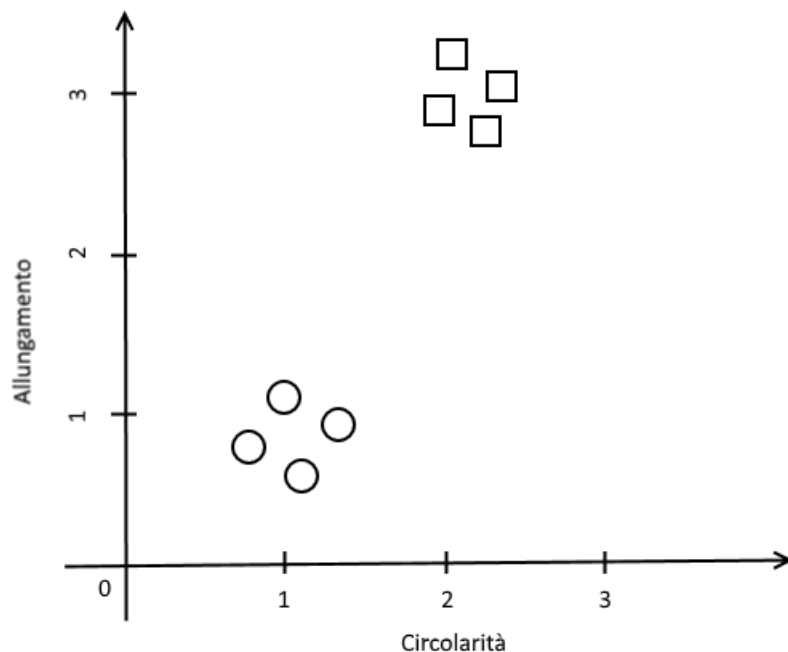


Figura 3.3: Distribuzione dei campioni delle sfere (○) e delle viti (□) nel piano delle feature (circolarità in ascissa e allungamento in ordinata).

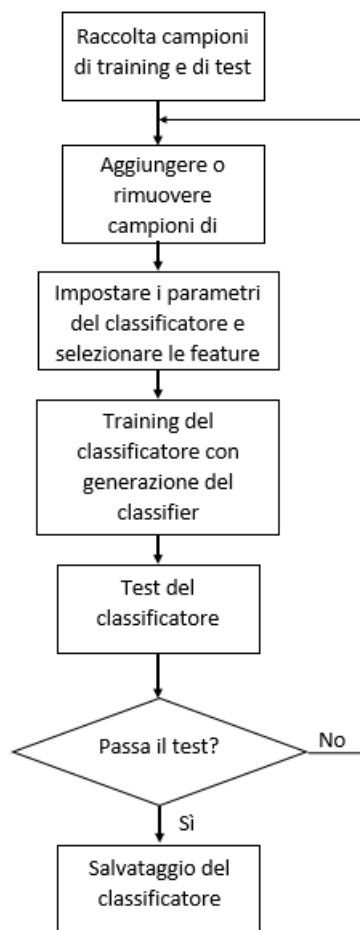


Figura 3.4: Illustrazione del processo di training e di test del classifier.

3.1.2 Classificazione

In questa fase sono svolte le seguenti operazioni:

1. **Elaborazione dell'immagine;**
2. **Estrazione delle feature e compilazione del feature vector;**
3. **Confronto del feature vector del campione sconosciuto con quelli dei campioni di training ed associazione della classe.**

Per quanto riguarda i parametri utilizzati per queste fasi si fa riferimento a quelli utilizzati e impostati nella fase di training.

Il campione sconosciuto verrà quindi associato alla classe del campione le cui caratteristiche vi assomiglieranno di più. La similarità viene misurata con il calcolo della distanza tra il vettore delle caratteristiche del campione sconosciuto e del campione di una classe, che viene calcolata in base al distance metric selezionato. Il metodo di scelta della classe più rappresentativa del campione sconosciuto avviene in base al classification method scelto nella fase di training.

Per valutare la validità della classificazione si fa riferimento al **Classification Confidence score** e all' **Identification Confidence Score** (Capitolo 3.3.3).

In Figura 3.5 si ha un esempio di campione da classificare con il classifier delle "viti" e delle "sfere" menzionato in precedenza. Dopo le operazioni di image processing e di estrapolazione delle feature, si calcola la distanza rispetto a ogni campione di training. In questo caso, poiché il classificatore analizza solo due feature, è possibile identificare la classe di appartenenza visivamente nel piano delle caratteristiche; il campione viene inserito nel piano delle feature (Figura 3.6) e si nota come questo sia più attiguo alla classe "sfera" rispetto alla classe "vite". Ne risulta che la classe associata al campione sarà per l'appunto la classe "sfera".

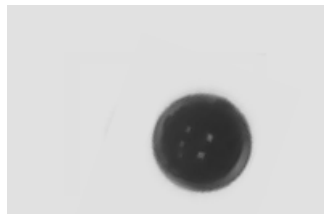


Figura 3.5: Campione sconosciuto da classificare.

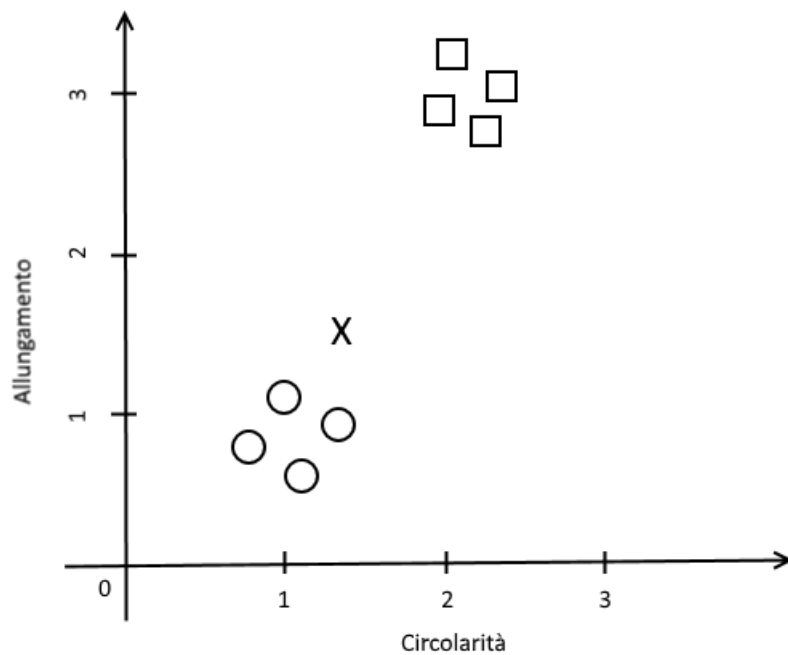


Figura 3.6: Inserimento campione da classificare (X) nel piano delle feature.

3.2 Parametri del classificatore

3.2.1 Feature: estrazione e tipi

L'estrazione delle feature consiste nella misurazione delle caratteristiche significative di un campione in ingresso; questa operazione consente di ridurre i dati necessari per caratterizzare un campione. I campioni di una classe presenteranno valori delle feature simili. A seconda dell'obiettivo del classificatore si utilizzano feature diverse: possono essere righe di pixel o rappresentazioni astratte dei dati dell'immagine. Per le applicazioni di identificazione, le feature da selezionare devono rappresentare efficacemente la separabilità delle classi, ciò significa che il valore di una feature per una classe dev'essere considerevolmente differente da quello di un'altra classe. Per le applicazioni di ispezione è necessario scegliere quelle che permettono la distinzione tra accettabile e non accettabile.

LabVIEW mette a disposizione un Particle Classifier, con feature non selezionabili dall'utente, e un Custom Particle Classifier, in cui le feature possono essere scelte dall'utente, però devono essere estratte e inserite nel feature vector dal programmatore.

NI Particle Classifier

NI Particle Classifier classifica i campioni utilizzando diversi descrittori di forma cioè un feature vector basato su una o più misurazioni effettuate sull'immagine del campione. Le misurazioni si effettuano sull'immagine binarizzata del campione.

Gli Shape Descriptor di default di NI Particle Classifier sono basati su caratteristiche invarianti a variazioni di scala o simmetria speculare. Tuttavia vengono messe a disposizione delle caratteristiche che possono rendere il classificatore:

- dipendente dalle dimensioni del campione (classificatore Scale dependent), utile nel caso in cui le classi presentino caratteristiche di forma simili ma di dimensione diversa, ad esempio per il classificatore di monete;
- dipendente dalla simmetria speculare (classificatore Mirror dependent), utile nel caso in cui si necessiti distinguere classi con caratteristiche di forma simili ma che presentano una simmetria speculare, ad esempio per la classificazione delle lettere p e q);

Lo Scale e il Mirror dependence possono essere applicati singolarmente o insieme; per poter essere applicati NI Particle Classifier utilizza tre classificatori in parallelo, come mostrato in Figura [3.7](#), ciascuno dei quali utilizza caratteristiche:

- dipendenti dalla forma ma non dalla scala e da simmetria speculare;
- dipendenti dalla scala;

- dipendenti dalla simmetria speculare.

I tre classificatori lavorano in parallelo e ciascuno fornisce un risultato. La classe di assegnazione finale è data dalla combinazione pesata dei risultati ottenuti dai classificatori, il peso con cui considerare ciascuno di questi è scelto dall'utente. Bisogna fare attenzione nel caso del classificatore Scale dependent che l'area inquadrata dal sistema di visione che acquisisce i campioni di training e i campioni da classificare sia la stessa, dato che un cambiamento dell'area inquadrata falserebbe i risultati ottenuti con il classificatore che utilizza caratteristiche dipendenti dalla scala. Le caratteristiche di forma di NI non sono fornite, ma sono illustrati i loro concetti:

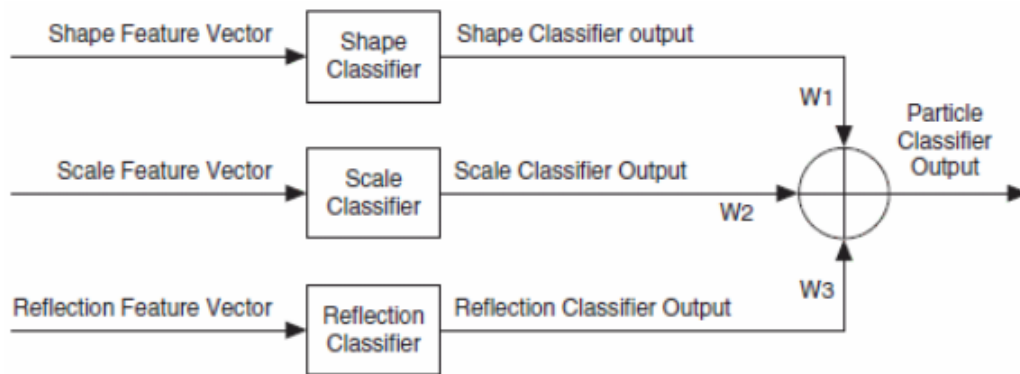


Figura 3.7: Illustrazione del classificatore parallelo di NI Particle Classifier.

- Feature 1: descrive la circolarità di un campione;
- Feature 2: descrive il grado di elongazione di un campione;
- Feature 3: rappresenta la convessità della forma di un campione;
- Feature 4: descrizione della convessità più dettagliata;
- Feature 5: permette di discriminare campioni con fori;
- Feature 6: descrizione dei fori in modo più dettagliato dei campioni con fori;
- Feature 7: descrive la distribuzione del campione;
- Feature 8: descrive la sottigliezza di un campione.

Per rendere il classificatore dipendente dalla scala e da riflessioni speculari, si utilizzano due indici che indicano il peso da dare a queste caratteristiche; possono assumere un valore da 0 (non dipendente) a 1000 (completamente dipendente). La somma dei due indici non deve essere pari o superiore a 1000 e per capire il peso con cui si considerano le caratteristiche di forma si effettua il calcolo $Shape\ Factor = 1000 - (Scale\ Factor + Mirror\ Factor)$.

Custom Particle Classifier

In questa tesi è stato utilizzato il Custom Particle Classifier con cui è possibile selezionare gli stessi distance metric e il classifier method del NI Particle Classifier. Con questo classificatore si definisce il **Personal Feature Vector (PFV)** in cui è possibile inserire le caratteristiche scelte dall'utente, estratte da un campione. In questo elaborato sono state utilizzate le feature che descrivono, come nel caso del NI Particle Classifier, le caratteristiche geometriche del campione, tuttavia è possibile inserire feature differenti come il colore del campione, distanze relative tra elementi, ecc.; l'importante è che tale caratteristica sia esprimibile con un valore numerico. Di seguito sono illustrate le caratteristiche geometriche che si sono utilizzate per compilare il PFV estraibili da un'immagine binarizzata con l'ausilio della Particle Analysis (Capitolo 2.3):

Heywood Circularity Factor : calcolato come il rapporto tra il perimetro di un oggetto e la circonferenza di un cerchio che ha la stessa area dell'oggetto:

$$\frac{P}{2\sqrt{\pi A}} \quad (3.1)$$

dove P è il perimetro dell'oggetto e A è la sua area senza considerare l'area di eventuali buchi. Il fattore può assumere valori da 1 a ∞ ; più la forma dell'oggetto è simile a un disco più il fattore assumerà un valore vicino a 1. Questa feature è utile nel caso in cui una delle classi da identificare presenti una forma circolare rispetto alle altre.

Elongation Factor : fattore definito da alcune grandezze particolari misurate in un'immagine quali:

- il **Massimo Diametro di Feret** (F) è il segmento che collega i due punti più lontani del perimetro dell'oggetto (Figura 3.8);
- la **Convex Hull Area** (A_{CH}) rappresenta l'area del più piccolo poligono convesso che contiene tutti i punti dell'oggetto (Figura 3.9);
- il **Lato Corto Equivalente di Feret** costituisce il lato corto di un rettangolo che ha la stessa area dell'oggetto e il lato lungo pari alla lunghezza del Massimo Diametro di Feret:

$$RF_b = \frac{A_{CH}}{F} \quad (3.2)$$

L'**Elongation Factor** è calcolato come il rapporto tra il Massimo Diametro di Feret e il Lato Corto Equivalente di Feret:

$$\frac{F}{RF_b} \quad (3.3)$$

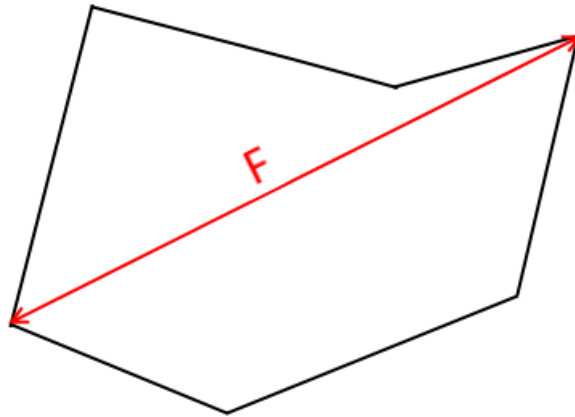


Figura 3.8: Illustrazione del Massimo Diametro di Feret (D).



Figura 3.9: Illustrazione della Convex Hull Area. In nero è rappresentato l'oggetto, mentre la linea rossa indica il più piccolo poligono convesso che contiene tutti i punti dell'oggetto.

Il fattore può assumere valori da 1 a ∞ : più la forma dell'oggetto è allungata, maggiore sarà il valore del fattore. Questa feature è vantaggiosa nel caso in cui una delle classi da identificare presenti una forma allungata rispetto alle altre, oppure le classi si distinguono in base a quanto la loro forma è allungata.

Compactness Factor : vengono prima definite le grandezze particolari che è possibile misurare in un'immagine:

- le **Linee di bordo destra** (B_R) e **sinistra** (B_L) rappresentano le coordinate in x rispettivamente della particella più a destra e più a sinistra;
- le **Linee di bordo alta** (B_T) e **bassa** (B_B) descrivono invece le coordinate in y rispettivamente della particella più alta e più bassa;

- le **Larghezza** (W) e **Altezza** (H) di bordo vengono definite come:

$$W = B_R - B_L \quad H = B_B - B_T \quad (3.4)$$

Queste feature permettono di quantificare l'ingombro dell'oggetto (Figura 3.10).

Il **Compactness Factor** è definito quindi come il rapporto tra l'area dell'oggetto e il prodotto della larghezza e dell'altezza di bordo:

$$\frac{A}{W \cdot H} \quad (3.5)$$

Può assumere valori da 0 a 1 e permette di rilevare come un oggetto è posizionato rispetto ai suoi ingombri massimi. Può essere sfruttato ad esempio nel caso in cui le classi identificano diverse posizioni in cui trova l'oggetto.

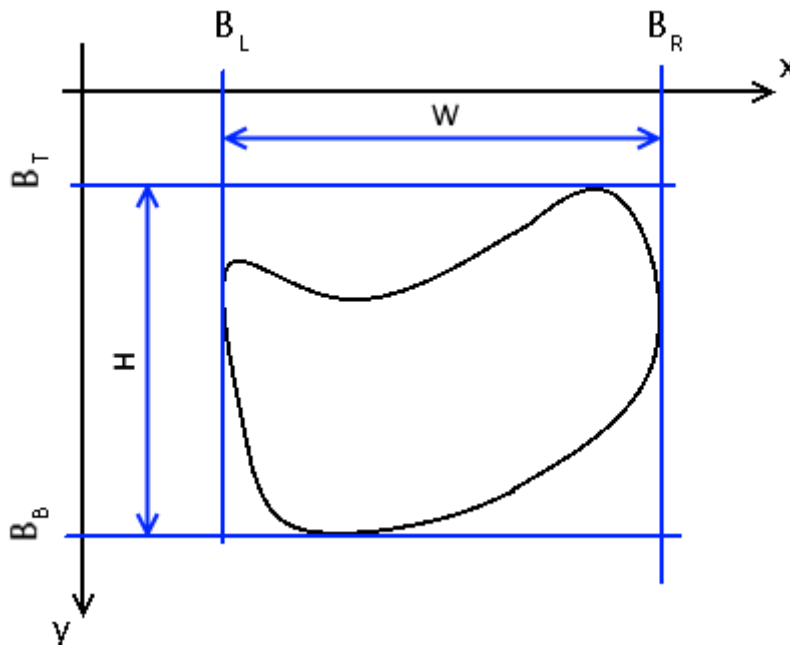


Figura 3.10: Illustrazione dei valori B_R , B_L , B_T , B_B , W e H rispetto al profilo di un oggetto.

Convex's Perimeter Ratio : definito come il rapporto tra il perimetro della Convex Hull Area (P_{CH}) e il perimetro dell'oggetto:

$$\frac{P_{CH}}{P} \quad (3.6)$$

Permette di evidenziare la presenza di concavità in un oggetto, come ad esempio nel caso di una ruota dentata che presenta numerose cavità dovute alla dentatura (Figura 3.11). Se questa feature non permettesse di raggiungere risultati accettabili si può utilizzare il Convex's Area Ratio o viceversa.

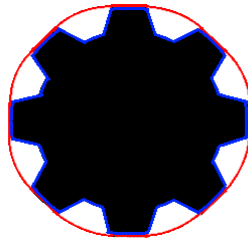


Figura 3.11: Esempio di applicazione del Convex's Perimeter Ratio applicato su una ruota dentata.

Convex's Area Ratio : rapporto tra l'area dell'oggetto, comprensiva dei suoi buchi, e la Convex Hull Area:

$$\frac{A_T}{A_{CH}} \quad (3.7)$$

dove A_T = Area totale dell'oggetto comprensiva dell'area dei fori. Questo è un ulteriore rapporto che evidenzia la presenza di oggetti con concavità (Figura [3.12](#)).

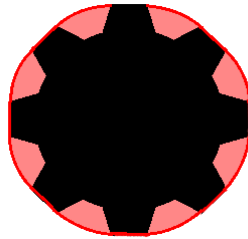


Figura 3.12: Esempio di applicazione del Convex's Area Ratio applicato su una ruota dentata.

Number Of Holes : conteggio del numero di fori presenti in un oggetto. Questa feature è molto importante per poter identificare oggetti in base al numero di fori che presentano. Si pensi alla classificazione di componenti meccanici quali ruote dentate alleggerite, che possono essere distinte in base al numero delle cavità di alleggerimento (Figura [3.13](#)). Utilizzando questa funzione è opportuno fare molta attenzione in quanto è possibile che l'immagine analizzata presenti dei disturbi che potrebbero essere conteggiati come fori.

Hole's Area Ratio : rapporto tra l'area dei fori e l'area totale dell'oggetto comprendente i fori:

$$\frac{A_H}{A_T} \quad (3.8)$$

dove A_H = Area dei fori. Questa feature è utile per poter caratterizzare componenti che presentano fori, calcolando la quantità di area forata che ricopre l'area totale dell'oggetto. Inoltre questa feature non ha problemi nel caso in cui ci siano disturbi nell'immagine (Figura [3.14](#)).

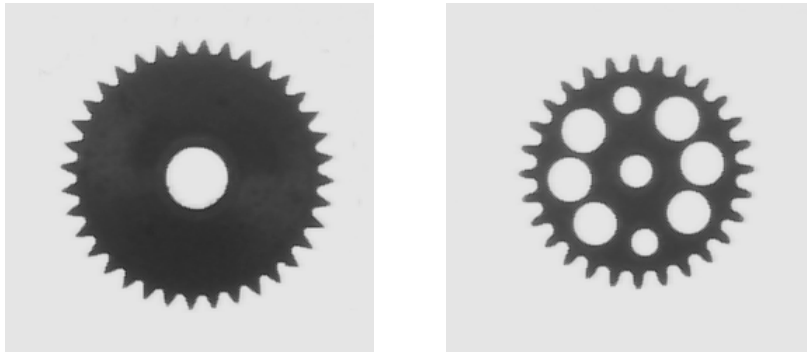


Figura 3.13: Esempio in cui il Number of Hole permette di discriminare la ruota alleggerita da quella non alleggerita.

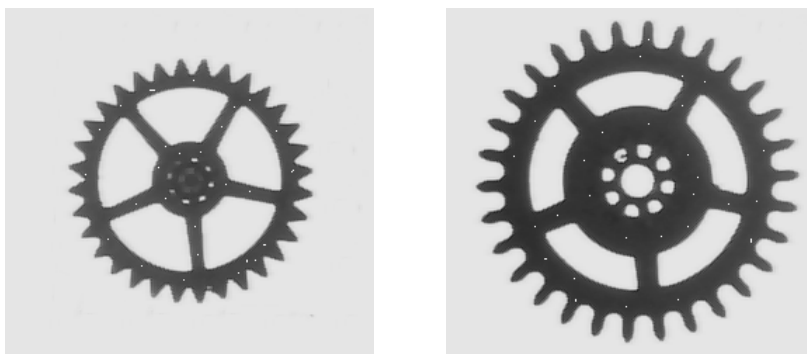


Figura 3.14: Esempio in cui l'Hole Area Ratio permette di discriminare le due ruote dentate in base alla parte di area totale dell'oggetto costituita da fori, tutto ciò nonostante il rumore che può essere percepito come tanti fori.

Scale Area ratio ‰ : rapporto tra l'area totale dell'oggetto comprendente i fori e l'area di tutta l'immagine che racchiude l'oggetto:

$$\frac{A_T}{A_I} \cdot 1000 \quad (3.9)$$

dove A_I è l'area di tutta l'immagine. Questa feature, se utilizzata, permette di tenere in considerazione la grandezza dell'oggetto, che equivale a rendere il classificatore dipendente dalle dimensioni dell'oggetto. Bisogna fare attenzione che l'area inquadrata dal sistema di visione che acquisisce i campioni di training e i campioni da classificare sia la stessa, dato che un cambiamento dell'area inquadrata falserebbe il valore dello Scale Area ratio calcolato (Figura 3.15).

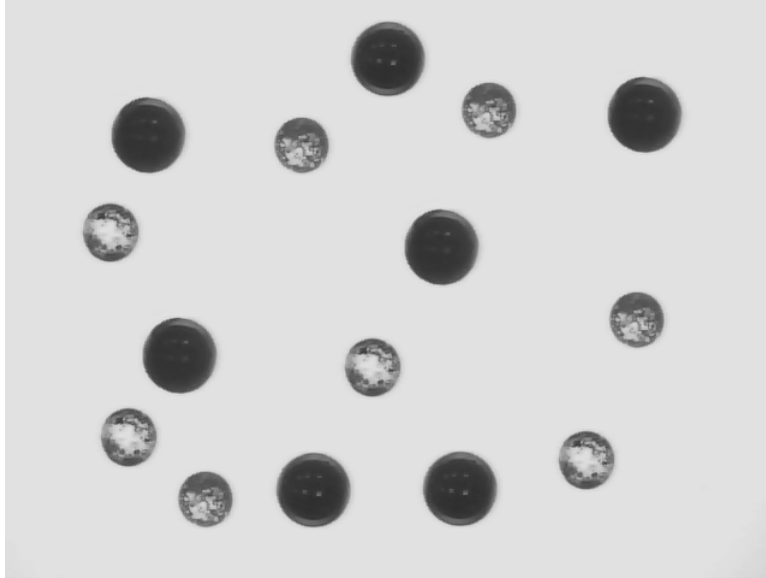


Figura 3.15: Esempio in cui l'unica caratteristica che contraddistingue le due sfere è l'area occupata da queste nell'immagine. In questo caso, lo Scale Area Ratio permette l'identificazione delle due classi.

3.2.2 Distance metrics

Per identificare la classe di appartenenza di un campione per prima cosa ne si estraggono le n caratteristiche geometriche e si inseriscono all'interno del feature vector associato ($X = [x_1, x_2, x_3, \dots, x_n]$), dopodiché si calcolano le distanze di quest'ultimo rispetto al feature vector di ogni campione associato alle classi, si indichi con $Y = [y_1, y_2, y_3, \dots, y_n]$ il feature vector di un generico campione. Infine valutando i risultati ottenuti si assegna la classe al campione sconosciuto.

La funzione di classificazione permette di calcolare la distanza tra il featur vector del campione sconosciuto X e quello di un generico campione Y nei seguenti modi:

- **Euclidean distance** ($L2$)

$$d(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.10)$$

Questo metodo è utilizzato nel caso in cui i campioni associati alle classi presentino piccole differenze all'interno della stessa, essendo questo metodo meno sensibile alle piccole variazioni tra i campioni [11].

- **Sum distance** ($L1$)

$$d(X, Y) = \sum_{i=1}^n |x_i - y_i| \quad (3.11)$$

Questo metodo è utilizzato nella maggior parte delle applicazioni di classificazione. È anche conosciuto come **Manhattan metric** o **Taxicab metric** [11].

- **Maximum distance** (L_∞)

$$d(X, Y) = \max_i |x_i - y_i| \quad (3.12)$$

Utilizzato per classificare campioni con piccolissime differenze tra le diverse classi, in quanto risulta essere il più sensibile tra i metodi proposti [11].

3.2.3 Classifier method

Per l'assegnazione della classe al campione sconosciuto si possono utilizzare tre criteri: **Nearest Neighbor Classifier**, **K-Nearest Neighbor Classifier** e **Minimum Mean Distance Classifier**. In tutti i criteri la distanza d viene calcolata con il metodo scelto nella fase di training.

Nearest Neighbor Classifier (NN)

L'algoritmo NN valuta la distanza del campione sconosciuto X rispetto a tutti i campioni di training e tiene in considerazione solo la distanza del campione più vicino per ciascuna classe:

$$d(X, C_j) = \min_j d(X, X_i^j) \quad (3.13)$$

dove $d(X, X_i^j)$ è la distanza tra X e X_i^j . Il metodo assegna al campione X sconosciuto la classe del campione più vicino:

$$X \in \text{Class } C_j, \text{ se } d(X, C_j) = \min_i d(X, C_i) \quad (3.14)$$

È l'approccio più diretto e intuitivo alla classificazione.

In alcune applicazioni di classificazione una classe può essere rappresentata da campioni multipli che non appartengono allo stesso gruppo come in Figura 3.16. Per capire meglio il tipo di campo d'applicazione basti pensare a un classificatore di due classi che utilizza due feature. I campioni della prima classe presentano valori della feature 1 che oscillano attorno a un valore a_1 , mentre per quanto riguarda la feature 2 alcuni oscillano attorno a b_1 e altri attorno a c_1 . La seconda classe invece ha tutti i campioni che oscillano attorno al valore della feature 2 a_2 , mentre per quanto riguarda la feature 1 alcuni campioni oscillano attorno a b_2 e altri attorno a c_2 . In questo tipo di applicazioni il NN è molto efficace.

K-Nearest Neighbor Classifier (KNN)

Nel KNN un campione sconosciuto in ingresso X è associato a una classe C_j in base a un meccanismo di voto attraverso la selezione dei K campioni più vicini, indipendentemente dalla classe. Il campione sconosciuto verrà assegnato alla classe con il maggior numero di voti ottenuti dai K campioni più vicini.

In Figura 3.17 si può notare come il campione esterno alla classe 1, causato dai disturbi nelle applicazioni reali, può causare classificazioni errate quando si utilizza il NN classifier. Come mostrato in Figura 3.17, la classificazione KNN è molto più robusta ai disturbi rispetto alla NN. Ad esempio il campione X , nel caso si considerino i tre campioni più vicini ($K = 3$), sarà associato alla classe 2, in quanto due dei tre campioni appartengono a quest'ultima.

Minimum Mean Distance (MMD)

Siano $\{X_1^j, X_2^j, \dots, X_{n_j}^j\}$ gli n_j feature vector che rappresentano la classe C_j . Il centro della classe j è definito come:

$$M_j = \frac{1}{n_j} \sum_{i=1}^{n_j} X_i^j \quad (3.15)$$

Al campione in ingresso X viene associata una classe in funzione della sua distanza rispetto al centro delle classi:

$$X \in \text{Classe } C_j, \text{ se } d(X, M_j) = \min_i d(X, M_i) \quad (3.16)$$

In applicazioni dove la variabilità delle feature è piccola e con pochi disturbi, i campioni tenderanno ad avvicinarsi al centro della classe; in queste condizioni, il MMD funziona efficacemente (Figura 3.18). Si evidenzia che vengono calcolate solo le distanze del feature vector in ingresso rispetto ai centri delle classi e non rispetto a ciascun campione; ciò comporta una diminuzione considerevole della quantità di calcolo se il classificatore ha molti campioni di training.

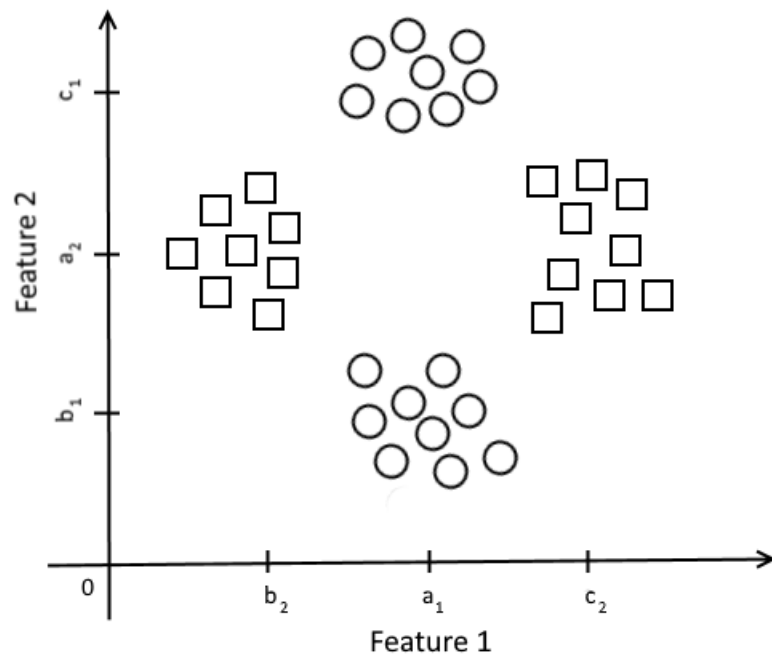


Figura 3.16: Esempio di distribuzione dei campioni nel piano delle feature in cui il NN classifier funziona bene. \circ = Classe 1 e \square = Classe 2.

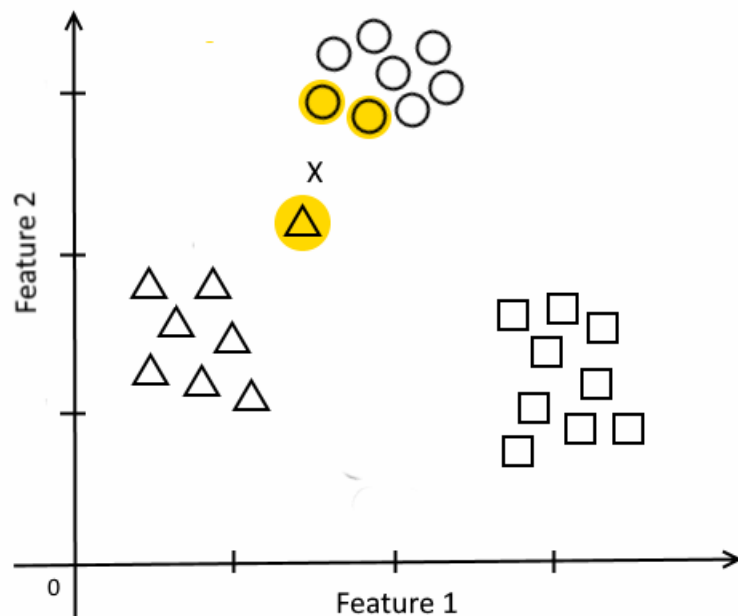


Figura 3.17: Esempio di distribuzione dei campioni nel piano delle feature in cui il KNN classifier funziona bene. \triangle = Classe 1, \circ = Classe 2 e \square = Classe 3.

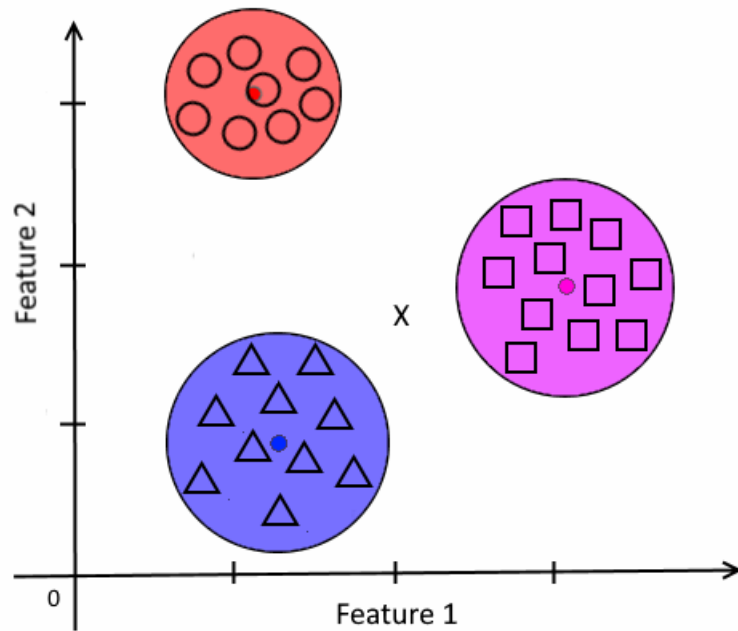


Figura 3.18: Esempio di distribuzione dei campioni nel piano delle feature in cui il MMD classifier funziona bene. \circ = Classe 1, \square = Classe 2 e \triangle = Classe 3.

3.3 Valutazione del classificatore

Per valutare il classificatore possono essere analizzati i dati delle feature raccolte nella fase di training oppure valutare i risultati che si ottengono dal classificatore con una serie di campioni di test.

3.3.1 Valutazione dei dati delle feature raccolte

La funzione di classificazione fornisce in uscita informazioni riguardanti le deviazioni standard delle feature dei dati appartenenti ad una classe, e una tabella che indica la distanza che c'è tra tutte le diverse classi. Un valente insieme di dati, raccolti durante la fase di training, dovrebbe avere come caratteristiche:

- una piccola deviazione per i dati che appartengono alla stessa classe;
- una grande differenza tra i campioni delle diverse classi.

Array delle deviazioni intraclassa (Intraclass Deviation Array)

Si definisca l'insieme delle coppie $[Q_j, N_j]$, $j = 1, 2, \dots, L$, dove N_j è il numero di campioni della classe j e L è il numero di classi. Si considerino $\{X_1^j, X_2^j, \dots, X_{N_j}^j\}$ che sono gli N_j feature vector, uno per ciascun campione, di dimensione n (numero delle feature) che rappresentano la

classe C_j . Ciascun vettore è definito come $X_i^j = [x_{i1}^j, x_{i2}^j, \dots, x_{in}^j]^T$ $i = 1, \dots, N_j$. Si definisca $M^j = [m_1^j, m_2^j, \dots, m_n^j]$ come il vettore delle medie della classe j per ciascuna feature. Quindi:

$$M^j = \frac{1}{N_j} \sum_{i=1}^{N_j} X_i^j \quad (3.17)$$

dove ogni elemento del vettore delle medie è:

$$m_k^j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_{ik}^j \quad (3.18)$$

La deviazione standard della k -esima feature della classe j è:

$$\sigma_k^j = \frac{1}{N_j} \sum_{i=1}^{N_j} (x_{ik}^j - m_k^j)^2 \quad (3.19)$$

da cui deriva:

$$Q_j = \max_k \sigma_k^j \quad (3.20)$$

L'NI Particle Classifier fornisce solo Q_j , tale valore se non contestualizzato con almeno il valore medio, a cui è collegata tale deviazione standard, non ha molto senso ai fini di valutazione del set di dati. Mentre, nel Custom Particle Classifier è possibile visionare le deviazioni standard di ciascuna feature per ciascuna classe. In altre parole, in output è possibile controllare m_k^j per $k = 1, \dots, n$ e $j = 1, \dots, L$.

Tabella della distanza interclasse (Class Distance Table)

Si consideri $M^j = [m_1^j, m_2^j, \dots, m_n^j]$ e $M^i = [m_1^i, m_2^i, \dots, m_n^i]$ i vettori delle medie delle classi j e i definiti in precedenza. La distanza tra due classi i e j è definita come:

$$d_{ij} = D(M^i, M^j) \quad (3.21)$$

dove D è il Distance Metric selezionato nella fase di training. La tabella viene compilata con le distanze tra tutte le classi (Tabella [3.1](#)) e permette di effettuare una valutazione generale dei dati associati alle classi, ad esempio si possono individuare le due classi più vicine e le due classi più distanti. Inoltre si può usare la class distance table con la intraclass deviation array per valutare i differenti set di dati dei training. Si sottolinea che la tabella presenta una diagonale di zeri, dato che la distanza di una classe rispetto se stessa è zero, ed è simmetrica, dato che la distanza della classe i dalla classe j è uguale alla distanza della classe j dalla classe i .

Classe	1	2	3	...	L
1	0	$d_{1,2}$	$d_{1,3}$...	$d_{1,L}$
2	$d_{2,1}$	0	$d_{2,3}$...	$d_{2,L}$
3	$d_{3,1}$	$d_{3,2}$	0	...	$d_{3,L}$
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
L	$d_{L,1}$	$d_{L,2}$	$d_{L,3}$...	0

Tabella 3.1: Illustrazione della Class Distance Table di un generico classificatore con L classi.

3.3.2 Determinazione della qualità di un classificatore

Per determinare la qualità di un classificatore, come già detto, si effettua una fase di test classificando una serie di campioni di cui si conosce già la classe di appartenenza. I risultati della fase di test vengono riportati in una tabella chiamata **Classification distribution table**, dove nella prima colonna viene indicata la classe di appartenenza dei campioni di test, mentre nella prima riga viene indicata la classe di assegnazione. Nella Classification Distribution table i campioni correttamente classificati sono contenuti lungo la diagonale. Come si può vedere nella Tabella 3.2 è riportato un esempio di una Distribution Table in cui al classificatore, costituito dalle classi A, B o C, è stato chiesto di classificare 15 campioni di test per ciascuna classe.

Dopo la fase di test e di compilazione della tabella è possibile calcolare due coefficienti probabilistici che permettono la valutazione del classificatore (vedi Tabella 3.2):

- **Classifier Accuracy:** indica la probabilità che un campione sia classificato nella classe in cui appartiene. Si calcola come:

$$\frac{\text{Numero di campioni classificati correttamente}}{\text{Numero totale di campioni di cui si conosce la classe}} \quad (3.22)$$

Facendo riferimento all'esempio illustrato in precedenza, sono presenti 15 campioni della classe B e il classificatore li classifica come: 2 campioni classe A, 4 classe C e 9 campione classe B; allora l'Accuracy di B è $\frac{9}{15} = 60\%$. In Tabella 3.2 questo indice è riportato nell'ultima colonna. Si calcola anche un ultimo fattore che indica l'accuratezza totale data da:

$$\frac{\text{Numero totale di campioni classificati correttamente}}{\text{Numero totale di campioni totale applicati}} \quad (3.23)$$

- **Classifier Predictability:** indica la probabilità che un campione classificato in una data classe appartenga a tale classe:

$$\frac{\text{Numero di campioni classificati correttamente}}{\text{Numero totale di campioni classificati nella classe}} \quad (3.24)$$

Facendo riferimento all'esempio illustrato in precedenza, 13 campioni sono stati classificati

nella classe B e di tali campioni 9 appartengono alla classe B e 4 alla classe C; allora la Predictability di B è $\frac{9}{13} = 69.2\%$. In Tabella [3.2](#) questo indice è riportato nell'ultima riga.

	A	B	C	Totale	Accuracy
Campioni della classe A	15	0	0	15	$\frac{15}{15} = 100\%$
Campioni della classe B	2	9	4	15	$\frac{9}{15} = 60\%$
Campioni della classe C	1	4	10	15	$\frac{10}{15} = 66.7\%$
Totale	18	13	14	45	$\frac{24}{45} = 75.6\%$
Predictive Value	$\frac{15}{18} = 83.3\%$	$\frac{9}{13} = 69.2\%$	$\frac{10}{14} = 71.4\%$		

Tabella 3.2: Esempio di una Classification Distribution Table.

3.3.3 Valutazione della classificazione

La funzione di classificazione fornisce in output due indici: **Identification confidence** e **Classification confidence**, i quali servono per la valutazione dei risultati di classificazione. Il Classification confidence è un indice molto significativo sia per le operazioni di selezione che ispezione. Si utilizza l'Identification confidence solo quando non è possibile effettuare decisioni con il solo Classification confidence, dopo la definizione dei due indici verrà illustrata la situazione in cui l'Identification confidence è utile al fine di valutare la classificazione.

Classification Confidence Score

Indica il grado con il quale la classe assegnata rappresenta al meglio il campione identificato rispetto alla rappresentazione che le altre classi forniscono. Viene definito come:

$$\text{Classification confidence score} = \left(1 - \frac{d_1}{d_2}\right) * 1000 \quad (3.25)$$

dove d_1 è la distanza del campione rispetto alla classe più vicina e d_2 rispetto alla seconda più vicina. I due valori vengono calcolati e scelti in base al distance metric e classification method scelti nella fase di training.

Permette inoltre di filtrare gli errori di classificazione mettendo una soglia di score minima tale per cui una classificazione può essere accettata o meno. La soglia di accettabilità è individuata attraverso l'analisi della **Classification Confidence Distribution** (Capitolo [3.3.3](#)).

Identification Confidence Score

Indica la somiglianza tra il campione identificato e la classe assegnata:

$$\text{Identification confidence score} = (1 - d) \cdot 1000 \quad (3.26)$$

Dove d è la distanza normalizzata tra il campione in ingresso e la classe assegnata:

$$d = \frac{\text{Distanza tra campione di input e la classe assegnata}}{\text{fattore di normalizzazione}} \quad (3.27)$$

dove il fattore di normalizzazione è la massima distanza interclasse.

L'Identification score è utile nel caso in cui il classificatore sia stato istruito con solo una classe, ciò comporta che il Classification score è sempre pari a mille dato non esiste una seconda classe con cui calcolarlo, mentre l'Identification score viene calcolato considerando come fattore di normalizzazione la distanza tra i due campioni più lontani della classe rendendo l'indice utile per la valutazione della classificazione di campioni sconosciuti.

Esempio

Si assuma che sia stato generato un classificatore che utilizza una sola feature, che può assumere valori da 0 a 1, per classificare due classi A e B. In Figura 3.19 è riportata la distribuzione rispetto alla feature dei campioni di training delle due classi. Le due distribuzioni sono centrate in 0.22 per A e 0.83 per B. Di seguito viene illustrato il calcolo dei due score appena illustrati per un campione da classificare la cui feature ha un valore di 0.41; ipotizzando che il campione sia associato in base al MMD Classification Method si associa il campione alla classe A:

$$\text{Classification confidence} = \left(1 - \frac{|0.41 - 0.22|}{|0.41 - 0.83|}\right) \times 1000 = 547.62 \quad (3.28)$$

$$\text{Identification confidence} = \left(1 - \frac{|0.41 - 0.22|}{|0.22 - 0.83|}\right) \times 1000 = 688.52 \quad (3.29)$$

Nel caso ideale tali valori dovrebbero assumere il valore più alto possibile.

Classification Confidence Distribution e soglia minima di sbarramento

Si effettui la classificazione di un insieme di campioni di test con un classificatore e si memorizzino tutti i Classification Confidence Score ottenuti per ciascun campione. Con il set di dati raccolto è possibile generare il Classification Confidence Distribution la cui ampiezza rappresenta il numero di campioni di test classificati con uno specifico classification score. Questa distribuzione è un buon indicatore delle performance di classificazione.

Nelle Figura 3.20 viene illustrata la Classification Confidence Distribution di un set di campioni

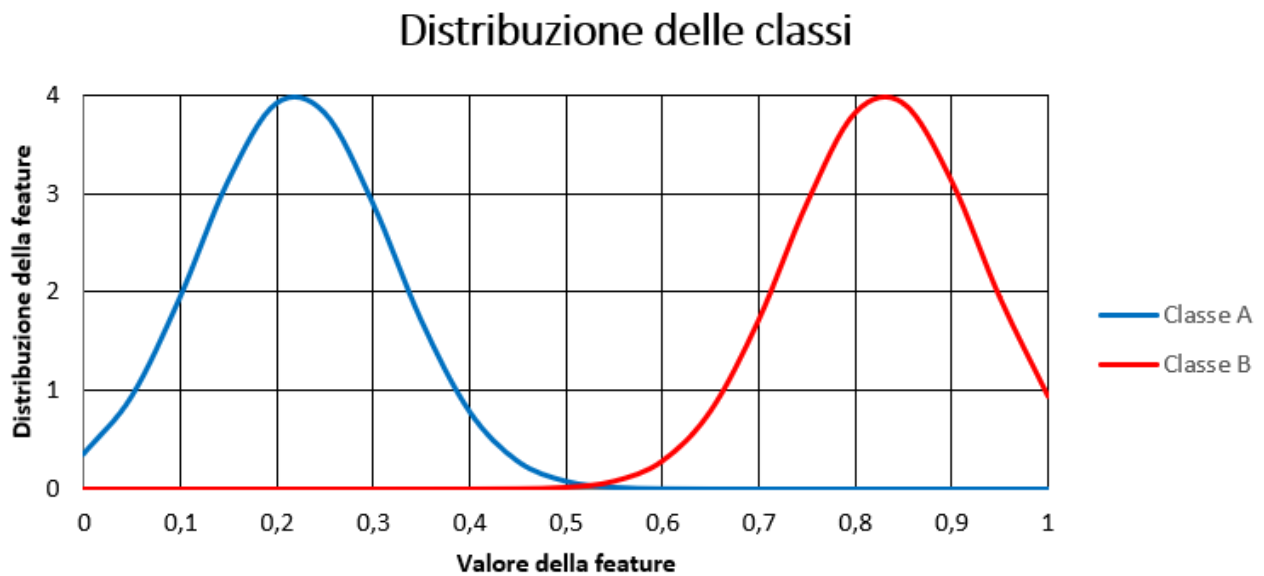


Figura 3.19: Esempio di distribuzione di probabilità che un certo numero di campioni, della classe A o B, assumano un certo valore della caratteristica.

di test di componenti meccanici. Al fine di ottenere il maggior numero di campioni classificati correttamente si rende necessario fissare ad 800 il valore della soglia di sbarramento.

La Figura [3.21](#) illustra la classification confidence distribution per un set di campioni di test di componenti elettronici. Se si utilizza la stessa soglia di classification score definita in precedenza per i componenti meccanici, si ottiene un alto tasso di campioni classificati non in modo corretto, ma questo risultato è dato dal fatto che una buona parte dei campioni ha un indice di classificazione inferiore a 800, perciò si rende necessario abbassare la soglia minima del classification score.

Il Classification Confidence Distribution di un set di campioni di test è un buon indicatore per la scelta di un buon indice di soglia per una specifica applicazione di ispezione e selezione.

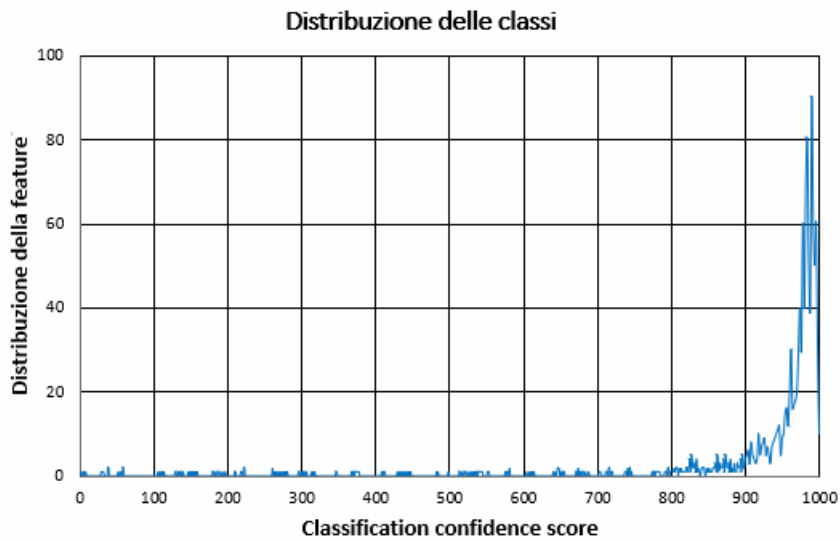


Figura 3.20: Distribuzione della classification confidence: probabilità con cui un certo numero di campioni gli venga assegnato un certo classification score. Questa distribuzione è stata generata con un set di campioni di test, differenti dai campioni di training, di componenti meccanici.

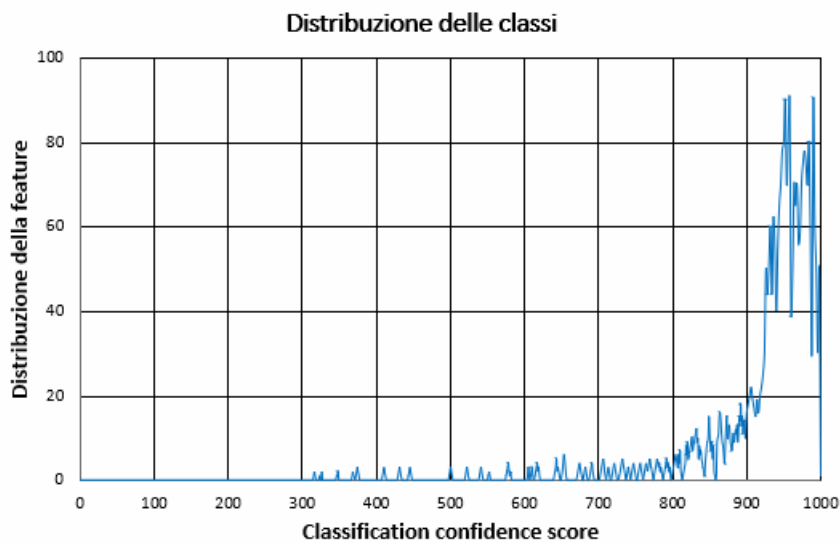


Figura 3.21: Distribuzione della classification confidence: probabilità con cui un certo numero di campioni gli venga assegnato un certo classification score. Questa distribuzione è stata generata con un set di campioni di test, differenti dai campioni di training, di componenti elettronici.

3.4 Requisiti fondamentali per la classificazione degli oggetti

3.4.1 Gli illuminatori

La scelta di un sistema di illuminazione è fondamentale per qualsiasi applicazione di machine vision. [29] La scelta del tipo di illuminazione permette di:

- ottenere contrasti maggiori dei dettagli a cui si è interessati;
- diminuire il contrasto di tutto ciò che non è rilevante;
- attenuare i disturbi provenienti dall'ambiente.

Le componenti fondamentali da controllare per ottenere condizioni di illuminazione adatte sono:

- La geometria della sorgente luminosa. Ciò significa sfruttare al meglio la geometria di riflessione tra la sorgente luminosa, l'oggetto e la telecamera. Piccoli cambiamenti delle posizioni relative tra questi potrebbero portare a risultati più o meno accettabili.
- Le proprietà direzionali della luce. Sono coinvolti tre effetti: luce diffusa o diretta, localizzazione dell'illuminatore e la direzione dei raggi luminosi.
- La lunghezza d'onda (colore) della sorgente. La componente spettrale della luce può essere sfruttata per enfatizzare uno specifico colore e attenuare i restanti.
- Il filtraggio. L'utilizzo di un filtro infrarossi o di un filtro polarizzante può aiutare nella rilevazione di parti di interesse o bloccare disturbi luminosi.

Prima di entrare nello specifico di alcuni tipi di illuminatori si spiegano le principali categorie in cui questi si suddividono: nel caso in cui la sorgente luminosa sia dallo stesso lato della telecamera rispetto all'oggetto si definisce *Front Light*, viceversa se si trova dal lato opposto è detta *Back Light*; se la sorgente è posizionata con un'angolazione rispetto all'oggetto tale che la maggior parte della radiazione luminosa è riflessa verso la telecamera l'illuminatore si definisce di tipo *Bright-Field*, mentre nel caso contrario in cui solo una piccola parte di radiazione viene riflessa verso la telecamera si definisce *Dark-Field*.

Tenendo in considerazione gli aspetti elencati in precedenza si illustrano 3 tipi di illuminatori: **Dark-field front light** (Figura 3.22(a)), **Bright-field front light** (Figura 3.22(b)) e **Bright-field back light** (Figura 3.22(c)).

L'illuminatore **Dark-field front light** è ottenuto direzionando la radiazione luminosa mantenendo un piccolo angolo di inclinazione rispetto alla superficie dell'oggetto; per ottenere tali caratteristiche l'anello di illuminazione è posto vicino all'area di lavoro. Questo illuminatore

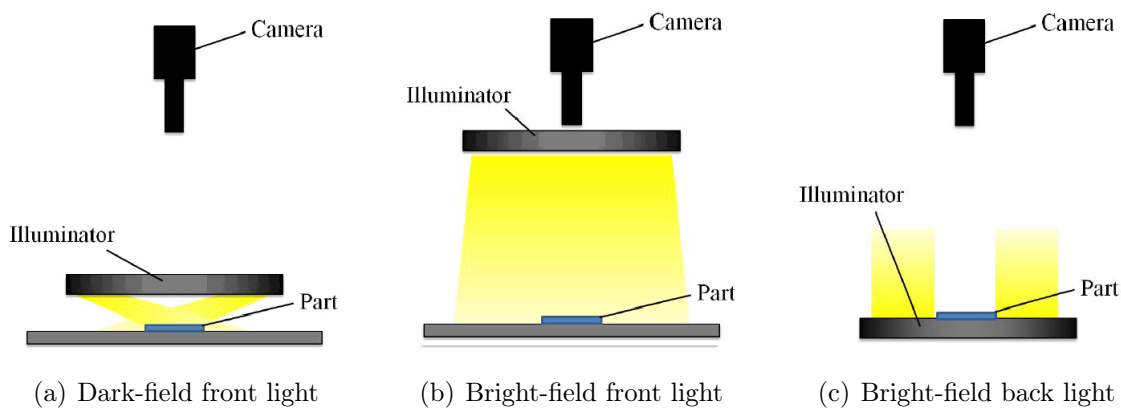


Figura 3.22: Tipi di illuminatori [18].

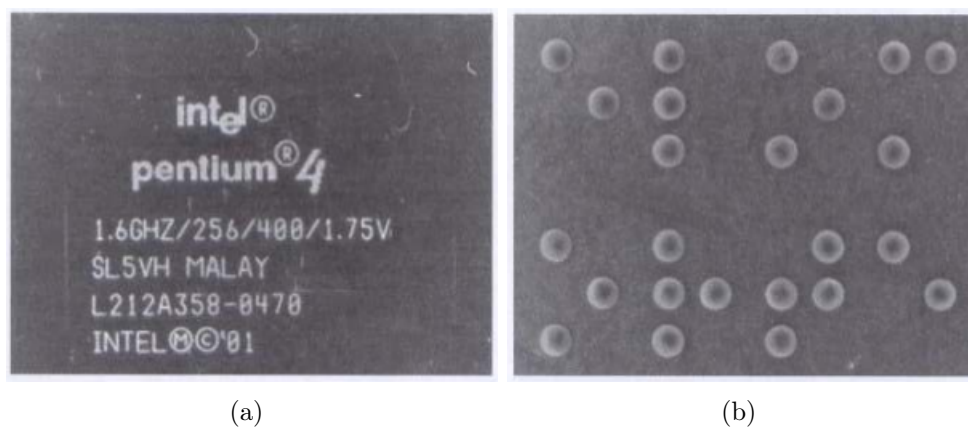


Figura 3.23: Acquisizioni di immagini con un illuminatore Dark-field. (a) Il numero di serie inciso su una CPU, notare come i graffi sulla CPU vengono risaltati; (b) la stampa di caratteri Braille su una scatola di farmaci messi in evidenza.

permette di mettere in risalto rientranze o protrusioni di un oggetto, si può quindi utilizzare per aumentare la visibilità di caratteristiche come graffi, incisioni o texture (Figura 3.23).

L'illuminazione **Bright-field front light** può essere ottenuta utilizzando due tipi di dispositivi: pannelli o anelli LED montati assialmente alla telecamera, oppure illuminatori coassiali che usano uno specchio semi-trasparente, attraverso il quale la telecamera può anche acquisire l'immagine. Nella prima tipologia l'illuminazione non sarà mai uniforme dato che non si hanno sorgenti di luce nel punto in cui la telecamera inquadra, esistono però diffusori studiati per risolvere questa problematica; nella seconda tipologia si ha un'illuminazione più regolare in quanto la telecamera inquadra attraverso lo specchio causando però distorsioni dell'immagine. Questi tipi di illuminatori sono tipicamente usati per prevenire le ombre e per ridurre e prevenire riflessioni speculari. Possono essere usati per la visione di oggetti contenuti in contenitori (Figura 3.24).

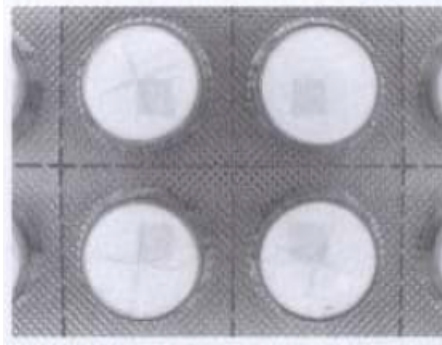


Figura 3.24: Esempio di un blister illuminato con un sistema Bright-field front light. Si noti che le pastiglie all'interno della copertura sono visibili.

Il **Bright-field back light** è ottenuto posizionando la sorgente luminosa dietro all'oggetto da inquadrare. Questo illuminatore mette in mostra solo il profilo di oggetti opachi, può essere utilizzato nel caso in cui le informazioni necessarie derivino dal profilo dell'oggetto. La Bright-field back light permette la visione dei componenti contenuti in oggetti trasparenti dato che evita la riflessione sulla superficie dell'oggetto.

Nel caso in cui venga illuminato un oggetto con un alto spessore alcune parti riflettono la luce verso la telecamera rendendo la visione dell'oggetto meno nitida (Figura 3.25): la causa è la sorgente luminosa di tipo diffuso; perciò si predilige illuminare con questa modalità solo oggetti che presentano uno spessore ridotto.

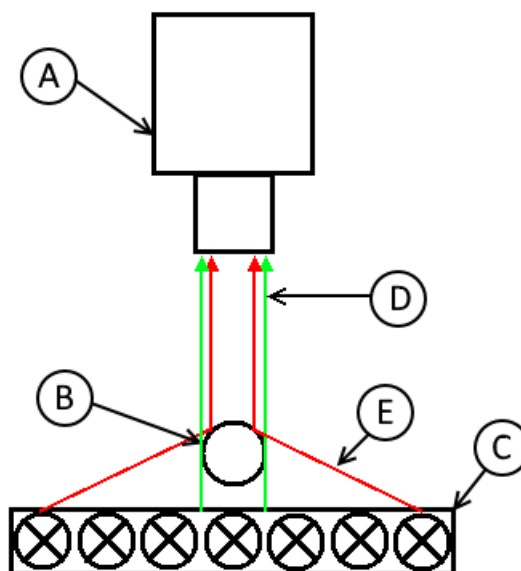


Figura 3.25: Bright-field Back Light: caso di riflessione che può accadere per oggetti con grandi spessori. (A) Telecamera; (B) oggetto; (C) Sorgente luminosa Bright-field back light; (D) raggio luminoso corretto che risalta i contorni dell'oggetto; (E) raggio luminoso riflesso dal bordo dell'oggetto.

Utilizzando questo illuminatore si possono estrarre le seguenti proprietà di un oggetto (Figura 3.26):

- la presenza o meno di fori, intervalli o parti;
- la posizione o l'orientamento di un componente;
- la dimensione di una certa parte.

Tutti i tipi di illuminatori elencati forniscono una luce diffusa e uniforme rendendo il sistema robusto rispetto alle condizioni di illuminazione ambientali.

Per l'utilizzo del metodo della classificazione il tipo di illuminazione migliore è il Bright-field back light poiché permette di estrarre più facilmente le caratteristiche geometriche del componente (Figura 3.27). Ciò non toglie che sia possibile utilizzare la Bright-field front light se permette di evidenziare i bordi dell'oggetto.

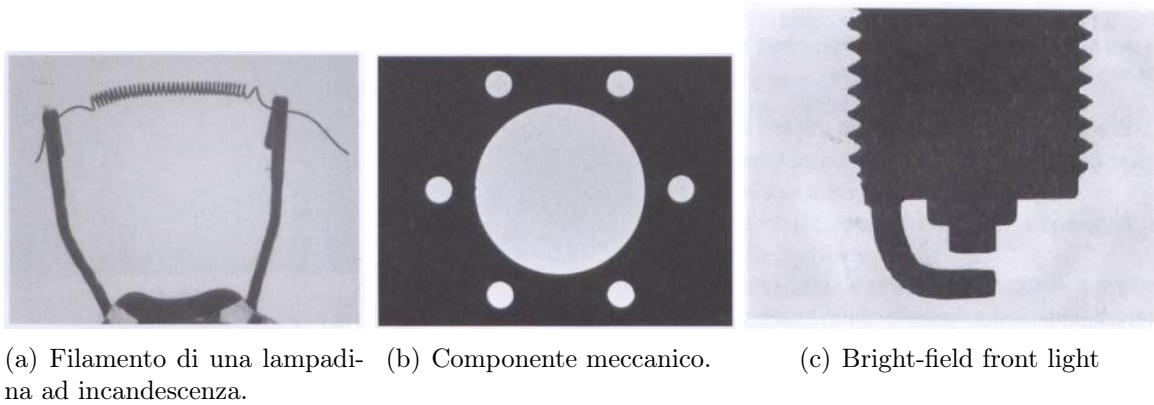


Figura 3.26: Esempi di oggetti illuminati con un illuminatore Bright-field Back Light.

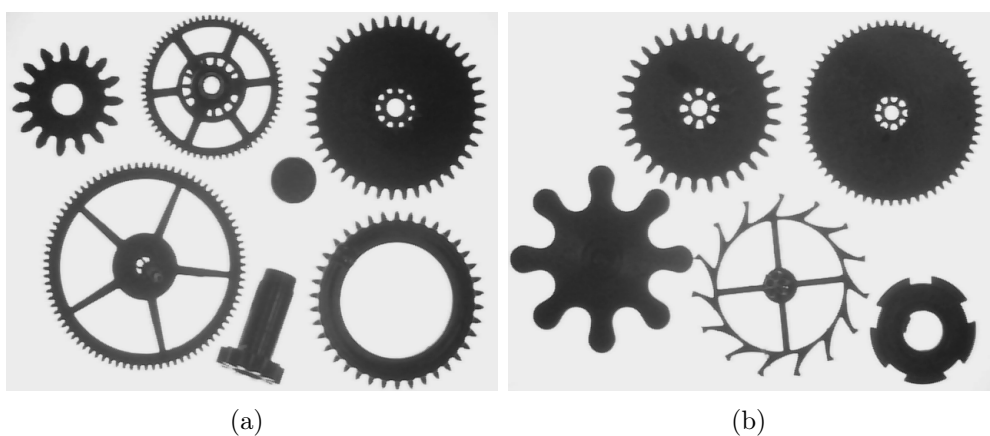


Figura 3.27: Esempi di oggetti da classificare illuminati con un Bright-field Back light.

3.4.2 Tipo di oggetto

L'oggetto contenuto nell'immagine dopo l'operazione di sogliatura e di image processing dev'essere una particella unica per poter effettuare la Particle Analysis. Questo significa che gli oggetti che si possono classificare devono essere costituiti da un'unica particella (Figura 3.28).

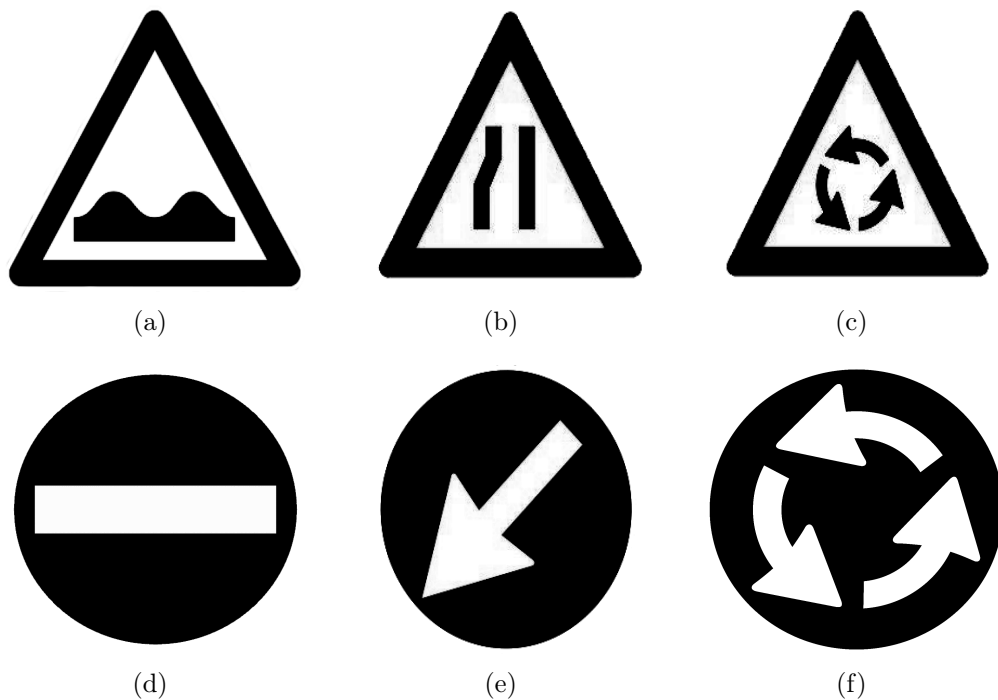


Figura 3.28: (a-c) Esempi di oggetti non classificabili a causa della presenza di particelle interne. (d-f) Esempi di oggetti classificabili perché formati da una particella unica.

I due metodi di classificazione studiati sono, come già detto, basati sull'analisi delle caratteristiche geometriche e morfologiche degli oggetti da identificare. Si prediligono quindi oggetti che presentano un colore uniforme e caratteristiche di forma ben definite e misurabili mediante le feature. Gli oggetti che presentano come fattori di identificazione le sole caratteristiche interne legate alla scala di grigi non sono classificabili (Figura 3.29), a meno che tali caratteristiche siano secondarie e non influenzino il risultato della classificazione (Figura 3.30).

Come già detto nel Sottoparagrafo 3.2.1, è sempre necessario prestare attenzione alle caratteristiche delle classi che si vogliono identificare ma soprattutto è indispensabile verificare che le classi posseggano caratteristiche geometriche e morfologiche rilevabili dalle feature; in caso contrario, il metodo non risulta applicabile.

Infine, la classificazione non è applicabile nel caso in cui gli oggetti siano attaccati o sovrapposti poiché in questo caso non si otterrebbero particelle separate ma un'unica particella sulla quale si effettuerà l'estrazione delle feature che a loro volta non permetteranno l'identificazione dell'oggetto.



Figura 3.29: Esempio in cui non è possibile generare un classificatore in quanto le monete si distinguono per le caratteristiche interne legate alla scala di grigi.



Figura 3.30: Esempio in cui è possibile generare un classificatore, nonostante le caratteristiche interne, perché ciascuna moneta ha una dimensione diversa.

Capitolo 4

Sviluppo del Software di classificazione

Il software sviluppato in questa tesi permette l'applicazione di metodi di classificazione di componenti (Capitolo [3](#)) per creare applicazioni di classificazione e ispezione per micro-manipolazione automatizzata tramite sistemi di visione 2D.

Il software è stato sviluppato in tre moduli, ciascuno dei quali svolge una delle fasi principali del metodo:

1. generazione del file di classificazione (training del classifier);
2. test del classificatore;
3. applicazione della classificazione.

Ciascun modulo è stato sviluppato in modo da poter essere eseguito indipendentemente dagli altri, rendendo il software più flessibile e facilmente inseribile nel programma di gestione della cella di micro-manipolazione.

Come detto nel Capitolo [3.2.1](#), sono stati studiati due tipi di classificatori: Particle Classifier (NI) e Custom Particle Classifier (PFV). Questi differiscono principalmente nella possibilità di:

- scegliere o meno le feature da estrarre;
- applicare o meno operazioni aggiuntive di image processing.

Tali differenze rendono lo sviluppo dei due tipi di classificatori a livello software e di gestione dei dati molto differenti, quindi si sono sviluppati due programmi, uno per metodo. Ogni software contiene un modulo di training del metodo e uno di test, e un modulo di classificazione uguale per entrambi i metodi. La scelta di creare un modulo di classificazione comune faciliterà l'implementazione della classificazione nel programma di gestione. Dalla pagina principale dei due software (NI e PFV) è possibile accedere alle tre fasi della classificazione (Figura [4.1](#)).

Si torna a precisare che il programma è stato sviluppato nell'ambiente LabVIEW, il quale fornisce una serie di librerie che permettono di implementare entrambi i tipi di classificatori. Il

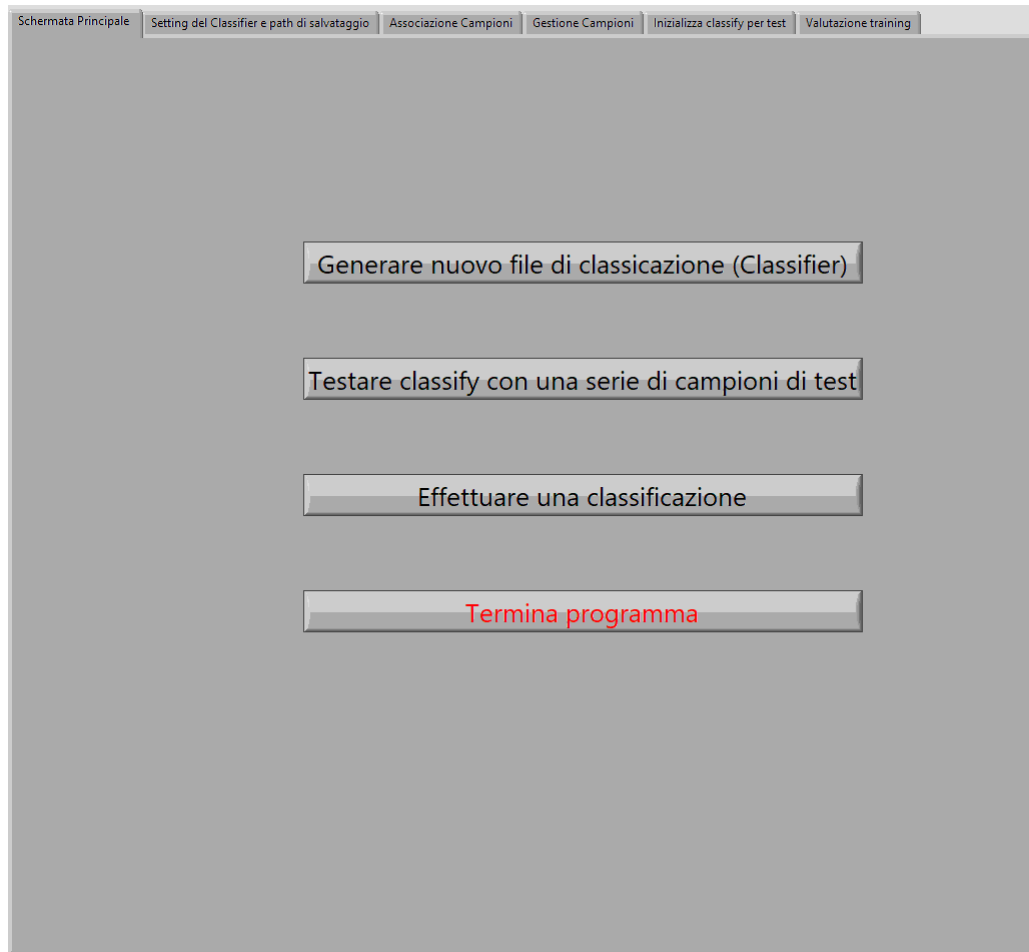


Figura 4.1: Front panel dei due software sviluppati.

programma si basa su una macchina a stati e le HMI guidano l'utente nelle varie operazioni. Le funzioni utilizzate per applicare la classificazione di tipo NI richiedono in ingresso, sia per il training che per la classificazione, l'immagine contenente il campione in scala di grigi senza l'applicazione di operazioni di image processing, dato che queste verranno effettuate internamente alla funzione stessa. Per quanto riguarda le funzioni dedicate al classificatore PFV non è richiesta l'immagine del campione, bensì il vettore delle caratteristiche di quest'ultimo. Per questa ragione, nello sviluppo del software, è stata creata una funzione che, ricevendo in ingresso un'immagine in scala di grigi, fornisce in uscita il vettore delle caratteristiche del campione; tutto ciò è effettuato applicando una serie di elaborazioni dell'immagine (come threshold, erosione, rimozioni particelle che toccano i bordi e apertura o chiusura) e l'applicazione della Particle Analysis (Capitolo [2.3](#)) che estrae tutte le caratteristiche geometriche richieste. Infine, tutte le immagini utilizzate nella fase di training, di test e di classificazione devono essere acquisite dallo stesso sistema di visione, al fine di ottenere una maggiore affidabilità con le stesse condizioni ambientali in modo tale da mantenere un'uniformità delle caratteristiche delle immagini e quindi un'uniformità nei parametri del classificatore.

4.1 Training nel caso del software Particle Classifier NI

Di seguito verranno illustrate tutte le parti che riguardano la generazione del file di classificazione (Modulo 1) del Particle Classifier NI.

4.1.1 Selezione dei parametri di training

La scelta dei parametri di training gioca un ruolo fondamentale per adattare al meglio il classificatore al tipo di applicazione in cui verrà utilizzato (Figura 4.2). Tutti i parametri, una volta selezionati, saranno applicati in tutte le fasi del metodo della classificazione. La prima scelta è relativa ai parametri per l'elaborazione dell'immagine con cui si effettua la binarizzazione e un eventuale miglioramento dell'immagine. Le elaborazioni applicate sono le seguenti:

- threshold (Capitolo 2.1): effettua la binarizzazione dell'immagine ed è un'operazione fondamentale per l'estrazione delle caratteristiche; può essere automatico o manuale e

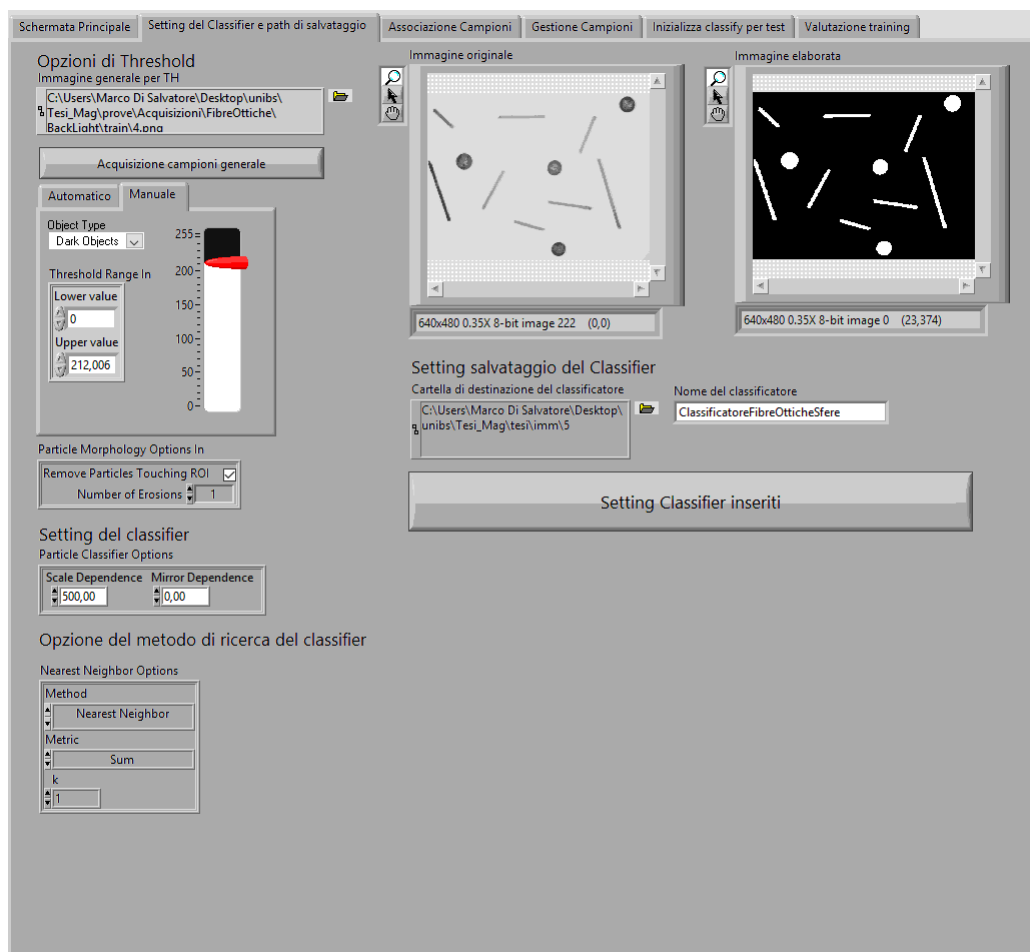


Figura 4.2: Pagina del software NI dedicata alla selezione dei parametri del classificatore.

richiede di definire il tipo di oggetto da mettere in evidenza: chiaro (Bright Objects), scuro (Dark Objects) o grigio (Gray Object);

- Remove Particles Touching ROI: effettua la cancellazione delle particelle che toccano il bordo dell'immagine o la ROI (Region Of Interest) utilizzata per la selezione di un campione (è l'equivalente della Border function (Capitolo [2.2.5](#)));
- Erosion (Capitolo [2.2.1](#)): applica l'operazione morfologica di erosione per eliminare le particelle piccole che rappresentano rumore, ed è possibile scegliere il numero di iterazioni con cui applicarla. Il numero di iterazioni indica quante volte si deve applicare l'operazione di erosione, perché in alcuni casi per eliminare una piccola particella è necessario applicare l'erosione più volte.

La scelta attenta di questi parametri permette di risaltare al meglio le caratteristiche geometriche degli oggetti contenuti nell'immagine e permette di eliminare tutte quelle parti che creano disturbo (come piccole particelle o parti tagliate dal bordo dell'immagine). Per verificare se i valori scelti siano adatti al tipo di immagine da cui si estrarranno i campioni si utilizza un'immagine di riferimento ("Immagine originale") su cui sono applicate le operazioni di image processing ("Immagine elaborata"). Nel caso in cui non si abbia a disposizione nessuna immagine è possibile acquisirne una utilizzando l'apposita funzione "Acquisisci campione generale" (Figura [4.3](#)); questa permette di acquisire nuove immagini da una qualsiasi sistema di visione 2D collegato al PC. La funzione di acquisizione permette inoltre di applicare le stesse operazioni di image processing descritte in precedenza, consentendo di verificare la conformità all'estrazione delle caratteristiche geometriche del campione; i parametri utilizzati sono trasferiti nei setting del classificatore alla fine dell'acquisizione.

La fase successiva è la selezione di tutti i parametri che riguardano il classificatore. Si parte dalla selezione del peso da assegnare alle caratteristiche di scala (*Scale dependence*) e a quelle di simmetria speculare (*Mirror dependence*), ciascuno dei quali può assumere un valore tra 0 e 1000. Se il peso assegnato è pari a zero la caratteristica non viene considerata; i due pesi sommati devono avere un valore inferiore a 1000 e per capire il peso con cui sono considerate le caratteristiche di forma si effettua il calcolo $Shape\ Factor = 1000 - (Scale\ Factor + Mirror\ Factor)$ (Capitolo [3.2.1](#)). Un'ulteriore coppia di parametri necessari per il classificatore sono il Method, con cui eleggere la classe di un campione sconosciuto (Capitolo [3.2.3](#)), e il Distance Metric, con cui calcolare la distanza tra il campione sconosciuto e i campioni di training (Capitolo [3.2.2](#)); nel caso si scelga il k-Nearest Neighbour è necessario definire con il parametro k quanti campioni di training considerare nella fase di assegnazione.

Infine si deve indicare il nome del file di classificazione ed il percorso della cartella in cui salvarlo. Al momento del salvataggio il programma aggiunge al nome la desinenza "NIInterf" permettendo di identificare più facilmente il classificatore di tipo NI.

Tutti i setting definiti fino a questo punto possono essere modificati e sovrascritti, per essere utilizzati in fase di classificazione (e training), in qualsiasi momento durante l'associazione dei campioni, che verrà illustrata in seguito.

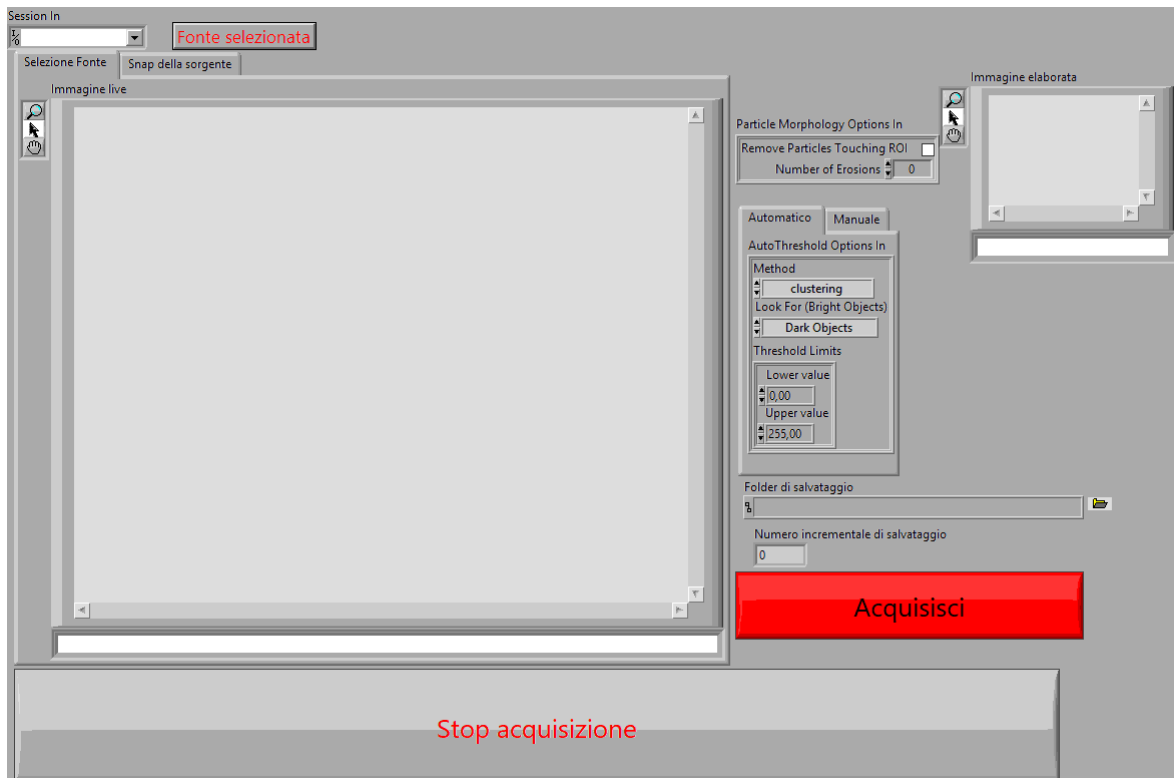


Figura 4.3: Pagina della funzione, dedicata al metodo NI, che permette l'acquisizione di nuovi campioni da un sistema di visione collegato al PC.

4.1.2 Generazione delle classi ed associazione dei campioni

La fase successiva consiste nella creazione e nel popolamento delle classi ed è svolta nella pagina del programma NI mostrata in Figura [4.4](#).

Per prima cosa si seleziona la cartella contenente le immagini di training e ciascuna immagine può contenere più di un campione. Nel caso in cui non si abbiano immagini dei campioni, queste possono essere acquisite come nella fase precedente; la funzione di acquisizione lavora allo stesso modo, con l'unica differenza che le operazioni di image processing applicate utilizzano i valori selezionati nella pagina di setting del classificatore e non sono modificabili, comunque è possibile tornare indietro e modificarli.

Nella finestra "Immagine contenente i campioni" si possono visualizzare le immagini contenute nella cartella; incrementando o decrementando il contatore "Selezione immagine" è possibile cambiare l'immagine visualizzata. Prima di poter effettuare l'associazione dei campioni bisogna creare le classi con il comando "Aggiungi classe", tramite il quale si accede a una sezione dove

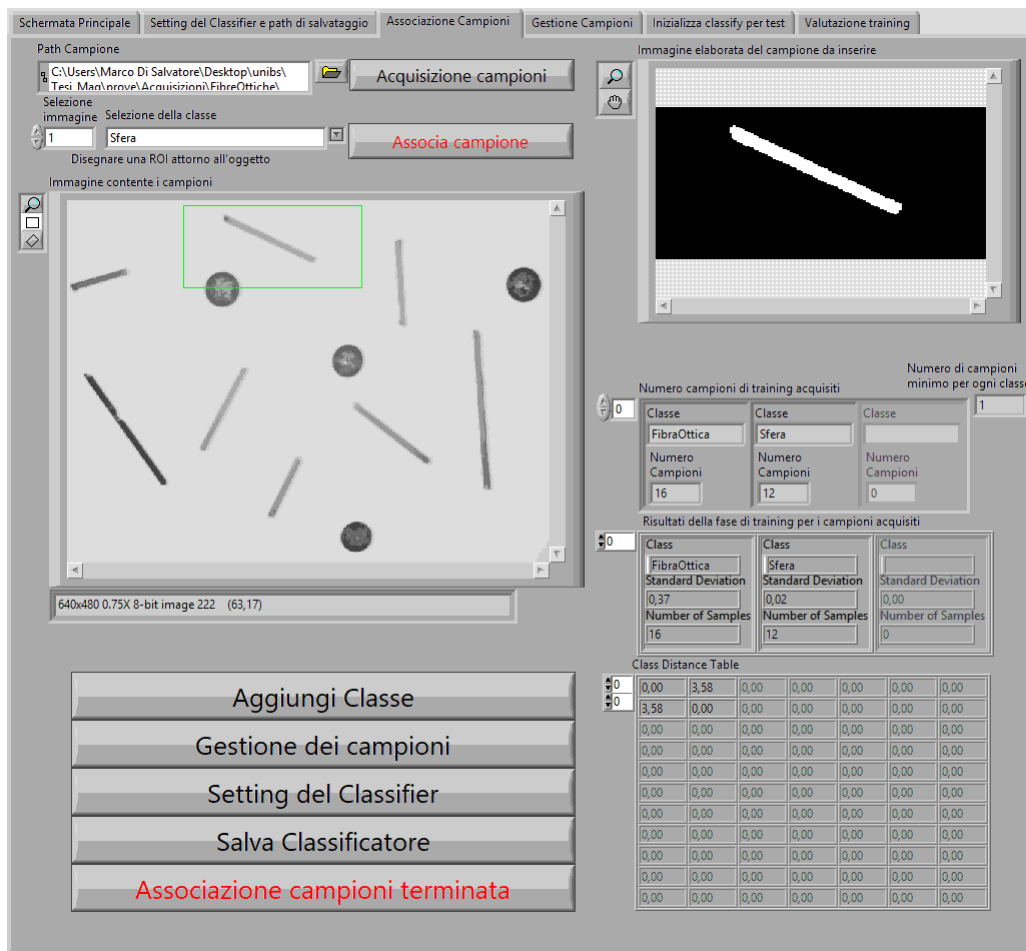


Figura 4.4: Pagina del programma NI in cui si effettua la creazione e il popolamento delle classi.

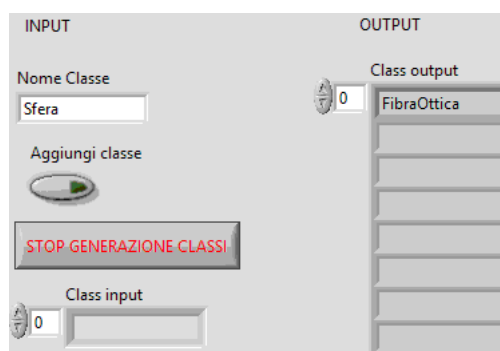


Figura 4.5: Pagina della funzione dedicata alla generazione dell'elenco delle classi.

è possibile generare una lista contenente i nomi di tutte le classi (Figura 4.5); le classi possono essere aggiunte in qualsiasi momento durante l'associazione dei campioni, non c'è bisogno quindi che le classi vengano generate in un unico momento.

Una volta generate le classi si seleziona con una ROI l'area dell'immagine contenente il campione dell'oggetto (uno ed uno solo) che verrà sottoposta alle operazioni di image processing, con i valori selezionati all'inizio, e visualizzata nella finestra "Immagine elaborata del campione da inserire", la quale permette di controllare se l'area selezionata contiene o meno un singolo campione. Una volta accertata la selezione del campione si definisce la classe di appartenenza attraverso il comando "Selezione della classe" e vi si associa il campione con il comando "Associa campione". Man mano che si procede con l'associazione si sono compilate in automatico tre tabelle:

- "Numero campioni di training acquisiti": è utilizzata sia dal software che dall'utente per controllare il numero di campioni associati a ciascuna classe, in modo tale da effettuare il training una volta che il numero di campioni di ogni classe è pari o superiore al "Numero di campioni minimo per ogni classe"; quest'ultimo numero è pari a uno nel caso del Nearest Neighbour e del Minimum Mean Distance, mentre è pari a k per il k -Nearest Neighbour.
- "Risultati della fase di training per i campioni acquisiti": sono riportati i risultati della fase di training, Deviazione Standard massima e numero di campioni di training.
- "Class distance table": riporta la distanza tra tutte le classi.

Le ultime due tabelle elencate sono utilizzate per la valutazione dei dati raccolti per il classificatore (Capitolo 3.3.1).

Terminata la fase di associazione si salva il file di classificazione. Il programma, inoltre, dà la possibilità di salvare il classificatore corrente senza terminare la fase di associazione. Tuttavia bisogna prestare attenzione a cambiare il nome o la cartella di destinazione del classificatore una volta effettuato il salvataggio, per evitare che il precedente classificatore venga sovrascritto; questa funzionalità permette di generare più classificatori, durante l'associazione, che presentino piccole o grandi differenze nei parametri di setting o nei campioni di training utilizzati.

4.1.3 Gestione delle classi e dei campioni associati

Il software presenta una terza parte nella quale si può effettuare la gestione dei campioni associati e delle classi. Nella pagina di "Gestione campioni" (Figura 4.6) viene indicato il numero totale di campioni che sono stati associati fino a quel momento ed è possibile visualizzarli incrementando o decrementando il cursore "Sample Index". Per il campione visualizzato nella finestra "Campioni" viene indicata la sua "Classe di appartenenza"; nel caso in cui ci sia stato un errore nell'associazione e il campione non sia rappresentativo della classe a cui è associato, il

campione può essere cancellato ("Cancella campione"), oppure riassegnato ("Cambiare classe") ad una classe diversa appartenente al gruppo di quelle generate in precedenza. Si possono infine cancellare tutti i campioni collegati a una classe ed eliminare quest'ultima ("Cancella classe"). Si precisa che la gestione dei campioni associati può essere fatta in qualsiasi momento della fase di training.

Tutte le operazioni effettuabili in questa fase sono seguite da un aggiornamento del classificatore, a patto che il numero di campioni associato a ciascuna classe rispetti le condizioni definite in precedenza.

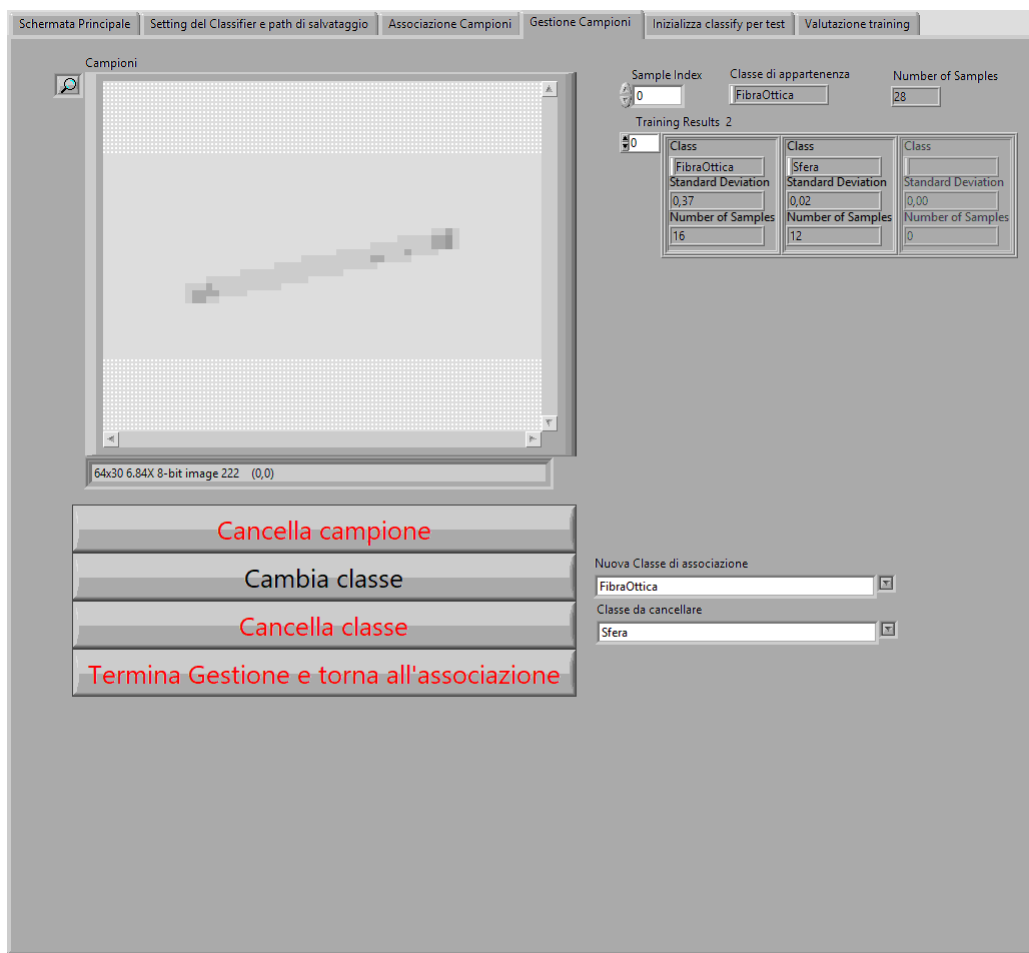


Figura 4.6: Pagina del programma NI in cui si effettua la gestione dei campioni associati alle classi.

4.2 Training nel caso del software Custom Particle Classifier PFV

Ciò che cambia nel training del Custom Particle Classifier rispetto al Particle Classifier (NI) è la possibilità di selezionare le feature da estrarre dai campioni e la possibilità di applicare ulteriori operazioni di image processing. Inoltre, è possibile effettuare una valutazione più approfondita del classificatore generato, dato che si può accedere ai valori delle feature estratte, a differenza del caso NI in cui non è possibile accedere a tutti i dati essendo un programma proprietario. Per questo motivo verranno illustrate solo le differenze del software PFV rispetto al software NI.

4.2.1 Selezione dei parametri di training

Per quanto riguarda la selezione dei parametri del classificatore (Figura [4.7](#)), è possibile applicare una tra le operazioni morfologiche binarie di chiusura (Capitolo [2.2.4](#)) o apertura (Capitolo [2.2.3](#)), di cui si possono definire la dimensione della maschera e il numero di iterazioni con cui applicarla. Per quanto riguarda le feature estraibili dall'immagine binarizzata, si fa riferimento alla sezione del Custom Particle Classifier del Capitolo [3.2.1](#) in cui sono tutte illustrate in modo approfondito. Il Custom Particle Classifier non permette di considerare la dimensione o la riflessione speculare dell'oggetto con l'utilizzo di un peso, come nel caso del Particle Classifier NI; tuttavia per rendere il classificatore dipendente dalla dimensione del campione, si deve selezionare la feature "Scale Area Ratio" (Capitolo [3.2.1](#)).

Per quanto riguarda il nome associato al file di classificazione, al momento del salvataggio viene aggiunta la desinenza "PFV" che, come nel caso del metodo NI, permette di identificare più facilmente il classificatore di tipo PFV.

La funzione usata per l'acquisizione è stata realizzata ad hoc per il metodo PFV per permettere di applicare tutte le operazioni di image processing possibili del metodo.

Come nel caso NI, tutti i setting del classificatore possono essere modificati durante la fase di associazione dei campioni, comprese le feature.

4.2.2 Generazione delle classi ed associazione dei campioni

Nella pagina di associazione dei campioni (Figura [4.8](#)), quando si disegna la ROI attorno al campione dell'oggetto, è possibile visionare i valori che assumono le features selezionate in precedenza, estratte dal campione utilizzando la particle analysis ("Valori feature campione selezionato").

Una volta che il numero di campioni minimo per ciascuna classe è stato raggiunto, si effettua il training che fornisce, oltre alla Deviazione Standard massima e al numero di campioni per cia-

scuna classe, la media e la Deviazione Standard di tutte le feature per ogni classe ("Deviazione Standard e media di ciascuna feature in una classe"). Con questa serie di dati è possibile effettuare una valutazione più approfondita dei campioni raccolti per ciascuna classe per valutare se le feature selezionate sono adatte per caratterizzare le classi generate.

4.2.3 Gestione delle classi e dei campioni associati

Per quanto riguarda la gestione dei campioni non si ha nessuna variazione nell'interfaccia e nelle funzionalità rispetto al programma NI.

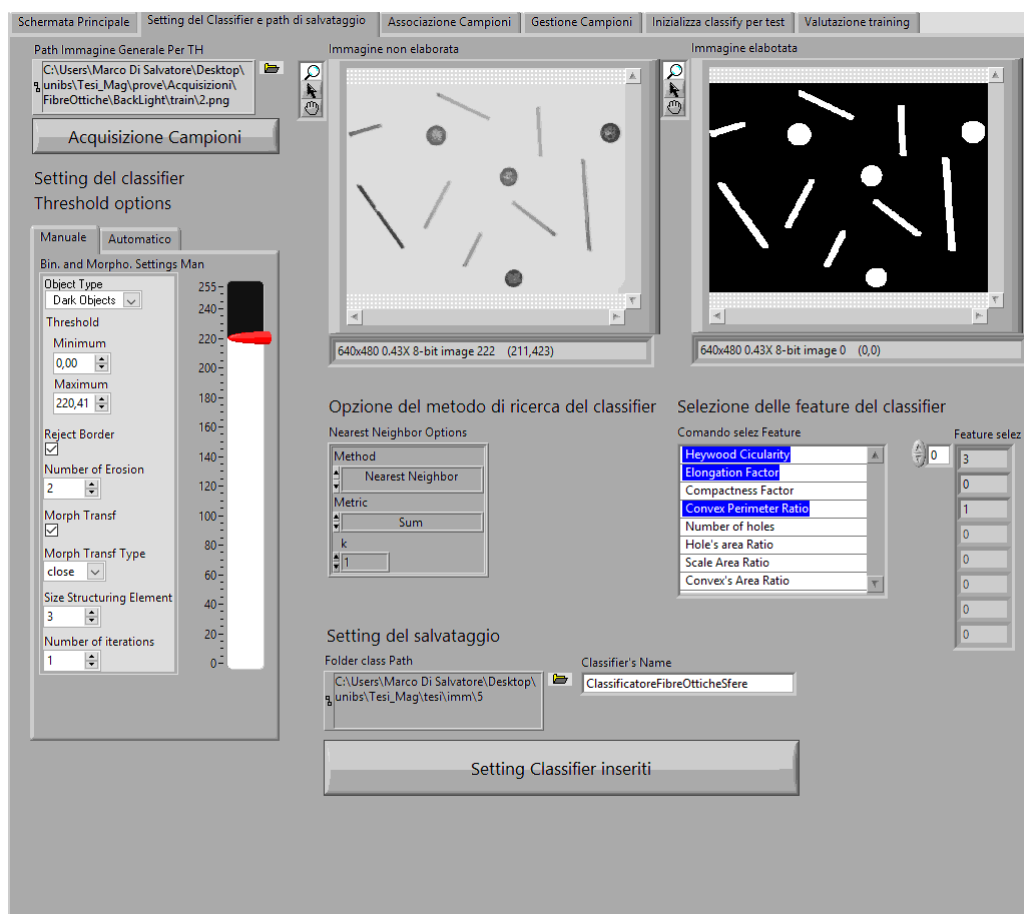


Figura 4.7: Pagina del software PFV dedicata alla selezione dei parametri del classificatore.

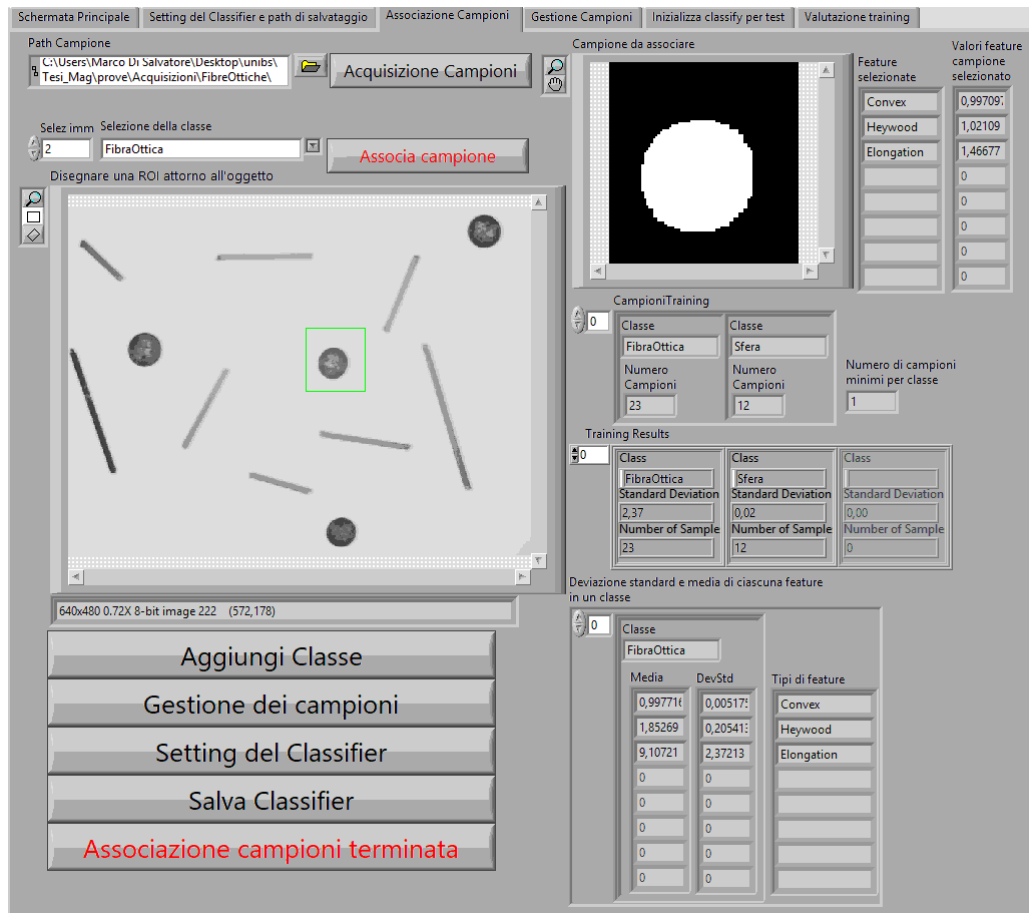


Figura 4.8: Pagina del programma PFV in cui si effettua la creazione e il popolamento delle classi.

4.3 Test del classificatore

La fase di test, fondamentale per verificare il corretto funzionamento del classificatore, può essere effettuata subito dopo la generazione del file di classificazione oppure con un classificatore generato precedentemente. In Figura 4.9 è riportata l'interfaccia del programma per l'esecuzione della fase di test (Modulo 2). La modalità di esecuzione e di restituzione dei risultati del modulo di test sono le stesse per entrambi i metodi, quello che cambia sono le operazioni di gestione ed elaborazione delle informazioni. Come per la fase di associazione dei campioni alle classi, le immagini di test possono essere raccolte in una cartella o possono essere acquisite con la stessa funzione della fase di associazione. Le immagini, contenenti i campioni di test, sono visualizzate nella finestra "Immagine di test" incrementando o decrementando l'indice "Selezione immagine di test". Una volta scelta un'immagine contenente uno o più campioni, si disegna un ROI attorno ad un campione e se ne seleziona la classe di appartenenza. Una volta selezionata la classe ("Indicare la classe di appartenenza") si effettua la classificazione dell'oggetto, ottenendone così la classe e il Classification Score (Capitolo 3.3.3). Tali opera-

zioni vengono ripetute per ciascun campione contenuto nelle immagini di test. Con i risultati raccolti è possibile compilare la "Classification distribution table" con i relativi indici di accuracy e predictability (Capitolo 3.3.2), mentre con i Classification Score ottenuti si compila la "Classification confidence distribution" (Capitolo 3.3.3) la quale identifica ogni classe, compreso il caso di errata classificazione (campione in cui la classe selezionata e quella identificata dal classificatore differiscono), con un colore differente per poter evidenziare il comportamento del classificatore al momento della classificazione (in basso a sinistra Capitolo 4.9).

Durante la fase di test è possibile variare, in modo permanente, il method e la distance selezionati al momento della generazione del classificatore, permettendo così d'identificare al meglio i due parametri per ottenere i migliori risultati possibili.

Poiché la selezione di tutti i campioni di test è un'operazione molto lunga, è possibile testare altri classificatori con gli stessi campioni selezionati, con l'unica condizione che i nomi e il numero delle classi di questo nuovo classificatore siano gli stessi di quello con cui si sono raccolti i campioni di test. Questa funzionalità consente di testare più classificatori che presentano settaggi differenti ma che hanno lo stesso insieme di classi.

Figura 4.9: Pagina dei programmi NI e PFV in cui si effettua il test del classificatore.

4.4 Applicazione della classificazione

Il modulo sviluppato (Modulo 3) può essere utilizzato con entrambi i tipi di classificatore perché è in grado di riconoscere automaticamente il tipo di classificatore che si carica, permettendo di applicare la sequenza di operazioni necessarie atte ad applicare il tipo di classificatore identificato. Il programma è suddiviso in tre parti che verranno illustrate nelle seguenti sezioni.

4.4.1 Scelta dell'immagine sorgente e del classificatore

Nella prima parte della classificazione si devono scegliere la sorgente dell'immagine e il classificatore da utilizzare, come si può vedere in Figura [4.10](#).

La classificazione può essere fatta su due tipi di sorgenti d'immagine:

- **Immagine statica:** la sorgente è un'immagine acquisita precedentemente, si parla quindi di classificazione Off-line. Per applicare la classificazione su un'immagine presente in memoria basta selezionarla con la gestione delle risorse.
- **Immagine dinamica:** la sorgente è un'immagine che viene acquisita in quel momento da un sistema di visione che inquadra o inquadrerà uno o più componenti da identificare, in questo caso si parla di classificazione On-line. Per applicare la classificazione su un'immagine acquisita in quel momento da un sistema di visione basta selezionare un sistema di visione di quelli collegati al PC.

Questi due tipi di sorgenti devono però essere caratterizzati dalle stesse condizioni di acquisizione dei campioni di training (FOV, distanza focale, illuminazione, ecc.), come già detto in precedenza, e vengono visualizzate nella finestra "Immagine generale".

Una volta scelta la sorgente è possibile selezionare il classificatore da utilizzare.

4.4.2 Scelta degli oggetti da identificare

Una volta scelta la sorgente e il classifier si passa allo step successivo (Figura [4.11](#)) in cui è presente una tabella dove per ogni riga è possibile selezionare una delle classi fornite dal classifier e indicare il numero di istanze che si desidera identificare: tale selezione è effettuabile con tutte le classi associate al classificatore in fase di training.

Un campione è considerato ben classificato se il Classification score (o l'Identification score) ottenuto ha un valore maggiore o uguale al controllo "Score minimo", che ha un valore di default pari a 800; quale score utilizzare per valutare la bontà della classificazione è scelto con il "Selettore Score per controllo". La selezione dello score a cui applicare il valore di soglia sarà illustrata nel Capitolo [5](#).

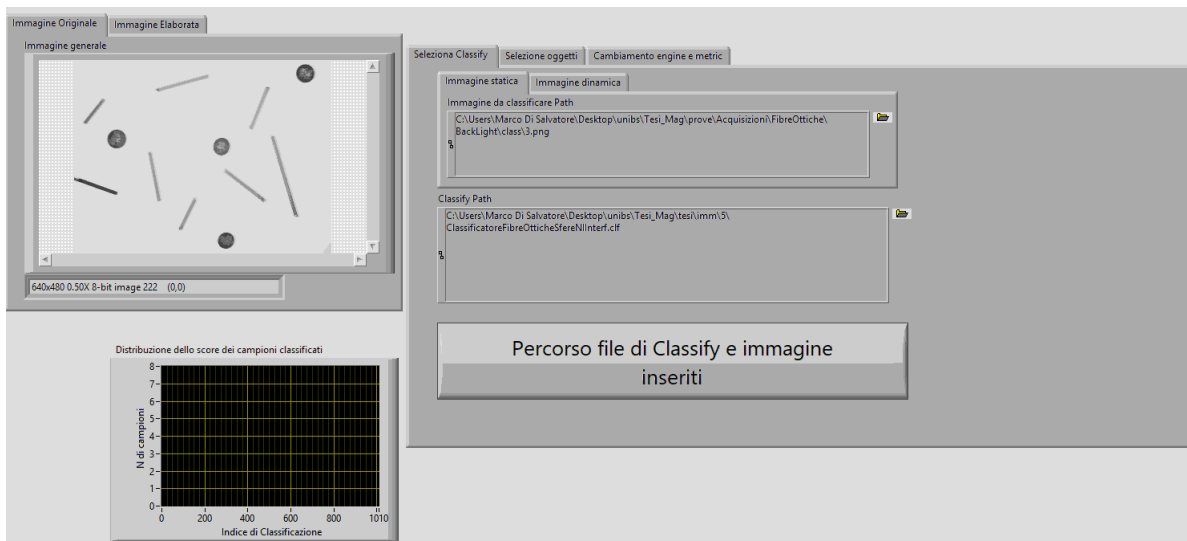


Figura 4.10: Pagina iniziale del modulo di classificazione.

Prima di effettuare la classificazione viene applicata la Particle Analys per identificare le posizioni e le ROI delle particelle nell'immagine rappresentanti i campioni da identificare. Nel caso in cui venga identificata una particella la cui ROI sia interna ad un'altra ROI, la particella della ROI interna non verrà classificata.

La classificazione, applicata a ciascuna particella, viene effettuata in modo continuo, cioè, sia nel caso della sorgente statica che dinamica, la classificazione viene effettuata ciclicamente, quindi il classificatore effettua una continua identificazione delle particelle dell'immagine. I risultati ottenuti vengono filtrati ed etichettati come ben classificati se il tipo di score, scelto precedentemente, supera il valore di score minimo selezionato in precedenza. Dagli oggetti etichettati come ben classificati vengono estratti gli oggetti richiesti nella tabella. Se il numero degli oggetti richiesti è maggiore o uguale al numero di quelli disponibili allora verranno forniti tutti i risultati etichettati come ben classificati; nel caso in cui invece il numero di campioni identificati sia maggiore rispetto a quello richiesto, verranno forniti i risultati con lo score più alto. Gli oggetti identificati e richiesti dall'utente vengono mostrati nella finestra "Risultati della classificazione" e vengono forniti in uscita raggruppati in classi e per ciascun oggetto vengono riportati: lo score ottenuto in fase di classificazione, la posizione del baricentro, il suo orientamento e la ROI che lo contiene. Con i risultati ottenuti viene anche creato il grafico della distribuzione della classification o identification confidence dei campioni identificati ("Distribuzione dello score dei campioni classificati").

Nel caso in cui le operazioni di image processing non risaltino al meglio le caratteristiche delle particelle, è possibile abilitare la modifica di tali parametri, sia per il software NI che per PFV, adattandoli per ottenere un migliore estrazione delle feature; c'è da evidenziare però che le modifiche a questi parametri sono temporanee, ovvero quando la classificazione sarà conclusa

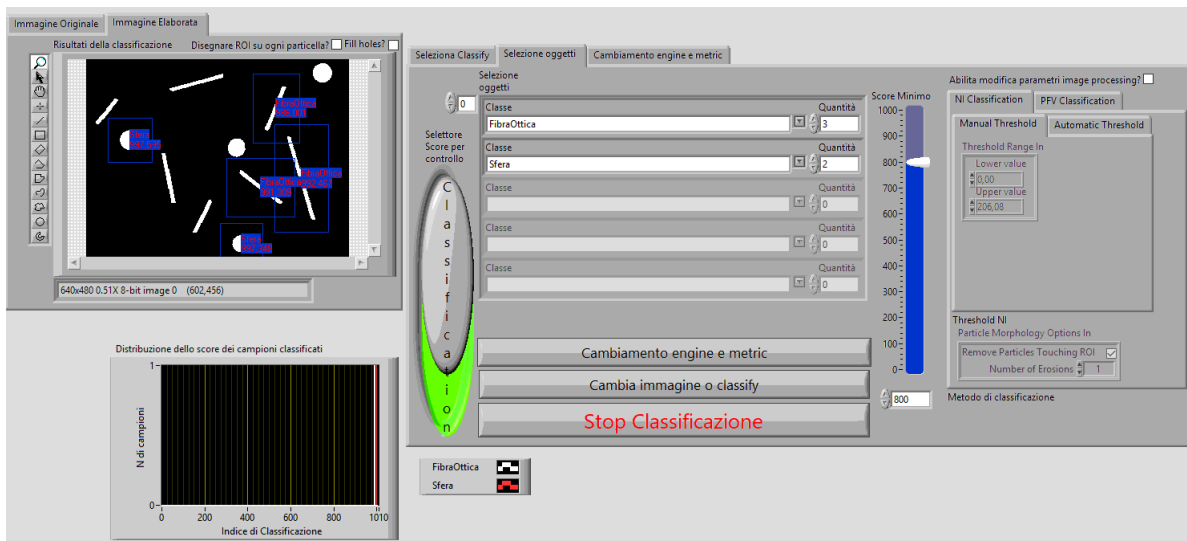


Figura 4.11: Pagina di selezione degli oggetti da identificare del software di classificazione.

le modifiche andranno perse. Nel caso in cui si effettuino le modifiche ai parametri di image processing i campioni di training non vengono rielaborati con i nuovi parametri.

Per identificare le particelle dei campioni da classificare nell'immagine si applicano le operazioni di image processing con i parametri selezionati durante la fase di training. Nel caso di NI, ciascuna regione dell'immagine in scala di grigi contenente la particella viene data in ingresso al classificatore che ne fornisce la classe di appartenenza e i relativi score; nel caso del PFV, ciascuna particella viene analizzata con la Particle Analysis per estrarre tutte le feature selezionate nella fase di training e compilarne il feature vector che a sua volta sarà posto in ingresso al classificatore che fornirà in uscita la classe di appartenenza con i relativi score.

4.4.3 Cambiamento del method e del distance metric

Durante la fase di classificazione è possibile modificare il method e il distance metric accedendo alla pagina "Cambiamento engine e metric" (Figura 4.12), nella quale è possibile consultare la "Class distance table" del classificatore, la Deviazione Standard massima e il numero di campioni per ciascuna classe. Una volta cambiati il method o il distance metric si torna alla fase di classificazione per verificare i cambiamenti dei risultati ottenuti; le modifiche vengono applicate permanentemente al classificatore. Questa funzione permette di identificare i miglior method e distance metric da utilizzare per il tipo di oggetti da identificare.

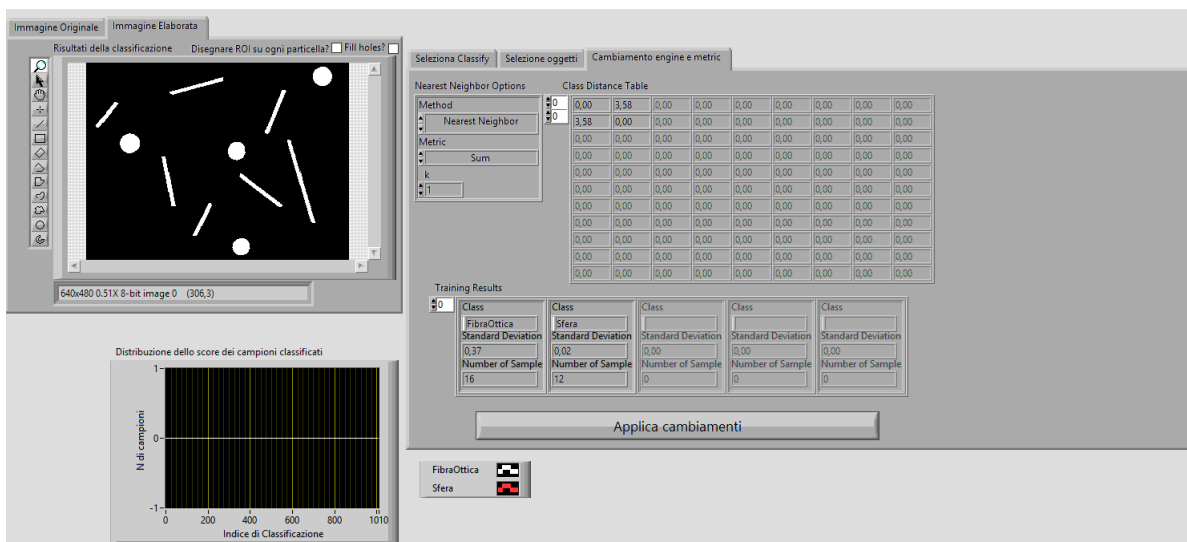


Figura 4.12: Pagina per il cambiamento del method e del distance metric del software di classificazione.

Capitolo 5

Test del software

Il metodo della classificazione ha tutte le caratteristiche per poter essere un ottimo algoritmo di machine vision dedicato all'identificazione e all'ispezione di oggetti. Quello che si è voluto dimostrare è la sua applicabilità ai fini industriali, in particolare nel settore micro.

Prima di procedere, bisogna però esprimere alcune caratteristiche degli oggetti micro per rendere idea a cosa ci si sta riferendo. Un componente micro è un oggetto che può avere la dimensione massima di qualche millimetro, come ingranaggi per orologi, fino ad arrivare ad oggetti con dimensioni inferiore a $10 \mu m$, come alcuni componenti elettronici SMD.

Sono stati effettuati una serie di test per verificare il funzionamento di diversi tipi di classificatori e, con i risultati ottenuti, è stato possibile individuare le diverse tipologie di applicazioni in cui è possibile utilizzare la classificazione; è stato inoltre possibile approfondire la valutazione degli indici legati alla fase di classificazione per la validazione della classe assegnata. Le prove effettuate sono state svolte applicando sia il Particle Classifier di NI che il Custom Particle Classifier PFV per poterne evidenziare gli aspetti positivi e negativi. I micro-componenti utilizzati per lo studio sono stati forniti gentilmente dal gruppo MEDIS dell'ITIA-CNR di Milano. I test sono stati eseguiti presso il laboratorio di micro-manipolazione dell'ITIA CNR di Milano e in particolare sono stati usati i sistemi di visione e gli illuminatori presenti nella cella di micro-manipolazione per effettuare le acquisizioni.

Nei test effettuati sono stati generati i seguenti tipi di classificatore:

- classificatori Scale Dependent ed Independent con monoclasse o multiclasse applicati a un insieme di blocchetti prismatici;
- classificatore Multiclasse Specifico applicato a un insieme di aste e dischi;
- classificatore Multiclasse Generale applicato a un insieme di viti e ruote dentate;
- classificatore Monoclasse applicato a un cilindro, un cilindro dentato e un insieme di ruote dentate;

- classificatore Multiclasse Mirror Dependent applicato all'orientamento di due oggetti.

Tutti questi test sono stati effettuati con immagini statiche, per questa ragione è stato effettuato un ultimo test in cui si sono applicati una serie di classificatori, con caratteristiche simili a quelli elencati precedentemente, ad immagini on-line acquisite da un sistema di visione.

5.1 Esempio di classificatori Scale Dependent ed Independent

Il primo test ha avuto come obiettivo quello di comprendere in modo più approfondito la funzionalità "Scale" attraverso l'analisi dei classificatori Scale Dependent ed Independent. Come illustrato nei capitoli precedenti, i parametri da impostare per considerare la dimensione di un oggetto come una caratteristica sono diversi per NI e PFV (Capitolo 3.2.1); nel primo caso, si indica il peso con cui considerare la caratteristica "dimensione", nel secondo caso invece basta semplicemente selezionare lo Scale Area Ratio come caratteristica da inserire nel feature vector.

Gli oggetti scelti per la prova sono tre blocchetti di forma rettangolare, con dimensione massima di $1\text{mm} \times 2\text{mm}$, che presentano come unica differenza la dimensione (Figura 5.1). Per ogni tipo di classificatore è stata generata una serie di classificatori suddivisi in due categorie, Scale Dependent ed Independent, in ciascuna delle quali sono presenti un classificatore multiclasse, costituito da più di una classe, e tre classificatori monoclasse, uno per ciascun tipo di oggetto, per un totale di 16 classificatori. Si precisa che, per generare il classificatore monoclasse, nella fase di training si utilizzano solo i campioni di una classe, mentre nel caso multiclasse si utilizzano tutti i campioni e si ripartiscono tra le varie classi. Saranno riportati solo i risultati dei training dei classificatori multiclasse in quanto i risultati ottenuti con il classificatore monoclasse sono uguali siccome i campioni utilizzati sono gli stessi per il training.

Per verificare il comportamento dei differenti classificatori si è deciso di effettuare:

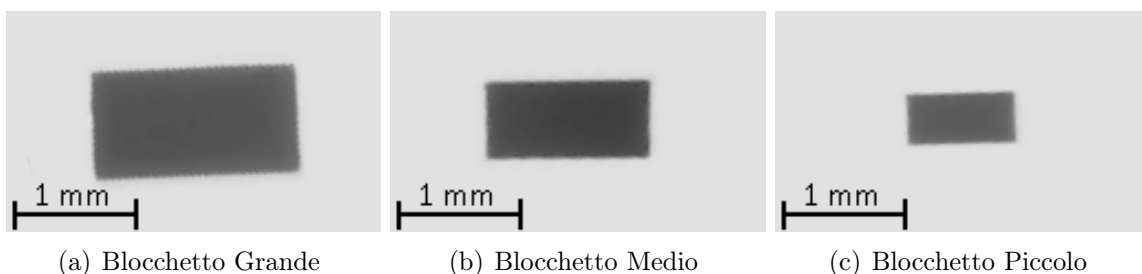


Figura 5.1: Oggetti utilizzati nella prova.

- una fase di test per i classificatori multiclasse per poterne valutare l'accuracy, la predictability e la media del classification score di ciascuna classe (non si analizza l'identification score perché in fase di test non viene fornito);
- una fase di classificazione per i classificatori monoclasse per poterne valutare il massimo classification ed identification score associati a ciascun tipo di campione.

Per rendere i risultati confrontabili si sono scelti gli stessi parametri di image processing (Tabella 5.1) e di metodo d'assegnazione della classe (Tabella 5.2) sia per NI che per PFV; per quest'ultimo non sono state applicate operazioni morfologiche binarie e la scelta delle feature è stata fatta per poter misurare le peculiarità degli oggetti.

Threshold	Tipo	Manuale
	Tipo di oggetto	Dark
	Valore (0-255)	97,85
Reject border	Applicato	
Numero di iterazioni di erosione	1	

Tabella 5.1: Valori dei parametri di image processing applicati in tutti i classificatori Scale Dependent ed Independent.

Method	Nearest Neighbor
Distance Metric	Sum
k	1

Tabella 5.2: Method e Distance Metric applicati in tutti i classificatori Scale Dependent ed Independent.

5.1.1 Classificatori Multiclasse

Le impostazioni della fase di training dei classificatori multiclasse sono riportati nelle Tabelle 5.3, 5.4, 5.7, 5.8, nelle quali non sono state riportate tutte le combinazioni di impostazioni ottenute ma solo le più significative e, nel caso dei classificatori PFV, è stato riportato per ciascuna classe il tipo di feature che possiede la deviazione standard massima e il suo relativo valore medio.

I risultati ottenuti dalla fase di test eseguita con 57 campioni (Tabelle 5.5, 5.6, 5.9, 5.10), effettuata analizzando entrambi i metodi di classificazione (NI e PFV) Scale Independent, hanno mostrato un'accuratezza totale intorno al 70%, una prevedibilità intorno al 72% ed classification score medi tutti inferiori a 600. Questi valori indicano che il classificatore utilizzato non è affidabile. Questo si spiega nel fatto che le classi sono dal punto di vista delle caratteristiche di forma molto simili, e quindi usando un classificatore scale independent l'appartenenza univoca

ad una classe specifica risulta difficoltosa. Nel caso invece dei classificatori Scale Dependent si ha un evidente miglioramento con entrambi i metodi di classificazione: l'accuratezza totale è mediamente intorno al 99% e le medie dei classification score assumono valori superiori a 900. Dopo l'analisi di questi dati si capisce come la scelta di considerare il fattore "dimensione" permetta di ottenere risultati più affidabili laddove la dimensione sia un fattore caratterizzante la classe. Si fa notare però che, nel caso NI, il peso con cui è stato considerato il fattore di scala è di 500, poiché l'utilizzo di uno scale factor pari al valore massimo (ovvero 1000 completa dipendenza dal fattore di scala) avrebbe portato il classificatore ad assumere valori prossimi al caso Scale Independent dato che si considererebbe solo la dimensione come caratteristica per applicare il metodo.

Classi	Blocco Grande	Blocco Medio	Blocco Piccolo	Totale campioni
Numero di campioni di training	15	23	15	53
Deviazione standard massima	0	0.02	0.01	
Scale factor	0			
Mirror factor	0			

Tabella 5.3: Caratteristiche di training del classificatore NI Multiclasse Scale Independent.

Classi	Blocco Grande	Blocco Medio	Blocco Piccolo	Totale campioni
Numero di campioni di training	15	23	15	53
Deviazione standard massima	0.02	0.09	0.03	
Feature con dev. std. max	Elongation Factor	Elongation Factor	Elongation Factor	
Valore medio feature con dev. std. max	2.37	2.53	2.6	
Feature selezionate	Heywood Circularity Factor e Elongation Factor			

Tabella 5.4: Caratteristiche di training del classificatore PFV Multiclasse Scale Independent.

Classi	Blocco Grande	Blocco Medio	Blocco Piccolo	Totale campioni
Numero di campioni di test	14	24	19	57
Media del classification score	414	464.385	435.765	
Accuracy	78.57	54.17	89.47	Acc. Tot. 71.93
Predictability	73.33	72.22	70.83	

Tabella 5.5: Risultati fase di test del classificatore NI Multiclasse Scale Independent.

Classi	Blocco Grande	Blocco Medio	Blocco Piccolo	Totale campioni
Numero di campioni di test	14	24	19	57
Media del classification score	477.364	548.923	452.8	
Accuracy	73.33	54.17	78.95	Acc. Tot. 68.42
Predictability	78.57	65	62.5	

Tabella 5.6: Risultati fase di test del classificatore PFV Multiclasse Scale Independent.

Classi	Blocco Grande	Blocco Medio	Blocco Piccolo	Totale campioni
Numero di campioni di training	15	23	15	53
Deviazione standard massima	61.43	141.14	5.64	
Scale factor	500			
Mirror factor	0			

Tabella 5.7: Caratteristiche di training del classificatore NI Multiclasse Scale Dependent.

Classi	Blocco Grande	Blocco Medio	Blocco Piccolo	Totale campioni
Numero di campioni di training	15	23	15	53
Deviazione standard massima	0.21	0.46	0.03	
Feature con dev. std. max	Scale area Ratio	Scale area Ratio	Scale area Ratio	
Valore medio feature con dev. std. max	15.00	8.35	3.48	
Feature selezionate	Heywood Circularity Factor, Elongation Factor e Scale Area Ratio			

Tabella 5.8: Caratteristiche di training del classificatore PFV Multiclasse Scale Dependent.

Classi	Blocco Grande	Blocco Medio	Blocco Piccolo	Totale campioni
Numero di campioni di test	14	24	19	57
Media del classification score	978.643	946.042	979.211	
Accuracy	100	100	100	Acc. Tot. 100
Predictability	100	100	100	

Tabella 5.9: Risultati fase di test del classificatore NI Multiclasse Scale Dependent.

Classi	Blocco Grande	Blocco Medio	Blocco Piccolo	Totale campioni
Numero di campioni di test	14	24	19	57
Media del classification score	989.357	989.75	992.368	
Accuracy	93.33	100	100	Acc. Tot. 98.25
Predictability	100	96	100	

Tabella 5.10: Risultati fase di test del classificatore PFV Multiclasse Scale Dependent.

5.1.2 Classificatori Monoclasse

Si passa adesso all'analisi dei classificatori monoclasse che in totale sono 12, metà NI e metà PFV. I risultati ottenuti dalla fase di training, come detto in precedenza, sono gli stessi di quelli del caso multiclasse. Per verificare le performance dei classificatori è stata effettuata la classificazione di un'immagine contenente almeno un campione per ciascun tipo di blocchetto (Figura 5.2) e sono stati rilevati il classification ed l'identification score massimi assunti da ogni gruppo di campioni appartenenti a un certo tipo di oggetto. Dai risultati ottenuti (Tabelle 5.11, 5.12, 5.13, 5.14), si nota subito che il classification score non è utile per valutare la bontà della classe applicata ad un campione in quanto assume sempre un valore pari a 1000, questo deriva dal fatto che tale indice per definizione (Capitolo 3.3.3) è calcolato tenendo in considerazione anche la seconda classe che rappresenta meglio il campione. Tuttavia, non è possibile calcolare quest'ultimo indice poiché il classificatore è composto da una sola classe; per questo motivo gli viene attribuito il suo valore massimo. Non resta che analizzare gli identification score ottenuti: questi, nel caso di classificatori Scale Independent (Tabelle 5.11, 5.12), assumono valori elevati con tutti gli oggetti senza permettere alcuna discriminazione. Nel caso dei classificatori Scale Dependent (Tabelle 5.13, 5.14) la situazione migliora: vengono assegnati identification score superiori a 900 a tutti quei campioni che appartengono al tipo di campioni rappresentati dal classificatore. Per i restanti campioni, nel caso NI vengono assegnati valori inferiori 850. Questo può essere un problema perché non vi è una netta distinzione tra l'indice assegnato ad un campione appartenente ad una classe e ad uno non appartenente. Al contrario, nel caso PFV gli identification score sono pari a 0 quindi vi è una differenza consistente che permette di avere una maggior certezza dell'appartenenza o meno del campione alla classe. In conclusione, l'utilizzo del fattore di scala nel caso monoclasse, permette il riconoscimento dell'appartenenza o meno di un campione a una classe nel caso in cui la dimensione sia caratterizzante per quest'ultima. La valutazione della conformità (o appartenenza) ad una classe è effettuata mediante l'imposizione di una soglia minima di identification score che permette di decretare la conformità dell'oggetto classificato.

Classificatore Campione \ Score	Blocco Grande		Blocco Medio		Blocco Piccolo	
	Classific.	Identific.	Classific.	Identific.	Classific.	Identific.
Blocco Grande	1000	993.94	1000	993.34	1000	908.562
Blocco Medio	1000	993.696	1000	997.326	1000	995.128
Blocco Piccolo	1000	918.263	1000	991.109	1000	995.997

Tabella 5.11: Risultati fase di classificazione del classificatore NI Monoclasse Scale Independent.

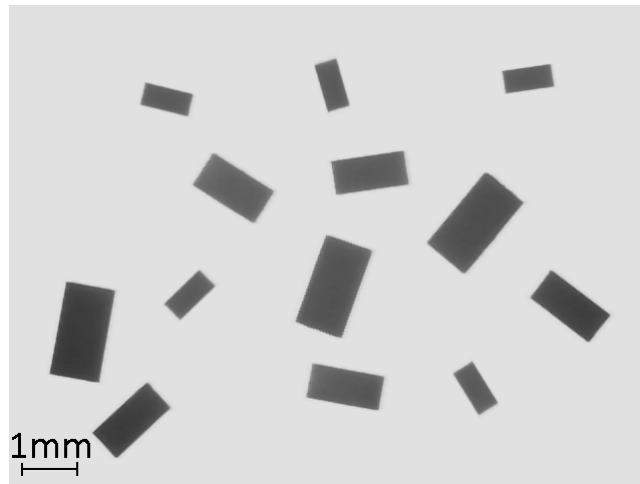


Figura 5.2: Immagine contenente differenti campioni su cui si è applicato il classificatore monoclasse.

Classificatore	Blocco Grande		Blocco Medio		Blocco Piccolo	
	Campione \ Score	Classific.	Identific.	Classific.	Identific.	Classific.
Blocco Grande	1000	996.97	1000	993.903	1000	821.996
Blocco Medio	1000	995.028	1000	996.687	1000	993.621
Blocco Piccolo	1000	848.649	1000	996.243	1000	995.849

Tabella 5.12: Risultati fase di classificazione del classificatore PFV Monoclasse Scale Independent.

Classificatore	Blocco Grande		Blocco Medio		Blocco Piccolo	
	Campione \ Score	Classific.	Identific.	Classific.	Identific.	Classific.
Blocco Grande	1000	995.508	1000	816.939	1000	571.332
Blocco Medio	1000	800.49	1000	996.293	1000	726.525
Blocco Piccolo	1000	580.101	1000	726.525	1000	997.809

Tabella 5.13: Risultati fase di classificazione del classificatore NI Monoclasse Scale Dependent.

Classificatore	Blocco Grande		Blocco Medio		Blocco Piccolo	
	Campione \ Score	Classific.	Identific.	Classific.	Identific.	Classific.
Blocco Grande	1000	969.304	1000	0	1000	0
Blocco Medio	1000	0	1000	953.733	1000	0
Blocco Piccolo	1000	0	1000	0	1000	989.338

Tabella 5.14: Risultati fase di classificazione del classificatore PFV Monoclasse Scale Dependent.

5.2 Esempio di classificatore Multiclasse Specifico

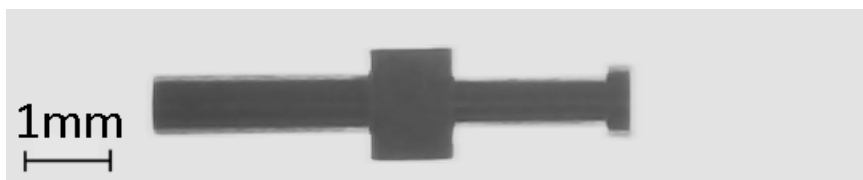
Una volta approfondito il fattore di scala si è passati all'analisi di un classificatore multiclasse che, dato un insieme di oggetti da classificare, avrà una classe per ogni tipo di oggetto e sarà definito come Classificatore Multiclasse Specifico. In questo esempio si sono generati due classificatori, uno NI e uno PFV, e per ciascuno di questi sono stati analizzati:

- i risultati ottenuti dopo una fase di test con una serie di oggetti uguali a quelli usati nella fase di training (oggetti non estranei);
- i risultati ottenuti dalla classificazione applicata a un'immagine in cui sono contenuti una serie di oggetti diversi da quelli utilizzati per il training (oggetti estranei).

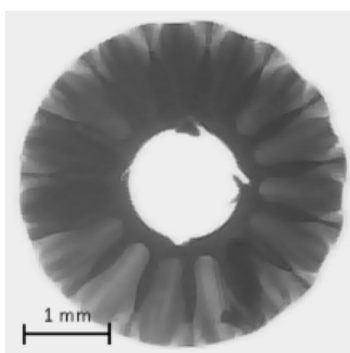
Gli oggetti utilizzati per la fase di training sono mostrati in Figura 5.3 e sono caratterizzati da forme circolari o allungate. Questi oggetti sono stati scelti per valutare la capacità del classificatore di distinguere campioni di oggetti con caratteristiche comuni ma che appartengono a classi diverse. Le prove sono state effettuate anche per verificare la robustezza nel classificare oggetti estranei che non appartengono a nessuna classe.



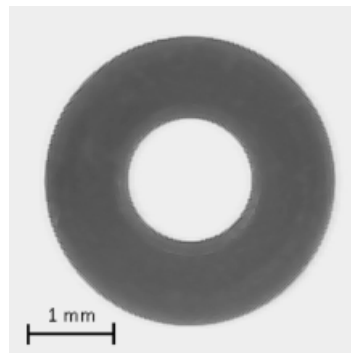
(a) Asta Grande



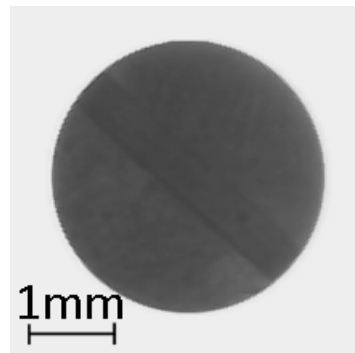
(b) Asta Piccola



(c) Disco Forato Dentato



(d) Disco Forato



(e) Disco

Figura 5.3: Oggetti utilizzati per la fase di training.

Come nei test precedenti, vengono utilizzati gli stessi parametri di image processing (Tabella 5.15) e di metodo di assegnazione delle classi (Tabella 5.16) per permettere la comparazione dei risultati ottenuti tra tutti i classificatori. In particolare, per il classificatore PFV non si applicano trasformazioni morfologiche binarie all'immagine.

Le impostazioni della fase di training sono riportati nelle Tabelle 5.18 e 5.17, i classificatori sono Scale Dependent perché gli oggetti hanno una dimensione diversa che li contraddistingue.

Threshold	Tipo	Manuale
	Tipo di oggetto	Dark
	Valore (0-255)	128
Reject border	Applicato	
Numero di iterazioni di erosione	1	

Tabella 5.15: Valori di image processing applicati in tutti i classificatori Multiclasse Specifico.

Method	Nearest Neighbor
Distance Metric	Sum
k	1

Tabella 5.16: Method e Distance Metric applicati in tutti i classificatori Multiclasse Specifico

Classi	Asta Grande	Asta Piccola	Disco Forato Dentato	Disco Forato	Disco	Totale campioni
Numero di campioni di training	4	5	6	7	8	30
Deviazione standard massima	0.44	0.41	0.93	0.25	0.22	
Feature con dev. std. max	Scale Area Ratio	Scale Area Ratio	Scale Area Ratio	Scale Area Ratio	Scale Area Ratio	
Valore medio feature con dev. std. max	48.61	39.2	103.813	83.902	73.876	
Feature selezionate	Heywood Circularity Factor, Elongation Factor, Convex Perimeter Ratio, Hole's Area Ratio e Scale Area Ratio					

Tabella 5.17: Caratteristiche di training del classificatore PFV Multiclasse Specifico.

Classi	Asta Grande	Asta Piccola	Disco Forato Dentato	Disco Forato	Disco	Totale campioni
Numero di campioni di training	4	5	6	7	8	30
Deviazione standard massima	144.56	193.28	295.89	111.24	70.5	
Scale factor	400					
Mirror factor	0					

Tabella 5.18: Caratteristiche di training del classificatore NI Multiclasse Specifico.

5.2.1 Campioni non estranei

Nella prima prova si sono effettuati due test, uno per metodo di classificazione, con una serie di campioni non estranei. Con il classificatore NI si è ottenuta una accuratezza totale del 100% e il classification score medio per ogni classe è superiore a 900, come si può vedere da Tabella 5.19. Per quanto riguarda il classificatore PFV si ha invece un'accuratezza totale del 96.87%, che è leggermente inferiore a quella ottenuta con NI a causa un campione Disco Forato che è stato classificato come Disco Forato Dentato, questo errore causato dal fatto che i due oggetti sono molto simili e perché in questo caso il campione, dopo le operazioni di image processing, presentava delle irregolarità che hanno portato il classificatore a sbagliare. Il problema appena illustrato ha causato lo stesso problema alla prevedibilità del Disco Forato Dentato che è pari al 85.71%, mentre per le altre classi è pari al 100%. Infine il classification score medio per ciascuna classe assume valori superiori a 900 come si può vedere da Tabella 5.19. Questo esempio ha dimostrato come il classificatore specifico sia in grado di poter identificare oggetti con caratteristiche comuni ma appartenenti a classi differenti. Si precisa che le feature scelte per il classificatore PFV giocano un ruolo fondamentale per la corretta identificazione dei campioni; sono state quindi scelte le feature Convex Perimeter Ratio, Hole's Area Ratio e Scale Area Ratio per poter evidenziare le peculiarità che differenziano le classi.

Classi	Asta Grande	Asta Piccola	Disco Forato Dentato	Disco Forato	Disco	Totale campioni
Numero di campioni di test	5	5	6	7	8	31
Media del classification score	984.6	993.2	944.667	987.571	996.125	
Accuracy	100	100	100	100	100	Acc.Tot. 100
Predictability	100	100	100	100	100	

Tabella 5.19: Risultati della fase di test del classificatore NI con campioni non estranei.

Classi	Asta Grande	Asta Piccola	Disco Forato Dentato	Disco Forato	Disco	Totale campioni
Numero di campioni di test	5	5	6	7	8	31
Media del classification score	967.333	994.4	989.83	995.833	911	
Accuracy	100	100	100	85.71	100	Acc.Tot. 96.87
Predictability	100	100	85.71	100	100	

Tabella 5.20: Risultati della fase di test del classificatore PFV con campioni non estranei.

5.2.2 Campioni estranei

Nella seconda prova si effettua la classificazione di una serie di campioni estranei (Figura 5.4) contenuti in un'unica immagine (Figura 5.5). L'obiettivo della prova è stato quello di analizzare la classe di assegnazione e i valori di classification ed identification score assegnati all'oggetto estraneo con entrambi i tipi di classificatori (Tabelle 5.21 e 5.22). Da una prima analisi si nota che l'utilizzo dell'identification score non è adatto per scartare l'oggetto estraneo in quanto in alcuni casi assume un valore superiore a 900 ed in particolare nel caso NI assume valori prossimi a 1000. I classification score assumono valori al di sotto di 600, ad eccezione dei campioni Asta Media e Disco Forato Piccolo che, nel caso PFV, assumono valori superiori a 900 poiché le feature selezionate, sotto il profilo del feature vector, rendono questi campioni molto simili a quelli che invece appartengono realmente alle classe assegnate ai campioni estranei. Da quest'ultima analisi si evidenzia che il classification score assume valori più alti nel caso in cui la forma del campione estraneo assomiglia a quella dell'oggetto rappresentante la classe. In conclusione si può dire che imporre una soglia minima di classification score per la selezione dei risultati ottenuti è una buona soluzione per scartare oggetti estranei con entrambi i tipi di classificatori. Nel caso PFV è opportuno prestare sempre molta attenzione alla selezione del tipo di feature, poiché devono permettere l'eventuale discriminazione rispetto ad un eventuale campione estraneo, pur mantenendo la capacità di caratterizzare la classe.

Campione	Classe assegnata	Score	
		Classification	Identification
Gancio	Asta Piccola	96.132	999.844
Rivetto	Asta Piccola	564.094	999.925
Asta Media	Asta Piccola	616.299	999.965
Spina	Disco	146.985	999.92
Cilindro	Disco	328.763	999.959
Disco Foro Piccolo	Disco Foro Grande	16.507	999.98

Tabella 5.21: Risultati della fase di classificazione del classificatore NI con campioni estranei.

Campione	Classe assegnata	Score	
		Classification	Identification
Gancio	Asta Piccola	435.63	699.337
Rivetto	Asta Piccola	367.51	598.86
Asta Media	Asta Piccola	902.71	979.976
Spina	Asta Piccola	490.56	757.976
Cilindro	Asta Piccola	568.31	823.421
Disco Foro Piccolo	Disco Forato Dentato	952.65	985.285

Tabella 5.22: Risultati della fase di classificazione del classificatore PFV con campioni estranei.

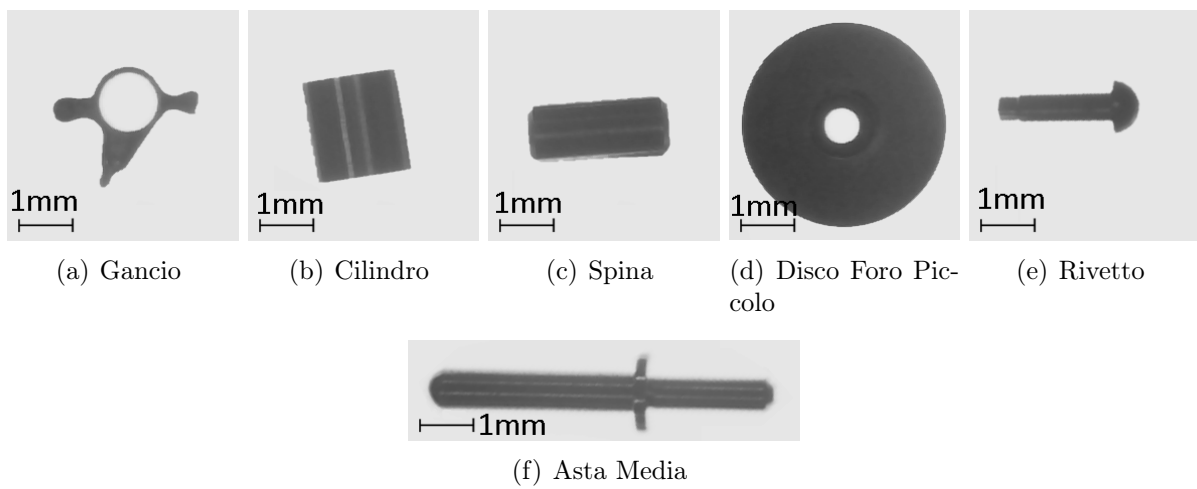


Figura 5.4: Oggetti utilizzati come campioni estranei.

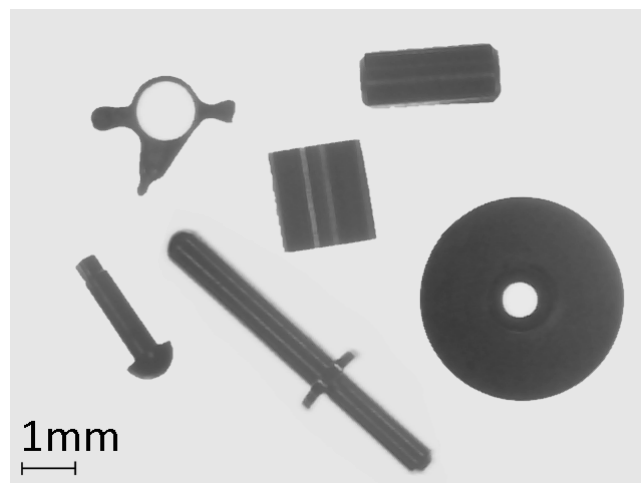


Figura 5.5: Immagine su cui è stato applicato il classificatore Multiclasse Specifico.

5.3 Esempio di classificatore Multiclasse Generale

Il prossimo esempio permette di analizzare il caso dell'applicazione di due classificatori, uno per NI e uno PFV, di tipo Multiclasse Generale. Questo tipo di classificatore è caratterizzato da classi costituite da più di un oggetto, non più uno solo come nel caso dei classificatori Multiclasse Specifici. Gli oggetti appartenenti ad una classe presentano caratteristiche di forma simili: ad esempio si può creare la classe "dischi" che sarà costituita da un insieme di dischi forati o meno di differente dimensione, ma tutti accumulati dalla forma circolare. Questo tipo di classificatore può essere utilizzato per la macro-identificazione della categoria di appartenenza di un oggetto. Tale operazione può essere seguita dall'utilizzo di un ulteriore classificatore specializzato per la categoria identificata dal classificatore. Le classi di oggetti generati per questo esempio sono due: Vite e Ruota Dentata (Figure 5.7, 5.6). Le caratteristiche comuni nel gruppo delle viti sono la forma allungata e la convessità creata dalla testa della vite; nel gruppo delle ruote dentate sono la forma circolare, la presenza di numerose convessità dovute alla dentatura e in alcuni casi la presenza di fori.

Per verificare la validità del tipo di classificatore si sono analizzati i risultati ottenuti:

- dopo una fase di test con una serie di oggetti uguali a quelli usati nella fase di training (non estranei);
- dalla classificazione applicata ad un'immagine in cui sono contenuti una serie di oggetti diversi da quelli utilizzati per il training (estranei);

Il method e il distance metric scelti sono gli stessi per entrambi i metodi NI e PFV (Tabella 5.23), mentre sono stati scelti due valori di threshold differenti. Come si può appunto vedere in Tabella 5.24, nel caso del classificatore PFV si utilizza un valore di soglia inferiore dato che, per migliorare l'immagine, si applica la trasformazione morfologica binaria di chiusura (Tabella 5.25). Le impostazioni del training dei due classificatori sono riportati nelle Tabelle 5.26 e 5.27, per quanto riguarda il classificatore PFV le feature selezionate permettono di misurare le caratteristiche comuni di ciascuna classe elencate in precedenza.

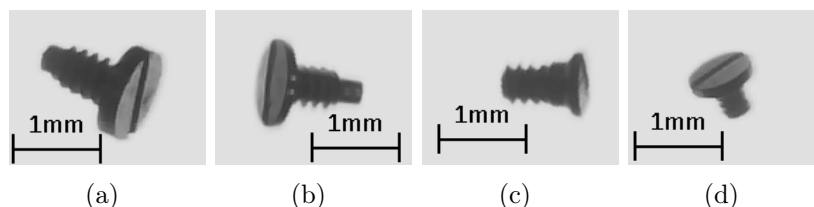


Figura 5.6: Oggetti della classe Vite.

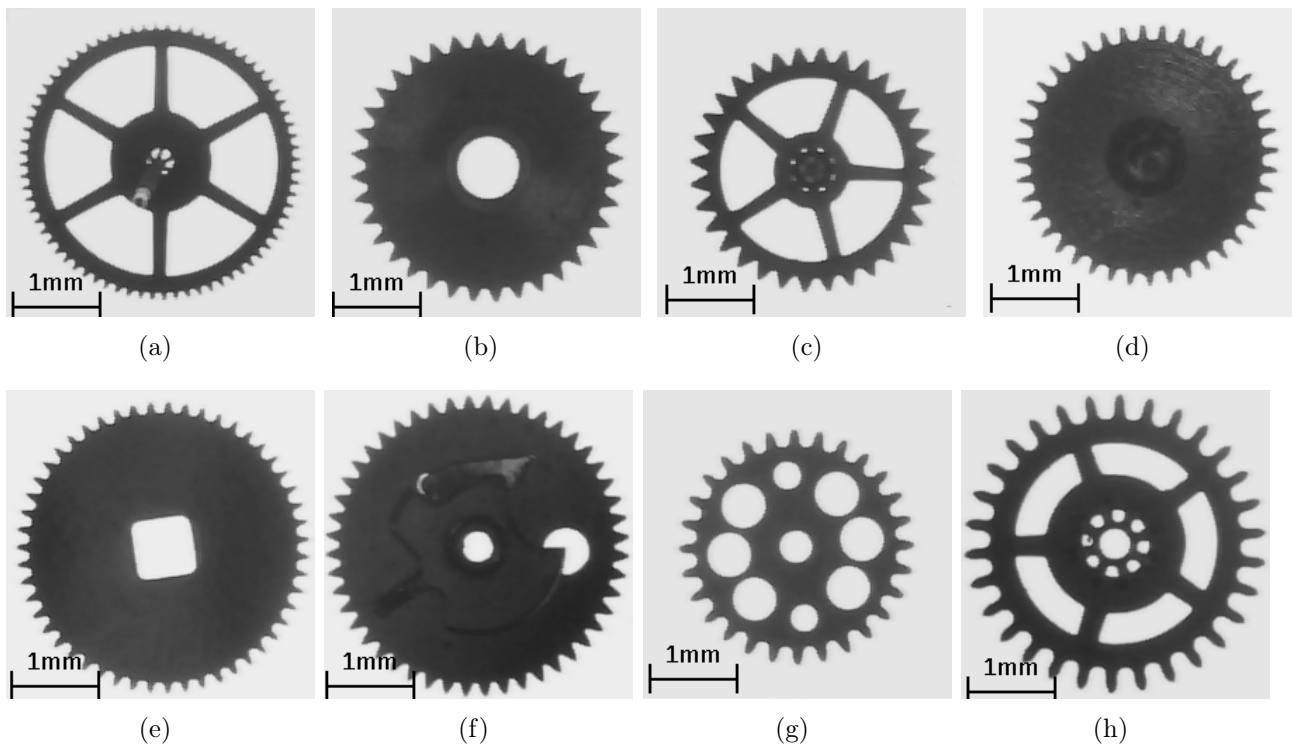


Figura 5.7: Oggetti della classe Ruota Dentata.

Method	Nearest Neighbor
Distance Metric	Sum
k	1

Tabella 5.23: Method e Distance Metric applicati a tutti i classificatori Multiclasse Generale.

Threshold	Tipo	Manuale
	Tipo di oggetto	Dark
	Valore (0-255)	NI:154 PFV:107
Reject border	Applicato	
Numero di iterazioni di erosione	1	

Tabella 5.24: Valori di image processing applicati a tutti i classificatori Multiclasse Generale.

Morphologic Transformation Type	Tipo	Chiusura
	Size	3
	n. di iter.	2

Tabella 5.25: Morphologic Transformation Type applicata ai classificatori PFV Multiclasse Generale.

Classi	Vite	Ruota Dentata	Totale campioni
Numero di campioni di training	21	46	67
Deviazione standard massima	0.06	0.22	
Scale factor	0		
Mirror factor	0		

Tabella 5.26: Caratteristiche di training del classificatore NI Multiclasse Generale.

Classi	Vite	Ruota Dentata	Totale campioni
Numero di campioni di training	21	46	67
Deviazione standard massima	0.19	0.42	
Feature con dev. std. max	Heywood Circularity	Heywood Circularity	
Valore medio feature con dev. std. max	1.34	1.77	
Feature selezionate	Heywood Circularity Factor, Elongation Factor, Hole's Area Ratio e Convex Area Ratio		

Tabella 5.27: Caratteristiche di training del classificatore PFV Multiclasse Generale.

5.3.1 Campioni non estranei

Nella prima prova è stata effettuata una fase di test con una serie di campioni uguali a quelli utilizzati nella fase di training. I risultati ottenuti con entrambi i metodi (Tabelle 5.28 e 5.29) sono molto soddisfacenti. Si ha un'accuratezza e una prevedibilità pari al 100% e un classification score medio per ciascuna classe superiore a 900. Dai risultati ottenuti si dimostra l'applicabilità del classificatore Multiclasse Generale come macro-identificatore.

Classi	Vite	Ruota Dentata	Totale campioni
Numero di campioni di test	24	53	77
Media del classification score	955.375	980	
Accuracy	100	100	Acc. Tot. 100
Predictability	100	100	

Tabella 5.28: Risultati della fase di test del classificatore NI con campioni non estranei.

Classi	Vite	Ruota Dentata	Totale campioni
Numero di campioni di test	24	53	77
Media del classification score	922.167	965.264	
Accuracy	100	100	Acc. Tot. 100
Predictability	100	100	

Tabella 5.29: Risultati della fase di test del classificatore PFV con campioni non estranei.

5.3.2 Campioni estranei

Nella seconda prova si è voluto verificare il comportamento del classificatore nel caso in cui sia richiesto di classificare campioni estranei, diversi dagli oggetti utilizzati per il training. La prova consiste nell'applicare la classificazione ad un'immagine contenente oggetti estranei. È stata analizzata la classe associata al campione estraneo ed i relativi classification ed identification score attribuiti. In Figura 5.8 sono riportati gli oggetti estranei utilizzati per la prova. In Figura 5.9 è riportata l'immagine utilizzata per la classificazione in cui è presente, oltre a tutti i campioni estranei, il campione di una vite appartenente alla categoria non estranei.

Dai risultati ottenuti (Tabelle 5.30 e 5.31) si evidenzia che a tutti i campioni estranei che hanno caratteristiche di forma simili a quelle rappresentative di una classe, viene associata tale classe, come si può vedere dalle classi assegnate alle ruote dentate. Con entrambi i metodi il classification score associato ai campioni estranei con caratteristiche simili agli oggetti di training, assume valori prossimi ai valori medi di classification score associati nella prova precedente dei campioni non estranei. Ad esempio nel caso del campione estraneo Corona Dentata a cui gli viene assegnata la classe Ruota Dentata con un classification score medio di 815 tra i due metodi, e si consideri che il classification score medio della ruota dentata nell'esempio precedente dei campioni non estranei è mediamente pari a 770 ; quindi, ponendo una soglia minima di sbarramento del classification score di poco inferiore ai valori medi ottenuti nella fase di test, è possibile considerare tali campioni estranei come classificati correttamente. Lo stesso ragionamento non vale per l'identification score da cui non è possibile ricavare alcuna informazione utile per lo scopo della prova.

In conclusione si può dire che, se si vuole effettuare una macro-identificazione il classificatore ottimale, per far sì che questo associ ad un campione sconosciuto la classe che più lo rappresenta, è il classificatore Multiclasse Generale.

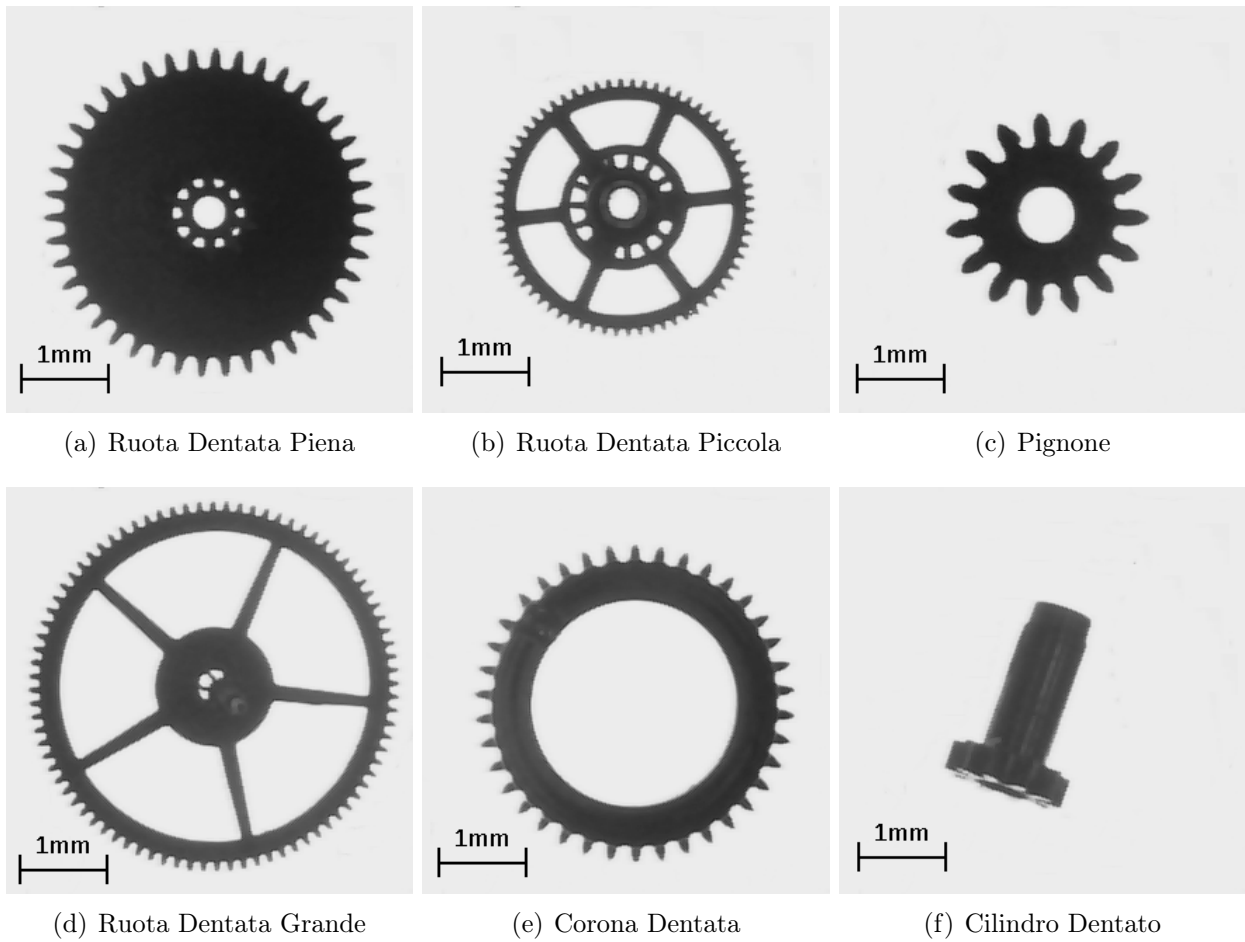


Figura 5.8: Oggetti estranei.

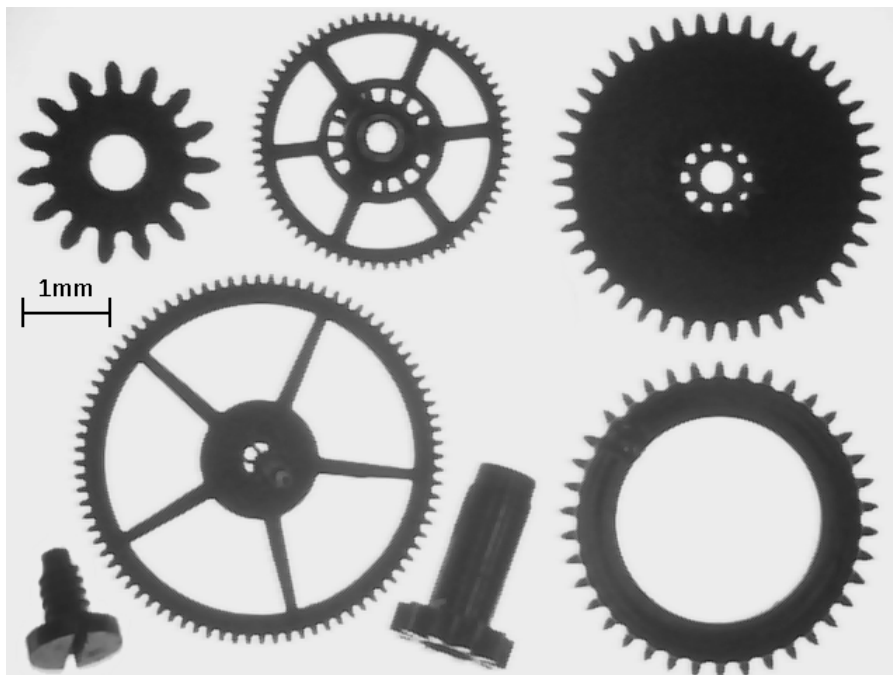


Figura 5.9: Immagine utilizzata per la classificazione.

Campione	Classe assegnata	Score	
		Classific.	Identific.
Ruota Dentata Alleggerita Grande	Ruota Dentata	862.74	760.934
Ruota Dentata Alleggerita Piccola	Ruota Dentata	887.542	836.526
Ruota Dentata Non Alleggerita	Ruota Dentata	759.613	817.513
Corona Dentata	Ruota Dentata	803.733	706.2
Pignone	Ruota Dentata	605.854	627.773
Cilindro dentato	Vite	590.403	717.515
Vite	Vite	932.072	960.625

Tabella 5.30: Risultati della fase di classificazione del classificatore NI con campioni stranieri.

Campione	Classe assegnata	Score	
		Classific.	Identific.
Ruota Dentata Alleggerita Grande	Ruota Dentata	838.64	774.937
Ruota Dentata Alleggerita Piccola	Ruota Dentata	856.14	857.43
Ruota Dentata Non Alleggerita	Ruota Dentata	723.25	753.442
Corona Dentata	Ruota Dentata	825.7	750.442
Pignone	Ruota Dentata	733.35	687.418
Cilindro dentato	Vite	758.19	788.19
Vite	Vite	952.85	974.408

Tabella 5.31: Risultati della fase di classificazione del classificatore PFV con campioni stranieri.

5.4 Conclusioni sui classificatori Multiclasse

Dagli esempi dei classificatori Multiclasse che si sono analizzati si giunge alla conclusione che per valutare se la classe associata dal classificatore al campione è corretta o meno bisogna valutare il classification score. Si definisce un valore minimo di score tale per cui la classificazione è considerata corretta. Come già detto nel Capitolo 3.3.3, per individuare il valore minimo di soglia si studia la distribuzione del classification score ottenuta dopo aver effettuato una fase di test con numerosi campioni. La soglia minima viene scelta in modo che buona parte dei campioni di test classificati sia considerato corretto, senza però ridurre di troppo il limite che porterebbe a considerare buone anche delle classificazioni errate. Mentre per l'identification score, dato che calcola la somiglianza del campione alla classe assegnata, nel caso di classificazione Multiclasse non ha utilità.

5.5 Esempio di classificatore Monoclasse

L'attenzione ora passa ai classificatori di tipo Monoclasse, sia con il metodo NI che con il metodo PFV. Questi tipi di classificatori hanno la caratteristica di essere costituiti da una sola classe, generata con un solo tipo di oggetto. In questo esempio si è voluto verificare se il classificatore in esame può essere impiegato nelle applicazioni di ispezione di oggetti, che verificano la conformità o meno di un oggetto. Per effettuare le prove sono stati scelti degli oggetti (Figura 5.11) e per ciascuno di essi è stato generato un classificatore Monoclasse. Per verificare il comportamento di ciascun classificatore è stata effettuata la classificazione di un insieme di oggetti. L'insieme è costituito in parte dagli stessi oggetti con cui si sono creati i classificatori (Figura 5.11), in parte da una serie di oggetti diversi da quelli usati nella fase di test (Figura 5.10) ma che presentano in alcuni casi caratteristiche comuni con quest'ultimi, pertanto possono essere definiti come campioni estranei. Per effettuare la prova tutti gli oggetti sono presenti in un'unica immagine (Figura 5.13). Nelle Tabelle 5.32, 5.33 e 5.34 sono riportati i parametri applicati ai classificatori. I risultati della fase di training sono riportati nelle Tabelle 5.35 e 5.36.

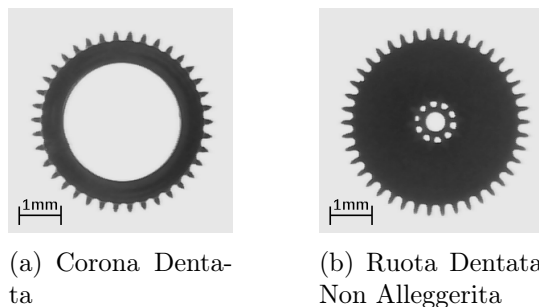


Figura 5.10: Oggetti utilizzati per la classificazione (estranei).

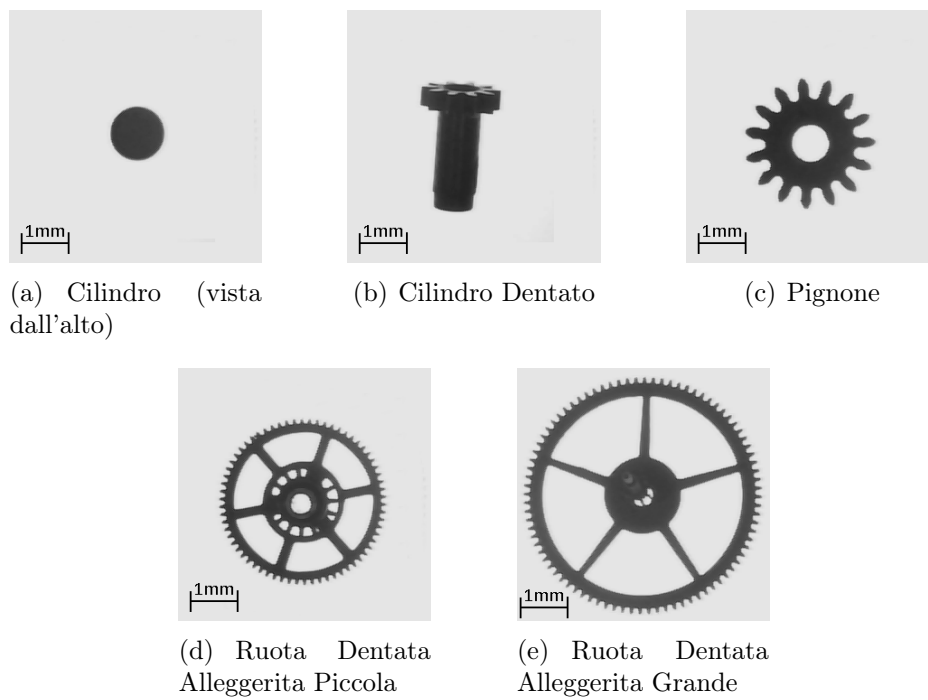


Figura 5.11: Oggetti utilizzati per creare i classificatori (non estranei).



Figura 5.12: Immagine contenente tutti gli oggetti a cui è stata applicata la classificazione monoclasse.

Threshold	Tipo	Manuale
	Tipo di oggetto	Dark
	Valore (0-255)	NI:148 PFV:109
Reject border	Applicato	
Numero di iterazioni di erosione	1	

Tabella 5.32: Valori di image processing applicati in tutti i classificatori Monoclasse.

Method	Nearest Neighbor
Distance Metric	Sum
k	1

Tabella 5.33: Method e Distance Metric applicati in tutti i classificatori Monoclasse.

Morphologic Transformation Type	Tipo	Chiusura
	Size	3
	n. di iter.	2

Tabella 5.34: Morphologic Transformation Type applicata ai classificatori PFV Monoclasse.

Dai risultati ottenuti (Tabelle [5.37](#) e [5.38](#)) si nota subito che il classification score associato da tutti i classificatori, sia NI che PFV, a tutte le classi; è pari a 1000: questo fattore quindi non può essere utile al fine di decretare la conformità di un campione. Si ottiene tale valore di classification score perché per definizione questo indice confronta la prima e la seconda classe che rappresentano di più il campione, nel momento in cui viene a mancare la seconda classe all'indice viene associato il massimo calore che può assumere. Il medesimo risultato si è ottenuto anche nell'esempio dei classificatori monoclasse scale dependent Capitolo [5.5](#). Per quanto riguarda l'identification score sono stati ottenuti risultati utili per la valutazione della conformità del campione: nel caso di applicazione del metodo NI, i classificatori monoclasse assegnano un identification score superiore a 900 al campione che rappresenta l'oggetto con cui il classificatore è stato istruito. Al contrario, a tutti i campioni che rappresentano oggetti diversi da quello utilizzato per il training del classificatore, viene assegnato un identification score inferiore a 600. Lo stesso vale per i classificatori PFV, con l'unica differenza che l'identification score associato agli oggetti rappresentativi della classe è superiore a 800, mentre per i restanti oggetti è pari a 0. Quest'ultimo valore appena illustrato deriva dal fatto la distanza vettoriale tra il campione da classificare e il campione di training più vicino è maggiore del fattore di normalizzazione e il loro rapporto ha quindi un valore maggiore di 1. Inserendo quindi tale rapporto nella formula di calcolo dell'identification score si otterrebbe un numero negativo che viene approssimato a 0.

Con questo esempio è stato dimostrato come il classificatore monoclasse possa essere utilizzato per applicazioni di riconoscimento della conformità di un oggetto. Nel caso NI la differenza del classification score tra campione conforme e non è circa 300, mentre nel caso PFV è circa pari a 900. Si evidenzia, inoltre, che i campioni estranei permettono di dimostrare che il classificatore monoclasse, sia NI che PFV, non assegna valori elevati, oltre a 800, di identification score a quei campioni che presentano caratteristiche comuni con l'oggetto rappresentato dalla classe. Ad esempio il classificatore monoclasse NI Ruota Dentata Alleggerita Grande associa al campione estraneo Corona Dentata un Identification score pari a 703. Si giunge quindi alla conclusione che, nel caso dei classificatori monoclasse, non va valutato il classification score,

bensì l'identification score che, attraverso la definizione di un valore di soglia, permette di dichiarare un campione conforme o meno.

Classi	Ruota Dentata Alleggerita Grande	Ruota Dentata Alleggerita Piccola	Pignone	Cilindro Dentato	Cilindro
Numero di campioni di training	5	3	3	5	3
Deviazione standard massima	492.16	513.94	189.1	266.32	56.96
Scale factor	500				
Mirror factor	0				

Tabella 5.35: Caratteristiche di training dei classificatori NI Monoclasse.

Classi	Ruota Dentata Alleggerita Grande	Ruota Dentata Alleggerita Piccola	Pignone	Cilindro Dentato	Cilindro
Numero di campioni di training	5	3	3	5	3
Deviazione standard massima	0.89	0.488	0.469	1.21	0.22
Feature con dev. std. max	Scale Area Ratio	Scale Area Ratio	Scale Area Ratio	Scale Area Ratio	Scale Area Ratio
Valore medio feature con dev. std. max	164.087	79.67	35.563	22.645	8.648
Feature selezionate	Number of Hole, Convex Perimeter Ratio, Heywood Circularity Factor, Elongation Factor, Hole's Area Ratio e Scale Area Ratio				

Tabella 5.36: Caratteristiche di training dei classificatori PFV Monoclasse.

Classificatore	Ruota Dentata Alleggerita Grande		Ruota Dentata Alleggerita Piccola		Pignone	
Campione\Score	Classific.	Identific.	Classific.	Identific.	Classific.	Identific.
Ruota Dentata Alleggerita Grande	1000	973.508	1000	508.112	1000	244.603
Ruota Dentata Alleggerita Piccola	1000	526.536	1000	981.043	1000	393.479
Pignone	1000	243.816	1000	394.615	1000	976.727
Cilindro Dentato	1000	171.622	1000	277.769	1000	360.556
Cilindro	1000	65.768	1000	106.445	1000	138.17
Corona Dentata	1000	703.169	1000	501.685	1000	288.974
Ruota Dentata Non Alleggerita	1000	302.96	1000	192.433	1000	404.778

Classificatore	Cilindro Dentato		Cilindro	
Campione\Score	Classific.	Identific.	Classific.	Identific.
Ruota Dentata Alleggerita Grande	1000	184.392	1000	68.181
Ruota Dentata Alleggerita Piccola	1000	296.62	1000	109.678
Pignone	1000	283.181	1000	141.685
Cilindro Dentato	1000	919.531	1000	201.286
Cilindro	1000	194.548	1000	998.212
Corona Dentata	1000	217.84	1000	80.548
Ruota Dentata Non Alleggerita	1000	106.585	1000	39.411

Tabella 5.37: Risultati della fase di classificazione dei classificatori NI Monoclasse.

Classificatore	Ruota Dentata Alleggerita Grande		Ruota Dentata Alleggerita Piccola		Pignone	
	Campione\Score	Classific.	Identific.	Classific.	Identific.	Classific.
Ruota Dentata Alleggerita Grande	1000	827.96	1000	0	1000	0
Ruota Dentata Alleggerita Piccola	1000	0	1000	890.505	1000	0
Pignone	1000	0	1000	0	1000	896.278
Cilindro Dentato	1000	0	1000	0	1000	0
Cilindro	1000	0	1000	0	1000	0
Corona Dentata	1000	0	1000	0	1000	0
Ruota Dentata Non Alleggerita	1000	0	1000	0	1000	0

Classificatore	Cilindro Dentato		Cilindro	
	Campione\Score	Classific.	Identific.	Classific.
Ruota Dentata Alleggerita Grande	1000	0	1000	0
Ruota Dentata Alleggerita Piccola	1000	0	1000	0
Pignone	1000	0	1000	0
Cilindro Dentato	1000	959.784	1000	0
Cilindro	1000	0	1000	981.25
Corona Dentata	1000	0	1000	0
Ruota Dentata Non Alleggerita	1000	0	1000	0

Tabella 5.38: Risultati della fase di classificazione dei classificatori PFV Monoclasse.

5.6 Esempio di classificatore Multiclasse Mirror Dependent

L'ultima caratteristica non ancora analizzata attribuibile a un classificatore, è la sensibilità alla riflessione speculare (Mirror Dependence). Tale caratteristica è considerabile con il metodo NI, mentre con il metodo PFV non è possibile dato che non è stata analizzata nessuna caratteristica che evidenzia tale peculiarità. Al momento della selezione dei parametri del classificatore per far sì che questo sia indipendente dalla riflessione speculare si imposta il valore del Mirror Factor a 0, in caso contrario, si sceglie un valore compreso tra 0 e 1000. Il Mirror factor indica il peso con cui il fattore di riflessione speculare viene considerato dal classificatore. Per verificare la differenza tra l'applicare o meno la dipendenza dalla riflessione speculare sono stati generati due classificatori, uno Mirror Dependent e uno Mirror Independent. Per la generazione delle classi sono stati scelti due oggetti asimmetrici, un gancio e una leva, per ognuno dei quali sono state generate due classi che identificano due posizioni dell'oggetto, come si può vedere in Figura 5.13. I parametri dei classificatori selezionati sono riportati nelle Tabelle 5.39, e 5.40 ed le impostazioni della fase di training sono riportati nella Tabelle 5.41, 5.42. Per verificare il funzionamento dei due tipi di classificatori è stata effettuata una fase di test, per ciascun classificatore, con una serie di campioni raffiguranti i due oggetti con differenti orientamenti e posizioni. La scelta di questo tipo di test è stata fatta per poter verificare se il classificatore identifica la posizione corretta in cui si trova l'oggetto.

I risultati ottenuti nel caso del classificatore Mirror Independent (Tabella 5.43) hanno permesso di riscontrare l'incapacità del classificatore di riconoscere la classe che indica la posizione del campione. Tale affermazione deriva dal fatto che l'accuratezza totale del campione è pari al 55% e che i valori di prevedibilità sono al massimo del 60%. Infine, i valori medi di classification score sono inferiori a 600. Nel caso del classificatore Mirror Dependent si ottengono invece ottimi risultati. Per prima cosa l'accuratezza totale è pari al 95% e i valori di prevedibilità sono superiori all'83% per tutte le classi. Non si sono ottenuti dei valori di accuratezza e prevedibilità pari al 100% in quanto le immagini degli oggetti, al momento della binarizzazione, presentavano dei difetti che hanno portato ad una leggera degradazione della qualità delle misurazioni

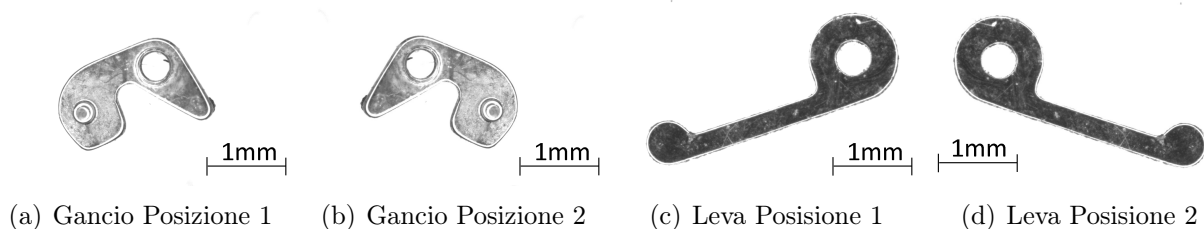


Figura 5.13: Oggetti rappresentanti le classi generate.

geometriche effettuate. Infine, i classification score medi assumono valori superiori a 850. In conclusione, si può dire che l'utilizzo del Mirror factor è ottimo per riconoscere l'orientamento in cui si trova l'oggetto, tuttavia può anche essere utilizzato per identificare componenti che presentano le stesse caratteristiche geometriche ma speculari. Si fa notare che, nell'esempio illustrato, il classificatore Mirror Dependent ha un Mirror Factor pari a 600 poiché non si è voluto vincolare il classificatore alla sola caratteristica della riflessione speculare. Si ricorda inoltre che, come nel caso dello Scale Factor, selezionare un valore di Mirror Factor pari a 1000 porterebbe ad un degrado delle prestazioni del classificatore in quanto si rende il classificatore completamente dipendente dalla riflessione speculare.

Threshold	Tipo	Manuale
	Tipo di oggetto	Dark
	Valore (0-255)	250.552
Reject border	Applicato	
Numero di iterazioni di erosione	4	

Tabella 5.39: Valori di image processing applicati a tutti i classificatori Mirror Dependent ed Independent.

Method	Nearest Neighbor
Distance Metric	Sum
k	1

Tabella 5.40: Method e Distance Metric applicati a tutti i classificatori Mirror Dependent ed Independent.

Classi	Gancio Pos.1	Gancio Pos.2	Leva Pos.1	Leva Pos.2	Totale campioni
Numero di campioni di training	5	5	5	5	20
Deviazione standard massima	0.15	0.13	0.12	0.1	
Scale factor	0				
Mirror factor	0				

Tabella 5.41: Caratteristiche di training del classificatore NI Multiclasse Mirror Independent.

Classi	Gancio Pos.1	Gancio Pos.2	Leva Pos.1	Leva Pos.2	Totale campioni
Numero di campioni di training	5	5	5	5	20
Deviazione standard massima	1.58	1.91	0.37	0.27	
Scale factor	0				
Mirror factor	600				

Tabella 5.42: Caratteristiche di training del classificatore NI Multiclasse Mirror Dependent.

Classi	Gancio Pos.1	Gancio Pos.2	Leva Pos.1	Leva Pos.2	Totale campioni
Numero di campioni di training	5	5	5	5	20
Media del classification score	488	575	440.5	469	
Accuracy	60	60	40	60	Acc. Tot. 55
Predictability	60	60	50	50	

Tabella 5.43: Risultati della fase di test del classificatore NI Mirror Independent

Classi	Gancio Pos.1	Gancio Pos.2	Leva Pos.1	Leva Pos.2	Totale campioni
Numero di campioni di training	5	5	5	5	20
Media del classification score	926.2	852.2	929.75	883.4	
Accuracy	100	100	80	100	Acc. Tot. 95
Predictability	100	100	100	83.33	

Tabella 5.44: Risultati della fase di test del classificatore NI Mirror Dependent

5.7 Esempio di classificazione On-line

Tutte le prove precedenti di classificazione sono state effettuate con immagini statiche. Si va quindi ora ad analizzare il comportamento di tutti i tipi classificatori nel caso di sorgenti dinamiche. In questo esempio sono stati generati differenti tipi di classificatori per la classificazione di componenti in acquisizione on-line con un sistema di visione. I classificatori sono stati applicati a due sistemi di visione: il primo configurato con un illuminatore back light, il secondo con illuminatore front light (Capitolo [3.4.1](#)). Il secondo sistema di visione è stato utilizzato al solo scopo di verificare se la classificazione restituisce risultati soddisfacenti anche utilizzando illuminatori che non risaltano solo il profilo dell'oggetto. Per ciascun sistema di visione è stato utilizzato un insieme di oggetti. Per il sistema di visione con back light sono stati utilizzati gli oggetti riportati in Figura [5.14](#) e sono stati generati i seguenti classificatori:

- Classificatore Multiclasse Specifico costituito da una classe per ciascun oggetto;
- Classificatore Multiclasse Generale costituito da due classi rappresentanti le categorie di oggetti Asta e Disco;
- Classificatore Monoclasse costituito da una classe rappresentante l'oggetto Asta Piccola.

Per quanto riguarda il sistema di visione configurato con front light sono stati utilizzati dei frammenti di fibra ottica di lunghezza variabile (Figura [5.15](#)), e delle sfere con diametro medio di circa 0.8mm (Figura [5.16](#)). Con questi oggetti sono stati generati i seguenti classificatori:

- Classificatore Multiclasse costituito dalle classi Fibra Ottica e Sfera;
- Classificatore Monoclasse costituito dalla classe Sfera.

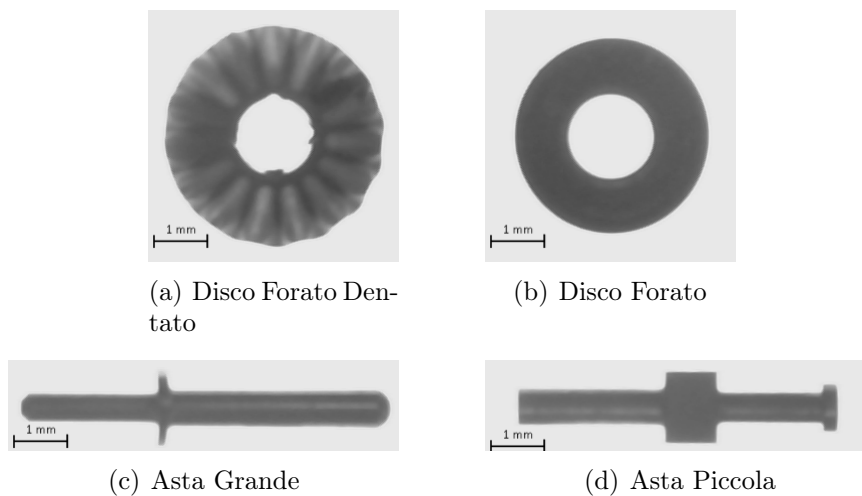


Figura 5.14: Oggetti utilizzati per la classificazione illuminati con back light.

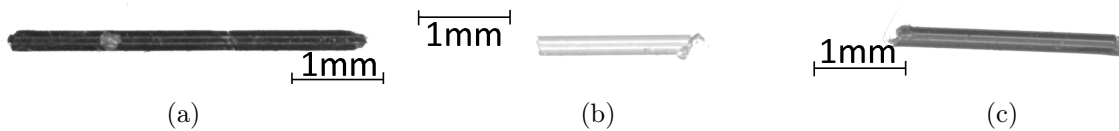


Figura 5.15: Frammenti di fibra ottica illuminato con front light.

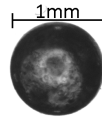


Figura 5.16: Sfera illuminata con front light.

Tutti i classificatori sono stati generati sia con metodo NI che PFV. A tutti i classificatori che utilizzano le immagini di certo sistema di visione sono stati applicati gli stessi parametri di image processing riportati in Tabella 5.45 e 5.46. I parametri Method e Distance Metric sono gli stessi per tutti i classificatori generati (Tabella 5.47). Infine, per i classificatori di tipo PFV, nei casi in cui si utilizzi l'illuminatore back light non sono state applicate trasformazioni morfologiche binare. Al contrario, nel caso di utilizzo dell'illuminatore front light è stata applicata la trasformazione morfologica binaria con i parametri riportati in Tabella 5.48.

I risultati ottenuti con tutti i classificatori sono soddisfacenti. A titolo dimostrativo sono state riportate delle istantanee rappresentanti i risultati della classificazione effettuata con una sorgente dinamica (Figure 5.17, 5.18, 5.19, 5.20 e 5.21). Dall'analisi dei risultati ottenuti, si nota che il classification e l'identification score variano leggermente. Tale fenomeno è causato dalle piccole variazioni d'intensità dei pixel di bordo degli oggetti. Queste piccole variazioni al momento dell'applicazione di sogliatura si trasformano in continui cambiamenti di stato (0 o 1) dei pixel binarizzati. Il numero di pixel soggetti a questa variazione aumentano al diminuire della nitidezza dei bordi dell'oggetto ripreso e alla variazioni istantanee dell'illuminazione. Il fenomeno appena spiegato comporta una variazione della forma dell'oggetto più o meno pronunciata in base alla nitidezza dei suoi bordi. La continua variazione della forma causa, di conseguenza, la continua variazione delle misurazioni effettuate per l'estrazione delle caratteristiche geometriche, che a loro volta causano la continua variazione del classification e dell'identification score. L'ampiezza delle variazioni riscontrate negli score delle prove sono al

Threshold	Tipo	Manuale
	Tipo di oggetto	Dark
	Valore (0-255)	201.628
Reject border	Applicato	
Numero di iterazioni di erosione	2	

Tabella 5.45: Valori di image processing applicati in tutti i classificatori che utilizzano le immagini del sistema di visione con illuminatore back light.

Threshold	Tipo	Manuale
	Tipo di oggetto	Dark
	Valore (0-255)	242.79
Reject border	Applicato	
Numero di iterazioni di erosione	5	

Tabella 5.46: Valori di image processing applicati in tutti i classificatori che utilizzano le immagini del sistema di visione con illuminatore front light.

Method	Nearest Neighbor
Distance Metric	Sum
k	1

Tabella 5.47: Method e Distance Metric applicati in tutti i classificatori dell'esempio On-line.

Morphologic Transformation Type	Tipo	Chiusura
	Size	3
	n. di iter.	3

Tabella 5.48: Morphologic Transformation Type applicate ai classificatori PFV che utilizzano le immagini del sistema di visione con front light.

massimo pari a 100 nel caso peggiore. Per far sì che questo fenomeno non influenzi negativamente la scelta dei campioni correttamente classificati si può tarare il limite di classification o identification score ad un valore tale che questo comprenda le oscillazioni dello score; tuttavia tale scelta può rivelarsi pericolosa, poiché se venisse scelto un valore di soglia troppo basso potrebbero essere considerati buoni alcuni campioni classificati in modo non corretto. Una seconda soluzione potrebbe essere porre un filtro ai dati dell'immagine per eliminare almeno in parte le variazioni e ottenere un'immagine più stabile che potrà ad avere score più stabili, ma è necessario comunque prestare attenzione in quanto applicare dei filtri all'immagine fornita dal sistema di visione rende il software meno reattivo ai cambiamenti nel campo visivo del sistema di visione.

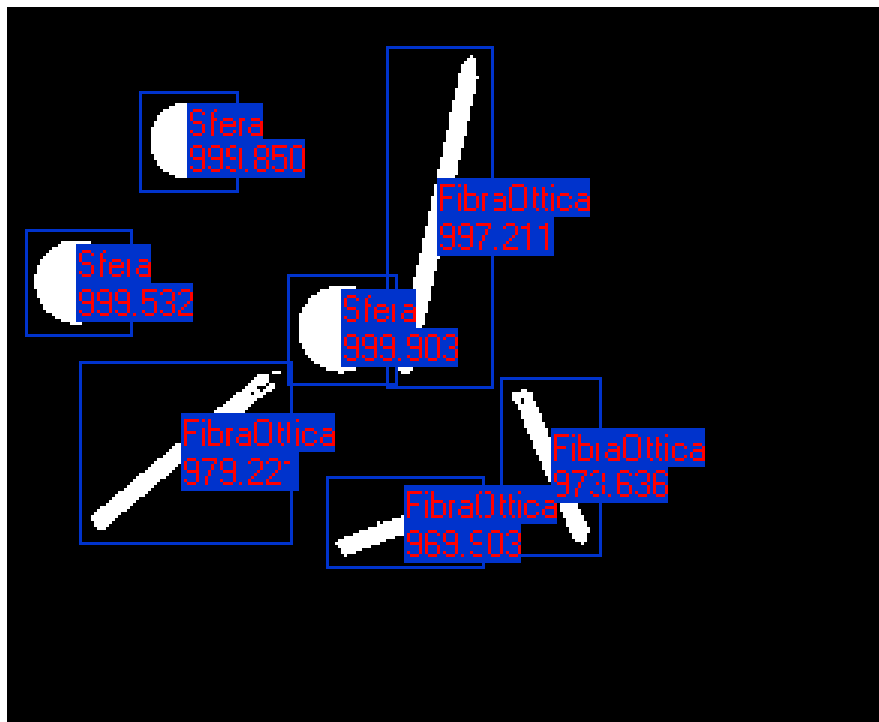


Figura 5.17: Esempio con classificatore Multiclasse Specifico PFV (illuminatore front light).

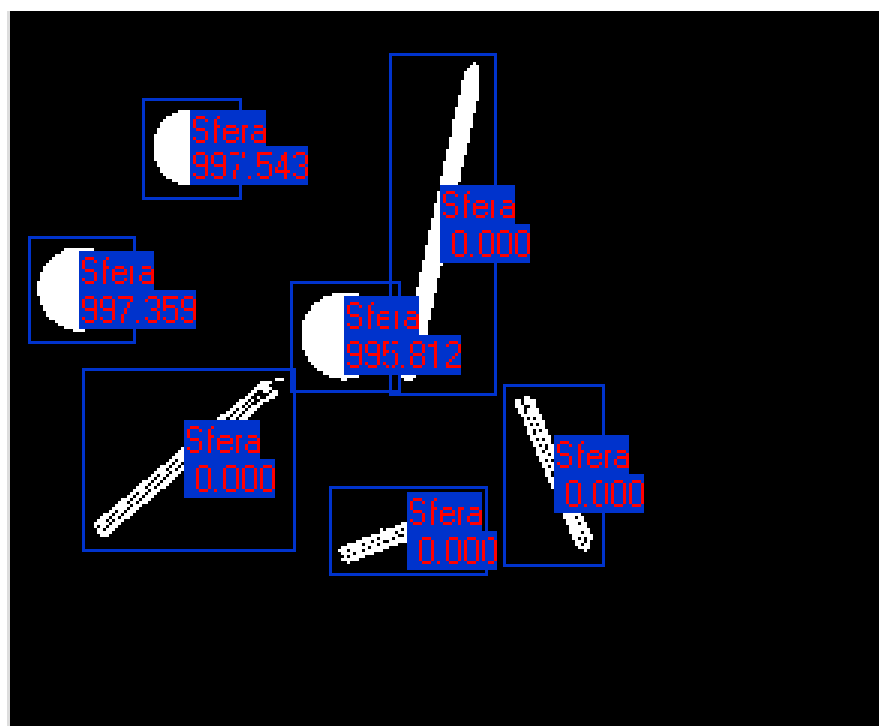


Figura 5.18: Esempio con classificatore Monoclasse NI (illuminatore front light).

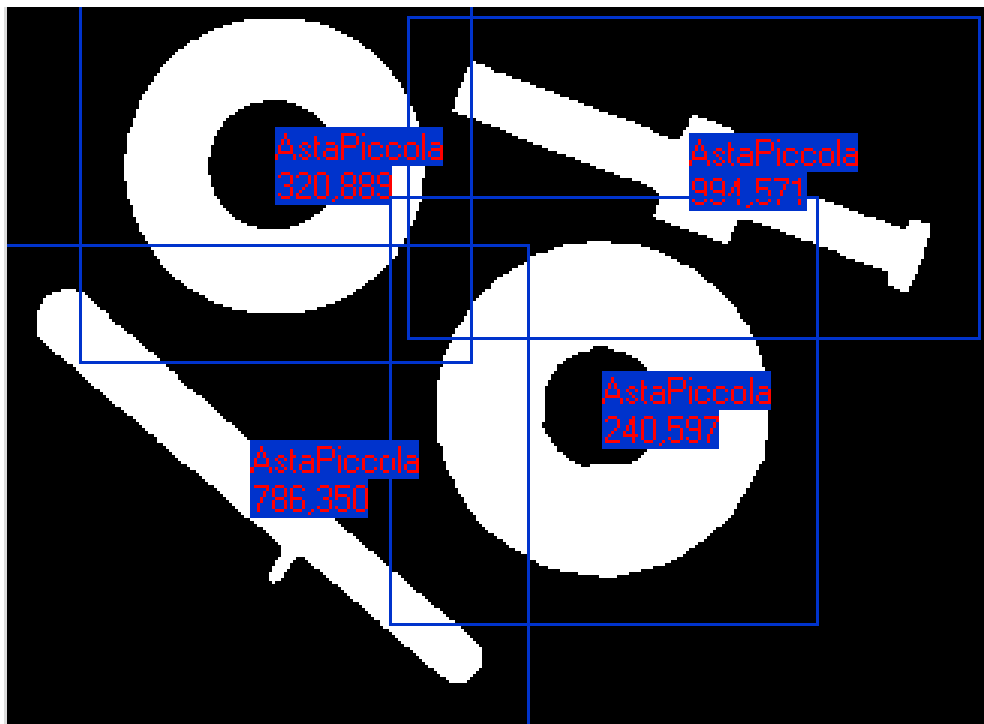


Figura 5.19: Esempio con classificatore Monoclasse NI (illuminatore back light).

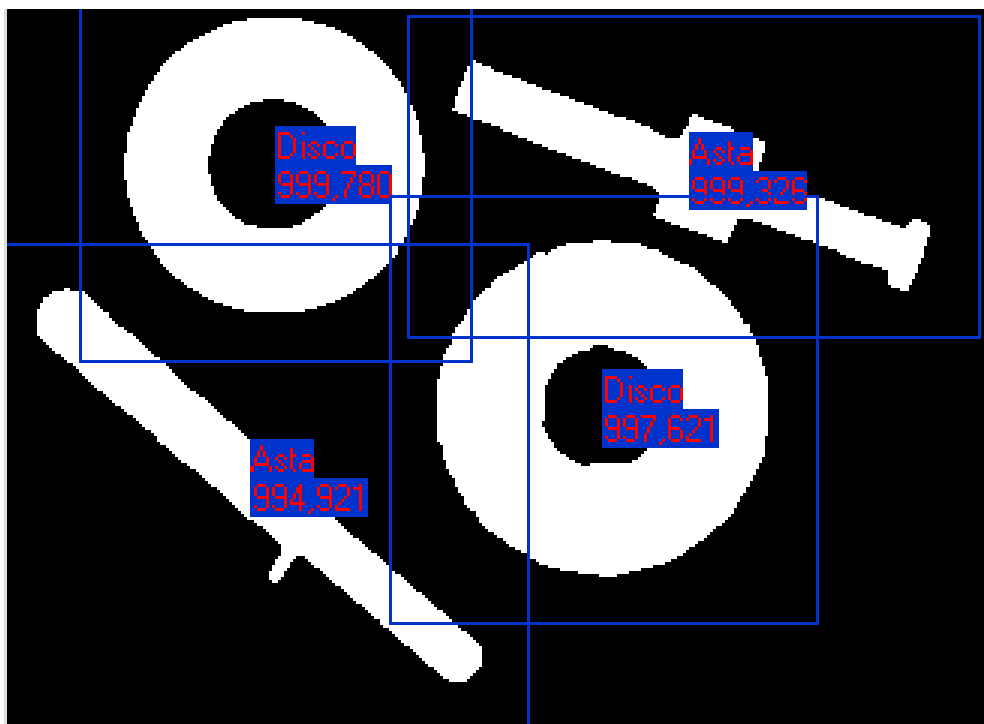


Figura 5.20: Esempio con classificatore Multiclasse Generale PFV (illuminatore back light).

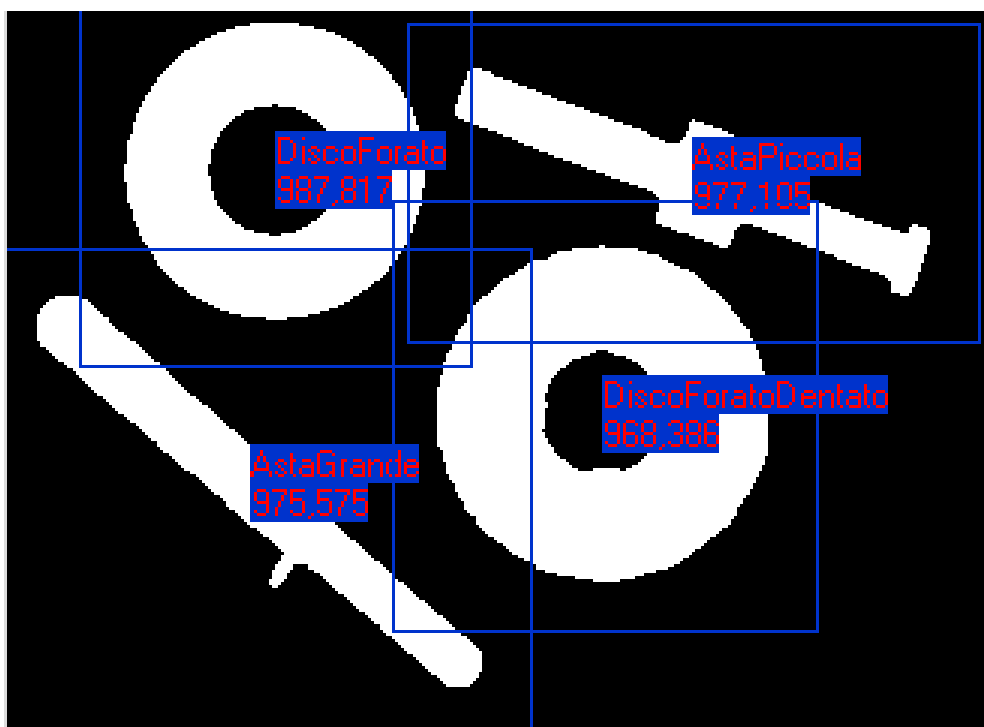


Figura 5.21: Esempio con classificatore Multiclasse Specifico NI (illuminatore back light).

Conclusioni

In questo lavoro di tesi è stato studiato ed implementato un algoritmo di machine vision con il relativo software in grado di effettuare la classificazione di particelle in base a fattori di forma, estraendo le informazioni da immagini in scala di grigi. In particolare sono stati sviluppati due tipi di classificatori, sfruttando le librerie di visione di LABView della NI: il Particle Classifier NI e Custom Particle Classifier PFV.

L'algoritmo sviluppato è stato testato per verificare le sue performance in diversi casi d'uso. Sono state infatti effettuate prove con differenti oggetti, condizioni operative (ad esempio configurazioni del sistema di visione 2D) e tipi di classificatore (ad esempio Multiclasse Specifici o Ggenerali, Monoclasse).

Dai risultati delle prove effettuate si può dire che la Classificazione è un efficace algoritmo di machine vision. I risultati della classificazione di oggetti di dimensione inferiore al millimetro, con i metodi Particle Classifier NI e con il Custom Particle Classifier PFV, hanno permesso di evidenziare in parte le potenzialità e le criticità della strategia.

Non sono state evidenziate particolari differenze sulle prestazioni dei due metodi di classificazione (NI e PFV). Si fa notare però che esistono alcune differenze significative legate a: elaborazione dell'immagine ed estrazioni e valutazione delle caratteristiche misurate.

Nel caso del metodo NI, non è possibile sapere il tipo e il valore delle misurazioni geometriche effettuate, è quindi impossibile effettuare una valutazione approfondita del classificatore al momento della fase di training. Ciò comporta che eventuali errori nella fase di training verranno rilevati durante la fase di test o durante la classificazione. Uno dei vantaggi del classificatore NI è la possibilità di considerare le caratteristiche di dimensione e di riflessione speculare attraverso la selezione degli Scale Factor e Mirror Factor.

A valle di queste considerazioni si può dire che il metodo NI ha buone prestazioni con tutti i tipi di classificatori analizzati nei test effettuati (Multiclasse Specifico, Multiclasse Generale e Monoclasse).

Nel caso dei classificatori PFV è possibile selezionare le caratteristiche geometriche da estrarre dal campione. Le misurazioni effettuate possono essere visionate e durante la fase di training è possibile inoltre valutare la media e la deviazione standard di ciascuna feature dei campioni associati alle classi. Tutto ciò permette di effettuare una valutazione più approfondita del clas-

sificatore durante la fase di training, permettendo di intervenire sul classificatore prima che il training venga terminato e non si possano più gestire i campioni associati o cambiare le feature scelte. Si ha quindi un maggior controllo dei dati raccolti a differenza del metodo NI. Tuttavia uno dei limiti di questo metodo è che i risultati che si ottengono sono dipendenti dalle feature selezionate. Se non si scelgono le caratteristiche adeguate le prestazioni del classificatore non sono soddisfacenti. Infine, a differenza del metodo NI, non è possibile selezionare una caratteristica che permetta di rilevare la riflessione speculare.

Alla fine di queste considerazioni si può dire che il metodo PFV ha buone prestazioni con tutti i tipi di classificatori analizzati nella fase di test, come per il metodo NI.

Per poter considerare corretta l'identificazione di un componente, con entrambi i metodi, si impone una soglia minima di classification score in caso di classificatori Multiclasse, o di identification score nel caso di classificatori Monoclasse.

Tutti i test effettuati in questo lavoro hanno utilizzato lo stesso tipo di metodo di selezione della classe da assegnare e lo stesso metodo di calcolo della distanza tra campione da identificare e di training. Per questo motivo in futuro potrebbe essere utile approfondire maggiormente l'analisi degli effetti delle variazioni di questi due parametri sui risultati finali.

Uno sviluppo futuro di questa tesi, con riferimento al metodo PFV, potrebbe essere l'applicazione di un sistema di apprendimento (per esempio rete neurale), da utilizzare alla fine della fase di training, per individuare una serie di pesi con cui considerare le caratteristiche geometriche misurabili. Tale sviluppo permetterebbe di sollevare l'utente dalla scelta delle feature più rappresentative per gli oggetti da classificare.

Un ulteriore sviluppo futuro include la ricerca di ulteriori caratteristiche geometriche utili per caratterizzare un oggetto, e in particolare potrebbe focalizzarsi sulla ricerca di caratteristiche che permettono di rilevare la riflessione speculare.

Infine il metodo può essere esteso anche per sistemi di visione a colori, considerando oltre alle caratteristiche di forma anche le caratteristiche di colore.

Ringraziamenti

Un ringraziamento a Irene Fassi per avermi permesso di fare questa esperienza nei laboratori di MEDIS di ITIA-CNR Milano.

Un ringraziamento particolare va a Serena Ruggeri e Gianmauro Fontana per avermi accolto nel loro gruppo e avermi trasmesso la loro conoscenza e guidato per la realizzazione di questa tesi.

Voglio ringraziare mio papà e mia mamma che mi hanno sostenuto in questi anni di studio spronandomi a dare il meglio.

Ringrazio i miei famigliari, i miei amici e i miei compagni di squadra per avermi supportato e sopportato sempre in qualsiasi momento.

Voglio ringraziare anche i miei compagni di corso con cui ho condiviso questo percorso di vita e con cui sono cresciuto.

Bene se mi fossi dimenticato di qualcuno non me ne voglia, ma se mi conosce bene sa che la mia memoria è più che breve.

Stavo scherzando, come facevo a dimenticarmi di lei che mi ha aiutato sempre e comunque sacrificando il suo tempo in molte occasioni.

Grazie Vale perché tu c'eri, ci sei e ci sarai.

Bibliografia

- [1] URL: <https://de.wikipedia.org/wiki/Halcon>.
- [2] URL: <https://www.mathworks.com/products/computer-vision.html>.
- [3] URL: <https://it.wikipedia.org/wiki/MATLAB>.
- [4] URL: www.treccani.it/vocabolario.
- [5] URL: http://www.scholarpedia.org/article/Local_Binary_Pattern.
- [6] URL: https://it.wikipedia.org/wiki/Ray_tracing.
- [7] URL: <https://it.wikipedia.org/wiki/LabVIEW>.
- [8] URL: https://en.wikipedia.org/wiki/Harris_Corner_Detector.
- [9] Schacklock A e Sun W. «Integrating microscope and perspective views». In: *International Conference on Robotis and Automation (ICRA '05)*.
- [10] Tamadazte B. et al. «CAD Model-based Tracking and 3D Visual-based Control for MEMS Microassembly». In: *The International Journal of Robotics Research* (2010).
- [11] Jalil Bin Lias. «Analysis Of Red Blood Cell (RBC) Classification Using NI Vision Builder AI». Tesi di laurea mag. Faculty of Electrical e Electronic Engineering University Tun Hussein Onn Malaysia, 2015.
- [12] Tamadazte Brahim, Le-Fort Piat Nadine e Demélé Sounkalo. «Robotic Micromanipulation and Microassembly Using Monoview and Multiscale Visual Servoing». In: *IEEE/ASME TRANSACTIONS ON MECHATRONICS* (2011).
- [13] Enikov E e Nelson BJ. «Three-dimensional microfabrication for a multidegree-of-freedom capavitive force sensor using fibre-chip coupling». In: *Micromechatronics and Microengineering* (2000).
- [14] Klaver Eric. *Embendding components into PCBs*. Rapp. tecn. Embendding components into PCBs.
- [15] «Esperimental Robotics VI, Lecture Notes in Control and Infomation Sciences». In: Springer, 1999. Cap. Force and vision Feedback for robotic manipulation on the microword.

- [16] Shen Fei et al. «High-Precision Automated 3-D Assembly With Attitude Adjustment Performed by LMTI and Vision-Based Control». In: *IEEE/ASME TRANSACTIONS ON MECHATRONICS* (2015).
- [17] Yang G e Nelson BJ. «Wavelet-based autofocusing and unsupervised segmentation of microscopic images.» In: *Preceeding of the IEEE/RSJ Internation Conference on Intelligent Robots and System*.
- [18] Fontana Gianmauro. «Assembly at the Microscale: Design and Implementation of a Robotised Work-Cell». Tesi di dott. Università degli studi di Brescia, 2014.
- [19] Fontana Gianmauro et al. «FLEXIBLE VISION BASED CONTROL FOR MICRO-FACTORIES». In: *Proceedings of the 7th International Conference on Micro- and Nano-systems*.
- [20] Chen H et al. «CAD-guided manufacturing of nanostructures using nanoparticles.» In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System (IROS'04)*.
- [21] «Handbook of industrial Robotics». In: Wiley & Sons. Inc., 1999. Cap. Microassembly.
- [22] Wang Huaping et al. «Automated Assembly of Vascular-Like Microtube With Repetitive Single-Step Contact Manipulation». In: *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING* (2015).
- [23] «IEEE Transaction on Pattern Analysis and Machine Intelligence». In: 2000. Cap. A flesible new technique for camera calibration.
- [24] «IEEE Transactions on Electronics Packaging Manufacturing». In: 2000. Cap. Micropositioning of a weakly calibrated microassembly system using coarse-to-fine visual servoing strategies.
- [25] Fassi Irene. *Guida alle tecnologie di micro-manufacturing e alle loro applicazioni*. CNR Edizioni, 2016.
- [26] Cecil J, Powell D e Vasquez D. «Assembly and manupulation of micro devices-A state of the art survey.» In: *Robotics and Computer-Integrated Manufarturing* (2007).
- [27] Fatemeh Karimirad et al. «Modelling a Precision Loadcell using Neural Networks for Vision-Based Force Measurement in Cell Micromanipulation.» In: *International Conference on Advanced Intelligent Mechatronics (AIM)*.
- [28] «Lecture Notes in Computer Science, Springer». In: 2006. Cap. Trifocal trasfer based novel view synthesis for micromanipulation, Advances in Visul Computing.
- [29] «Machine Vision Algorithms and Application». In: WILEY-VCH, 2008. Cap. 2 Image Acquisition.

- [30] «Microrobotics for Micromanipulation». In: Wiley ISTE, 2010. Cap. Architecture of a Micromanipulation Station.
- [31] Matthew P. Edgara et al. «Real-time 3D video utilizing a compressed sensing time-of-flight single-pixel camera». In: *Optical Trapping and Optical Micromanipulation XIII* (2016).
- [32] Rougeot P et al. «Vision-based control of AFM-based micromanipulation.» In: *In Proc. of International Advanced Robotic Program*.
- [33] «Photogrammetric Engineering and Remote Sensing». In: Brown DC, 1986. Cap. Lens distortion for close-range photogrammetry.
- [34] Ruggeri S., Fontana G. e Fassi I. *Embedding Test PON2016*. Rapp. tecn. MEDIS, ITIA CNR, 2016.
- [35] Brahim Tamadazte et al. «Robotic micromanipulation for microassembly: modelling by sequential function chart and achievement by multiple scale visual servings». In: *J. Micro-Nano Mech* (2009).
- [36] *Una introduzione a OpenCV*.
- [37] *Vision Concepts Help*.
- [38] *Vision for LabVIEW Help*.
- [39] Lu Zhe et al. «A micromanipulation system with dynamic force-feedback for automatic batch microinjection». In: *JOURNAL OF MICROMECHANICS AND MICROENGINEERING* (2007).
- [40] Cenev Zoran, Venäläinen Janne e Zhou Quan. *Vision based event classification in robotic micromanipulation*. Rapp. tecn. Department of Electrical Engineering e Automation, School of Electrical Engineering, Aalto University. Espoo, Finland.
- [41] Cenev Zoran et al. *Object Tracking in Robotic Micromanipulation by Supervised Ensemble Learning Classifier*. Rapp. tecn. Department of Electrical Engineering e Automation, School of Electrical Engineering, Aalto University, Espoo, Finland, 2017.



UNIVERSITÀ DEGLI STUDI DI BRESCIA
UNIVERSIDAD DE ALMERÍA



UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

Resumen/Abstract

Now a days using vision system in industry allows innumerable activities such as: identification of flaws, measuring tolerance, robot's identification, contactless measure or object classification [1]. In this thesis, I analysed and implemented an algorithm that allows the identification of classes of objects to support the execution of microassembly tasks using 2D vision systems. In automatic manipulation, identification algorithms are fundamental for identifying, distinguishing, and locating components with different shapes and purpose in the work field. The information extracted from the images can guide the robot and make it more autonomous and versatile for various applications. The identification method analysed in this thesis is called Particle Classification [5]. It was implemented and used in the micromanipulation work-cell in the Laboratory of Micro-robotics of the Micro-Enabled Devices and Systems (MEDIS) group at the Intelligent Industrial Systems and Technologies for Advanced Manufacturing (STIIMA). of the National Research Council of Italy (STIIMA-CNR) in Milan (Figure 1). The algorithm is used in the main management program of the work-cell for the development of automatic manipulation tasks.

Doble Título UNIBS-UAL

Mechatronics for Industrial Automation

Ingegneria dell'Automazione Industriale

Grado en Ingeniería Electrónica Industrial

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL, 2017/2018