

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

DIPARTIMENTO DI INFORMATICA, SCIENZA E INGEGNERIA

TESI DI LAUREA

in
Elaborazione dell'Immagine M

**UN ALGORITMO DI VISIONE ARTIFICIALE PER
L'INDIVIDUAZIONE DELLO SWITCHING POINT NEL
PROCESSO DI ETICHETTATURA MARKERLESS**

CANDIDATO
Erica Barone

RELATORE
Chiar.mo Prof. Luigi Di Stefano

CORRELATORE
Michele Benedetti

Anno Accademico 2012/13

Sessione I

Indice

INTRODUZIONE	5
1. SCENARIO DI STUDIO.....	7
1.1. Il processo di confezionamento	7
1.2. Identificazione dello switching point.....	9
1.3. Il marker sensor	10
1.3.1. Principio di funzionamento	11
1.3.2. Caratteristiche dei sensori di contrasto	12
1.3.3. Problematiche dei sensori di contrasto	13
2. APPROCCIO ALTERNATIVO: UN SISTEMA DI VISIONE	17
2.1. Principio di funzionamento.....	17
2.2. I vantaggi del sistema di visione rispetto al sensore di contrasto ...	19
3. ALGORITMO A DUE FASI PER L'INDIVIDUAZIONE DELLO SWITCHING POINT	23
3.1. Fase 1: Teaching	24
3.1.1. Il procedimento di individuazione dei pattern	25
3.1.2. Analisi delle problematiche.....	30
3.1.3. Soluzioni proposte	33
3.2. Fase 2: Matching	34
3.2.1. Analisi delle problematiche.....	36
3.2.2. Soluzioni proposte	37
3.3. Controlli di integrità	40

4. TEST E RISULTATI SPERIMENTALI.....	43
4.1. Immagini acquisite tramite scanner	45
4.2. Immagini acquisite tramite webcam.....	51
5. ANALISI DEL SISTEMA E POSSIBILI SVILUPPI FUTURI	61
5.1. Condizioni di illuminazione e distorsioni prospettiche.....	61
5.2. Variazioni di scala.....	63
5.3. Teaching su una porzione di immagine	66
6. CONCLUSIONI.....	69
7. BIBLIOGRAFIA	71
8. RINGRAZIAMENTI.....	73

Introduzione

Oggi giorno la maggior parte dei prodotti presenti sul mercato viene venduta impacchettata all'interno di confezioni aventi le più svariate forme e colori. La presenza di imballaggi trova le sue fondamenta in ragioni di carattere igienico-sanitario, ma anche funzionale e pubblicitario. Si pensi ad esempio ai sacchetti di caffè macinato, patatine, cibo per animali: impossibile pensare di acquistarli senza un confezionamento adeguato. Un altro caso tipico riguarda le confezioni di cibo tascabili: merendine e snack vengono generalmente incartati uno ad uno, in modo da essere facilmente e comodamente trasportabili ovunque.

Le strategie di marketing e pubblicità, inoltre, impongono colori sgargianti in grado di catturare l'attenzione del consumatore, immagini evocative e frasi ad effetto per qualunque tipologia di prodotto. L'acqua e le bibite, ad esempio, vengono tipicamente vendute all'interno di bottiglie di plastica, sopra le quali sono incollate delle etichette aventi scopo prevalentemente pubblicitario.

Un'altra funzione fondamentale degli imballaggi è di carattere informativo: l'etichetta è proprio ciò che guida il consumatore all'acquisto del prodotto, pertanto è necessario che contenga informazioni esaustive che possano indirizzarlo nella scelta dell'articolo più adatto alle proprie esigenze. Alcune di queste informazioni sono, inoltre, vincolate da specifiche norme di legge, quali l'elenco degli ingredienti, la tabella nutrizionale, l'indicazione di provenienza, la data di scadenza o di confezionamento dell'articolo specifico.

Infine, uno degli elementi che nel corso del tempo ha acquisito un'importanza sempre maggiore è il codice a barre. Si tratta di un insieme di elementi grafici ad elevato contrasto in grado di essere letti tramite un sensore a scansione e decodificati per restituire le informazioni associate al prodotto, tipicamente il suo prezzo al momento del pagamento. Tale

elemento, che ha evidentemente assunto un valore fondamentale nel mercato moderno, deve necessariamente trovarsi su ogni singolo prodotto, stampato o applicato sulla confezione.

A questo punto risulta chiaro che gli imballaggi delle merci hanno ad oggi diverse ed importanti funzionalità. Questo significa che è indispensabile, a livello industriale, avere a disposizione metodologie efficaci ed efficienti per la loro creazione, a partire dal progetto per arrivare al packaging vero e proprio dei prodotti. Tipicamente, una volta ideati, gli imballaggi vengono stampati uno di seguito all'altro su nastri di grandi dimensioni, che dovranno poi essere montati su appositi macchinari per il taglio dei singoli elementi e il confezionamento vero e proprio degli articoli.

L'obiettivo di questa tesi è quello di analizzare il procedimento industriale che ad oggi viene utilizzato per la separazione dei singoli imballaggi a partire dal nastro, valutarne pro e contro ed ideare un procedimento innovativo, in grado di apportare miglioramenti significativi al sistema attualmente in uso. Per fare ciò, è stato analizzato nel dettaglio lo scenario di studio, focalizzando l'attenzione sulle soluzioni adottate fino a questo momento e sulle evoluzioni possibili. Quindi, è stato progettato e implementato un algoritmo innovativo basato su un sistema di visione.

Il progetto in analisi è stato realizzato presso Datalogic Automation s.r.l., azienda leader nel mercato dell'acquisizione automatica dei dati e dell'automazione industriale.

La tesi è organizzata come segue: nel Capitolo 1 verranno introdotti lo scenario di studio e le soluzioni adottate sino ad oggi; nel Capitolo 2 saranno descritti funzionamento e vantaggi derivanti dall'adozione di un sistema di visione; il Capitolo 3 conterrà i dettagli implementativi dell'algoritmo proposto; nel Capitolo 4 saranno presentati i risultati sperimentali che hanno influenzato alcune importanti scelte progettuali; infine il Capitolo 5 fornirà una panoramica sulle problematiche e le rispettive soluzioni analizzate per ottenere risultati soddisfacenti, oltre ai possibili sviluppi futuri.

1. Scenario di studio

1.1. Il processo di confezionamento

Il processo di confezionamento consiste nell'applicazione di un imballaggio al prodotto, a scopo in primo luogo protettivo, minimizzando l'influenza dell'ambiente esterno. Si pensi ad esempio all'industria alimentare, per la quale i materiali di confezionamento devono rispettare caratteristiche specifiche: non-tossicità, capacità di fornire un'adeguata protezione sanitaria, impedimento di ingresso e uscita di gas e odori, facilità di apertura, chiusura, smaltimento.

Il confezionamento ha, come già accennato in precedenza, un'importanza fondamentale in riferimento ad alcuni aspetti della vita di un prodotto: il contenimento, la conservazione, la commercializzazione ed il trasporto. I materiali utilizzati presentano, evidentemente, una notevole varietà: dal vetro, al cartone, ai metalli, fino a una vastissima scelta di materiali plastici.

Durante il processo di confezionamento, l'imballaggio del prodotto viene creato a partire da una bobina di elementi stampati uno di seguito all'altro. Uno o più macchinari si occupano della separazione dei singoli package e della realizzazione del prodotto finito, correttamente imballato ad esempio tramite tecniche di termosaldatura. In questa tesi l'attenzione è focalizzata sul processo di separazione dei singoli imballaggi che costituiscono la bobina: occorre un metodo efficace ed efficiente per individuare dove termina l'elemento precedente e inizia quello successivo.



Figura 1.1 – Bobine di imballaggi

Si analizzano ora le esigenze specifiche che un'industria di confezionamento ha la necessità di soddisfare. Il parametro che più di ogni altro è fondamentale tenere in considerazione durante il processo appena descritto è sicuramente la velocità del dispositivo nel riconoscimento del punto di taglio. E' infatti indispensabile sfruttare al meglio la rapidità dei nastri su cui sono montate le bobine, per ovvi motivi di efficienza. Non va tuttavia sottovalutata l'importanza della precisione e della ripetibilità della misura, poiché è indispensabile che le confezioni ottenute al termine del processo di packaging del prodotto siano una identica all'altra.

Un altro elemento a cui tipicamente si presta particolare attenzione è quello della semplicità di installazione e di utilizzo del sensore, in modo da rendere operativo il processo nel minor tempo possibile. Infine, robustezza e ingombro del dispositivo possono giocare un ruolo importante in particolari condizioni di utilizzo, quando lo spazio o le vibrazioni del macchinario possono influenzare significativamente l'andamento del processo produttivo.

1.2. Identificazione dello switching point

Una prima soluzione, estremamente semplice ma efficace, a cui si potrebbe pensare per la gestione del suddetto problema consiste nella valutazione delle dimensioni fisiche dell'imballaggio e nel calcolo della distanza tra uno *switching point* e il successivo. Lo *switching point* rappresenta il punto di taglio tra i due imballaggi adiacenti sulla bobina, come mostrato in figura 1.2. Si può trattare sia di una distanza spaziale, misurabile run-time tramite un *encoder*, sia di una distanza temporale, valutata in base alla velocità del nastro trasportatore.



Figura 1.2 – Switching point

Appare tuttavia evidente che tale soluzione non risulta sufficientemente flessibile, in quanto non presenta alcuna capacità di adattamento alle condizioni esterne. Si pensi ad esempio ad un'anomalia del macchinario che causi uno stallo, anche temporaneo, della bobina: un sistema di taglio a tempo o basato su riferimenti spaziali continuerebbe ad inviare gli impulsi senza accorgersi del guasto, rischiando di tagliare le confezioni in punti sbagliati e rovinando in questo modo tutti gli imballaggi, fino all'intervento di un operatore. Da qui si deduce che un sistema di questo tipo, molto economico e di immediata realizzazione, risulta però poco

pratico, difficilmente gestibile e in fin dei conti anche dispendioso, avendo necessità di un controllo visivo da parte di un operatore.

Una soluzione invece largamente utilizzata, anch'essa piuttosto semplice ed economica, ma indiscutibilmente migliore in termini di flessibilità, è data dall'uso del sensore fotoelettrico di contrasto o *marker sensor*, il cui funzionamento sarà spiegato ed analizzato nel dettaglio nei paragrafi successivi.

1.3. Il marker sensor

Il sensore di contrasto, come si evince dal nome, è in grado di rilevare in modo automatico aree che presentano una variazione significativa dell'intensità luminosa in punti adiacenti. Questo tipo di sensori è noto anche col nome di *lettore di tacche colorate*. Avendo a disposizione una bobina di imballaggi tutti uguali, è evidente che identificare un'area ad elevato contrasto, denominata *marker*, all'interno del singolo elemento garantisce una buona ripetibilità. Nel caso in cui tale zona sia posizionata in modo strategico, in prossimità del punto di taglio che separa due elementi adiacenti, utilizzando un *marker sensor* si ottiene esattamente il risultato desiderato: l'identificazione del punto di taglio tra i due imballaggi, in modo estremamente efficace, veloce ed efficiente.



Figura 1.3 - marker

1.3.1. *Principio di funzionamento*

Il sensore di contrasto si basa sul principio di funzionamento dei sensori di tasteggio focalizzato. Essi sono in grado di rilevare la presenza di oggetti sfruttando la diffusione in riflessione che si ha quando la radiazione elettromagnetica proveniente da una sorgente incontra l'oggetto da rilevare. Sorgente e ricevitore sono montati sullo stesso lato di un unico dispositivo fisico. Nel caso specifico dei lettori di tacche colorate, la radiazione elettromagnetica di un emettitore viene focalizzata sull'oggetto recante il marker da rilevare, in uno spot di dimensioni confrontabili con quelle della tacca. A questo punto la radiazione diffusa dalla superficie dell'oggetto viene raccolta dal sistema di ricezione e trasformata in un livello di segnale. Il sensore è dotato, al suo interno, di un comparatore, che provoca una commutazione delle uscite quando il segnale rilevato è superiore ad una soglia minima di riferimento. Tale livello di riferimento viene determinato in una fase iniziale come valore medio tra il segnale acquisito sullo sfondo (*valore di background*) e il segnale acquisito su una tacca avente stesso colore e dimensioni dei marker che si vogliono individuare.

Lo spettro della radiazione dell'emettitore dipende dal colore della tacca. Il colore di un oggetto è infatti dovuto principalmente all'assorbimento selettivo della radiazione su di esso incidente, ovvero dalla capacità dell'oggetto stesso di assorbire alcune lunghezze d'onda della radiazione e di rifletterne altre. Pertanto, risulta indispensabile che la differenza di intensità tra la radiazione riflessa dalla tacca e quella riflessa dallo sfondo sia sufficientemente elevata da permettere il corretto riconoscimento del segnale da parte del sensore fotoelettrico.

1.3.2. *Caratteristiche dei sensori di contrasto*

Esistono vari tipi di sensori di contrasto, che si differenziano per il formato della lente, il tipo di emettitore, la frequenza di commutazione. Di seguito saranno analizzate le principali differenze tra le categorie, con relativi vantaggi e svantaggi in funzione delle esigenze di un'industria di confezionamento.

La frequenza di commutazione, e quindi di rilevamento del marker, gioca ovviamente un ruolo fondamentale. Essa dipende inevitabilmente dall'elettronica presente sul dispositivo, la quale incide poi sul costo.

Per quanto riguarda l'affidabilità e la ripetibilità della misura, si focalizza l'attenzione sul tipo di emettitore luminoso utilizzato. Nel corso del tempo si è passati dalla lampada alogena al LED (*Light Emitting Diodes*), e ad oggi è possibile usarne di diverse tipologie. Esistono infatti marker sensor che emettono solo luce bianca, luce rossa o verde, luce RGB. La capacità di riconoscimento delle tacche è diversa nei tre casi: quelli a luce rossa o verde sono caratterizzati da una maggiore affidabilità rispetto alla sola luce bianca, in quanto effettuano il rilevamento del marker con entrambe le radiazioni, scegliendo poi quella che fornisce in uscita un segnale migliore in riferimento alla soglia. Il segnale riflesso, infatti, a parità di colore del materiale, avrà un valore diverso se illuminato con radiazione elettromagnetica rossa o verde, ed in particolare uno dei due segnali risulterà più robusto in termini di differenza dal valore di soglia. Un ulteriore miglioramento è stato introdotto nel tempo dal led RGB: si tratta di un singolo led in grado di emettere luce rossa, verde e blu, e di selezionare di conseguenza il colore col quale si ottiene la migliore differenza tra il valore della tacca e quello dello sfondo.

Un altro elemento che distingue le varie tipologie di sensori di contrasto è l'ottica, che però influenza inevitabilmente le dimensioni e quindi anche l'ingombro del dispositivo, oltre al costo. Esistono anche modelli con

obiettivi mobili e intercambiabili, per garantire la massima flessibilità. In alcuni casi può essere indispensabile avere a disposizione un'ottica in grado di rilevare un segnale stabile anche in condizioni di stress, quali ad esempio vibrazioni significative del rullo sul quale scorrono le bobine di imballaggi. Legato ad un discorso di robustezza è anche il materiale con cui è costruito il sensore: tipicamente plastica oppure metallo.

In merito alla semplicità di installazione e configurazione, va menzionata l'esistenza di modelli in grado di eseguire un'acquisizione dinamica, oltre che statica. Questo significa sostanzialmente che la fase di acquisizione del segnale di tacca e sfondo può essere effettuato direttamente run-time, facendo scorrere i rulli con le bobine di imballaggi, e risparmiando quindi il tempo necessario per le varie misurazioni su elementi campione al fine di stabilire il valore di soglia.

Infine, sono in commercio alcuni sensori di contrasto con display, elemento che li rende molto flessibili in fase di configurazione e di utilizzo.

1.3.3. *Problematiche dei sensori di contrasto*

Nei paragrafi precedenti sono state elencate le caratteristiche dei sensori di contrasto, le differenze in termini di prestazioni e di adattabilità alle esigenze dell'industria di confezionamento. Si analizzano ora gli aspetti negativi dei suddetti dispositivi, che portano all'idea dell'utilizzo di un sistema di visione in sostituzione del lettore di tacche.

Il primo elemento da sottolineare è la necessità di introdurre fisicamente il marker all'interno dell'imballaggio. Un sensore di contrasto è infatti in grado di rilevare elementi adiacenti aventi una sostanziale differenza di intensità luminosa. Poiché i colori del package sono tipicamente dettati da esigenze di marketing o brand, la soluzione classicamente adottata consiste nell'inserimento di tacche rettangolari nere o bianche, in modo da avere un elemento fortemente distintivo da rilevare. Questo ha però come

conseguenza un risultato inevitabilmente poco gradevole a livello estetico, che magari distoglie l'attenzione da un design studiato appositamente per attirare il consumatore. In alcuni casi tale problema si risolve aumentando la dimensione del package, in modo da ricoprire e rendere invisibile la parte in cui si trova il marker. In tal modo, però, subentra un problema di spreco di materiale, se si considera l'insieme completo di tutti gli imballaggi presenti nella bobina, come mostrato in figura 1.4.



Figura 1.4 – Spreco di materiale per coprire il marker

Esistono inoltre alcuni casi particolari in cui un sensore di contrasto può presentare dei problemi nel rilevamento del marker. All'interno dell'etichetta possono, ad esempio, essere presenti elementi fuorvianti, per i quali il dispositivo rileva un segnale simile a quello della tacca, identificando quindi uno *switching point* non corretto. Una soluzione che è possibile adottare per risolvere il problema è quella di utilizzare finestre temporali per limitare i rilevamenti da considerarsi validi.

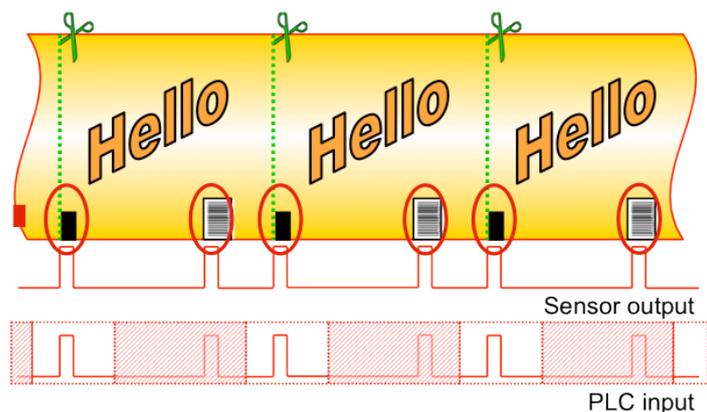


Figura 1.5 – Sensore di contrasto con finestre statiche

Si focalizzi ora l'attenzione sul fenomeno dello *stretching*, rappresentato in figura 1.6, nella pagina seguente: la bobina di imballaggi presenta un'anomalia per cui l'estensione in senso orizzontale è maggiore rispetto alle altre bobine. Un sensore di contrasto configurato con la tecnica delle finestre temporali statiche non è in grado di gestire il fenomeno, perde il sincronismo e quindi non riesce ad identificare in modo corretto lo *switching point*. Si ricorda inoltre che l'errore si propaga poi su tutti gli elementi successivi, fino all'intervento dell'operatore o al termine della bobina che presenta l'anomalia. In figura 1.6 l'errore di taglio è evidenziato dal cerchio blu.

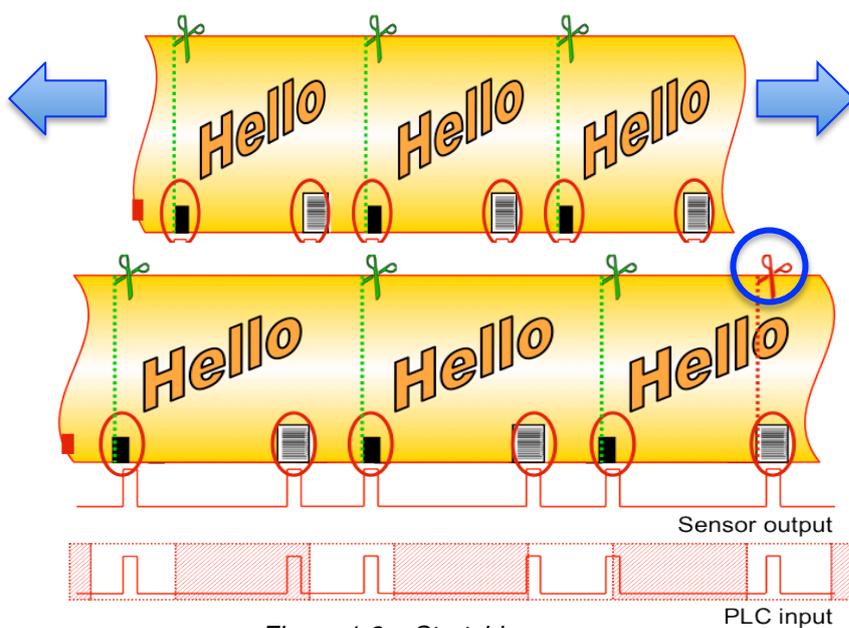


Figura 1.6 – Stretching

Un altro caso di studio che è stato preso in considerazione è quello di *overlapping*, mostrato in figura 1.7. Si tratta della sovrapposizione tra la bobina che termina e quella che inizia. Tale sovrapposizione può ovviamente non essere perfetta, e quindi introdurre uno sfasamento che un sensore di contrasto con finestre statiche non sarebbe in grado di gestire. Le conseguenze sarebbero analoghe a quelle descritte per lo stretching: una perdita di sincronismo che comporterebbe la propagazione dell'errore nell'individuazione del punto di taglio fino all'intervento di un operatore. La figura sottostante evidenzia gli errori di taglio tramite i cerchi blu.

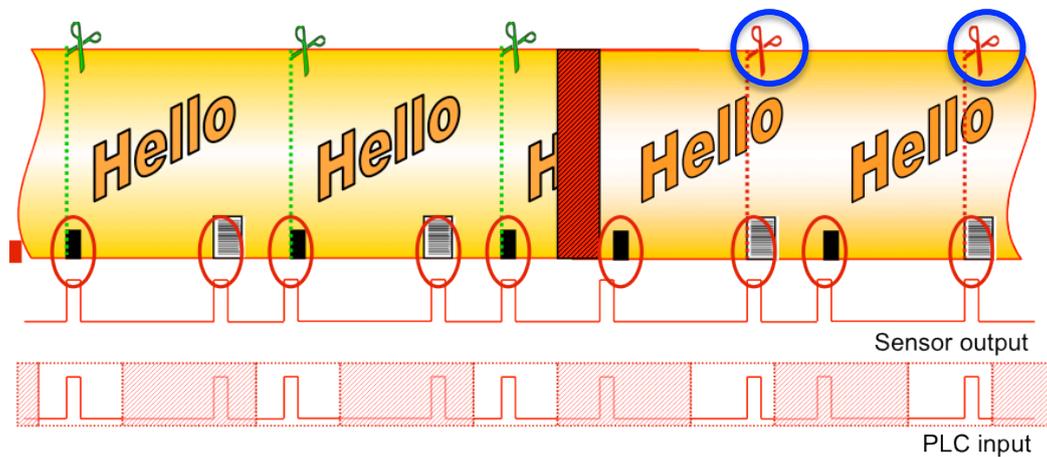


Figura 1.7 - Overlapping

Le ragioni sopraelencate portano all'idea di utilizzare un sistema di visione per sopperire alle mancanze dei sensori di contrasto, al fine di migliorare il processo di individuazione dello *switching point* in termini di affidabilità, precisione, e resistenza ad alcune tipologie di guasti. Nel capitolo successivo saranno esposti il principio di funzionamento di tale sistema ed i vantaggi che ne derivano.

2. Approccio alternativo: un sistema di visione

Il sensore di contrasto, come ampiamente descritto nel capitolo precedente, è estremamente efficiente ed efficace in normali condizioni di utilizzo. Esistono tuttavia alcune circostanze che possono causare problemi di notevole rilevanza. L'approccio alternativo che è stato analizzato per la gestione del processo di individuazione del punto di taglio nella bobina di imballaggi consiste nell'uso di un sistema di visione. In tal modo si ha infatti la possibilità di monitorare costantemente le etichette che scorrono sul rullo, minimizzando quindi il rischio di perdita di sincronismo ed i conseguenti errori. Risulta però fondamentale non rinunciare ai vantaggi dei sensori di contrasto, in primo luogo la velocità nel riconoscimento dello *switching point*. Di seguito sarà esposto nel dettaglio il principio di funzionamento del sistema di visione, insieme alle prospettive di miglioramento che si potrebbero ottenere dal suo impiego.

2.1. Principio di funzionamento

Il sistema di visione si basa sull'utilizzo di una telecamera e sull'analisi dell'immagine che da essa proviene. Questo significa che si potrà utilizzare tutto ciò che è stampato sull'imballaggio per capire se si è in prossimità del punto di taglio. Ovviamente, utilizzare una telecamera che analizzi ogni pixel presente sull'etichetta riduce drasticamente la velocità di utilizzo del sistema, requisito essenziale per poter competere con i sensori di contrasto. L'idea è allora di focalizzare l'attenzione non sull'intera stampa, ma solo su alcune zone particolarmente caratteristiche, mantenendo però i requisiti fondamentali di robustezza e ripetibilità. Il

problema si riduce quindi all'individuazione delle zone più adatte a questo scopo all'interno di un'immagine qualsiasi. Lo *switching point* sarà poi individuato in funzione delle suddette aree. La figura sottostante rappresenta il funzionamento del sistema, la freccia mostra la direzione di scorrimento della bobina di etichette sotto la telecamera, rappresentata dalla colonna in rosso.

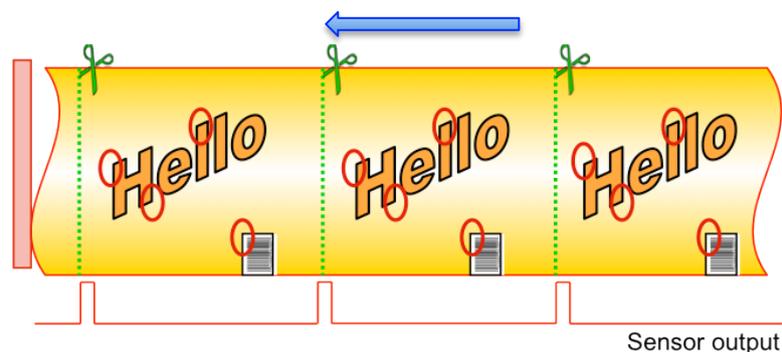


Figura 2.1 – Principio di funzionamento del sistema di visione: individuazione delle zone caratteristiche all'interno dell'imballaggio

Il procedimento appena descritto suggerisce la necessità di una prima fase di "addestramento", durante la quale devono essere individuate le aree che caratterizzano l'immagine dell'imballaggio, e calcolato di conseguenza il punto di taglio. A seguire, sarà poi possibile incrementare la velocità del sistema analizzando soltanto le zone caratteristiche degli imballaggi che scorrono sul rullo.

Occorre infine una precisazione in merito al sistema di visione utilizzato per questo progetto. Esistono ovviamente moltissime tipologie di telecamera, ma il requisito fondamentale di velocità impone una restrizione sulle possibilità di scelta. In particolare, per ottenere prestazioni accettabili, risulta conveniente utilizzare un dispositivo con una frequenza di acquisizione elevata, in grado di elaborare immagini aventi un numero ridotto di pixel. La scelta si indirizza quindi verso una *telecamera lineare*, in grado di acquisire una singola linea di pixel per ogni scansione,

garantendo però velocità e prestazioni adeguate al caso di studio. Mentre la bobina scorre sul rullo, la telecamera, fissa, acquisisce una per volta le linee di pixel adiacenti, permettendo una eventuale bufferizzazione nel caso in cui si abbia la necessità di visualizzare l'etichetta nella sua interezza.

In riferimento alla figura 2.1 e a tutte le immagini successive, si ipotizzi che la telecamera lineare si trovi a sinistra in corrispondenza della colonna rossa e sia in grado di leggere una linea di pixel orientata verticalmente. Il rullo di etichette scorre da destra verso sinistra, come mostrato dalla freccia.

2.2. I vantaggi del sistema di visione rispetto al sensore di contrasto

Nel capitolo precedente sono state analizzate le problematiche legate all'uso di un sensore di contrasto per l'individuazione dello *switching point*; un sistema di visione può garantire una migliore gestione di alcune di queste problematiche.

Il primo elemento che occorre sottolineare è l'eliminazione dei marker dagli imballaggi. Si parla infatti di *algoritmo di etichettatura markerless*, in quanto la caratteristica fondamentale che differenzia il sistema progettato in questa tesi è proprio l'eliminazione delle tacche necessarie ai sensori di contrasto. Si ha quindi la possibilità di lasciare l'intero spazio al brand, alle informazioni che è necessario esplicitare per legge, a scritte, colori e forme dettate da esigenze di marketing. La tacca nera o bianca scompare dalla vista del consumatore, senza la necessità di sprecare del materiale aggiuntivo per ricoprirlo, tecnica adottata, ad esempio, nell'etichettatura delle bottiglie d'acqua, come mostrato nel capitolo precedente. Questo si traduce, di conseguenza, in una riduzione di costi da parte del produttore,

nonché in un risparmio in termini di materiale che può essere considerato positivo anche dal punto di vista ecologico.

Un altro elemento che era stato definito problematico per i sensori di contrasto consisteva nella possibile presenza, all'interno dell'imballaggio, di elementi fuorvianti per il sensore. Tale problema può scomparire utilizzando invece un sistema di visione, nel caso in cui le aree di interesse riescano ad essere selezionate proprio in base alla loro unicità all'interno di un singolo elemento della bobina. Questo procedimento richiede ovviamente un'analisi dettagliata dell'intera immagine, da effettuare in una prima fase tramite un'immagine campione. Non appare quindi necessario ricorrere a tecniche quali le finestre temporali, a cui possono legarsi, come è già stato accennato in precedenza, problemi di sincronizzazione. Il dettaglio in merito alla selezione delle aree più idonee all'interno dell'imballaggio sarà fornito nel capitolo successivo.

Un altro caso di studio riportato in precedenza è il fenomeno di *stretching*. Mentre un sensore di contrasto con finestre temporali perde il sincronismo, un sistema di visione che si basi su alcuni elementi fissi dell'etichetta ha una capacità di adattamento decisamente migliore. Basando le proprie valutazioni su alcuni elementi che caratterizzano l'immagine, il sistema analizza la bobina di imballaggi che scorre sotto la sua visione indipendentemente da vincoli di tempo o spazio: quando le aree caratteristiche vengono individuate il sistema sa con precisione quanta distanza intercorre tra la zona corrente e il punto di taglio. Occorre però precisare che il sistema garantisce una buona robustezza solo nel caso in cui lo *stretching* non abbia peso variabile tra le diverse zone caratteristiche dell'immagine. Se l'etichetta risulta "stirata" in modo non uniforme non è possibile garantire la correttezza dello *switching point* individuato. Ad ogni modo, anche nel caso di errore non si avrà una propagazione fino all'intervento dell'operatore, come nel caso dei sensori di contrasto, in quanto l'analisi di ciascun imballaggio è completamente indipendente dai precedenti e dai successivi.

Per quanto concerne il fenomeno di *overlapping*, il principio di funzionamento del sistema di visione garantisce, analogamente allo *stretching*, una buona robustezza. Infatti, anche se l'imballaggio affetto dal fenomeno potrebbe essere non recuperabile, il sistema è comunque in grado di permettere l'individuazione dell'anomalia, garantendo sempre l'indipendenza dell'analisi di tutti gli altri elementi della bobina. Questo si traduce in un'unica etichetta non conforme, senza propagazione dell'errore.

Un ultimo elemento avente una certa rilevanza è dato, infine, dalla capacità del sistema di effettuare un controllo di qualità dell'imballaggio durante la sua analisi per l'individuazione del punto di taglio. Il procedimento consiste, infatti, nella ricerca degli elementi caratteristici dell'immagine e nella verifica che la loro posizione sia conforme a quella dell'etichetta campione. Ciò significa che l'individuazione di un imballaggio anomalo può essere segnalata all'operatore, senza però dover interrompere il processo e senza avere conseguenze sugli altri imballaggi della bobina. Il suddetto controllo di qualità ha il pregio di essere effettuato direttamente durante il processo di taglio, con un evidente risparmio di tempo durante le fasi successive, in cui si potrà evitare l'analisi degli eventuali elementi anomali individuati dal sistema di visione.

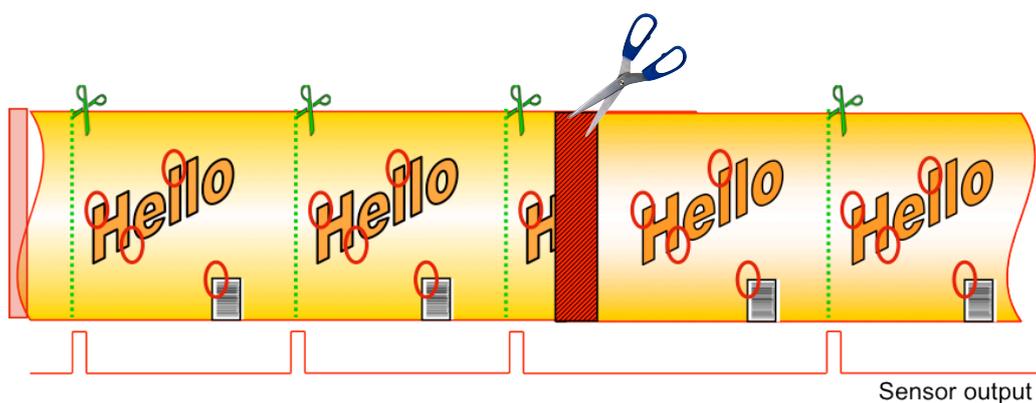


Figura 2.2 – Etichetta anomala in caso di overlapping

Tuttavia, è importante precisare che il procedimento in esame non può garantire l'individuazione di tutte le non conformità, ed è inoltre possibile che alcuni elementi segnalati come anomali siano in realtà recuperabili. Si pensi al caso di *overlapping*: l'etichetta affetta dal fenomeno potrebbe essere facilmente recuperabile con un taglio addizionale, non gestito dal sistema di visione, come riportato in figura 2.2.

3. Algoritmo a due fasi per l'individuazione dello switching point

Il principio di funzionamento del sistema esposto nel capitolo precedente prevede, come già accennato, due fasi distinte di esecuzione. Tale differenziazione si rende necessaria per garantire da un lato la correttezza e la ripetibilità dei rilevamenti, dall'altro una velocità di esecuzione a regime che possa garantire una concorrenza accettabile con i sensori di contrasto. Il sistema di visione, infatti, è in grado di trattare qualsiasi tipo di immagine, ma, in prima battuta, ha la necessità di analizzarla al fine di individuare le aree di maggiore interesse, adatte ad essere considerate come elementi distintivi dell'imballaggio. Pertanto, una prima elaborazione di un'etichetta "campione" è indispensabile. Tale procedimento risulterà inevitabilmente più lento rispetto al funzionamento del sistema *run-time*, ma potrà essere effettuato una sola volta per l'intero rullo di etichette, o anche per più bobine dello stesso imballaggio. Inoltre, prevedendo la possibilità di salvataggio dell'elaborazione effettuata sull'immagine, sarà possibile riutilizzarla in momenti futuri, evitando di ripetere inutilmente la procedura iniziale.

Nella seconda fase, il riconoscimento dell'etichetta e l'identificazione dello *switching point* potranno essere effettuati prendendo in considerazione unicamente le zone caratteristiche, la loro posizione e la loro distanza.

Le due fasi sono identificate rispettivamente come *teaching* e *matching*. La prima prevede infatti l'apprendimento da parte del sistema delle caratteristiche dell'immagine, l'analisi completa e l'individuazione delle zone di interesse, mentre la seconda fase consiste nell'esecuzione vera e propria, con l'identificazione dei punti di taglio per tutte le etichette che scorrono sul rullo. Proprio grazie alla combinazione di due tecniche ben

note in letteratura, ovvero la caratterizzazione dell'immagine tramite *keypoints* e il *pattern matching*, l'algoritmo proposto è in grado di ottenere buone prestazioni in termini di affidabilità e velocità. Dati più specifici in merito saranno esposti nel capitolo 4, mentre nei paragrafi che seguono sarà descritto nel dettaglio il funzionamento di ciascuna delle due fasi del sistema.

3.1. Fase 1: Teaching

Lo scopo della prima fase dell'algoritmo proposto è quello di individuare le zone caratteristiche dell'immagine campione, in modo da ridurre la mole di dati da utilizzare run-time per ottenere prestazioni accettabili.

L'idea è quella di individuare un certo numero di *patch* significative, identificare tra queste i *pattern* "migliori" (secondo alcuni criteri i cui dettagli saranno spiegati nel seguito), e considerare come *switching point* l'ultima linea di pixel dell'immagine, calcolando l'*offset* esistente tra questa e l'ultimo pattern. La figura 3.1 fornisce una rappresentazione grafica degli elementi principali che vengono considerati durante la fase di *teaching*.

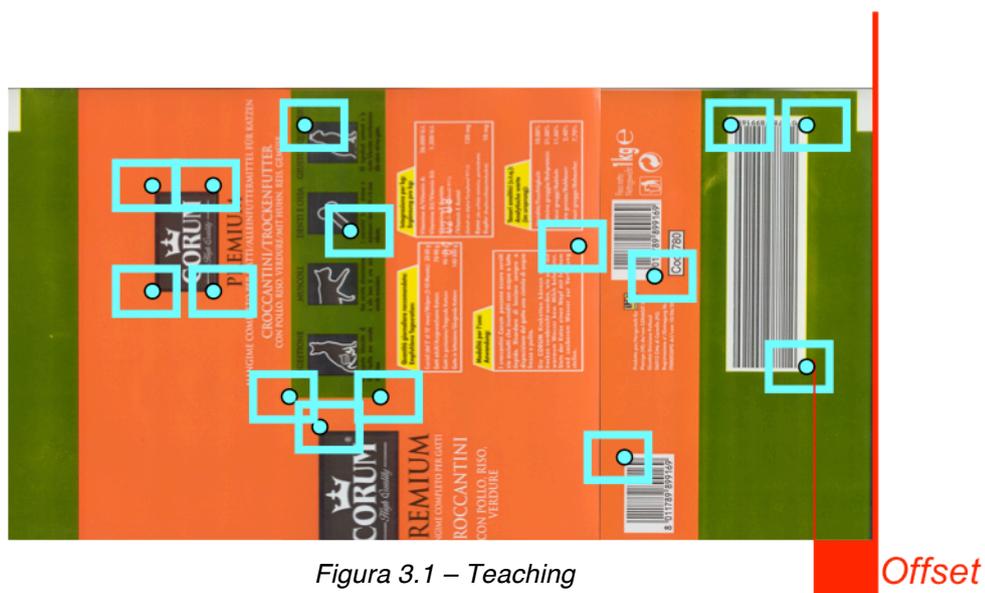


Figura 3.1 – Teaching

Offset

Di seguito verrà fornita una descrizione dettagliata della modalità di selezione dei pattern sufficientemente distintivi, seguita dall'analisi delle problematiche riscontrate durante i primi test, e dai miglioramenti introdotti al fine di ottenere risultati soddisfacenti.

3.1.1. Il procedimento di individuazione dei pattern

Appare chiaro fin da subito che il problema fondamentale è capire come determinare le aree di un'immagine che possano essere considerate adatte allo scopo. Occorre ottenere patch che siano facili da individuare durante la successiva fase di riconoscimento, ma anche sufficientemente diverse le une dalle altre, in modo da avere una buona copertura dell'intera immagine e ridurre la possibilità di falsi positivi. Esistono in letteratura diverse proposte, che vanno dall'individuazione degli *edge*, a quella dei *corner*, fino alle più generiche *features*. Tutti questi algoritmi, però, sono in grado di individuare un numero estremamente elevato di *keypoint*, che risulta eccessivo per le esigenze dell'algoritmo proposto in questa tesi. Pertanto, dopo un'attenta analisi, si è deciso di ricorrere ad alcuni di questi algoritmi, ai quali però segue una post-elaborazione studiata ad-hoc per il problema in esame.

Il primo elemento da considerare è l'immagine stessa: una trasformazione in scala di grigi comporta una perdita d'informazione che può essere considerata trascurabile, avendo però l'effetto di ridurre il peso dei dati da memorizzare, e velocizzando di conseguenza l'intero sistema.

Per quanto riguarda gli algoritmi di individuazione di *features*, sono state analizzate nel dettaglio varie proposte: i *corner detector* di Harris^[1] e Moravec^[2], FAST^[3], Good Features To Track^[4], SURF^[5]. Al fine di ottenere un numero adeguato di elementi sufficientemente distintivi, si è deciso di utilizzare in prima battuta un *corner detector*, seguito, nel caso in cui i pattern individuati non siano sufficienti, da un *feature detector*. Gli algoritmi

proposti sono in grado di fornire in uscita i *keypoint* dell'immagine secondo un certo ordine di "qualità", che è stato sfruttato come punto di partenza per le successive valutazioni. La fase di post-elaborazione continua comunque ad essere necessaria, in quanto l'algoritmo di etichettatura markerless proposto in questa tesi ha l'esigenza di individuare *patch* distintive, e non singoli pixel.

Un elemento che può essere considerato strategico per la successiva fase di riconoscimento è la *posizione* delle patch: infatti, poiché gli imballaggi sono tutti uguali, le patch dovranno trovarsi circa nella stessa area in ciascuna etichetta. Conoscendo la loro localizzazione si può quindi ridurre lo spazio di ricerca ad un sottoinsieme ristretto di pixel. Si è dunque pensato di considerare come elemento caratteristico anche la posizione delle patch individuate dal *corner detector*: a partire dall'elemento con maggiore *cornerness* e quindi potenzialmente più distintivo, si individuano alcune sezioni orizzontali all'interno dell'immagine, si confrontano le une con le altre tutte le patch interne, e si seleziona infine al più un pattern per ciascuna sezione. Inoltre, allo scopo di snellire il processo eliminando rapidamente i dati non accettabili, viene preso in considerazione un valore di soglia per la misura di similarità utilizzata nella valutazione delle singole patch: le coppie di patch che risultano "troppo simili" vengono immediatamente eliminate, in quanto non sufficientemente distintive. La figura 3.2 mostra un esempio di patch evidentemente non accettabili, in quanto caratterizzate da un'eccessiva similarità.



Figura 3.2 – Patch aventi una misura di similarità molto alta

Un ulteriore controllo di affidabilità delle *patch* individuate consiste in un *pattern matching* dell'elemento all'interno della sezione orizzontale di

appartenenza. Se la massima misura di similarità rilevata non supera la soglia, allora ci sono buone probabilità che l'elemento non possa essere confuso con altre zone dell'immagine durante la fase di riconoscimento, pertanto è idoneo ad essere trattato come *pattern*. La successiva fase di riconoscimento, infatti, opererà in una specifica zona dell'immagine per ciascuno dei pattern da individuare.

In figura 3.3 è mostrato il procedimento sopra descritto: in blu sono rappresentate le patch individuate inizialmente, in rosso la patch caratterizzata da maggiore *cornerness*, in verde le patch appartenenti alla sezione orizzontale in esame, le quali verranno sottoposte a tutti i successivi controlli. Al più, soltanto uno di questi ultimi elementi potrà essere considerato come pattern. La medesima elaborazione verrà poi effettuata su tutti gli altri gruppi di patch, fino a quando resteranno in memoria solamente i pochi pattern idonei.

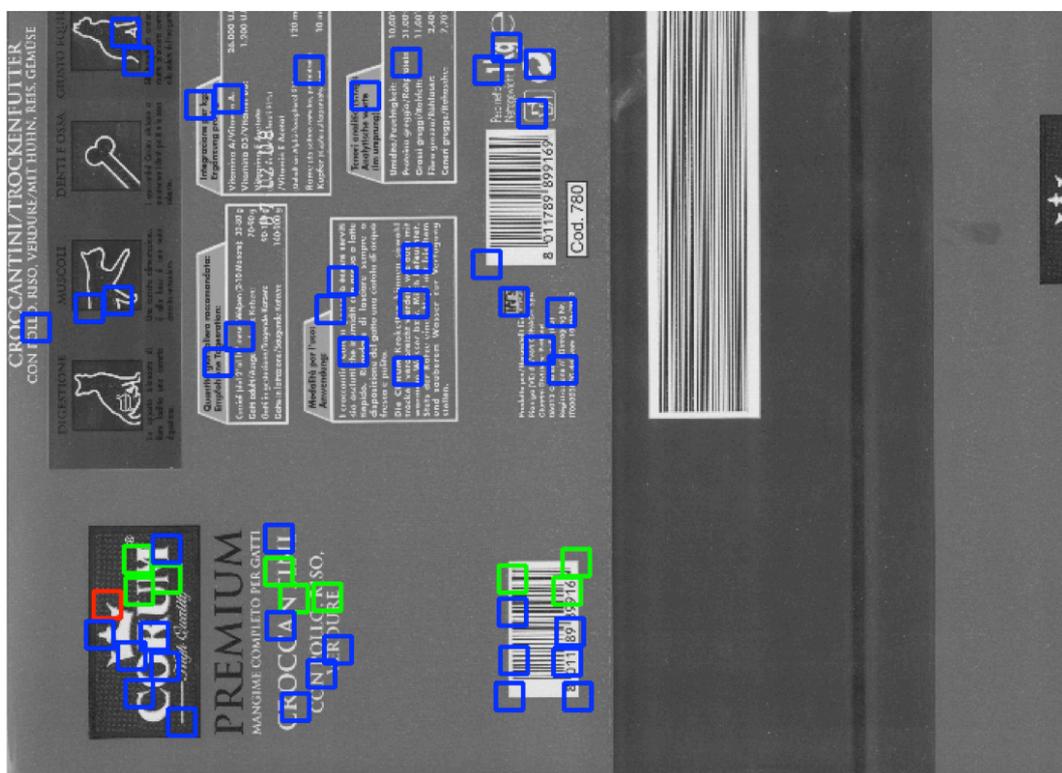


Figura 3.3 – Teaching: post-elaborazione delle patch per l'identificazione dei pattern

Il risultato di questa post-elaborazione è una drastica riduzione degli elementi che possono essere considerati sufficientemente distintivi per l'immagine, i quali però sono caratterizzati da una robustezza tale da garantire buoni risultati in fase di riconoscimento. Occorre tuttavia precisare che, considerando la posizione orizzontale come uno dei fattori distintivi delle patch, l'algoritmo prevede l'eliminazione di elementi che potrebbero essere accettabili in termini di misura di similarità e quindi sufficientemente caratteristici all'interno dell'immagine, ma sono posizionati in zone non strategiche per la successiva fase di *matching*.

Per quanto riguarda le misure di similarità, durante la descrizione del processo di post-elaborazione si è affermato che se una coppia di patch presenta un valore superiore alla soglia, allora entrambe le patch sono da considerarsi non accettabili. Questo perché la misura di similarità a cui si è fatto ricorso è la *Normalized Cross Correlation*^[6], i cui valori restituiti vanno da 0 se le immagini sono completamente diverse, a 1 in caso di match perfetto. In tal modo si è raggiunto un buon compromesso tra robustezza e rapidità di elaborazione.

Un altro elemento che è stato preso in considerazione per la selezione dei pattern è la *distanza orizzontale*: al fine di permettere alla fase di riconoscimento di poter trattare un singolo pattern per volta, localizzato in una zona orizzontale ben specifica dell'immagine, occorre evitare di avere elementi che si sovrappongono verticalmente anche in maniera parziale, poiché richiederebbero una elaborazione contemporanea di due pattern distinti, aumentando inutilmente la complessità del sistema e causando rallentamenti.

Infine, nel caso in cui il numero di pattern individuati dall'algoritmo sia inferiore rispetto al necessario, lo stesso procedimento si ripete partendo dai punti individuati dal *feature detector* e considerando come pattern finali, al termine dell'intera fase di *teaching*, l'insieme dei risultati di entrambe le post-elaborazioni. Questo permette, in molti casi, di

aumentare il numero complessivo di patch idonee al termine della fase di *teaching*.

A questo punto i pattern che hanno superato tutte le selezioni sopra descritte costituiscono l'insieme degli elementi di interesse dell'imballaggio in esame: viene valutata la loro posizione, vengono memorizzate le distanze relative tra i punti centrali dei pattern, viene calcolato l'offset come distanza tra l'ultima patch e la linea di pixel finale dell'immagine campione, e al termine della fase di *teaching* si ottiene la coordinata verticale dello *switching point*. E' importante sottolineare che tale coordinata è relativa al primo pattern, in ordine orizzontale, il quale viene considerato come zero per l'etichetta in esame. Questo perché, azzerando il conteggio per ciascuno degli imballaggi della bobina, si ha una buona garanzia di indipendenza degli elementi, che vengono trattati singolarmente tutti allo stesso modo, valutando distanze molto brevi. In tal modo si ha un vantaggio anche in termini di dimensione dei dati da trattare durante la fase di *matching*.

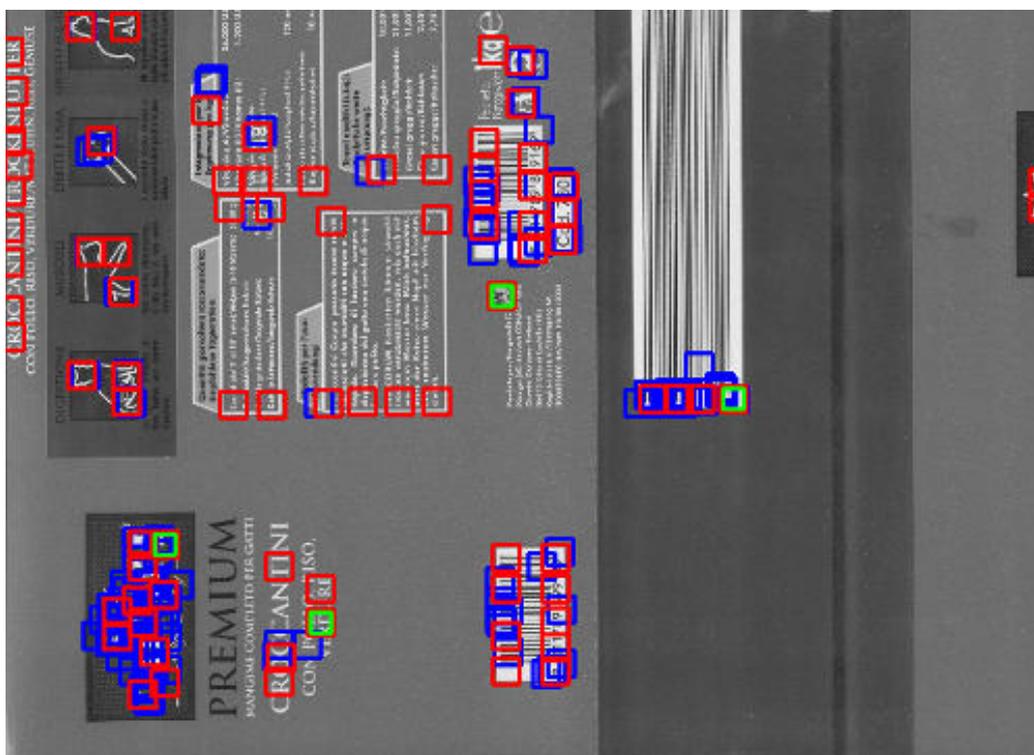


Figura 3.4 – Teaching: dalle patch ai pattern definitivi

L'immagine riportata in figura 3.4 rappresenta le diverse fasi del procedimento di *teaching*: in rosso sono mostrate le zone di interesse individuate dal *corner detector*, in blu quelle individuate dal *feature detector* e in verde i pattern ottenuti al termine di tutte le post-elaborazioni.

3.1.2. *Analisi delle problematiche*

Il primo problema da porsi è identificare quale sia il numero di pattern più adatto per una singola etichetta. Occorre infatti valutare diversi fattori: il requisito fondamentale di velocità in fase di *matching* impone un numero di pattern ridotto, ma, allo scopo di avere una robustezza accettabile e un buon controllo sull'intera estensione dell'immagine, un singolo elemento o una coppia di pattern non può essere sufficiente. Dopo una prima valutazione, si è considerato che 5 potesse essere un valore adeguato. Occorre tuttavia precisare che un vincolo sul numero di patch significative non influenza in alcun modo il procedimento di *teaching* in merito alla loro posizione: a seconda dell'immagine in esame è infatti possibile che tutti i pattern siano localizzati in una zona avente elementi particolarmente caratteristici. Gli unici vincoli riguardanti la posizione, infatti, sono quelli inerenti all'impossibilità di sovrapposizione orizzontale e verticale dei pattern.

A seguito dei primi test, tuttavia, nonostante l'utilizzo di due distinti algoritmi di *feature detection* e l'unione dei risultati di entrambe le elaborazioni, nella maggior parte dei casi il numero complessivo di pattern in grado di superare tutti i controlli si è rivelato insufficiente. Sono stati di conseguenza effettuati svariati esperimenti al fine di valutare quali fossero gli elementi con influenza maggiore sui risultati ottenuti, sui quali occorreva intervenire per migliorare le prestazioni dell'intero sistema: la soglia per la misura di similarità, la dimensione dei pattern considerati e la loro forma.

Si consideri a titolo di esempio la figura 3.5 riportata nella pagina successiva. L'evidente ridondanza di elementi presenti nell'etichetta comporta una difficoltà considerevole nell'individuazione di patch sufficientemente distintive.



Figura 3.5 – Immagine caratterizzata da elementi ridondanti

Per quanto riguarda i valori di soglia, una serie di test, che saranno riportati nel capitolo 4, sono stati sufficienti per individuare i valori ottimali. Al contrario, identificare le migliori forma e dimensione dei pattern ha richiesto un'analisi del problema estremamente approfondita e alcune implementazioni aggiuntive. Ovviamente, patch di dimensioni ridotte comportano un minore tempo di elaborazione e una minore esigenza di risorse, qualità indispensabili per soddisfare il requisito di velocità. Per contro, patch di grandi dimensioni sono evidentemente più distintive, in quanto contengono porzioni più grandi dell'immagine originale e sono quindi più facilmente riconoscibili.

3.1.3. Soluzioni proposte

Per risolvere il problema del dimensionamento dei pattern sono state valutate due soluzioni alternative: patch aventi dimensioni variabili e patch di forma rettangolare.

La prima soluzione ha il vantaggio, partendo da dimensioni ridotte, di riuscire a limitare l'occupazione di risorse e i tempi di elaborazione, se si ottengono risultati accettabili senza dover fare ricorso a tutto il pool di dimensioni. Si introduce tuttavia una complessità aggiuntiva: occorre tenere traccia delle dimensioni di ciascun pattern durante tutta la fase di *teaching*. Inoltre, è necessario precisare che, al fine di evitare sovrapposizioni in fase di *matching*, la minima distanza accettabile tra i pattern diventa pari alla dimensione maggiore del pool, con conseguente influenza sull'intero processo di *teaching*.

La seconda soluzione mantiene invece la semplicità della dimensione fissa, introducendo però un miglioramento derivante dal numero maggiore di pixel presenti all'interno della patch, che la rendono evidentemente più distintiva. Anche l'inevitabile *overhead* introdotto da questo sistema, però, consiste nel maggior numero di pixel da trattare in fase di *teaching* e da analizzare in fase di riconoscimento.

Per valutare quale delle due soluzioni fosse la migliore in termini di numero di pattern effettivamente identificati al termine dell'intera fase di *apprendimento*, sono state implementate entrambe e sono stati effettuati ulteriori test. I risultati, che verranno mostrati nel dettaglio nel Capitolo 4, hanno mostrato che la soluzione del pool di dimensioni garantisce la robustezza desiderata. Al fine di ottenere una maggiore configurabilità del sistema, infine, si è deciso di lasciare all'utente la possibilità di scegliere il numero di pattern che desidera identificare e le dimensioni che è disposto ad inserire nel pool per l'esecuzione ottimale dell'intera fase di *teaching*.

Le dimensioni testate e proposte vanno da un minimo di 9x9 pixel, fino ad un massimo di 17x17 pixel.

3.2. Fase 2: Matching

La seconda fase dell'algoritmo consiste nel riconoscimento vero e proprio dell'etichetta che scorre sul rullo, mediante analisi dell'immagine in un intorno di ciascuno dei pattern memorizzati in fase di *teaching*, valutazione della misura di similarità, individuazione del punto di taglio sulla base dell'offset calcolato nella fase precedente e infine alcuni controlli di integrità sul singolo elemento della bobina.

Come già ampiamente discusso nei capitoli precedenti, uno dei requisiti fondamentali per questo sistema è la velocità. Durante la precedente fase di *teaching* si sono ottenute informazioni fondamentali che permettono di ridurre notevolmente il carico di lavoro. In primo luogo, confrontare immagini di dimensioni che vanno da 9 a 17 pixel è evidentemente molto meno oneroso rispetto ad un'analisi dell'intera immagine. Inoltre, conoscendo la posizione dei pattern, si ha la possibilità di analizzare solo un'area localizzata con precisione. Avendo memorizzato le zone di interesse in ordine di coordinata orizzontale (x), è possibile effettuare il *pattern matching* di un elemento alla volta, evitando sovrapposizioni e risparmiando risorse. Nel momento in cui la telecamera lineare inizia la scansione dell'etichetta, viene aperta una finestra orizzontale di lettura localizzata in prossimità della coordinata verticale (y) del primo pattern da individuare. Vengono quindi bufferizzate un numero di linee di pixel pari alla dimensione della zona di interesse da riconoscere e viene eseguito un comune procedimento di *pattern matching* tra l'immagine corrente e il pattern che deve essere individuato, utilizzando ancora la cross-correlazione (NCC^[6]). Quando la similarità tra i due elementi supera il

valore di soglia, significa che il pattern è stato riconosciuto, se ne memorizza quindi la coordinata orizzontale (x), si chiude la finestra di lettura corrente e si apre quella che corrisponde al pattern successivo.

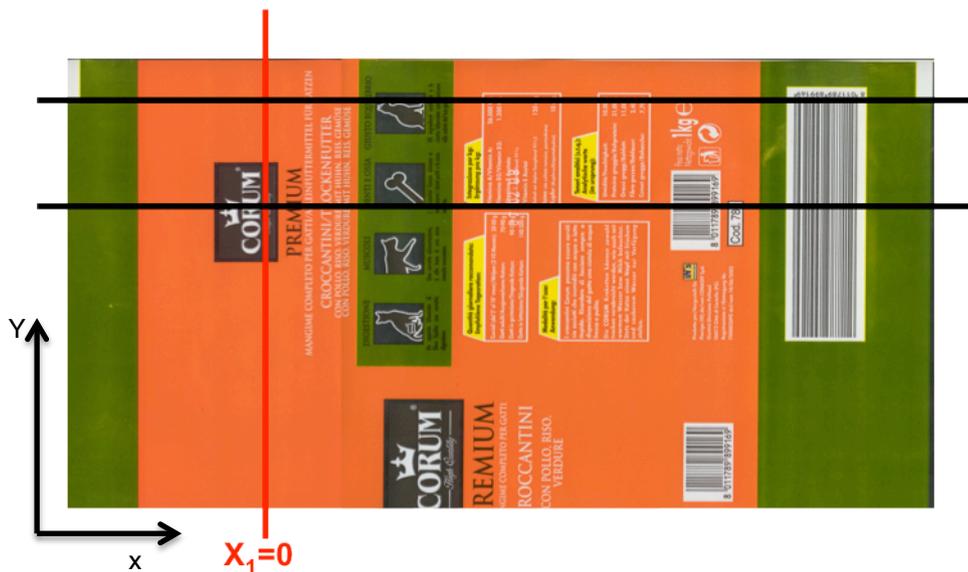


Figura 3.7 – Matching: primo pattern

In figura 3.7 sono riportati in azzurro i pattern da individuare, le linee orizzontali nere rappresentano la finestra di lettura del primo pattern, la linea verticale rossa rappresenta la coordinata orizzontale del pattern memorizzata per i controlli di integrità che verranno descritti in seguito. Come già accennato in precedenza, il primo pattern viene assunto come riferimento per i successivi, in modo da poter effettuare tutte le elaborazioni necessarie usando distanze relative, con dimensione ridotta. Quando anche l'ultimo pattern viene individuato, si utilizza l'offset ricavato dalla fase di *teaching* per ottenere la coordinata orizzontale (x) dello *switching point* ed inviare quindi alla macchina l'impulso di taglio, come mostrato in figura 3.8.

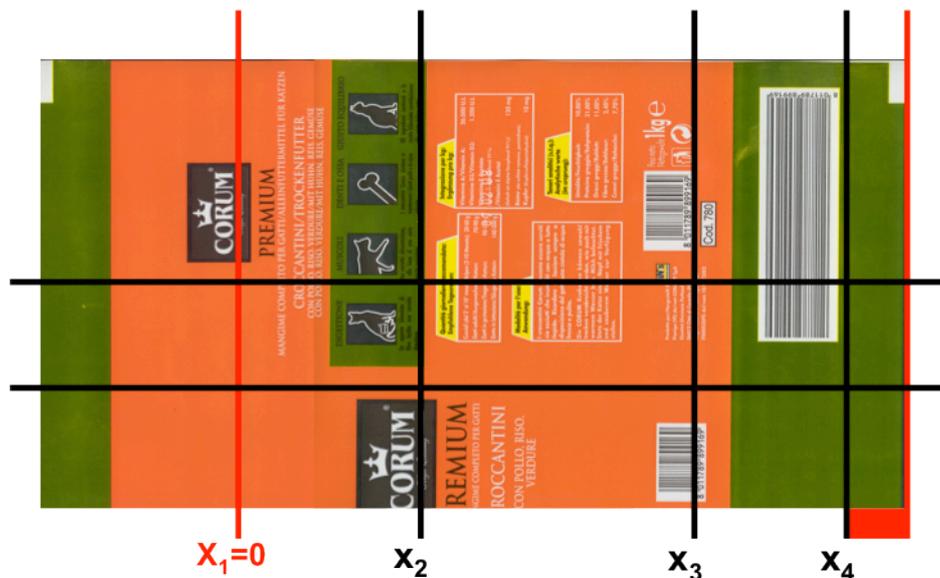


Figura 3.8 – Matching: ultimo pattern, offset, switching point

3.2.1. **Analisi delle problematiche**

Sin dalle analisi preliminari, si è preso in considerazione il caso di mancato rilevamento di un pattern durante la fase di *matching*. Uno degli intenti principali dell’algoritmo proposto è infatti quello di evitare la propagazione degli errori da un’etichetta alla successiva. Come già spiegato in precedenza, i pattern vengono ricercati in maniera sequenziale, aprendo una finestra di lettura in corrispondenza della posizione verticale del pattern corrente. Se questo, per qualsiasi ragione, non viene rilevato, l’approccio sequenziale causa una situazione anomala: il sistema continuerà a cercare lo stesso elemento fino a quando non lo troverà, presumibilmente nell’etichetta successiva. Questo significa, però, che due elementi della bobina vengono erroneamente classificati come non conformi, in base ai controlli di integrità che verranno descritti nel paragrafo 3.3. Per evitare questa situazione, si è pensato di mantenere sempre aperta la finestra di lettura dell’ultimo pattern, in modo da individuare sempre lo *switching point*. Tale approccio ha indubbiamente

migliorato il disaccoppiamento tra imballaggi adiacenti, ma non ha risolto il problema dei “*missing*”. Durante i primi test, infatti, è emersa una significativa instabilità in termini di numero di pattern rilevati con successo durante la fase di *matching*, pur lavorando con immagini identiche. Si è effettuata di conseguenza una minuziosa analisi in merito alle cause del fenomeno e ai possibili miglioramenti da introdurre per ottenere risultati soddisfacenti. I parametri che maggiormente influiscono sui risultati sono: la qualità dei pattern individuati in fase di *teaching*, la robustezza della misura di similarità utilizzata per il procedimento di *pattern matching*, i valori di soglia.

Per quanto riguarda il primo elemento, un’ampia trattazione è stata presentata nei paragrafi precedenti riguardanti la fase di *teaching*, mentre per garantire una maggiore robustezza è stata implementata una soluzione che prevede l’utilizzo di ZNCC^[7] come misura di similarità. Svitati test sono stati indispensabili per individuare valori di soglia in grado di fornire i risultati desiderati.

Infine, le ultime problematiche che sono state valutate riguardano i fenomeni di *stretching* e *overlapping*, presentati nel Capitolo 2. La gestione del primo è stata assegnata ai controlli di integrità, mentre per il secondo non sono state necessarie elaborazioni ad-hoc, in quanto l’algoritmo è in grado di gestire la situazione in maniera efficiente.

3.2.2. Soluzioni proposte

Allo scopo di aumentare l’indipendenza tra imballaggi adiacenti, la finestra di lettura dell’ultimo pattern rimane sempre aperta. Questo significa che, indipendentemente dalla sequenza, quando l’ultimo pattern viene individuato scatta il calcolo dello *switching point*, con conseguente invio dell’impulso di taglio. In tal modo, anche se non sono state rilevate tutte le zone di interesse, l’etichetta viene comunque tagliata, e solo un

imballaggio verrà segnalato come anomalo. L'elaborazione dell'elemento successivo risulterà dunque del tutto indipendente dal precedente.

Resta da analizzare come si comporta il sistema nel caso in cui sia proprio l'ultimo pattern a fallire il *matching*: il comportamento è del tutto analogo a quello descritto in precedenza, ed è stato rappresentato in figura 3.9.



Figura 3.9 – Missing dell'ultimo pattern

La struttura dell'algoritmo, in questo caso, non permette di accorgersi in tempi brevi dell'anomalia: l'esecuzione procede mantenendo aperta soltanto la finestra di lettura dell'ultimo pattern, e causando quindi un malfunzionamento che coinvolge due imballaggi. Si potrebbe comunque considerare l'introduzione di un controllo aggiuntivo sulla base del tempo che intercorre tra due successive individuazioni dell'ultimo pattern.

Per quanto riguarda la percentuale di *hit* che l'algoritmo è in grado di ottenere, è stato necessario effettuare alcuni test preliminari allo scopo di valutare quale fosse il valore di soglia ottimale da utilizzare per NCC^[6]. Non riuscendo però ad ottenere risultati soddisfacenti, si è deciso di implementare una soluzione che facesse ricorso ad una misura di similarità più robusta, valutando se il rallentamento dovuto al maggiore onere computazionale potesse essere tollerabile. Dopo alcuni esperimenti,

si è deciso di lasciare all'utente la possibilità di scegliere quale misura di similarità utilizzare per la specifica immagine campione.

Infine, per quanto riguarda il fenomeno di *overlapping*, sono stati presi in esame entrambi i casi possibili, riportati in figura 3.10: una sovrapposizione che nasconde una parte di etichetta, eliminando uno dei pattern da individuare, oppure una sovrapposizione che causa una replica dello stesso pattern in due posizioni distinte.

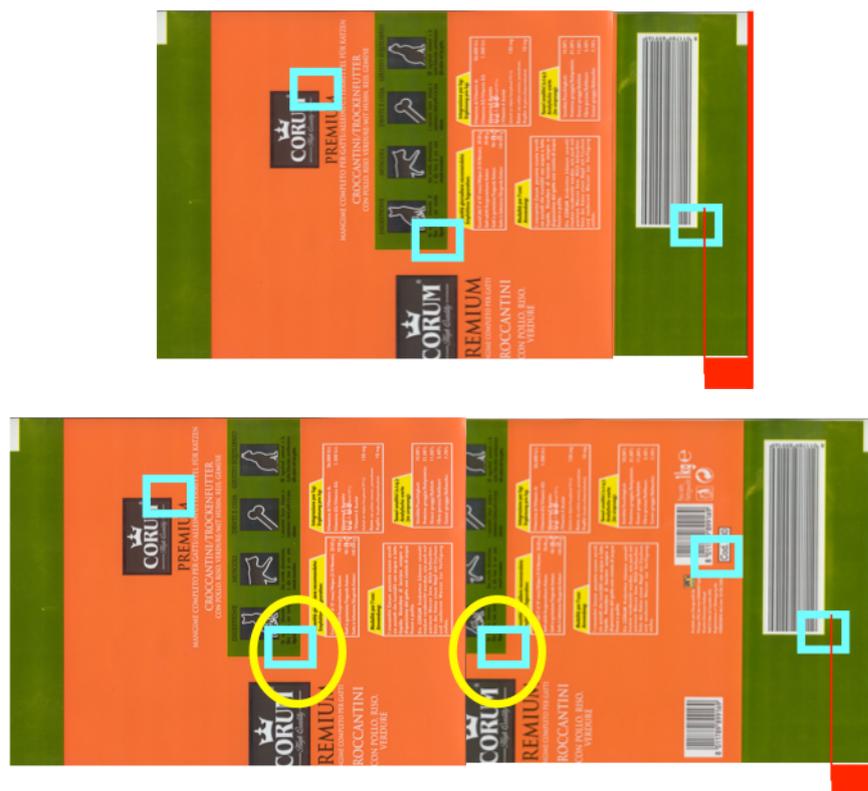


Figura 3.10 – Overlapping

In entrambi i casi l'algoritmo si dimostra in grado di individuare correttamente il punto di taglio, accorgendosi dell'anomalia grazie ai controlli di integrità che saranno descritti nel paragrafo successivo.

3.3. **Controlli di integrità**

Un importante valore aggiunto che l'algoritmo proposto in questa tesi è in grado di fornire è dato dai *controlli di integrità* delle etichette. Si tratta di semplici verifiche sui pattern individuati, che permettono di effettuare una valutazione preliminare in merito alla qualità dei singoli elementi della bobina. Infatti, se uno o più pattern non sono stati riconosciuti, è possibile che l'etichetta presenti dei difetti proprio in corrispondenza delle zone di interesse, e potrebbe quindi essere direttamente scartata o almeno segnalata allo scopo di valutarne poi la qualità.

I controlli proposti consistono nella stima del numero di *hit* di ciascun pattern all'interno della singola etichetta: nel caso in cui un elemento venga riconosciuto più volte si segnala l'anomalia, così come nel caso in cui uno o più pattern non vengano del tutto rilevati. Inoltre, è stata introdotta anche una valutazione delle distanze relative tra i pattern, in modo da gestire al meglio il fenomeno dello *stretching*. Considerando i rapporti tra le distanze relative esistenti tra i pattern, e dimensionando opportunamente un valore di tolleranza, si è infatti in grado di gestire correttamente il caso di uno "stiramento" orizzontale uniforme dell'etichetta.

La figura 3.11 nella pagina successiva mostra come sono valutate le distanze tra i pattern: definendo x_t le coordinate memorizzate in fase di *teaching* e x_m quelle rilevate in fase di *matching*, il controllo di integrità prevede che il rapporto tra le coordinate orizzontali di due pattern consecutivi individuati in fase di *teaching* (x_t) sia simile allo stesso rapporto utilizzando le coordinate relative alla fase di *matching* (x_m), a meno di un valore di tolleranza opportunamente stabilito.

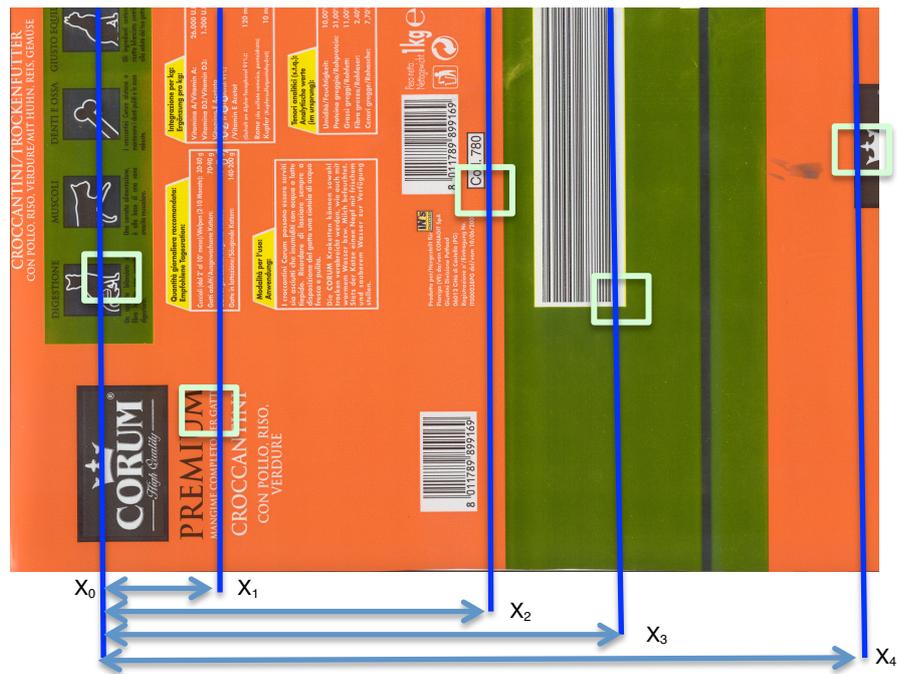


Figura 3.11 - distanze relative tra i pattern

Lo stesso principio viene utilizzato per il calcolo dello *switching point*, in modo da rendere l'algoritmo sufficientemente tollerante al fenomeno dello *stretching* anche nell'individuazione del punto di taglio.

Nel capitolo successivo saranno presentati i primi test effettuati per valutare robustezza e affidabilità del sistema, in base ai quali sono state intraprese alcune importanti scelte progettuali, come già accennato in precedenza.

4. Test e risultati sperimentali

Le problematiche riportate nel capitolo precedente sono state analizzate, nella maggior parte dei casi, durante la fase di studio preliminare per il progetto dell'algoritmo. Allo scopo di valutare le diverse possibilità di scelta a livello implementativo, molti test sono stati effettuati parallelamente allo sviluppo del codice, influenzando le scelte di progetto e le valutazioni che hanno portato alla realizzazione dell'algoritmo nella sua versione finale.

Il primo elemento che occorre sottolineare è l'ampiezza del *dataset* di immagini a disposizione. La possibilità di lavorare con imballaggi di svariati materiali, colori e dimensioni ha fornito un significativo valore aggiunto alle numerose prove sul campo effettuate per valutare la robustezza dell'algoritmo ed il comportamento del sistema. Questo ha permesso di individuare gli elementi di maggiore criticità, ed identificare di conseguenza eventuali soluzioni o miglioramenti, proseguendo quindi nella creazione della migliore strategia di elaborazione delle etichette.

Per quanto riguarda la strumentazione utilizzata per i test, durante la fase iniziale si è fatto ricorso ad immagini ad alta risoluzione acquisite tramite scanner, in un secondo momento si è deciso di simulare il comportamento di una telecamera lineare utilizzando una webcam e scomponendo l'immagine acquisita in singole colonne di pixel tramite un semplice modulo software. In questo modo sono emerse tutte le problematiche dovute all'influenza dell'ambiente esterno, quali differenze di luminosità e spostamento delle zone di interesse rispetto alla loro posizione nominale.

Inoltre, allo scopo di lasciare all'utilizzatore la possibilità di configurare facilmente tutti i parametri necessari per l'esecuzione dell'algoritmo, è stata implementata una semplice interfaccia grafica, con l'ausilio delle librerie QT. In tal modo è possibile esaminare i risultati intermedi prima di

procedere alla fase di *matching*, ed eventualmente valutare un possibile compromesso tra le proprie esigenze e le prestazioni che il sistema è in grado di fornire per la specifica immagine in esame. In figura 4.1 è mostrata la schermata che si presenta all'utente all'avvio del programma, affiancata all'immagine campione su cui lavorare, acquisita da file o da webcam.

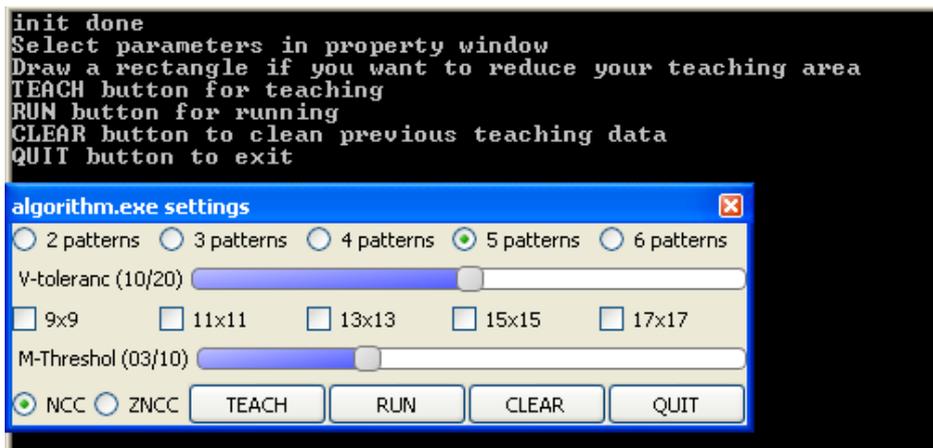


Figura 4.1 – Interfaccia grafica e istruzioni per l'uso

Nei paragrafi che seguono, saranno analizzati nel dettaglio tutti i test effettuati parallelamente allo sviluppo dell'algoritmo e i relativi risultati, allo scopo di giustificare le principali scelte progettuali che hanno portato all'implementazione finale del sistema.

4.1. Immagini acquisite tramite scanner

Per comprendere le principali problematiche da affrontare nel progetto, occorre innanzitutto avere un'idea degli imballaggi che dovranno essere trattati dal sistema in esame. Per questa ragione, il primo passo è stata la creazione di un dataset di immagini campione, da utilizzare per i primi test. Tali immagini sono state acquisite tramite scanner, pertanto sono caratterizzate da dimensioni limitate, risoluzione elevata, eliminazione dei problemi riguardanti superfici trasparenti o riflettenti. È importante però precisare che tali problematiche non sono da considerarsi irrilevanti, anzi si sono dimostrate molto critiche ed hanno richiesto analisi approfondite, ma sono state prese in considerazione in un secondo momento e saranno trattate nel paragrafo successivo.

I primi test si sono focalizzati sul dimensionamento del valore di soglia per la misura di similarità. Esaminando i valori di similarità tra le patch individuate durante la fase di *teaching*, è stato possibile analizzare quanto il valore di soglia utilizzato per il *pattern matching* influisca, in combinazione con la dimensione del descrittore, sul numero di pattern che l'algoritmo è in grado di individuare. In figura 4.2 sono riportati i grafici relativi a 5 diverse immagini campione, per le quali sono stati registrati, al variare delle dimensioni delle patch, i valori di similarità tra le coppie analizzate in fase di *teaching*. La prima particolarità che salta all'occhio è che l'andamento dei grafici varia in relazione all'immagine campione. Questo è dovuto al fatto che in alcuni casi, in particolare per le immagini denominate "voli" e "montere", il contenuto del campione è volutamente ridondante, ovvero l'immagine contiene quasi per intero due elementi della bobina di etichette. In tal modo è stato possibile valutare il comportamento del sistema anche in casi particolarmente sfavorevoli, in cui le patch sufficientemente distintive fossero difficili da individuare.

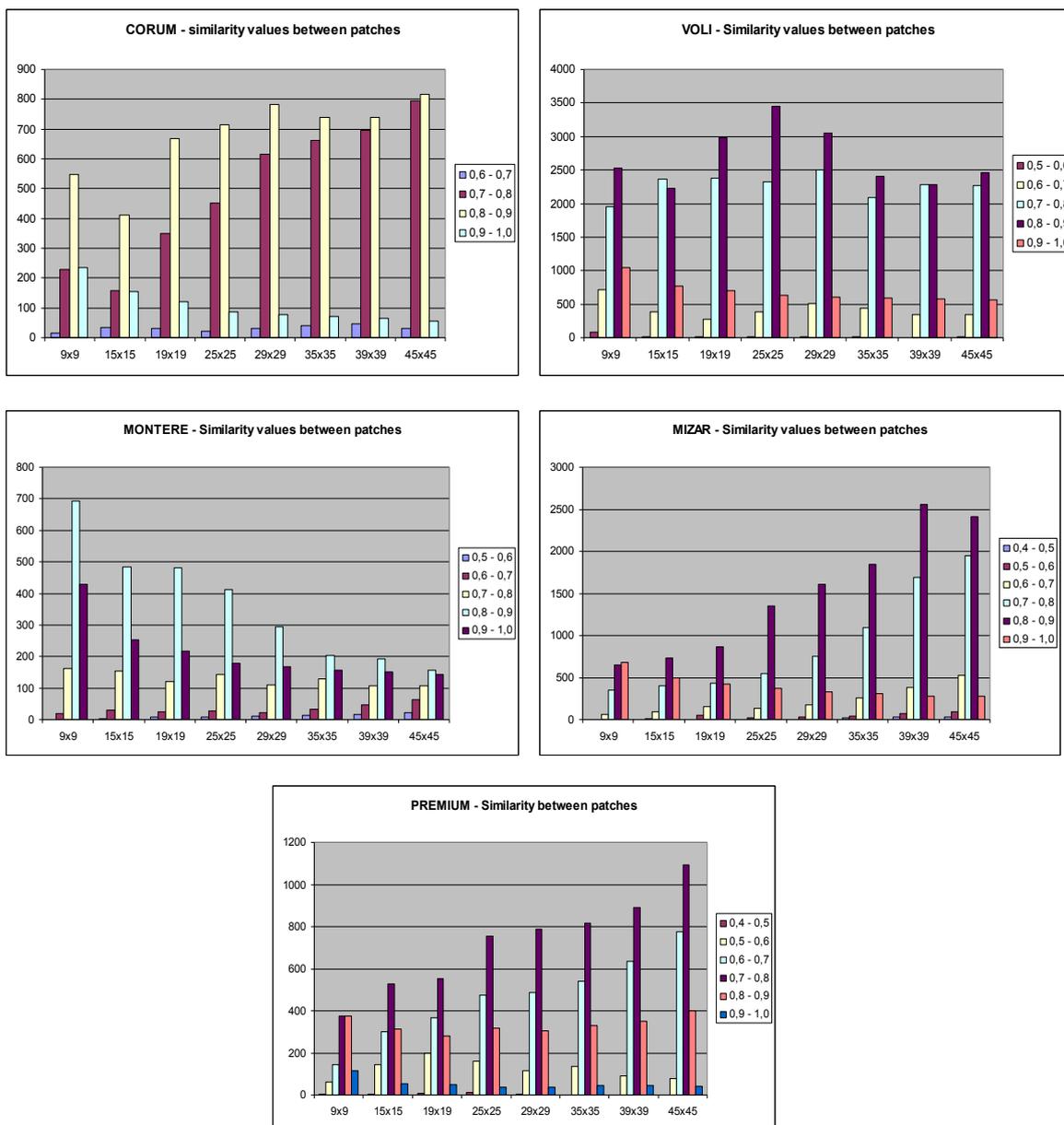


Figura 4.2 – similarità tra le coppie di patch

In generale, i grafici confermano che l'aumento della dimensione dei descrittori influisce sul valore di similarità registrato tra le coppie. In ogni caso, tale valore difficilmente raggiunge quote inferiori a 0.7, pertanto la soglia dovrà necessariamente essere piuttosto elevata per poter ottenere un numero sufficiente di pattern al termine della prima fase dell'algoritmo. Con un valore inferiore a 0.9, infatti, quasi tutte le patch sarebbero

eliminate dal confronto reciproco, e le restanti potrebbero non essere considerate idonee dai controlli successivi. Anche il numero totale di patch varia a seconda delle dimensioni, a causa della necessità di gestione delle distanze tra i singoli elementi per evitare sovrapposizioni, di cui si è già ampiamente discusso nel capitolo precedente.

In un secondo test l'attenzione si è focalizzata sulla quantità di patch accettate e rifiutate dalla post-elaborazione della fase di *teaching*, anche in questo caso al variare delle dimensioni. In figura 4.3 sono riportati i grafici per le diverse immagini campione, impostando una soglia di 0.9 per la misura di similarità, sulla base dei risultati del test mostrato in precedenza.

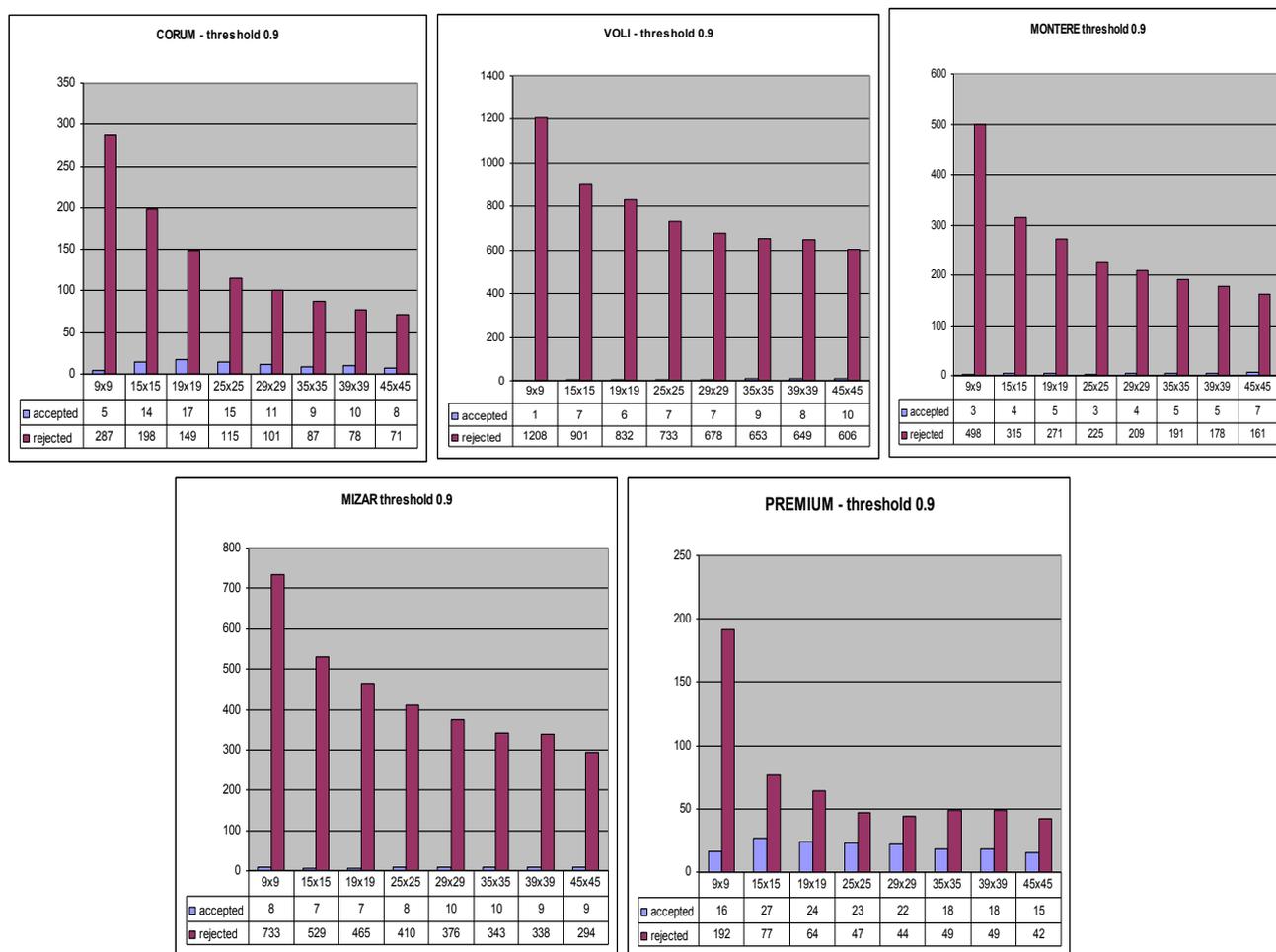


Figura 4.3 – Patch accettate e rifiutate dalla fase di *teaching*

L'elemento di maggior risalto nei grafici riportati in figura 4.3 è la differenza in termini quantitativi tra il numero di patch accettate e rifiutate. In particolare, per le immagini denominate “voli”, “montere” e “mizar” le colonne dei descrittori che superano i controlli sono quasi invisibili. In termini percentuali, si è registrato un leggero aumento delle patch accettate in funzione della dimensione, da ricondursi alla migliore caratterizzazione di descrittori più grandi. Come si può notare dai grafici, infatti, le colonne che rappresentano gli elementi rifiutati tendono a diminuire con l'aumento della dimensione delle patch, sebbene i miglioramenti in termini di descrittori accettati non siano apprezzabili visivamente. Il divario tra le due colonne rimane comunque significativo per tutte le dimensioni e tutte le immagini campione: la maggior parte delle patch viene scartata in fase di *teaching*.

A questo punto occorre focalizzare l'attenzione proprio sulle patch rifiutate dall'algoritmo, in quanto considerate non idonee. La post-elaborazione descritta nel capitolo precedente è costituita da diversi controlli, è quindi importante capire quale di questi incide maggiormente sulla selezione. Confrontando il numero di patch rifiutate dalla comparazione delle coppie di descrittori con quelle eliminate dal *pattern matching* all'interno della sezione orizzontale di appartenenza, è stato possibile valutare quale fosse l'elemento più critico della post-elaborazione, permettendo quindi di intervenire con miglioramenti mirati. In figura 4.4, nella pagina successiva, sono rappresentati in marrone gli elementi rifiutati dalla comparazione tra coppie di patch, in azzurro quelli eliminati nel procedimento di *pattern matching*.

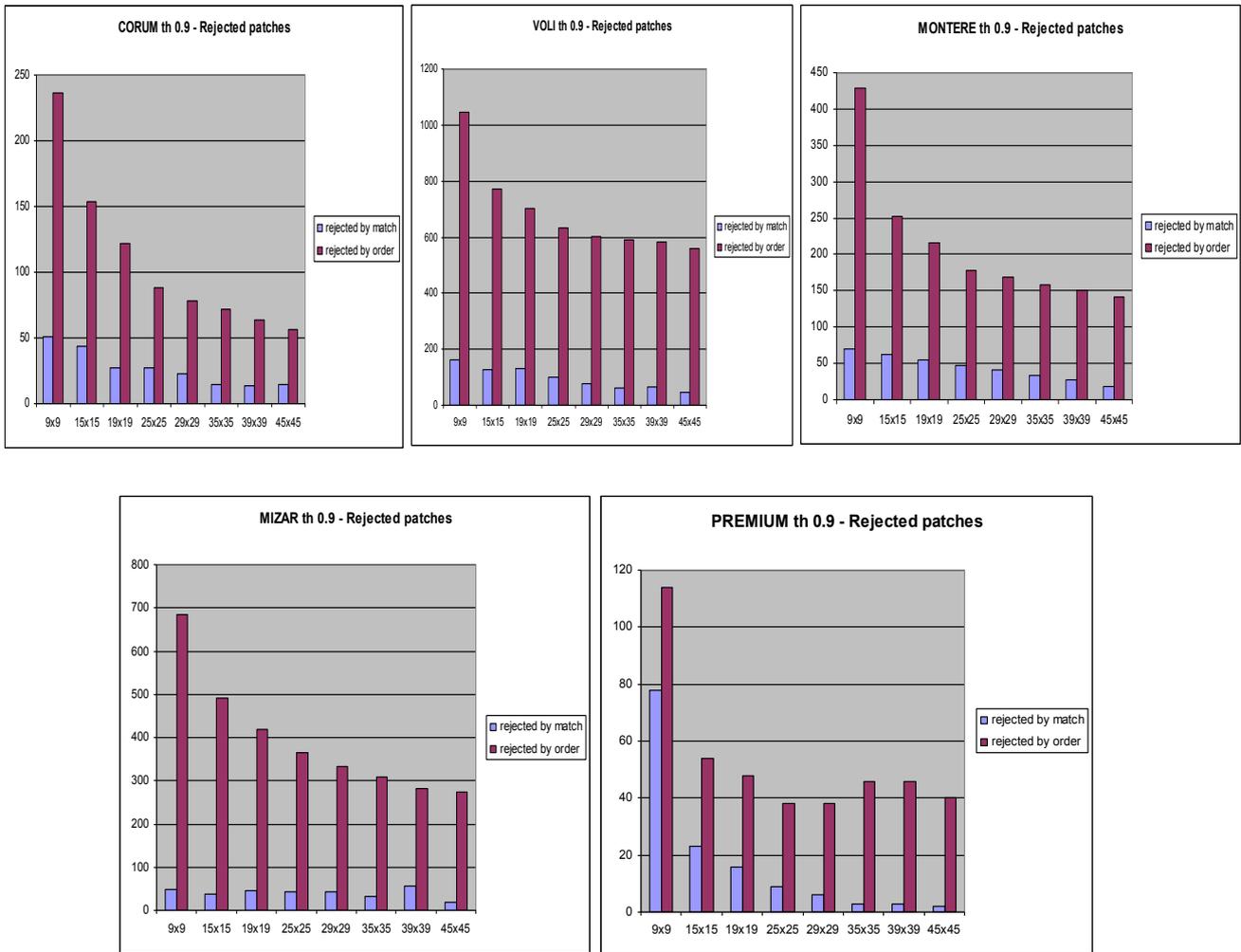


Figura 4.4 – Patch scartate in fase di teaching

Risulta quindi evidente che la fase più critica è la comparazione tra le coppie di descrittori, in termini di misura di similarità. Per gestire al meglio questo fenomeno è importante valutare bene la misura di similarità da utilizzare ed il relativo valore di soglia.

L'elemento fondamentale su cui si focalizza l'intero algoritmo, tuttavia, è il numero di pattern individuati al termine del *teaching*, a seguito di tutti i controlli e le verifiche descritti nel capitolo 3. E' dunque indispensabile esaminare in termini quantitativi le prestazioni della prima fase

dell'algoritmo, valutando, al variare della dimensione delle patch, quanti sono i descrittori che potranno essere utilizzati in fase di *matching*.

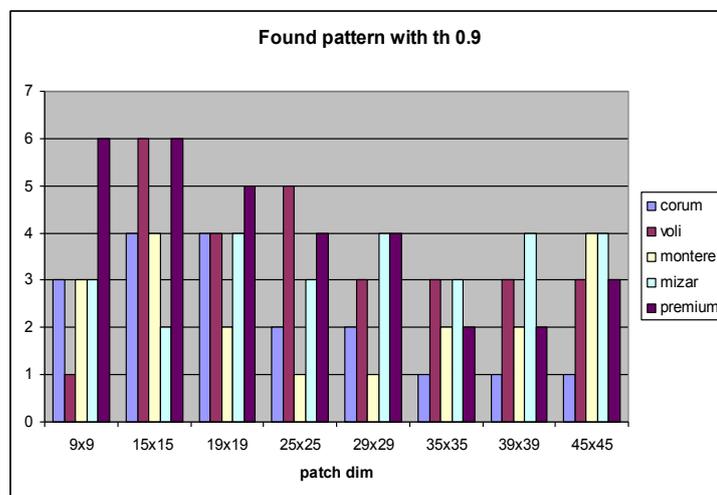


Figura 4.5 – Numero di pattern al termine della fase di teaching. Soglia: 0.9

Le figure 4.5, 4.6, 4.7 riportano il numero di pattern individuati dall'algoritmo per tutte le immagini campione. Come si può facilmente notare, tali valori non dipendono in modo significativo dalle dimensioni, quanto piuttosto dalle peculiarità di ciascuna immagine. Lo stesso test è stato ripetuto usando valori di soglia leggermente diversi, esaminandone l'influenza effettiva al termine dell'intera fase di *teaching*.

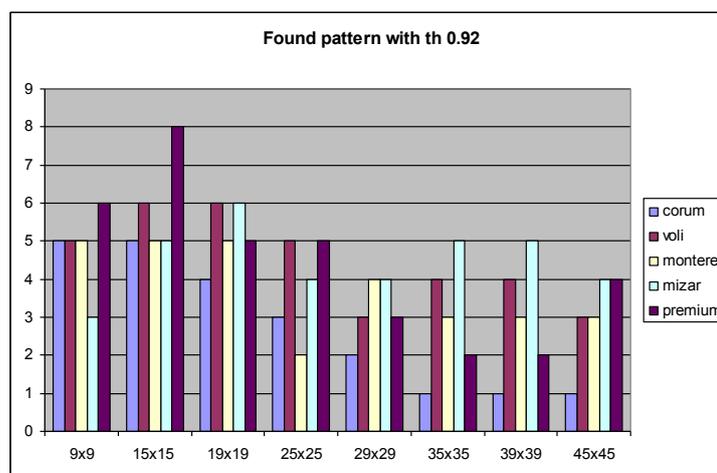


Figura 4.6 - Numero di pattern al termine della fase di teaching. Soglia: 0.92

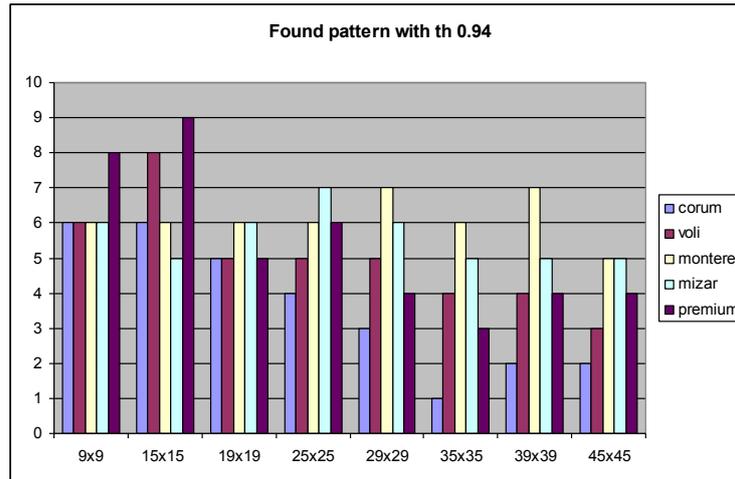


Figura 4.7 - Numero di pattern al termine della fase di teaching. Soglia: 0.94

L'elemento comune a tutti i test fin qui descritti è la variazione della dimensione delle patch. Uno degli obiettivi principali di tutti questi esperimenti, infatti, era la valutazione della grandezza ottimale, in modo da ottenere risultati adeguati per tutte le immagini campione. Dai grafici si evince che dimensioni elevate non sono accettabili, in quanto nel complesso il numero finale di pattern risulta in molti casi insufficiente. Dimensioni ridotte permettono invece di ottenere risultati finali accettabili, seppure al limite, ma comportano un maggiore overhead nella post-elaborazione della fase di *teaching*. Sulla base di questi dati, in definitiva, è stata presa in considerazione l'idea di utilizzare patch di dimensioni variabili, come già accennato nel capitolo 3.

4.2. Immagini acquisite tramite webcam

Le immagini campione utilizzate per i primi test sono state molto utili per analizzare il comportamento della fase di *teaching* del sistema, valutando la dimensione delle patch, il numero di pattern finali, il carico di lavoro ed i risultati ottenuti al termine della prima fase dell'algoritmo. Al fine di

ottenere elementi utili anche alla successiva fase di *matching* è stato indispensabile lavorare con immagini diverse, acquisite tramite webcam. Inoltre, allo scopo di simulare il comportamento di una telecamera lineare, è stato realizzato un semplice modulo software per la scomposizione dell'immagine in singole colonne di pixel, analizzate una per volta in sequenza, simulando così lo scorrimento della bobina di etichette sul rullo. Grazie a queste condizioni di lavoro, sono emerse diverse problematiche riguardanti la fase di *matching*: le variazioni di luminosità, i problemi derivanti da materiali riflettenti, il potenziale spostamento della posizione dei pattern da individuare nel caso in cui l'immagine non sia identica al campione trattato in fase di *teaching*. In figura 4.8 sono riportate due delle immagini acquisite tramite webcam ed utilizzate per i test che verranno descritti nei prossimi paragrafi.



Figura 4.8 – Immagini acquisite tramite webcam

Si può facilmente notare quanto le condizioni di illuminazione possano influenzare le prestazioni dell'algoritmo. Nelle immagini riportate in figura 4.8, infatti, se uno dei pattern viene individuato dalla fase di *teaching* in corrispondenza di una zona caratterizzata da una particolare riflessione di luce, ad esempio quelle nei cerchi gialli, la successiva fase di *matching* non sarà probabilmente in grado di riconoscerla se le condizioni di

illuminazione non sono esattamente identiche. Questo fenomeno influenza in maniera significativa le prestazioni dell’algoritmo, pertanto sono stati introdotti vincoli in termini di condizioni di luminosità da rispettare nell’ambiente di lavoro della telecamera lineare, ma al contempo è stata implementata una versione dell’algoritmo che sfrutta una misura di similarità più robusta alle variazioni di intensità luminosa: la *Zero-mean Normalized Cross-Correlation*^[7].

Al fine di esaminare le prestazioni del sistema in condizioni particolarmente sfavorevoli, per una serie di test sono state utilizzate immagini che presentano alcune anomalie, quali ad esempio traslazioni orizzontali, *smoothing*, rumore uniforme, rotazione dell’etichetta, distorsione con conseguente variazione dell’effetto della luce sul materiale riflettente. La robustezza alle traslazioni orizzontali è ovviamente implicita nella struttura stessa dell’algoritmo. Per quanto riguarda le altre anomalie, di seguito saranno fornite alcune immagini di riferimento per una più corretta valutazione dei risultati ottenuti.

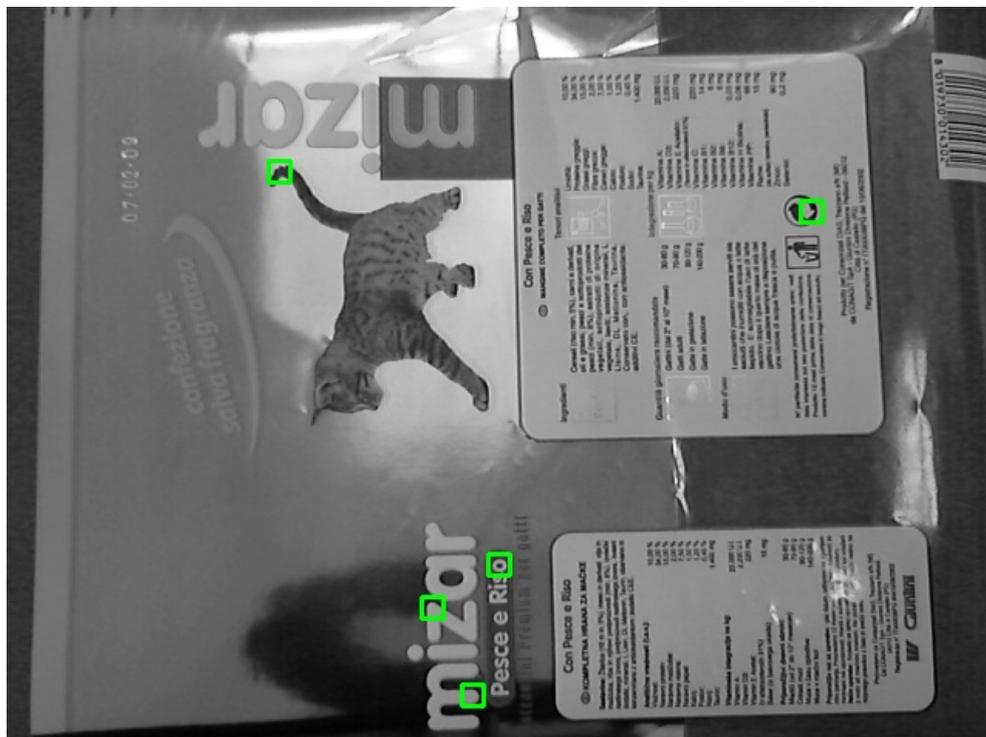


Figura 4.9 - Teaching

In figura 4.9 sono mostrati i pattern individuati al termine della fase di *teaching*. Si analizzerà ora la fase di *matching*, applicata alle immagini affette dalle diverse anomalie descritte in precedenza.

Per il primo test è stato introdotto un semplice filtro di *smoothing*. I risultati, come si può notare in figura 4.10, sono soddisfacenti, sebbene siano ovviamente dipendenti dall'entità del filtraggio. In termini operativi, tuttavia, si consideri che la telecamera lineare dovrà necessariamente essere posizionata ad una distanza ridotta dal rullo di etichette che deve inquadrare, pertanto lo sfocamento dovuto ad errori di posizionamento, spostamento o taratura non potrà essere eccessivo.



Figura 4.10 - smoothing

La figura 4.11, nella pagina successiva, è invece affetta da rumore uniforme. Anche in questo caso le prestazioni dipendono fortemente dall'entità del disturbo, ma per l'esempio riportato in figura, in cui il

degrado dell'immagine è visibilmente apprezzabile, i risultati in termini di pattern riconosciuti sono comunque soddisfacenti.

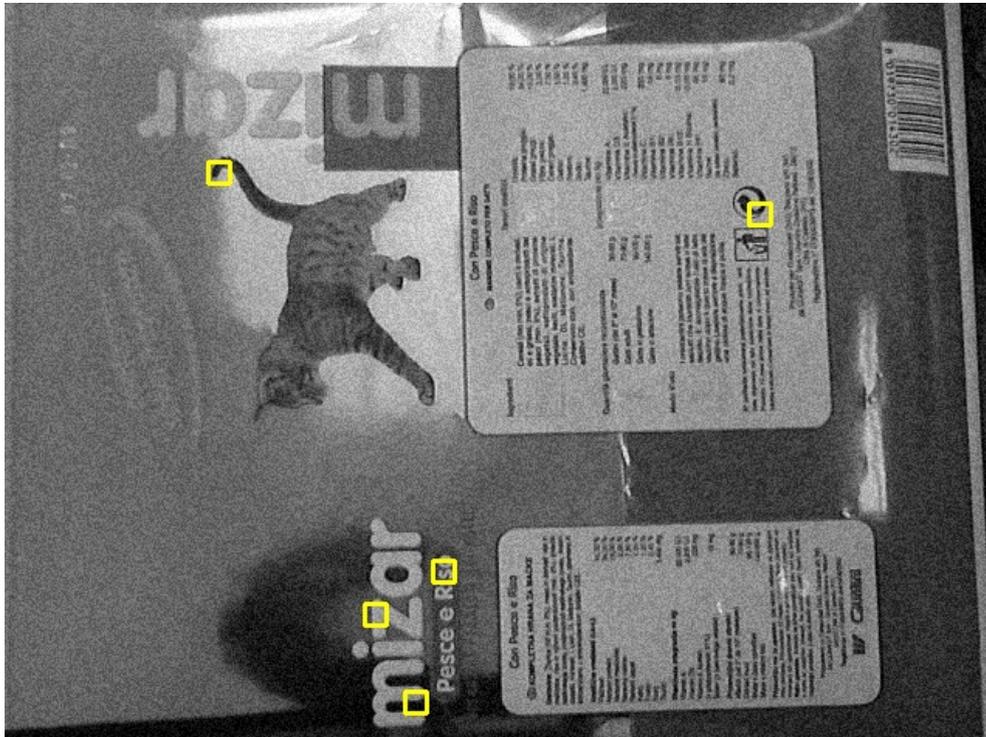


Figura 4.11 - Noising

Un'altra situazione anomala che potrebbe verificarsi nell'ambiente di lavoro del sistema è data dallo spostamento della bobina sul rullo, che causa una leggera rotazione delle immagini registrate dalla telecamera lineare. In tal caso l'immagine è qualitativamente buona, ma le posizioni dei pattern sono diverse rispetto all'immagine campione. Questo fenomeno può quindi causare problemi in fase di *matching*, nel caso in cui il pattern fuoriesca, del tutto o in parte, dalla sezione orizzontale in cui se ne effettua la ricerca. Per gestire al meglio questo problema occorre quindi dimensionare opportunamente le sezioni orizzontali, introducendo una tolleranza adeguata rispetto alla posizione nominale rilevata in fase di *teaching*. In figura 4.12 è riportato un esempio in cui l'ultimo pattern non

viene riconosciuto. Una semplice modifica della sezione orizzontale di ricerca è stata sufficiente per migliorare i risultati ottenuti.

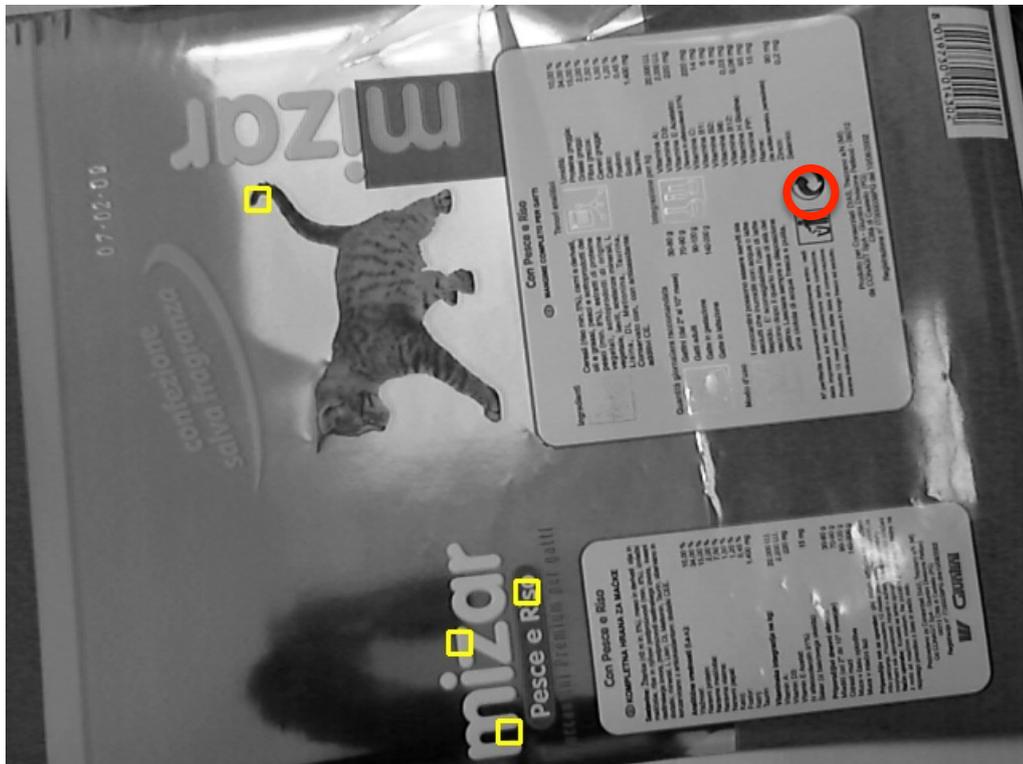


Figura 4.12 – Rotazione

L'elemento più critico per il sistema in esame è sicuramente dato dall'illuminazione dell'ambiente di lavoro. Il materiale riflettente di cui è costituito l'imballaggio, per esigenze di marketing, influenza significativamente le prestazioni dell'algoritmo. E' stato sufficiente ripiegare leggermente l'etichetta per ottenere un'immagine visibilmente molto diversa dal campione utilizzato per il *teaching*, in cui le variazioni in scala di grigi dovute alle riflessioni non sono in alcun modo riconducibili all'immagine campione. In figura 4.13 sono mostrati i risultati della fase di *matching*: l'algoritmo non è in grado di supportare variazioni di intensità luminosa così evidenti. L'unico miglioramento che si può pensare di

introdurre è una misura di similarità più robusta: la ZNCC^[7]. Le prestazioni dell'algoritmo complessivamente migliorano, ma quella dell'illuminazione resta una delle principali problematiche che affliggono il sistema.

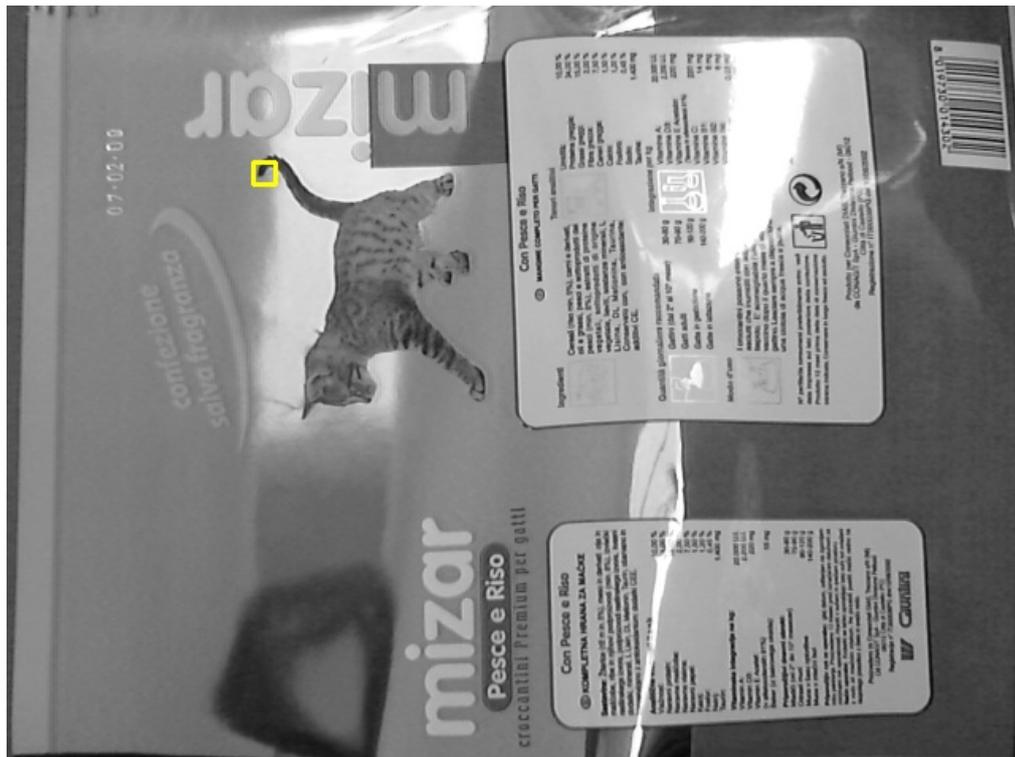


Figura 4.13 – Bending e variazione di intensità luminosa

L'implementazione della nuova misura di similarità ha richiesto un attento esame in merito al rapporto tra il miglioramento in termini di prestazioni e l'aumento di tempo richiesto per le elaborazioni. La modifica è stata inserita in entrambe le fasi dell'algoritmo, ed è stato necessario ricorrere ad un processo di validazione per individuare un valore di soglia adeguato. In particolare, i primi test effettuati con lo stesso valore utilizzato per la NCC^[6] avevano evidenziato un numero di fallimenti nel riconoscimento dei pattern pari al 47%. Una volta individuato un valore di soglia opportuno, i risultati si sono invece rivelati buoni ma non eccezionali, pertanto si è

deciso di non eliminare l'implementazione iniziale con NCC^[6], ma lasciare all'utente la possibilità di selezionare la misura di similarità più adatta alle proprie esigenze.

Infine, sono state analizzate le prestazioni del sistema nella sua implementazione finale, focalizzando l'attenzione sui tempi di elaborazione dell'intera fase di *teaching* e sul riconoscimento di ogni singolo pattern all'interno della porzione di immagine di appartenenza. Allo scopo di ottenere una certa stabilità nella scansione delle immagini, in particolare durante la fase di *matching*, in cui occorre simulare lo scorrimento del rullo di etichette sotto la telecamera, è stato costruito un supporto metallico per la webcam, mostrato in figura 4.14, adattabile alle varie tipologie di imballaggi a disposizione per i test. In tal modo è stato possibile eseguire un elevato numero di esperimenti, ricavando dati sufficientemente robusti e ripetibili.

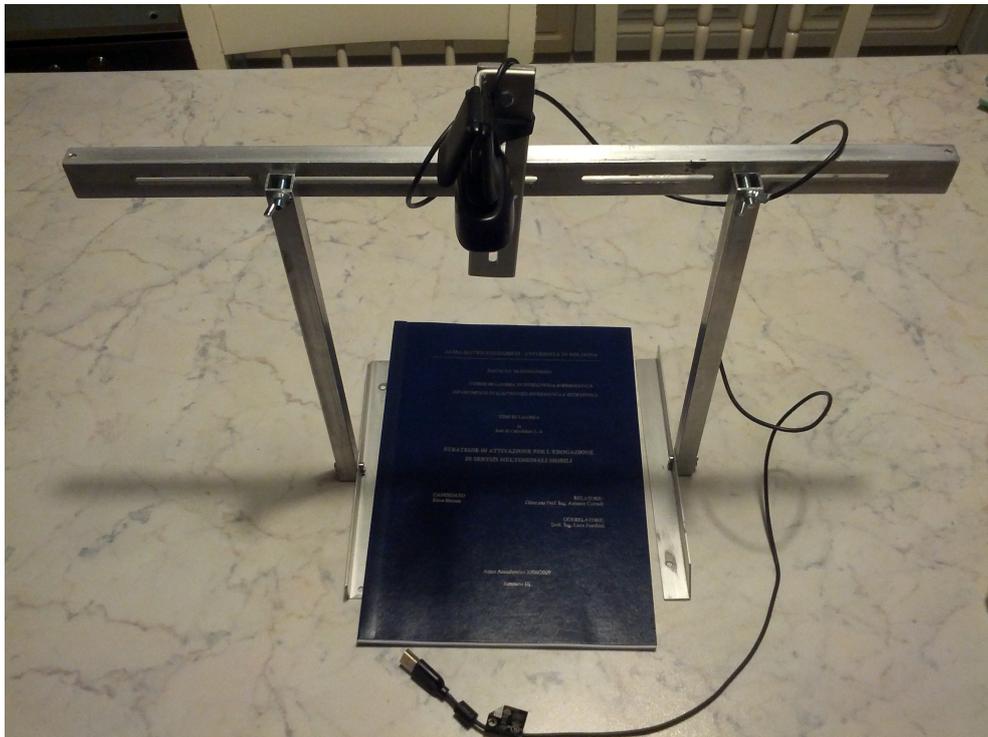


Figura 4.14 – Supporto metallico per i test con webcam

Per quanto riguarda la valutazione dei tempi di esecuzione, i dati rilevati in fase di *teaching* sono piuttosto variabili, in quanto influenzati da diversi elementi fondamentali, quali le dimensioni dell'immagine campione, il pool di dimensioni selezionato, oltre a tutte le altre elaborazioni descritte nel capitolo 3. Per immagini acquisite da webcam, i valori oscillano tra i 230 ms e i 700 ms, con un valore medio intorno ai 400 ms. Per le immagini acquisite tramite scanner, invece, si sono raggiunti valori oltre i 4000 ms. Le tempistiche rilevate in fase di *matching* sono invece indipendenti dalle dimensioni dell'immagine campione: infatti è stato considerato significativo soltanto il tempo necessario all'algoritmo per rendersi conto che all'interno delle colonne di pixel che in un determinato momento si trovano sotto la telecamera lineare, in corrispondenza della sezione di ricerca, il pattern è stato riconosciuto. Questo tempo è un indicatore fondamentale delle prestazioni del sistema, in quanto strettamente legato alla velocità di scorrimento delle bobine sul rullo. I valori registrati durante i test oscillano tra 0.01 ms e 0.6 ms, con una media di 0.209 ms. Occorre tuttavia precisare che tali valori fanno riferimento all'esecuzione dell'algoritmo su PC, un'implementazione su hardware dedicato richiederebbe un'ulteriore e più approfondita analisi dei tempi di esecuzione dell'algoritmo, soprattutto per quanto riguarda la fase di *matching*, in quanto le prestazioni dipendono in maniera significativa dall'hardware a disposizione. I test presentati in questa tesi forniscono una semplice valutazione preliminare. In conclusione, i risultati dei vari esperimenti hanno fornito materiale utile per arrivare ad una implementazione finale soddisfacente in termini di prestazioni. Alcune problematiche, quali il controllo dell'illuminazione, la gestione dei materiali riflettenti, e la robustezza alle variazioni di scala non sono state gestite a livello implementativo. Nel capitolo 5 saranno però analizzate nel dettaglio, al fine di valutare la possibilità di miglioramenti futuri dell'algoritmo.

5. Analisi del sistema e possibili sviluppi futuri

Nei capitoli precedenti sono state descritte diverse problematiche affrontate durante l'implementazione dell'algoritmo, che hanno richiesto test e studi specifici, influenzando le scelte progettuali e, di conseguenza, il funzionamento finale del sistema. Per alcune criticità, tuttavia, non è stato possibile trovare una soluzione definitiva: ad esempio l'influenza delle condizioni di illuminazione sul comportamento del sistema. Inoltre, sono state valutate alcune possibilità di sviluppo al fine di potenziarne l'utilità e i vantaggi per l'utente, quali ad esempio la possibilità di utilizzare per la fase di *teaching* solo una porzione dell'immagine, o la robustezza a variazioni di scala e la possibilità di non dover utilizzare necessariamente immagini campione acquisite con telecamera lineare. Nei paragrafi che seguono tali idee saranno analizzate nel dettaglio, valutando accuratamente costi e benefici di ciascuna possibile soluzione implementativa.

5.1. Condizioni di illuminazione e distorsioni prospettiche

Una delle criticità per la quale non è stato possibile trovare una soluzione definitiva riguarda le differenze di illuminazione, combinate ai materiali riflettenti di cui possono essere costituiti gli imballaggi: tale problematica non può essere gestita unicamente dal software. E' indispensabile introdurre vincoli per quanto riguarda l'ambiente di lavoro, e valutare

l'inserimento di un illuminatore sul dispositivo fisico, in grado di mantenere pressoché invariate le condizioni di illuminazione durante le due fasi dell'algoritmo. Questa seconda soluzione, però, porta alla necessità di valutazione di un ulteriore problema: puntare un illuminatore su un'etichetta come quelle utilizzate in fase di test può causare riflessioni intollerabili dal sistema. La soluzione più comunemente adottata è quella di introdurre un angolo di inclinazione adeguato tra la lampada e la superficie che la telecamera deve inquadrare. A questo punto, però, l'immagine acquisita dalla telecamera sarà affetta da una distorsione prospettica tanto più rilevante quanto maggiore è il suddetto angolo di inclinazione. Occorre quindi analizzare l'effetto di questa distorsione sul comportamento del sistema.

Grazie all'ausilio della webcam, è stato possibile sperimentare come l'inclinazione della telecamera possa influenzare la capacità dell'algoritmo di riconoscere i pattern. Acquisendo l'intera immagine campione per la fase di *teaching* in un singolo scatto, e procedendo alla successiva fase di *matching* facendo fisicamente scorrere le etichette sotto la telecamera, si è notato come alcuni dei fallimenti nel riconoscimento dei pattern fossero dovuti proprio alla variazione di inclinazione della webcam. Occorre tuttavia considerare che il dispositivo di visione che si dovrebbe utilizzare per l'implementazione di questo progetto è una telecamera lineare, che legge una singola linea di pixel per volta. Questo significa che, nel caso in cui l'immagine campione sia anch'essa acquisita tramite lo stesso dispositivo ed ottenuta dall'assemblaggio delle singole colonne, in maniera del tutto analoga alla fase di *matching*, allora la distorsione prospettica sarebbe esattamente identica per entrambe le fasi dell'algoritmo, e di conseguenza ininfluenza. Questo perché il confronto effettuato in fase di riconoscimento prende in considerazione un insieme di pixel appartenenti a colonne adiacenti: se il processo di acquisizione è lo stesso per entrambe le fasi, allora anche l'immagine risultante dalla bufferizzazione

dei suddetti pixel adiacenti sarà del tutto identica, e quindi il riconoscimento avrà esito positivo.

5.2. Variazioni di scala

L'algoritmo proposto in questa tesi non è ovviamente in grado di garantire robustezza alle variazioni di scala. Effettuando un *pattern matching* nella fase di riconoscimento, è infatti indispensabile che le patch dell'immagine campione abbiano la stessa dimensione di quelle da rilevare in fase di *matching*. Questa non è stata inizialmente considerata una limitazione pesante, in quanto una delle ipotesi di partenza riguardava proprio l'acquisizione dell'immagine campione, da effettuarsi tramite la stessa telecamera lineare che opera in fase di riconoscimento. Potrebbe tuttavia essere interessante permettere all'utente di utilizzare anche immagini acquisite in altri modi, o magari già disponibili in formato digitale per svariati motivi. Ad esempio, sarebbe utile poter eseguire la fase di *teaching* sul disegno al CAD dell'etichetta, o magari su un file JPEG. In tale prospettiva occorre ovviamente rivalutare le considerazioni presentate nel paragrafo precedente in merito alle distorsioni prospettiche, che assumerebbero una rilevanza fondamentale, impattando significativamente sulle prestazioni del sistema.

Un algoritmo robusto alle variazioni di scala avrebbe anche il vantaggio di non essere strettamente vincolato al posizionamento del dispositivo fisico: in caso di spostamento accidentale del sensore non sarebbe necessario ripetere la fase di *teaching*. Tale fase, inoltre, potrebbe essere effettuata un'unica volta, magari in una sede diversa rispetto al luogo in cui si procede alla fase di *matching* e al taglio vero e proprio delle bobine di imballaggi. Infine, sarebbe molto interessante avere la possibilità di riutilizzare gli stessi dati acquisiti in fase di *teaching* per etichette uguali

aventi dimensioni diverse, si pensi al caso di prodotti in confezioni *mini* e *maxi*.

Tutte queste considerazioni portano alla necessità di valutare l'implementazione di una versione dell'algoritmo in grado di supportare le variazioni di scala, possibilmente senza stravolgere la struttura del progetto presentato in questa tesi. Esistono in letteratura diversi metodi per ottenere un algoritmo in grado di garantire tale robustezza: si potrebbe pensare di utilizzare già in fase di *teaching* descrittori invarianti per scala, come ad esempio SURF^[5], ma questo richiederebbe l'acquisizione dell'intera immagine anche per la fase di *matching*, con conseguenze intollerabili per i requisiti di velocità di esecuzione del sistema. Allo scopo di mantenere il più possibile invariata la struttura dell'algoritmo, si è deciso allora di focalizzare l'attenzione sulle tecniche di *Fast Template Matching*^[8], ed in particolare sull'*approccio piramidale*. L'idea è quella di memorizzare una piramide per ciascuno dei pattern, ed introdurre una fase intermedia di *calibrazione* per individuare la dimensione corretta da utilizzare per la successiva fase di riconoscimento.

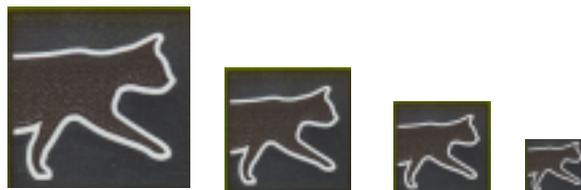


Figura 5.1 – Piramide di pattern

In figura 5.1 è mostrato un semplice esempio per spiegare il funzionamento di un'ipotetica implementazione dell'algoritmo che utilizzi l'approccio piramidale per ottenere robustezza alle variazioni di scala. Per ciascuno dei pattern individuati in fase di *teaching* sarebbe necessario memorizzare un vettore di patch, aventi dimensioni diverse, come

mostrato in figura. Uno dei punti più critici sarebbe proprio la valutazione del numero di immagini da memorizzare e delle loro dimensioni, in quanto da questi dati dipendono i risultati della fase di riconoscimento. A questo punto occorre capire quale di queste sia la dimensione più adatta per la bobina di imballaggi sulla quale deve essere identificato lo *switching point*. Si ritiene pertanto indispensabile introdurre una fase di calibrazione, in cui il *pattern matching* viene eseguito utilizzando una per volta tutte le dimensioni dei pattern, a cominciare da quella inferiore. Una prima idea di implementazione prevede che la fase di riconoscimento venga eseguita prima con tutte le patch di dimensione più piccola, poi con tutte le patch leggermente più grandi, e così via fino ad arrivare ai pattern aventi dimensioni maggiori. In tal modo, però, le tempistiche della fase di calibrazione sarebbero certamente superiori rispetto alla normale acquisizione dei pattern mediante la fase di *teaching*, riducendo drasticamente i vantaggi per l'utente. Si può allora pensare ad una ricerca in parallelo di tutte le dimensioni di ciascun pattern, valutando opportunamente l'estensione delle finestre di ricerca utilizzate in fase di *matching*. E' evidentemente possibile che siano rilevati dei falsi positivi, pertanto occorre un sistema di *voting* per identificare la dimensione che nel complesso ottiene risultati migliori in termini di numero di pattern correttamente identificati.

Al termine della fase di calibrazione, una volta individuata la grandezza delle patch da considerare, si potrebbe procedere normalmente con la fase di *matching*. Il tempo necessario per il passo di calibrazione è teoricamente calcolabile come il prodotto tra il tempo impiegato comunemente per ciascun *match* moltiplicato per il numero di immagini dello stesso pattern, con l'aggiunta di un *overhead* introdotto dall'aumento della dimensione della finestra di ricerca in fase di riconoscimento.

Dopo questa minuziosa analisi, si è deciso di non implementare una versione dell'algoritmo robusta alle variazioni di scala, in quanto computazionalmente onerosa e anche potenzialmente svantaggiosa in termini di velocità di esecuzione del sistema nel suo complesso.

5.3. *Teaching su una porzione di immagine*

Un'altra possibilità di sviluppo del sistema consiste nell'introduzione di *blanking areas* durante la fase di *teaching*, allo scopo di ridurre la zona in cui individuare i pattern. In questo modo l'utente ha la possibilità di selezionare una porzione di immagine che ritiene particolarmente adatta all'individuazione di patch caratteristiche, velocizzando così entrambe le fasi dell'algoritmo. Il vantaggio fondamentale di questa implementazione consiste, infatti, nella possibilità di interrompere la ricerca in fase di *matching* quando la camera lineare acquisisce le linee dell'immagine che appartengono alla *blanking area*. Per contro, riducendo la porzione di immagine su cui eseguire la fase di *teaching*, è probabile che si riduca anche il numero di pattern che l'algoritmo è in grado di individuare, nonostante la tecnica del pool di dimensioni descritta nel capitolo 3. Occorre pertanto un'attenta valutazione dei pro e dei contro per il corretto utilizzo di questa tecnica.

Una prima implementazione di questa funzionalità addizionale è stata realizzata e testata con le immagini acquisite tramite webcam. I risultati dipendono, ovviamente, in maniera significativa dalle dimensioni della porzione di immagine utilizzata per la fase di *teaching*. Pertanto è indispensabile lasciare all'utilizzatore la possibilità di configurare tutti i parametri, valutando i risultati ottenuti al termine della fase di apprendimento prima di procedere con la successiva fase di *matching*.

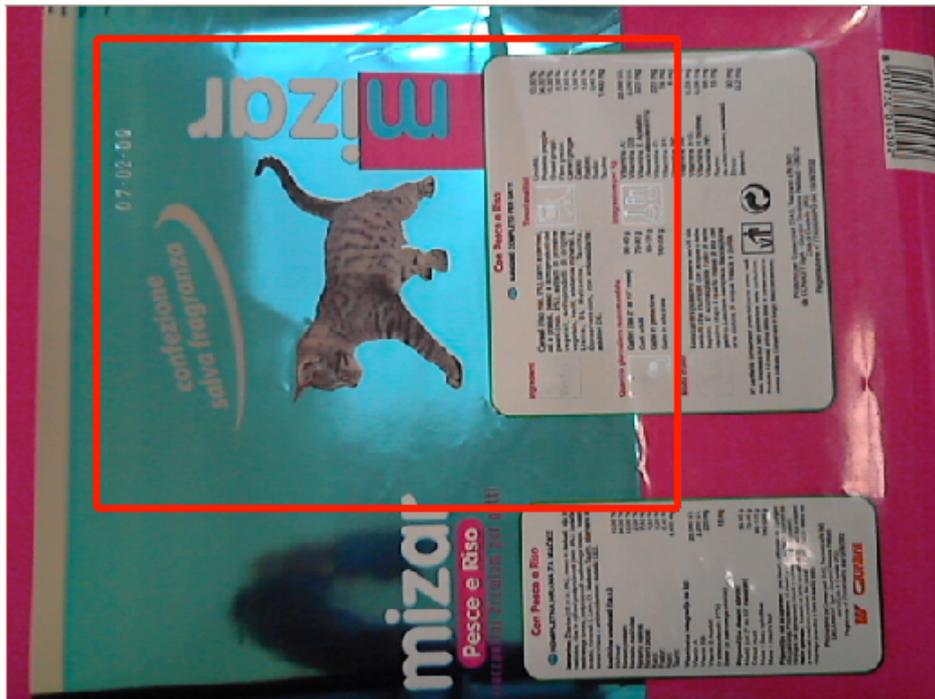


Figura 5.2 – Selezione di una porzione di immagine per il teaching

In figura 5.2 è rappresentata con un rettangolo rosso la porzione di immagine selezionata, in figura 5.3 sono riportati i pattern individuati al suo interno, che vengono poi normalmente utilizzati per la fase di riconoscimento, mostrata in figura 5.4.



Figura 5.3 - Teaching



Figura 5.4 – Matching

È importante sottolineare che, per ottenere il risultato mostrato in figura 5.4 durante la fase di *matching*, è stata necessaria un'operazione di *mapping* delle coordinate tra i dati registrati nelle due fasi dell'algoritmo, allo scopo di garantire una corretta valutazione dei controlli di integrità.

6. Conclusioni

L'obiettivo principale con il quale si è iniziato questo lavoro di tesi era la progettazione di un sistema di visione da utilizzare per l'individuazione dello *switching point* durante lo scorrimento delle bobine di imballaggi sul nastro trasportatore. Trattandosi di un ambito piuttosto specifico, in primo luogo è stato necessario analizzare lo scenario di studio, al fine di comprendere le condizioni di lavoro in cui applicare la tecnologia in esame. Durante una prima fase di analisi approfondita in merito alle tecniche attualmente utilizzate nell'industria di confezionamento, l'attenzione si è focalizzata sui *sensori di contrasto*, che rappresentano la tecnologia ad oggi più diffusa per questo tipo di attività. Grazie al supporto tecnico di un progettista, sono state chiarite le caratteristiche e le particolarità dei diversi modelli presenti sul mercato, oltre alle esigenze tipiche di un ipotetico cliente.

La valutazione dal punto di vista dell'utilizzatore è stata di fondamentale importanza per definire i requisiti iniziali del sistema: la velocità di esecuzione della fase di riconoscimento è stato considerato il punto basilare su cui incentrare l'intero progetto. Per questa ragione si è deciso di separare l'algoritmo in due fasi distinte, in modo da riuscire ad ottenere delle tempistiche accettabili. L'idea di utilizzare descrittori sull'intera immagine anche in fase di riconoscimento, infatti, è stata immediatamente scartata, proprio perché non avrebbe potuto reggere il confronto con i sensori di contrasto. Il sistema di visione presentato in questa tesi, infatti, deve competere con una tecnologia ad oggi molto diffusa ed efficace, pertanto non deve essere soltanto equiparabile, ma fornire anche un valore aggiunto rispetto ad essa. La possibilità di memorizzazione dei dati acquisiti in fase di *teaching*, il controllo di qualità, la robustezza e la

possibilità di eliminazione dei *marker* sono da considerarsi caratteristiche supplementari di una certa rilevanza per un ipotetico cliente.

Per quanto riguarda l'implementazione vera e propria, un supporto fondamentale è stato fornito dal dataset di immagini campione e dai test, eseguiti parallelamente allo sviluppo del codice. Il costante monitoraggio dei risultati ottenuti ha permesso di valutare accuratamente vantaggi e svantaggi di ogni possibile soluzione o miglioramento, ottenendo alla fine un sistema sufficientemente robusto, affidabile e veloce. In alcuni casi è stato necessario ricorrere a compromessi tra i diversi requisiti del sistema, ad esempio per quanto riguarda l'onere computazionale introdotto da una misura di similarità robusta alle variazioni di intensità luminosa. In altre circostanze, invece, le problematiche riscontrate in linea teorica hanno costretto a rinunciare ad uno sviluppo potenzialmente utile per l'utilizzatore, ma difficilmente gestibile a livello implementativo. Si pensi ad esempio alla robustezza alle variazioni di scala, che avrebbe introdotto un overhead non accettabile in termini di velocità di esecuzione.

I risultati ottenuti al termine di questo lavoro di tesi sono stati considerati accettabili, ma occorre precisare che tutti i test sono stati eseguiti simulando al computer l'ambiente di lavoro. Uno sviluppo tanto interessante quanto fondamentale sarebbe l'implementazione su dispositivo fisico della fase di *matching*, in modo da poter valutare le prestazioni reali del sistema e i requisiti hardware e software da soddisfare per ottenere risultati equiparabili a quelli dei sensori di contrasto. Solo in questo modo, infatti, si può completare lo studio di fattibilità per valutare in maniera opportuna l'introduzione sul mercato di una tecnologia innovativa come quella proposta in questo progetto.

7. Bibliografia

- [1] C. Harris and M.J. Stephens; “A Combined Corner and Edge detector”; Alvey Conference, pp 147-152, 1988
- [2] H. Moravec; “Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover”; Ph.D., p. CS, 1980
- [3] E. Rosten and T. Drummond; “Machine learning for high speed corner detection”; European Conference of Computer Vision, May 2006
- [4] J. Shi and C. Tomasi; “Good Features To Track”; 9th IEEE Conference of Computer Vision and Pattern Recognition; June 1994
- [5] H. Bay, A. A. Ess, T. Tuytelaars, L. Van Gool; “SURF: Speeded Up Robust Features”; Computer Vision and Image Understanding (CVIU) Vol 110, No 3, pp. 346-359, 2008
- [6] <http://en.wikipedia.org/wiki/Cross-correlation>
- [7], [8] http://www.t3lab.it/wp-content/uploads/download/20111129/Pattern_Matching_Techniques.pdf
- <http://docs.opencv.org/>
- http://en.wikipedia.org/wiki/Corner_detection

8. Ringraziamenti

Molte sono le persone che vorrei ringraziare per aver contribuito al raggiungimento di questo importante traguardo. A cominciare dal Professor Luigi di Stefano, che è stato il mio punto di riferimento durante tutto lo sviluppo di questa tesi di laurea, sostenendomi non solo nell'evoluzione del progetto, ma anche e soprattutto nei momenti in cui ho avuto bisogno di consigli professionali e personali.

Un sentito grazie va a Michele Benedetti e Roberto Musiani, per il loro costante supporto e la disponibilità che hanno dimostrato durante tutto il periodo di lavoro su questa tesi. Desidero inoltre ringraziare Datalogic Automation e tutti coloro che mi hanno dimostrato fiducia e hanno voluto scommettere su di me, apprezzando il mio lavoro e dandomi la possibilità di dimostrare il mio valore per il futuro. Un ringraziamento particolare va a LucaDB e a tutti i compagni delle partite a biliardino, per avermi accolta con entusiasmo sin dal primo giorno.

A questo punto voglio ringraziare chi mi è stato vicino durante tutto il percorso di studi che mi ha condotta a questo traguardo, condividendo con me le soddisfazioni e i momenti di difficoltà. Sara è sicuramente la prima della lista: grazie per la pazienza, il supporto e la certezza di poter contare su di te in ogni circostanza. E poi Baio, che al momento vive dall'altra parte del mondo ma riesce ad essere presente affianco a me in tutti i momenti di necessità: grazie per le belle chiacchierate notturne, le battutacce e i consigli che non mi fai mai mancare. E ancora Miki, sempre pronto ad ascoltarmi e a farmi dimenticare i problemi con una bella risata, Sebi, Lucal, Simi, Marti, Nico e tutti gli altri compagni di studio e di viaggio, che hanno reso questi anni divertenti e indimenticabili.

Ringrazio anche tutti gli amici non-ingegneri, che hanno sopportato i miei periodi di clausura pre-esame, dimostrandosi però sempre pronti per una bella serata quando finalmente riuscivo a trovare un po' di tempo libero. L'elenco dei nomi sarebbe troppo lungo: sentitevi tutti presi in causa!

Per concludere, i ringraziamenti più sentiti. Il primo va alla mia famiglia: mamma, papà, Alex, Andrea, Livia e Giorgia. Non so dire quanto vi sia grata per il supporto morale (ma anche economico) che mi avete dato in questi anni, so che non è stato facile tollerare gli alti e bassi di questo mio lungo percorso di studi, e so che avete sempre fatto di tutto per sostenermi e incoraggiarmi in ogni circostanza.

Infine Matteo. Sei la persona che più di ogni altra ha dovuto sopportare sbalzi di umore, crisi di nervosismo, insicurezze e tensioni. E sei la persona che più di ogni altra ha saputo tranquillizzarmi, sostenermi, incoraggiarmi, farmi sorridere e rendermi felice. Grazie di cuor

Erica