
Università Politecnica delle Marche
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Elettronica

# Sviluppo e test di algoritmi di Optical Character Recognition per sistemi di visione industriali con finalità di manutenzione predittiva.

**Development and testing of Optical Character Recognition algorithms applied to industial vision systems for predictive maintenance purposes.**

Candidato:
**Noemi Tridenti**

Relatore:
***Prof.ssa.*** **Paola Pierleoni**

Correlatore:
***Ing.*** **Enrico Santini**

Anno Accademico 2019 - 2020


Università Politecnica delle Marche
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Elettronica

# Sviluppo e test di algoritmi di Optical Character Recognition per sistemi di visione industriali con finalità di manutenzione predittiva.

**Development and testing of Optical Character Recognition algorithms applied to industial vision systems for predictive maintenance purposes.**

Candidato:
**Noemi Tridenti**

Relatore:
***Prof.ssa.*** **Paola Pierleoni**

Correlatore:
***Ing.*** **Enrico Santini**

Anno Accademico 2019 - 2020

Università Politecnica delle Marche
Corso di Laurea in Ingegneria Elettronica
Facoltà di Ingegneria
Via Brecce Bianche – 60131 Ancona (AN), Italy

# Ringraziamenti

Un ringraziamento particolare va al mio relatore la Prof.ssa Paola Pierleoni per la sua infinita disponibilità e per la fiducia mostratami. Grazie per i suoi indispensabili consigli e per le conoscenze trasmesse durante tutto il percorso di stesura dell'elaborato. Ringrazio l'Ing. Enrico Santini e tutto il personale dell'azienda iGuzzini Illuminazione per avermi dato la possibilità di svolgere il mio lavoro di tesi in un luogo interessante e dinamico, che mi ha permesso di mettermi in gioco e fare un'esperienza che sarà preziosa per il mio futuro. Una dedica speciale va a tutta la mia famiglia, per il loro costante sostegno ed i loro insegnamenti senza i quali oggi non sarei ciò che sono. Senza di voi, tutto questo non sarebbe stato possibile. Un grazie di cuore a Riccardo e a tutti i miei amici per avermi sempre supportata in ogni decisione e per esserci stati sempre soprattutto nei momenti di difficoltà. Infine, dedico questa tesi a me stessa, ai miei sacrifici e alla mia tenacia che mi hanno permesso di arrivare fin qui.

*Ancona, Ottobre 2020*

Noemi Tridenti

# Abstract

The following thesis describes the work carried out during the internship period at *iGuzzini illuminazione S.p.A.*, under the attention of Professor Paola Pierleoni and the Manufacturing Director Enrico Santini.

The main goal to be achieved at the end of the experience was the implementation of an OCR (Optical Character Recognition) algorithm through the MATLAB® software, for the recognition of number plates for predictive maintenance purposes. MATLAB® (Matrix Laboratory) is an environment for numerical computation and statistical analysis written in C, which also includes the homonymous programming language created by MathWorks®. It is a software widely used in industry and universities due to its many tools to support different fields of study.

The development of the recognition algorithm arises from the company's need for a vision system, to apply to its painting system, which is able to recognize correctly the number plates that identify the product to be painted. These plates tend to be covered with paint residues, which make difficult to read the number correctly. The purpose of the algorithm is to perform predictive maintenance, alerting the operator when the reading accuracy falls below a pre-established threshold.

The starting point of the internship was the study of existing OCR algorithms and implementation techniques in Industry 4.0 and Smart Factories. These algorithms are widely used not only in industrial fields with the aim of recognizing alphanumeric codes but also in other sectors such as scientific research, public safety and medical diagnostics.

The second step for the development of the algorithm was the carrying out of an image acquisition campaign with a mechanical set up of the acquisition cameras, similar to the final one with which the vision system will be installed on the painting plant. This allowed us to have a large dataset of images to work on in order to train the algorithm and make it suitable for the company's need.

The development of the OCR algorithm consists of three main phases: the Preprocessing of the images, the algorithm Training and finally the Testing to validate the algorithm. In the first phase of Preprocessing, the acquired images are processed to make the numbers clearly visible and to eliminate any disturbing elements or non-uniformity of the background. To do that, the functions that MATLAB® makes available in Image Processing Toolbox™ and in Computer Vision Toolbox™ were used.

The adequately processed images were then used in the Training phase to instruct the algorithm to make it capable of recognizing the images. For this purpose, the MATLAB® OCR Trainer App was used as it allows you to create a custom ocr function capable of recognizing custom languages and fonts. You can use this App to segment and classify characters interactively and to carry out algorithm training with extracted samples. The function generated by the OCR Trainer performs the OCR process using the trained linguistic model and returns the read number and the percentages of accuracy with which the single characters have been read.

In the last phase of work, the Optical Character Recognition algorithm was validated, testing it on a large number of images. The results obtained were in line with expectations, as the number plates were read correctly with high precision. The cleanest plates free of paint residues were recognized with high accuracy, while the very dirty ones were recognized but with accuracy percentages below the pre-established threshold. The developed algorithm respects the needs of the company as it has the function of processing the image acquired by the camera of the vision system and communicating to the system manager if a plate maintenance is required.

The fundamental steps that led to the development of the OCR algorithm will be explored in the following chapters.

# Sommario

Nella seguente tesi viene descritto il lavoro svolto durante il periodo di tirocinio presso l'azienda *iGuzzini illuminazione S.p.A.*, sotto l'attenzione della Professoressa Paola Pierleoni e del Manufacturing Director Enrico Santini.

L'obbiettivo principale da raggiungere al termine dell'esperienza era l'implementazione di un algoritmo di OCR (Optical Character Recognition) attraverso il software MATLAB®, per il riconoscimento di targhe numeriche con finalità di manutenzione predittiva. MATLAB® (Matrix Laboratory) è un ambiente per il calcolo numerico e l'analisi statistica scritto in C, che comprende anche l'omonimo linguaggio di programmazione creato dalla MathWorks®. È un software molto usato nell'industria e nelle università per via dei suoi numerosi strumenti a supporto di diversi campi di studio.

Lo sviluppo del algoritmo di riconoscimento nasce dalla necessità da parte dell'azienda di un sistema di visione, da applicare al proprio impianto di verniciatura, che sia in grado di riconoscere correttamente le targhe numeriche che identificano il prodotto da verniciare. Tali targhe tendono ad essere ricoperte da residui di vernice che rendono difficile la corretta lettura del numero. Lo scopo dell'algoritmo è quello di svolgere una manutenzione predittiva, avvisando l'operatore quando l'accuratezza della lettura scende sotto una soglia prestabilita.

Il punto di partenza del tirocinio è stato lo studio di algoritmi di OCR già esistenti e delle tecniche implementative nel mondo dell'Industria 4.0 e delle Smart Factories. Questi algoritmi sono ampiamente usati non soli in ambiti industriali con il fine di riconoscere codici alfanumerici ma anche in altri settori come quello della ricerca scientifica, della pubblica sicurezza e della diagnostica medica.

Il secondo passo necessario per lo sviluppo dell'algoritmo è stato lo svolgimento di una campagna di acquisizione immagini con un set up meccanico delle telecamere di acquisizione simile a quello definitivo con cui è stato istallato il sistema di visione presso l'impianto di verniciatura. Questo ha permesso di avere un ampio dataset di immagini su cui lavorare per poter allenare l'algoritmo e

renderlo adeguato alle esigenze dell'azienda.

Il processo di OCR si compone di tre fasi principali: il Preprocessing delle immagini, il Training dell'algoritmo ed infine il Testing per validare l'algoritmo. Nella prima fase di Preprocessing le immagini acquisite sono state elaborate così da rendere ben visibile il numero della targa e per eliminare eventuali elementi di disturbo o di non uniformità dello sfondo. Per fare ciò sono state utilizzate le funzioni che MATLAB® mette a disposizione nel Image Processing Toolbox™ e nel Computer Vision Toolbox™.

Le immagini elaborate in maniera adeguata sono state poi impiegate nella fase di Training per istruire l'algoritmo così da renderlo capace di riconoscere le immagini. A questo fine è stata utilizzata OCR Trainer App di MATLAB® in quanto permette di creare una funzione ocr personalizzata in grado di riconoscere linguaggi e font custom. È possibile utilizzare questa App per segmentare e classificare i caratteri in modo interattivo e per svolgere il training dell'algoritmo con i campioni estratti. La funzione generata dall' OCR Trainer esegue il processo di OCR utilizzando il modello linguistico addestrato e restituisce il numero letto e le percentuali di accuratezza con cui sono stati letti i singoli caratteri.

Nell'ultima fase di lavoro l'algoritmo di Optical Character Recognition è stato validato, testandolo su un elevato numero di immagini. I risultati ottenuti sono stati conformi alle aspettative, dato che le targhe numeriche sono state lette correttamente con alta precisione. Le targhe più pulite e prive di residui di vernice sono state riconosciute con un'elevata accuratezza mentre quelle molto sporche sono state riconosciute ma con percentuali di accuratezza sotto lo soglia prestabilita. L'algoritmo sviluppato rispetta le esigenze dell'azienda in quanto ha la funzione di elaborare l'immagine acquisita dalla telecamera del sistema di visione e di comunicare al gestore dell'impianto se è necessaria una manutenzione delle targhe.

Nei capitoli seguenti verranno approfonditi i passi fondamentali che hanno portato allo sviluppo dell'algoritmo di OCR.

# Indice

# Elenco delle figure

# Elenco delle tabelle

# Capitolo 1

# Industria 4.0 e algoritmi di OCR

Con il rapido sviluppo della tecnologia elettrica ed elettronica, della tecnologia dell'informazione e di tecnologie produttive sempre più avanzate, la modalità di produzione delle imprese si sta trasferendo dal digitale all'intelligente. Questo processo di rinnovamento tecnologico che integra ai sistemi di produzione già esistenti tecnologie innovative, è noto come Industria 4.0.

L'esigenza di avere prodotti personalizzati richiede alle industrie di essere estremamente competitive, reattive e rapide ad adattare le proprie produzioni alle specifiche dei clienti. A causa di queste nuove sfide, i vantaggi dei processi manifatturieri tradizionali si stanno gradualmente riducendo. Di conseguenza, la tecnologia di produzione intelligente sta diventando una delle aree che attira maggiore interesse da parte del governo, delle imprese e dei ricercatori accademici.

La concretizzazione dei principi dell'Industria 4.0. è la fabbrica intelligente o Smart Factory. Questa applica all'impresa tradizionale tutte le tecnologie tipiche del nuovo concetto di industria. In particolare, sfruttando le tecnologie dell'informazione come l'Internet of Things, il Machine Learning che comprende gli algoritmi di OCR e la Computer Vision, le imprese riescono a migliorare la produzione e a rispondere in tempo reale alla variabile domanda di mercato.

La Smart Factory è una fabbrica indipendente, dotata di sensori e orientata al supporto di persone e macchine nello svolgimento delle loro attività. I suoi sistemi sono in grado di elaborare le informazioni acquisite e di valutarle, al fine di prendere decisioni, interagire con l'ambiente circostante e di svolgere manutenzione predittiva così da ridurre al minimo i problemi agli impianti. Grazie all'integrazione delle nuove tecnologie le imprese sono più flessibili ed efficienti, con sistemi produttivi capaci di ridurre al minimo l'intervento dell'uomo e di migliorarne le condizioni lavorative.

## 1.1 Industria 4.0 e Smart Factory

Il termine Industria 4.0 sta per quarta rivoluzione industriale che può essere definita come un nuovo livello di organizzazione e di controllo sopra l'intero ciclo di vita del prodotto [1]. Lo scopo dell'Industria 4.0 è quello di convertire le macchine attuali in macchine consapevoli di sé e capaci di autoapprendere per migliorare le loro prestazioni e la gestione della manutenzione.

A questo scopo il paradigma dell'Industria 4.0 promuove la connessione di elementi fisici come sensori, dispositivi e risorse aziendali alla rete Internet. Alla tecnologia già esistente viene infatti integrata la tecnologia cyber e in particolare i Cyber Physical System (CPS), strumenti hardware in grado di connettersi alla rete e di interfacciarsi con le tecnologie informatiche. La fabbrica intelligente, che si basa sul digitale e sull'automazione industriale, utilizza le moderne tecnologie come ad esempio le piattaforme Cloud e l'Industrial Internet of Things per migliorare la gestione della produzione e la Quality of Service [2]. Tramite l'analisi dei dati di produzione, la fabbrica intelligente può realizzare produzione flessibile, riconfigurazione dinamica e ottimizzazione della produzione, mirando ad adattare il sistema ai cambiamenti del modello di business e del comportamento d'acquisto dei consumatori.

Pertanto, la Smart Factory rappresenta un sistema di ingegneria che consiste principalmente di tre aspetti: interconnessione, collaborazione ed esecuzione [2]. Come mostrato in Figura 1.1, l'architettura di una fabbrica intelligente comprende quattro livelli:

- Physical Resource Layer
- Network Layer
- Data Application Layer
- Terminal Layer

### 1.1.1 Physical Resource Layer

Le risorse fisiche includono tutte quelle risorse di produzione coinvolte nell'intero ciclo di vita del prodotto e che rappresentano la base per il raggiungimento della produzione intelligente [2]. La produzione efficiente di prodotti personalizzati propone nuove esigenze in materia di attrezzature, di linea di produzione e di acquisizione dati. A causa della mancanza di flessibilità e configurabilità, l'attuale attrezzatura ha una forte specificità e una gamma di applicazioni relativamente ristretta, che si traduce in un debole adattamento ai cambiamenti.

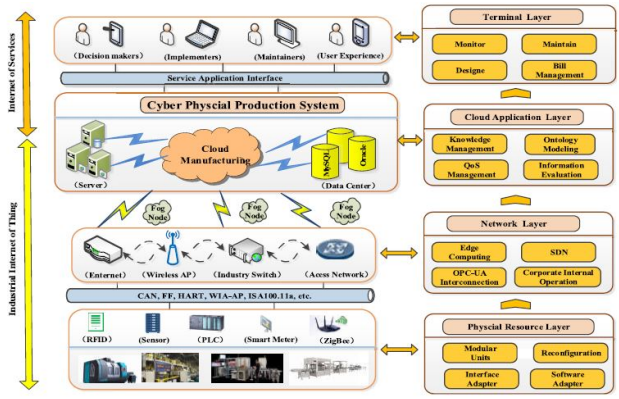


Figura 1.1: Gerarchia strutturale di una Smart Factory. Immagine tratta da Chen et al. [2].

Per questo motivo all'interno delle Smart Factories sono state introdotte nuove unità produttive, linee di produzione e controllori riconfigurabili, con lo scopo di garantire una maggiore flessibilità, scalabilità e la generazione di un'ampia gamma di prodotti diversi.

Per il monitoraggio, l'acquisizione e la registrazione dei dati vengono utilizzate le reti di sensori wireless (WSN) e la Computer Vision così da acquisire dati in tempo reale e permettere alle macchine di prendere decisioni in presenza di problematiche differenti [1]. Queste tipologie di sensori hanno una fondamentale importanza all'interno della Smart Factory dato che riescono a segnalare all'operatore la presenza di problemi o di errori lungo la linea di produzione dei prodotti e a organizzare tempestivamente l'attività di manutenzione necessaria.

### 1.1.2 Network Layer

Il Network layer, con il compito di controllare e connettere i sistemi intelligenti del Physical Layer, gioca un ruolo fondamentale nella Smart Factory. A causa dei miglioramenti della tecnologia di cloud computing, tecnologie di rete affidabili e in tempo reale sono necessarie per la trasmissione dei dati e la condivisione delle informazioni tra le apparecchiature intelligenti e la piattaforma cloud di produzione.

La rete wireless industriale deve essere a bassa latenza, ad alta affidabilità e alta precisione soprattutto quando ha a che fare con un servizio di controllo e deve avere un basso consumo energetico nell'acquisizione dei dati [2]. Inoltre le reti devono essere caratterizzate da un'elevata velocità di trasmissione soprattutto per i servizi interattivi. Con lo scopo di soddisfare queste esigenze sono

state introdotte reti moderne e dinamiche come le Software Defined Industrial Networks (SDN) che presentano la scalabilità e la versatilità necessaria per il controllo dei diversi sensori wireless [2]. Queste reti possono essere utilizzate insieme alle tecnologie di Machine Learning per rendere la rete più intelligente e autoadattabile alle esigenze dell'azienda.

### 1.1.3 Data Application Layer e Terminal Layer

Il Data Application Layer consiste in tutte quelle tecnologie informatiche indispensabili per l'elaborazione della grande quantità di dati provenienti in tempo reale dai sensori e dal processo di produzione. Con questo livello stiamo indicando i server e le piattaforme cloud che si occupano del trattamento e della memorizzazione dei dati. L'uso di Big Data Analytics in ambito industriale permette di operare un'analisi predittiva con lo scopo di prevenire guasti e svolgere manutenzione. Infatti il corretto funzionamento delle apparecchiature intelligenti è garanzia per l'impresa di una continua produzione [1].

Inoltre, le tecnologie di analisi basate su big data vengono utilizzate anche nella progettazione del prodotto, partendo da previsioni fatte sui dati delle tendenze di acquisto. Il Terminal Layer comprende i dispositivi degli utenti finali, come smartphone, computer desktop e dispositivi elettrici che sono distribuiti negli impianti, negli uffici, in centri di monitoraggio e altre regioni. I dispositivi terminali sono utilizzati dagli operatori per visualizzare i risultati dell'elaborazione nel cloud, e supportare il monitoraggio remoto della produzione e della manutenzione [2].

## 1.2 Manutenzione predittiva nell'Industria 4.0

L'algoritmo di OCR è stato sviluppato con lo scopo di svolgere manutenzione predittiva all'interno dell'impianto di verniciatura dell'impresa *iGuzzini Illuminazione S.p.A.* La possibilità di predire quando è necessaria una manutenzione e la prevenzione di possibili guasti all'interno degli impianti è uno dei vantaggi introdotti dalle tecnologie che costituiscono l'Industria 4.0 e le Smart Factories.

La manutenzione mira a ridurre ed eliminare il numero di guasti che si possono verificare durante la produzione in quanto problemi alla macchina o all'attrezzatura potrebbero portare a interruzioni per la filiera. Infatti la manutenzione predittiva è un'attività fortemente consigliata per quei componenti cruciali il cui guasto può provocare gravi perdite di funzionalità e rischi per la sicurezza [3]. La manutenzione può essere definita come tutte le azioni necessarie per mantenere o ripristinare un sistema a uno stato che è necessario per adempiere

alla funzione prevista. L'obiettivo principale della manutenzione è quindi quello di preservare la capacità e la funzionalità del sistema, controllando il costo indotto dalle attività di manutenzione e la potenziale perdita di produzione [3]. I guasti possono essere definiti come una qualsiasi modifica o anomalia nel sistema che causa un livello di prestazioni insoddisfacente e problemi che spesso sono fastidiosi, scomodi e costosi. Proprio per questo motivo i piani di manutenzione sono progettati per ridurre o eliminare il numero di guasti e i costi ad essi correlati.

### 1.2.1 Fasi della manutenzione predittiva e tecnologie usate

La manutenzione predittiva si basa molto su tecniche dell'ingegneria e su strumenti statistici per elaborare i dati e analizzare lo stato di salute delle apparecchiature, al fine di prevedere possibili guasti. La previsione delle condizioni dell'attrezzatura si basa sulla constatazione che dei possibili guasti si verificano dopo un certo processo di degrado dal normale stato della macchina [3]. Pertanto, attraverso il monitoraggio della degradazione e la previsione dei guasti, la manutenzione predittiva consente di identificare e risolvere i problemi prima del potenziale danno.

Una manutenzione basata sull'osservazione delle condizioni delle macchine, necessita di ispezioni in tempo reale utilizzando sensori wireless e sistemi di acquisizione dati. Di conseguenza, una fabbrica intelligente caratterizzata da un elevato numero di sensori genera e raccoglie grandi quantità di dati. Questi big data possono essere utilizzati nella manutenzione predittiva, migliorando notevolmente le condizioni del sistema e aumentando la velocità e la precisione nel processo decisionale in materia di manutenzione. Pertanto moderne tecnologie dell'informazione come l'intelligenza artificiale e Big Data Analytics vengono applicate alle Smart Factories per classificare i probabili modelli di fallimento e stimare le condizioni delle macchine che costituiscono gli impianti [4].

Nella Figura 1.2 è mostrato un modello di manutenzione predittiva dove è messo in evidenza il flusso di dati da elaborare. Possono essere individuati tre passaggi chiave nel processo di manutenzione predittiva:

- Il monitoraggio delle macchine e l'elaborazione dei dati
- Diagnosi e prognosi
- Decisione sulla manutenzione

Lo stato di salute delle apparecchiature all'interno degli impianti è monitorato da una fitta rete di sensori e sistemi di acquisizione dati. Questi dati prodotti

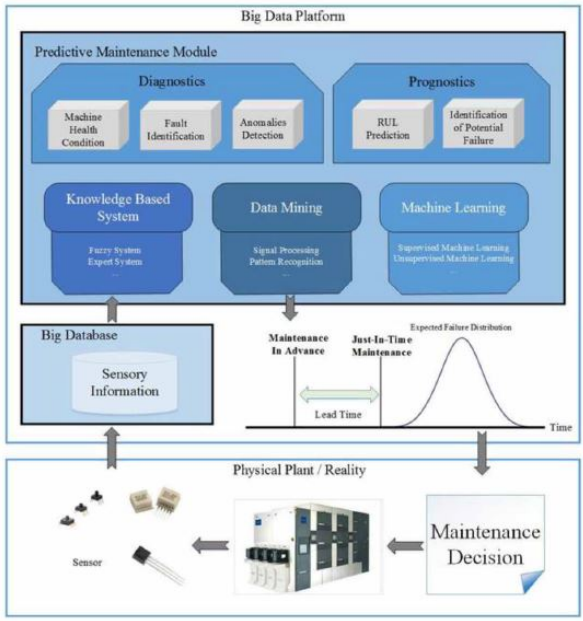


Figura 1.2: Modello di manutenzione predittiva. Immagine tratta da Lee et al. [3].

da sensori di vibrazione, calore e pressione e da telecamere per la visione artificiale forniscono le informazioni sensoriali necessarie a valutare le condizioni della macchina e allo sviluppo di opportuni modelli di manutenzione. I big data raccolti in tempo reale o provenienti da precedenti acquisizioni vengono memorizzati in database ed elaborati attraverso Knowledge Based System e attraverso tecniche informatiche di Machine Learning e Data Mining [3].

L'elaborazione di dati attraverso sistemi basati sulla conoscenza tenta di estrarre le regole per la generazione degli algoritmi di manutenzione tramite intelligenza umana e opinione di esperti. I Knowledge Based Systems incoraggiano un modo più flessibile per aumentare la qualità nella risoluzione dei problemi e nell'estrazione di dati rilevanti per l'attività di manutenzione [4]. L'obiettivo del processo di Data Mining è invece quello di creare un modello di analisi delle caratteristiche, che sia in grado di classificare i dati in gruppi, rilevare caratteristiche irregolari e individuare relazioni esistevano tra i dati. Nella tecnica di Data Mining, l'accuratezza delle informazioni scoperte aumenta insieme all'aumento dei dati raccolti dai sensori e dai dati storici di manutenzione. Attraverso queste tecniche di elaborazione si è in grado di prevedere il guasto dal modello di comportamento della macchina operativa [4].

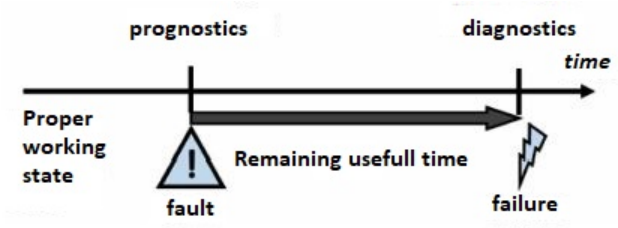


Figura 1.3: Progressione da errore a fallimento. Immagine tratta da Lee et al. [3].

I meccanismi di elaborazione basati su Knowledge Based Systems e Data Mining sono usati per individuare informazioni in anticipo per il processo di lavoro dell'algoritmo, estraendo conoscenze da enormi quantità di dati. Al contrario il Machine Learning si occupa di ragionamento automatico e risoluzione cognitiva artificiale, sfruttando un'intelligenza agente. Il Machine Learning funziona come una misurazione online della salute dell'attrezzatura con lo scopo di rivelare degrado della macchina e anomalie dai modelli [3]. L'autoapprendimento, unitamente all'analisi del normale degrado, consente la previsione di guasti casuali della macchina in modo efficace ed efficiente al fine di pianificare manutenzione prima che si verifichi un guasto.

Il modulo di manutenzione predittiva è costituito da due sistemi principali: un sistema diagnostico e un sistema prognostico [3]. Il processo di diagnostica determina il modo in cui il sistema è stato danneggiato e indaga la causa o la natura di tale condizione degradata. I guasti vengono rilevati, isolati e identificati per creare un record diagnostico e le possibilità di guasti vengono calcolate per trovare il potenziale pattern di guasto. Le operazioni diagnostiche seguono un approccio di analisi dall'effetto alla causa. D'altra parte, i sistemi prognostici considerano il tempo un fattore vitale e si concentrano sulla previsione delle condizioni future del sistema o del componente e ne calcolano accuratamente la vita utile residua prima il fallimento. Nella Figura 1.3 viene evidenziata la differenza tra diagnostica e prognostica e mostrata la progressione da errore a fallimento.

Individuate le cause che hanno compromesso e degradato la produttività della macchina si procede con lo svolgimento della attività di manutenzione secondo i dati raccolti. L'obiettivo è quello di svolgere una manutenzione che sia in anticipo o al limite just-in-time con lo scopo di intervenite prima che ci sia un drastico peggioramento nelle condizioni di salute della macchina e si presentino dei guasti sulle attrezzature degradate [3]. Incorporando l'intelligenza artificiale nel monitoraggio dello stato di salute della macchina è possibile per i produttori migliorare l'affidabilità e la produzione complessiva del sistema, soprattutto

riducendo i tempi di fermo macchina.

## 1.3 Algoritmi di Optical Character Recognition

Con l'avvento dell'Industria 4.0 i processi manifatturieri si stanno spostando sempre più verso la produzione intelligente, integrando tecnologia informatica e di automazione industriale. Questi sistemi di produzione innovativi che costituiscono le Smart Factories, dipendono fortemente dalla disponibilità di dati acquisiti online e dalla possibilità di identificare gli innumerevoli codici marcati su prodotti e su componenti degli impianti. Per garantire infatti la corretta gestione dei processi di automazione è indispensabile avere dei sistemi che siano in grado di individuare e codificare le informazioni contenute all'interno dei codici QR, a barre o alfanumerici.

### 1.3.1 Definizione e usi

Il riconoscimento ottico dei caratteri (OCR) può essere definito come un processo di digitalizzazione di un immagine nei suoi caratteri costitutivi. L'Optical Character Recognition è un pezzo di software che converte testo e immagini stampati in file in forma digitalizzata in modo che possa essere manipolato dalla macchina [5]. L'OCR è un problema complesso a causa della varietà di lingue, caratteri e stili in cui è possibile scrivere il testo, e le complesse regole delle lingue ecc. I primi sistemi OCR non erano computer ma dispositivi meccanici in grado di riconoscere i caratteri, ma che erano molto lenti e con bassa precisione. Negli ultimi trent'anni sono state effettuate ricerche sostanziali sui processi di OCR. Ciò ha portato alla nascita di algoritmi in grado di analizzare immagini, documenti multilingua e manoscritti e allo sviluppo di OCR omni-font. Nonostante queste ricerche approfondite, lo sviluppo di un OCR con capacità paragonabili a quelle umane rimane ancora un sfida aperta [5].

Gli algoritmi di OCR sono usati in numerosi contesti e ambiti di applicazione: riconoscimento delle targhe di veicoli, riconoscimento captcha, creazione di archivi e biblioteche digitali, riconoscono di fatture ed estratti conto bancari, estrazione di dati da moduli sanitari, da documentazione o posta e come immissione dati dai record di dati in carta stampata [6]. In ambito industriale tecniche di Machine Learning e di intelligenza artificiale si affidano spesso a questo processo di riconoscimento ottico di caratteri con elevata precisione. La capacità di ridurre il coinvolgimento del personale per aumentare la produttività e la memorizzazione di codici in modo efficace sono i principali vantaggi di questo sistema di Optical Character Recognition.

Il processo di OCR è composto in diversi passaggi:

- Acquisizione dell'immagine
- Pre-elaborazione
- Segmentazione dei caratteri
- Estrazione delle caratteristiche
- Classificazione dei caratteri
- Post-elaborazione

Di seguito verranno approfonditi e descritti i passi fondamentali per lo sviluppo dell'algoritmo.

### 1.3.2 Acquisizione dell'immagine

L'acquisizione delle immagini è il passaggio iniziale dell'OCR. In questa fase si procede con la cattura di un'immagine da una fonte esterna attraverso l'uso di scanner, fotocamere, ecc. Una volta ottenuta l'immagine digitale si procede con la sua conversione in una forma adatta che può essere facilmente elaborata dal computer [5]. Ciò può comportare la quantizzazione e la compressione dell'immagine. Un caso speciale di quantizzazione è la binarizzazione che coinvolge solo due livelli dell'immagine. Nella maggior parte dei casi, l'immagine binaria è sufficiente per caratterizzare l'immagine originare e contiene al suo interno tutte le informazioni necessarie per svolgere i passi successivi.

### 1.3.3 Pre-elaborazione

Una volta acquisita l'immagine, diversi passaggi di pre-elaborazione possono essere eseguiti per migliorarne la qualità. Lo scopo della pre-elaborazione è eliminare le caratteristiche indesiderate o il rumore in un'immagine senza perdere alcuna informazione significativa. Sono necessarie tecniche di pre-elaborazione che si possano applicare su immagini a colori, a livello di grigio o binarie contenenti testo e/o grafica. Poiché l'elaborazione di immagini a colori è computazionalmente più costosa, la maggior parte delle applicazioni nei sistemi di riconoscimento dei caratteri utilizza immagini binarie o grigie [7]. La pre-elaborazione riduce i dati incoerenti e il rumore, migliora l'immagine e la prepara per le fasi successive. Poiché la pre-elaborazione controlla l'idoneità dell'ingresso per le fasi successive, questa è una fase di primaria importanza prima della fase di estrazione delle caratteristiche. Il primo passo da svolgere in questa fase è l'estrazione dall'immagine della regione di interesse (ROI). Questa fase è importante per diminuire l'interferenza da aree chiare e scure

sulla regione presa in considerazione. Infatti le immagini in ingresso possono avere luminosità diversi, specialmente a causa dell'interferenza della luce solare o per l'illuminazione artificiale spesso mal regolata o assente [6]. Dopo aver selezionato la ROI si possono applicare diversi filtri con lo scopo di aumentare il contrasto globale e di ottenere un'immagine bimodale. La diminuzione del rumore che causa la riduzione del tasso di riconoscimento dei caratteri è la questione principale e più importante nella fase di pre-elaborazione. Pertanto, tutti gli oggetti più piccoli di un possibile carattere devono essere eliminati, stando attenti a non filtrare via anche i possibili caratteri rotti [6]. Alcune operazioni che si possono eseguire nella fase di pre-elaborazione sono elencate e brevemente descritte nella Tabella 1.1.

Tabella 1.1: Alcune importanti operazioni di pre-elaborazione

| Operazioni | Descrizione |
|---|---|
| **Binarizzazione** | Separa i pixel dell'immagine come testo o sfondo. |
| **Riduzione del rumore** | Miglioramenti nei dispositivi di acquisizione immagini prodotti dai progressi della tecnologia. Rimozione di elementi di disturbo. |
| **Correzione di immagini inclinate** | A causa della possibilità di avere immagine ruotate prodotte dal dispositivo di acquisizione immagine, l'inclinazione del documento dovrebbe essere corretta. |
| **Operazioni morfologiche** | Aggiunta o rimozione di pixel ai caratteri che hanno buchi o pixel in eccesso. |

### 1.3.4 Segmentazione dei caratteri

Nella fase di segmentazione, l'immagine viene segmentata in candidati a caratteri prima di passare alla fase successiva. La segmentazione è il processo di isolamento della componente di testo all'interno di un'immagine dallo sfondo dell'immagine. In questa fase vengono selezionati tutti gli elementi che hanno caratteristiche simili ai caratteri, come dimensioni, proporzioni altezza x larghezza e regione dell'immagine [6]. Tuttavia in situazioni complesse, dove i caratteri sono sovrapposti, interrotti o è presente del rumore nell'immagine vengono usate delle tecniche di segmentazione con caratteristiche più avanzate. Il metodo generale per riconoscere il testo in un'immagine è usare estrattori di caratteristiche nel file immagine, per poi classificare i caratteri usando queste caratteristiche e applicare l'algoritmo di apprendimento automatico nella fase di classificazione dei caratteri. In questa fase di segmentazione, i caratteri nel file immagine sono separati in modo da poterli passare al motore di riconoscimento. In generale, la segmentazione del testo da un'immagine opera prima

una segmentazione delle linee di testo, poi procede con la segmentazione delle parole e infine con la segmentazione dei singoli caratteri [7].

### 1.3.5 Estrazione delle caratteristiche

I caratteri segmentati sono quindi processati per estrarre le caratteristiche che identificano in modo univoco i singoli caratteri. Sulla base di queste caratteristiche, i caratteri verranno poi riconosciuti nella fase di classificazione. Le caratteristiche estratte dovrebbero essere efficientemente computabili, minimizzando le variazioni intra-classe e massimizzando variazioni inter-classe. In questa fase si procede con l'estrazione di caratteristiche pertinenti da oggetti o alfabeti per costruire vettori di caratteristiche [8]. Questi vettori vengono poi utilizzati dai classificatori per identificare l'unità di input con l'unità di output. Pertanto diventa facile per il classificatore classificare gli oggetti in input in classi molto diverse tra loro basandosi proprio su questi vettori.

### 1.3.6 Classificazione dei caratteri

La classificazione è la procedura di distribuzione degli input rispetto alle informazioni rilevate nella loro classe di appartenenza al fine di creare gruppi con qualità omogenee, separando i diversi input in classi diverse. L'immagine preelaborata viene inviata a un file classificatore che trova una classe a cui appartiene l'oggetto preelaborato con il fine di creare classi di caratteri con caratteristiche differenti. Per svolgere la classificazione dei caratteri possono essere usati diversi strumenti. I classificatori più comuni includono reti neurali, macchine a vettori di supporto, classificatori a distanza minima [9]. Quelli comunemente più usati sono le reti neurali convoluzionali (CNN) in quanto hanno mostrato degli ottimi risultati nelle attività di apprendimento e di elaborazione di immagini [10]. Un altro classificatore è il Tesseract, un riconoscimento ottico dei caratteri open source, con capacità di riconoscere più di 100 lingue e che può essere addestrato a riconoscerne altre [11]. Le tecniche di classificazione strutturale si basano invece sulle caratteristiche estratte dalla struttura dell'immagine e utilizzano differenti regole di decisione per classificare i caratteri. In generale i classificatori sono studenti supervisionati che richiedono un set di formazione etichettato correttamente in input, così da essere istruiti al corretto riconoscimento dei dati in ingresso. Una volta che il classificatore è stato istruito questo deve essere testato con un diverso dataset per verificare la corretta esecuzione della fase di apprendimento [9]. In Figura 1.4 sono stati riassunti i passi principali delle fasi di training e testing del classificatore.

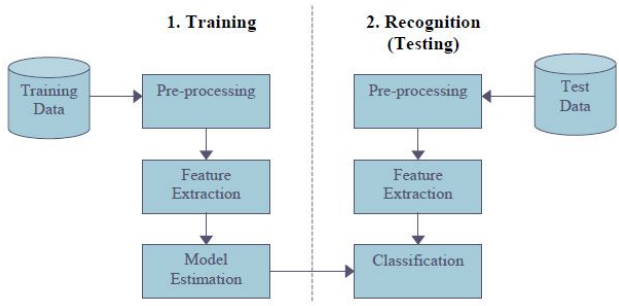


Figura 1.4: Fasi di Training e Testing del classificatore

### 1.3.7 Post-elaborazione

Dopo la classificazione i risultati non sono corretti al 100%, soprattutto per le lingue complesse e per immagini non elaborate correttamente. Una volta che il carattere è stato classificato, ci sono vari approcci che possono essere utilizzati per migliorare l'accuratezza dei risultati dell'OCR. Uno di questi è usare più di un classificatore per la classificazione dell'immagine. I risultati dei classificatori possono quindi essere combinati utilizzando vari metodi. Oppure possono essere eseguite delle analisi di contesto o possono essere utilizzati dei sistemi di correzione errori. Questi possono essere dei sistemi di correzione degli errori di post-elaborazione manuale, semiautomatico o completamente automatico. Nel semiautomatico il sistema di post-correzione rileva automaticamente gli errori e propone correzioni ai correttori umani mentre in quello completamente automatico il sistema di post-correzione fa il rilevamento e la correzione di errori autonomamente [12].

# Capitolo 2

# Sistema di visione e campagna d'acquisizione immagini

## 2.1 Presentazione azienda e contesto di lavoro

*iGuzzini illuminazione S.p.A.* è una azienda fondata nel 1959 a Recanati (MC), che conta numerose sedi non solo sul territorio italiano ma anche nel resto del mondo, con attività produttive in oltre 20 paesi distribuiti in 5 continenti. L'azienda recanatese è leader nel settore dell'illuminotecnica occupandosi dello studio, del design e della produzione di sistemi di illuminazione intelligenti per interni ed esterni.

Per quanto riguarda la produzione, lo stabilimento di Recanati si occupa delle lavorazioni meccaniche di elementi in alluminio, dello stampaggio di elementi in materiale plastico, della fase di verniciatura dei semilavorati in alluminio e infine della fase di assemblaggio e di confezionamento. Gli impianti produttivi di *iGuzzini* presentano elementi all'avanguardia che si inseriscono nell'ambito dell'Industria 4.0 e delle Smart Factories. Infatti l'azienda è da sempre attenta al continuo rinnovamento tecnologico, seguendo la tendenza dell'automazione industriale che integra nuove tecnologie produttive per migliorare le condizioni di lavoro, creare nuovi modelli di business e aumentare la produttività e la qualità produttiva degli impianti.

### 2.1.1 Il nuovo impianto di verniciatura

Esempio del rinnovamento tecnologico all'interno dell'impresa è l'impianto 2230 del reparto verniciatura (REVER). Questo moderno impianto di veniciatura affianca quello già esistente nella verniciatura a solvente dei componenti in alluminio. Il 2230 è costituito da un nastro trasportatore aereo che si muove alla velocità di 2 metri al minuto e che percorre tre aree principali: la zona di carico, l'area di sgrassamento e preparazione dei prodotti e infine le cabine di

Figura 2.1: Ganci per la verniciatura su cui sono appoggiati i prodotti da verniciare

Figura 2.2: Bilanciella agganciata al nastro trasportatore

verniciatura con forni per l'asciugatura.

Al nastro trasportatore sono collegate delle bilancelle che vengono identificate da delle targhe numeriche a tre cifre affisse sopra di esse. Nella zona di carico gli operatori appendono manualmente alle bilancelle dei ganci su cui vengono appoggiati i prodotti da verniciare. In Figura 2.1 e in Figura 2.2 sono mostrati i ganci per la verniciatura e un esempio di bilancella. Questi prodotti in alluminio passano in una prima cabina in cui si procede allo sgrassaggio tramite solvente. Getti di prodotto permettono la rimozione di possibili residui oleosi presenti sulla superficie dei pezzi che potrebbero essere rimasti da lavorazioni meccaniche precedenti e garantiscono una finitura in grado di assicurare un ancoraggio idoneo per i successivi trattamenti di verniciatura. Dopo delle fasi di risciacquo si procede con la conversione e la passivazione dei prodotti, così da ottenere una maggiore resistenza alla corrosione e una perfetta adesione del film di vernice.

I pezzi pretrattati passano in forno per l'asciugatura per poi procedere verso la verniciatura vera e propria. In una prima cabina di verniciatura viene passata la mano di fondo, per poi passare la mano definitiva nella cabina immediatamente successiva. Per la verniciatura l'impianto 2230 usa dei robot antropomorfi ( Figura 2.3 ) in grado di svolgere 15 programmi di verniciatura

Figura 2.3: Robot antropomorfo usato per la verniciatura nell'impianto 2230.

ognuno caratterizzato da movimenti differenti in base al prodotto da verniciare. L'impiego di questi moderni robot ha permesso all'azienda di svolgere la fase di verniciatura in maniera completamente automatizzata e di avere un notevole risparmio per quanto riguarda gli sprechi di vernice. I robot reciprocatori usati nell'impianto già esistente possiedono una possibilità di movimento limitata in quanto sono in grado di muoversi solo dall'alto al basso e viceversa. Al contrario i robot antropomorfi usati nel 2230 riescono a svolgere movimenti diversi in base alla forma del prodotto così da avere un verniciatura più accurata e con meno sprechi. I pezzi, dopo la fase di verniciatura, passano di nuovo all'interno del forno per il processo di polimerizzazione delle polveri che garantisce la formazione di un film regolare e omogeneo e il suo indurimento. Dopo il raffreddamento attraverso dei buffer lungo il nastro trasportatore i prodotti in alluminio sono pronti per le lavorazioni successive.

## 2.2 Descrizione del problema

Attualmente si è lavorato sull'impianto 2230 per l'istallazione di un sistema di visione artificiale. Lo scopo che il sistema di visione ha all'interno dell'impianto è quello di acquisire, attraverso apposite telecamere, l'immagine della targa che

Figura 2.4: Targa identificativa della bilancella

identifica la bilancella e di riconoscere correttamente il numero inciso su di essa. Nella Figura 2.4 viene mostrato un esempio di targa.

Queste targhe non sono altro che delle placche metalliche lunghe 70 mm e alte 60 mm su cui è stato tagliato un numero a tre cifre che può andare da 000 a 450. Il numero riconosciuto dal sistema deve essere comunicato al gestionale dell'impianto, per informarlo del fatto che in quel particolare punto sta transitando la bilancella identificata dal numero letto e che questa possiede un certo numero di ganci su cui è appeso un determinato numero di pezzi. Il gestionale dell'impianto si occupa infatti di associare ad una bilancella, identificata dalla propria targa, un certo prodotto che deve essere verniciato con una determinata lavorazione dal robot antropomorfo. È proprio il gestionale che informa il robot su quali movimenti compiere o in che momento cambiare la colorazione della vernice.

Risulta chiaro il ruolo fondamentale del sistema di visione e dell'algoritmo di OCR su di esso applicato. Infatti un'errata lettura delle targhe numeriche comporterebbe il passaggio al gestionale di una informazione sbagliata, causando una verniciatura non adeguata al prodotto. Inoltre ci sarebbe anche il rischio di danneggiare il prodotto stesso o i robot antropomorfi. Infatti con una errata lettura, il gestionale comunicherebbe ai robot dei movimenti che non sono quelli coretti, rischiando così di urtare la vasca di verniciatura e di causare danni ingenti all'azienda. Letture errate della targhe possono essere causate dalla presenza di sporco e di residui di vernice, accumulati sulla targa a causa


Figura 2.5: Targa molto sporca che può causare una lettura errata

del continuo passaggio in cabina di verniciatura. Da qui nasce l'esigenza di sviluppare un algoritmo di OCR in grado di avvisare l'operatore umano su un abbassamento della percentuale di accuratezza della lettura, così da intervenire manualmente con la manutenzione della targa in modo da prevenire letture errate dei numeri. In Figura 2.5 viene mostrata una targa molto sporca che potrebbe causare una lettura errata.

## 2.3 Il sistema di visione

### 2.3.1 Computer Vision

La Computer Vision, o visione artificiale, è un sotto campo dell'intelligenza artificiale che si occupa di capire come i computer possano riprodurre processi e funzioni dell'apparato visivo umano. In particolare la sfida più ambiziosa della Computer Vision riguarda la visione high level (ad alto livello di astrazione e comprensione). Il sistema, partendo dall'immagine in 2D, deve riuscire a elaborare, ricostruire ed analizzare l'intero contesto in 3D in cui l'immagine è inserita. Un sistema di visione artificiale è costituito dall'integrazione di componenti ottiche, elettroniche e meccaniche che permettono di acquisire, registrare ed elaborare immagini sia nello spettro della luce visibile che al di fuori di essa. Il risultato dell'elaborazione è il riconoscimento di determinate caratteristiche dell'immagine per varie finalità di controllo, classificazione e per l'estrazione di informazioni utili per prendere decisioni.

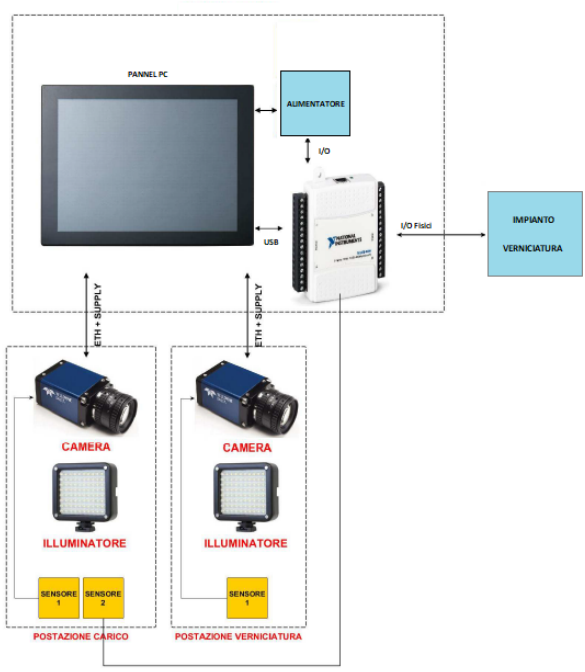


Figura 2.6: Architettura del sistema di visione

### 2.3.2 Installazione del sistema di visione

Un sistema di visione è costituito da diverse componenti: telecamere e ottiche, sistema di illuminazione, l'oggetto da esaminare, il sistema di acquisizione e di elaborazione dell'immagine e le interfacce uomo macchina. Le parti da ispezionare vengono posizionate di fronte a una o più telecamere ed illuminate in modo da evidenziare difetti nell'immagine. L'immagine, catturata e resa "comprensibile" da un calcolatore, potrà quindi essere elaborata con un apposito software in grado di individuare le caratteristiche dell'immagine allo scopo di eseguire i controlli e le verifiche per i quali il sistema è stato concepito.

La soluzione proposta per il sistema di visione applicato all'impianto di verniciatura 2230 di REVER si compone di: due camereNano, due illuminatori, dei sensori ottici usati sia come trigger per l'acquisizione dell'immagine sia per il conteggio dei ganci appesi alla bilancella, un modulo USB - I/O e un Panel PC 15" come interfaccia con l'operatore umano. Il Panne PC è collegato tramite cavo GigabitEthernet alle telecamere e si occupa della gestione di queste per il riconoscimento dei numeri. La restante gestione dei sensori, per il riconoscimento del numero di ganci e il passaggio della bilancella, avviene per mezzo di sensori a sbarramento, gestiti da un apposito Modulo I/O. Quest'ultimo

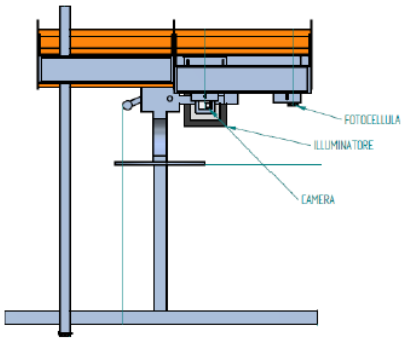


Figura 2.7: Struttura per l'installazione di telecamere e sensori- Vista frontale.

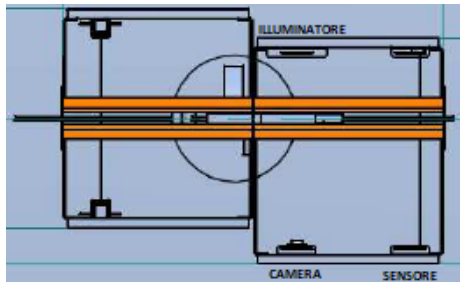


Figura 2.8: Struttura per l'installazione di telecamere e sensori- Vista in pianta.

comunica con il sistema di visione per mezzo di una connessione USB. In ultimo, lo scambio dati con il gestionale dell'impianto di verniciatura è gestito sempre con il modulo I/O. In Figura 2.6 viene schematizzata l'architettura del sistema di visione e vengono mostrati i vari collegamenti tra i componenti che lo vanno a costituire. In Figura 2.7 e in Figura 2.8 sono mostrate rispettivamente la vista frontale e la vista in pianta del progetto per la struttura che sorregge telecamere, illuminatori e sensori. Infine nella Figura 2.9 è possibile vedere la struttura definitiva montata sull'impianto.

Il sistema di visione è suddiviso in due stazioni come si può vedere in Figura 2.10. La prima è in corrispondenza della fine della zona di carico, dove gli operatori terminano di appendere i ganci e i prodotti da verniciare. Questa zona è stata individuata al fine di verificare, attraverso l'uso di una fotocellula, se la bilancella è vuota o presenta dei ganci appesi ed eventualmente quanti. Questa stazione è costituita dalla telecamera, dall'illuminatore, dal sensore per la cattura dell'immagine e dal sensore per il conteggio dei ganci. La seconda stazione è posizionata all'ingresso della cabina di verniciatura. In questa area le bilancelle hanno un punto di stop e poi ripartono, quindi, è stato necessario posizionare la telecamera, l'illuminatore e la fotocellula subito dopo. Questa zona è stata individuata al fine di comunica al gestionale il numero della bilancella e quindi il prodotto che sta per essere verniciato.

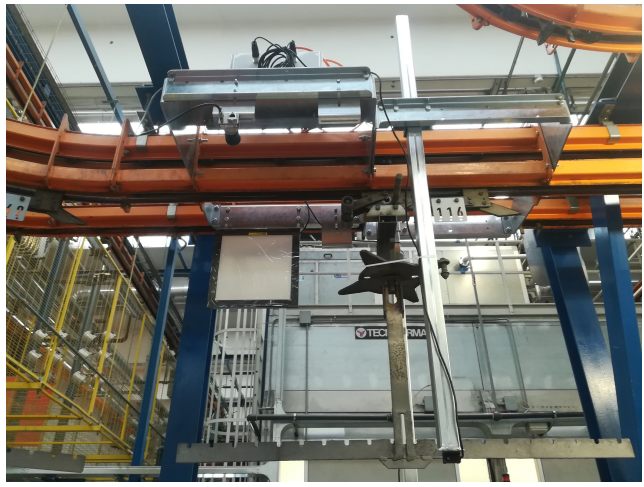

Figura 2.9: Struttura definitiva usata per l'installazione del sistema di visione

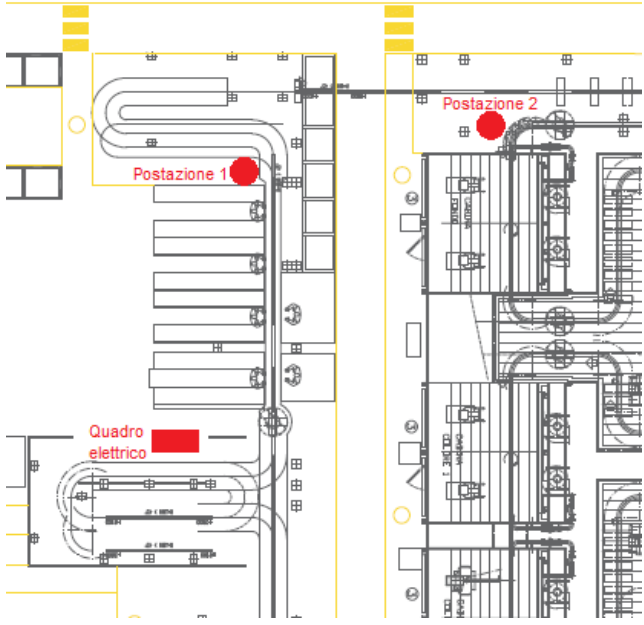


Figura 2.10: Planimetria dell'impianto con evidenziate le due stazioni che costituiscono il sistema di visione

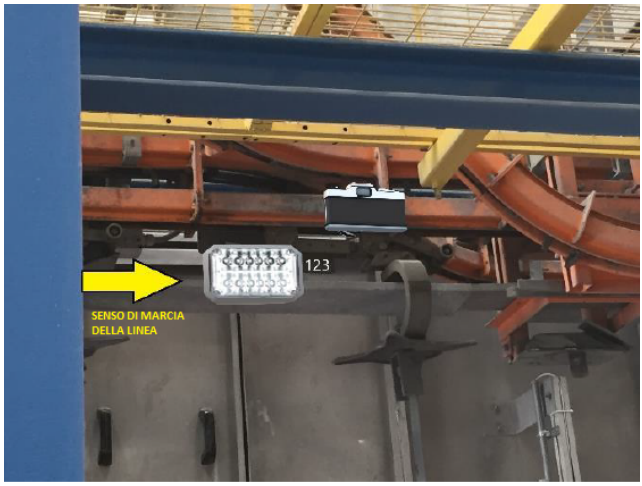


Figura 2.11: Schematizzazione del set up per l'acquisizione immagini

## 2.4 Campagna d'acquisizione immagini

Per lo studio e lo sviluppo dell'algoritmo di Optical Character Recognition il primo passo è stato quello di svolgere una campagna di acquisizione immagini, indispensabile per istruire l'algoritmo al corretto riconoscimento delle targhe numeriche. Questa fase si è svolta direttamente sull'impianto con il nastro trasportatore aereo in movimento alla velocità di 2 metri al minuto. Per l'acquisizione delle immagini è stata usata una cameraNano posizionata su un treppiedi e allineata perfettamente con la targa numerica. Applicando alla camera un obiettivo da 50 mm, la distanza ottenuta tra questo e il target è stata di circa 80 cm. Per il set-up definitivo questa distanza è proibitiva pertanto sono stati usati altri obiettivi da 12 mm, così da ottenere una distanza di 35 cm circa tra la camera e la targa. All'interno della linea di trasporto, dalla parte opposta rispetto la camera, è stato posizionato l'illuminatore retto da un'asta. In Figura 2.11 è stato schematizzato il set up per l'acquisizione delle immagini mentre in Figura 2.12 viene mostrato il set up definitivo usato. L'uso di questa configurazione ha permesso di acquisire delle immagini in controluce in cui i numeri e la targa sono ben visibili, senza i numerosi elementi di disturbo che sarebbero stati presenti in una immagine catturata senza l'uso dell'illuminatore. Con la velocità della linea a 2 metri al minuto e un tempo di campionamento della telecamera a 500 ms si ottiene che il numero da acquisire compare all'interno dell'obbiettivo per almeno 3 volte nitidamente. In un periodo di acquisizione di durata 6 ore circa, al rate di acquisizione di 500 ms sono state ottenute complessivamente 2559 immagini utili in cui le targhe di interesse compaiono completamente o solo parzialmente. Le immagini acquisite già in scala di


Figura 2.12: Set up definitivo usato per l'acquisizione delle immagini.

grigio con dimensioni pari a 2064x1544 pixel, sono state salvate in formato BMP.

Per lo sviluppo dell'algoritmo di Optical Character Recognition le immagini sono state divise in due gruppi: uno contenente le immagini in cui il numero a tre cifre compare solo parzialmente, cioè delle tre cifre ne sono ben visibili solo una o due, mentre l'altro gruppo contiene le immagini in cui la targa e il numero sono visibili interamente. Il primo gruppo, contenente 1609 immagini (62,88%), è stato usato soprattutto per la fase di Training così da istruire l'algoritmo al corretto riconoscimento dei numeri e per la generazione di una fuzione di OCR personalizzata. Il secondo gruppo, contenente 950 immagini (37,12%), è stato impiegato nella fase di Testing in modo da validare l'algoritmo e verificarne il corretto funzionamento.
Le fasi principali che hanno portato allo sviluppo del codice per l'algoritmo di OCR sono:

- Fase iniziale di pre-processing in cui le immagini sono state elaborate e migliorate, per eliminare elementi di disturbo e per rendere ben visibili i numeri.
- Fase di Training in cui le immagini processate sono state segmentate e attraverso l'applicazione OCR Trainer si è prodotta una funzione ocr personalizzata.
- Fase di Testing in cui la funzione ocr ottenuta è stata testata sulla restante parte di immagini non utilizzata nella fase precedente.

Nel capitolo seguente verranno approfondite queste 3 fasi di lavoro.

# Capitolo 3

# Sviluppo dell'algoritmo di OCR

Il primo passo per lo sviluppo dell'algoritmo di Optical Character Recognition è stato quello di svolgere una fase di elaborazione delle immagini acquisite, in modo da renderle adeguate alla fase successiva di addestramento dell'algoritmo. È indispensabile eliminare tutti gli elementi di rumore dalle immagini in quanto costituiscono un disturbo per l'OCR, rendono difficile il corretto riconoscimento delle cifre numeriche. Prima di ripercorrere le operazioni fondamentali svolte durante la fase di Preprocessing è indispensabile soffermarsi sulla definizione di immagine digitale.

## 3.1 Immagine digitale

Un'immagine digitale è la rappresentazione numerica di una immagine bidimensionale. Questa è composta da una matrice di punti, detti pixel, la cui colorazione è definita tramite uno o più valori numerici. I pixel vengono organizzati secondo una matrice 2D o 3D di numeri compresi tra 0 e 255, a seconda che l'immagine sia in scala di grigi o a colori. Le dimensioni di una immagine individuano il numero di righe e colonne con cui la matrice è fatta. Nel nostro caso le immagini hanno 2064 righe e 1544 colonne. I numeri all'interno della matrice di pixel individuano la scala di un determinato colore e quindi la sua intensità. Ad esempio, in un'immagine binaria (in bianco e nero) si ha solamente una matrice 2D in cui il numero 0 rappresenta il nero e il numero 255 rappresenta il colore bianco per un determinato pixel della matrice. In un'immagini in scala di grigi, ciascun pixel rappresenta l'intensità di un solo colore (immagine con un solo canale). La matrice 3D di un'immagine a colori, invece, include 3 canali: quello rosso, verde e blu (dall'inglese RGB, ovvero Red, Green e Blue). Ogni canale è composto da una matrice 2D, pertanto si ha una matrice per il canale rosso, una per il canale verde e una per quello blu, che poi vengono impilate una sopra l'altra per ottenere la matrice 3D che mostra l'immagine a colori.

Figura 3.1: Esempio di immagine usata nella fase di preprocessing - Targa pulita.

Figura 3.2: Esempio di immagine usata nella fase di preprocessing - Targa sporca.

## 3.2 Preprocessing

In Figura 3.1 e in Figura 3.2 è possibile vedere due esempi di foto usate nelle fasi di preprocessing. L'acquisizione delle immagini in controluce ha permesso di ottenere delle foto con uno sfondo completamente bianco e la targa che al contrario risulta essere scura. L'uso dell'illuminatore ha permesso di mettere in evidenza i residui di vernice che con il tempo vanno a riempire i tagli che costituiscono i numeri. Questi sono un forte elemento di disturbo per l'algoritmo di OCR in quanto non riesce a riconoscere il numero nella sua interezza ma lo considera costituito da tanti pezzi. Le immagini acquisite presentano la targa, cioè la regione di interesse (ROI), già in evidenza rispetto lo sfondo quindi non è stato necessario ritagliare la foto o svolgere dei filtraggi per migliorare la qualità dell'immagine. A volte infatti risulta impossibile procedere alla soluzione del problema con la stesura dell'algoritmo a causa della somiglianza tra le tonalità di grigio dell'oggetto e quelle dello sfondo. Di seguito è riportato lo script usato nella fase di preprocessing.

---

```
% CARICAMENTO DELLE IMMAGINI COME VARIABILI.
file_list = dir('ORIGINALI');

n_files=length(file_list);
dir_name='ORIGINALI';
for i=3:n_files
 eval(['im_' num2str(i) '=imread(file_list(i).name);']);
end

% PREPROCESING IMAGES DEF.
```

```
I=im_3;
%stampa dell'immagine
figure; imshow(I);

%Binarizzazione dell'immagine, con soglia a 70/255
bn= imbinarize(I,70/255);

%stampa dell'immagine binarizzata
figure; imshow(bn);

%BlobAnalysis per individuare lo sfondo.
%Ha area maggiore a quella della targa.
Hblob= vision.BlobAnalysis('MinimumBlobArea',
1104320,'LabelMatrixOutputPort',2);
[area,centroid,bbox,label]=Hblob(bn);

% sfondo bianco convertito in nero
im_n=I;
for i=1:1544
    for j=1:2064
        if label(i,j)==1
            im_n(i,j)=0;
        end
        j=j+1;
    end
    i=i+1;
end
figure; imshow(im_n);

%Binarizzazione dell'immagine, con soglia a 70/255
bn_n=imbinarize(im_n,70/255)
figure; imshow(bn_n);

%Chiusura morfologica dell'immagine per
%eliminare i tratti neri presenti sui numeri
disk=strel("disk",20);
im_c=imclose(bn_n, disk);

% Rimozione da immagine bianaria di oggetti
% con area minore a 7500 pixel.
% Produce un'altra immagine binaria.
% Operazione nota come area opening.
im_s= bwareaopen(im_c,7500);
```

```
figure; imshow(im_s);

%salvataggio dell'immagine elaborata
imwrite(im_s,'im_3.bmp');
```

### 3.2.1 Blob Analysis

Il primo passo è stato quello di mettere in evidenza i numeri rispetto il resto dell'immagine, per fare ciò si è cercato di rendere scuro lo sfondo bianco che contorna la targa. Da come è possibile vedere nello script, per ottenere questo risultato è stata usata la Blob Analysis di Computer Vision Toolbox ™. Nella visione artificiale la Blob Analysis è una tecnica che ha come obiettivo la rilevazione in una immagine di regioni di pixel connessi tra loro che differiscono in proprietà come luminosità o colore rispetto l'ambiente circostante.

Il blocco Blob Analysis calcola le statistiche per le regioni connesse individuate in un'immagine binaria, restituendo quantità come l'area, il centroide, il riquadro di delimitazione, la matrice dell'etichetta e il conteggio dei blob. Viene creato l'oggetto *vision.BlobAnalysis* e se ne impostano le proprietà. In questo caso è stata impostata la proprietà di *MinimumBlobArea* a 1104320 pixel per specificare che il blob di interesse non può avere un area minore di 1104320 (area della targa in pixel) così da non essere confuso con la targa. Inoltre è stata impostata la proprietà di *LabelMatrixOutputPort* a 2. Questa proprietà indica il numero massimo di regioni da etichettare in ciascuna immagine in input, specificato come numero intero scalare positivo. Attivando questa proprietà, viene restituita una matrice di etichette che descrive i blob trovati nell'immagine.

La Blob Analysis genera un oggetto che può essere usato come una funzione che accetta in input una immagine binaria e restituisce in output le proprietà dei blob trovati. Come si può vedere dallo script in questo caso è stata restituita l'area del blob, il centroide, il riquadro di delimitazione (Bounding Box) e la matrice di etichette. Quest'ultima è stata usata per convertire lo sfondo da bianco a nero in quanto la matrice presenta degli 1 in corrispondenza dei pixel del blob trovato che coincide proprio con lo sfondo. Per ottenere il risultato voluto è stato sufficiente impostare a 0 il valore dei pixel in corrispondenza dei quali la matrice di etichette valeva 1. In Figura 3.3 viene mostrata l'immagine ottenuta dopo questa operazione.

### 3.2.2 Binarizzazione

Il passo successivo è stato quello di uniformare la foto binarizzandola attraverso la funzione *imbinarize* di Image Processing Toolbox ™ con una soglia

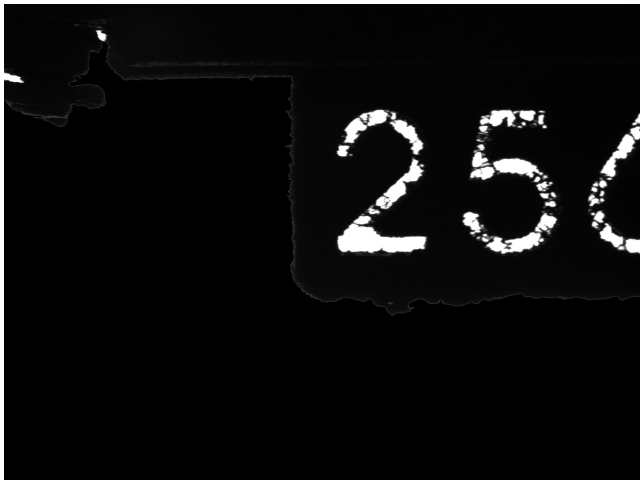

Figura 3.3: Immagine ottenuta dopo aver convertito lo sfondo da chiaro a scuso grazie all'uso della Blob Analysis.

opportuna. La funzione permette di creare un'immagine binaria da un'immagine in scala di grigi, sostituendo tutti i valori al di sopra di una soglia determinata globalmente con 1 (bianco) e impostando tutti gli altri valori su 0 (nero). Per impostazione predefinita, *imbinarize* utilizza il metodo di Otsu, che calcola la soglia ottima per separare le classi di pixel bianchi e neri in modo da ridurre al minimo la varianza intra classe. Questo tipo di sogliatura utilizza tecniche di analisi dell'istogramma dell'immagine ed è particolarmente adatto per immagini che presentano istogrammi con una netta separazione tra due picchi principali. La binarizzazione di Otsu si occupa proprio di trovare il punto di scissione fra tali picchi, che risulta essere il valore di soglia ottimale. Come è possibile vedere in Figura 3.4 l'istogramma dell'immagine con targa 256 presenta due picchi ben distinti, uno in corispondenza dei toni scuri e l'altro in corispondenza di quelli chiari. La soglia applicata è intorno al valore 70. Questo valore è stato ottenuto tramite sperimentazioni e sembra essere quello migliore per la maggior parte degli esempi considerati. Rendere l'immagine binaria da un lato migliora l'estrazione delle caratteristiche dei numeri, mentre dall'altro elimina gli elementi di rumore, consentendo un ritaglio più pulito e mirato delle cifre in fase di segmentazione. In Figura 3.5 è mostrata l'immagine risultante dall'operazione di binarizzazione.

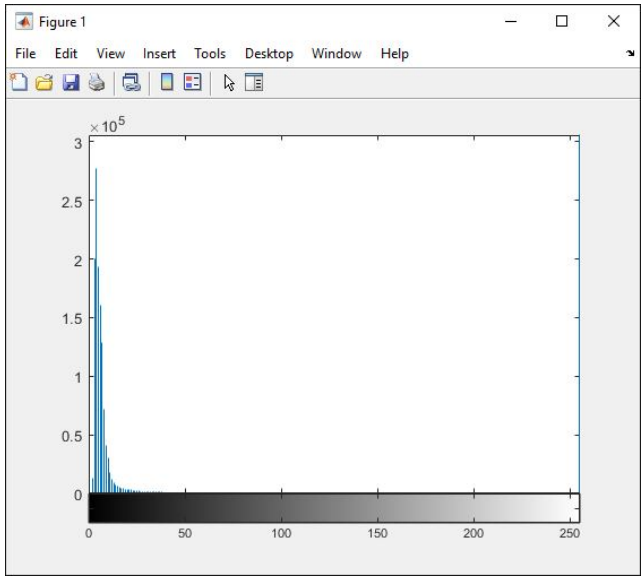


Figura 3.4: Istogramma dell'immagine parziale con targa numero 256.

Figura 3.5: Immagine binaria ottenuta attraverso la funzione *imbinarize*

Figura 3.6: Immagine ottenuta dopo la chiusura morfologica, priva dei filamenti neri in corrispondenza dei numeri.

### 3.2.3 Chiusura morfologica e rimozione degli elementi di disturbo

L'immagine ottenuta dalla binarizzazione presenta ancora i residui di vernice in corrispondenza dei numeri. Per la loro rimozione l'immagine è stata sottoposta ad alcune operazioni morfologiche con lo scopo di migliorare la definizione delle cifre. In particolare è stata svolta la chiusura morfologica sull'immagine binaria attraverso la funzione *imclose* di Image Processing Toolbox ™. La chiusura morfologica permette di riempire gli spazi vuoti all'interno di una immagine, rendendo nel nostro caso bianchi i tratti neri presenti sul numero. Questa operazione è una dilatazione dell'immagine binaria seguita da una sua erosione, utilizzando lo stesso elemento strutturante per entrambe le fasi. Con la dilatazione si allargano gradualmente i contorni delle regioni di pixel più chiari che individuano l'oggetto di interesse. In questo modo l'area dei pixel cresce in dimensione, mentre i vuoti in queste regioni diventano più piccoli fino a connettere oggetti separati da una distanza minore della dimensione dell'elemento strutturante. Nella chiusura morfologica questa operazione è seguita dall'erosione dell'immagine che opera in maniera contraria rispetto alla dilatazione, rimuovendo i contorni delle regioni di pixel espansi nella fase precedente senza ripristinare i vuoti. L'elemento strutturale scelto per la chiusura morfologica è stato un disco di raggio 20 pixel creato attraverso la funzione *strel*. In Figura 3.6 viene mostrata l'immagine ottenuta dopo l'applicazione della chiusura morfologica, in cui sono stati eliminati gli elementi neri di disturbo.

L'immagine ottenuta dopo l'applicazione di *imclose* presenta ancora delle

Figura 3.7: Immagine ottenuta dopo la rimozione degli elementi di disturbo bianchi attraverso la funzione *bwareaopen*.

aree di pixel bianchi che potrebbero essere confusi con il numero nella fase di riconoscimento. Questi oggetti sono stati eliminati attraverso la funzione *bwareaopen* messa a disposizione da Image Processing Toolbox ™. La funzione elimina gli oggetti che hanno un area inferiore a un certo numero di pixel dall'immagine binaria, producendone un'altra. L'area che è stata impostata è pari a 7500 pixel così da eleminare tutti gli oggetti più piccoli delle cifre numeriche. In Figura 3.7 è mostrata l'immagine ottenuta dopo la rimozione degli oggetti bianchi attraverso la funzione *bwareaopen*.

## 3.3 Training dell'algoritmo

Le immagini elaborate opportunamente nella fase di preprocesing sono state impiegate all'interno dell'applicazione OCR Trainer di MATLAB® per addestrare la funzione ocr per riconoscere lingue e font custom. Infatti il font usato per le targhe non è di comune uso in quanto è tipico per oggetti tagliati su metallo. Questa applicazione permette di svolgere le fasi di segmentazione dei numeri, di classificazione e il training della funzione in maniera semplice per l'utente grazie all'interfaccia interattiva.

Una volta aperta l'applicazione, attraverso l'apposita icona o attraverso linea di comando, si procede con l'avvio di una nuova sessione di training. Nell' OCR

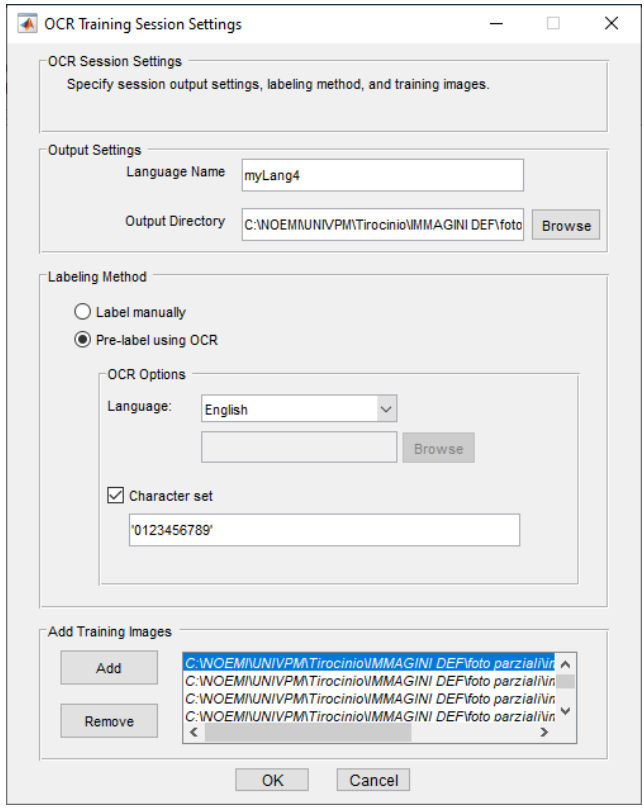


Figura 3.8: Finestra di dialogo per le impostazioni della sessione di training dell'OCR.

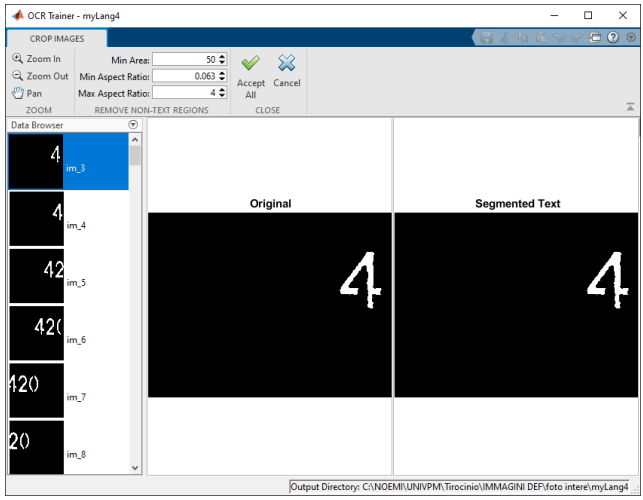


Figura 3.9: Finestra di dialogo per la segmentazione e la selezione della ROI.

Trainer è possiblie cliccare su Nuova sessione per aprire la finestra di dialogo per le impostazioni della sessione di training dell'OCR. Nelle impostazioni di output, è necessario inserire un nome per il file di dati della lingua OCR e scegliere la posizione della cartella di output per il file. La posizione specificata deve essere scrivibile. Nella sezione Metodo di etichettatura si sceglie se etichettare i dati manualmente o pre-etichettarli utilizzando il riconoscimento ottico dei caratteri. Se si utilizza l'OCR, è possibile selezionare la lingua inglese o giapponese pre-installata oppure è possibile scaricare file di supporto per la lingua aggiuntivi. Come è possibili vedere in Figura 3.8, il file è stato chiamato MyLang4 ed è stato salvato nella cartella C:\NOEMI\UNIVPM\Tirocinio\IMMAGINI DEF\foto intere. Come metodo di labeling (etichettatura) è stato scelto quello automatico usando l'OCR e la lingua scelta è stata l'Inglese, specificando anche il set di caratteri '0123456789'. È possibile aggiungere le immagini per il training in qualsiasi momento della sessione. Oggi volta che queste vengono aggiunte viene aperta l'interfaccia mostrata in Figura 3.9 in cui l'utente può disegnare una regione di interesse per selezionare una parte dell'immagine così da rimuovere eventuale rumore e migliorare i risultati della segmentazione. Il display mostra l'immagine originale a sinistra e quella modificata a destra. Conclusa la selezione della ROI è necessario cliccare su Accept all.

A questo punto l'OCR Trainer svolge automaticamente la fase di segmentazione e di classificazione delle cifre numeriche, esaminando le immagini caricate. Al completamento di queste fasi, viene aperta l'interfaccia visibile in Figura 3.10. Questa è caratterizzata da una finestra per la visualizzazione dei caratteri (character view window), una finestra per i dati riconosciuti (data browser

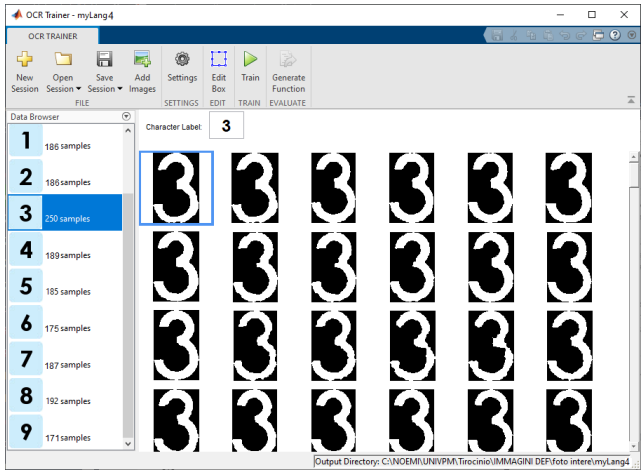


Figura 3.10: Finestra di dialogo per la correzione della segmentazione e della classificazione dei caratteri.

window) e dei pulsati per il controllo della app. Questi ultimi sono:

- Sessions: Per iniziare una nuova sessione, aprire una sessione già salvata, o aggiungere una sessione a quella corrente. È possibile nominare una sessione e salvarla come una file MATLAB.
- Add Images: Usato per aggiungere immagini all'inizio di una nuova sessione o dopo aver accettato un gruppo di immagini già inserito.
- Settings: Per settare o cambiare il font dei dati riconosciuti.
- Edit Box: Seleziona l'immagine che contiene il carattere selezionato, insieme ai riquadri di delimitazione. È possibile creare regioni aggiuntive, unirle, modificarle o eliminare immagini esistenti.
- Train: Crea un file di dati OCR dalla sessione.
- Generate Function: Crea una funzione di valutazione generata automaticamente per la verifica dei risultati dell'addestramento.

È possibile modificare i campioni estratti dalla finestra di visualizzazione dei caratteri. Per correggere i campioni, selezionare un numero nella finestra di visualizzazione dei caratteri e modificare le etichette utilizzando il campo Etichetta del carattere. Per escludere un campione dall'addestramento bisogna cliccare con il pulsante destro del mouse sul campione e selezionare l'opzione per spostare quel campione nella categoria Unknown. I campioni sconosciuti sono elencati nella parte superiore della finestra del browser dei dati e non vengono

Tabella 3.1: Numero di campioni usati nel training per ogni cifra.

| **Numero** | **Numero di campioni** |
|---|---|
| 0 | 188 |
| 1 | 186 |
| 2 | 186 |
| 3 | 189 |
| 4 | 189 |
| 5 | 185 |
| 6 | 175 |
| 7 | 187 |
| 8 | 192 |
| 9 | 171 |

utilizzati per l'addestramento. Se il riquadro di delimitazione ha ritagliato un carattere in maniera errata, fare doppio clic sul carattere e modificarlo nell'immagine da cui è stato estratto. Per ottenere un training dell'algoritmo uniforme si è cercato di utilizzare lo stesso numero di campioni per ogni cifra con lo scopo di avere la stessa accuratezza di lettura per ogni numero. Il numero di campioni usati nel trainig per ogni cifra è riassunto nella Tabella 3.1.

Dopo aver corretto i campioni, è stato avviato il training cliccando sul tasto Train. Quando il trainer ha completato l'addestramento, l'app crea un file di dati in lingua OCR e lo salva nella cartella specificata. Attraverso il tasto Generate Function è stata generata la funzione ocr personalizzata per il riconoscimento delle cifre numeriche tagliate sulle targhe.

```
% FUNZIONE OCR CUSTOM
function [ocrI, results] = myOcrDef(I, roi)

% Recupero dei dati dal file OCR prodotto
% dall'addestramento.
trainedLanguage = 'C:\NOEMI\UNIVPM\Tirocinio\
\IMMAGINI DEF\foto intere\myLang4\tessdata\
\myLang4.traineddata';

% Esecuzione OCR utilizzando il linguaggio addestrato.
layout = 'Block';
if nargin == 2
    results = ocr(I,roi, ...
    'Language',trainedLanguage, ...
    'TextLayout',layout);
else
```

```
    results = ocr(I, ...
    'Language',trainedLanguage, ...
    'TextLayout',layout);
end

ocrI = insertOCRAnnotation(I,results);

% Annota I con i risultati dell'OCR.
function J = insertOCRAnnotation(I,results)
text = results.Text;

I = im2uint8(I);
if isempty(deblank(text))
% Testo non riconosciuto.
text = 'Unable to recognize any text.';
[M,N,~] = size(I);
J = insertText(I,[N/2 M/2],text, ...
    'AnchorPoint','Center','FontSize', 24,
    'Font', 'Arial Unicode MS');

else
location = results.CharacterBoundingBoxes;

% Rimuove nuove righe dai risultati.
newlines = text == char(10);
text(newlines) = [];
location(newlines, :) = [];

% Rimuove spazi dai risultati.
spaces = isspace(text);
text(spaces) = [];
location(spaces, :) = [];

% Converte array di testo in array di celle di stringhe.
text = num2cell(text);

%Riempie l'immagine per aiutare ad annotare i risultati
% al bordo dell'immagine.
I = padarray(I,[50 50],uint8(255));
location(:,1:2) = location(:,1:2) + 50;

% Inserisce annotazioni di testo.
J = insertObjectAnnotation(I,'rectangle',location,text);
end
```


Figura 3.11: Esempio d'immagine usata nella fase di testing dell'algoritmo.

## 3.4 Testing

Per verificare la correttezza del Trainig e della funzione *ocr* generata è stato necessario testare l'algoritmo su un campione di immagini. Il materiale usato in questa fase consiste di 950 immagini in cui le targhe compaiono nella loro interezza come si può vedere in Figura 3.11. Questo set di immagini è stato diviso in due gruppi: il primo gruppo contiene quelle immagini in cui i numeri non sono ricoperti da residui di vernice e quindi sono ancora ben riconoscibili. Il secondo gruppo contiene invece quelle immagini in cui i numeri non sono riconoscibili a causa della elevata presenza di sporco. Ci aspettiamo che il software segnali la necessità di svolgere una manutenzione solo per il secondo gruppo di foto, cioè per quelle in cui la corretta lettera della targa è resa difficile dalla presenza di residui di vernice.

- **Primo gruppo** - targhe pulite - manutenzione NON NECESSARIA : 795 targhe (83,68%)

- **Secondo gruppo** - targhe sporche - manutenzione NECESSARIA: 155 targhe (17,89%)

Di seguito è stato riportato lo script usato per il testing.

---

```
% VALIDAZIONE FUNZIONE CUSTOM
```

```
I=int_1;
%stampa dell'immagine
figure; imshow(I);

%Binarizzazione dell'immagine, con soglia fissa a 70/255
bn= imbinarize(I,70/255);

%stampa dell'immagine binarizzata
figure; imshow(bn);

%BlobAnalysis per individuare lo sfondo.
%Ha un'area maggiore rispetto quella della targa
Hblob= vision.BlobAnalysis('MinimumBlobArea',1000000,
'LabelMatrixOutputPort',true);
[area,centroid,bbox,label]=Hblob(bn);

% sfondo bianco convertito in nero
im_n=I;
for i=1:1544
    for j=1:2064
        if label(i,j)==1
            im_n(i,j)=0;
        end
        j=j+1;
    end
    i=i+1;
end
figure; imshow(im_n);

%Binarizzazione dell'immagine, con soglia fissa a 70/255
bn_n= imbinarize(im_n,70/255);
figure; imshow(bn_n);

%Chiusura morfologica dell'immagine per eliminare
%i tratti neri presenti sui numeri
disk=strel("disk",8);
im_c=imclose(bn_n, disk);

% rimozione da immagine binaria di oggetti
% con area minore 10000 pixel.
% Produce un'altra immagine binaria.
%Operazione nota come area opening.
im_s = bwareaopen(im_c,10000);
figure; imshow(im_s);
```

```
%applicazione MYOCRDEF
[ocrI, results] = myOcrDef(im_s);
results.Text
results.CharacterConfidences

%CONTROLLO ACCURATEZZA PER MANUTENZIONE

filename = fopen ( 'letturatarghe.txt', 'w');
tx=results.Text;
Rc=results.CharacterConfidences;
if (Rc(1)>=0.80) && (Rc(2)>=0.80) && (Rc(3)>=0.80)
fprintf(filename,'%s %f %f %f manutenzione NON
NECESSARIA\n',tx,Rc(1),Rc(2),Rc(3));
else
fprintf(filename,'%s %f %f %f manutenzione
NECESSARIA\n',tx,Rc(1),Rc(2),Rc(3));
end
fclose(filename);
```

Le immagini sono state opportunamente elaborate in modo da mettere in evidenza i numeri rispetto lo sfondo. Nel fare ciò sono stati ripercorsi gli stessi passi svolti nella fase precedente di preprocessing. Per eliminare lo sfondo bianco è stata usata la Blob Analysis di Computer Vision Toolbox$^{TM}$. In questo caso è stata impostata la proprietà di *MinimumBlobArea* a 1000000 pixel in modo tale che la targa non venga considerata come parte dello sfondo. È stata impostata la proprietà di *LabelMatrixOutputPort* a 2 per indicare il numero massimo di regioni da etichettare. La matrice di etichette restituita dalla Blob Analysis è stata usata per convertire lo sfondo da bianco a nero, settando a zero tutti i pixel in corrispondenza dei quali la matrice label vale 1. Attraverso la chiusura morfologica svolta dalla funzione *imclose* di Image Processing Toolbox$^{TM}$sono stati eliminati i filamenti di vernice più piccoli che presentano un area minore uguale di 8 pixel. Questi residui non rendono il numero irriconoscibile ma restano comunque un elemento di disturbo per la funzione *ocr* per cui è stato necessario eliminarli. Attraverso la funzione *bwareaopen* messa a disposizione da Image Processing Toolbox$^{TM}$sono state rimosse tutte le regioni bianche che possiedono un area minore uguale a 10000 pixel. Questa operazione è stata necessaria in quanto l'algoritmo può confondere l'aree bianche con i numeri provocando una lettura errata.

L'immagine opportunamente processata va a costituire l'input della funzione di ocr personalizzata. Come è possibile vedere dallo script MatLab, la funzione *MyOcrDef* è in grado di riconoscere a partire dalla immagine in ingresso i

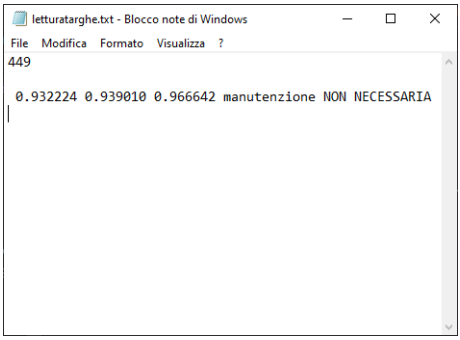


Figura 3.12: Messaggio di manutenzione NON NECESSARIA scritto sul file *letturatarga.txt*.

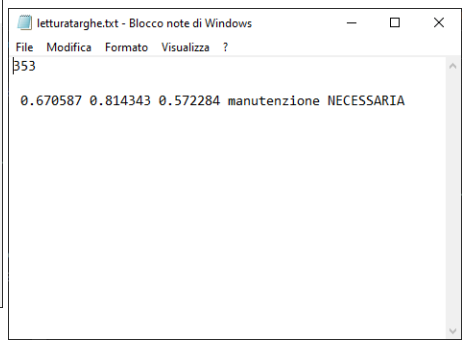


Figura 3.13: Messaggio di manutenzione NECESSARIA scritto sul file *letturatarga.txt*.

numeri tagliati con il font custom. Questa funzione può restituire in uscita: il testo riconosciuto, le bounding box che delimitano i caratteri, la precisione di lettura dei caratteri, le parole riconosciute, le bounging box che delimitano le parole e infine la precisione di lettura delle parole. In particolare per lo scopo dell'algoritmo sono stati di fondamentale importanza il testo riconosciuto e la precisione con cui i singoli caratteri sono stati letti. Infatti, per avvisare l'operatore della necessità di svolgere una manutenzione delle targhe, lo script verifica se l'accuratezza di lettura dei singoli caratteri è al di sotto di una soglia fissata all' 80% di accuratezza.

Nel caso in cui anche una sola cifra della targa è molto sporca e il taglio è fortemente occluso dai residui di vernice, l'accuratezza della lettura del carattere si presenta al di sotto della soglia stabilita. In questo caso lo script riporta in un file di testo *letturatarga.txt* il numero letto, le precisioni di lettura dei singoli caratteri e un avviso con scritto 'manutenzione NECESSARIA', proprio per sollecitare l'operatore ad intervenire con una pulizia delle targhe prima che avvenga una lettura errata. Al contrario se i numeri si presentano puliti e leggibili, si hanno delle percentuali di accuratezza molto elevate, dell'ordine del 90-95% . In questo caso sul file di testo viene riportato il messaggio 'manutenzione NON NECESSARIA' oltre al numero letto e l'accuratezza della lettura. In Figura 3.12 e in Figura 3.13 sono mostrati i due avvisi che possono essere scritti sul file *letturatarga.txt*.

La validazione dell'algoritmo con le 950 immagini ha portato ai seguenti risultati:

- Immagini riconosciute correttamente con accuratezza della lettura sopra la soglia: 798 immagini (84%)

- Immagini riconosciute correttamente con accuratezza della lettura sotto la soglia: 85 immagini (8,95%)

- Immagini non riconosciute correttamente: 67 immagini (7,05%)

Nella Tabella 3.2 sono riportate le percentuali di accuratezza medie della lettura dei singoli caratteri, considerando solo il gruppo d'immagini per cui non è stata segnalata la necessità di una manutenzione.

Tabella 3.2: Percentuali di accuratezza medie della lettura dei singoli caratteri.

| **Numero** | **Numero di campioni** | **Percentuale di accuratezza media** |
|---|---|---|
| 0 | 291 | 93,178% |
| 1 | 319 | 90,398% |
| 2 | 336 | 94,299% |
| 3 | 321 | 94,816% |
| 4 | 273 | 92,775% |
| 5 | 158 | 93,935% |
| 6 | 131 | 95,067% |
| 7 | 165 | 91,750% |
| 8 | 163 | 93,177% |
| 9 | 126 | 95,276% |

# Conclusioni e sviluppi futuri

In questo lavoro di tesi sono stati descritti e analizzati i passi fondamentali che hanno portato allo sviluppo di un algoritmo di Optical Character Recognition in grado di riconoscere correttamente targhe numeriche caratterizzate da un font custom. Lo scopo dell'algoritmo sviluppato è quello di svolgere una manutenzione predittiva di un impianto di verniciatura, con il fine di evitare guasti e problemi per la catena produttiva. L'uso di algoritmi di OCR in ambito industriale risulta essere particolarmente adatto soprattutto per scopi di manutenzione e di controllo dell'intero processo produttivo. L' impiego di questi software su sistemi di visione avanzati permette alle aziende una corretta gestione dei processi di automazione, monitorando il corretto svolgimento delle attività e l'adeguato funzionamento degli impianti produttivi. Il corretto riconoscimento di codici alfanumerici, impressi su prodotti o su elementi identificativi, permette il controllo del percorso seguito dai prodotti lungo la catena produttiva e l'acquisizione di tutta una serie di dati indispensabili per monitorare lo stato di degrado degl'impianti.

La scelta di sviluppare l'algoritmo di OCR attraverso il software MATLAB® è risultata vincente, in quanto ha permesso di semplificare alcune fasi dello sviluppo che risultavano essere complesse da implementare. Grazie alla configurazione usata durante la fase di acquisizione, è stato ottenuto un ampio dataset d'immagini di buona qualità, con in evidenza la targa e i residui di vernice in corrispondenza dei numeri. Queste immagini sono state elaborate facilmente attraverso le funzioni messe a disposizione dai Toolbox del software MATLAB®. In particolare l'uso dell'OCR Trainer ha permesso di svolgere in maniera semplice e interattiva le fasi di segmentazione e classificazione dei caratteri, garantendo un continuo controllo da parte dell'utente. Come è possibile vedere dai risultati mostrati al paragrafo 3.4, l'algoritmo di OCR sviluppato è in grado di riconoscere correttamente le targhe con elevata precisione. In aggiunta in presenza di targhe molto sporche questo procede con la segnalazione della necessità di manutenzione al gestore, prima che avvenga una lettura errata della targa. Pertanto il software risulta essere conforme alle specifiche date dall'azienda e risponde pienamente alle sue esigenze.

In definitiva, i risultati ottenuti sono validi e possono costituire la base per

eventuali miglioramenti o sviluppi futuri. Possibili miglioramenti potrebbero essere effettuati per aumentare le prestazioni dell'algoritmo e per migliorarne la precisione di lettura. A questo scopo si potrebbe svolgere una nuova campagna d'acquisizione immagini con il sistema di visone istallato con il set up definitivo. In questo modo si riuscirebbe ad aumentare la dimensione del dataset usato per il training dell'algoritmo e ad avere delle immagini più accurate. Per ottenere migliori prestazioni nella fase di training, nuove tecnologie di Machine Learning e d'Intelligenza Artificiale potrebbero essere impiegate per addestrare il software adeguatamente con un set d'immagini più ampio.

# Bibliografia


[1] S. Vaidyaa, P. Ambadb, S.Bhoslec: *Industry 4.0 – A Glimpse*; 2nd International Conference on Materials, Manufacturing and Design Engineering, 11-12 December 2017, MIT Aurangabad, Maharashtra, INDIA.

[2] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, B. Yin: *Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges*; IEEE Access 14 December 2017.

[3] C.K.M. Lee, Y. Cao, K. Hung: *Big Data Analytics for Predictive Maintenance Strategies*; Supply Chain Management in the Big Data Era (2017).

[4] J. Yan, Y. Meng, L. Lu, L. Li: *Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes and Applications for Predictive Maintenance*; IEEE Access (2017).

[5] N. Islam, Z. Islam, N. Noor: *A Survey on Optical Character Recognition System*; Journal of Information and Communication Technology-JICT Vol. 10 Issue. 2 December 2016.

[6] T. Caldeira, P.M. Ciarrelli, G.A. Neto: *Industrial Optical Character Recognition System in Printing Quality Control of Hot-Rolled Coils Identification*; Journal of Control, Automation and Electrical Systems (2019).

[7] K. Hamad, M. Kaya: *A Detailed Analysis of Optical Character Recognition Technology*; International Journal of Applied Mathematics, Electronics and Computers (2016).

[8] P. Chaturvedi, M. Saxena and B. Sharma: *A Bounding Box Approach for Performing Dynamic Optical Character Recognition in MATLAB*; Emerging Trends in Expert Applications and Security pp 117-123 (2018).

[9] O.P. Verma, E. Agarwal, C. Agrawal, A. Gupta: *Optical Character Recognition Using Minimal Complexity Machine and Its Comparison with Existing Classifiers*; Advances in Communication, Devices and Networking (2018).

[10] D. Hinduja, R. Dheebhika,T. Prem Jacob: *Enhanced Character Recognition using Deep Neural Network- A Survey*; International Conference on Communication and Signal Processing, April 4-6, 2019, India.

[11] R. Smith: *An Overview of the Tesseract OCR Engine*; Ninth International Conference on Document Analysis and Recognition, 23-26 Sept. 2007.

[12] S. Singh: *Optical Character Recognition Techniques: A survey*; International Journal of Advanced Research in Computer Engineering and Technology (IJARCET) Volume 2, Issue 6, June 2013.