

Riconoscimento di difetti di produzione sulla superficie di pompe meccaniche

Boller Mattia, De Rosso Daniel

Industry Lab

Corso magistrale in Data Science, Università di Milano Bicocca

Sommario: In questo progetto si è affrontato il problema della identificazione automatica di possibili difetti di produzione sulla superficie di pezzi meccanici. In particolare, si è cercato di riconoscere e localizzare anomalie dovute alla mancanza di materiale sulla faccia anteriore di pompe meccaniche per l'innessione di carburante, prodotte dall'azienda Bosch. La soluzione proposta si basa sull'utilizzo di una rete neurale con architettura autoencoder. L'intento è stato quello di ottenere un modello predittivo in grado di ricostruire l'immagine privandola di difetti, per poi valutarne le differenze con l'immagine originale. I risultati ottenuti dopo l'allenamento del modello, preceduto da preprocessing e data augmentation, hanno consentito la corretta classificazione delle immagini in anomale o meno, nel 85.19% dei casi, mentre, escludendo le immagini che il modello non riesce a ricostruire adeguatamente, richiedendo di conseguenza l'intervento umano, l'accuratezza raggiunta è stata del 95.83%.

1 Introduzione

L'era dell'Industria 4.0 ha portato ad un utilizzo massiccio dell'automazione nei contesti lavorativi. L'analisi dei dati va ad inserirsi nelle varie catene di produzione per permettere di ridurre al minimo il carico di lavoro nei compiti più ripetitivi e monotoni. Il valore aggiunto di queste pratiche non va ritrovato solamente nell'efficienza e nella maggiore produttività, ma anche nel perfezionamento della qualità generale, dovuto alla rimozione dei margini di errore umano e alla continua rifinitura dei modelli sviluppati, i quali hanno raggiunto (se non superato) le performance umane.

Per questo motivo, sempre più aziende effettuano investimenti volti al miglioramento delle infrastrutture e alla revisione dei piani strategici per massimizzare il ritorno dei capitali spesi.

In questo contesto si è voluto, tramite un caso studio reale, implementare delle tecniche di automazione per rilevare eventuali anomalie in pezzi di produzione. Questo problema è all'ordine del giorno nelle fabbriche a produzione intensiva, le quali ricercano un'elevata quantità di fab-

bricazione senza però rinunciare alla continuità di qualità nei loro prodotti. Risolvere questo problema significherebbe portare ad un livello superiore il rapporto efficienza su costo, aggiungendo anche componenti di scalabilità e di maggiore flessibilità nella catena di produzione.

In particolare, nel caso in questione, l'azienda Bosch, all'interno del suo ciclo di produzione di pompe meccaniche per l'innessione di carburante nei motori, sfrutta un controllo da parte di un operatore umano per rilevare possibili difetti sulla superficie del pezzo dovuti a mancanza di materiale. Nel seguente elaborato si propone una soluzione per automatizzare la fase sopradescritta.

1.1 Dataset fornito

Il dataset fornito consiste in 123 immagini in scala di grigi, ognuna ritraente una singola pompa meccanica per il carburante, fotografata dall'alto in modo che ne sia visibile la superficie. I pezzi non occupano sempre la stessa posizione all'interno della foto, ma sono tutti rivolti allo stesso modo, con differenze di angolazione minime. Le immagini presentano leggere differenze in termi-



(a) Pezzo non difettoso



(b) Pezzo difettoso

Figura 1: Esempi di immagini contenute nel dataset originale

ni di colore, segno che non sono state catturate sempre con le stesse condizioni di luce.

104 foto ritraggono esemplari privi di difetti, mentre nelle restanti 19 sono presenti delle anomalie sulla superficie del pezzo, visibili come piccole macchie scure dovute alla mancanza di materiale.

In figura 1 si possono vedere due esempi di immagini presenti nel dataset, una di un pezzo non difettoso, l'altra di un esemplare difettoso, in cui è possibile notare una macchia più scura nella parte sinistra.

1.2 Problema e soluzione proposta

L'approccio proposto in questo elaborato, consiste nello sfruttare le abilità di ricostruzione delle immagini degli autoencoder, un particolare tipo di rete neurale. In breve, il processo consiste nell'allenare il modello a rimuovere i difetti dalle immagini, in modo poi da poter confrontare la foto originale con quella ricostruita e tramite la loro differenza identificare e localizzare le possibili anomalie. Questo metodo prende spunto dal lavoro proposto da Anne-Sophie Collin e Christophe De Vleeschouwer nel paper "Improved anomaly detection by training an autoencoder with skip connections on images corrupted with Stain-shaped noise". [1] Particolare attenzione è stata posta sul rendere spiegabile il processo di identificazione dei difetti, andando quindi non solo a classificare un'immagine come anomala o non

anomala, ma andando anche a mostare quali parti della foto hanno portato a tale decisione. Questo perchè, in un contesto industriale, sapere come una determinata decisione sia stata presa da un certo processo automatico, è fondamentale ai fini di poterlo considerare affidabile.

Nel capitolo seguente vengono presentati i dettagli dell'approccio scelto, le modifiche apportate ai dati, il metodo di individuazione delle anomalie e i risultati ottenuti.

2 Note metodologiche

2.1 Preparazione dei dati

Il dataset originale è stato sottoposto ad una serie di operazioni di preprocessing per renderlo adatto all'allenamento del modello proposto. Le trasformazioni effettuate, in ordine sono:

- Crop: dall'immagine originale si è estratto il solo pezzo meccanico, andando ad eliminare lo sfondo. L'obiettivo è stato raggiunto andando ad applicare una funzione di threshold sul colore dei singoli pixel, portando a 0 (nero) tutti i pixel sotto il valore di 120 e a 1 (bianco) gli altri. Precedentemente a questo, si è applicato un filtro gaussiano all'immagine, andando a sfocare la foto per rendere netta la divisione tra oggetto e sfondo. Da questa maschera ottenuta, si sono estratte le coordinate del contorno rettangolare della zona bianca centrale (rappresen-

tante il pezzo meccanico) e si sono sfruttate per ritagliare l'immagine originale. In figura 2 si può vedere un esempio di maschera ottenuta attraverso l'operazione di threshold e il risultato del ritaglio dell'immagine.

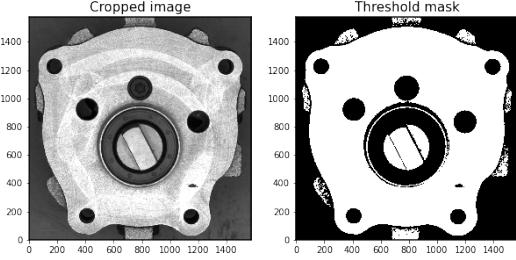


Figura 2: Esempio di operazione di crop ottenuto tramite una maschera di threshold

- Resize: le immagini ritagliate sono state portate ad una grandezza di 1024x1024 pixel, mantenendo le proporzioni originali dell'oggetto.
- Divisione in parti: le immagini sono state divise in 16 parti, ognuna di 256x256 pixel. Questa divisione è stata effettuata per aumentare la capacità della rete di considerare anche i dettagli più piccoli della foto da ricostruire.

Si è effettuata una divisione del dataset, destinando il 90% delle immagini di pezzi non difettosi al training e il restante al test. Mentre per quanto riguarda le foto di componenti difettosi, si è destinato il 20% dei dati per la fase di training e i rimanenti per la fase di test.

Sulle immagini adibite alla fase di training dei pezzi che non presentano anomalie, si è applicata data augmentation, applicando una rotazione casuale di 4° e modificando casualmente i valori di luminosità e contrasto. Questa operazione è stata effettuata per rendere il modello robusto a possibili foto scattate con angolazioni e condizioni di luci diverse. Si è quindi raddoppiato il numero di dati disponibili per allenare la rete, ottenendo 2976 immagini.

Come ultimo passaggio della preparazione dei dati, per ogni immagine di training si è effettuata una copia, aggiungendo ad essa del rumore

casuale in una forma simile alle anomalie presenti nei pezzi difettosi. In figura 3 si possono vedere due esempi di macchie aggiunte come simulazione di anomalie.

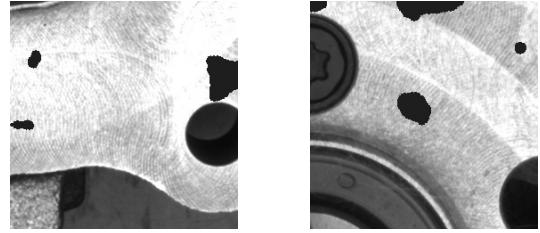


Figura 3: Esempi di rumore aggiunto alle immagini

Questo nuovo dataset rappresenta l'input del modello in fase di training, mentre le corrispondenti immagini prive di rumore vengono utilizzate come obiettivo di predizione dalla rete. Ulteriori informazioni sulle tecniche utilizzate per l'allenamento sono presentate nel paragrafo 2.3.

2.2 Modello - Autoencoder

Definito l'obiettivo e i risultati da ottenere, si è dedicato del tempo per effettuare ricerche sull'attuale stato dell'arte delle architetture utilizzate in casi analoghi.

Ne è emerso da tali indagini che un possibile approccio per la risoluzione del problema potesse essere l'utilizzo di modelli che imparassero in maniera autonoma a rimuovere le macchie, così da permettere un confronto con l'immagine passata in input. Successivamente a questo step è necessario l'impiego di metriche di similarità tra l'immagine predetta dal modello e quella passata gli, in modo da riuscire a discriminare se l'elaborazione del modello abbia portato cambiamenti significativi (indice di rimozione delle macchie).

In prima battuta si è quindi affrontato il problema utilizzando un autoencoder con layer convoluzionali al suo interno, notoriamente adatti per problemi trattanti le immagini. Un autoencoder è un particolare tipo di rete neurale, la cui architettura si compone di due parti: encoder e decoder. La prima parte si occupa di ridurre la dimensionalità del dato in input mentre la seconda, partendo dalla rappresentazione ridotta creata dal-

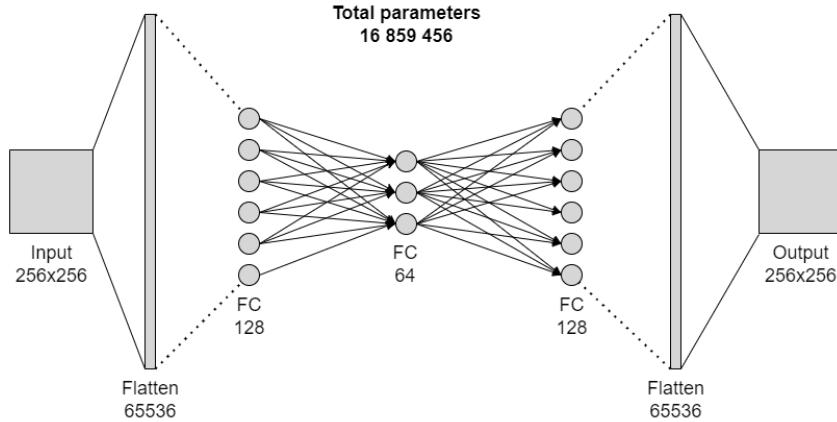


Figura 4: Descrizione architettura autoencoder

l'encoder, cerca di ricostruire il dato originale. I risultati ottenuti con i layer convoluzionali, non sono stati tuttavia soddisfacenti. I layer di upsampling infatti non erano in grado di ricostruire un'immagine sufficientemente simile a quella inserita nel modello, rendendo quindi difficile la discriminazione delle macchie artificiali rispetto alla scarsa precisione ottenuta dall'architettura.

Si è quindi deciso di rimuovere i diversi layer convoluzionali per passare ad un modello con soli layer fully connected. Questo perchè l'obiettivo da raggiungere è stato quello di ricreare un'immagine il più possibile simile a quella di partenza, senza quindi perdere troppa informazione nei passaggi tra i diversi layer. Dopo diverse prove di configurazioni e architetture, si è raggiunto un risultato soddisfacente con il modello mostrato in figura 4.

Come si può notare dalla figura citata, l'autoencoder riceve in input un'immagine di 256x256x1 dimensioni, poi appiattita da un layer flatten che si connette ad un layer fully connected prima di 128 neuroni e poi di 64. La stessa architettura va poi specchiata per formare il decoder.

2.3 Training

Il modello presentato in precedenza è stato studiato per lavorare con le immagini preprocessate secondo i vari step, descritti dettagliatamente nel paragrafo 2.1. L'intera fase di training viene infatti svolta passando al modello una grande quantità di immagini (originariamente

senza difetti) a cui sono state applicate delle trasformazioni per inserire del rumore artificiale in esse. Il training e l'ottimizzazione vengono quindi condotti imponendo al modello di ridurre l'errore MSE (mean squared error) tra l'immagine con rumore passata in input e l'immagine originale passata come output. Il modello sfrutta poi l'optimizer adamax per la ricerca dei migliori parametri che riducano la loss MSE. Non essendo chiaro a priori il tempo richiesto per l'allenamento ottimale del modello, si è applicata la tecnica di regolarizzazione di early stopping, che permette di terminare la fase di training dopo 20 epoche che non hanno riportato miglioramenti, di almeno 0.0001 punti, nella loss di validazione. Dopo una fase di training durata 82 epoche, si è raggiunto un valore MSE di 0.0037 nel set di validazione. Il modello ottenuto viene salvato per essere utilizzato anche nelle fasi successive.

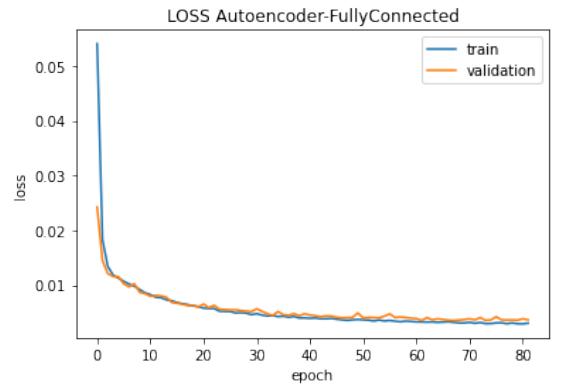


Figura 5: Loss train e validation per epoche

Dal grafico in figura 5 si può osservare come il modello non entri mai in uno stato di overfitting, ma piuttosto che l'errore di validazione riesca a seguire in maniera ottimale quello di training, rendendo il modello ottimizzabile con una buona granularità.

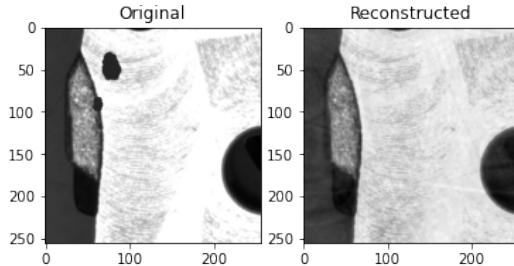


Figura 6: Esempio di predizione del modello con immagine ricostruita senza rumore

Come risultato finale della fase di training vengono mostrati in figura 6 un sample di training, a partire da un'immagine con l'aggiunta di rumore artificiale, e il conseguente output del modello che riesce a ricostruire l'immagine originale priva di difetti con un'elevata precisione.

2.4 Identificazione e localizzazione di anomalie

Attraverso le immagini ricostruite dal modello e quindi prive di anomalie, è stato possibile identificare la presenza di uno o più difetti ed eventualmente individuarne la posizione all'interno della foto.

Il processo di riconoscimento delle anomalie viene presentato graficamente in figura 7 e descritto di seguito. Il primo punto consiste nel processamento delle 16 parti di una singola imma-

gine (figura 7a) da parte dell'autoencoder, ogni divisione viene quindi ricostruita e privata degli eventuali difetti presenti. Si concatenano in ordine le sezioni in modo da ricomporre la foto completa (figura 7b), la quale viene confrontata con l'immagine originale tramite differenza assoluta. Questa operazione consente di mettere in risalto le zone diverse tra le due figure, quindi, se l'immagine è stata ricostruita correttamente, di evidenziare i difetti (descritti da zone di colore più chiaro nell'immagine di differenza) (figura 7c). Attraverso l'applicazione di un threshold, come si è fatto nell'operazione di crop (paragrafo 2.1), è possibile filtrare le leggere differenze dalle diversità più marcate, andando a creare una maschera che mette in risalto le sole anomalie (figura 7d). Dalla maschera appena descritta, si estraggono i confini rettangolari delle zone di colore bianco, in modo da poter localizzare i difetti rilevati (figura 7e).

Il metodo descritto restituisce buoni risultati solo se l'immagine viene ricostruita correttamente dall'autoencoder, in quanto, se la ricostruzione presenta differenze in alcune zone non dovute alla presenza di difetti, queste vengono riconosciute come anomalie e quindi incorrettamente classificate come imperfezioni. In figura 8 è presente un esempio della differenza tra un'immagine e la sua ricostruzione non corretta da parte del modello.

Come si può notare, l'immagine di differenza presenta molteplici zone di colore chiaro, soprattutto nei bordi, segno di una ricostruzione non perfetta. Per gestire questo problema si è introdotta una misura di somiglianza tra la figura ori-

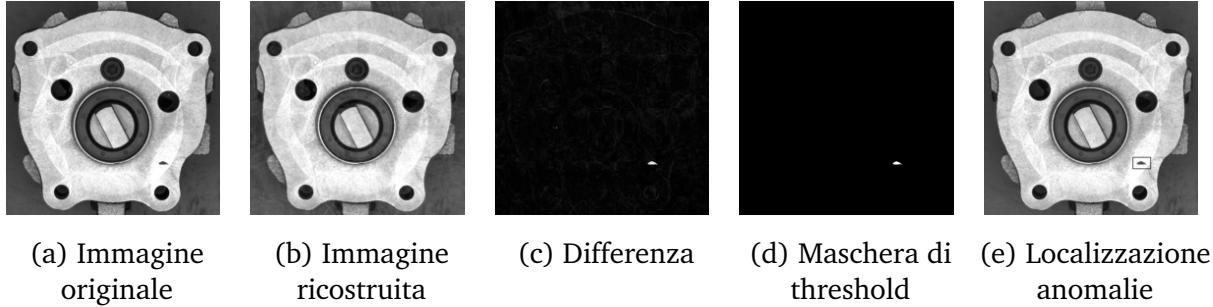


Figura 7: Rappresentazione grafica degli step che portano all'identificazione e localizzazione dei difetti

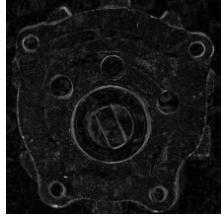


Figura 8: Differenza tra un'immagine e la sua ricostruzione errata

ginale e quella processata dal modello, in modo da rilevare i casi in cui la ricostruzione non sia andata a buon fine. La metrica scelta è stata la SSIM (Structural Similarity Index), una metrica perettiva che quantifica il degrado della qualità dell'immagine causato da elaborazioni come la compressione dei dati o da perdite nella trasmissione dei dati. Il suo valore è compreso tra -1 e 1, con 1 che indica totale uguaglianza tra le immagini. [2] Lo Structural similarity index tra due sezioni x e y di grandezza comune $N \times N$, si calcola come:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (1)$$

con:

- μ_x la media di x ;
- μ_y la media di y ;
- σ_x^2 la varianza di x ;
- σ_y^2 la varianza di y ;
- σ_{xy} la covarianza di x e y ;
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ due variabili per stabilizzare la divisione con denominatore debole;
- L l'intervallo dinamico dei valori di pixel;
- $k_1 = 0.01$ e $k_2 = 0.03$ di default.

Per portare alcuni esempi, il valore di SSIM calcolato sull'immagine in figura 7a e la sua ricostruzione in figura 7b è pari a 0.95, mentre la SSIM calcolata sulle foto che hanno generato l'immagine di differenza presente in figura 8 ha valore pari a 0.63.

2.5 Risultati

Le performance dell'intero processo di indentificazione e localizzazione dei difetti, composto dalla ricostruzione delle immagini da parte del modello autoencoder e dalla successiva analisi delle differenze, sono state valutate sul dataset riservato alla fase di test, composto da 11 foto di pezzi non difettosi e 16 foto di componenti difettosi. Le immagini sono state classificate come difettose se almeno un'anomalia è stata rilevata al loro interno, e non difettose altrimenti. Le ricostruzioni che presentavano un valore di SSIM inferiore a 0.75, sono state considerate come non valide, quindi impossibili da classificare da parte del modello. In ottica di utilizzazione della soluzione proposta in ambito industriale, un valore di Structural similarity index troppo basso può essere utilizzato per indicare la necessità di un controllo umano sul pezzo in questione.

In termini di performance, considerando le immagini non processabili ($SSIM < 0.75$) come errori di classificazione, il processo ha ottenuto un valore di accuratezza pari al 85.19%, con tutti i componenti non difettosi classificati correttamente, 1 componente difettoso classificato come non difettoso e 3 componenti difettosi non processabili. Un controllo visuale sull'unico pezzo anomalo classificato come non anomalo, ha fatto notare come non si riscontrino effettive imperfezioni sulla sua superficie, facendo pensare ad un possibile errore in fase di raccolta di immagini e costruzione del dataset più che un errore del modello. Non considerando le figure non processabili dal modello nel calcolo delle performance (necessitando di un controllo umano), il processo ha ottenuto un'accuratezza pari al 95.83% (contando sempre l'unico errore sulla possibile immagine priva di difetti, ma etichettata nel dataset come difettosa).

In figura 9 sono presentati degli esempi di classificazione del processo di identificazione di anomalie, compresi di etichetta identificativa di pezzi difettosi o non (True o False), valore di SSIM per identificare immagini non processabili e localizzazione delle imperfezioni sulle foto.

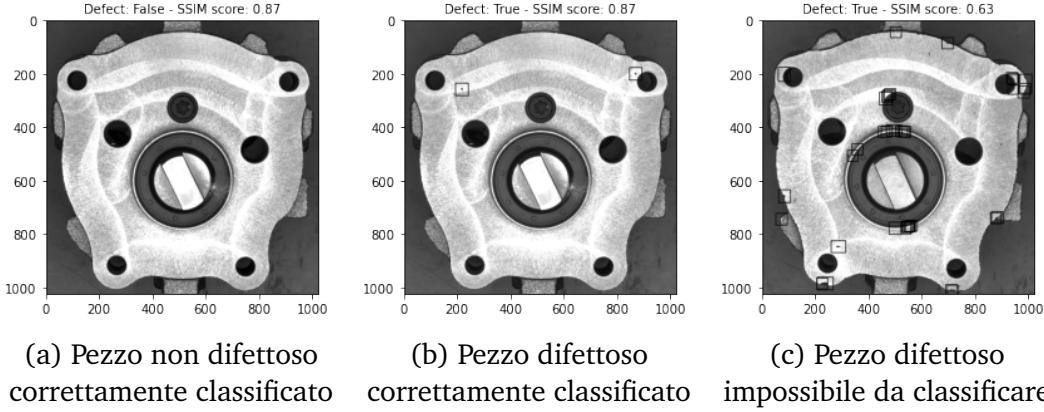


Figura 9: Esempi di classificazione eseguiti dal processo

3 Conclusioni e sviluppi futuri

Tramite il seguente elaborato si è stati in grado di dimostrare come sia possibile portare l’automazione anche in contesti solitamente affidati a mansioni umane. Con una corretta modellazione del problema e le relative funzioni di manipolazione dei dati, si è infatti raggiunto un grado di precisione soddisfacente e che ha margine per possibili sviluppi e perfezionamenti.

Concludendo questo elaborato si vogliono riportare possibili migliorie in fase di raccolta dati, utili per rendere al meglio le performance del modello. Come precedentemente esposto, per affrontare il problema è stato necessario svolgere un’ampia fase di data augmentation per via del dataset ridotto. La possibilità di avere a disposizione un grande dominio di immagini reali renderebbe il problema più generalizzabile, in termini di varietà di difetti, e allo stesso tempo ne aumenterebbe la specificità al caso d’uso in esame, migliorando di conseguenza le performance.

Inoltre sarebbe utile poter standardizzare il processo di acquisizione delle immagini e la posizione dei vari pezzi nella scena. Come emerso nella fase di test infatti, il modello in alcune eccezioni non riesce a ricostruire le immagini ad un livello tale da permetterne l’identificazione di difetti sulla superficie. Probabilmente questo è causato dal fatto che alcune foto dei pezzi siano state scattate con angolature e prospettive differenti rispetto alla media. Ciò va ad inficiare

le performance finali, poichè il modello identifica come anomalie anche le zone che non sono effettivamente problematiche.

Essendo a conoscenza dei dettagli implementativi del processo in azienda, un suggerimento che si può proporre, nel caso venisse implementata la soluzione proposta, riguarda l’aggiunta di attuatori nell’infrastruttura per riportare informazioni aggiuntive. Nel dettaglio si pensa a segnali luminosi specifici in caso di scarso punteggio di ricostruzione delle immagini, per comunicare la necessità di un controllo umano, e di schermi dedicati per mostrare a video dove sono stati riscontrati problemi nei pezzi difettosi.

Possibili sviluppi futuri possono riguardare il test di architetture diverse per la ricostruzione delle immagini, come ad esempio reti GAN (generative adversarial network) [3], oppure approcci diversi dall’anomaly detection, come potrebbe essere una object detection tramite modelli specifici come YOLO (You Only Look Once) [4] o Fast R-CNN [5].

Riferimenti bibliografici

- [1] A.-S. Collin and C. De Vleeschouwer, “Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise,” 2020. [Online]. Available: <https://arxiv.org/abs/2008.12977>
- [2] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 13, no. 4, pp. 600–612, 2004.
- [3] Z. Wang, Z. Wang, Q. She, and T. E. Ward, “Generative adversarial networks in computer vision: A survey and taxonomy,” *arXiv: Learning*, 2020.
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” 2015. [Online]. Available: <https://arxiv.org/abs/1506.02640>
- [5] R. Girshick, “Fast r-cnn,” 2015. [Online]. Available: <https://arxiv.org/abs/1504.08083>