

# Autoencoder per l’Anomaly Detection applicati in ambito AutoMotive

Laureando: **Tristano Munini**  
Relatore: **Prof. Giuseppe Serra**  
Correlatore: **Ing. Daniele Fornasier**



Università di Udine, Dipartimento di Scienze Matematiche, Informatiche e Fisiche

## Sommario

In questa tesi si affronta il problema di rilevare la presenza di colla sul fondo di carcasse metalliche per motori elettrici. Il *dataset* fornito, composto da qualche migliaio di immagini, comprende due classi, ma una di queste è rappresentata da un numero veramente esiguo di esemplari. Si è deciso di utilizzare tecniche di *Machine Learning* per il riconoscimento di anomalie. In particolare è stato sviluppato un modello basato su reti neurali del tipo *autoencoder*, poiché permettono di essere allenati utilizzando solamente una classe. Applicando algoritmi di manipolazione e trasformazione delle immagini si è migliorato il *dataset*. La soluzione proposta si dimostra all’altezza dei risultati attesi, perché ha permesso di identificare quasi tutte le colle, senza però generare troppi falsi positivi. In questo studio non è stato possibile determinare se la soluzione proposta sia applicabile in modo affidabile in ambienti industriali. A tal fine sarebbe necessario ampliare il *dataset* ed effettuare delle verifiche sul campo. Il lavoro viene comunque considerato positivo perché ha permesso di esplorare le capacità combinate di algoritmi di manipolazione immagini e degli *autoencoder*.

## Introduzione

I pezzi da analizzare hanno una forma cilindrica cava con una delle due estremità sigillata. All’intero della cavità verrà alloggiato un motore elettrico per tergicristalli. Per poter fissare il motorino è richiesto che un anello di colla sia versato sulla parete verticale interna del pezzo. Poiché la colla è liquida è possibile che goccioli fino a raggiungere il fondo del pezzo, questo comporta il malfunzionamento del motore. Da ciò nasce la necessità di rilevare in modo automatizzato quali pezzi presentano colla sul fondo, questo comporta il malfunzionamento del motore. Da ciò nasce la necessità di rilevare in modo automatizzato quali pezzi presentano colla sul fondo, queste verranno poi rimosse automaticamente dal processo di produzione. La soluzione proposta è stata sviluppata durante un tirocinio della durata di sei mesi svolto presso beanTech. Il principale strumento utilizzato è stato il linguaggio di programmazione *python*, arricchito da librerie come: *opencv*, *numpy*, *pytorch*, *matplotlib*.

## Obiettivi

Per quanto riguarda gli obiettivi che si vogliono raggiungere sappiamo che: la colla viene depositata su circa 5000 pezzi al giorno; la probabilità che gocce di colla cadano sul fondo delle carcasse è estremamente bassa. Purtroppo non esistono dati numerici esatti ma si stima che il processo di produzione abbia un tasso d’errore di una carcassa al mese o poco più. Questi dati possono essere trasformati in probabilità approssimative osservando che:

colla depositata al mese:  $5000 * 31 = 155000$   
colla depositata male al mese: 2

Quindi la probabilità che venga prodotta una carcassa non idonea è pari allo 0.00001%. Il sistema di intelligenza artificiale deve riconoscere i pezzi non conformi ma soprattutto, tenendo conto della probabilità di cui sopra, deve generare un numero bassissimo di falsi positivi. Ricordano che per falsi positivi (detti anche FP o *False Positive*) si intendono tutte le carcasse che il sistema considera non conformi ma che in realtà non presentano difetti. Un’intelligenza artificiale troppo rigida, che, quando indecisa, propende per scartare il pezzo, inciderebbe negativamente sulla produzione. Si rischierebbe infatti di creare un enorme danno economico andando a scartare molte più carcasse del necessario. Il nostro obiettivo è creare un sistema che generi un numero di falsi positivi che sia inferiore al 2%, cercando di riconoscere più carcasse non conformi possibili.

## Soluzione Proposta

La soluzione proposta può essere divisa in tre momenti: nel primo l’immagine in ingresso viene manipolata da algoritmi creati appositamente, così da ottenere un’immagine che favorisce le future trasformazioni; l’immagine appena generata viene fornita in ingresso ad un *autoencoder* con una specifica architettura; l’*output* della rete neurale viene gestito da un ulteriore algoritmo che permette di sintetizzare una classificazione binaria. Si è deciso di utilizzare gli *autoencoder* perché possono essere allenati utilizzando una sola classe, il nostro scopo è quello di creare una rete capace di “rimuovere” la colla dalle immagini. Un *autoencoder* è formato da due componenti principali:

l’*encoder* comprime il dato in ingresso, mappandolo in uno spazio di dimensionalità inferiore; il *decoder* prende il dato compresso e lo rimappa nello spazio di partenza. Poiché la rete viene allenata solo sui conformi è lecito pensare che, in fase di compressione, l’informazione della colla non venga codificata. Quindi non è possibile che uno scarto venga ricostruito interamente. Un algoritmo di *post-processing* verifica dove *input* ed *output* differiscono, sintetizzando la classificazione binaria definitiva.

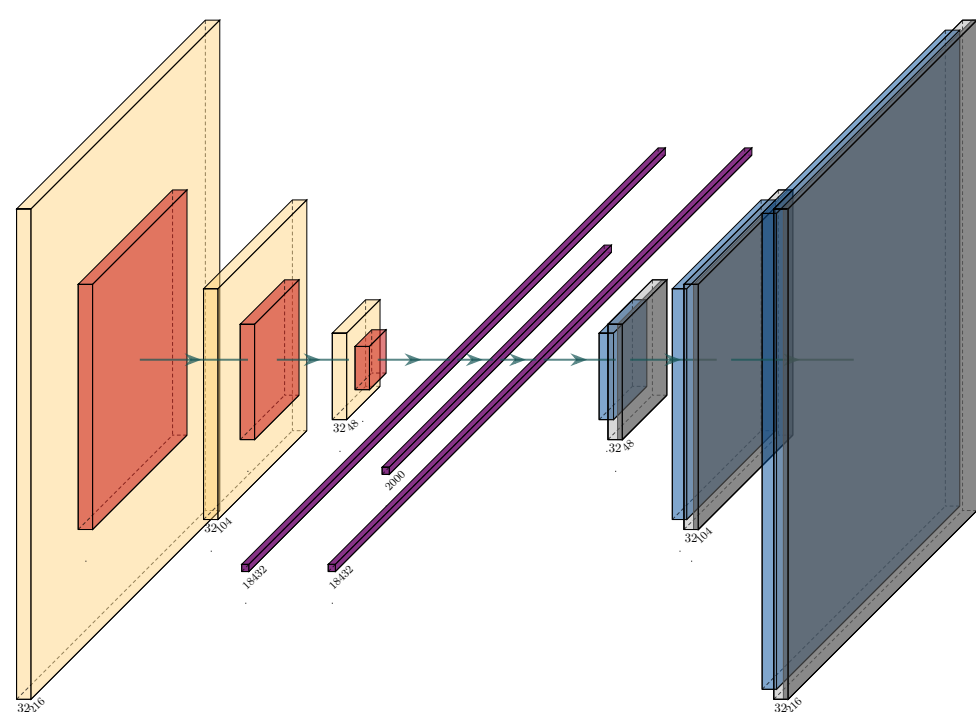


Figura 1: Architettura della rete proposta

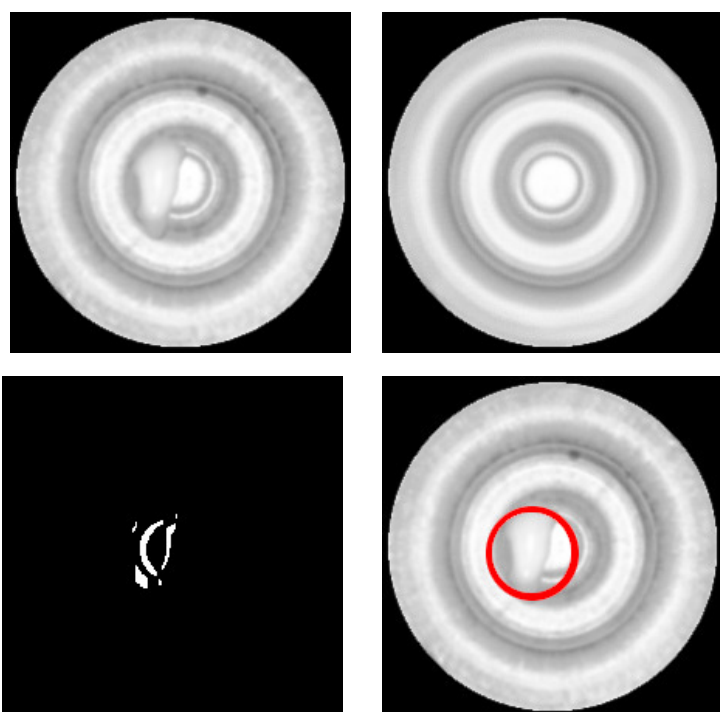


Figura 2: Immagini di: ingresso, uscita, differenza e risultato

## Analisi Sperimentale

**Dataset** Il *dataset* comprende 1749 immagini di cui solo 30 sono scarto, ovvero presentano colla sul fondo. In casi come questo si dice che il *dataset* è sbilanciato. Solitamente un *dataset* comprende centinaia di migliaia di immagini e le classi di cui è composto sono distribuite equamente. Ulteriori problematiche delle immagini fornite riguardano: centratura e distanza dal fondo differenti; il pezzo si presenta ruotato in modo casuale; le superfici delle carcasse sono simili ma hanno delle caratteristiche uniche (macchie, effetto “sale e pepe”, ...); luminosità variabile. Quest’ultimo punto, assieme alla rotazione, ha rappresentato una delle difficoltà principali.

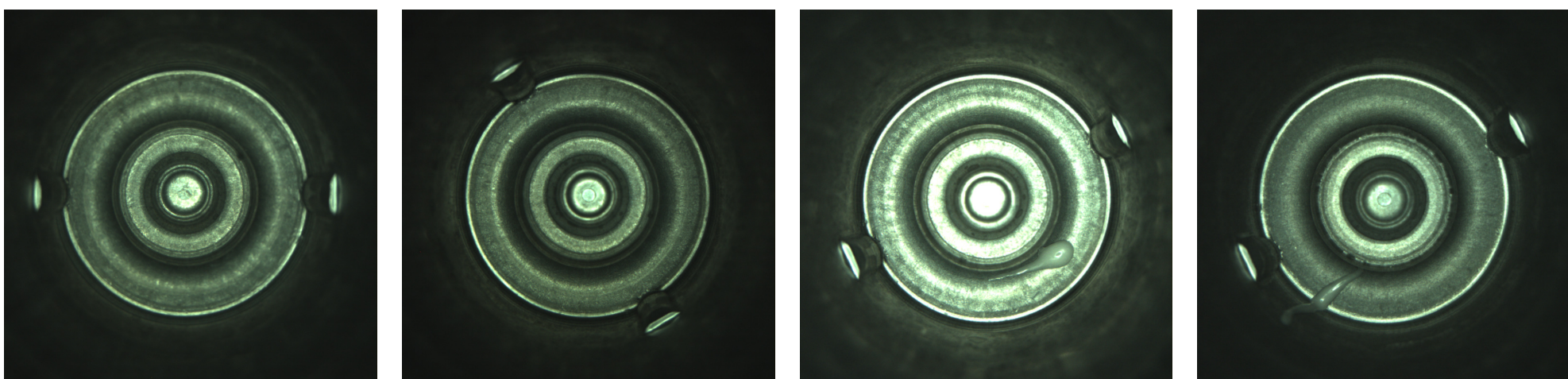


Figura 3: Due conformi e due scarti

**Risultati Ottenuti** Nelle tabelle sono riportati i risultati della soluzione proposta. Si fa notare che con *recall* si indica la percentuale di scarti effettivamente riconosciuta.

Classe	Elementi	Predetti come KO	Percentuale
Conformi	314	0	0%
Scarti	30	28	93.3%
Scarti sintetici	30	28	93.3%

Accuracy	0.989%
Precision	1.000%
Recall	0.933%

## Conclusioni

Osservando i risultati ottenuti si può vedere che, nonostante il campione a disposizione abbia una numerosità ridotta, nessun conforme dell’insieme di *test* è stato classificato in modo scorretto. Questo significa che c’è una quantità di falsi positivi, cioè di carcasse conformi che verrebbero scartate, pari allo 0%. Verificando le capacità del sistema anche sull’insieme d’allenamento, risulta che lo 1.1% di elementi è stato classificato come scarto. Tra questi si trovano principalmente foto scattate ad una distanza superiore alla media. Il numero di scarti correttamente identificato corrisponde al 93.3%. Si può concludere che la soluzione proposta ha dato risultati vicini a quelli attesi e che quindi sia soddisfacente. Il *dataset* ristretto, però, non permette di determinare se la soluzione proposta sia applicabile in modo affidabile in ambienti industriali. Studi futuri potrebbero comprendere un campione più ampio sia di conformi che di scarti e sperimentazioni con *autoencoder* con architetture differenti.