

CNN-WheelClassifier

Modello di IA per l'analisi delle ruote da pattinaggio:
valutazione dell'usura e ottimizzazione della loro
posizione.



Agenda

- 01 **Contesto e Obiettivi**
- 02 Dataset
- 03 Modelli
- 04 Applicazione Web
- 05 Conclusioni



500 m Sprint JUNIORES F - FINALE

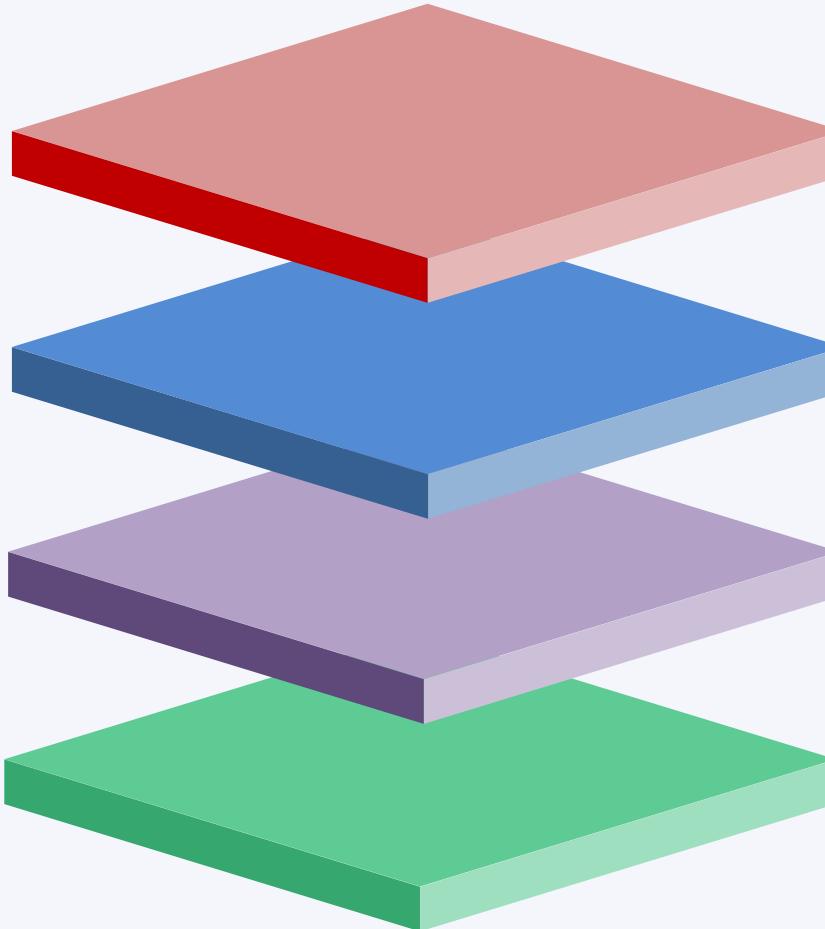
Obiettivi Progetto

PROBLEMA

Le ruote dei pattini sono soggette a usura. Per ottimizzarne l'utilizzo, il pattinatore fa uno screening **visivo** delle ruote, e le riposiziona.

SVILUPPO SOLUZIONE AI

Tecniche di computer vision applicate con CNN in Python



OBIETTIVI

Ingegnerizzare il processo di screening del consumo delle ruote e di riposizionamento ottimale nella scarpa.

APP FRONT-END

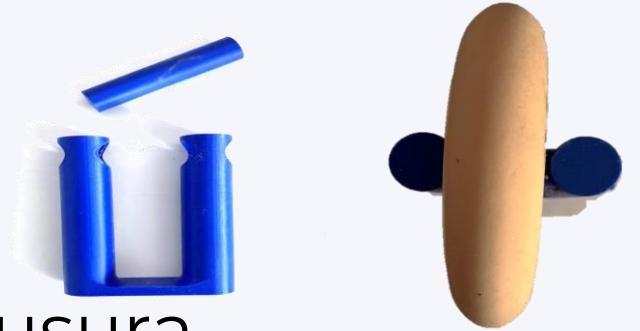
Rilascio in produzione di un'applicazione di assistenza per la manutenzione delle ruote.



Agenda

- 01 Contesto e Obiettivi
- 02 **Dataset**
- 03 Modelli
- 04 Applicazione Web
- 05 Conclusioni

Creazione del dataset



- Raccolta di **248 ruote** con diversi livelli di usura
- Ogni ruota è stata **fotografata** utilizzando un **supporto** realizzato appositamente con stampante 3D
- Specchiato ogni foto per ampliare il data-set arrivando a **476 foto**



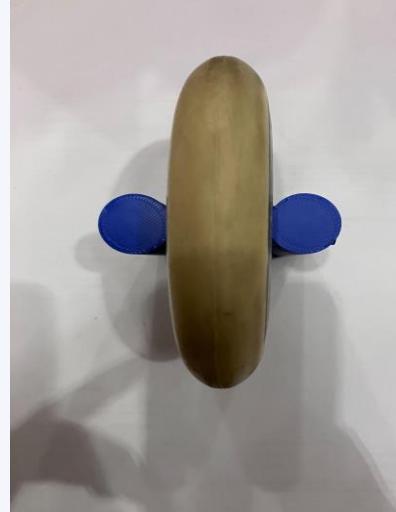
Classificazione del Dataset

Ogni immagine è stata **etichettata manualmente**

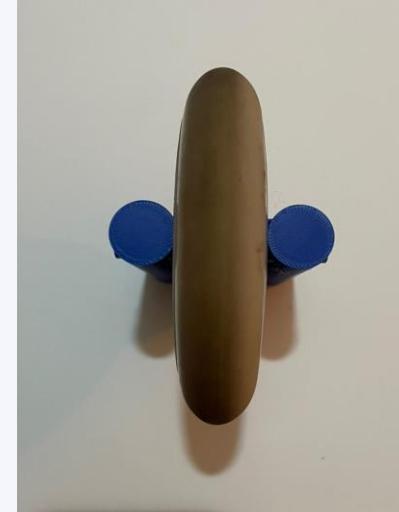
Come? Assegnazione di un valore di usura per ciascun lato della ruota.

Scala valori di usura:

- 0** Ottimo
- 1** Buono
- 2** Scarso
- 3** Pessimo



0,0



2,1

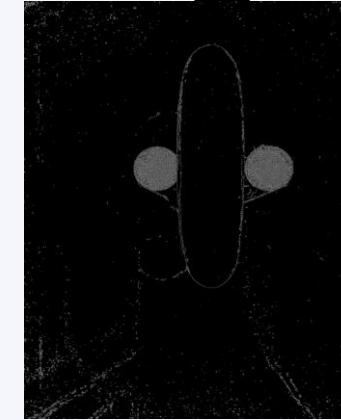


2,3



Pre processing delle immagini

- Aggiunta di tre canali di bordo, fatto con **Canny**
- Normalizzazione dei colori dei tre canali **RGB**
- Raddrizzamento dell'immagine correggendo il metadato **EXIF**
- Normalizzazione delle dimensioni mantenendo le proporzioni originali e tramite **padding**
- Concatenazione dei tre canali RGB con i tre canali di bordo



Agenda

- 01 Contesto e Obiettivi
- 02 Dataset
- 03 **Modelli**
- 04 Applicazione Web
- 05 Conclusioni

Fine tuning dei modelli



python™

- **Primo layer adattato a 6 canali** per accettare RGB + bordo
- **Layer finale con due output** per ottenere un coefficiente di consumo per il lato sx e uno per il lato dx
- **Wrapper del layer finale** con una sigmoide per ottenere valori tra 0 e 1
- **MSE** come loss function
- **Adam** come optimizer del modello
- **Testing** tecniche di **model validation**: K-fold vs holdout



Confronto tra Modelli

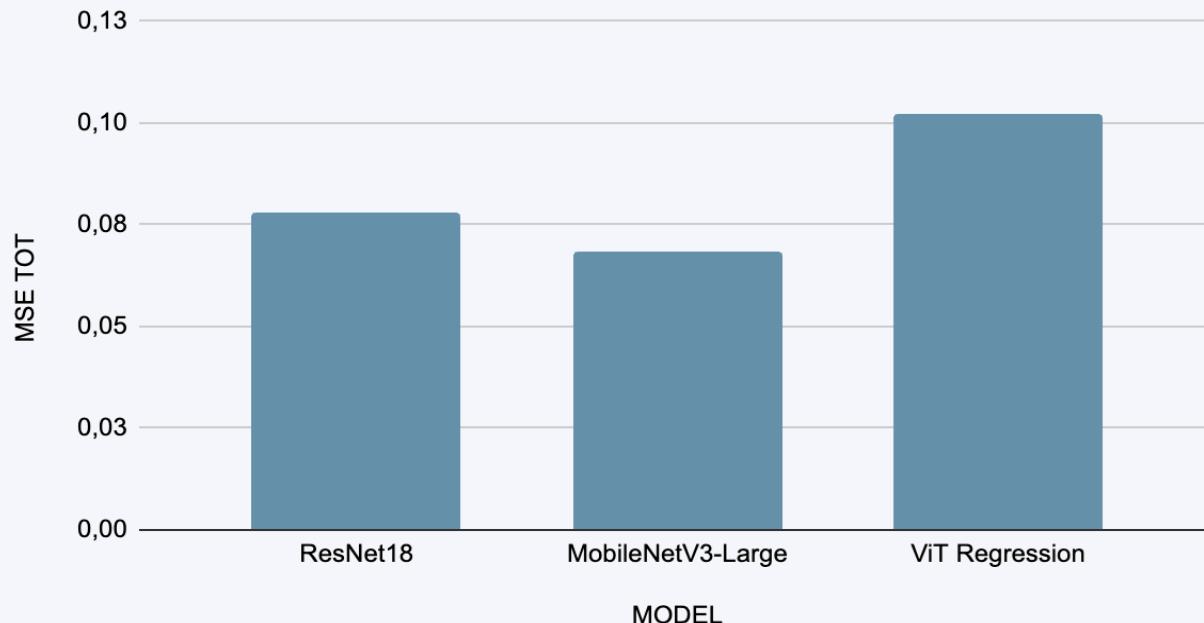
Model	Pro	Contro	Use Case su Bordi
ViT Regression	Ottima comprensione globale della forma; molto flessibile; buona sensibilità ai pattern complessi	Richiede molti dati; computazionalmente pesante; meno performante con immagini semplici	Se i bordi contengono pattern complessi o variazioni globali da catturare
MobileNetV3-Large	Leggero e veloce; buono sui dettagli locali; ottimo per dataset piccoli e deploy	Meno preciso sui pattern globali; può perdere info strutturali	Se serve velocità, efficienza e un modello adatto a immagini semplificate
ResNet18	Robusto; buone performance con dataset medio-piccoli; ottimo sulle feature derivate dai bordi	Meno leggero di MobileNet; meno globale di ViT	Se si vuole un equilibrio stabile e affidabile su bordi/silhouette



Model assessment

MODEL	MSE X	MSE Y	MSE TOT	Numero di predizioni con errore > 0.33	Numero di predizioni con errore > 0.55
ResNet18	0.07819677117628462	0.06409289049789449	0.07114483083708956	52	14
MobileNetV3-Large	0.046223680648390664	0.050178183378282434	0.04820093201333655	31	3
ViT Regression	0.10238996161921807	0.09217103354527717	0.09728049758224762	61	19

MSE TOT rispetto a MODEL



MobileNetV3-Large è il modello che
performa meglio, nato **per elaborare
immagini provenienti da smartphone.**



Agenda

- 01 Contesto e Obiettivi
- 02 Dataset
- 03 Modelli
- 04 **Applicazione Web**
- 05 Conclusioni

Streamlit App

localhost:8502

Retex AI e-Learning - UNIMI... VMware Horizon Colab Posta in arrivo (33) ... Il mio Drive - Googl... Our World in Data DB-Engines Ranking... Tutti i p



Rollers Check-Up

Controlla l'usura delle due ruote e trova l'assalto ottimale per le gare.

Progetto sviluppato da

Andrea De Tomasi
Giovanni Pichieri
Giuliana Chiarella

Versione 1.0 (Streamlit Frontend)

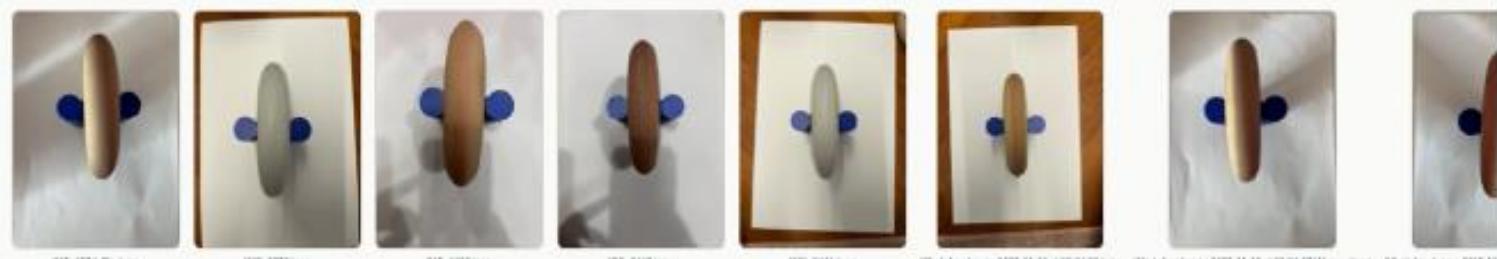
WhatsApp Image 2025-11-18 at 20.04.37 (1).copy.jpeg 1400x1400

WhatsApp Image 2025-11-18 at 20.04.36.jpeg 1400x1400

Stampa page 1 of 1

Hai caricato il rotelle!

Anteprima Ruote Caricate



IMG_3014.jpg
IMG_6216.jpeg
IMG_6800.jpeg
IMG_3440.jpeg
IMG_3441.jpeg
WhatsApp Image 2025-11-18 at 20.04.30.jpeg
WhatsApp Image 2025-11-18 at 20.04.31 (1).copy.jpeg
WhatsApp Image 2025-11-18 at 20.04.31 (1).copy.jpeg

Analisi Ruote

Avvia analisi

Assistente Virtuale – RollersBot



Agenda

- 01 Contesto e Obiettivi
- 02 Dataset
- 03 Modelli
- 04 Applicazione Web
- 05 **Conclusioni**

Conclusioni



LIMITI

- **Dataset piccolo.** Ampliare il dataset per migliorare la generalizzazione del modello, oppure cambiare logica di validazione per il Training
- **Metriche di classificazione soggettive,** evitando analisi “a occhio”, come avvenuto nella fase attuale in cui le label sono state assegnate manualmente
- **Performance** computazionali app



MIGLIORAMENTI

- **Sperimentare modelli alternativi** confrontando architetture e prestazioni
- **Esplorare nuove tecniche di pre-processing** per ottimizzare i dati in input
- **Sviluppare un agente AI integrabile in un'app mobile**, per portare il sistema in un contesto reale di utilizzo
- **Tempi di risposta** del chatbot

28 Novembre 2025

Grazie!

Giovanna Picherri

Andrea De Tomasi

Gianluca Chiarello

