28 Novembre 2025

CNN-WheelClassifier

AI model for the analysis of skating wheels: wear assessment and optimization of their positioning.

Team: I rotellisti

Giovanna Pichierri

Andrea De Tomasi

Gianluca Chiarello

Agenda


Gara Finale 500m Sprint Junior F/M
Guarda più...
Condividi
500 m Sprint JUNIORES F - FINALE

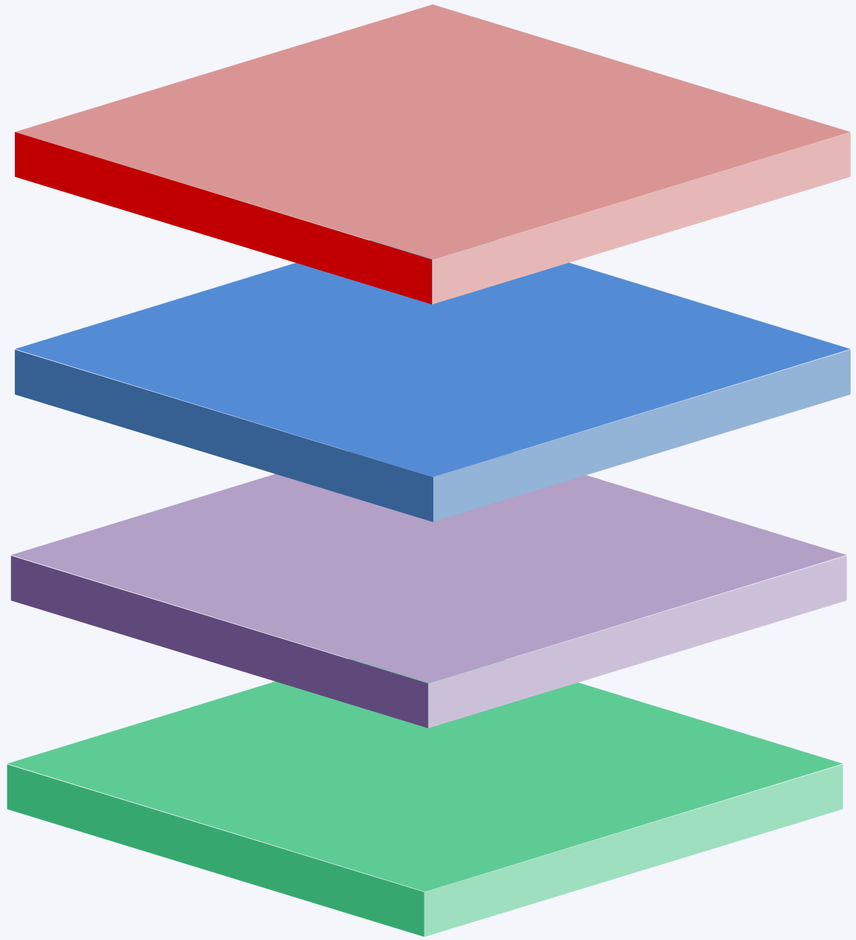
Project Objectives

PROBLEM

Skate wheels are subject to wear and tear. To optimize their use, skaters visually inspect the wheels and reposition them.

AI SOLUTION DEVELOPMENT

Computer vision techniques applied with CNN in Python



OBJECTIVES

Engineering the process of screening wheel wear and optimal repositioning in the shoe.

FRONT-END APP

Release into production of a wheel maintenance assistance application.

Agenda

01 Context and Objectives

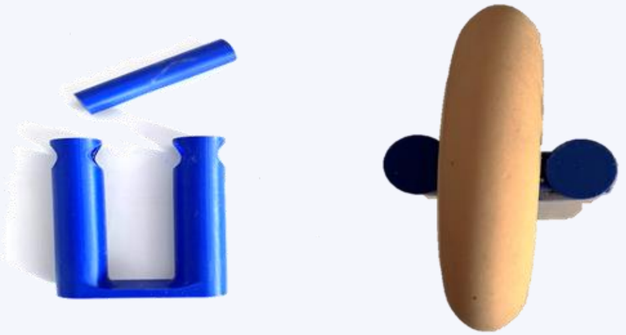
02 **Dataset**

03 Models

04 Web Application

05 Conclusion

Creation of the dataset



- Collection of **248 wheels** with varying degrees of wear

- Each wheel was **photographed** using a specially made stand created with a 3D printer.

- Mirrored each photo to expand the data set to **476 photos**

Dataset Classification

Each image was **labelled manually**

How? By assigning a wear value to each side of the wheel.

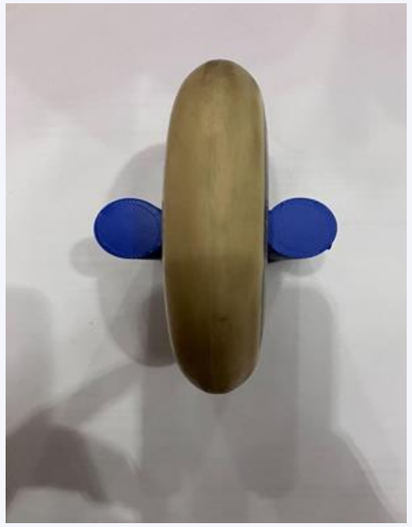
Wear rating scale:

0 Great

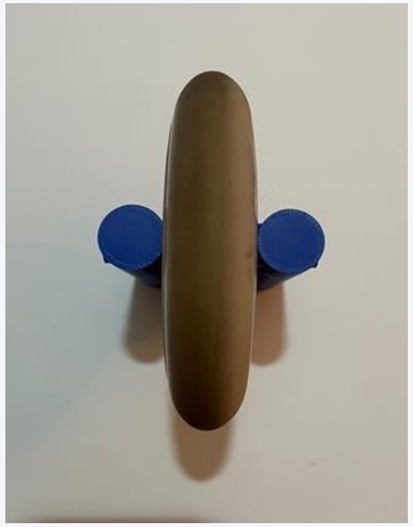
1 Good

2 Poor

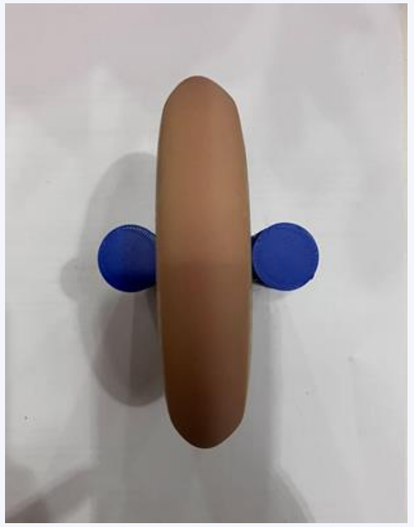
3 Terrible



0,0

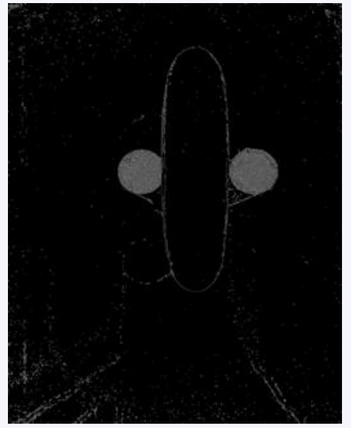


2,1



2,3

Image pre-processing



- Addition of three edge channels, done with **Canny**

- Normalisation of the colours of the three **RGB channels**

- Straightening of the image by correcting the **EXIF** metadata

- Normalisation of the dimensions while maintaining the original proportions and using **padding**

- Concatenation of the three RGB channels with the three edge channels

Agenda

01 Context and Objectives

02 Dataset

03 **Models**

04 Web Application

05 Conclusion

Fine tuning of models

- **First layer** adapted to 6 channels to accept RGB + edge

- **Final layer** with two outputs to obtain a consumption coefficient for the left side and one for the right side

- Final layer wrapper with a **sigmoid** to obtain values between 0 and 1

- **MSE** as loss function

- **Adam** as model optimiser

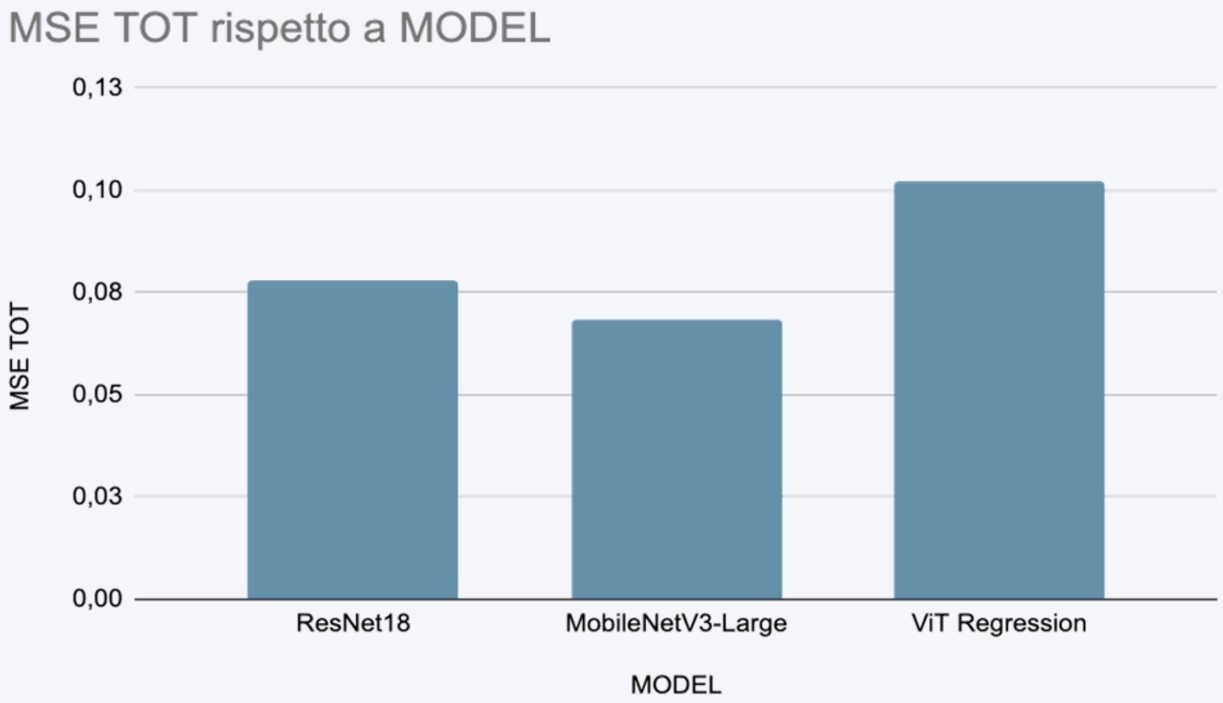
- **Testing** model validation techniques: K-fold vs holdout

Models Comparison

Model	Pro	Contro	Use Case on Edges
ViT Regression	Excellent overall understanding of form; very flexible; good sensitivity to complex patterns	Requires a lot of data; computationally intensive; less effective with simple images	If the edges contain complex patterns or global variations to capture
MobileNetV3-Large	Lightweight and fast; good on local details; great for small datasets and deployments	Less accurate on global patterns; may lose structural information	If you need speed, efficiency and a model suitable for simplified images
ResNet18	Robust; good performance with small to medium datasets; excellent on edge-derived features	Less lightweight than MobileNet; less comprehensive than ViT	If you want stable and reliable balance on edges/silhouettes

Model assessment

MODEL	MSE X	MSE Y	MSE TOT	Number of predictions with error > 0.33	Number of predictions with error > 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



MobileNetV3-Large is the best-performing model, designed to process images from smartphones.

Agenda

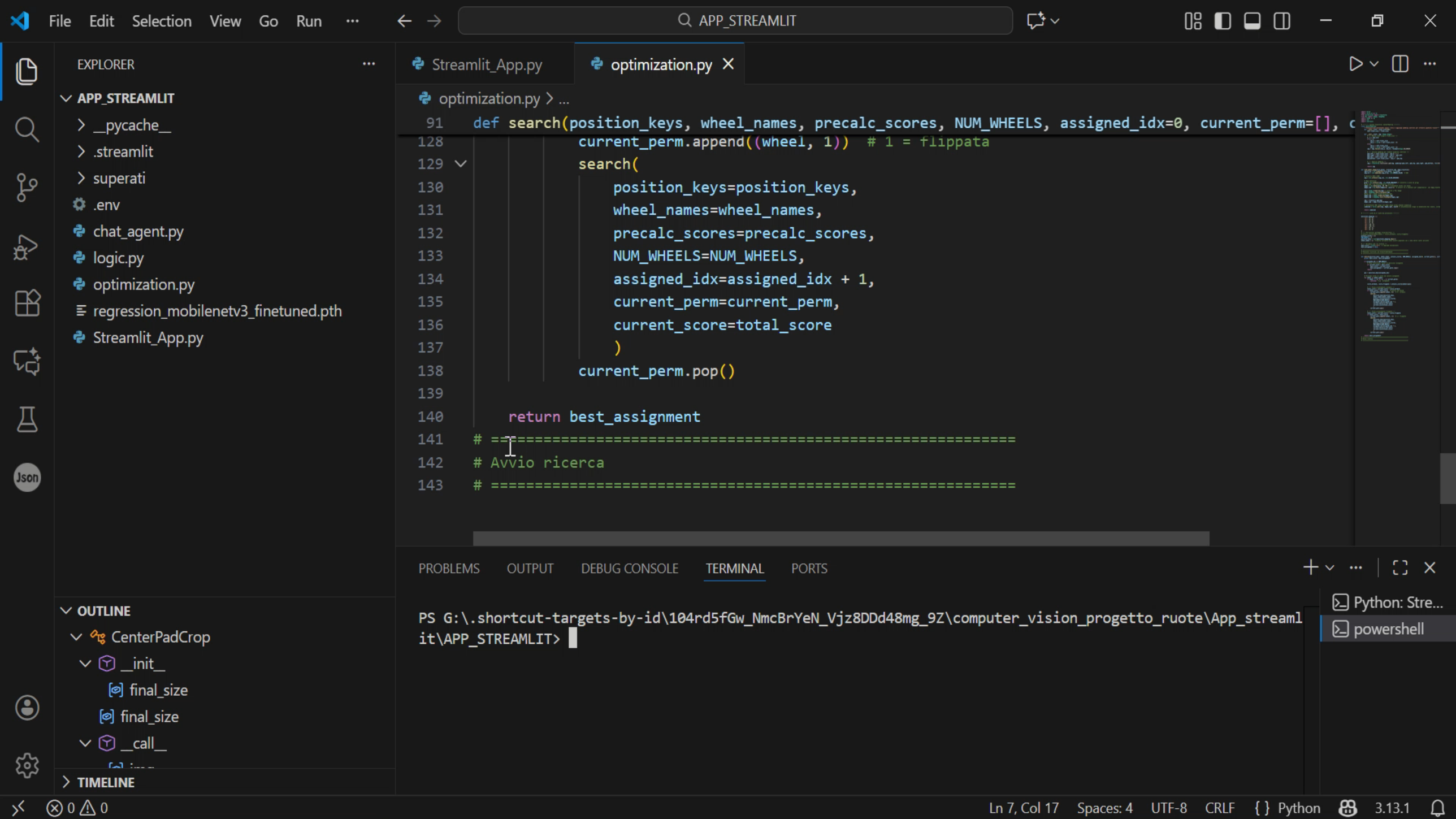
01 Context and Objectives

02 Dataset

03 Models

04 **Web Application**

05 Conclusion


File Edit Selection View Go Run
APP_STREAMLIT
EXPLORER
APP_STREAMLIT
__pycache__
.streamlit
superati
.env
chat_agent.py
logic.py
optimization.py
regression_mobilenetv3_finetuned.pth
Streamlit_App.py
Streamlit_App.py
optimization.py
optimization.py > ...
91 def search(position_keys, wheel_names, precalc_scores, NUM_WHEELS, assigned_idx=0, current_perm=[], c
128 current_perm.append((wheel, 1)) # 1 = flippata
129 search(
130 position_keys=position_keys,
131 wheel_names=wheel_names,
132 precalc_scores=precalc_scores,
133 NUM_WHEELS=NUM_WHEELS,
134 assigned_idx=assigned_idx + 1,
135 current_perm=current_perm,
136 current_score=total_score
137)
138 current_perm.pop()
139
140 return best_assignment
141 # ==
142 # Avvio ricerca
143 # ==
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS G:\.shortcut-targets-by-id\104rd5fGw_NmcBrYeN_Vjz8DDd48mg_9Z\computer_vision_progetto_ruote\App_streaml
it\APP_STREAMLIT>
Python: Stre...
powershell
OUTLINE
CenterPadCrop
__init__
final_size
final_size
__call__
TIMELINE
Ln 7, Col 17 Spaces: 4 UTF-8 CRLF { } Python 3.13.1

Agenda

01 Context and Objectives

02 Dataset

03 Models

04 Web Application

05 **Conclusion**

Conclusion

LIMITS

- **Small dataset.** Expand the dataset to improve model generalization, or change the validation logic for Training.
- **Subjective classification metrics.** Avoiding ‘eyeballing’ analyses, as has been the case in the current phase where labels have been assigned manually.
- **Computational performance app**

IMPROVEMENTS

- **Sperimentare modelli alternativi** confrontando architetture e prestazioni
- **Exploring new pre-processing** techniques to optimise input data
- **Develop an AI agent** that can be integrated into a mobile app, to bring the system into a real-world context of use.
- **Chatbot** response times

28 Novembre 2025

Thank you!

Giovanna Pichierri

Andrea De Tomasi

Gianluca Chiarello