

PROGETTO CORSO SISTEMI DIGITALI M

Loris Giannatempo – Davide Guidetti

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

OBIETTIVI DEL PROGETTO

- Sviluppo di una **app** Android per **rilevare** la presenza o meno di **mascherine sui volti**
- Si farà uso di una rete neurale in grado di effettuare **Object Detection**.
- **Object Tracking** dei volti al fine di realizzare un **conta persone**:
 - conteggio individui entrati con mascherina
 - conteggio individui entrati senza mascherina

DATASET

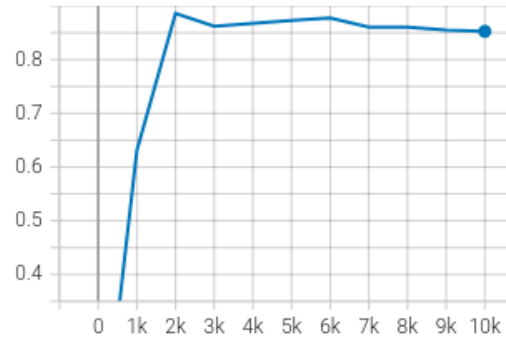
- Dataset disponibile su [Kaggle](#)
- Circa **4000 immagini** annotate
- box appartenenti a **20 label** diverse, organizzate in una gerarchia
- Le label sono state **rimappate** in 2 macro classi:
 - **Con mascherina**
 - **Senza mascherina**



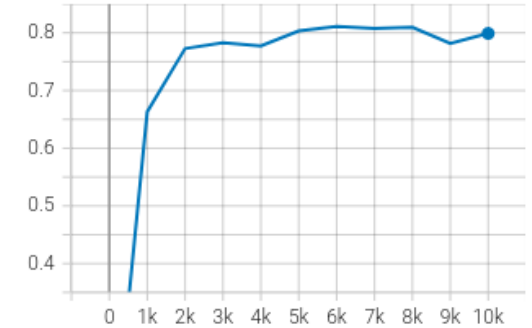
ADDESTRAMENTO

- Uso di **Tensorflow Object detection API**
- Pretrained Model scelto: **SSD MobileNet V2 FPNLite 320x320**
- Durante il training il modello viene testato usando le **COCO API**
- Test successivi hanno evidenziato che il modello con 5000 steps di training risulta migliore di quello con 10000 steps

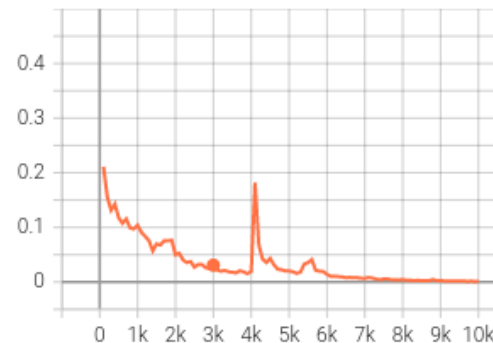
DetectionBoxes_Precision/mAP@.50IOU
tag: DetectionBoxes_Precision/mAP@.50IOU



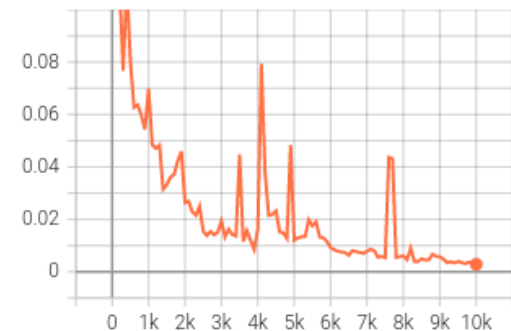
DetectionBoxes_Recall/AR@100 (large)
tag: DetectionBoxes_Recall/AR@100 (large)



Loss/classification_loss
tag: Loss/classification_loss



Loss/localization_loss
tag: Loss/localization_loss



ESTRAZIONE e CONVERSIONE MODELLO

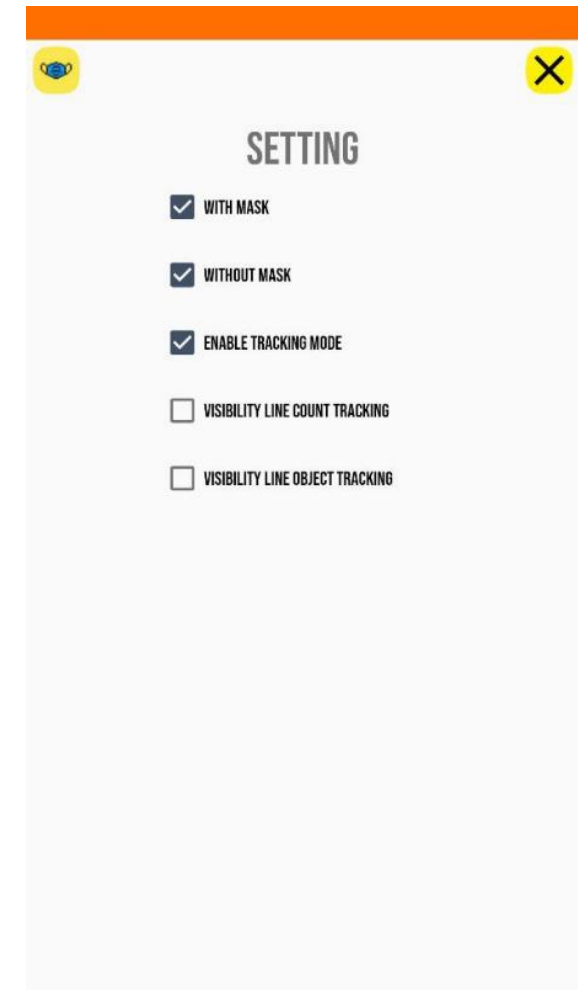
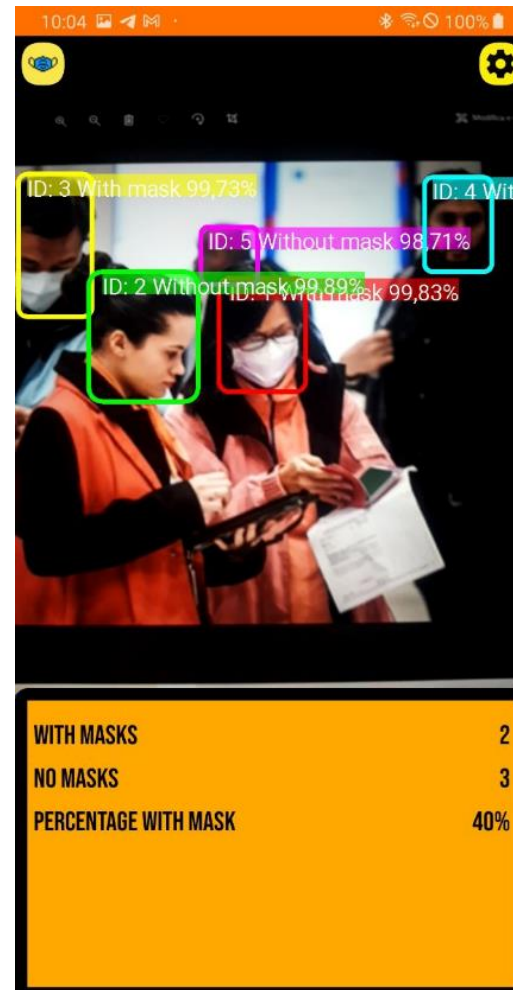
- Salvato un **checkpoint** ogni 1000 steps.
- Dai checkpoint si estrarre il modello in formato **SavedModel**, che include il file **.pb**
- Modello convertito in **.TFLite** per uso su mobile
- Vengono aggiunti **metadati** utilizzati dalla **Task Library ObjectDetector API**
- Full integer quantization porta a risultati troppo inaccurati

```
69 #tensorflow lite conversion
70 converter = tf.lite.TFLiteConverter.from_saved_model(_SAVED_MODEL_PATH)
71 converter.optimizations = [tf.lite.Optimize.DEFAULT]
72 converter.allow_custom_ops = True
73 converter.experimental_new_converter = True
74 converter.target_spec.supported_ops = [
75     tf.lite.OpsSet.TFLITE_BUILTINS_INT8, # Ensure that if any ops can't be c
76     tf.lite.OpsSet.TFLITE_BUILTINS, # enable TensorFlow Lite ops.
77     tf.lite.OpsSet.SELECT_TF_OPS # enable TensorFlow ops.
78 ]
79 converter.representative_dataset = representative_data_gen
80 tflite_model = converter.convert()
81 with open(_TFLITE_MODEL_PATH, 'wb') as f:
82     f.write(tflite_model)
```

```
104 from tflite_support.metadata_writers import object_detector
105 from tflite_support.metadata_writers import writer_utils
106 writer = object_detector.MetadataWriter.create_for_inference(
107     writer_utils.load_file(_TFLITE_MODEL_PATH), input_norm_mean=[127.5],
108     input_norm_std=[127.5], label_file_paths=[_TFLITE_LABEL_PATH])
109 writer_utils.save_file(writer.populate(), _TFLITE_MODEL_WITH_METADATA_PATH)
```

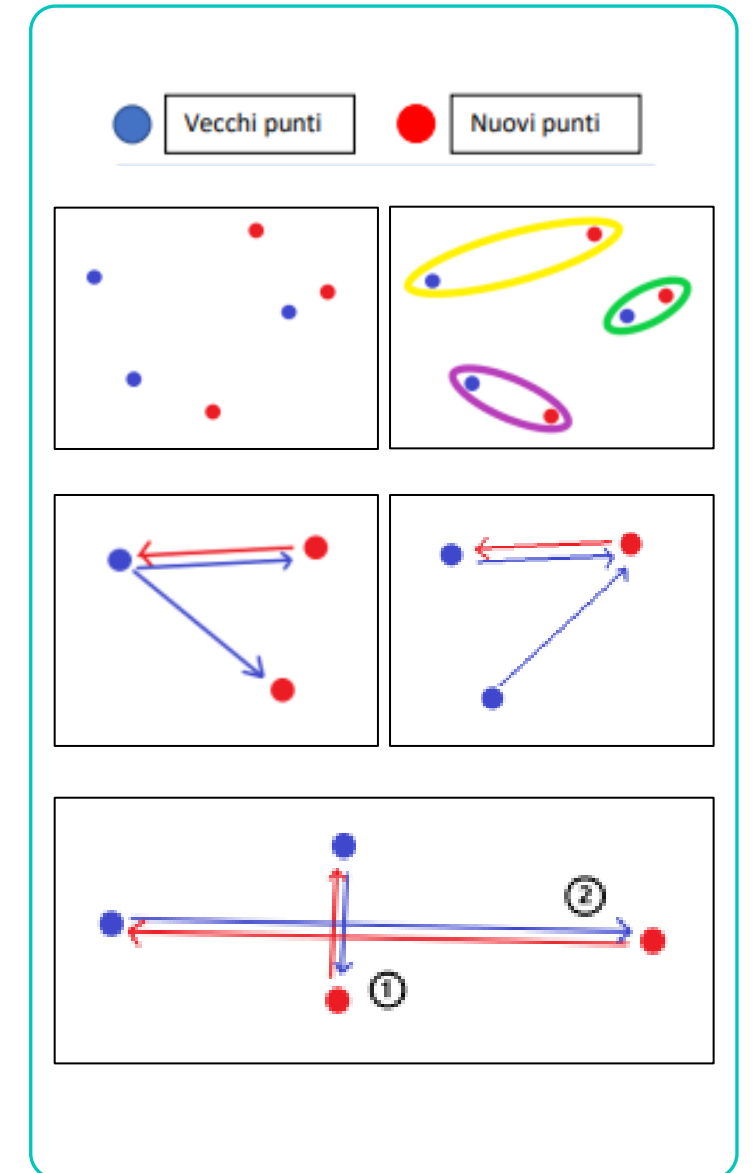
ANDROID

- Sviluppo dell'app Android a partire da un esempio offerto da Tensorflow
- L'applicazione è stata estesa, e si compone di due activity:
 - ❑ Camera Activity
 - ❑ Settings Activity



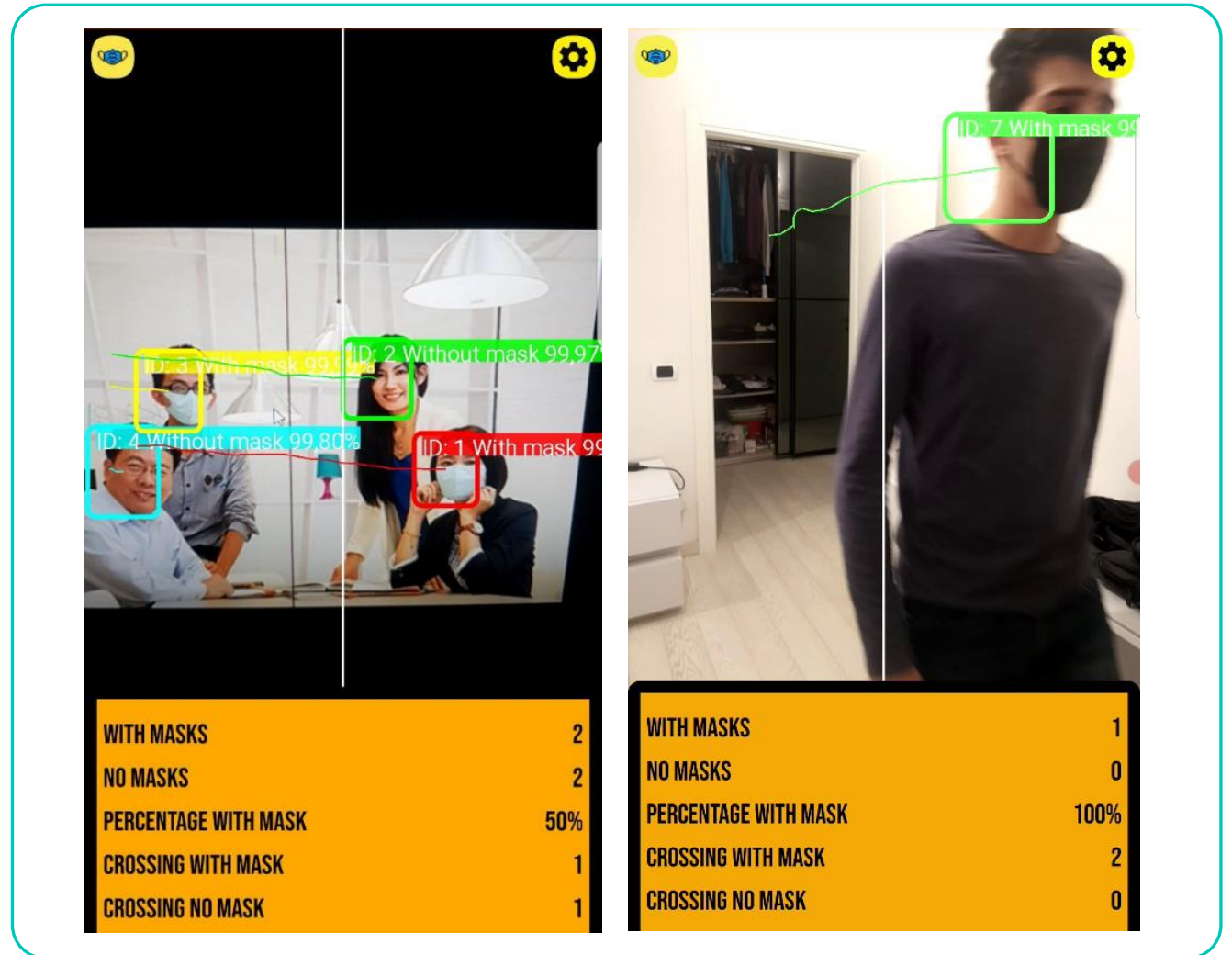
TRACKING

- Ogni **box** viene **trasformata in un punto**
- Tracciamento degli oggetti si **basa**, volta per volta, sulla **posizione dei punti** nei nuovi frame
- Algoritmo **associa** ad ogni nuovo punto il vecchio punto **reciprocamente più vicino**
- Collegamento tra punti associabili può richiedere **più passi**
- Vecchi punti non associati mantenuti in memoria per un certo tempo



CONTA PERSONE

- Un algoritmo verifica se un punto supera o meno una linea
- Si **aggiorna il contatore corrispondente alla classe corrente** dell'oggetto che supera la linea



CONCLUSIONI

- I risultati ottenuti riteniamo essere soddisfacenti
- La rete neurale è sufficientemente affidabile, ma risultati migliori potrebbero essere raggiunti con un dataset più ampio e di migliore qualità
- L'app è fluida e di facile utilizzo
- La funzionalità «Conta persone» risulta essere piuttosto efficace