

# FaceMask Detection

---

## Intro

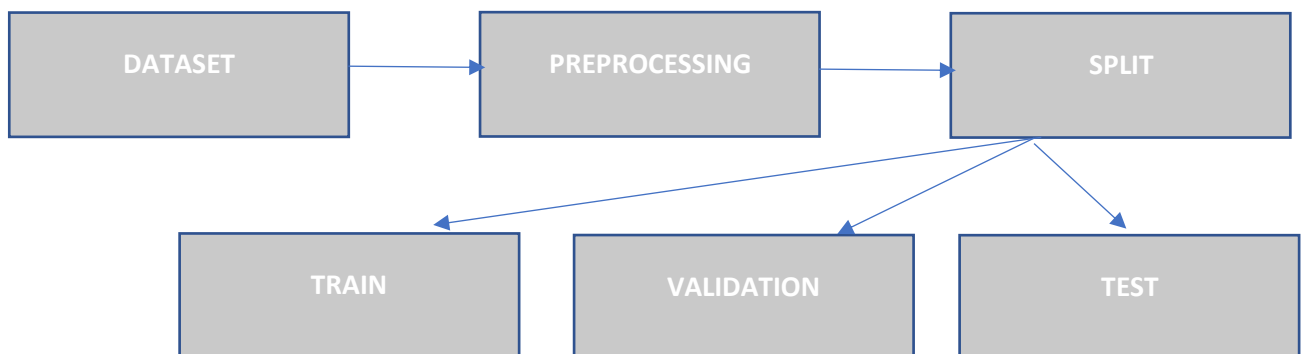
Il primo task riguardo il progetto di Sistemi ad agenti che stiamo portando avanti, si basa sul riconoscimento della mascherina. In questo periodo di forte crisi dovuto alla presenza del COVID-19, abbiamo pensato, insieme alla prof De Carolis, di realizzare un modello che potesse essere utile nel riconoscimento della mascherina sul volto della gente, sia in Real time, che “staticamente” (analizzando la singola immagine).

## Idee per il progetto

L'idea che abbiamo utilizzato per portare a termine tale task, è stato quello di utilizzare una delle tecniche più conosciute nell'ambito dell'AI, l'uso di una Neural Network. Il primo step è stato quello di approfondire tale tematica, in maniera piuttosto teorica, aiutandoci sia con l'utilizzo di libri, articoli presenti sul web e consultando la documentazione delle diverse librerie che ci sarebbero servite.

## Organizzazione del progetto

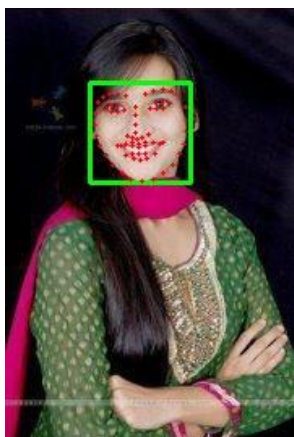
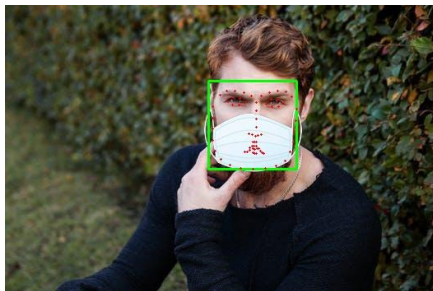
### 1.FASE DI TRAIN



- **Dataset :**  
[https://drive.google.com/drive/folders/1JtzDcqpgAuh15PMcoc6UHQEgT\\_SEMT8J?usp=sharing](https://drive.google.com/drive/folders/1JtzDcqpgAuh15PMcoc6UHQEgT_SEMT8J?usp=sharing)

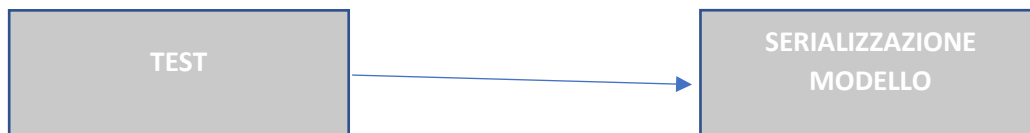


- **Preprocessing** : tale fase si è basata sull'processing delle immagini presenti nel dataset. Esso è utile farlo, al fine di addestrare la rete esclusivamente sulla parte dell' immagine utile a noi (il viso). Per fare ciò abbiamo seguito una metodologia consigliataci dal Dr. Nicola Macchiarulo; tale metodologia consiste nel:
  - Individuare il volto attraverso l'uso di un **Face Detector** (abbiamo utilizzato la libreria *dlib*)
  - Una volta individuato il volto, abbiamo effettuato il *crop* esclusivamente sulla zona individuata dal precedente modello
  - Questo ci ha portato a creare un nuovo dataset composto da due sottocartelle (*cropped\_with\_mask*, *cropped\_without\_mask*) : [https://drive.google.com/drive/folders/1JtzDcqpgAuh15PMcoc6UHQEgT\\_SEMT8J](https://drive.google.com/drive/folders/1JtzDcqpgAuh15PMcoc6UHQEgT_SEMT8J)
  - A causa della presenza di immagine duplicate all'interno del dataset, abbiamo utilizzato un programma: "Similar Images". Questo ha fatto sì che il dataset diventasse sbilanciato, ( $\text{cropped\_with\_mask} < \text{cropped\_without\_mask}$ ). Per risolvere tale problema, sono state aggiunte altre immagini, opportunamente *croppate*, nella sottocartella di nostro interesse (*cropped\_with\_mask*).



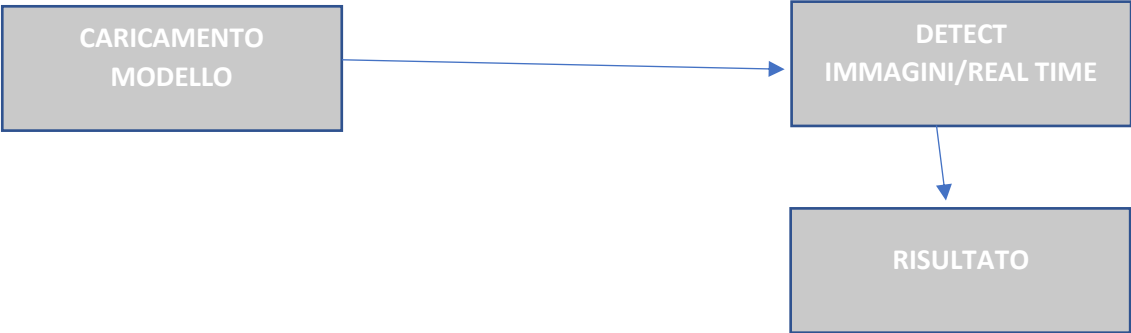
- **Split** : al fine di addestrare la rete (MobileNetV2), abbiamo suddiviso il dataset in
  - train → 80%
  - validation → 10%
  - test → 10 %
- **Train** : abbiamo effettuato il training basandoci su N\_epochs (numero di epoche)= 20. Abbiamo fatto fine tuning della rete menzionata precedentemente

## 2.FASE DI TEST



- Dopo aver effettuato il test, abbiamo definito l'accuratezza nella predizione. I valori sono riportati in basso nella sezione: "metriche utilizzate". Ci siamo serviti anche di una Confusion matrix (presente in basso) utile a valutare gli errori commessi e non nella fase di predizione. Infine il modello viene serializzato sottoforma di "*file.h5*"

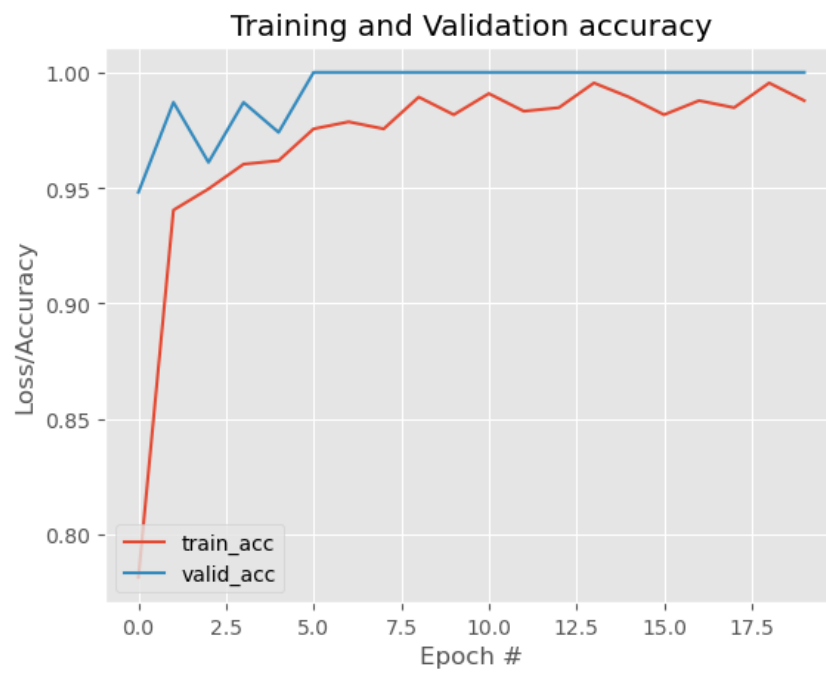
**3.FASE APPLICATIVA**



- Al termine del tutto ci siamo concentrati sull'utilizzo del modello creato, affinché riconoscesse la presenza o meno della mascherina da immagini esterne scelte su internet, per poi effettuare un detect in tempo reale (mediante l'utilizzo della webcam del pc) sfruttando le funzionalità della libreria *OpenCV*.
- Intefaccia Grafica → al fine di utilizzare tale progetto in maniera molto più rapida ed intuitiva, abbiamo considerato la creazione di una semplice interfaccia grafica.



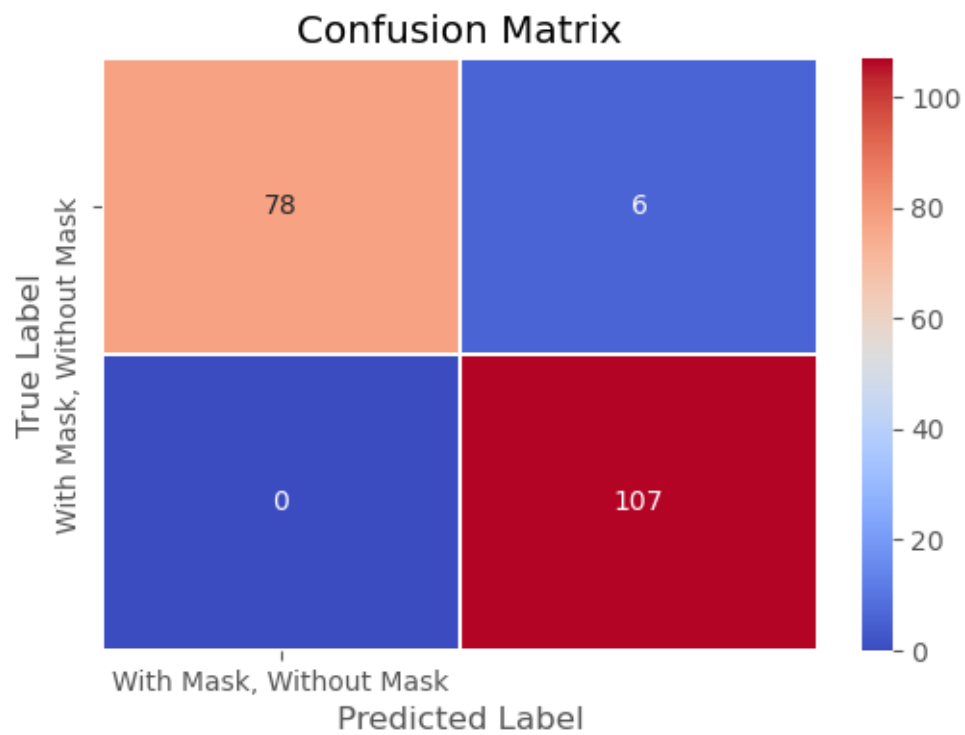
## Training & Validation Accuracy



## Training & Validation Loss



## Confusion Matrix



## Metriche utilizzate

|                      | Precision | Recall | f1-score | Support |
|----------------------|-----------|--------|----------|---------|
| cropped_with_mask    | 0.99      | 0.93   | 0.96     | 84      |
| cropped_without_mask | 0.95      | 0.99   | 0.97     | 107     |
| accuracy             |           |        | 0.96     | 191     |
| macro avg            | 0.97      | 0.96   | 0.96     | 191     |
| weighted avg         | 0.96      | 0.96   | 0.96     | 191     |