



Covid Protection Checker

Cristian Davide Conte

0001034932

Simone Morelli

0001034742

Punti principali

01

Reti Neurali

02

Architettura Applicazione

03

Problemi Affrontati

04

Risultati

Reti Neurali

1 Dataset utilizzato

Dataset WWMR-DB composto da **1200 foto** divise in **8 classi** appartenenti a **42 soggetti**.

3 Training e Risultati

Allenamento di **50 epoche** per ciascuna rete neurale.
Precisioni ottenute: **61% e 81%**.

2 Architettura Reti Neurali

Sono state usate **2 architetture** diverse per due funzionalità diverse:
EfficientDet - Lite 0 ed
EfficientDet - Lite 4

Dataset

8 modi di indossare la mascherina

- indossata correttamente
- non indossata
- pendente da un'orecchio
- sulla fronte
- sulla punta del naso
- sotto al naso
- sopra al mento
- sotto al mento

3 inclinazioni dei volti

- visione frontale - 0°
- visione inclinata - 45°
- visione di profilo - 90°

4 tipi di mascherina

- mascherina chirurgica
- mascherina non medica
- respiratore senza valvola
- respiratore con valvola

Architettura Reti Neurali

EfficientDet - Lite 0

- Basata su EfficientNet
- Lavora con immagini a risoluzione **320x320_{px}**
- **Maggiore velocità** di esecuzione
- **Minore precisione** sulle predizioni
- Utilizzata per la **live detection**
- 30 frame per secondo

EfficientDet - Lite 4

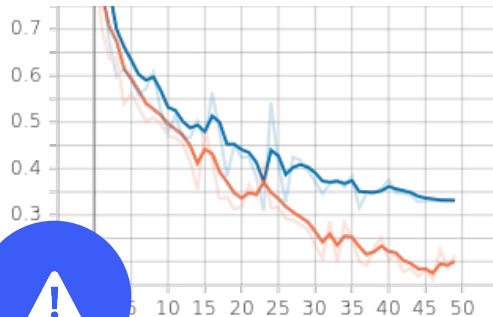
- Basata su EfficientNet
- Lavora con immagini a risoluzione **640x640_{px}**
- **Minore velocità** di esecuzione
- **Maggiore precisione** sulle predizioni
- Utilizzata per l'**analisi in differita**
- 4 frame per secondo

Training e Risultati

Dopo un allenamento di **50 epoche** per entrambe le reti neurali abbiamo raggiunto delle **precisioni medie** del:

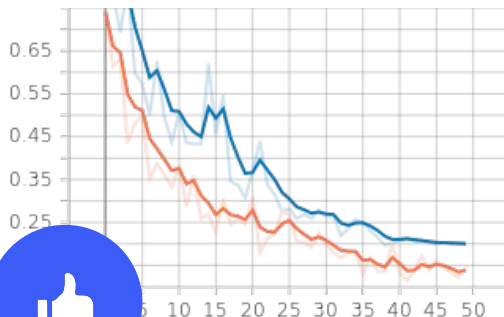
- **61%** -> EfficientDet - Lite 0
- **66%** -> EfficientDet - Lite 4

Il secondo risultato ottenuto **non rispecchiava però ciò che ci aspettavamo.**



Overfitting

Durante l'allenamento della rete **EfficientDet-Lite 4** abbiamo notato che l'**accuracy** sul **training set aumentava** mentre quella sul **validation set stagnava**.



No overfitting

Abbiamo quindi deciso di fare **data augmentation** sul dataset WWMR-DB di partenza.

L'accuracy finale è dell'**81%**.

Architettura Applicazione

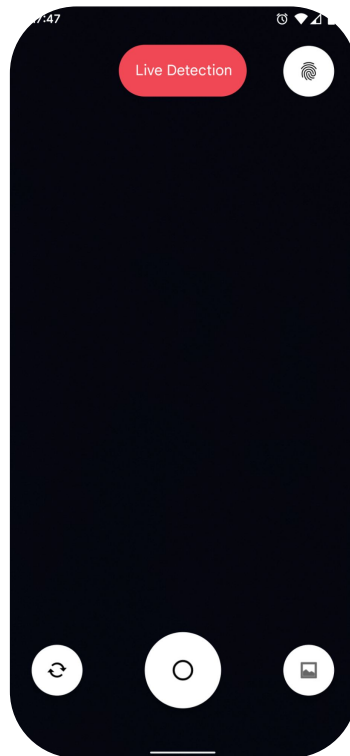
Di cosa si tratta?

È un'applicazione **Android** basata su **modello asincrono a callback** che sfrutta le API **CameraX** e **MediaStore**.

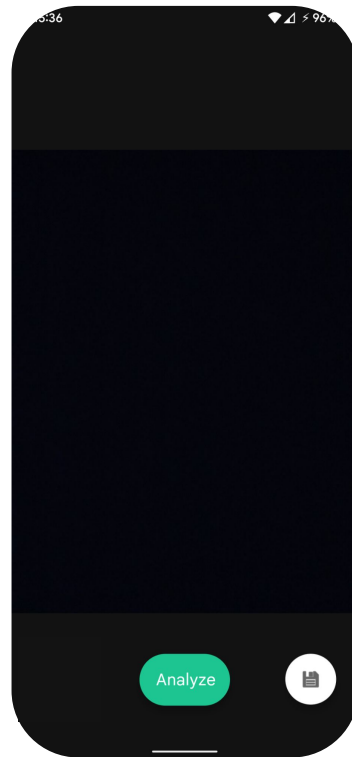
Tre classi principali:

- Una classe **MainActivity** per gestire i **permessi** e lo **startup**
- Una classe **CameraProvider** per gestire la **camera** e le funzionalità di **live detection** tramite la **CNN EfficientDet - Lite 0**
- Una classe **AnalyzeActivity** per le funzionalità di **analisi avanzata** tramite la **CNN EfficientDet - Lite 4**

MainActivity



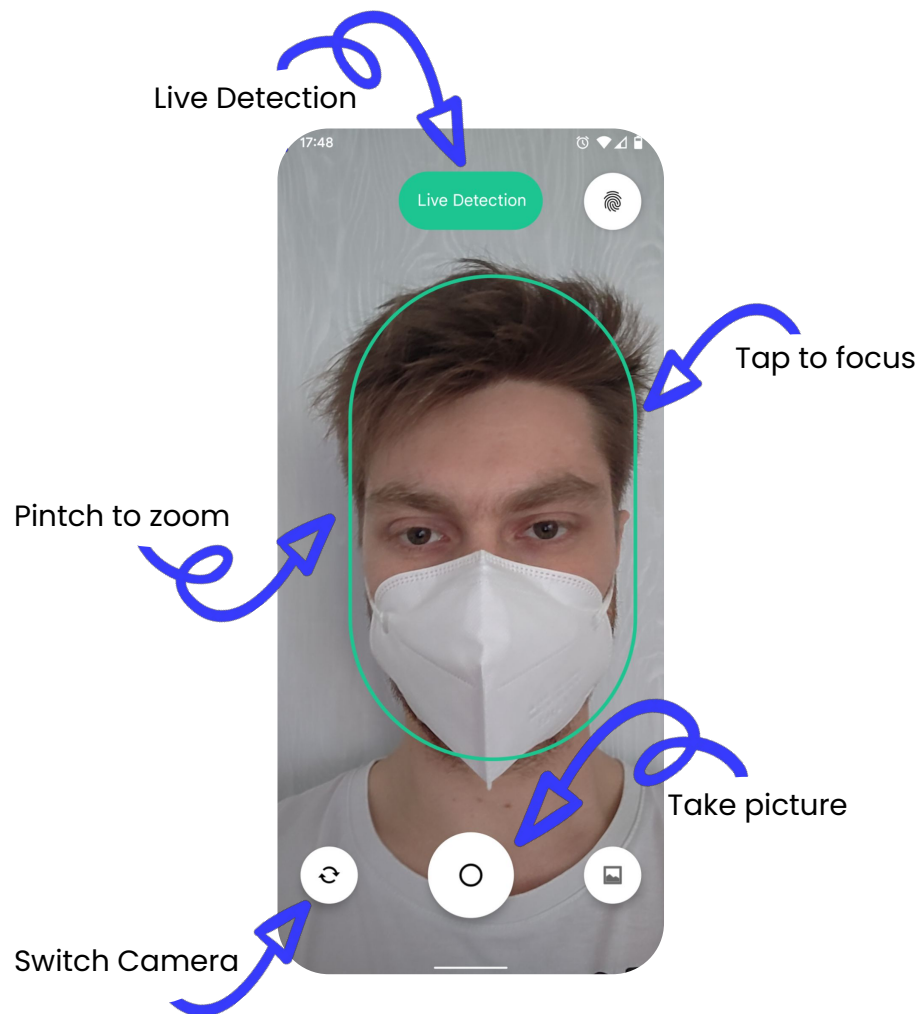
AnalyzeActivity



API CameraX

CameraProvider utilizza gli usecase **CameraX** per implementare le funzionalità principali utilizzate da **MainActivity**:

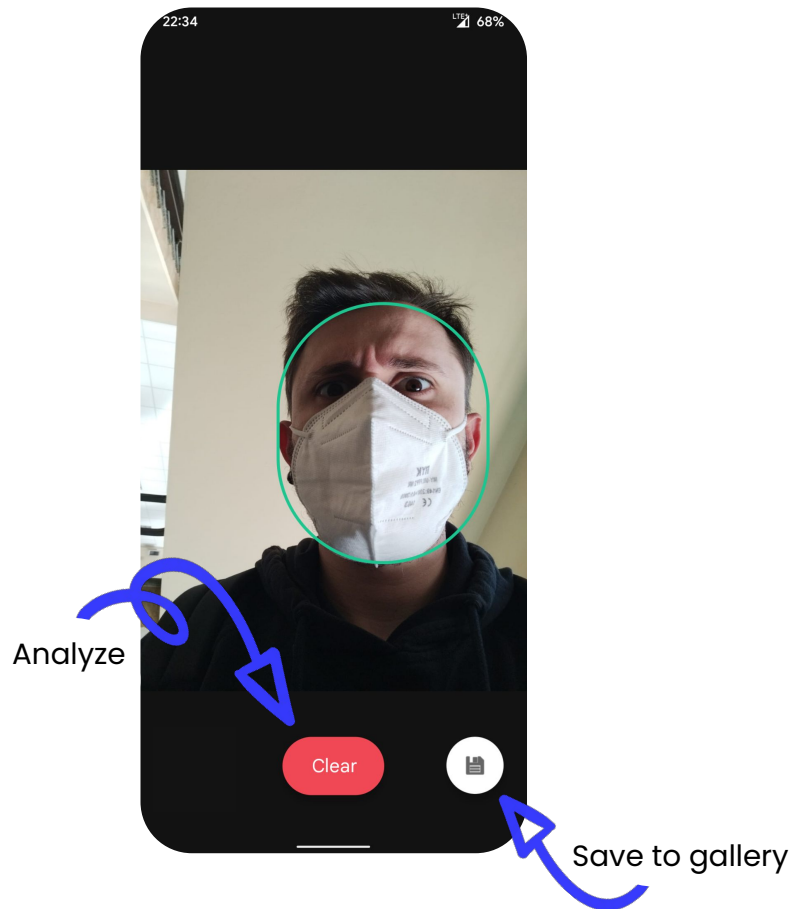
- **ImagePreview** permette lo **streaming** dei frame, la **selezione della camera**, il **pinch-to-zoom** ed il **tap-to-focus**
- **Image Capture** permette di **acquisire i frame** e di lanciare istanze di **AnalyzeActivity**
- **Image Analysis** permette di sfruttare la rete EfficientDet - Lite 0 per la **live detection** delle mascherine



API MediaStore

AnalyzeActivity utilizza il frame acquisito da **CameraProvider** per:

- Eseguire **un'analisi accurata** tramite la rete **EfficientDet -Lite 4**
- Salvarlo nella galleria del telefono tramite l'API MediaStore



Sfide Affrontate

Rotazione della fotocamera

- Ogni **smartphone** moderno in commercio dispone di un **sistema di camere**
- **L'orientamento** di ciascuna camera **non è standard** e varia da dispositivo a dispositivo
- **Algoritmo custom** basato sulle informazioni fornite da **CameraX**

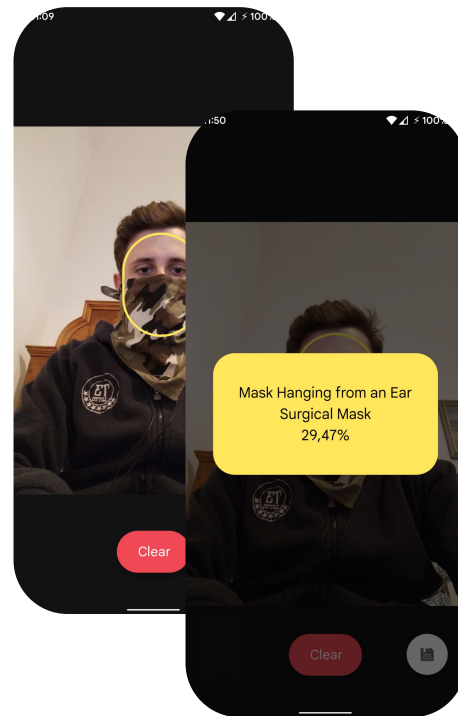
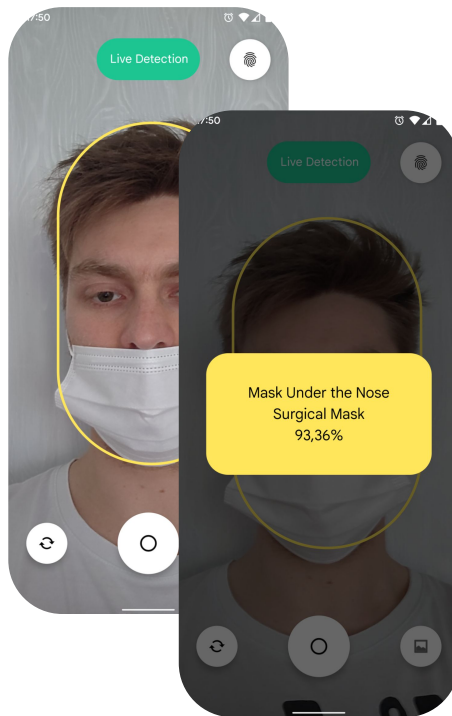
Data augmentation

- Atto a **risolvere l'overfitting** della rete EfficientDet - Lite 4
- **Tool custom** in python3 basato su **ereditarietà** e **pattern adapter**
- Sfrutta operazioni di **convoluzione** sulle immagini originali e accelera i calcoli con la **GPU** tramite l'utilizzo della libreria **CuPy**

Risultati

Risultati coerenti con ciò che ci aspettavamo.

Introducendo variabili non presenti in fase di training (indumenti particolari, oggetti nel frame, monili indossati, ecc...), abbiamo notato delle **variazioni più o meno significative**.





Estensione

Metodo di risoluzione



Assunzioni base

Si lavora solo con volti umani i quali rispettano proporzioni note che scalano uniformemente con la distanza dall'osservatore



Depth Map

Viene sfruttata la rete neurale MiDas per ottenere le depth map dei frames e stimare la gerarchia delle profondità



Trigonometria

Le persone in un frame sono equiparate a punti nello spazio su cui operare operazioni trigonometriche

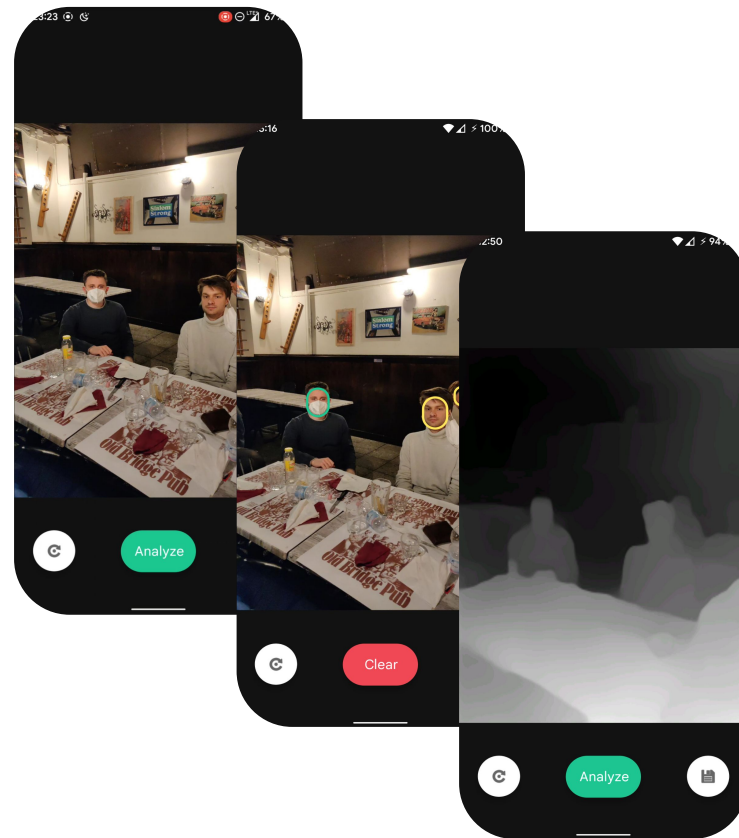
Passo 1

Si acquisisce un'immagine tramite le funzionalità dell'applicazione base.

Si utilizza la rete neurale **EfficientDet - Lite 4** per individuare i **volti** delle persone.

Si utilizza la rete neurale **MiDas** di Intel per creare una **depth map** del frame acquisito.

Si **calcola** la **profondità media**, all'interno della depth map, **dei volti** rilevati.



Passo 2

Si prende il **volto** che dalle misurazioni precedenti è risultato il **più lontano**.

Si fa la **proporzione** tra le sue misure e quelle di un **volto standard** si e si ottiene la **distanza tra l'osservatore e quella persona**.

Sfruttando i valori delle distanze relative contenute nella **depth map** ottenuta al passo 1, si ottiene anche la **distanza tra l'osservatore e la persona più vicina**.



Volto Standard

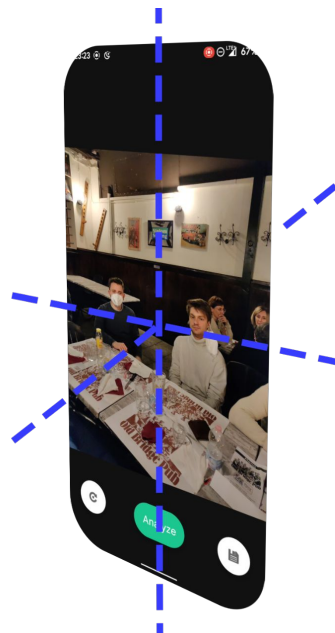
1.5x Volto Standard

Passo 3

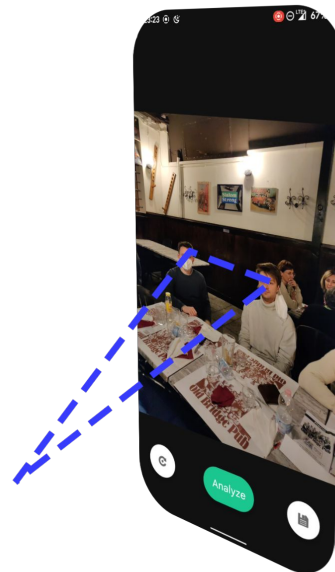
Immaginando un piano passante per il centro dello schermo si calcolano le **coordinate** (in px) **dei volti** su quel piano.

Queste coordinate sono affette dalla **distorsione** data dalla profondità.

Le distorsioni vengono corrette grazie alle distanze volto-osservatore calcolate al passo 2.



Piano immaginario

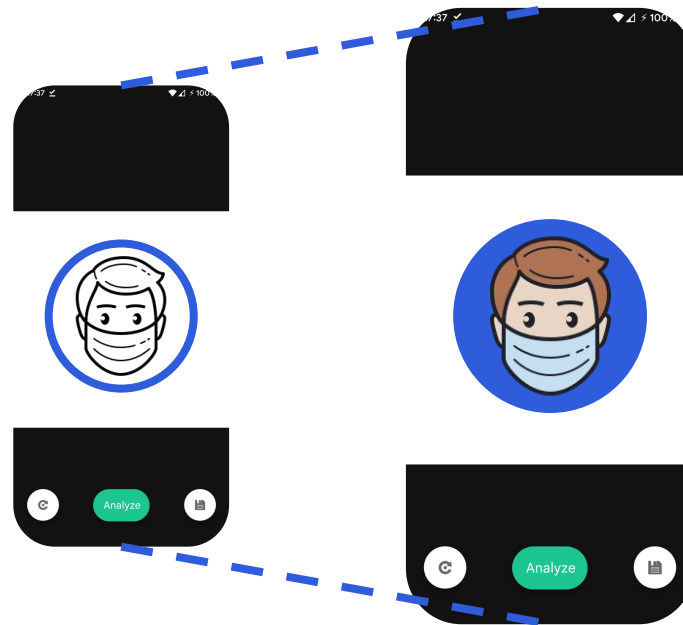


Distanze
volto-osservatore

Passo 4

Si prende la **risoluzione** della foto **corrente** e la si mette in **proporzione** con quella della foto contenente il volto **standard** del passo 2, scalando di conseguenza le coordinate ottenute al passo 3.

A questo punto si usa la formula della **distanza tra due punti** e si valuta se questa è maggiore di 1m per capire se le due persone rispettano o meno la distanza di sicurezza.



Risoluzione standard:
1728x2304px

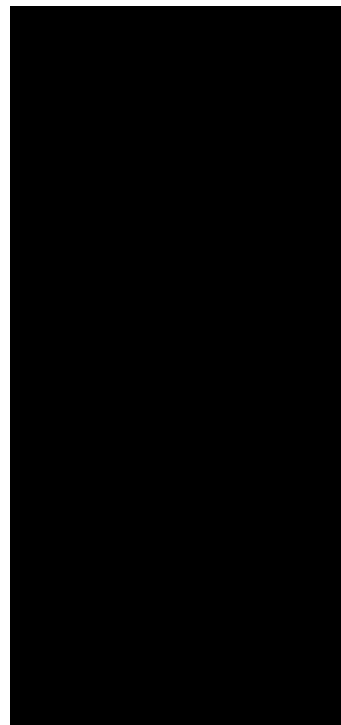
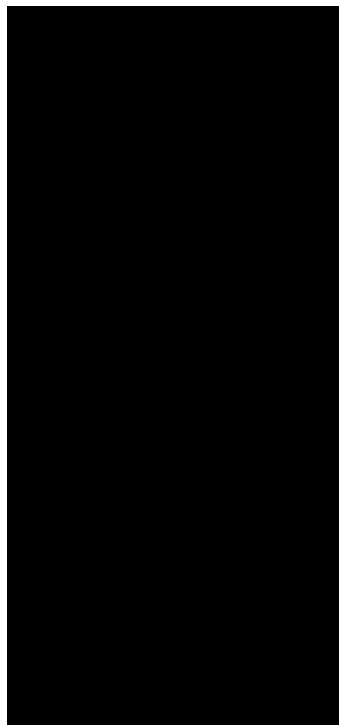
Risultati

Risultati soddisfacenti nella maggior parte dei casi.

Il metodo non tiene però in considerazione le **micro-variazioni** delle dimensioni dei **volti** nè le **diverse lunghezze focali** che le lenti delle fotocamere moderne possono avere.

Inoltre la rete neurale **MiDas** non sempre fornisce stime accurate.

Questi fattori possono introdurre **errori in fase di misurazione**, alterando le distanze finali.



A blue speech bubble with a small tail pointing downwards and to the left. Inside the bubble, the word "Grazie" is written in white, bold, sans-serif font.

Grazie