

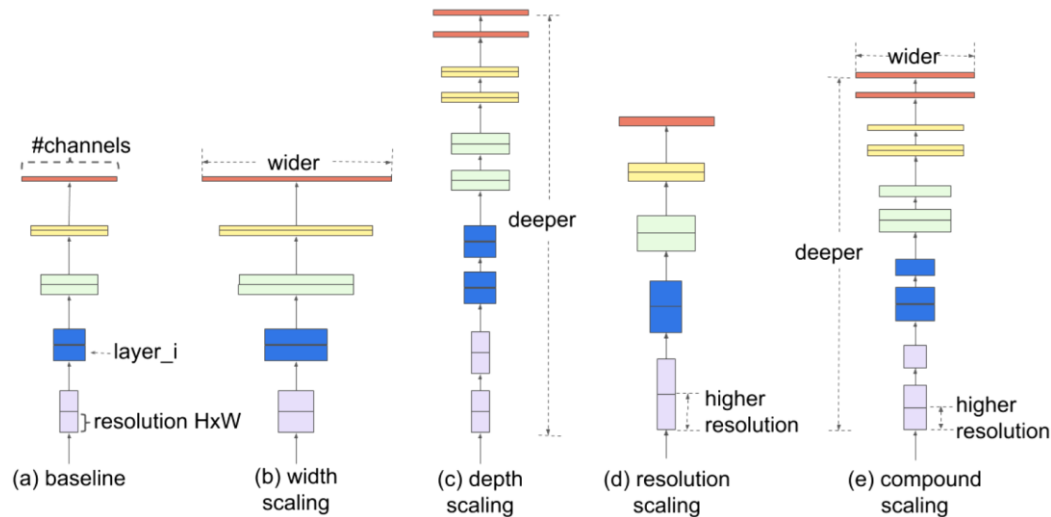


# SIGNTRACKER

ANTONIO SARCHIONE

SISTEMI DIGITALI M

# INTRODUZIONE



- Realizzazione di un'applicazione Android in grado di individuare le mani di uno o più utenti, di analizzarle e di tradurle in lettere dell'alfabeto **ASL** (American Sign Language)
- Riconoscimento in tempo reale tramite l'utilizzo di entrambe le fotocamere dello smartphone
- Rete Neurale Convolutionale (CNN) EfficientNetB0, basata sul ridimensionamento composto
- Pulsanti aggiuntivi per aumentare la consapevolezza dell'utente e le funzionalità

# RETE NEURALE - DATASET

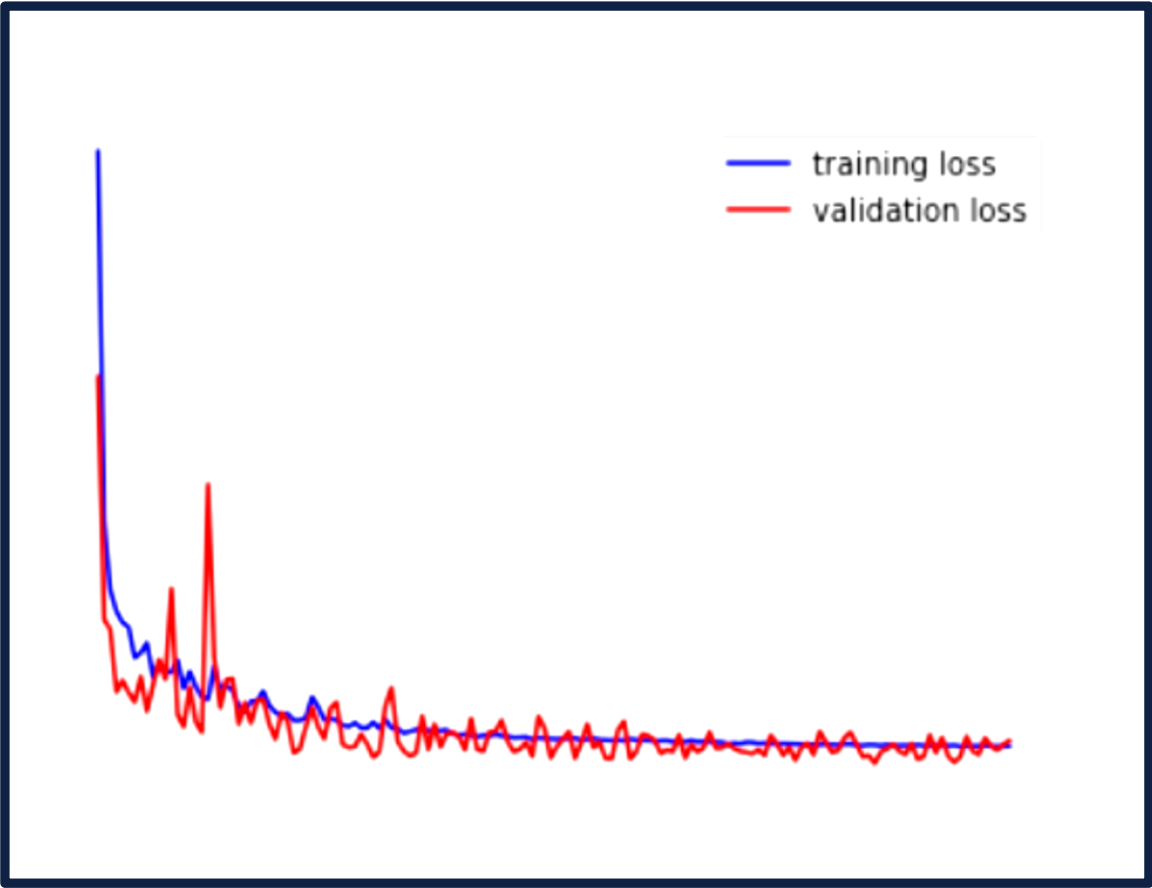
## HANDS

- Dataset pubblico fornito da Kaggle contenente oltre 15 mila immagini raffiguranti mani di una o più persone contemporaneamente, sia di palmo che di dorso

## SIGNS LANGUAGE

- Dataset pubblico fornito da Kaggle contenente oltre 35 mila immagini raffiguranti mani di una o più poste a simboleggiare una lettera dell'alfabeto ASL

# RETE NEURALE - TRAINING



The graph displays two lines: a blue line for 'training loss' and a red line for 'validation loss'. Both lines start at high values and decrease rapidly, then level off with some fluctuations. The training loss is generally slightly higher than the validation loss after the initial drop.

— training loss  
— validation loss

- L'addestramento della rete è stato effettuato tramite Python, utilizzando le librerie di TensorFlow e le API di Keras
- Utilizzo di una particolare architettura di rete neurale convoluzionale, chiamata **EfficientNet**, basata su una metodologia di ridimensionamento uniforme per tutte le dimensioni di profondità, larghezza e risoluzione, utilizzando un coefficiente composto
- EfficientNetB0 si basa sui blocchi residui dell' *'inverted bottleneck method'*, utilizzato dalla rete MobileNetV2, oltre ai blocchi di compressione e stimolazione, dunque risulta leggero e veloce sugli smartphone

# RETE NEURALE – TENSORFLOW LITE

```
model.load_weights(check_path)
converter=tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model=converter.convert()
with open("SignTrackerModel.tflite","wb") as f:
    f.write(tflite_model)
```

- Una volta terminato il processo di addestramento del modello, è stato finalmente caricato il migliore dei checkpoint registrati, in modo da ottenere un grado di riconoscimento ancora maggiore
- Una volta accertato che la rete neurale funzioni in modo equo e che abbia dei livelli di precisione significativamente alti, il modello addestrato viene convertito nel formato TensorFlow Lite, necessario per essere importato all'interno dell'applicazione Android come file di Machine Learning (ML)

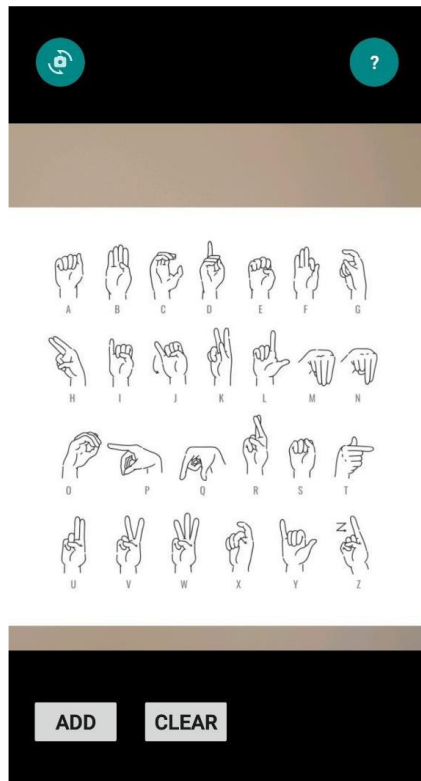
# ANDROID APP



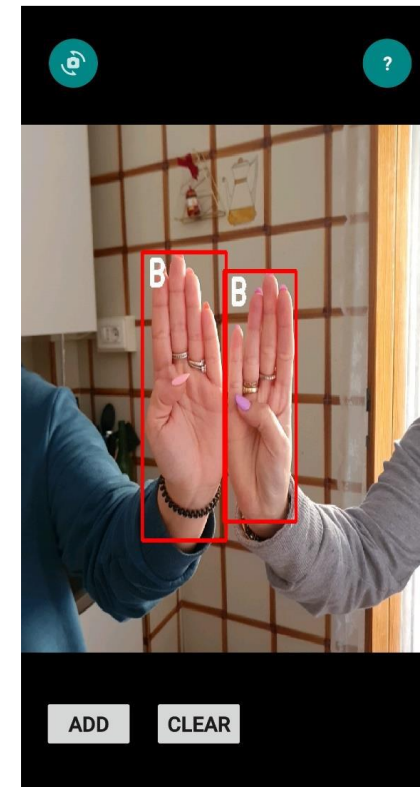
- L'applicazione ha per pregio la reattività in real-time, perciò si è deciso di mantenere un'unica schermata, che permettesse ugualmente di svolgere più azioni, mantenendo una grande semplicità di utilizzo per l'utente
- Questa schermata principale contiene quattro pulsanti che permettono all'utente di creare una propria consapevolezza della potenza del linguaggio dei segni

# ANDROID APP - BUTTONS

## DICTIONARY

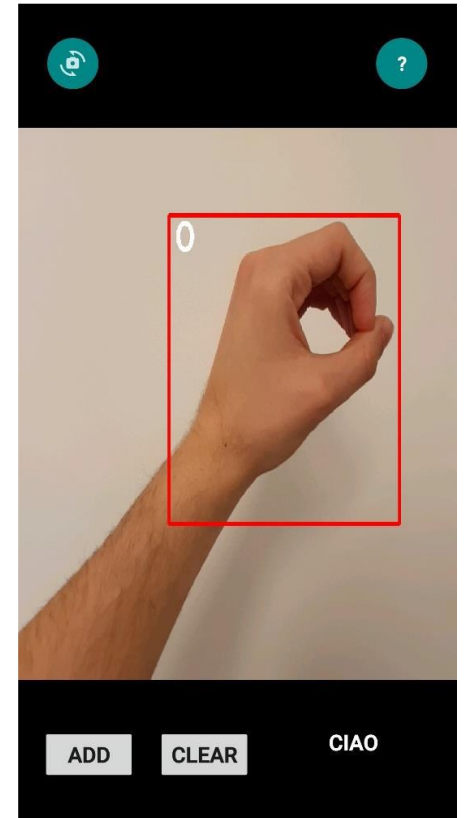
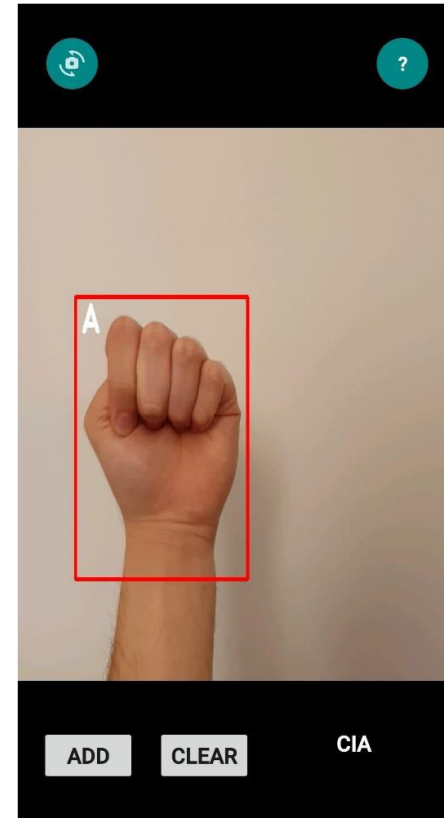
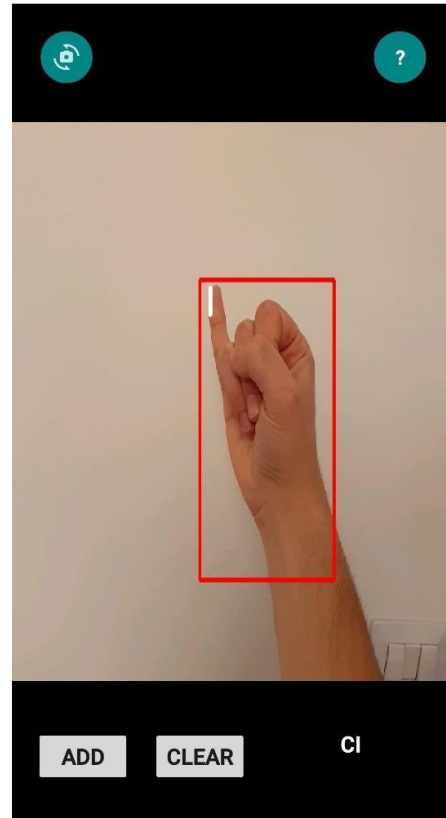
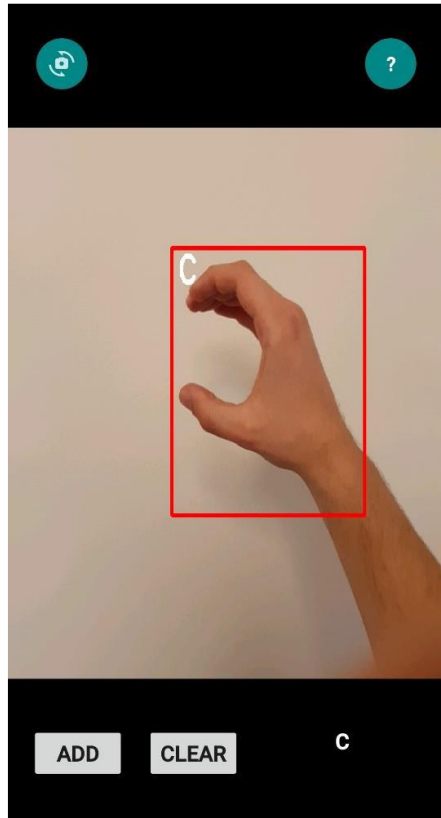


## CLEAR



# ANDROID APP - BUTTONS

## ADD





# CONCLUSIONI



L'obiettivo principale del progetto, ovvero il riconoscimento di tutte le lettere dell'alfabeto ASL con il corrispondente tracking in real time, è stato raggiunto con risultati piuttosto soddisfacenti



L'applicativo Android rispetta le aspettative e svolge i compiti, principali e secondari, in maniera adeguata, senza un eccessivo impiego della CPU.



Il modello addestrato restituisce un valore in uscita con una precisione elevata:

- precisione del 96% per quanto riguarda l'hand tracking
- precisione del 89% per quanto riguarda il sign language tracking



GRAZIE