

---

# EYE TRACKER

---

PROGETTO E ATTIVITÀ PROGETTUALE SISTEMI DIGITALI M

**Luigi di Nuzzo, Daniele Foschi, Filippo Veronesi**

**Dipartimento di Informatica - Scienza e Ingegneria (DISI)**

**Alma Mater Studiorum - Università di Bologna**

Aprile 2022



## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Rete Neurale</b>	<b>2</b>
2.1	Dataset . . . . .	2
2.2	Training . . . . .	2
2.3	TFLite . . . . .	3
<b>3</b>	<b>Progetto Android</b>	<b>4</b>
3.1	Nerd Mode . . . . .	4
3.2	Calibration . . . . .	4
3.3	Game . . . . .	4
<b>4</b>	<b>Conclusioni</b>	<b>6</b>

## 1 Introduzione

Questo progetto prevede la realizzazione di un applicativo Android che, sfruttando una rete neurale, sia in grado di riconoscere il volto dell'utente e di questo ultimo, anche i propri occhi. Il sistema sfrutterà un modello di rete neurale addestrato in modo tale da essere in grado di riconoscere i visi e gli occhi di uno o più utenti grazie ad un dataset di 10 mila immagini di visi umani. L'applicativo permetterà sia di svolgere un filtraggio "live", cioè aprendo la camera frontale ed esterna dello smartphone e individuando real-time la faccia e gli occhi dell'utente, !!!! sia uno "statico" in cui si selezionano dalla galleria una foto e in output vengono riconosciuti gli occhi !!!! DA VEDERE SE METTERE. !!!!!

Inoltre, è stato implementato un gioco (SPIEGAZIONE) aiutato da un tool di calibrazione.

Tale caso di studio è un classico esempio di applicazione di machine learning e il software farà ricorso a una rete neurale convoluzionale (CNN). Tale scelta è dovuta al fatto che una rete neurale rappresenta il modo più comodo e pratico per problemi di object detection, come quello di questa attività in cui vengono individuati gli occhi.

L'applicativo, inoltre, è pensato per la piattaforma Android e quindi tale progetto pone attenzione anche all'uso di risorse in quanto dovrà funzionare su smartphone, ovvero dispositivi embedded.

## 2 Rete Neurale

L'obiettivo primario di questo task è produrre un modello di rete neurale addestrata in grado di riconoscere gli occhi di una o più persone. A tal scopo si è utilizzato un modello messo a disposizione dalla libreria TensorFlow ottimizzato per il training di modelli specifici per object detection di immagini. Visto l'ambito dei sistemi embedded nel quale il progetto complessivo si colloca si è optato per ResNet e MobileNet, una CNN pensata per dispositivi mobile. MobileNet è il primo modello di computer vision pensato per dispositivi embedded basato su TensorFlow. MobileNet è sufficientemente leggera e veloce da essere eseguita su smartphone senza consumo di risorse eccessivo mantenendo comunque una precisione adeguata.

### 2.1 Dataset

Per il training del modello si è utilizzato un dataset pubblico rilasciato da Google (LINK DA INSERIRE) contenente 10 mila immagini di visi umani, dove ogni foto conteneva da uno a più visi umani.

Oltre alle immagini il dataset conteneva un utile file Excel che indicava le coordinate della posizione degli occhi delle persone all'interno dell'immagine.

Prima dell'addestramento della rete era però necessario convertire questi dati di input in file XML in modo tale da poterli poi convertire in TFRecord, utili per TensorFlow. Per questo motivo è stato implementato uno script Java in grado di creare un file XML per ogni foto del dataset con al suo interno le informazioni geografiche della posizione degli occhi.

### 2.2 Training

Si è quindi proceduto all'addestramento della rete tramite Python usando TensorFlow e le API di Keras. Si è ottenuto in output un modello addestrato e pronto all'uso.

Prima di testare direttamente su Android, abbiamo deciso prima di testare e analizzare la nostra nuova rete utilizzando la libreria Matplotlib, utile per visualizzare graficamente via shell i nostri risultati. In particolare abbiamo usato Tkinter, l'unico framework GUI incluso nella libreria standard di Python.

A fini di testing si è preso ad esempio il volto di un personaggio pubblico, quello di Barack Obama, e anche un'immagine contenente più persone in modo tale da poter verificare la correttezza e la precisione della rete anche in condizioni più difficili. (INSERIRE FOTO GUI tkAgg OBAMA e PIU PERSONE)

## **2.3 TFLite**

Dopo esserci accertati che la rete funzionasse correttamente e avesse dei livelli di precisione sopra una certa soglia, si è ottenuto in output un modello addestrato e pronto all'uso, che poi è stato convertito in un formato adatto ai sistemi embedded, quello di TensorFlow Lite.

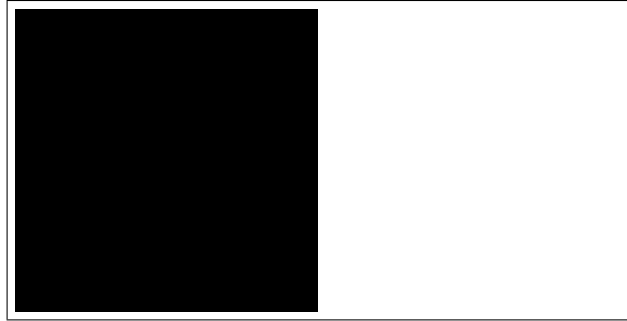


Figura 1: Sample figure caption.

## 3 Progetto Android

### 3.1 Nerd Mode

Citations use `natbib`. The documentation may be found at

<http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf>

Here is an example usage of the two main commands (`citet` and `citep`): Some people thought a thing (Kour and Saabne, 2014a; Hadash et al., 2018) but other people thought something else (Kour and Saabne, 2014b). Many people have speculated that if we knew exactly why Kour and Saabne (2014b) thought this...

### 3.2 Calibration

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi. See Figure 1. Here is how you add footnotes. <sup>1</sup> Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

### 3.3 Game

See awesome Table ??.

---

<sup>1</sup>Sample of the first footnote.

The documentation for `booktabs` ('Publication quality tables in LaTeX') is available from:

<https://www.ctan.org/pkg/booktabs>



## 4 Conclusioni

Gli obiettivi del progetto sono stati raggiunti a pieno e con risultati soddisfacenti.

L'applicativo è conforme alle aspettative e svolge i compiti adeguatamente. Le parti realizzate in Android risultano con prestazione adeguate.

Il modello addestrato restituisce un output corretto la maggior parte delle volte con una precisione di eye tracking del 95%.

## Riferimenti bibliografici

George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014a.

Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.

George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014b. doi:[10.1109/SOCPAR.2014.7008025](https://doi.org/10.1109/SOCPAR.2014.7008025).