

**ALMA MATER STUDIORUM - UNIVERSITÀ
DI BOLOGNA**

DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA

**PROGETTO E ATTIVITÀ PROGETTUALE IN SISTEMI
DIGITALI M**

EYE TRACKER

Luigi di Nuzzo - Daniele Foschi - Filippo Veronesi

Anno Accademico 2021/2022

Maggio 2022

Indice

1	Introduzione	1
2	Rete Neurale	2
2.1	Dataset	2
2.2	Training	2
2.3	TFLite	3
3	Progetto Android	4
3.1	Nerd Mode	4
3.1.1	Bottone Analyze	5
3.2	Calibration	5
3.3	Game	6
4	Conclusioni	7

Elenco delle figure

1	Homepage App	4
2	Schermata Calibration	5
3	Schermata Game	6

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)

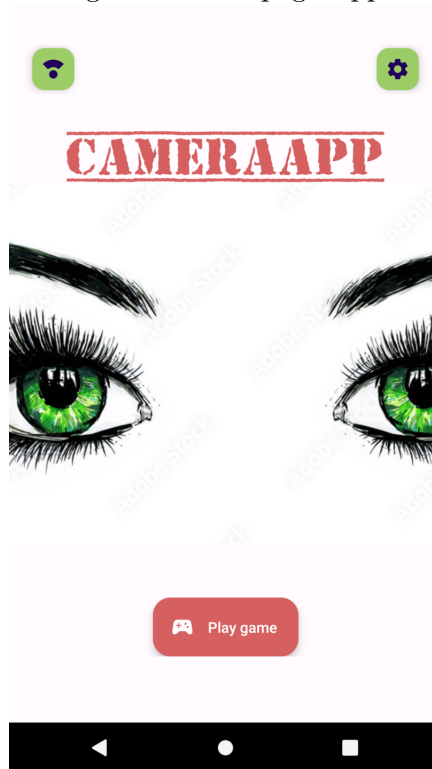
2.3 TFLite

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.

3 Progetto Android

L'applicazione Android una volta avviata presenta la seguente interfaccia:

Figura 1: Homepage App



Sono da segnalare i 3 principali bottoni dell'applicazione:

- Nerd Mode in alto a destra;
- Calibration in alto a sinistra;
- Play Game al centro dello schermo.

3.1 Nerd Mode

Modalità la cui utilità è quella di individuare real time gli occhi della persona, in particolare permette di vedere i boxes creati dalla rete neurale attorno agli occhi dell'utente. Interfaccia che presenta diversi bottoni:

- Preview che permette di avviare la propria fotocamera frontale o esterna;
- Analyze che permette appunto la visualizzazione dei boxes creati dalla rete neurale in real time.

- Fotocamera frontale/esterna per cambiare da una fotocamera all'altra.

3.1.1 Bottone Analyze

3.2 Calibration

Modalità usata per calibrare al meglio la fotocamera e ottenere risultati i più veritieri possibili. Qui di sotto la sua schermata.

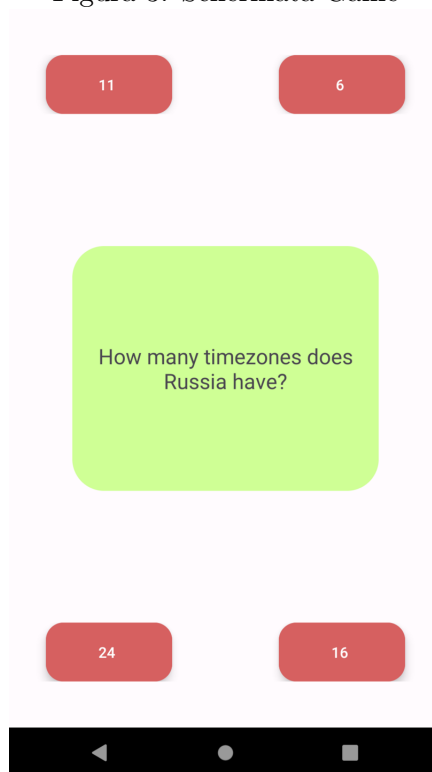
Figura 2: Schermata Calibration



3.3 Game

Modalità usata per completare un mini-gioco sfruttando le potenzialità della rete neurale sottostante. Tramite un mini gioco Quiz infatti, l'utente può rispondere alle domande del Quiz semplicemente spostando lo sguardo sulle possibili risposte e soffermandosi il più tempo possibile sulla sua scelta. I dati delle domande e delle risposte sono state recuperate da uno dei tanti servizi di Api pubblici che si trovano in rete: tramite appunto una richiesta Url, vengono ricevuti i dati casuali in formato JSON per una singola domanda, e una volta ricevuta la risposta, corretta o sbagliata che sia, viene generata un'altra richiesta in modo tale da poter continuare il mini gioco.

Figura 3: Schermata Game



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%.