**Alma Mater Studiorum – Università di Bologna**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SCUOLA DI INGEGNERIA E ARCHITETTURA

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

**TESI DI LAUREA**

In

Calcolatori Elettronici – T

Creazione applicazione che stima la posa delle persone per oscurare il volto.

Privacy nel XXI secolo.

CANDIDATO: Relatore:

Francesco Luzzi Prof. Stefano Mattoccia

Correlatore:

Filippo Aleotti

Fabio Tosi

Sessione

Anno Accademico 2020/21

Sommario

[1 Introduzione 2](#_Toc86760073)

[1.1 OpenPose 2](#_Toc86760074)

[1.2 OpenCv 2](#_Toc86760075)

[2 Obbiettivo dell’applicazione 2](#_Toc86760076)

[3 Sviluppo 2](#_Toc86760077)

[3.1 Input 2](#_Toc86760078)

[3.2 Logica 2](#_Toc86760079)

[3.3 Primi output 2](#_Toc86760080)

[3.4 Modifiche dei threshold 2](#_Toc86760081)

[3.5 Nuovi output 3](#_Toc86760082)

[4 Conclusione 3](#_Toc86760083)

# Introduzione

Alla base di questo progetto vi è la necessità di nascondere l’identità, in modo automatico, delle persone presenti all’interno di una fotografia o di un video.   
Questa rete non riconosce le persone, semplicemente ne riconosce la posa di ognuna di esse concentrandosi maggiormente sulla posizione del volto (naso-collo) cosi da poterlo oscurare.

Bla bla bla approfondimento di questo problema mi ha spinto a creare questa applicazione…

Bla bla bla la presentazione sarà suddivisa in 5 capitoli:  
 -…

# Strumenti utilizzati

Gli strumenti utilizzati per questo progetto sono stati:

## OpenPose

Sfruttando il machine learning è stata creata una rete neurale che passata come input un’immagine restituirà una mappa di probabilità per diverse parti del corpo.

Questa specifica rete utilizza un formato di output delle articolazioni chiamato MPI, questa reta ritorna anche mappe di probabilità per le diverse “giunture” tra le diverse articolazioni, molto utili per riconoscere di chi è il preciso keypoint infatti ( spiegazione per quale motivo usare un algoritmo che lega diversi keypoint a seconda della distanza è sbagliato).

## OpenCv

Libreria open source praticamente essenziale nel panorama della computer vision, sviluppata in C++ e resa disponibile tramite un wrapper anche su python. L’implementazione da me utilizzata è leggermente modificata, è stata compilata con CMake utilizzando il codice della versione “originale” di opencv ed il codice della versione “contrib”, tutto ciò per abilitare/sfruttare l’accelerazione GPU data da CUDA (difatti la mia gpu è di marca Nvidia) che è molto efficiente per questo tipo di workload di tipo algerico/matriciale.

## NumPy

Libreria OpenSource indispensabile per fare operazioni algebriche in python con overhead e durata dell’esecuzione minori, di fatto è una libreria in C con un wrapper ed è molto veloce e performante.

# Sviluppo

Immagini del codice, con spiegazione semplicistica di ciò che sta succedendo

## Input

Immagini utilizzati per il testing e problemi con questa rete neurale.

## Logica

Immagini del codice con immagini esplicatorie della logica dietro la probMap, il ritrovamento dei keypoints e l’utilizzo dei PAF per legare e collegare i diversi Keypoints.

## Primi output

Immagini di output degli input con spiegazione della problematica su immagine 000586

## Modifiche dei threshold

Spiegazione della modifica dei valori di threshold per la rilevazione dei keypoints, specificando che solitamente i keypoint meglio riconosciuti sono quelli del collo e del naso/viso, quindi anche perdere qualche keypoint dei piendi non ci interessa allo scopo dell’applicazione

## Nuovi output

Nuove immagini di output degli input

# Conclusione

Possibili ambiti di utilizzo e possibili migliorie, utilità di questa applicazione.

Ringraziamenti ed addio.