GDPR Compliant Facial Anonymizer for use in Embedded Systems

Kaden Suderman & Mark Slattery
Electronic and Computer Engineering

OÉ Gaillimh
NUI Galway

Project Supervisor: Dr. Peter Corcoran
Co-Assessor: Prof. John Breslin

Introduction

With the world becoming more modernized year by year and the privacy of an individual's information becoming more pertinent every day, new laws and regulations such as the GDPR are being put into place to protect this data. This project is for the development of a facial anonymizer for use in doorbell, security system, or any other real time video capture device to ensure GDPR compliance.

This project aims to consist of:

- A standalone real-time computer vision embedded platform
- Proof-of-concept usability on a Raspberry Pi 4b equipped with a camera module
- Capability to detect faces appearing in a video stream, anonymize them using Gaussian blur, and identify faces based on images stored in a connected repository.

Facial Recognition & OpenCV

- OpenCV, an open-source library developed by Intel and used with Python, is a programming library developed mainly for real-time computer vision.
- To scan for faces, OpenCV uses cascades: a process where images are broken into many blocks and scanned repeatedly. [Fig. 7]
- If the objects passes all scans, it is identified appropriately, and therefore detected as a face, eye, limb, etc.

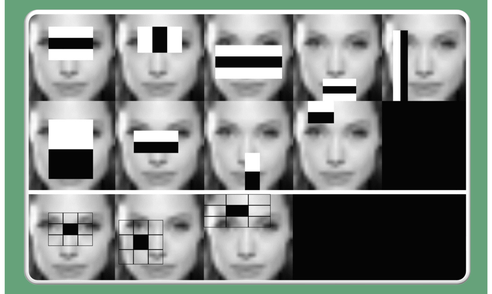


Figure 7. Visualisation of Cascades

Methodology

- Researched different options for which embedded computer system to use for real-time computer vision and processing and chose the Raspberry Pi 4b [Fig. 1] based on its native support for a camera module [Fig. 2], sufficient specs for software requirements, and strong user-base.
- Looked at similar projects in order to gain an understanding of the most effective way to write and implement the software, chose to utilize Python based code running with the OpenCV computer-vision libraries, and a virtual machine test environment (Oracle VirtualBox) running the Debian Linux distribution for Raspberry Pi Compatibility.
- Designed a processing flow [Fig. 3], to illustrate and explain how the facial detection and tracking will operate, focusing on the individual steps which would be taken by the algorithm as well as where facial data would come from, and the desired response based on recognition of specific faces during analysis (i.e. "Paparazzi Mode").

Figure 1. Raspberry Pi 4b

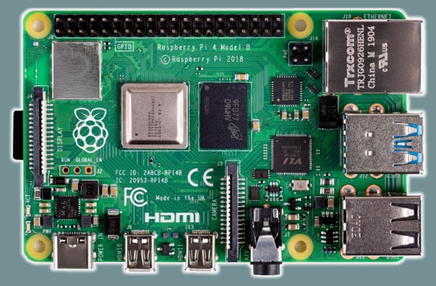




Figure 2. Raspberry Pi 4b Camera Module

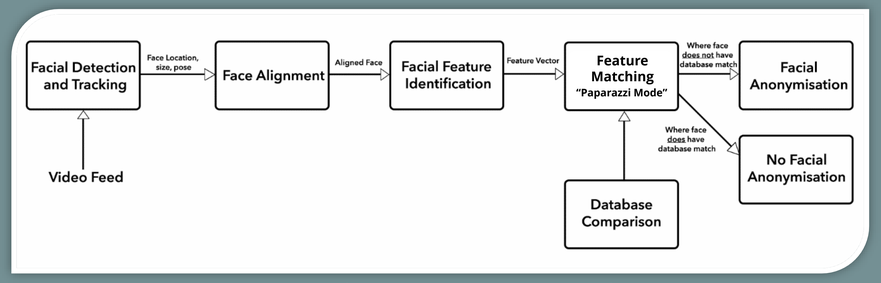


Figure 3. Facial Detection Processing Flow



Figure 4. Eye-tracking with OpenCV

- Implemented eye tracking in order to identify partially obscured faces effectively. [Fig. 4]
- Moved from black-box style obfuscation to using Gaussian-blur techniques. Keeps faces identifiably human but removes specific details. [Fig. 5]
- Learned and implemented GitHub usage to simultaneously collaborate and contribute to project work.

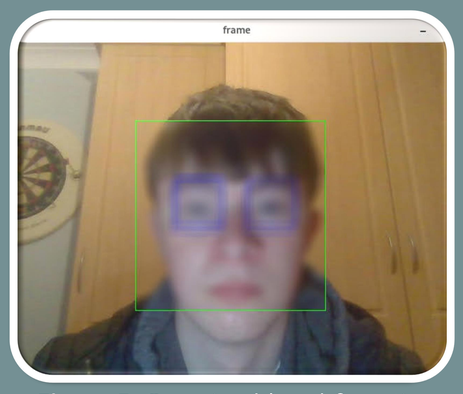


Figure 5. Gaussian-blur obfuscation

- Moved from black-box style obfuscation to using Gaussian-blur techniques which keeps faces identifiably human but removes specific details. [Fig. 5]

Potential Use Cases

This device, when completed, is intended to be distributed as a portable package for use in different real-time camera devices which may be utilized in fully or partially public spaces. These include:

- Doorbell cameras
- Public-area security cameras
- Live streamed footage in public spaces

Future Work

The remaining project objectives are as follows:

- Finalize facial identification features as well as add a "known-faces" repository recognized via neural networks [Fig. 6]
- Refine programming simplicity and improve overall video performance
- Create custom camera compatible housing for the Raspberry Pi

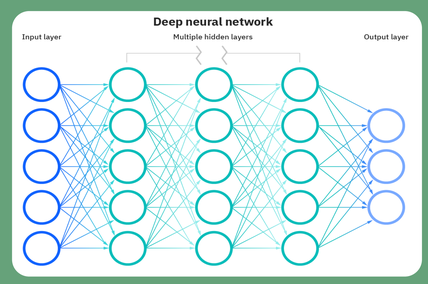


Figure 6. Example of nodes in a Deep Neural Network

Acknowledgements

We would like to thank our project Supervisor, Dr. Peter Corcoran and Co-Assessor John Breslin for their continued guidance and support with this project.