PPM Project Report

Magnus Frater System

Callum Axon (N0727303), Callum Carney (N0741707), Jordan Brightmore (N0732961), Finlay McKinnon(N0743587), Vital Harachka (N0731739), Wing Chiang (T0086366)

Abstract

Magnus Frater (or Big Brother) has been created to help tackle the ongoing issue of security within large open campuses and premises, these sorts of locations inherently have an increased potential for intrusion through unmonitored sections of land. The group analysed the recent spree of attacks on schools and offices (for example the shooting that occurred at the YouTube headquarters in 2018) and found that in a large amount of these attacks there were open doors and spaces that allowed the attacker to enter with ease. As a consequence to this, the idea of creating a facial recognition system to analyse and report known and unknown people within a campus/large open setting was conceived.

As mentioned, the main purpose of the project was to create a system that would accurately detect and report people walking around an area to the associated security team, this data would differentiate between employees or authorised users and unknown people by linking into the companies employee/student database. Not only would this allow a security team to monitor who is within a set area at any one time, but it would also allow administrative users to track any persons movements and activities within a set time frame, through tracking of the targets face across multiple cameras. Another advantage to this project is that administrative users can view analytics in relation to the usage of campus properties, an example use case for this would be within a University. Admins could check what buildings within the campus are being utilised most by students.

After the main purpose behind the project was defined, the group decided on how to proceed in regards to the requirements for the project, most importantly how we should proceed with splitting up the individual hardware and software components so that the system could functions within any scenario or environment. It was decided that there will be 4 different modules, these being:

1. A Raspberry Pi that would be responsible for processing any facial data that is captured by the camera
2. A Camera module that would connect directly to the Raspberry Pi and provide images to the Raspberry Pi
3. A website created for administrators and security personnel to administer and manage hits/rejections.
4. An API (Application Programming Interface) used within the website and the Raspberry Pi for collation and provision of data.

These modules will work together to create the Cameras that report facial data and the web interface that is used to manage the data received by the camera, the connection between these modules was outlined in the design documentation (for example the Data Flow Diagram and Entity Relationship Diagram).

Once the components and requirements were completed, the group began to consider which programming languages and setups would be best suited for the type of project this is (Facial Recognition with

Web Related components). It was clear that Python should be used for the facial recognition section of the project due to its strong existing libraries. NodeJS would be used for the Web Frontend, PHP would be used to power the backend API that links all of the components together and the API would be using a MySQL database to hold all of the data. The system would work in the following way:

1. The Camera feeds data to the Raspberry Pi
2. The Python application on the Raspberry Pi calculates if a face is present
3. Any potential face found is sent to the API where corresponding facial data is requested from the database
4. If no corresponding data is found, then the face is unknown, otherwise the image will be linked to the person the face associates with.
5. The Website will update using data from the API to show new detections, known or unknown.

Once the product had been developed, testing took place to ensure that the facial recognition software worked from a variety of different distances and in unfavourable circumstances (heavy rain, fog, etc). While some of the tests passed, others failed to detect faces when they were present, however this only occurred in extreme circumstances. We made small enhancements to the facial detection algorithm to improve its effectiveness during these scenarios.

Due to the nature of this system, there are a lot of potential legal and ethical issues, people may not consent to the recording of their faces, people may not wish to have their faces processed and stored by this system. Therefore it was important for us to implement a blacklist system that would stop the system from performing facial data processing, however, this is a complex system because we first need to process a persons face to understand what to blacklist, which could cause further legal or ethical issues.

Table of Contents

References

[] [@infotech_2017] [@wayner_2019]