



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

A Study of the use of Facial Recognition Software to improve security within an Academic/Educational Setting

By

Samuel Groves

Supervisor: Andrew Green

Second Marker: John Williams

Project submitted as part of the requirements for
the award of BSc (Hons) Computing (Computer
Networks and Cybersecurity)

April 2022

Declaration

I, Samuel Groves declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Form Of Consent

I Samuel Groves hereby consent that my Project, submitted in candidature for the B.Sc. (Hons) Computing (Computer Networks and Cybersecurity) degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Contents

Abstract	5
1. Introduction.....	5
2. Research & Review of Literature	5
2.1. Secondary Research	5
2.1.1. What is Facial Recognition Software?.....	5
2.1.2. Facial Recognition Environments & Tools.....	6
2.1.2.1. General Use.....	6
2.1.2.2. Governmental & Law enforcement Use	6
2.1.2.3. Educational Use	7
2.2. Initial Concerns	7
2.2.1. Data Protection & Data Storage.....	8
2.2.2. Student Age Related Concerns	8
2.3. Future Implementations & Discussions	8
2.4. Primary Research	9
2.4.1. Interviews.....	9
3. Research Methodology	10
3.1. Philosophy.....	10
3.2. Approach	10
3.3. Strategy	11
3.4. Time Horizon.....	11
3.5. Data Collection Method.....	11
3.6. Development Methodologies	11
3.6.1. SDLC	11
3.6.1. Frameworks & Paradigms	12
4. Design & Implementation	13
4.1. Aims & Objectives	13
4.1.1. Aims.....	13
4.1.2. Objectives.....	13
4.2. System Choices, Justification & Requirements.....	13
4.2.1. Hardware	13
4.2.2. Software & Programming Language	14
4.3. Proposed functionality (IPO).....	15
4.4. Design.....	17
4.5. Implementation	22
5. Testing	23

5.1. Testing Strategy & Audience.....	23
6. Evaluation	32
7. Conclusion & Recommendations.....	32
7.1. Suggestions for further work	33
8. Reflection.....	34
9. References	35
10. Bibliography.....	37

Abstract

FR has emerged as a solution to improve security across many global sectors. The manipulation and use of unique biometric data are used to identify and verify individuals. Safety within academic environments has become an important discussion throughout the past decade.

This report analyses the use of FR to improve security in such an environment. A solution is designed for an academic organisation and an analysis is conducted. The concerns associated with this technology are obtained and addressed, alongside the limitations and considerations. These concerns are found through research methods such as observations, interviews, and secondary literature reports.

The implementation, cost, and accuracy of the technology are discussed and reviewed to produce a designed solution to an academic organisation. The solution uses a range of hardware and software elements (Raspberry Pi 4, Python, and many other elements) to produce a FR application tailored to the needs of an academic organisation.

Future recommendations to improving the application; the accuracy and speed, are discussed. The legal and ethical considerations to implementing this an FR application within the academic sector are also addressed.

1. Introduction

This project assesses the use of facial recognition (FR) software to improve security within an academic/educational setting. FR is an emerging technology. The use and research of this technology within an academic/educational setting is limited. FR has proven to improve security within specific industries such as CCTV. This project will assess whether this is true within an academic/educational environment.

Future recommendations to improving the application; the accuracy and speed, are discussed. The legal and ethical considerations to implementing this an FR application within the academic sector are also addressed.

2. Research & Review of Literature

The data collection methods chosen provided a large range of relevant literature.

2.1. Secondary Research

2.1.1. What is Facial Recognition Software?

To understand the meaning of FR software, we must understand 'Biometrics'. Biometrics is '*the automatic recognition of a person using their distinguishing traits*' [1]. FR is an umbrella term for biometrics and consists of the automatic analysis of distinguishable facial features on an individual [2]. Software is used to achieve this.

Norton, a popular antivirus software, suggest that there are four stages to the process of FR.

Stage 1 - A picture or video of an individual is presented to the FR algorithm (the formula used to identify an individual).

Stage 2 - The geometry of the individuals' facial features is read. Key factors such as the distance between the eyes and distance from forehead to chin are measured. The software may also identify other distinct facial features such as skin tone or birthmarks.

Stage 3 - This facial signature is then compared to a database of known faces. Depending on the organisation or industry, the number of known faces may differ (for example, staff within a security firm or known criminals within a police database).

Stage 4 - The final process is the acceptance or rejection of the individual (and their facial signature)

[4]

This process may differ depending on the technologies used within the FR software.

2.1.2. Facial Recognition Environments & Tools

2.1.2.1. General Use

FR has been used in many different environments/industries ranging from the smartphone industry to law enforcement agencies. Over recent years, FR software has been implemented into several products designed by monopolistic smartphone giants. Samsung have described FR as the ability to 'unlock your smartphone in one quick move' [5].

This suggests that FR is used to assist in the security of the smartphone (the biometric data implemented can be used to unlock the device). Recent studies conducted by Juniper Research also suggest that by 2024, biometric FR will be present in more than 800 million mobile devices (90% of smartphones) [6].

2.1.2.2. Governmental & Law enforcement Use

Law enforcement agencies such as the metropolitan police and Interpol use biometric data to identify criminals (or missing persons) in their established geographical location [7] [8] [9]. To identify these individuals of interest, FR is implemented into CCTV and surveillance cameras owned by the organisation.

This is all in the importance of global/national security. The software is used to catch criminals, prevent a crime from occurring, and find missing persons. A documentary produced by KM Plus Media labelled 'Terrorism Close Calls' show that the use of biometric data, and devices to read such data, are vital in preventing security threats to the general public [10].

FR is also used by other governmental agencies to improve convenience. A number of airports have implemented 'e-gates' to combat the rapid growth of aviation passengers worldwide [11]. These 'e-gates' are automated FR scanners (or other forms of biometric scanners) at border control which are used to handle problems such as congestion caused by the arrival of passengers.

The UK government actively advise passengers arriving in the UK to use the e-gates if they are eligible for faster entry into the country. A passenger will supply their passport to the e-gate (with the image of the passenger facing the scanner). The FR software analyses the passengers face for biometric data through an external camera, alongside key security features provided on the passport (to ensure legitimate passports are presented). If the e-gate fails, the individual will be directed to a member of staff (security) at border control who will perform further checks to determine if the passport is legitimate [12].

The implementation of FR software at airports have shown evidence of “*improved accuracy of detecting fraudulent documents*” [13]. This can be further promoted following a study conducted in 2020. A world map of the number of countries who use FR, are considering using FR, show no evidence of use or have outright banned the technology was generated. The results generated from the study can be shown.

Facial Recognition Status	Total Countries
Currently Using FR	98
Approved but not yet implemented	12
Considering using FR	13
No evidence of use found	68
FR Banned	3

[14]

This data suggests that the number of countries who have adopted FR software has increased overtime. FR was first implemented in airports within the United States followed by an increase in terrorism in 2004 [15]. This study suggests that FR is a valid and accepted form of security within a number of countries.

2.1.2.3. Educational Use

The United States is also an early adopter of using FR within education. This has been implemented throughout the United States due to an increase in school shooting incidents. The use of FR has been marketed as a ‘shield’ against school shooting incidents alongside a way to identify unauthorised intruders [16]. The true effects of the implementation of this technology within these schools may require a longer study over time.

Although there is evidence to suggest FR improves security at places such as airports, there should be some scepticism as to the effects within an academic/educational setting (due to a different range of deterrents implemented and the motives of unauthorised individuals).

A public-school official in Denver [17] argued that implementing FR software could prevent dangerous people from entering the school/educational grounds (campus), such as an expelled student or an individual barred by a restraining order. There may be issues to consider though, students may feel ‘policed’ by such policies and may not feel they have any freedom whilst at school/on campus.

2.2. Initial Concerns

Technological software concerns have been observed in regard to the creation of FR software. There are different methods which can be adopted to create an effective solution. Recognition algorithms and machine learning are some examples [18].

Creating an algorithm or machine learning solution alongside an application for an educational organisation could prove time-consuming. The accuracy of the FR software is also a large concern. If the software is not accurate, it may produce more issues than what it is required to solve (such as overwhelming a staff member as multiple students cannot enter the campus, potentially creating congestion).

Accuracy can be affected by a number of different factors, both intrinsic and extrinsic.

Intrinsic Factors	Extrinsic Factors
Aging	Occlusion
Facial Surgery	Background Noise (visible noise)
Facial Expression	Illumination
	Pose Variation
	Resolution

[19]

Each intrinsic factor and extrinsic factor will need to be understood and thoroughly analysed in order to produce the most effective solution. This will require a number of different testing strategies at the appropriate time.

Occlusion is an increasing concern. Due to the COVID-19 Pandemic, the use of face masks has provided partial occlusion to the face for a large majority of the population (depending on the location and current rules set). This is prevalent in educational settings as throughout 2021 a number of rules have been imposed to ensure the use of face masks to combat the spread of COVID-19. This could prove problematic for FR software as the algorithm analyses biometric data from the face. If there is partial face covering, the accuracy may fall dramatically.

Some religions require individuals to cover parts of their face or surrounding facial features such as the use of Hijab's or Niqab's (worn by women who part take in the Islamic faith) [25]. It is extremely important to follow legislation when designing a product or application. The Equality Act 2010 states that it is unlawful to discriminate against someone based on their religion or belief [26]. This will require careful consideration as there is potential to discriminate accidentally by implementing such technologies.

2.2.1. Data Protection & Data Storage

The implementation of FR software must require students, staff, and other bodies within an academic organisation to understand the technology that they may face. Confidential data will have to be handled and must follow strict guidelines in order to protect said data.

The Data Protection Act 2018 [27] is a national legislation (within the UK) which organisations, businesses, and government bodies must follow to protect and control how personal information is used. There are many factors within this legislation that must be followed such as the fair use of personal data and the strict storage requirements associated.

2.2.2. Student Age Related Concerns

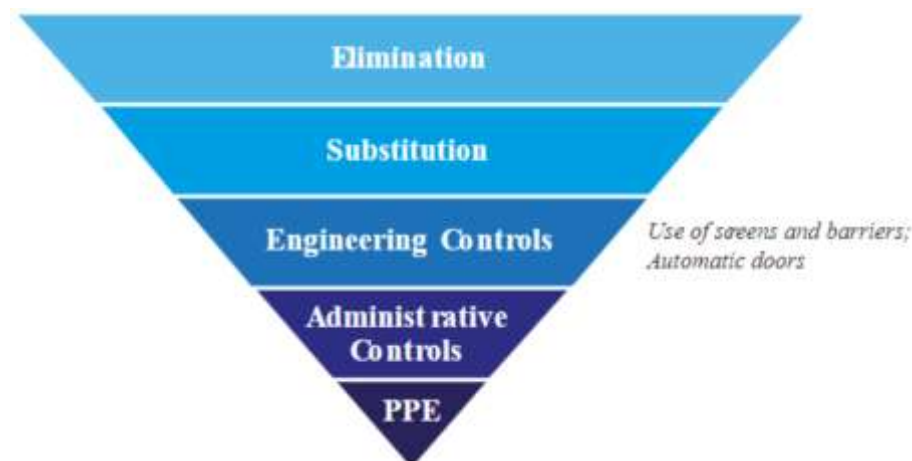
The personal data used throughout academic organisations may contain data on minors (individuals under the age of 18). The GDPR (general data protection regulation) [28] contains a specific subset of protection requirements that organisations must consider when managing this type of data.

Minors may not fully understand the risks involved in regard to providing their personal data to organisations or businesses. In order to use this type of data, a legal guardian to the individual (minor) will need to provide consent.

2.3. Future Implementations & Discussions

Due to the COVID-19 pandemic, there are changing requirements which businesses must follow (or are advised to follow) in order to reduce the spread of coronavirus. Under the Welsh government

guidance, organisations are encouraged to implement one-way systems, improved ventilation, and other methods to achieve this. These are known as 'Engineering controls' [29].



[29]

This might suggest that the implementation of limited contact technologies will be encouraged by the government. FR could be used to achieve this and should be considered when designing a solution.

2.4. Primary Research

2.4.1. Interviews

In order to obtain direct, valid, and important information, interviews were conducted. Relevant authorised individuals from the organisation/client (NPTC) were contacted and a discussion was generated.

A series of questions were devised to further understand the desired solution. Interviews allow further clarifications to be made, creating a more in-depth analysis within the discussion. These interviews will provide important information on the design of the proposed solution; what should be implemented and what should not be implemented.

The College Safety & Security Manager at NPTC was contacted by email. They accepted a video interview arranged through Microsoft Teams. The discussion proved vital in obtaining a detailed understanding of the current security implementations in place within the college, alongside important information that must be followed when a new security measure or technology is implemented.

A new sign-in system had recently been implemented at the entrance to campus grounds. This system is used by contractors and students only. This system is integrated with other technologies within the college, such as the staff database and the email server. This allows automation to play a key part in sending emails and other relevant information to the required members within the organisation (such as a student using the sign-in system to alert a staff member that they are available for a scheduled meeting).

The security manager suggested that the implementation of the new sign-in system took a considerable amount of time and that any security technologies that are to be implemented or discussed within the college must complete thorough checks and testing.

A large emphasise was placed on legislation during the discussion, specifically the GDPR, and how the security technologies within the college must comply with them. They suggested that this is a long process, and that it takes a long time to *“get the technology right”*.

The security manager suggested that FR technology could be *“the next stage of security implementations within the college to create, design, and solidify safe learning communities”*.

The security manager suggested that staff should not have to sign in through these means. They stated that staff do not like to be ‘clock-watched’ and that they may find the implementation of this type of technology to be controlling and restrictive. This could create an extensive list of other issues, especially for staff members who are unionised (could lead to teacher strikes and distrust amongst the organisation).

Through their experience within their current role, they suggested that the software is not the only consideration that should be made when developing a security feature of this type. *“The hardware security features do not work without the software and the software does not work without the hardware”*. This suggests that the hardware technologies used must collaborate with the software chosen.

The security manager expressed concerns on the length of time data is kept on an individual. The newly implemented system currently holds information (specifically photographs) of individuals for 30 days to comply with legislation. This could prove problematic for a system such as the one proposed.

Cost is another concern. The newly implemented sign-in system is expensive not only to build, implement and install, but to ensure year-round support is provided (through a yearly subscription fee). This may suggest that the design and the features implemented within the proposed FR application must not be complex so that limited knowledge is required to solve issues. This also suggests that the cost of the hardware and software must certainly be **considered**.

3. Research Methodology

3.1. Philosophy

The philosophical research methodology that will be required to conduct effective research will be positivism. Security based research will require the positivism philosophical research methodology due to the nature of the question proposed [20].

To understand if FR software can improve security within an academic/educational setting, facts and relevant evidence must be the focus. In order to understand and produce an outcome based on the research question, it is most acceptable to use this approach. This is due to the basis of requiring statistical, non-speculative information and **data**.

3.2. Approach

A deductive approach will be required to produce an outcome to the project. The ability to identify an individual using camera-related technologies is prevalent throughout the security industry (such as CCTV). FR software has already been used to improve security in other sectors, such as the smart

phone industry [3]. Therefore, a deductive approach will be used as a hypothesis which will expand upon an existing theory [21].

3.3. Strategy

The strategy chosen for this study will be experimental [22]. The study is focused on the analysis of FR software to improve security within a specific environment. To understand if security has been improved with the use of this software, experiments will need to be conducted to obtain useful information and data.

A comparison of the current security measures will be analysed, and single variables will be changed in order to produce an outcome. One such variable may be the implementation of FR software compared to the use of standard CCTV software.

3.4. Time Horizon

The time horizon required to conduct this study will be cross-sectional [30]. This is due to the time allocated to the study. Observations will be made before a solution is implemented. To ensure an effective solution, observations will be conducted after the design. Different groups of the academic population will be observed throughout these times such as students, staff members and visitors to the organisation.

3.5. Data Collection Method

There will be a range of data collection methods used throughout the analysis of research. To conduct suitable secondary research [24], previous case studies and literature will be analysed, tested, and compared to assist in producing a hypothesis, supporting the theory that the use of FR within a particular setting can improve security.

A range of primary data collection techniques will also be adopted [23]. Observations of current security measures within a particular academic setting will be analysed to produce important data/information. Interviews with relevant individuals within this educational setting will also be conducted. These data collection methods initially seem the most appropriate for the discussion at hand in providing important data.

3.6. Development Methodologies

3.6.1. SDLC

A sequential development methodology such as waterfall is not appropriate for the development required. Each stage must be completed before continuing with the waterfall methodology. The inability to go back if issues occur could prevent project success. The proposed solution will require a consistent communication with the client and its users. If changes were required, they could not be completed using this methodology.

Therefore, the methodology chosen to conduct this project is agile (iterative). The agile methodology allows regular interactions with the client and users. This is ideal for the project at hand to produce an effective solution; the needs of the client and user must be addressed.

The agile SDLC breaks the project down into smaller cycles/sprints, which allow for precise timings to be considered without the compromise of quality. At the end of each sprint, something is produced. For example, adding a specific feature to the application would require a sprint. Multiple sprints would produce the final outcome, satisfy both the user and the client.

Through the regular interactions with the client and user (creation of multiple project sprints), the application is tailored to suit their needs. This can be time consuming, but the result can be

effective. Other iterative SDLC's such as spiral were dismissed although initially they seem like an effective solution. Spiral manages risk, specifically risks that may be difficult to anticipate at the start of the project. The research conducted found the associated risks with the technology, therefore although a suitable development methodology, it was disregarded.

The application proposed has real clients (Neath Port Talbot College). This will mean that the design will need to be tested and adapted to suit their environment. Due to the nature of the technology, changes to the design are likely to occur in order to fully satisfy the client. This suggests that an iterative development methodology is again more appropriate.

3.6.1. Frameworks & Paradigms

Frameworks are a proposed development methodology which have been identified as a method of developing a solution. Software frameworks provide a foundation for the development [31]. This will allow the design of the solution to be developed faster without compromising required functionalities.

PyQt is a python-based framework used to develop a graphical user interface (GUI) for an application [32]. This python widget is available on many platforms. Related to the current project, detailed research has been performed to assess where this framework may be most suitable. There are other frameworks which have been considered.

WinForms (Windows Forms) [41] is an interface designer used primarily on windows operating systems. This framework uses the programming language C# and allows the implementation of multiple common GUI features or elements (such as text boxes or buttons) to be added to an application with ease using the simple 'drag-and-drop' technique.

WinForms could be used to produce the entire application, but this would have to be on a machine that uses the Windows operating system. This could become costly for the organisation (compared to other solutions) which may produce an ineffective solution.

Alongside the use of frameworks, a programming principle or methodology must be identified and adopted. The proposed design of the solution will require some user input to identify the user (a student ID number).

The proposed design could adopt the object-oriented programming (OOP) paradigm [33]. The OOP paradigm focuses on the data (or objects) presented or called. This paradigm allows specific elements of code, or data structures, to be reused when called. This allows code to be reusable and may assist the debugging process (as errors can be identified easier as each object or class is self-contained).

This is the exact focus required for the solution proposed. Data will be presented and made available to the application ready for identification. The FR algorithm will need to keep updating and checking the live camera feed. Objects, which contain data and code, will be called, and reused. In this scenario, the object would contain code that instructs the algorithm the process required to identify/reject a user and the data would be the users' facial image within the data store (if identified).

There are other paradigms which have been researched. The functional programming [42] paradigm requires subprograms to be read as functions. Further research found that this paradigm was more appropriate to applications that manage large amounts of data. The proposed application will not manage large amounts of data therefore this paradigm was disregarded.

The imperative programming paradigm requires each function to be implicitly coded along every step of the application. As each operation is coded, this means that pre-coded models are not called upon. This paradigm is a reasonable choice as the application can be created exact to specifications. The path of data is also very easy to understand as the path is linear. This issue with this paradigm is that code can quickly become very extensive and confusing. This could lead to a higher risk of errors occurring and difficulty debugging.

4. Design & Implementation

4.1. Aims & Objectives

These aims and objectives were originally proposed during the initialisation of the project. Research completed may already meet some of the objectives set, yet the project is still incomplete.

4.1.1. Aims

- The aim of the project is to assess how FR software can be used to improve the security of current systems within an academic/educational setting.
- Sufficiently test and evaluate a designed solution within a suitable setting (educational setting) and obtain feedback where necessary.

4.1.2. Objectives

- To understand the ethical considerations of implementing FR software in an academic/educational setting.
- To understand the security considerations of implementing FR software.
- To understand the type of data generated and used to produce an effective FR program.
- Obtain first-hand statistics from suitable sources to understand how and if FR software should be used in an educational setting.
- Design a suitable program for an educational setting.
- Develop a suitable program for an educational setting.
- To achieve a clear understanding of the benefits and drawbacks of implementing FR software within an educational setting.
- To understand the different hardware and software requirements of building an effective FR program.

4.2. System Choices, Justification & Requirements

4.2.1. Hardware

Some technologies are required to allow FR technologies to function. A standard peripheral which is required for the design to function is a USB camera. This peripheral can be customised. The USB camera must provide fully HD video quality (at 1080p). This will allow the FR algorithm used the best possibility to identify a user (better accuracy).

Based on these requirements, the USB camera chosen for the solution is the Logitech C920 HD Pro [34]. This technology provides full HD 1080p video at a resolution of 1920 x 1080. It also provides a steady framerate of 30 fps (frames per second). This was chosen due to the low cost (~£50) and the simplicity of the implementation.

The main hardware component proposed and used within the design is the Raspberry Pi 4GB. The Raspberry Pi is a single-board computer that uses the Linux operating system, specifically an extension of the Linux operating system known as Raspbian [35]. The Raspberry Pi provides extensive

compatibility with a range of standard computing peripherals specifically USB cameras and keyboards.

Other hardware components were researched such as the Arduino [40]. This hardware technology is similar to the Raspberry Pi but is less powerful. The Arduino does not have a pre-packaged operating system which could prove problematic during the development of the solution. The greatest benefit to using the Arduino is the cost. The Arduino costs on average, £25 (depending on the specification chosen) but due to the resource intensive technology used within the proposed solution, this technology was deemed unsuitable.

The solution chosen was the Raspberry Pi due to the ability to manage resource hungry technologies, the pre-packaged operating system provided (Raspbian) and the large range of compatibilities available.

4.2.2. Software & Programming Language

The hardware requirements of the project have determined many software choices. The operating system (OS) used for the design proposed is the pre-packaged operating system available on the Raspberry Pi (Raspbian). This is due to the compatibility (of the technologies chosen).

The larger scope of this operating system and hardware chosen is the ability to install compatible libraries and other developer-authorised applications at limited to no cost. Developing a FR algorithm can be extremely difficult and can take a long time to develop. The hardware and software chosen to develop the solution allows the ability for a FR algorithm to be used that already exists.

This will then be implemented into the application and will be tailored to suit the needs of the client (Neath Port Talbot College). The FR algorithm decided is one created by Adam Geitgey [36]. This algorithm is the most supported FR technology on the platform, it is a trusted library, and it is the most up-to-date FR algorithm. The full library can be found on GitHub [37].

Alongside this chosen software technology, a few other libraries must be installed to support the FR algorithm. These are:

- OpenCV – This is a library of programming functions mainly aimed at real-time computer vision [38].
- Dlib – This is a general-purpose software library used to detect the facial landmark of a user [39]. This is completed through estimating the co-ordinates of facial points on a person's face (the FR algorithm uses this to compare an image to the real-time live video stream).

The software discussed specifically relates to the programming and implementation of the technology. Software must be adopted to **design** the proposed application. PyQt and Tkinter are GUI toolkits available on the Raspberry Pi. They were proposed as potential solutions to designing the application. Tkinter provides extreme customisability through designing functions, buttons, and other features within a typical GUI (graphical user interface) through coding.

PyQt on the other hand also provides a large range of customisability but instead of developing the graphical user interface through coding techniques, a pre-packaged design tool can instead be adopted. This tool can save a lot of time depending on the GUI required. Due to the proposed GUI of this application not requiring complex user input or a large range of features, PyQt was adopted as the tool used to develop the graphical user interface.

Finally, the programming language adopted to design the solution. The language chosen is Python. This is due to the large range of resources available, specifically related to the hardware and

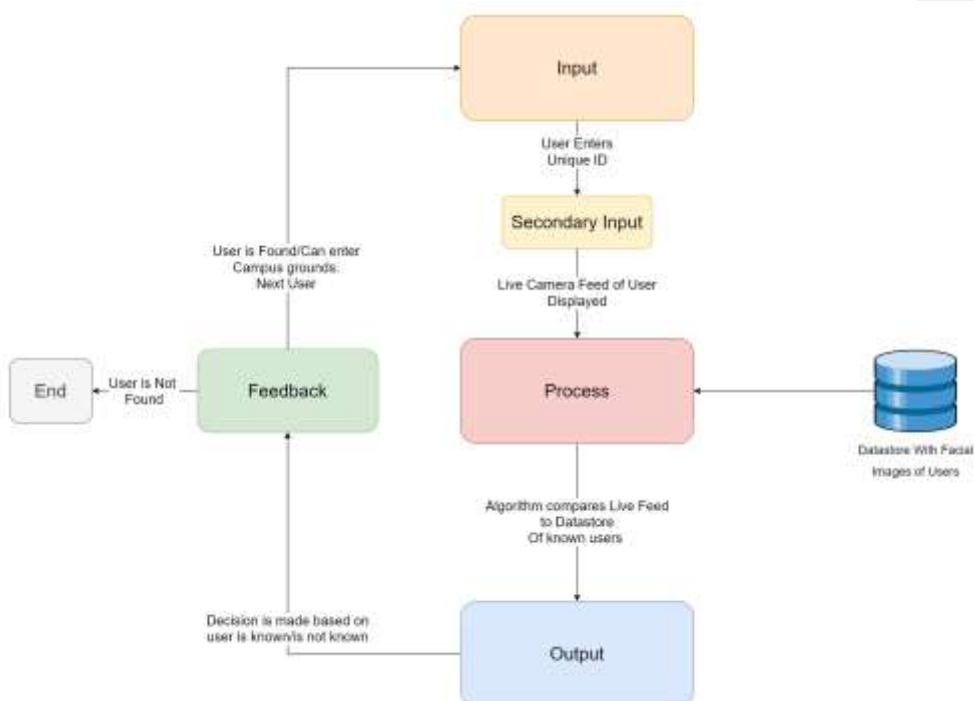
software technologies already chosen. Furthermore, the technologies chosen are **all** compatible with this programming language.

Other languages such as C# or C++ could be used to design the solution. This would require using a different framework to design the GUI such as the .NET framework supplied by Visual Studio. This could create compatibility issues as the framework is not native to the Raspberry Pi [43] and there is limited support available to non-native languages and technologies (errors or bugs encountered could be difficult to solve due to the limited support).

Python is the most appropriate programming language to use in relation to the application proposed as it is the native programming language to the hardware. This will assist in the implementation of the application by reducing the risk of errors or bugs as there is a large range of support available for native technologies.

4.3. Proposed functionality (IPO)

There are three main functionalities of the FR algorithm that must be known. These are the Input, the Process, and the Output (IPO).



The input is based on what the user will provide to the algorithm, or the application proposed. In this case the user will enter their unique ID number (their student number).

Welcome to the NPTC Facial Recognition Scanner!

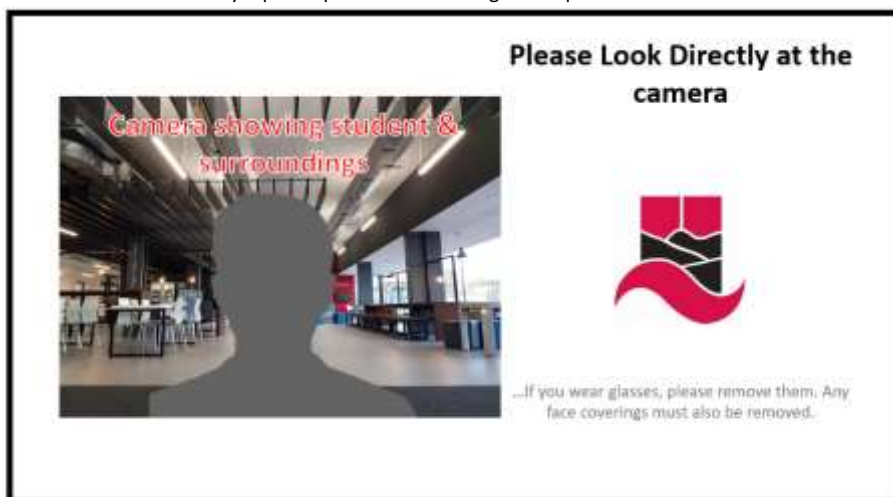


Please enter your student number below

Student Number:

Enter

Once the unique identifier has been entered the user will then be required to face the live camera feed. This is the secondary input required for the FR algorithm process to occur.



With this information, the FR algorithm will compare the user shown on the live camera feed to a datastore of unique known users. This is where the processing within the algorithm occurs.

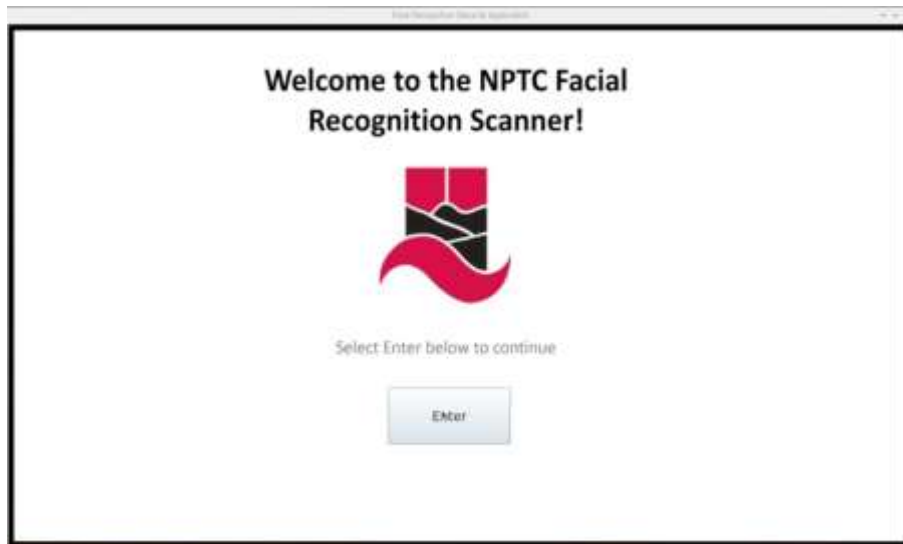
Each image held within the datastore is named after the users' unique identifier (student number). A form of verification will check that the student number provided matches the name of the image. This will only occur if the user is known, and their image is found through the FR algorithm initially. If the user has no student number (no facial image within the datastore), they will be rejected and the student number they provided will not be checked.

The output display provided to the user will be an 'Accepted' interface, or a 'Declined' interface depending on the FR algorithm decision. This can be shown below.



4.4. Design

Throughout the design a number of changes were made to meet the requirements of both the user and client. The proposed login/enter screen changed to support limited user input. Through the research and interviews conducted, requiring input from a user can be tedious and may affect the overall performance of the application developed. The login screen was removed and replaced with a simple 'Enter' button.



Furthermore, discovered during the development was that the user login provided no real benefit to the application. Implementing the user login would require complex coding and with that, may place strain on the hardware (due to hardware limitations). Removing the login allowed less time to be spent using the application and would require less user input which further support this development decision. This decision also removed a hardware peripheral (keyboard) which will lower the overall cost of the solution.

With some development adjustments, the entire application could become touch screen. This could allow easy sanitising of the equipment and is more favourable to the organisation (based on the primary research conducted), especially in regard to recent events (COVID-19 pandemic). It is also more suited to match the other systems within the college (specifically the new touch screen sign-in system).

Once the 'Enter' button had been selected/pressed, the application would continue.

```

1  #Facial Recognition Application for NPFC by Samuel Groves
2  #Imports Below
3
4  import sys
5  from PyQt5.uic import loadUi
6  from PyQt5 import QtWidgets
7  from PyQt5.QtCore import pyqtSlot
8  from PyQt5.QtWidgets import QApplication, QDialog
9  import resource
10 from out_window import Ui_OutputDialog
11
12
13 class Ui_Dialog(QDialog):
14
15     # Initialise loading of qt Main menu/window interface (log in screen)
16
17     def __init__(self):
18         # Defined function for initialisation (Script Start)
19         super(Ui_Dialog, self).__init__()
20         loadUi("mainwindow.ui", self)
21         self.loginLabel.show()
22
23         # Login/enter button below will run the Defined function runCameraWindow
24
25         self.enterBtn.clicked.connect(self.runCameraWindow)
26         # Once button is clicked, load live camera feed based on defined function (line 35)
27
28         self._new_window = None
29         self.VideoCapture_ = None
30         # Ensure single window is shown to user
31         # Ensure camera does not start within this script
32
33     def refreshAll(self):
34         # Defined function to reset video camera
35
36         self.VideoCapture_ = "0"
37         # Direction of USB Camera port connected to Raspberry Pi
38
39     @pyqtSlot()
40     def runCameraWindow(self):
41
42         print("Student has Logged in")
43         # Guidance for command prompt when user has clicked login/enter button
44         self.refreshAll()
45         # Restart USB Camera (refresh)
46         ui.hide()
47         # Main Menu is hidden here
48         self.outputWindow_[]
49         # Create and open new output window (Live camera feed)
50
51     def outputWindow_(self):
52
53         self._new_window = Ui_OutputDialog()
54         # Connection to secondary script, initialises live camera feed interface
55         self._new_window.show()
56         # Live camera feed window shows
57         self._new_window.startVideo(self.VideoCapture_)
58         # Start the live camera feed
59         print("Live Camera Feed has Started")
60         # Guidance for command prompt when live camera feed has started

```

Line 24 is the function that is run when the enter/login button on the main menu is pressed. One this function is activated (the button is pressed), other defined functions are initiated. The live camera feed begins, and the second window (the live camera feed window) is started. The list of imports (lines 4 to 10) ensure connection to the second script and include the navigation elements used by the application designer interface (PyQt).

The original design of the live camera feed window did not change but the coding techniques were adjusted. Originally, the FR algorithm would scan all faces to see if the user is recognised or is unknown. To ensure less strain was placed on the system, the algorithm now only scans for known faces. After a set amount of time (20 seconds), if no face is recognised, the declined screen is displayed.

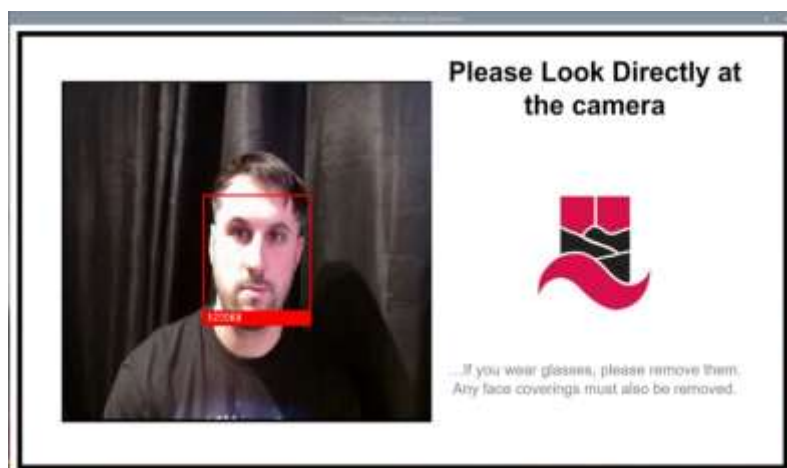
```

76 faces_cur_frame = face_recognition.face_locations(frame) # Connects Box to specific user
77 encodes_cur_frame = face_recognition.face_encodings(frame, faces_cur_frame) # Box location
78 for encodeFace, faceLoc in zip(encodes_cur_frame, faces_cur_frame): # Face recognition algorithm starts
79     match = face_recognition.compare_faces(encode_list_known, encodeFace, tolerance=0.50) # Defines the match variable. Tolerance relates to accuracy of algorithm
80     face_dis = face_recognition.face_distance(encode_list_known, encodeFace) # Estimate and understand distance of user
81     name = "Unknown" # User name unknown till user identified
82     best_match_index = np.argmin(face_dis) # Find best match based on user distance
83
84
85
86
87 if match[best_match_index] and isActive == False: # IF statement implemented for Users who are Identified
88     name = class_names[best_match_index].upper() # Name equals identified user image name
89     y1, x1, y2, x2 = faceLoc # Co-ordinates of face location
90     cv2.rectangle(frame, (x1, y1), (x2, y2), [0, 0, 255], 2) # OpenCV frame and box styles/properties
91     cv2.rectangle(frame, (x1, y2 - 20), (x2, y2), [0, 0, 255], cv2.FILLED) #
92     cv2.putText(frame, name, (x1 + 5, y2 - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, [255, 255, 255], 1) #
93     self.close() # Close Live camera feed window as user is Identified
94     print("Face has been successfully recognized. User is accepted") # Guidance for command prompt to show a user has been accepted
95     loadit("Accepted.ui", self) # Load PyQt GUI Interface "Accepted.ui" for identified user
96     isActive = True # isActive variable changed to true so that script stops updating
97     self.show() # Show "Accepted.ui" interface
98
99
100
101 f = open("../logfile.txt", "a") # Defining log file information and path/directory
102 with open("logfile.txt", "a") as f: #
103     date_time_string = datetime.datetime.now().strftime("%d-%m-%Y %H-%M-%S") # Add date and time to log file
104     f.writelines("In Name: " + name + " | Date time string: " + date_time_string + "\n") # Write to log file with name (ID) of accepted user (Name of image)
105     break # Break code here to ready restart
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139

```

From lines 76 to 82 we have the FR algorithm. This algorithm is obtained through the 'face_recognition' python library designed by Adam Geitgey and has been tailored to meet the design requirements of the application. Lines 87 to 119 are conditions which are followed when a user is found (lines 87 to 97) within the database and also if a user is not found (lines 113 to 117).

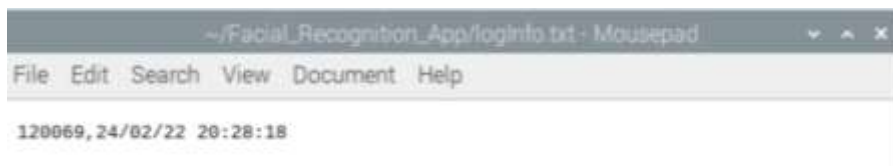
When a user is recognised, they are notified by a square box around their face alongside their identification number within the live camera feed interface. The 'Accepted' interface or window is then shown.



Lines 110 to 111 restart the application after 20 seconds of a user being accepted (the 'Accepted' interface has been shown for 20 seconds). This feature was implemented as it was not fully considered during the initial design phase. Without this feature, the script would have to be manually restarted each time a user has been accepted.

If a user is not identified over 20 seconds of the live camera feed being active, the user is declined and are informed through the display of the 'Declined' interface. They must then contact a nearby staff member for assistance. This occurs within lines 113 to 117. A decision was made to not restart the script after this occurs as the system could potentially be abused. Unidentified/unauthorised users could keep attempting to enter the system in order to access the college. A staff member should be responsible for assessing if the user is an authorised individual at this stage.

An extra feature was implemented during this stage of development. When users are identified (a face has been recognised) a log of the user is created (if it is their first time using the system) or/and updated based on the time and date they were accepted onto campus grounds. This is shown within lines 101 to 106. A file known as 'logInfo.txt' is updated and written to, which produces a log. An example of an automated user log can be shown below.



The FR algorithm is implemented but a path to the database store, with the images of known users, had to be identified and directed.

```
23 class UI_OutputDialog(QDialog):
24     def __init__(self):
25         super(UI_OutputDialog, self).__init__()
26         loadUi("./outputdialog.ui", self)
27
28         now = QDate.currentDate()
29         current_date = now.toString("dd/MM/yyyy")
30         current_time = QDateTime.currentDateTime().toString("HH:mm:ss")
31
32         logInfo()
33
34
35
36     def startVideo(self, camera_name):
37
38         if len(camera_name) == 1:
39             self.capture = cv2.VideoCapture(int(camera_name))
40         else:
41             self.capture = cv2.VideoCapture(camera_name)
42         self.timer = QTimer(self)
43         path = "faces/known"
44         if os.path.exists(path):
45             os.mkdir(path)
46         images = []
47         self.class_names = []
48         self.encode_list = []
49         self.TimerList1 = []
50         self.TimerList2 = []
51         students_list = os.listdir(path)
52
53
54
55         for cl in students_list:
56             cur_img = cv2.imread(f"{path}/{cl}")
57             images.append(cur_img)
58             self.class_names.append(os.path.splitext(cl)[0])
59
60         for img in images:
61             img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
62             boxes = face_recognition.face_locations(img)
63             encodes_cur_frame = face_recognition.face_encodings(img, boxes)[0]
64             encode = face_recognition.face_encodings(img)[0]
65             self.encode_list.append(encodes_cur_frame)
66             self.timer.timeout.connect(self.update_frame)
67             self.timer.start(20)
```

```
# Class which comments to initial script (made menu script)
# What to run when script starts (live camera feed script)
#
# Load second still window live camera feed
#
# Variable 'now' is changed to the current date for future use within log file.
# Format for date when added to log file is defined here
# Format for time when added to log file is defined here
#
# Allows python to read specific elements within Pyqt Interface Design (live camera feed label)
#
# Defined function to start video/live camera feed
#
# OpenCV configuration
# Location USB camera port (directs to the USB path)
#
#
# Location directory for images of known faces
#
# Configuration of images to watch
# Disregard image name until required
# Further OpenCV configuration
#
# Ensure directory is found
#
# Directory found, Read images of known faces
#
# Allow images to be readable/understood by FR algorithm
#
# Camera configuration required by OpenCV
# Ensure camera presents RGB colour scheme (OpenCV uses BGR by default) which cannot be read by algorithm
# Location of box
# Creates frame for box
# Places box around specific facial features
#
# Initialise the update of each frame
#
```

Under the defined function 'startVideo' (lines 36 to 66) we have the initialisation of the camera. Within this function we also have the connection made between the FR algorithm and the live

camera feed. Line 43 direct the FR algorithm to the file directory used to store the images of known users.

The path has been implemented alongside the FR algorithm. Lines 55 to 66 obtain the identification number of a known face alongside elements for the webcam to function (using the python library OpenCV), specifically the orientation of the camera and the style the live feed must be transformed so that the facial algorithm can read or scan a user's facial signature.

Lines 28 to 30 are used to assist the automated log of accepted users by providing the current date and time in a specific format (dd/mm/yy for the date and hh:mm:ss for the time).

```
121 def update_frame(self):  
122     global count  
123  
124     ret, self.image = self.capture.read()  
125     self.displayImage(self.image, self.encode_list, self.class_names, 1)  
126     count = count + 1  
127     #print(count)  
128  
129  
130  
131 def displayImage(self, image, encode_list, class_names, window=1):  
132     image = cv2.resize(image, (640, 480))  
133     try:  
134         image = self.face_rec(image, encode_list, class_names)  
135     except Exception as e:  
136         print(e)  
137         gformat = QImage.Format_Invalid  
138         if len(image.shape) == 3:  
139             if image.shape[2] == 4:  
140                 gformat = QImage.Format_RGBA8888  
141             else:  
142                 gformat = QImage.Format_RGB888  
143         outImage = QImage(image, image.shape[1], image.shape[0], image.strides[0], gformat)  
144         outImage = outImage.rgbSwapped()  
145  
146         if window == 1:  
147             self.inLabel.setPixmap(QPixmap.fromImage(outImage))  
148             self.inLabel.setScaledContents(True)
```

Defined function to update every frame of live camera feed when active
Connection to Face recognition algorithm to keep 'scanning' for faces
Further connection obtain image information for Face Recognition algorithm
Each frame that updates, +1 is added to the count (over X seconds, produce outcome)
Direction for command prompt during testing & Bug fixing
Defined function to display live camera feed with PyQt GUI interface
Set camera feed label size

Exception if error occurs and camera feed cannot be displayed
Guidance for command prompt when this error occurs
Formatting of live camera feed label

Ensure image is readable/formatted correctly

Ensure live camera feed colour is correct
Attach live camera feed to label
Import QPixmap for images within resources
Scale camera feed label to size property allocated

Lines 121 to 126 (the update_frame function) tell the FR algorithm to update each frame the live camera is active. This means that the algorithm is constantly attempting to identify a known face based on the data presented by the live camera until certain conditions are met (user is found or declined).

4.5. Implementation

Due to logistical issues and potential data protection issues. This is due to the technology. The time allocated to the project does not allow sufficient and complete GDPR testing to be conducted. Initial research has shown that technologies implemented within these specific environments must be fully vetted. This is a **long process**.

However, the college environment was simulated. The camera faced a white wall with sufficient lighting supplied. The campus grounds were visited, and similar conditions can be found. This is an assumption. The actual location of the application/solution may differ depending on the logistics within the college. They may decide to place the solution at a different location. An example of a user using the simulated environment can be shown below.



10 potential users were instructed to stand 1.5ft away from the camera (estimate). Each user was different from one another. Whether it be age, gender, or skin colour. Ages ranged from 17 to 56 years old. The users were contacted through connections made within the College.

The users found the experience simple and easy to use. All users found the limited amount of user input to be refreshing. The oldest user said that they usually struggle with new technologies as they can be overwhelming. They felt relieved and stated that when they hear discussions on emerging technologies such as 'facial recognition' they automatically assume that the technology will be difficult to use.

This might suggest that there are some users who are afraid to use the technology because they do not understand it. Some may rather stick to 'what they know' specifically technology that they are used to using. The largest concern found from most users is the long load times between the main window (login/enter screen) and the live camera feed screen.

Each user was satisfied with the speed of the algorithm matching their face. The FR algorithm was able to identify every individual who used the technology within the simulated environment. This may mean that the sample size of users should be increased but due to time limitations, this was not possible. The current implementation testing is not enough to prove the functionality of the application; further testing must be conducted.

5. Testing

To ensure the application is suitable, and meets the proposed requirements, thorough testing must be conducted.

5.1. Testing Strategy & Audience

The audience and user base required to perform some methods of testing are broad. This is due to the diverse range of students, contractors, and other relevant members of the organisation that will require access to campus grounds. This suggests that the only type of testing required regarding the type of user is age, skin colour, facial expression, or whether the user has had facial surgery.

These are the intrinsic factors which may affect the accuracy of the FR algorithm implemented. To test the accuracy of the FR algorithm, a list of extrinsic factors must also be tested. These are as follows:

- The image of a known user supplied to the data store (database of known faces). This specifically relates to the quality of the image.
- The illumination of the live camera feed area (lighting).
- The background noise of live camera feed area (potentially other faces).
- The distance of the user from the live camera feed.
- The height at which the live camera is located.

The Test plan will assess multiple different elements of the application. This will include the testing of usability features (such as buttons or other interactive features) and each application window designed.

The accuracy of the FR algorithm will have a direct impact on the success of the application. The tests conducted on the solution can be shown below.

Intrinsic Test Conditions

Test Number	Test Data	Purpose	Expected Result	Actual Result
1.1	Facial Images of potential users aged 17, 21, 23, and 55 were added to the known faces database. Half of these were female.	To assess if age is a factor for recognising a known user.	Age of the user should not matter due to their unique facial signature.	Age of user had no effect on identification of a known user. An advisory is that up-to-date facial images of users should be implemented into the database so that accuracy of the system is not compromised.
1.2	A variety of facial expressions presented to the live camera feed.	Assess if facial expression may affect the accuracy of the FR algorithm.	Facial expression will not affect the accuracy of the algorithm.	The FR algorithm was able to identify known users based on their facial signature.

Extrinsic Test Conditions

Test Number	Test Data	Purpose	Expected Result	Actual Result
2.1	A known user will have a range of different quality images implemented within the data store of known faces. 	Assess image quality required for the FR algorithm to identify a known user.	FR algorithm should identify a known user. A minimum requirement for the pixel quality of a facial image should be obtained.	Images using the image quality of 720p or above allowed the FR algorithm to identify known users. Images below 720p proved problematic and would sometimes struggle to identify users.

Commented [AG23]: Interesting

				
2.2	A variety of different illuminated environments in which a known user is shown.	Assess the illumination/lighting of the live camera feed and the affect it may have on the FR algorithm.	This test should obtain the minimum and maximum illumination levels required for the application to work without compromising accuracy.	The FR was able to identify known users within environments of low illumination. The FR was not able to identify user when highly illuminated facial image was used. A suggestion can be made that a well-lit area is ideal.

				
2.3	Multiple different known and unknown users within the radius of the live camera feed.	To assess if background noise influences the accuracy of the FR algorithm.	Background noise should <u>not</u> influence the accuracy of the algorithm.	The actual results disproved the expected results. The background noise negatively impacted the accuracy of the algorithm. Different

				users were logged each time this test was run. A suggestion should be made that users should use this system with a cover behind them so that no under users can be identified.
2.4	Known users will test the accuracy of the FR algorithm by facing the live camera from under 1 metre, 1.5 metres and over 2 metres away.	To identify the required distance and limitation at which the user must be for their face to be identified.	This test will obtain the ideal distance a user must be for their facial signature to be read.	The ideal distance found was 1.5 metres (roughly 5 ft). Users who stood under 1 metre away from the live camera provided inaccurate results. This was also true for users who stood further than 2 metres away (6.5 ft).
2.5	Known user will test if they are identified by the live camera feed which is placed at different angles and heights.	To identify the height at which the live camera feed should be located.	This test will obtain the ideal height at which the live camera feed must be placed without the compromise of FR accuracy.	FR was unable to identify user when camera height was too low. User was identified when camera was placed above the height over the user (5 ft 11in) and at a slight angle.

Commented [AG24]: That would be an issue.

Commented [AG25]: Again, quite specific.

2.6		To assess if occlusion affects the accuracy of facial identification/recognition.	Occlusion should affect the accuracy of the FR negatively. The level at which should be identified here.	Known user was unidentified by the FR algorithm. Hmm... I cannot seem to recognise you.  Please contact a nearby staff member.
-----	---	---	--	--

Technical/Design Test Conditions - Software

Test Number	Test Data	Purpose	Expected Result	Actual Result
3.1	Select the 'Enter' button on the main window screen.	To ensure the 'Enter' button directs the user to the live camera feed.	The 'Enter' button will open the live camera feed window.	The Enter button navigates to the desired location (live camera feed window).

Welcome to the NPTC Facial Recognition Scanner!



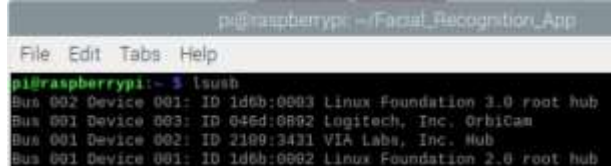
Select Enter below to continue

Enter

3.2	A new user will log in.	To ensure a log is created when a user enters campus grounds (has been accepted) for the first time.	A log will be created of the new user.	A new log was created for the new user. 160421, 02/03/22 16:08:07
3.3	A known user will log in.	To ensure a log is updated when a user enters campus grounds (has been accepted)	The date and time of an appropriate user log will be updated.	The log file updated. 120069, 02/03/22 16:14:31 120069, 02/03/22 17:21:02
3.4	A known and unknown user will use the application. An assessment will be carried out on the specific window that appears for each of the different users.	To identify if the correct interface/screen/window is shown when a user is declined or accepted.	If a user is found, they will be accepted. If a user is unrecognised, they are declined.	Once a user was recognised, the correct window was shown (the accepted window). However, when a user was unrecognised, nothing occurred. A feature was implemented so that when a face had not been recognised for over

				20 seconds, the declined screen would be shown.
3.5	A known user will use the system and once they are accepted, a countdown timer will begin.	To ensure that the application restarts after a user has been accepted on to campus grounds.	The application will restart 10 seconds after the last accepted user (the 'Accepted' window has been shown for 10 seconds).	The application <u>did not</u> restart after 10 seconds of a successful entry. Administration permissions were required. A different function was implemented. Line 111 within the live camera feed script were changed from <code>'os.execv(sys.argv[0], sys.argv)'</code> to <code>'os.execv(sys.executable, ['python'] + sys.argv)'</code> as the initial code created errors within the operating system.

Technical/Design Test Conditions – Hardware

Test Number	Test Data	Purpose	Expected Result	Actual Result
4.1	Python command 'lsusb' will be activated within a terminal window to identify connected USB devices to the Raspberry Pi.	To ensure that the webcam/live camera feed is identified by the Raspberry Pi.	The Raspberry Pi should identify the webcam for a live camera feed to be shown.	The Logitech camera was found/identified. 

6. Evaluation

To evaluate if the project has been successful, we must compare the solution and research conducted against the original requirements. A large range of ethical considerations were initially discovered. The interview (first-hand statistic data generated) conducted with the security manager also displayed a list of ethical considerations.

Due to these discoveries, the original design of the solution had changed. The staff at the college would not be required to use this application if it were implemented. Furthermore, many security considerations of implementing this technology within an educational environment had been found.

FR is a viable option to improve security within educational environments. Discussions with the college safety and security team suggest this software as being the 'next step' to improving security at NPTC.

This discussion produces a range of unanswered questions. Why has FR not already been implemented within the college? Is this technology not completely reliable yet? Is there not enough producers of this technology out there? This could be researched in the future.

A solution was designed, and the application is deemed appropriate. The issues where this project lies are with the hardware requirements. Although thorough research was conducted on the different hardware and software requirements needed to build an effective FR program, further testing and justifications should have been conducted. If this project had to be conducted again, a larger emphasis on the hardware requirements should be considered.

Many resources discovered and obtained used the Raspberry Pi 4 as a hardware peripheral to effectively use FR. However, the original design did not fully consider how resource heavy these applications can be, specifically when integrated with other software elements such as the GUI designer used (PyQt). This placed heavy strain on the Raspberry Pi 4 which led to long load times between the login screen and the live camera feed screen.

This is particularly disappointing as one benefit found when researching the use of FR software is the ability to ease congestion (e-gates at airports). The long load times produced by the application suggest the opposite. The solution can improve security, but this may be at the cost of increasing congestion throughout the college.

Users may argue that this produces an ineffective FR system but as the proposal is specifically related to the improvement of security, evidence can be provided that this objective has been met. A FR system/application has been designed and developed based on the initial research conducted, the client has provided positive feedback, therefore this project has met the aims implemented, ensuring project success.

Commented [AG26]: Failure of initial requirements discussions.

Commented [AG27]: Maybe.

7. Conclusion & Recommendations

FR can improve security within an educational setting. This is only applicable to educational settings that implement the software correctly alongside using the technology as proposed. Educational facilities that already have rigid security implementations can benefit from FR. Organisations that do not have sufficient security protocols already in place may not benefit from this type of technology.

Some potential users within these educational facilities should not have to use the FR technology. Specifically relating to staff members of the organisation. It is assumed that a different method (or

the current method; staff lanyards) will be used by the organisation without the compromise of security related to the FR application **implemented**.

The review of literature provided extensive and appropriate information on specific elements of FR software (such as the benefits and drawbacks). This review also allowed a prototype design to be created based on the existed information discovered and obtained.

The design included the implementation of the python libraries discussed, the FR algorithm, and the multiple user interfaces. During this stage, many changes were made to either solve existing issues or to implement new, appropriate features (such as the user log). This was a long process as each interface had to be designed, buttons and other GUI features had to be implemented, alongside the back-end programming (ensuring all technologies communicated as proposed including hardware and software).

Before the application is implemented into the proposed environment (NPTC College), a recommendation can be made that thorough legislation testing should be performed on the application. Only then can the implementation of the application be tested properly.

The evaluation provides evidence and justification for the project success. Although the project has been deemed successful, there are still issues which had to be addressed. The latency and loading times of the live camera screen application were relatively long which could increase congestion if many users are attempting to use the application at the same time.

7.1. Suggestions for further work

To develop a better tailored application, it is recommended as a future iteration to implement the Welsh language as an option due to NPTC promoting the Welsh language. Other languages could also be implemented so that the application is less exclusive.

Further spending/increasing the budget should be considered as the current hardware in place does not have the processing power/resources available to utilise FR technology fully. The current hardware works, but there is room for improvement.

Considerations should be made for transporting the current application into an interactive touchscreen application. This will be relatively simple as there is only one interactive element currently within the application. The current sign-in system at NPTC is touch screen therefore, this technology better suits the current theme within the college and may also be seen as a more hygienic solution.

Further research should be conducted on the current technology available specifically to these educational facilities. What is there currently to offer? A further analysis should be conducted on the types of software, whether they are bespoke or off-the-shelf, and the costs of each solution should be considered.

The integration of the current sign in system and the FR application proposed must be considered. As found through the initial interviews, if the technology can work together, this would provide great benefit to the college by being able to manage multiple technologies with ease. The application must be fully tested against current legislations in place such as the GDPR and any IEEE standards that the college is looking to obtain within the future. This can be a long process.

8. Reflection

The development of this technology and application has produced a new range of knowledge and information. The use of python to develop a solution is interesting. There are a range of different functionalities which must be followed when using this application (such as code is run from top to bottom always).

The ability to implement other technologies with python tools is shown to be relatively simple. Although this was initially discovered based on the research conducted, there was very little understanding of the limitations of the Raspberry Pi when incorporating multiple technologies or python tools. If this hardware was improved, the application would run faster, reducing the long load time between the main menu and the live camera feed.

The Raspberry Pi 4 would bottleneck the application. The initial research conducted found that FR works well with the Raspberry Pi 4, specifically the face recognition library used within the solution. However, this research did not factor in the implementation of numerous python libraries and technologies working collaboratively.

The entire system could have been improved through multiple iterations. This could have solved the bottleneck problem by producing more efficient code. Due to time constraints, this was not possible. Development should have started earlier to solve this problem and create a better iteration.

Time was taken to learn python, specifically how to implement the FR algorithm into a live camera feed. This meant that throughout the development, there were stages where understanding and learning how to implement specific features were present. This slowed the development of the solution/application quite considerably as sometimes simple feature implementations, such as adding a button, took considerably longer than anticipated.

Implementing the technology proved difficult due to a large range of extrinsic limitations. Thorough testing conducted shows a good use of the potential issues or concerns that the application may encounter. This might suggest that with the testing information found, the college may implement the application at a location of their choice when they are ready.

FR is an emerging technology, there is limited information on these types of technologies especially regarding the use within specific environments. The development provided an immense amount of learning through understanding basic python, to implementing emerging technologies, and understanding how to design an effective user interface followed by much more.

This project has produced an outcome to an industry where this technology can be used by using real clients, users, and performing real tests.

9. References

1. Woodward J, Horn C, Gatune J. Biometrics - A Look at Facial Recognition [Internet]. RAND; 2003 [cited 10 January 2022]. Available from: <https://apps.dtic.mil/sti/pdfs/ADA414520.pdf>
2. Sikender M. Facial Recognition Technology [Internet]. Wilmington University; 2020 [cited 7 December 2021]. Available from: https://www.researchgate.net/publication/342067011_Facial_Recognition_Technology
3. About Face ID advanced technology [Internet]. Apple Support. 2021 [cited 7 December 2021]. Available from: <https://support.apple.com/en-gb/HT208108>
4. Symanovich S. What is facial recognition? How facial recognition works [Internet]. Us.norton.com. 2021 [cited 7 December 2021]. Available from: <https://us.norton.com/internetsecurity-iot-how-facial-recognition-software-works.html>
5. Samsung. How does Face recognition work on Galaxy S20, S20+, S20 Ultra, and Z Flip? [Internet]. The Official Samsung Galaxy Site. 2021 [cited 7 December 2021]. Available from: <https://www.samsung.com/global/galaxy/what-is/face-recognition/>
6. Pascu L. Biometric facial recognition hardware present in 90% of smartphones by 2024 | Biometric Update [Internet]. Biometric Update |. 2020 [cited 7 December 2021]. Available from: <https://www.biometricupdate.com/202001/biometric-facial-recognition-hardware-present-in-90-of-smartphones-by-2024>
7. Metropolitan Police. Live Facial Recognition [Internet]. 2021 [cited 10 January 2022]. Available from: <https://www.met.police.uk/advice/advice-and-information/facial-recognition/live-facial-recognition/>
8. J. Del Greco K. Facial Recognition Technology: Ensuring Transparency in Government Use — FBI [Internet]. Fbi.gov. 2019 [cited 10 January 2022]. Available from: <https://www.fbi.gov/news/testimony/facial-recognition-technology-ensuring-transparency-in-government-use>
9. Identifying terrorist suspects - Biometrics [Internet]. Interpol.int. 2021 [cited 7 December 2021]. Available from: <https://www.interpol.int/en/Crimes/Terrorism/Identifying-terrorist-suspects>
10. Terrorism Close Calls (TV Series 2018–) - IMDb [Internet]. IMDb. 2018 [cited 7 December 2021]. Available from: https://www.imdb.com/title/tt6775816/episodes?season=1&ref=tt_eps_sn_1
11. Sanchez J, Moctezuma D, Conde C, et al. Automated border control e-gates and facial recognition systems [Internet]. 2016 [cited 14 December 2021]. Available from: <https://www.sciencedirect.com/science/article/pii/S0167404816300736?via%3Dihub>
12. UK Home Office. eGates eligibility and how to use the eGates [Internet]. 2019 [cited 14 December 2021]. Available from: <https://www.youtube.com/watch?v=9TG31dOttMM>
13. Khan N, Efthymiou M. The use of biometric technology at airports: The case of customs and border protection [Internet]. 2021 [cited 14 December 2021]. Available from: <https://www.sciencedirect.com/science/article/pii/S2667096821000422>
14. Ghosh I. Mapped: The State of Facial Recognition Around the World [Internet]. Visual Capitalist. 2020 [cited 14 December 2021]. Available from: <https://www.visualcapitalist.com/facial-recognition-world-map/>
15. Glick M. Authority To Collect Biometric Data From Additional Travelers [Internet]. Discover. 2008 [cited 14 December 2021]. Available from: <https://www.federalregister.gov/documents/2008/12/19/E8-30095/united-states-visitor-and-immigrant-status-indicator-technology-program-us-visit-enrollment-of>
16. Andrejevic M, Selwyn N. Facial recognition technology in schools: critical questions and concerns [Internet]. Taylor & Francis. 2020 [cited 14 December 2021]. Available from: <https://www.tandfonline.com/doi/full/10.1080/17439884.2020.1686014>

17. Heilweil R. Schools are using facial recognition to try to stop shootings. Here's why they should think twice. [Internet]. Vox. 2019 [cited 14 December 2021]. Available from: <https://www.vox.com/recode/2019/12/20/21028124/schools-facial-recognition-mass-shootings>
18. Introna L. Facial Recognition Technology - A survey of Policy and Implementation Issues [Internet]. Eprints.lancs.ac.uk. 2010 [cited 14 December 2021]. Available from: <https://eprints.lancs.ac.uk/id/eprint/49012/1/Document.pdf>
19. Anwarul S, Dahiya S. A Comprehensive Review on Face Recognition Methods and Factors Affecting Facial Recognition Accuracy [Internet]. University of Petroleum & Energy Studies; 2020 [cited 14 December 2021]. Available from: https://www.researchgate.net/publication/337446642_A_Comprehensive_Review_on_Face_Recognition_Methods_and_Factors_Affecting_Facial_Recognition_Accuracy
20. Positivism - Research Methodology [Internet]. Research-Methodology. [cited 30 December 2021]. Available from: <https://research-methodology.net/research-philosophy/positivism/>
21. Deductive Approach - Research Methodology [Internet]. Research-Methodology. [cited 30 December 2021]. Available from: <https://research-methodology.net/research-methodology/research-approach/deductive-approach-2/>
22. Mcleod S. Experimental Methods in Psychology | Simply Psychology [Internet]. Simplypsychology.org. 2012 [cited 30 December 2021]. Available from: <https://www.simplypsychology.org/experimental-method.html>
23. Primary research: Home [Internet]. Library.soton.ac.uk. 2021 [cited 30 December 2021]. Available from: <https://library.soton.ac.uk/sash/primary-research>
24. Secondary research [Internet]. Oxford Reference. [cited 30 December 2021]. Available from: <https://www.oxfordreference.com/view/10.1093/oi/authority.20110803100451291>
25. What's the difference between a hijab, niqab and burka? [Internet]. Bbc.co.uk. 2018 [cited 4 January 2022]. Available from: <https://www.bbc.co.uk/newsround/24118241>
26. Equality Act 2010 [Internet]. Legislation.gov.uk. 2021 [cited 4 January 2022]. Available from: <https://www.legislation.gov.uk/ukpga/2010/15/contents>
27. Data protection [Internet]. GOV.UK. 2018 [cited 4 January 2022]. Available from: <https://www.gov.uk/data-protection>
28. General Data Protection Regulation (GDPR) – Official Legal Text [Internet]. General Data Protection Regulation (GDPR). 2018 [cited 4 January 2022]. Available from: <https://gdpr-info.eu/>
29. Alert level 2: guidance for employers, businesses and organisations [Internet]. GOV.WALES. 2022 [cited 4 January 2022]. Available from: <https://gov.wales/alert-level-2-guidance-employers-businesses-and-organisations-html>
30. Setia M. Methodology Series: Cross-sectional Studies [Internet]. 2016 [cited 4 January 2022]. Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4885177/>
31. Andreica A. Integrated Software Framework for Academic Management [Internet]. ResearchGate. 2009 [cited 1 February 2022]. Available from: https://www.researchgate.net/publication/228966006_Integrated_Software_Framework_for_Academic_Management
32. What is PyQt? [Internet]. Pythonpyqt.com. 2020 [cited 1 February 2022]. Available from: <https://pythonpyqt.com/what-is-pyqt/>
33. Raut R. Object-Oriented Programming (OOP) [Internet]. Irjet.net. 2020 [cited 1 February 2022]. Available from: <https://www.irjet.net/archives/V7/I10/IRJET-V7I10247.pdf>
34. Logitech C920 Pro HD Webcam [Internet]. Logitech. 2022 [cited 16 February 2022]. Available from: https://www.logitech.com/en-gb/products/webcams/c920-pro-hd-webcam.960-001055.html?kpid=242791000_en_GB%2C242791000_en_GB

35. What is a Raspberry Pi? [Internet]. Raspberry Pi. 2022 [cited 16 February 2022]. Available from: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
36. Geitgey A. Adam Geitgey Projects [Internet]. Adam Geitgey. 2021 [cited 16 February 2022]. Available from: <https://www.adamgeitgey.com/>
37. GitHub - face_recognition: The world's simplest facial recognition api for Python and the command line [Internet]. GitHub. 2019 [cited 16 February 2022]. Available from: https://github.com/ageitgey/face_recognition
38. OpenCV Services [Internet]. OpenCV. 2022 [cited 16 February 2022]. Available from: <https://opencv.org/>
39. Dlib C++ Library [Internet]. Dlib.net. 2022 [cited 16 February 2022]. Available from: <http://dlib.net/>
40. Arduino - Products [Internet]. Arduino.cc. 2022 [cited 16 February 2022]. Available from: <https://www.arduino.cc/en/Main/Products>
41. Docs.microsoft.com - Windows Forms Documentation [Internet]. 2022 [cited 22 February 2022]. Available at: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-6.0>
42. Major programming paradigms [Internet]. Cs.ucf.edu. 2022 [cited 8 March 2022]. Available from: <http://www.cs.ucf.edu/~leavens/ComS541Fall97/hw-pages/paradigms/major.html>
43. C# and .NET to develop GUI Apps for the Raspberry Pi [Internet]. Tutorials-raspberrypi.com. 2022 [cited 16 March 2022]. Available from: <https://tutorials-raspberrypi.com/write-raspberry-pi-gui-apps-c-sharp-dot-net/#:~:text=NET%20to%20develop%20your%20own%20GUI%20apps%20for%20the%20Raspber%20Pi,-Facebook%20Twitter%20LinkedIn&text=In%20addition%20to%20creating%20graphical,Studio%20on%20the%20Raspberry%20Pi.>

10. Bibliography

- Groves S. Video Demo for NPTC Facial Recognition System [Internet]. Youtu.be. 2022 [cited 1 March 2022]. Available from: <https://youtu.be/LNbnj5aM8OM>
- LOHR S. Facial Recognition Is Accurate if You're a White Guy [Internet]. Cs.toronto.edu. 2018 [cited 1 March 2022]. Available from: <http://www.cs.toronto.edu/~bor/196f21/facial-recognition-nytimes.pdf>
- Denton E, Gebru T, et al. Investigating the Ethical Concerns of Facial Recognition [Internet]. ACM Conferences. 2020 [cited 2 March 2022]. Available from: <https://dl.acm.org/doi/abs/10.1145/3375627.3375820>
- Smith A. More Than Half of U.S. Adults Trust Law Enforcement to Use Facial Recognition Responsibly [Internet]. Pewinternet.org. 2019 [cited 2 March 2022]. Available from: https://www.pewinternet.org/wp-content/uploads/sites/9/2019/09/PDL_09.05.19_facial-recognition_fullreport_2.pdf
- Brey P. Ethical aspects of facial recognition systems in public places [Internet]. ResearchGate. 2004 [cited 18 March 2022]. Available from: https://www.researchgate.net/publication/235306881_Ethical_aspects_of_facial_recognition_systems_in_public_places