Aaron Burton, David Drohan, Tony Browne

TUD

Image processing group assignment research document

Contents

[Introduction 2](#_Toc181363045)

[Section 1: Theory 2](#_Toc181363046)

[What is the Project? 2](#_Toc181363047)

[Why this Project? 2](#_Toc181363048)

[Section 2: Competitors & Papers 3](#_Toc181363049)

[Corporations 3](#_Toc181363050)

[Visage SK 3](#_Toc181363051)

[Amazon Rekognition 3](#_Toc181363052)

[Final Year Projects 3](#_Toc181363053)

[Passwordless Authentication using Facial Recognition for Websites 3](#_Toc181363054)

[An Electronic Voting Web Application with Facial Recognition Verification 4](#_Toc181363055)

[Home Security Door Lock, Alarm and Facial Recognition Surveillance Camera 6](#_Toc181363056)

[References 8](#_Toc181363057)

Facial Feature Detection Application

# Introduction

This document will cover the research done in preparation for the design and implementation of the Facial Feature Detection Application Group Project.

The first section will cover the theory of our project, explaining what the app is and its use cases, what potential scenarios it can be implemented into to improve outputs or systems, and why a company or individual might want to use it.

The second section will cover the list of competitors with similar ideas. Understanding what features, they provide, and what approach they took to achieve them. It will identify the architecture used to make the project, the image processing techniques implemented, and their reason for implementing them. Proceeding with identifying their strength, weaknesses, and where we can improve or take a different approach. This section will also touch base on research papers explain techniques with similar ideas.

Finally, the third section will cover the risk assessment of the project. Identifying the potential risks of the project, categorizing them into levels of severity and likelihood, as well as covering the steps we will take to try mitigating these risks to the best of their ability.

# Section 1:

## What is the Project?

Our proposed project is a facial feature detection/ recognition program, that will capture a live image of a face and be able to provide details of the face in the image. These features we intend to include are hair colour, eye colour, skin colour and facial hair colour. This program will be able to describe the users face and highlight the areas of the features that the user has selected on a facial feature menu provided. The program will be able to compare other faces with each other based on the features they have in common. There will also be a smile intensity feature that will determine based on the facial images mouth how happy or sad the user is in the image.

## Why this Project?

The purpose of this project is for individuals or companies to be able to retrieve detailed information about an individual present on a camera without the use of machine learning. This information can be useful in identifying a person of interest or controlling an individual’s access to certain areas for corporations or government bodies. For an individual it can be useful for identifying their faces’ exact colour description and information about their facial features.

It can be further built upon or implemented in future applications for helping a system identify key features regarding individuals faces and help AI systems understand or be able to describe and compare different individuals better. Building upon this system can also be used to track changes in features and movements to identify potential health risks, aging patterns, or emotion identification.

# Section 2:

From our research we have identified two key corporate competitors, three final year projects, and one key research paper with similar implementation of features to our project. We’ll discuss each below.

## Corporations

### Visage SK

A business with a free demo to use on their website. The demo tracks the users face, identifying their age, gender, and emotional state. Although this has accurate gender detection and semi-accurate age detection, the emotion detection struggles with facial occultation’s such as glasses, hair, hat, scarfs, etc. It uses point distribution modelling when identifying facial features, and plots key landmarks around the eyebrows, eyes, nose, and mouth; similar to what we want to do with our project. They tackled this by branching off AI and implementing some sort of machine learning technique more than likely Convolutional Neural Networks on a large set of data with predetermined features and descriptions. Our project will want to try avoiding this as it further leans towards machine learning, instead of displaying more standard image processing techniques we want to demonstrate.[1]

### Amazon Rekognition

A large corporation branch off that focuses on automating and lowering the cost of image recognition and video analysis using machine learning. Again, machine learning is a tool we want to avoid taking inspiration from. However, understanding how it works and using it as a framework to compare our project to will be beneficial to the development and comparison of the project.[2]

## Final Year Projects from TU/DIT

### Passwordless Authentication using Facial Recognition for Websites

#### Description

The purpose of this project is to provide an alternative solution that would be simpler and faster way of logging a user into to their accounts on websites. The project is an API that can be easily implemented into websites to provided facial recognition authentication to allow users to sign in with a camera attached to their device.

#### Methodology

The application implements the OAuth 2.0 standard making it a pluggable API for web developers to easily integrate into their websites. The project was developed using a waterfall methodology, which is the methodology our team plans on implementing for the life cycle of the project.

#### Architecture

The architecture of this project involved the client-side including the user interface and the camera attached interacting with the 3rd party websites which held the Front-end and back-end code. This would then gather the necessary information and preform the necessary functions which are then sent to the cloud service. The cloud service contains the login/register page which runs on the Flask server, this calls the Facial Recognition API which then preforms the necessary image processing functions and compares the taken photo with the image stored in the Postgres SQL database and if a match, confirms the user’s authentication and logs them in.

#### Image Processing

The library used for the image processing techniques in this project was OpenCV. Their use of the OpenCV library will differ from how our group will tackle identifying features and faces as they use it for machine learning and image processing. Although our group will use it for image processing we will attempt to avoid machine learning as much as possible for this project.[3]

The features used from OpenCV for image processing include:

* Connects to the web camera on the device
* Reads in images from camera
* Changes Pixel Colour for Anti-Spoofing
* Image Saving to Folders
* Image Resizing
* Converting to RGB
* Image Sharpening
* Passes images to Face Recognition Library

Our project will implement most of these same features using OpenCV to read in images from a camera, save images write over them to highlight features in the images and reformatting them to have a consistent output including resizing, colour formatting, denoising and sharpening.

#### Strengths

The strengths of this project identified:

* Anti-spoofing preventing impersonating o masquerading
* Image sharpening for sharper images
* Denoising to remove noise from images
* Image formatting to provide a consistent output
* Use of different libraries
* Simple but effective project architecture

#### Weaknesses

The weaknesses identified through our research of this project involved:

* Reliance on machine learning for identifying facial features
* Load Balancing were too many requests caused the system to significantly slow down
* Lack of Error Detection for user camera quality

### An Electronic Voting Web Application with Facial Recognition Verification

#### Description

This purpose of this project is to have a web application that allows users to be verified using a picture of their face and a picture of their form of identification (Passport), then using image processing techniques to analyse the similarities of the two faces and if the faces have a match of 98% or higher and they do not already appear in the facial database then it casts their vote for the representative of their choice.

#### Methodology

The chosen methodology for this project by Katie Crowley was Agile, for the use of multiple sprints with the initial sprints on planning and design of the project, followed by subsequent sprints focused on implementing, testing and improving the system. Although our project will primarily work as a waterfall methodology, we do plan on implementing small sprints on improving minor errors or features throughout the project.

#### Architecture

The technical architecture of this project involves the Client-side, and web application. The client side is where the interactions between the user and the device are taken. The user selects their respective and then the device takes a photo of the user and their form of identification. This is then sent to the web application.

The web application uses the framework flask to communicate between the AWS Amazon Recognition and Amazon S3, with the MySQL Database. This part takes in the photos taken compares the face with the id and if there is a match of 98% or higher similarity with the AWS tools, it then checks if the person has already voted by checking if their face already appears in the MySQL Database. If not the vote is then casted and the users vote, face and id are stored in the database. [4]

#### Image Processing

Katie references the use of several image-processing libraries that are useful in python such as:

* OpenCV
* Pillow
* Scikit-image
* Pygame

These libraries are helpful for image inputs and the use of python also provides the capabilities for connecting with cloud services like:

* Amazon Web Services (AWS)
* Microsoft Azure
* Google Cloud Services

Katie goes on to use these services and OpenCV library to perform the following functions:

* Detect camera 1 and launch video stream
* Detect if users face is present with Haarcascade
* Capture face if detected and save it as a frame
* Detect camera 2 and launch video stream
* Capture frame and save it as live\_ID
* Setup connection to AWS Rekognition
* Perform face similarities checks
* Store face to Rekognition collection

Our project will perform similar functions such as using the Haarcascade to detect the users face. Although this is considered a type of machine learning, it is also a built-in function of OpenCV with already pre-trained data, this will be the only use of machine learning in our project. The other functions of using OpenCV to open a video stream and capture images as frames will also be used to perform feature analyse.

#### Strengths

Strengths of Katies project include:

* Leveraging use of OpenCV pre-trained models for facial detection Haarcascade
* Implementation of AWS frameworks for similarity checks
* Correct implementation of Model View Controller architecture
* Configuring the similarity check from 99-98% to account for low camera resolution

#### Weaknesses

Weaknesses identified through reading of this project:

* Lack of testing performed on this project (5 users were only tested)
* She stated the scalability of the system is limited
* Lack of security of the web application
* Amazon Rekognition performs the similarity checks instead of using image processing techniques

### Home Security Door Lock, Alarm and Facial Recognition Surveillance Camera

#### Description

The purpose of this project was to create a home security system that can be controlled from anywhere with an android device. The person can grant access to any individual from the use of their android device, the person can view through the camera on the system who is at the door and can decide to remotely unlock the door giving access to that person. A facial recognition system will also be implemented through the camera, allowing inhabitants to gain access to the property by simply standing in front of the camera. Once the face is recognised as an inhabitant of the home or facility the door will unlock.

#### Methodology

The methodology Sam has chosen to approach this project was the Agile methodology. His reasoning for this choice is the “methodology supports adaptive planning, continuous improvements and flexibility”. [5]

Sams reasonings for this approach are why agile has become one of the most adapted methodology approaches in project development today, it’s a reason why our team will incorporate a bit of agile into our project. But with our approach as stated before, we plan on moving through each stage of the life cycle of the project and completing it before moving forward to the next stage.

#### Architecture

The technical architecture of the Sam’s project involves multiple devices and databases all interacting with each other simultaneously. It begins with the mobile device communicating with the connect router. The device is under the client/server part of the architecture and uses Android Java programs to interact with the Webhost server of the application. The application then interacts with the PHP functions that call the User Pass Codes database and the Facial Images database. This then confirms the user’s identity and replies to the webhost server, which then communicates the users’ interactions to the router. The router then sends the interactions to the Arduino Printed Circuit Board which then controls the Camera, Door Lock and Security Alarm. These devices then send signals to the Arduino Circuit Board which then communicates the information back to the router and then to the webhost server which can then be interpreted on the user’s device allowing them to see the camera footage and be able to unlock their door and deactivate or activate their alarm.

#### Image Processing

After analysing the home security project report. The source of how its duology in which the Arduino camera identifies and recognises the face’s present are a C++ source code file found called **techbitarFaceDetection.cpp**. [6]

The facial recognition method used in the code relies on Haar Cascade Classifiers for detecting faces and eyes, a feature-based method for object detection in images. Here’s an overview:

1. Haar Cascade Classifiers

* Haar-like features: This method uses Haar-like features to detect objects (in this case, faces and eyes). These features are simple rectangular patterns that can capture light and dark areas of an image, such as edges and textures, which are common in facial features (e.g., the bridge of the nose or the contour of the eyes).
* Cascade of Classifiers: The algorithm works by applying a series of classifiers to different regions of an image. These classifiers are arranged in stages (a cascade), where each stage decides whether the region could potentially contain a face. If a region passes all stages, it’s classified as a face. This cascading approach allows for real-time detection since it quickly discards non-face regions.
* detectMultiScale(): This function is used to detect objects at multiple scales (different sizes) in the image, which is helpful for detecting faces of various sizes and distances from the camera. In this code, it’s used for both face and eye detection.

1. Grayscale Conversion and Histogram Equalization

* Before detecting faces, the image is converted to grayscale using cvtColor(). This reduces computational complexity because colour information is not necessary for face detection.
* equalizeHist() is used to normalize the image contrast, making it easier to detect features in varying lighting conditions.

1. Position Detection and Communication with Arduino

* Once a face is detected, its position (X, Y coordinates) is calculated and then communicated to an Arduino via a serial connection. While this code detects faces, it does not recognize specific faces (i.e., it doesn't compare the detected face against a database of known faces).

The facial recognition method in this system relies on Haar Cascade Classifiers for detecting faces and eyes, which is based on detecting Haar-like features across various scales in the image.

Another method that is within this project is the fact that this is actually not facial recognition but rather facial detection as the projects database would have contained the authorized users faces.

Our project will be implementing these facial identification features to be able, to isolate key features on faces, so that our functions can then proceed to analyse the features. Using the Grayscale conversion and Histogram might also be a viable option to regulate saturation of the image for more features to be identified in detail.

#### Strengths

The strengths identified of Sam’s project are as follows:

* Complex integration of multiple devices and services working together
* The use of Haarcascade for facial feature detection leveraging a pretrained machine learning algorithm
* Equalisation of histograms of images to detect features

#### Weaknesses

Weaknesses identified after researching this project are:

* Lack of image processing functions for identification
* Reliance on machine learning

## Research Paper - Face Detection Techniques: A Review

This paper covers a wide range of facial detection techniques and their uses. This paper covers Feature Based approaches and Image Based approaches when it comes to identifying and recognising a face. This review of the paper covers the research that was identified as beneficial to this group project. Although the paper covers the Image based approaches, these will not be discussed as this primarily focuses on the implementation of machine learning, instead of image processing techniques.

### Feature Based Approaches

Feature based approach tries extracting features from the images and match it against its knowledge of facial features.

### Active Shape Modelling

ASM automatically locates landmark points that define the shape of any statistically modelled object in an image e.g., eyes, lips, nose, mouth and eyebrows.  
There is a training stage for ASMs which is the building of a statistical facial model containing images with manually annotated landmarks. ASM can be broken down into 3 parts:

#### Snakes

**Snakes** are used to identify head boundaries. To achieve this a snake is first initialized at the proximity around a head boundary. Then look at the edges within that boundary and assumes the shape of head from that. The evolution of the snake is by minimizing an energy function,

Esnake = Einternal + Eexternal

Einternal and External are internal and external energy functions

Energy minimizing is done by optimizing the steepest gradient descent function.

**Internal energy** focuses on the intrinsic properties of the snake. Naturally shrinking or expanding.

**External energy** counteracts the internal energy and enables the contours to deviate from the natural shape and assume the shape of nearby features within the head boundary at a state of equilibria

**Snakes are not good for extracting non-convex features** [7]

#### Point Distribution Model

**PDM** focuses on representing shapes as vectors and applying standard statistical methods to them. These models learn allowable constellations of shape points from training examples and use principal components to build the PDM. [7]  
The use of PDMs in our project can help identify and mark out the facial features we are going to try analysing and provide feedback on.

#### Deformable Templates

Consider a priori of facial features to improve the performance of snakes. Based on narrow valley, edge, peak and brightness. Other than face boundary, salient feature extraction is a great challenge of face recognition.

**Salient features** are eyes, nose, mouth and eyebrows

E = Ev + Ee + Ep + Ei + Einternal

|  |  |
| --- | --- |
| Ev | External energy valley |
| Ee | External energy edges |
| Ep | External energy peak |
| Ei | External energy image brightness |
| Einternal | Internal energy |

[7]

### Low Level Analysis

#### Skin Colour base

The colour of a person’s skin is a very important feature of their face. It has many benefits such as:

* Faster processing than other facial features
* Is orientation invariant with certain lighting conditions
* Motion estimation easier cause translation model is only needed

It also has several problems as well:

* Ambient lighting can affect the image
* Object movement

**Simplest skin-colour algorithms** to be used for detecting skin pixels. Pixels for skin regions can be detected using a normalized colour histogram and can be normalized for changes in intensity on dividing by luminance.

#### RGB – RG Normalized Colour Conversion

Conversion of an [R, G, B] vector into an [r, g] vector of normalized colour.  
Normalization involves converting the RGB values into a relative proportion of red and green components, effectively making the colour more robust to lighting variations.  
By converting it, it simplifies the process of identifying skin tone regions.  
The problem with this method is it may fail when there are large skin regions in the image including arms or legs.

#### YCbCr Colour Space and Skin Detection

**YCbCr Colour Space** separates an image into **luminance (Y)** and **chrominance (Cb and Cr)** components. Skin tones tend to cluster within specific ranges of **Cb (blue difference chroma)** and **Cr (red difference chroma)** values.  
Using an algorithm to set Cr and Cb thresholds as [Cr1, Cr2] and [Cb1, Cb2] so if a pixel falls within those thresholds it’ll be classified as skin.  
Problem with this is it may fail using images that contain other skin regions besides the face.

#### Implementation Steps for Skin Detection

1. Classify the skin region using one of the following:
   1. RGB
   2. YCbCr
   3. HSV
2. Apply a threshold to mask out all non-skin regions
3. Draw a bounding box around the detected skin regions to extract the face from the image

### Feature Analysis

The purpose of feature analysis is to find structural features that exist even when the pose, viewpoint or lighting varies and then using those features to locate faces. Design for face localization.

#### Feature Searching:

Viola and Jones came up with an approach for object detection which minimized computation time while being very accurate at detection. The approach relies on the use of Haar-like features that are evaluated quickly through the use of a new image representation. Works of the integral image generating a large set of features and using boosting algorithms from AdaBoost to reduce the over complete set. The detector is applied in a scanning fashion on grey scaled images. There are three key supports:

* **Integral image** which allows the features used by our detector to be computed quickly.
* **Classifier** built using the AdaBoost learning algorithm to select a small number of critical visual feature from a large set of potential features.
* **Combining classifiers** in a cascade which allows background regions of the image to be quickly discarded while focusing on the face-like regions more.

Advantages:

* Admired for face detection in real time
* Uncompetitive detection speed
* Highly accurate
* Low false positive rate

Limitations:

* Long training time
* Limited head poses
* Do not detect black faces (Racist)

#### Local Binary Pattern

Technique is very effective to describe the image texture features. Used for **image retrieval**, **texture examination, face recognition, image segmentation**, etc. Is recently used to detect moving objects via background subtraction. In LBP **every pixel** **has a texture value**, which can be combined with target for tracking thermos graphic and monochromatic video.

Advantages:

* Describe Image texture
* Texture analysis, image retrieval, face recognition and image segmentation
* Detection of moving objects via background subtraction
* Tolerance against monotonic illumination changes

Limitations:

* Sensitive to small changes in face localization
* Using large local regions increases errors
* Insufficient for non-monotonic illumination changes
* Only used for binary and gray images

# Risks and Mitigation

This section will cover the potential risks that are associated with our group project. The risks will be assigned a priority level on their impact on the system. Additionally, the probability of occurrence will also be provided. Following the identification of risks, mitigation techniques will be discussed on what are team will implement into the development cycle to either prevent or reduce the likelihood of these risks occurring.

## Severity of Risks

**Impact Levels**

* **Negligible:** little to no effect
* **Low**: Minor inconvenience or minor data privacy issue. No significant harm/potential impact
* **Moderate**: Problematic, privacy concerns, biased outputs. Could harm user trust or cause regulatory issues.
* **High**: Significant issues, large scale data breach/ severe bias causing harm to certain user groups. Potential legal penalties
* **Catastrophic**: Catastrophic effect on individuals, widespread identity theft, public backlash, significant ethical violations, or severe legal consequences leading to program shutdown.

**Likelihood levels:**

* Highly Unlikely <11%
* Unlikely 11-40%
* Possible 41-60%
* 61-80%
* Highly Likely >91%

**Risk Assessment Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Likelihood/ Impact | Highly Unlikely (1) | Unlikely (2) | Possible (3) | Likely (4) | Highly Likely (5) |
| Negligible (1) | 1 |  |  |  |  |
| Low (2) |  | 4 | 6 |  |  |
| Moderate (2) |  |  | 9 |  |  |
| High (3) |  |  |  | 16 |  |
| Catastrophic (4) |  |  |  |  | 25 |

## Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk ID** | **Name of Risk** | **Description** | **Priority** | **Chance of Occurrence** |
| 1 | Lack of Scalability | The ability for a program to handle multiple users at once and be able to scale for future growth | Low | Unlikely |
| 2 | Slow Performance | The system runs at a barely runnable performance for execution of functions | Moderate | Highly Unlikely |
| 3 | Breach of Data | The system does not store the user's sensitive data properly in relation to GDPR and unauthorized users gain access | High | Likely |
| 4 | System incompatibility | The application is unable to run on the system for certain individuals | High | Unlikely |
| 5 | Can’t identify face | The application is unable to identify a user's face | High | Highly Unlikely |
| 6 | Malicious repurpose of code | The source code is repurposed in a negative way that was not the intended use of the code | High | Negligible |
| 7 | The feature identification fails | The application is unable to identify users features | Moderate | Unlikely |
| 8 | Low Accuracy | The application outputs are inaccurate to the description of the user | Low | Likely |
| 9 | Bias | The project can be bias to a specific demographic of users. | Moderate | Likely |

## Mitigation

1. Scalability: using an IaaS platform.
2. Performance: ensuring the machine running the application is sufficiently powerful.
3. Breach of Data: can be mitigated by not storing any user of the program, ensuring they are aware of how their data will be used, stored and processed.
4. System Incompatibility: can be mitigated by using a coding language that is compatible with all systems (python).
5. Unable to identify face: use of high-quality inputs could be implemented, using image processing techniques like noise reduction, sharpening kernels, adaptive thresholding for lighting adjustment or equalize histogram. Additionally, when the image is being extracted from a video, weighted images could be used to reduce motion blur.
6. Malicious repurpose of code: ensuring this program is only used within this college and assignment grading system.
7. Fails to identify features: Proper research on techniques and use of image processing
8. Low Accuracy: The need for high quality training data, if applicable, as we intend to not use machine learning as much as possible, can be rectified using pre-existing high quality, varied data sets.
9. Bias: can be addressed by use of various demographics for testing, and training if necessary.

# Conclusion

In summary, our research for our project provides unique approaches on how we can tackle this problem identifying and analysing facial features without the extensive reliance on machine learning technologies. This approach is to demonstrate our understanding and ability to implement image processing techniques.

Our research highlights both strengths and limitations of current solutions and projects within this domain. We reviewed similar work by corporations, final year projects, and research papers. By comparing them, our approach aims to demonstrate that traditional processing techniques can still offer practical applications in feature detection.

Additionally, the researched covered on the potential risks and mitigation procedures have prepared our group for implementation and development of the assignment.

Ultimately, this research has helped our understanding on the development of projects with image processing techniques and has now provided us with a plan on how we are going to tackle our goals of achieving our project with confidence.

# References

[1] “Visage Technologies | Showcase Demo.” Accessed: Oct. 25, 2024. [Online]. Available: https://visagetechnologies.com/HTML5/latest/Samples/ShowcaseDemo/ShowcaseDemo.html

[2] “Amazon Rekognition AWS.” Accessed: Nov. 01, 2024. [Online]. Available: https://aws.amazon.com/rekognition/

[3] “Karlo Bozic Passwordless Authentication FYP.” Accessed: Oct. 29, 2024. [Online]. Available: https://library.tudublin.ie/articles/5683285.6340/1.PDF

[4] “Katie Crowley Electronic Voting Facial Recognition.” Accessed: Oct. 31, 2024. [Online]. Available: https://library.tudublin.ie/articles/5683287.6341/1.PDF

[5] “Sam Hough Home Security Door Lock, Alarm and Facial Recognition Surveillance Camera.” Accessed: Nov. 01, 2024. [Online]. Available: https://library.tudublin.ie/articles/3243174.2433/1.PDF

[6] techbitar, “Face Detection and Tracking With Arduino and OpenCV,” Instructables. Accessed: Nov. 01, 2024. [Online]. Available: https://www.instructables.com/Face-detection-and-tracking-with-Arduino-and-OpenC/

[7] A. Kumar, A. Kaur, and M. Kumar, “Face Detection Techniques: A Review,” *Artif. Intell. Rev.*, vol. 52, Aug. 2019, doi: 10.1007/s10462-018-9650-2.