SECURITY MANAGEMENT SYSTEM

Project Work submitted in partial fulfillment of requirement for the award of degree of

Master of Computer Application

Submitted by

VIVEK KUMAR MISHRA
(18032030097 / 18SCSE2030084)
AMRESH PATHAK
(18032030080 / 18SCSE2030063)

Under the Supervision of

Mr. C. Vairavel (Assist. Prof.)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY

GREATER NOIDA – 201301

MAY 2020

Bonafide Certificate

I hereby certify that the work which is being presented in the project entitled, " **Security Management System**" , in partial fulfilment of the requirement for the award of degree of Master of Computer Application submitted in School of Computing Science and Engineering of Galgotias University, Gr. Noida, is an authentic record of my own work carried out under the supervision of **Mr. C. Vairavel** and refers other researcher's works which are duly listed in the reference section.

The matter presented in this project has not been submitted for the award of any other degree of this or any other university.

(Vivek Kumar Mishra)

(Amresh Pathak)

This is to certify that above statement made by the candidate is correct and true to the best of my knowledge.

Mr. C Vairavel

School of Computing science and engineering
Galgotias University

Countersigned by

Dr. Munish Shabarwal

PhD (Management), PhD (CS)

Professor & Dean

School of Computing science and engineering

Galgotias University

Abstract

The world is getting more and more advance day by day, we all relying on the technology; we want all the thing to be authentic or much security require. We all love technology and making it the part of our daily life. We are adapting the future and every new step into the technology taking us one step ahead into the world of future. We all are going to the future world of authentic and security. Our project is also one step into the world of security. With the increasing use of extensive IT systems for sensitive or safety-critical applications, the matter of IT security is becoming more important. In order to be able to make sensible decisions about security there is a need for measures and metrics for computer security.

There currently exist no established methods to assess the security of management systems. Identifying a person with an image has been popularized through the mass media. However, it is less robust to fingerprint or retina scanning. This report describes the face detection and recognition mini-project undertaken for the visual perception and autonomy module at Plymouth university. It reports the technologies available in the Open-Computer-Vision (OpenCV) library and methodology to implement them using Python. For face detection, Haar-Cascades were used and for face recognition Eigenfaces, Fisherfaces and Local binary pattern histograms were used. The methodology is described including ow charts for each stage of the system. Next, the results are shown including plots and screen-shots followed by a discussion of encountered challenges. The report is concluded with the authors' opinion on the project and possible applications.

Facial recognition has proven very effective in **security** critical environments, such as airports, border control areas, and high-**security** access control situations. ... Likewise, automated **facial recognition** is becoming a powerful business intelligence tool, beyond recognizing the specific identity of a person.

Facial recognition has proven very effective in **security** critical environments, such as airports, border control areas, and high-**security** access control situations. ... Likewise, automated **facial recognition** is becoming a powerful business intelligence tool, beyond recognizing the specific identity of a person.

Table of contents

1 Problem Statement

Issues related to security are intertwined in all areas of the life of an organization. Many of them are considered and documented organizational and provided within the respective area/subsystem of the management system of the organization. For example, information security management is detailed and reliably described in the international standards. And this is natural, considering the importance of this type of security and the fact that information technology lies at the core of almost all organizations. In the meantime, however, it is necessary to go a long way to reach the ultimate goal of improving security of the organizations by developing integrated management systems for business security. A Security Management System may be considered as that part of the overall management system, based manly of the quality management system, that provides the structure to enable identification of potential threats to an organization and which establishes, implements, operates, monitors, reviews and maintains all appropriate measures to provide assurance of the effective management of the associated security risks. A security management system, as with other management systems is based upon the model defined in ISO 9001:2008, Quality Management Systems –Requirements [3]. In a security risk-based, process-driven approach to security, the achievement of security objectives should start with a threat/security risk assessment. Having identified the security risks and planned mitigation measures, a security risk register may be established. The mitigation measures detailed in the security risk register are realized through resource management and security planning, thus arriving at a security solution (product), whether that is hard security measures, procedural requirements or a higher level security solution that supports strategic objectives, such as a crisis management strategy, or establishment of an intelligence gathering network.

A management system, as defined by ISO 9000:2005 [4] is a 'system to establish policy and objectives and to achieve those objectives'. In order to help in the achievement of those objectives, the system needs to be supported; that support comes in the form of approved standards and procedures. An effective management system should be such that it does not necessarily need a discipline expert to implement and manage it.

Teamwork is key in all disciplines, but arguably more so in security; the ability to stand in for each other seamlessly, be it stepping up or stepping down, and achieving coordination in operations through an understanding of each other’s' roles ensures and develops talent and progression.

In security terms, this translates as providing a 'best in class' support in all areas, and planning now to do the same in the future. Resource and strategy planning for new country entry well in advance is an example where security could sustain an outstanding performance

2 Features and Functionality

1 Connected through Wi – Fi.

2 Can be controlled from all over the world using internet.

3 Can be used to camera externally to connect the software.

4 Can be used to identified the person through the camera application

3 Software Required

1 PyCharm

2 Open Vision Classifiers

3 SQLlite3

4 Windows 10 OS

4 Hardware Required

1 Personal Computer Desktop or Laptop

2 Minimum Core i3 Processor at least 7^{th} generation

3 Minimum 512GB HDD or 265GB SSD

4 Camera 720p support (inbuild with your system)

Introduction

Face Detection & Recognition System is an application for automatically identifying or verifying a person from a digital image or a video frame. One of the ways to do this is by comparing selected facial features from the image and a facial database. This is a perfect way to empower web and desktop applications with face-based user authentication, automatic face recognition, and identification.
It is typically used in security systems and can be compared to other biometrics such as fingerprint or Iris recognition systems. Facial recognition use characteristics of the face to identify an individual.

The following document is a report on the mini project for Robotic visual perception and autonomy. It involved building a system for face detection and face recognition using several classifiers available in the open computer vision library (OpenCV). Face recognition is a non-invasive identification system and faster than other systems since multiple faces can be analyzed at the same time. The difference between face detection and identification is, face detection is to identify a face from an image and locate the face. Face recognition is making the decision "whose face is it? ", using an image database. In this project both are accomplished using different techniques and are described below. The report begins with a brief history of face recognition. This is followed by the explanation of HAAR-cascades, Eigenface, Fisherface and Local binary pattern histogram (LBPH) algorithms. Next, the methodology and the results of the project are described. A discussion regarding the challenges and the resolutions are described. Finally, a conclusion is provided on the pros and cons of each algorithm and possible implementations.

Security management can apply a systems theory approach, which develops and defines the security_management_plan using inputs, transformation within functions, and deliverable outputs. A system is considered to be an organized collection of components that integrate and operate at their optimum level, without decay. Inputs include strategic and tactical alignment, leadership, governance, accountability, ethics, culture, sustainability, and resilience. Security transformation comprises knowledge categories, such as security_risk management, business continuity, physical security, and personnel and technical security, with supporting management and business knowledge. Security managers should be a business manager first supported by security knowledge, with the ability to plan, organize, staff, lead, and control the security function.

LITERATURE REVIEW

Face recognition and face detection is an evolving area, it is changing and improving constantly. This section givesthe overview of various approaches and techniques along with their advantages anddisadvantages. Different approaches of face recognition can be categorized in three main groups such as holistic approach, feature-based approach, and hybrid approach with the help of these approach we can implement this these things.

Geometry Feature-based Approach

The geometry feature-based approach methods analyze local features such as nose, eyes andtheir geometric relationships. Sometimes this approach is known as only feature-basedapproach . Examples of this approach are Elastic bunch graph matching algorithm. This technique is not used now days.

Holistic Approach

Many researchers followed this approach. In the holistic approach whole face region is takeninto account as input data to the system. There are Various methods comes under this approach areeigenfaces, fisher faces, support vector machine, hidden Markova model (HMM) .

Hybrid-Approach

Under the hybrid approach the combination of local feature and whole feature is used.Modulareigenface, hybrid local feature methods are for hybrid approach. Human facialfeature plays important in face recognition. Many research and studies have determined that eyes mouth and nose are amongst the most significant feature for face recognition .

Face Recognition Techniques

This section gives the description of various techniques that are used by researchers and mostly apply on frontal faces. Various techniques are Eigen faces, neural network, fisher faces, and Genetic algorithm.

Eigenfaces

Eigenface is fast, simple, and practical method. First of all eigenfaces approach in FR was used by L. Sirovich and M. Kirby in 1986 . This method is based on the Principal Component Analysis and the goal was to represent an image in a lower dimension without losing much information, and then reconstructing it .This technique is later the foundation of the proposal of many new face recognition algorithms. In 1992 MathewTurk and Alex Pentland of the MIT presented a work which used eigenfaces for recognition. They used PCA projection as a feature vector to solve the problem of face recognition using Euclidean distance as a similarity function. The authors reported 96 percent, 85 percent,and 64 percent correct classifications averaged over lighting, orientation, and size variations,respectively.AlexPentland, BabackMoghaddam extended the work of eigenface to eigenfeaturecorresponding to eigencomponent. They uses modular approach with this they have been ableto demonstrate robustness to localized variations in object appearance.

Recently, K.Chang, K.W. Bowyer and S. Sarkar, experiments with ear and face recognition, usingthe standard principal component analysis approach , showed that the recognition performance is essentially identical using ear images or face images and combining the twofor multimodal recognition results is 90.9% ,71.6%, 70.5% respectively.

Fisherface

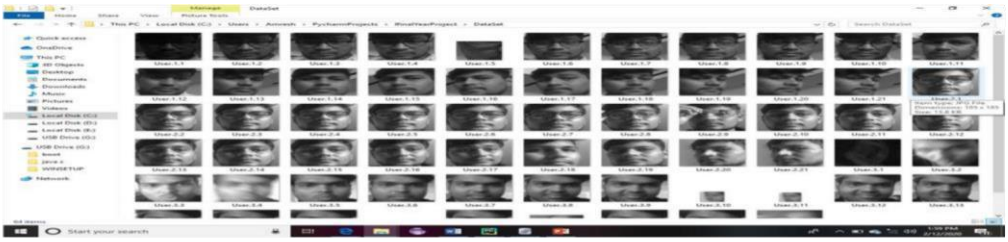
This Linear Discriminant is a "classical" technique in pattern recognition , firstdeveloped by Robert Fisher in 1936 for taxonomic classification . FLD (Fisher's Linear Discriminant) is an example of a *class specific method* , in the sense that it tries to get the shape the scatter in order to make it more reliable for classification. Belhumeur et al propose fisherfaces method by using PCA and Fisher‟s linear discriminant analysis.
The FLDremove the problem of eigenface method in pentland by taking the advantage of withinclass information, minimizing variation in same image due to lighting condition.

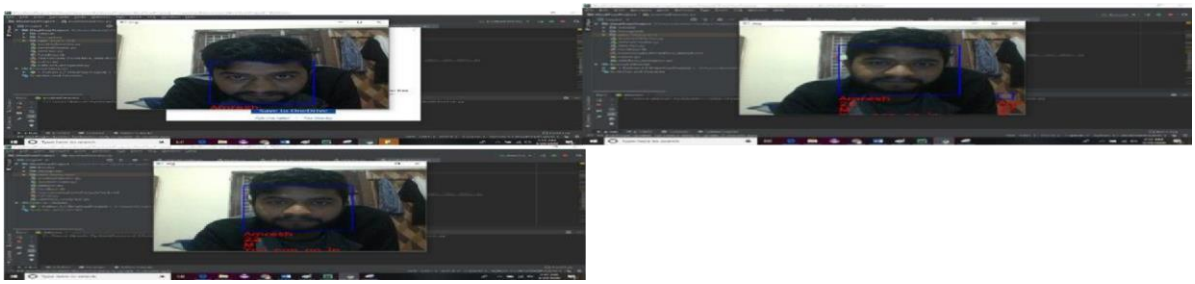
Face Detection using Haar-Cascades

Haar wavelet is a mathematical action that gives us a square-shaped waves create box shaped patterns to recognize signals with sudden transformations. An example is shown in gure 1. With the combination of several wavelets, a cascade can be created that can identify edges, lines and circles with different color intensities. These sets are used in Viola Jones face detection technique in 2001 and since then more patterns are introduced for object detection.

Using Haar cascades to analyze an image, a scale is selected smaller than the target image. It is then placed on the image, and the average of the values of pixels in each section is taken.

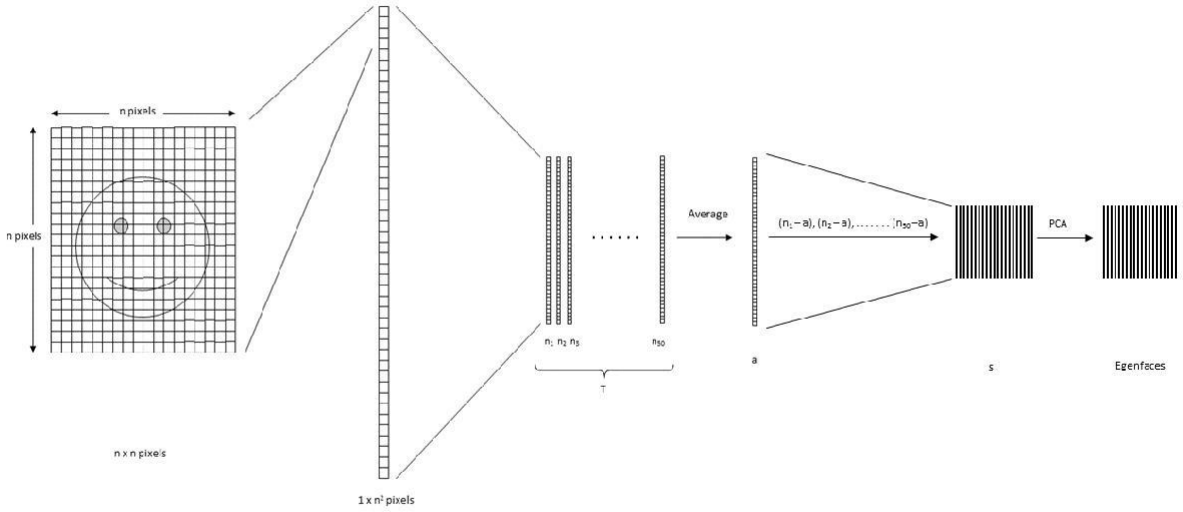
For face detection on a human face is performed by matching a combination of different Haar-like-features. Forehead, eyebrows and eyes contrast as well as the nose with eyes as shown below in figure A single classifier is not accurate enough. Several classifiers are combined as to provide an accurate face detection system as shown in the block diagram below in picture.





Eigenface

Eigenface is based on PCA that classify images to extract features using a set of images. It is important that the images are in the same lighting condition and the eyes match in each image. Also, images used in this method must contain the same number of pixels and in grayscale. For this example, consider an image with n x n pixels as shown in gure 4. Each raw is concatenated to create a vector, resulting a 1 n^2 matrix. All the images in the dataset are stored in a single matrix resulting a matrix with columns corresponding the number of images. The matrix is averaged (normalised) to get an average human face. By subtracting the average face from each image vector unique features to each face are computed. In the resulting matrix, each column is a representation of the di erence each face has to the average human face. A simpli ed illustration can be seen in gure 4.

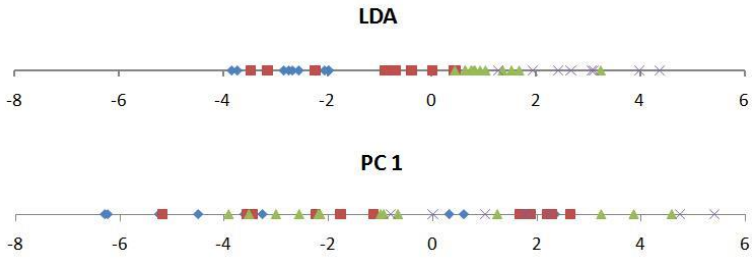


Pixels of the image are reordered to perform calculations for Eigenface

The next step is computing the covariance matrix from the result. To obtain the Eigen vectors from the data, Eigen analysis is performed using principal component analysis. From the result, where covariance matrix is diagonal, where it has the highest variance is considered the 1^{st} Eigen vector. 2^{nd} Eigen vector is the direction of the next highest variance, and it is in 90 degrees to the 1^{st} vector. 3^{rd} will be the next highest variation, and so on. Each column is considered an image and visualised, resembles a face and called Eigenfaces. When a face is required to be recognised, the image is imported, resized to match the same dimensions of the test data as mentioned above. By projecting extracted features on to each of the Eigenfaces, weights can be calculated. These weights correspond to the similarity of the features extracted from the di erent image sets in the dataset to the features extracted from the input image. The input image can be identi ed as a face by comparing with the whole dataset. By comparing with each subset, the image can be identi ed as to which person it belongs to. By applying a threshold detection and identi cation can be controlled to eliminate false detection and recognition. PCA is sensitive to large numbers and assumes that the subspace is linear. If the same face is analysed under di erent lighting conditions, it will mix the values when distribution is calculated and cannot be e ectively classi ed. This makes to di erent lighting conditions poses a problem in matching the features as they can change dramatically.

Fisherface

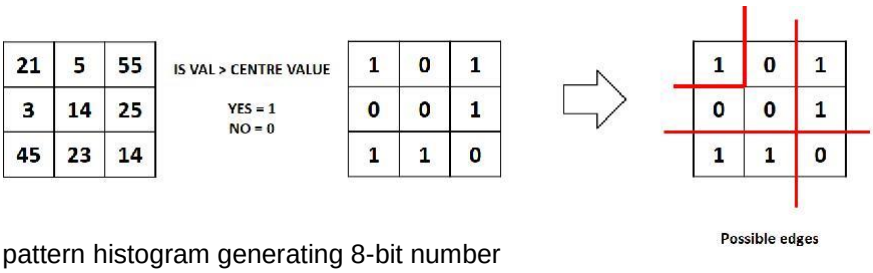
Fisherface technique builds upon the Eigenface and is based on LDA derived from Ronald Fishers' linear discriminant technique used for pattern recognition. However, it uses labels for classes as well as data point information [6]. When reducing dimensions, PCA looks at the greatest variance, while LDA, using labels, looks at an interesting dimension such that, when you project to that dimension you maximise the di erence between the mean of the classes normalised by their variance [6]. LDA maximises the ratio of the between-class scatter and within-class scatter matrices. Due to this, di erent lighting conditions in images has a limited e ect on the classi cation process using LDA technique. Eigenface maximises the variations while Fisherface maximises the mean distance between and di erent classes and minimises variation within classes. This enables LDA to di erentiate between feature classes better than PCA and can be observed in gure 5 [12]. Furthermore, it takes less amount of space and is the fastest algorithm in this project. Because of these PCA is more suitable for representation of a set of data while LDA is suitable for classi cation.



The first component of PCA and LDA. Classes in PCA looks more mixed than of LDA

Local Binary Pattern Histogram

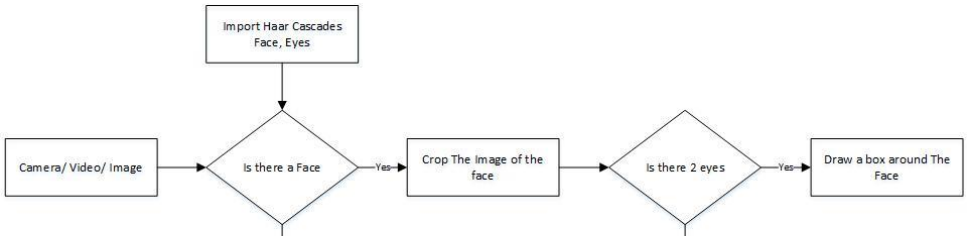
Local binary patterns were proposed as classi ers in computer vision and in 1990 By Li Wang [4]. The combination of LBP with histogram oriented gradients was introduced in 2009 that increased its performance in certain datasets [5]. For feature encoding, the image is divided into cells (4 x 4 pixels). Using a clockwise or counter-clockwise direction surrounding pixel values are compared with the central as shown in gure 6. The value of intensity or luminosity of each neighbour is compared with the centre pixel. Depending if the di erence is higher or lower than 0, a 1 or a 0 is assigned to the location. The result provides an 8-bit value to the cell. The advantage of this technique is even if the luminosity of the image



: Local binary pattern histogram generating 8-bit number

Face Detection

First stage was creating a face detection system using Haar-cascades. Although, training is required for creating new Haar-cascades, OpenCV has a robust set of Haar-cascades that was used for the project. Using face-cascades alone caused random objects to be identi ed and eye cascades were incorporated to obtain stable face detection. The owchart of the detection system can be seen in gure 8. Face and eye



The Flow chart of the face detection application

classi er objects are created using classi er class in OpenCV through the cv2.CascadeClassi er() and loading the respective XML les. A camera object is created using the cv2.VideoCapture() to capture images. By using the CascadeClassi er.detectMultiScale() object of various sizes are matched and location is returned. Using the location data, the face is cropped for further veri cation. Eye cascade is used to verify there are two eyes in the cropped face. If satis ed a marker is placed around the face to illustrate a face is detected in the location.

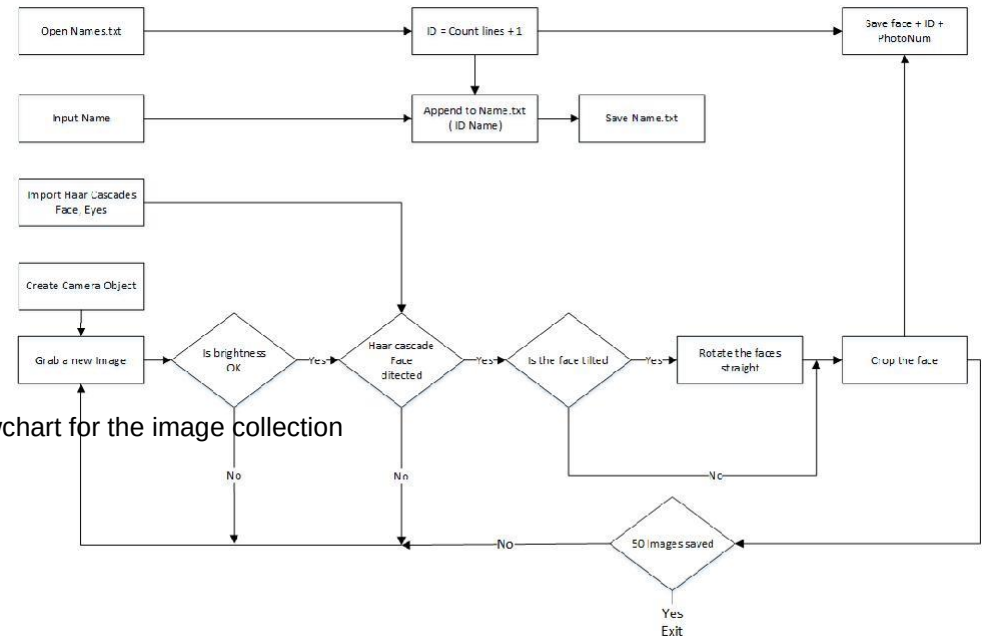
Face Recognition Process

For this project three algorithms are implemented independently. These are Eigenface, Fisherface and Linear binary pattern histograms respectively. All three can be implemented using OpenCV libraries. There are three stages for the face recognition as follows:

1. Collecting images IDs
2. Extracting unique features, classifying them and storing in XML les
3. Matching features of an input image to the features in the saved XML les and predict identity

Collecting the image data

Collecting classi cation images is usually done manually using a photo editing software to crop and resize photos. Furthermore, PCA and LDA requires the same number of pixels in all the images for the correct operation. This time consuming and a laborious task is automated through an application to collect 50 images with di erent expressions. The application detects suitable expressions between 300ms, straightens any existing tilt and save them. The Flow chart for the application is shown in gure 9.



The Flowchart for the image collection

:

Application starts with a request for a name to be entered to be stored with the ID in a text le. The face detection system starts the rst half. However, before the capturing begins, the application check for the brightness levels and will capture only if the face is well illuminated. Furthermore, after the face is detected, the position of the eyes are analysed. If the head is tilted, the application automatically corrects the orientation. These two additions were made considering the requirements for Eigenface algorithm. The Image is then cropped and saved using the ID as a lename to be identi ed later. A loop runs this program until 50 viable images are collected from the person. This application made data collection e cient.

Training the Classi ers

OpenCV enables the creation of XML les to store features extracted from datasets using the FaceRecognizer class. The stored images are imported, converted to grayscale and saved with IDs in two lists with same indexes. FaceRecognizer objects are created using face recogniser class. Each recogniser can take in parameters that are described below:

cv2.face.createEigenFaceRecognizer()

1. Takes in the number of components for the PCA for crating Eigenfaces. OpenCV documentation mentions 80 can provide satisfactory reconstruction capabilities.

2. Takes in the threshold in recognising faces. If the distance to the likeliest Eigenface is above this threshold, the function will return a -1, that can be used state the face is unrecognisable

cv2.face.createFisherfaceRecognizer()

1. The rst argument is the number of components for the LDA for the creation of Fisherfaces. OpenCV mentions it to be kept 0 if uncertain.

2. Similar to Eigenface threshold. -1 if the threshold is passed.

cv2.face.createLBPHFaceRecognizer()

1. The radius from the centre pixel to build the local binary pattern.

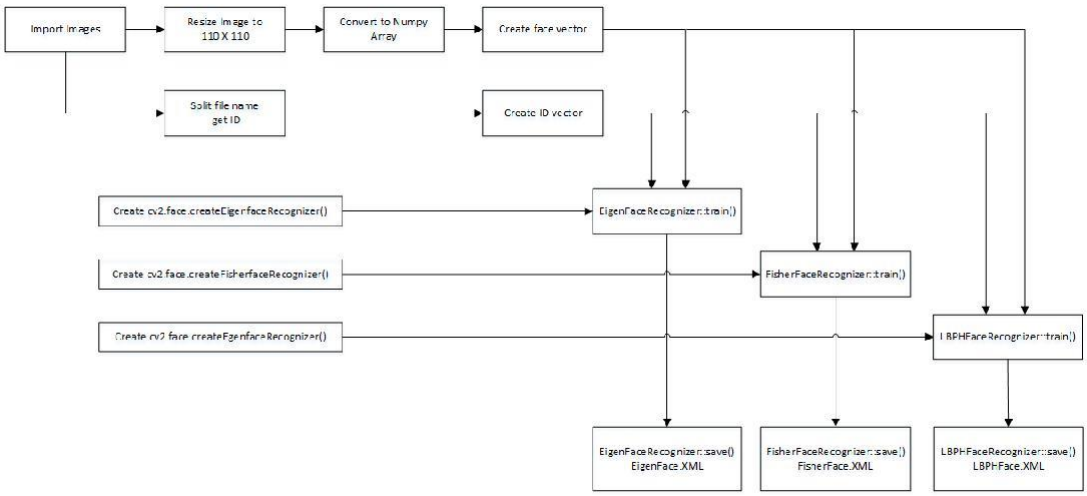
2. The Number of sample points to build the pattern. Having a considerable number will slow down the computer.

3. The Number of Cells to be created in X axis.

4. The number of cells to be created in Y axis.

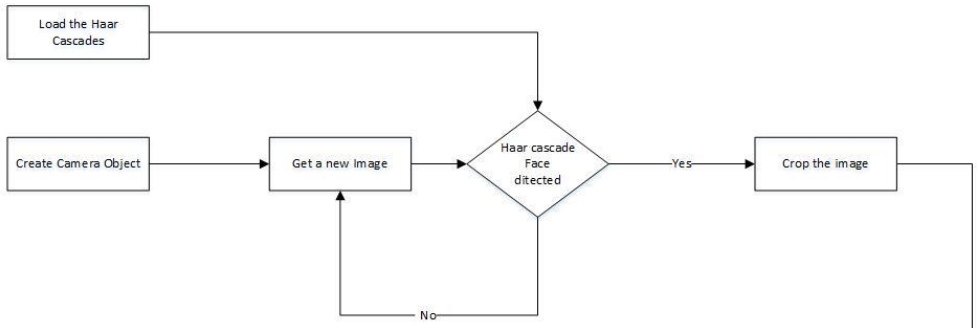
5. A threshold value similar to Eigenface and Fisherface. if the threshold is passed the object will return -1

Recogniser objects are created and images are imported, resized, converted into numpy arrays and stored in a vector. The ID of the image is gathered from splitting the le name, and stored in another vector. By using FaceRecognizer.train(NumpyImage, ID) all three of the objects are trained. It must be noted that resizing the images were required only for Eigenface and Fisherface, not for LBPH. Next, the con guration model is saved as a XML le using FaceRecognizer.save(FileName). In this project, all three are trained and saved through one application for convenience. The ow chart for the trainer is shown in gure 10.



The Face Recognition

Face recogniser object is created using the desired parameters. Face detector is used to detect faces in the image, cropped and transferred to be recognised. This is done using the same technique used for the image capture application. For each face detected, a prediction is made using FaceRecognizer.predict() which return the ID of the class and con dence. The process is same for all algorithms and if the con dence his higher than the set threshold, ID is -1. Finally, names from the text le with IDs are used to display the name and con dence on the screen. If the ID is -1, the application will print unknown face without the con dence level. The ow chart for the application is shown in gure 11.



the face as ID-17 and the rest are between ID-20 and ID-21, which is the same person. The change of

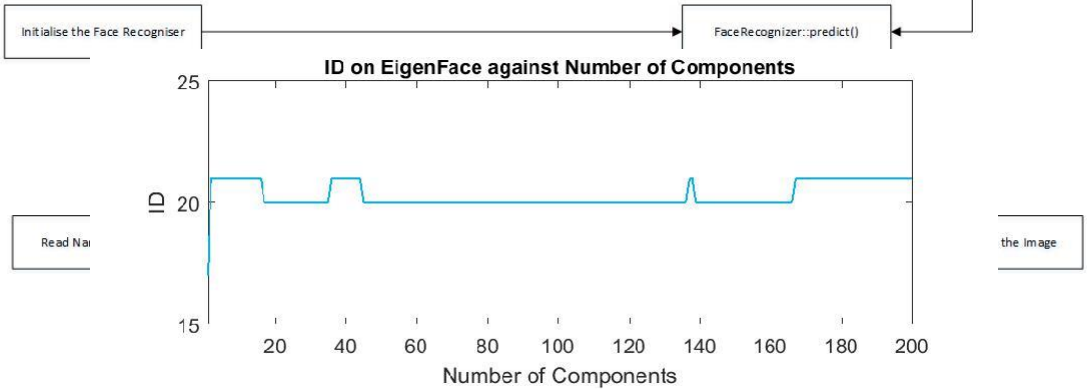
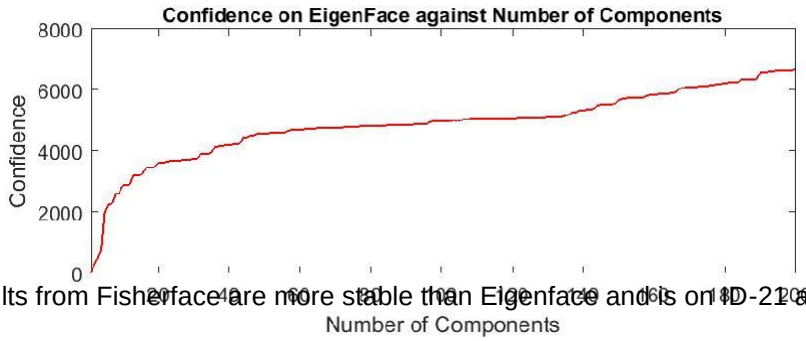


Figure 13: The ID from the face recogniser changes between two classes of the same person

Con dence is plotted in gure 14, increasing with components. From this plot it appears the best is when components are below 20.



The ID results from Fisherface are more stable than Eigenface and is on ID-21 as seen in gure 15.

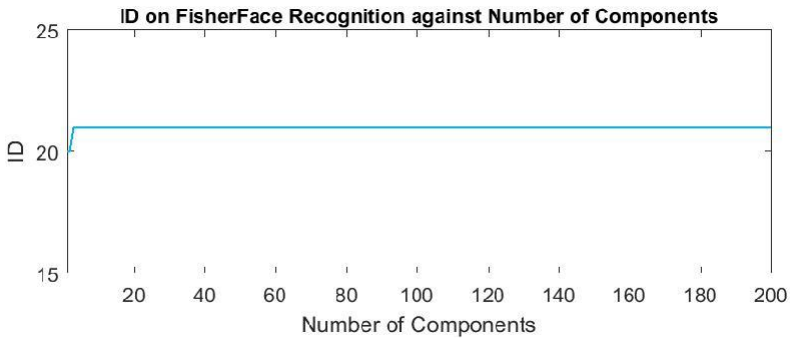


Figure 15: Fisherface ID is stable

Fisherface con dence increase in gure 16 until the number of components is 10 and will be used as the ideal value. LBPH has more than one parameter to change. All are incremented to the maximum

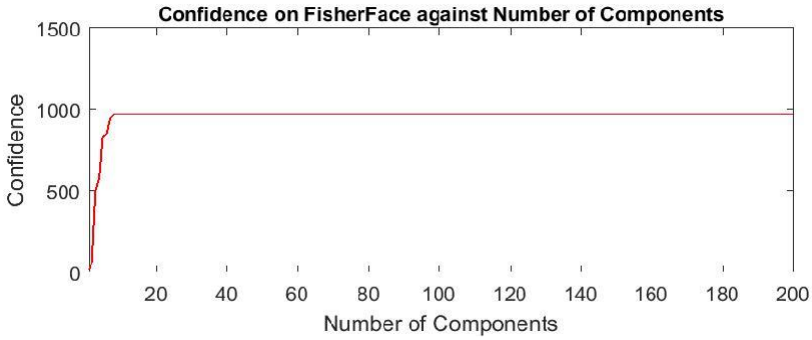


Figure 16: Stable con dence after 10 components

limit and the results are shown below. The rst is the radius from the centre pixel and since the image size is 110 X 110, maximum radius is 54. The ID is steady all the way to 50 as can be seen in gure 17.

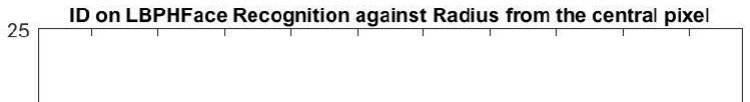


Figure 17: The ID returned from LBPH

Con dence level is graphed against the radius in gure 18. The con dence is uctuating after 40.

The lowest con dence level is at 2.

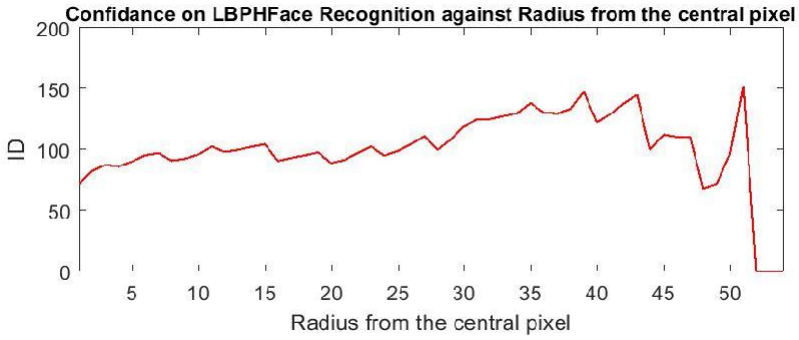


Figure 18: The con dence returned from LBPH

The number of neighbours are changed from 1 - 13. Further increase caused the computer to stall. The returned ID is plotted below in gure 19. ID steady until 9 neighbours and changed to ID-20.

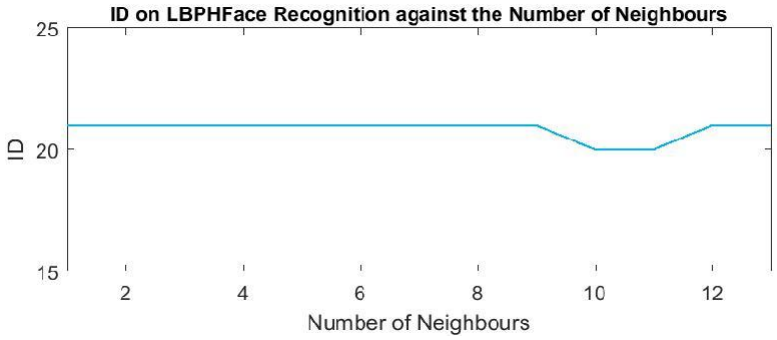
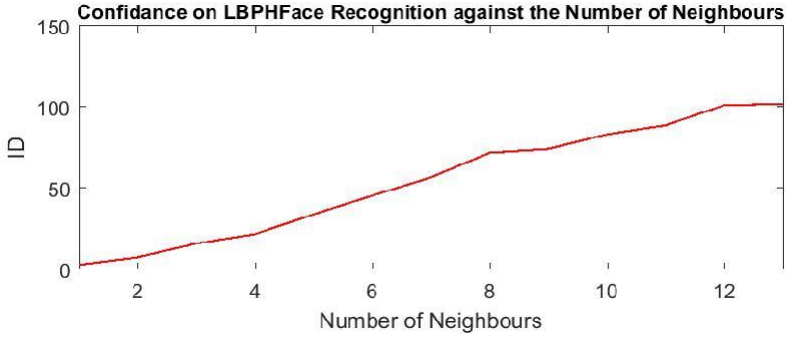


Figure 19: The ID returned from LBPH changing neighbours

The con dence continuously increased as can be seen in gure 20 and 1 neighbour will be included to the next test.



Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user

Example

```
#!/usr/bin/python

import Tkinter
top = Tkinter.Tk()
# Code to add widgets will go here...
top.mainloop()
```

This would create a following window −

Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table −

Sr.No.	Operator & Description
1	Button The Button widget is used to display buttons in your application.
2	Canvas The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
3	Checkbutton The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
4	Entry The Entry widget is used to display a single-line text field for accepting values from a user.
5	Frame The Frame widget is used as a container widget to organize other widgets.
6	Label The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
7	Listbox The Listbox widget is used to provide a list of options to a user.
8	Menubutton The Menubutton widget is used to display menus in your application.
9	Menu The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
10	Message The Message widget is used to display multiline text fields for accepting values from a user.

Discussion

Early attempts on Eigenface and Fisherface was disappointing since the LBPH alone recognised a face. By designing an application to test the algorithms, and after calibration with new data, both algorithms performed well. The tester applications also allowed accurate threshold settings. Another problem was people tilting their head when images taken for the data. This was xed with an application that identi es the locations of the eyes and rotate the image to correct the o -set. It was noticed that some early-stage images in the data-set di erent brightness's. To resolve this, before taking an image, the brightness was averaged to prevent dark images. These changes to the system improved the performance noticeably.

Conclusion

This paper describes the mini-project for visual perception and autonomy module. Next, it explains the technologies used in the project and the methodology used. Finally, it shows the results, discuss the challenges and how they were resolved followed by a discussion. Using Haar-cascades for face detection worked extremely well even when subjects wore spectacles. Real time video speed was satisfactory as well devoid of noticeable frame lag. Considering all factors, LBPH combined with Haar-cascades can be implemented as a cost e ective face recognition platform. An example is a system to identify known troublemakers in a mall or a supermarket to provide the owner a warning to keep him alert or for automatic attendance taking in a class.

References

[1] Takeo Kanade. Computer recognition of human faces, volume 47. Birkhauser Basel, 1977.

[2] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. Josa a, 4(3):519{524, 1987.

[3] M. Turk and A. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71{86, Jan 1991.