Artificially Intelligent Forensic Face Detection and Identification System Using Open Computer Vision and Deep Learning (AIFFD)

(A Case Study of Security in Kampala City Streets and Residences**)**

FACULTY OF SCIENCE AND TECHNOLOGY

A Project Report Submitted To The Faculty Of Science And Technology For A Study Leading To Partial Fulfillment Of The Requirements For The Award Of The Degree Of Bachelors In Software Engineering Of International University Of East Africa.

Faculty of Science and Technology: International University of East Africa

[OCTOBER, 2019]

DECLARATION

We, the undersigned, declare that to the best of our knowledge, this report is our original piece of work, and has never been published and/or submitted for any award in any other University or Higher Institution of Learning.

Name	Registration Number	Signature
MOYIGA MORIA NDURU	17/UG/572/BSSE-S	
KIYIMBA BILL	17/UG/584/BSSE-S	

Date: ………………..………………

APPROVAL

This report has been submitted with my approval as supervisor and my signature is appended here:

Signed……………………………………… Date: ………………………………

[MR. EDISON KAGONA]

International University of East Africa [IUEA]

DEDICATION

We dedicate this piece of work to our dear parents who basically sacrificed their lives to make certain that we got a head start. We also dedicate this report to our supportive lecturers who have been a source of incessant motivation and encouragement, and have always extended their unstinted support in writing this report and their endurance through the several months that we spent with them.

ACKNOWLEDGEMENT

We would like to thank our classmates, our parents who have enabled that we better ourselves as individuals by getting an education, the teaching staff of the International University Of East Africa most especially Madam lakshimi Babu,Mr Edison and Mr.watiti which of whom have helped, mentored and guided us successfully throughout this whole project.

Table of Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

CCTV	Closed Circuit Television
IDE	Integrated development Environment
IDLE	Integrated development and learning Environment
LBPH	Linear Binary Pattern Histogram
LDA	Lanten Dritchlet Allocation
ML	Machine learning
PCA	Principal Component Analysis
SVM	Hypertext Preprocessor
XML	Extensible markup language.

ABSTRACT

This research project aimed at designing and developing an artificially intelligent forensic face detection and identification system. The researchers used Design science approach as the methodology. System requirements were collected by using interviews and document reviews for the artificially intelligent forensic face identification system. Data was analysed using Microsoft word 2010. The findings from data analysis were used to design the core functionalities of the system. For system design the researchers used Microsoft Visio 2007 to model architectural designs and data flow diagrams that were used to develop the system. Implementation of the project the researchers used the following technologies: Python, OpenCV and IDL

As a result, the researchers developed a system to help combat the abnormal increase in crime rates and number of criminals has caused a great impact of insecurity in the nation. Crime prevention and criminal identification are the primary issue-challenges before security organizations because property and life protection are the basic concerns. Physical human security interventions are limited, hence the advent of security technology specifically cameras especially CCTV that have been installed in many public and private areas to ensure surveillance. Footage from the CCTV can be used to detect, Recognize &identify wanted criminals on scene. In this paper, an automated facial recognition system with a criminal database was proposed using known Haar feature-based cascade classifier. This system will be able to detect and recognize faces in real time. Accurate identification of faces is still a challenging task though the Viola-Jones framework has been widely used by researchers in order to detect the faces and objects in a given image. Face detection classifiers are shared by public communities, such as OpenCV, Tensor-Flow . Keywords: Criminal Identification; CCTV; facial recognition; Haar classifier; real-time; Viola-Jones; OpenCV.

CHAPTER ONE

INTRODUCTION

1 Introduction

This chapter presents the research background, problem statement solved by this research project, main and specific objectives of the study and the rational why this research was carried out.

1.1 Background to the Study

Face recognition is a biometric technique that entails identification of a given face. In recent years, face recognition has attracted much attention and its research has rapidly expanded by not only engineers but also neuroscientists and has potential application in computer vision, communication and automatic access control systems (R.Rajalakshmi, 2013). Face detection is an important part of recognition and its step of automatic face identification. However, face detection is not straightforward because it has lots of variations of image appearance, such as pose variation (front, non-front), occlusion, image orientation, illuminating conditions and facial expression (Sonia Ohlyan, 2013).

There are other biometric ways to identify individuals and their claimed identity and these are; fingerprint scanning, retina scan; DNA analysis etc. (Shailesh Wadhankar, 2018) says that "what makes face recognition different from other biometric techniques and the surreptitious answer is face recognition does not require an individual's participation for recognition". This holds the key for considering face recognition as the primary biometric technique for authentication and identification. Use cases for face recognition are spread across various domains which range from national security, cyber security to smart home security systems. In the past 10 to 15 years, the data generated from research has grown manifolds and so has the safety and security concerns of individuals' lives hence face recognition has been in consideration for quite some long time to address these concerns (PEI LI, 2019).

You face recognition is into two steps;

i. Face authentication/identification

ii. Face recognition

Face authentication involves a process of matching the queried face with a given dataset of images to authenticate identity. Face recognition is a category of biometric software that maps an individual's facial features mathematically and stores the data as a face dataset. The software uses deep learning algorithms to compare a live capture or digital image to the stored dataset in order to verify an individual's identity (TechTarget, 2019). In this paper, we have discussed a face recognition model based on OpenCv framework and a linear SVM classifier for webcam based face recognition.

China's facial recognition technology sector has found itself on a new development path: exporting artificial intelligence (AI) technologies to the vast land of Africa under the framework of the China-proposed Belt and Road initiative. Cloud Walk Technology Co, a domestic start-up based in Guangzhou, capital of South China's Guangdong Province, signed a strategic cooperation agreement with the Zimbabwean government in March on a mass facial recognition project, marking the entry of China's AI technology into Africa (Hongpe, 2018). The rollout of Huawei's smart CCTV technology into Uganda and Kenya is the latest example of China exporting its artificial intelligence to developing countries, particularly in Africa. Based on the recent news, (Tom Wilson, 2019) says that "Uganda police has confirmed that Chinese telecommunications giant Huawei is rolling out a massive surveillance system that uses facial recognition and other artificial intelligence software to fight crime in the central Africa country." The project, which includes the nationwide installation of the Huawei closed circuits television cameras and it, has been rolled out in more than 200 cities around the world. In Nairobi, the capital of the Neighboring Kenya, Huawei's small spherical cameras are ubiquitous in the city Centre, perched on slender lampposts at most intersections.

For the case of United States, there are no laws that specifically protect an individual's biometric data. Facial recognition systems are currently being studied or deployed for airport security and it is estimated that more than half the United States population has already had their faces captured. Data from a facial recognition system may be captured and stored, and an

Individual may not even know. The information could then be accessed by a hacker, and an individual's information spread without them ever knowing it. This data could be used by Government agencies or advertisers to track individuals as well. Even worse, a false positive may implicate an individual for a crime they are not guilty of. (Tom Wilson, 2019) Quotes Dorothy Mukasa, executive director of the unwanted witness, a non-profit body that advocates for uncensored online platforms in Uganda, says that "The concern is; we have no safeguards in Uganda. This is unregulated and there is lack of accountability and transparency when it comes to collection of personal data."

The human face is crucial for identity and it is the feature that best distinguishes a person. Face recognition is an interesting and challenging problem and impacts important applications in many areas such as identification for law enforcement, authentication for banking and security systems access, and personal identification among others (Peng Wu, 2019). Face recognition is an easy task for humans but it's entirely different task for a computer. Very little is known about human recognition to date on how to analyze an image and how the brain encodes faces and the inner features (eyes, nose, mouth) or outer features (head shape, hairline) which are used for successful face recognition. Neurophysiologist David Hubel and Torsten N. Wiesel (DAVID H, 2004) has shown that our brain has specialized nerve cells responding to specific local features of faces, such as lines, edges, angles or movement. Since we don't see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns.

Nowadays, face detection is used in many places especially the websites hosting images like Picassa, Photobucket and Facebook. The automatically tagging feature adds a new dimension to sharing pictures among the people who are in the picture and also gives the idea to other people about who the person in the image is. In this project, we have studied and implemented a pretty simple but very effective face detection, Identification & recognition algorithm based on deep learning that uses open computer vision as an additional forensic input to the CCTV cameras being used in the city and the different police surveillance cameras. The purpose & goal is to store criminal data for easy detection, recognition, verification & authentication. Data that is saved is used as input for forensic investigations and conviction of criminals.

1.2 Statement of the Problem

Pending and unsolved crime cases due to lack of concrete evidence has led to a pattern of criminal activities by the un-convicted which has caused an increase in the level of crime rates and a decrease of security in communities despite efforts to solve crime but lacking concrete evidence used for criminal conviction against committed offences. (Tom Wilson, 2019) Quotes Dorothy Mukasa, executive director of Unwanted Witness, a non-profit body that advocates for uncensored online platforms in Uganda, that "The concern is we have no safeguards in Uganda, unregulated and there is lack of accountability and transparency when it comes to collection of personal data." The face is crucial for human identity. It is the feature which best distinguishes a person. It is an easy task for humans but it's entirely a difficult task for a computer. Very little is known about human recognition to date on how to analyze an image and how the brain encode it and the inner features (eyes, nose, mouth) or outer features (head shape, hairline) used for a successful face recognition. Face recognition is interesting and challenging, impacts important applications in many areas such as identification for law enforcement, authentication for banking and security system access, and personal identification among others (Peng Wu, 2019).

Therefore, we have developed an artificially intelligent forensic face detection and identification system as an additional feature on the CCTV cameras installed around the country to support the governments and police departments at large in taking record, collection of evidence for criminal conviction against committed offences.

1.3 Purpose of the Study/ General Objective

This study aimed at designing and developing artificially intelligent forensic face detection and identification system that will be used by security organizations to control/regulate crime and to mitigate solution to the current pending and unsolved cases that lack criminal face input as evidence that is used for investigations for successful conviction of criminals

1.4 Specific Objectives of the Study

a) To study existing criminal detection, recognition and identification systems and to identify the requirements for the intelligent forensic face detection and identification system.
b) To design an intelligent forensic face detection and identification system for security agencies that will enhance justice and investigation activities within the Criminal investigation departments.
c) To implement, test and validate the intelligent forensic face detection and identification system in a real time environment.

1.5 Study Scope

1.5.1 Subject / Conceptual Scope

The system developed is limited to development of an artificially intelligent forensic face detection and identification system used by security agencies meant for criminal investigations. The system developed can be used by police or investigation departments to recognize criminals by their faces. The method of face recognition used is fast, robust, reasonably-simple and accurate with a relatively easily implementable and understandable algorithm and techniques

1.5.2 Geographical Scope

This study was carried out from data or requirements collected from existing systems and members of the community and security agencies meant for criminal investigations

1.5.3 Time Scope

The study was carried out in period of 3 and half months.

1.6 Justification/significance of the Study

This study was of importance to security investigators, the community and other researchers in the following ways:

a) Forensic face identification systems are essential in improving the security sector in Kampala and the system provides a software integration platform that will enhance accessibility for detection, identification and verification and can be trained to recognize faces, store faces in a database and perform real time reporting of criminals detected.

b) The system will help to ensure improvement of security in communities by enhancing the detection of faces that are linked to unsolved and pending cases hence serving justice.

c) The current on-going kidnaping situation in Kampala has made it difficult to find perpetuators easily hence the forensic recognition system saves time by enabling that kidnappers can be caught faster so that there is minimum damage to victims and emotional distress to the victims' families.

d) This Artificially intelligent system will reduce manual work of community & eye witness policing, maintaining records of wanted criminals by security personnel and the use of automation that reduces on costs and labour.

e) This system supports crowd image analysis that allows the easy detection of criminals among a vast number of citizens in Kampala

f) It can be installed on any operating system, it is light weight and can be used immensely applied and used to help in the reduction of crime

CHAPTER TWO

LITERATURE REVIEW

2 Introduction

This chapter presents the reviewed literature relevant to the study; this was done to give the researchers knowledge of the existing solutions and contributions of other researchers in order to identify the gaps in the existing solutions.

2.1 Computer Vision

Over the past decade face detection and recognition has transcended from esoteric to popular areas of research in computer vision which is one of the better and successful applications of image analysis and algorithm based understanding. A general statement of the face recognition problem (in computer vision) can be formulated as follows: given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces. Facial recognition generally involves two stages:

a) **Face Detection** where a photo is searched to find a face, then the image is processed to crop and extract the person's face for easier recognition.

b) **Face Recognition** where that detected and processed face is compared to a database of known faces, to decide who that person is.

Since 2002, face detection can be performed fairly easily and reliably with Intel's open source framework called OpenCV (Emami, 2010). This framework has an in- built Face Detector that works in roughly 90-95% of clear photos of a person looking forward at the camera. However, detecting a person's face when that person is viewed from an angle is usually harder, sometimes requiring 3D Head Pose Estimation. Also, lack of proper brightness of an image can greatly increase the difficulty of detecting a face, or increased contrast in shadows on the face, or maybe the picture is blurry, or the person is wearing glasses, etc. Face recognition however is much less reliable than face detection, with an accuracy of 30-70% in general. Face recognition has been a strong field of research since the 1990s, but is still a far away from a reliable method of user authentication. More and more techniques are being developed each year. The Eigen face technique is considered the simplest method of accurate face recognition, but many other (much more complicated) methods or combinations of multiple methods are slightly more accurate.

OpenCV was started at Intel in 1999 by Gary Bradski for the purposes of accelerating research in and commercial applications of computer vision in the world and, for Intel, creating a demand for ever more powerful computers by such applications. Vadim Pisarevsky joined Gary to manage Intel's Russian software OpenCV team. Over time the OpenCV team moved on to other companies and other Research. Several of the original team eventually ended up working in robotics and found their way to Willow Garage. In 2008, Willow Garage saw the need to rapidly advance robotic perception capabilities in an open way that leverages the entire research and commercial community and began actively supporting OpenCV, with Gary and Vadim once again leading the effort (Hewitt, 2010). Intel's open-source computer-vision library can greatly simplify computer- vision programming. It includes advanced capabilities - face detection, face tracking, face recognition, Kalman filtering, and a variety of artificial- intelligence (AI) methods - in ready-to- use form. In addition, it provides many basic computer-vision algorithms via its lower-level APIs.

OpenCV has the advantage of being a multi-platform framework; it supports both Windows and Linux, and more recently, Mac OS X.

OpenCV has so many capabilities though seems overwhelming at first. A good understanding of how these methods work is the key to getting good results when using OpenCV. Fortunately, only a selected few need to be known beforehand to get started. OpenCV functionality that will be used for facial recognition is contained within several modules. Following is a short description of the key namespaces: **CXCORE** namespace contains basic data type definitions, linear algebra and statistics methods, the persistence functions and the error handlers. Somewhat oddly, the graphics functions for drawing on images are located here as well.

i. **CV** namespace contains image processing and camera calibration methods. The computational geometry functions are also located here.

ii. **CVAUX** namespace is described in OpenCV's documentation as containing obsolete and experimental code. However, the simplest interfaces for face recognition are in this module. The code behind them is specialized for face recognition, and they're widely used for that purpose.

iii. **ML** namespace contains machine- learning interfaces.

iv. **High GUI** namespace contains the basic I/O interfaces and multi-platform windowing capabilities.

v. **CVCAM** namespace contains interfaces for video access through DirectX on 32- bit Windows platforms.

Eigenfaces is considered the simplest method of accurate face recognition, but many other (much more complicated) methods or combinations of multiple methods are slightly more accurate. Most resources on face recognition are for basic Neural Networks, which usually don't work as well as Eigenfaces does. And unfortunately there are only some basic explanations for better type of face recognition than Eigen faces, such as recognition from video and other techniques at the Face Recognition Homepage (Face Recognition Homepag, 2019) or 3D Face Recognition Wikipedia page (Wikipedia, Three-dimensional face recognition,, 2019) .But for other techniques, When you read some recent computer vision research papers from CVPR and other computer vision conferences. Most computer vision or machine vision conferences include new advances in face detection and face recognition that give slightly better accuracy.

2.2 face recognition

The following sections describe the face recognition algorithms; Eigenface, Fisherface, Local binary pattern histogram and how they are implemented in OpenCV.

2.2.1 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 eye 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 figure 4. Each raw is concatenated to create a vector, resulting a 1 x 2 matrix. All the images in the dataset are stored in a single matrix resulting to a matrix with columns in line with the number of images. The matrix is averaged (normalized) 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 difference each face has to the average human face. A simplified illustration can be seen in figure 1.

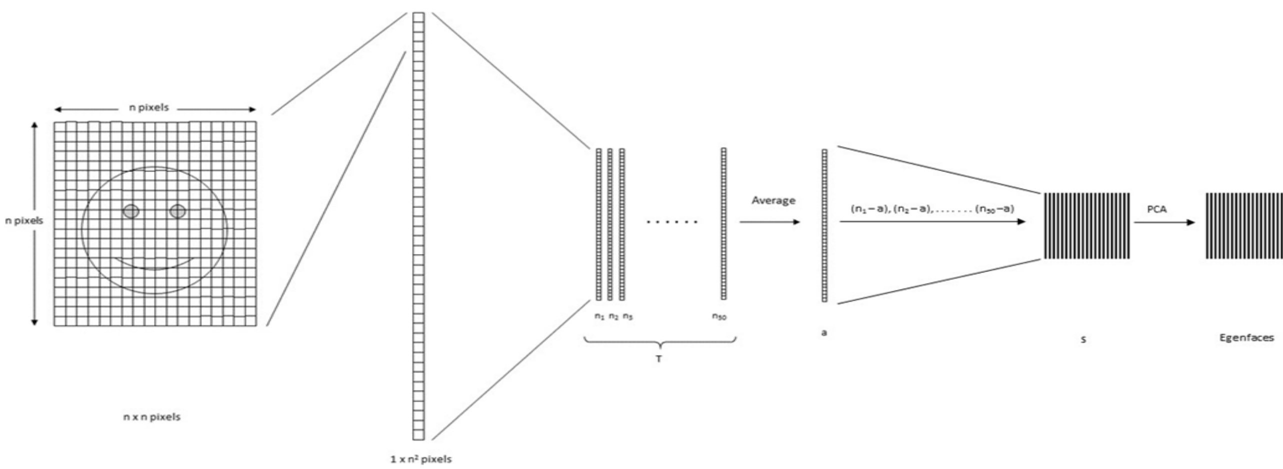


Figure 1 : Pixels of the image are recorded to perform calculations for Eigenface

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 1st Eigen vector. 2nd Eigen vector is the direction of the next highest variance, and it is in 90 degrees to the 1st vector. 3rd will be the next highest variation, and so son. Each column is considered an image and visualized, resembles a face and called Eigenface. When a face is required to be recognized, the image is imported, resized to match the same dimensions of the test data as mentioned above. By projecting extracted features on the tech of the Eigenface, weights can be calculated. These weights correspond to the similarity of the features on to each of the Eigenface, weights can be calculated. These weights correspond to the similarity of the features extracted from the different image sets in the datasets to the features extracted from the input image. The input image can be identified as a face by comparing with the whole dataset. By comparing with each subset, the image can be identified as to which person it belongs to. By applying a threshold detection and identification 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 analyzed under different lighting conditions, it will mix the values when distribution is calculated and cannot be effectively classified. This makes to different lighting conditions poses a problem in matching the features as they can change dramatically.

a) Haar features

A simple rectangular Haar-like feature can be defined as the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the original image. This modified feature set is called 2- rectangle feature. Viola and Jones also defined 3-rectangle features and 4-rectangle features. Faces are scanned and searched for Haar features of the current stage. The weight and size of each feature and the features themselves are generated using a machine learning algorithm from AdaBoost (Viola, 2001) (Peng Wu, 2019). The weights are constants generated by the learning algorithm

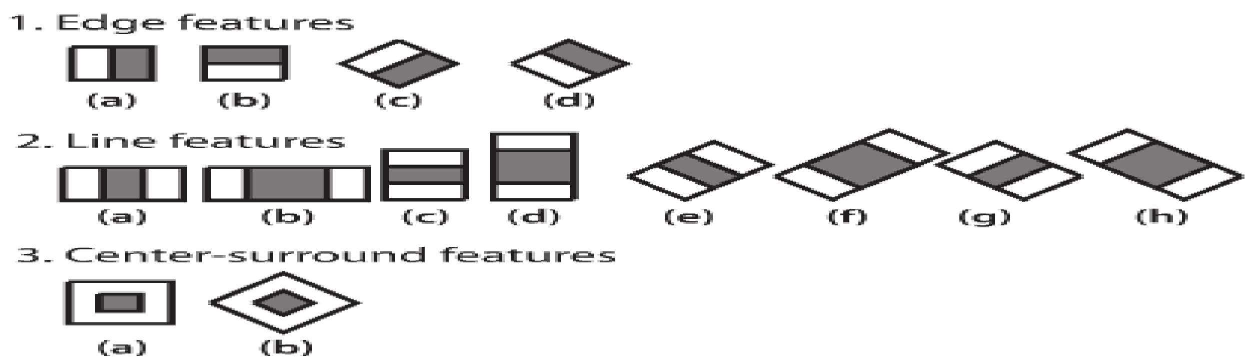


Figure 2: There's a variety of forms of features as seen above

Each Haar feature has a value that is calculated by taking the area of each rectangle, multiplying each by their respective weights, and then summing the results (Brunelli, 1993). The area of each rectangle is easily found using the integral image. The coordinate of the any corner of a rectangle can be used to get the sum of all the pixels above and to the left of that location using the integral image. By using each corner of a rectangle, the area can be computed quickly as denoted by "Figure 4". Since A is subtracted off twice it must be added back on to get the correct area of the rectangle.

The area of the rectangle R, denoted as the rectangle integral, can be computed as follows using the locations of the integral image: C + A - B - D

Figure 3: Examples of Haar features. Areas of white and black regions are multiplied by their respective weights and values added

2.2.2 Local Binary Pattern Histogram

Local binary patterns were proposed as classifiers in computer vision and I 1990 by Li Wang (Dong chen He, 1990). The combination of LBP with histogram orientated gradients was introduced in 2009 that increased its performance in certain datasets (T. X. Han, 2009). For feature encoding the image is divided into (4 x 4). Using a clockwise or counter-clockwise direction surrounding pixel values are compared with the central as shown in figure 6. The value of intensity or luminosity of each neighbor is compared with the Centre pixel. Depending if the difference 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 is changed as figure 7, the result is same as before. Histograms are used in large cells to find the frequency of occurrences of the values making process faster. By analyzing the result in the cell, edges can be detected as the values change. By computing the values of all cells and concatenating the histograms, feature vectors can be obtained. Images can be classified by processing with an ID attached. Input images are classified using txhe same process and compared with the dataset and distance is obtained. By setting up a threshold, it can be identified if it is a known or unknown face. Eigenface and Fishface compute the dominant features of the whole training set while LBPH analyze them individually.

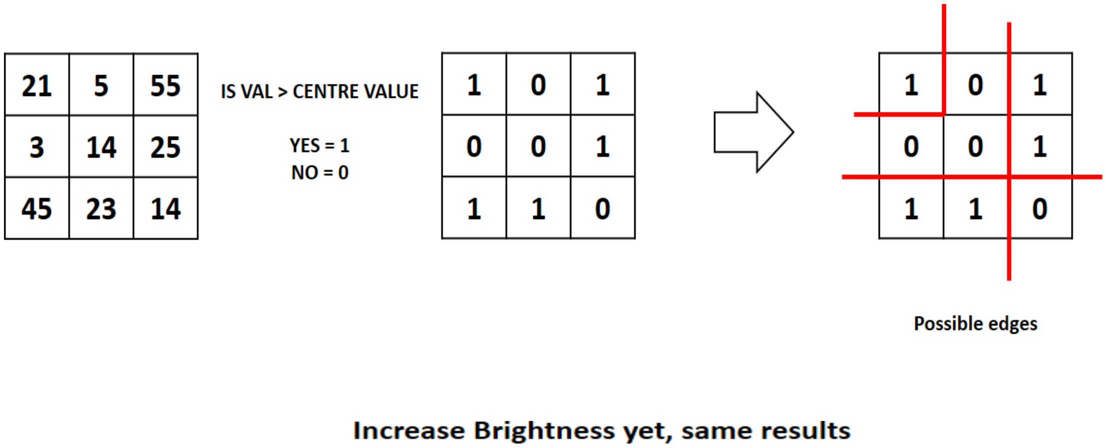


Figure 4: Local Binary Pattern Histogram generating 8-Bit number

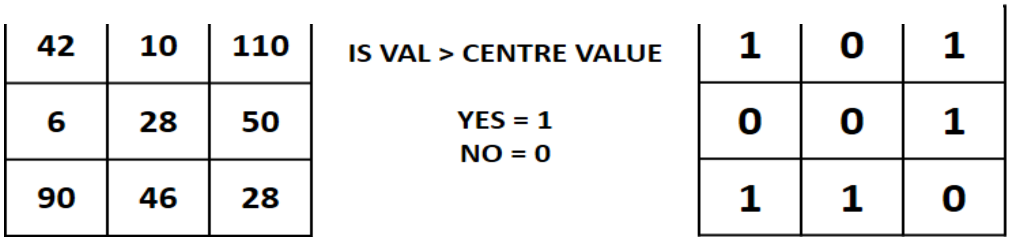


Figure 5 : The results are same even if brightness is changed

2.2.3. Face Detection using Haar classifier algorithm

The face detection algorithm proposed by Viola and Jones is used as the basis of our design. The face detection algorithm looks for specific Haar features and not pixels of a human face (Paul Viola, 2001). When one of these features is found, the algorithm allows the face candidate to pass to the next stage of detection. A face candidate is a rectangular section of the original image which is called as a sub-window. Generally, these sub windows have a fixed size (typically 24×24 pixels). This sub-window is often scaled in order to obtain a variety of different size faces. The algorithm scans the entire image with this window and denotes each respective section a face candidate (Viola, 2001).

a) Integral Image

The integral image is defined as the summation of the pixel values of the original image.

The value at any location (X, Y) of the integral image is the sum of the image's pixels above and to the left of location (X, Y). "Figure 1" illustrates the integral image generation

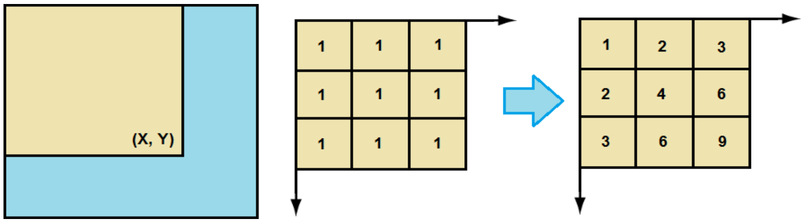


Figure 6: Integral image generation

The dark region represents the sum of the pixels up to position (X, Y) of the original image. It shows a 3×3 image and its corresponding integral image representation

b) Haar Feature Classifier

The cascade classifier contains a list of stages, where each stage consists of a list of weak learners. The system detects the required object by moving a window over the image. Each stage of the classifier labels the specific region defined by the current location of the window as either positive or negative where positive means that an object was found and negative means that the specified object was not found in the image (Wikipedia, /Cascading_classifiers, 2019).

If the labeling yields a negative result, then the classification of that particular region is over and the location of the window is moved to the next location. If the labeling gives a positive result, then the region moves to the next stage of classification. The classifier yields a final result as positive, when all the stages, including the last one, yield a positive result, which implies that the required object is found in the image (Wikipedia, /Cascading_classifiers, 2019)

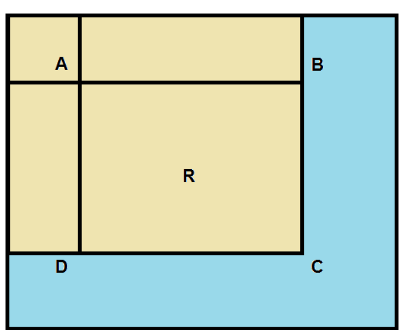


Figure 7: Calculating the area of a rectangle R is done using the corner of the rectangle: C + A - B - D.

c) Cascade

The Viola and Jones face detection algorithm eliminates face candidates quickly using a cascade of stages. The cascade eliminates candidates by making stricter requirements in each stage with later stages being much more difficult for a candidate to pass. Candidates exit the cascade if they pass all stages or fail any stage. A face is detected if a candidate passes all stages. This process is shown in “Figure 8”.

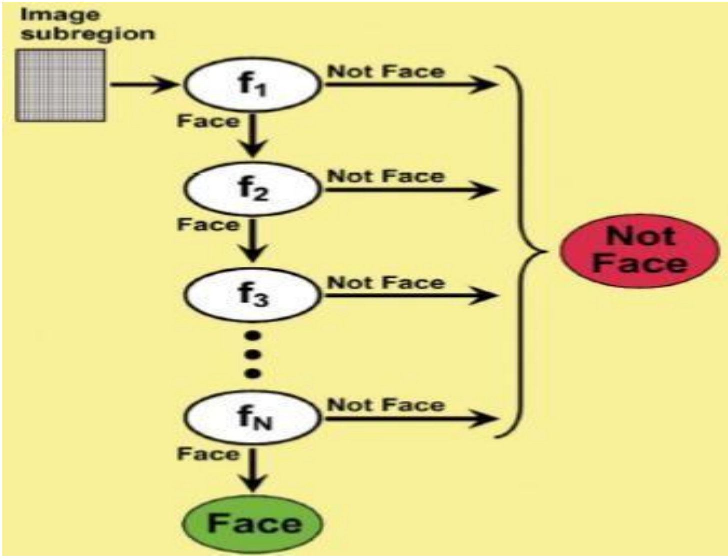


Figure 8: Cascade of stages. The face must pass all stages in the cascade to be concluded.

When image quality is taken into consideration, there is a plethora of factors that influence the system’s accuracy. It is extremely important to apply various image pre-processing techniques to standardize the images that you supply to a face recognition system. Most face recognition algorithms are extremely sensitive to lighting conditions, so that if it was trained to recognize a person when they are in a dark room, it probably won’t recognize them in a bright room, etc. This problem is referred to as "illumination dependent", and there are also many other issues, such as the face should also be in a very consistent position within the images (such as the eyes being in the same pixel coordinates), consistent size, rotation angle, hair and makeup, emotion (smiling, angry, etc.), position of lights (to the left or above, etc.). This is why it is so important to use a good image preprocessing filters before applying face recognition. You should also do things like removing the pixels around the face that aren't used, such as with an elliptical mask to only show the inner face region, not the hair and image background, since they change more than the face does.

2.3 Proposed Solution

OpenCV uses a type of face detector called a Haar Cascade classifier. Given an image, which can come from a file or from live video, the face detector examines each image location and classifies it as "Face" or "Not Face." Classification assumes a fixed scale for the face, say 50x50 pixels. Since faces in an image might be smaller or larger than this, the classifier runs over the image several times, to search for faces across a range of scales. This may seem an enormous amount of processing, but thanks to algorithmic tricks, classification is very fast, even when it's applied at several scales. The classifier uses data stored in an XML file to decide how to classify each image location. The OpenCV download includes four flavors of XML data for frontal face detection, and one for profile faces. It also includes three non-face XML files - one for full body (pedestrian) detection, one for upper body, and one for lower body.

You'll need to tell the classifier where to find the data file you want it to use. The one we used is called haarcascade_frontalface_default.xml. In OpenCV version 1.0, it's located at [OPENCV_ROOT]/data/haarcascades/haa rcascade_frontalface_default.xml where [OPENCV_ROOT] is the path to your OpenCV installation. For example, if you're on Windows XP, and you selected the default installation location, you'd use [OPENCV_ROOT] = "C:/Program Files/OpenCV" (If you're working with an older, 16-bit version of Windows, you'd use '\' as the directory separator, instead of '/'). It's a good idea to locate the XML file you want to use, and make sure your path to it is correct, before you code the rest of your face-detection program. It is very easy to use a webcam stream as input to the face recognition system instead of a file list. Basically you just need to grab frames from a camera instead of from a file, and you run forever until the user wants to quit, instead of just running until the file list has run out.

2.4 Conclusions

To improve the detection & recognition, many considerations & techniques can be implemented. For example, you could add colour processing, edge detection, etc. You can usually improve the face recognition accuracy by using more input images, at least 50 per person, by taking more photos of each person, particularly from different angles and lighting conditions. If you can't take more photos, there are several simple techniques you could use to obtain more training images, by generating new images from your existing ones:

a) You could create mirror copies of your facial images, so that you will have twice as many training images and it won't have a bias towards left or right.

b) You could translate or resize or rotate your facial images slightly to produce many alternative images for training, so that it will be less sensitive to exact conditions.

c) You could add image noise to have more training images that improve the tolerance to noise.

It is important to have a lot of variation of conditions for each person, so that the classifier will be able to recognize the person in different lighting conditions and positions, instead of looking for specific conditions. But it's also important to make sure that a set of images for a person is not too varied, such as if you rotated some images by 90 degrees. This would make the classifier to be too generic and also give very bad results, so if you think you will have a set of images with too much variance (such as rotation more than 20 degrees), then you could create separate sets of training images for each person. For example, you could train a classifier to recognize "John Facing Forward" and another one for "John F acing Left" and other ones "Mary Facing Forward", "Mary Facing Left", etc. Then each classifier can have a lot of variance but not too much, and you simply need to associate the different classifiers for each person with that one person (i.e.: "John" or "Mary").

That's why you can often get very bad results if you don't use good preprocessing on your images. As I mentioned earlier, Histogram Equalization is a very basic image preprocessing method that can make things worse in some situations, so you will probably have to combine several different methods until you get decent results. And something important to understand is that at the heart of the algorithm, it is matching images by basically doing the equivalent of subtracting the testing image with a training image to see how similar they are. This would work fairly well if a human performed it, but the computer just thinks in terms of pixels and numbers. So if you imagine that it is looking at one pixel in the test image, and subtracting the gray scale value of that pixel with the value of the pixel in the EXACT same location of each training image, and the lower the difference then the better the match. So if you just move an image by a few pixels across, or use an image that is just a few pixels bigger or has a few more pixels of the forehead showing than the other image, etc., then it will think they are completely different images! This is also true if the background is different, because the code doesn't know the difference between background and foreground (face), which is why it’s important to crop away as much of the background as you can, such as by only using a small section inside the face that doesn't include any background at all.

Since the images should be almost perfectly aligned, it actually means that in many cases, using small low-res images (such as by shrinking the images to thumbnail size) can give better recognition results than large hi-res images. Also, even if the images are perfectly aligned, if the testing image is a bit brighter than the training image then it will still think there is not much of a match. Histogram Equalization can help in many cases but it can also make things worse in other cases, so differences in lighting is a difficult & common problem. There are also issues such as if there was a shadow on the left of the nose in the training image and on the right in the test image then it will often cause a bad match, etc. That's why face recognition is relatively easy to do in real-time if you are training on someone and then instantly trying to recognize them after, since it will be the same camera, and background will be the same, their expressions will be almost the same, the lighting will be the same, and the direction you are viewing them from will be the same. So you will often get good recognition results at that moment. But once you try to recognize them from a different direction or from a different room or outside or on a different time of the day, it will often give bad results!

So it’s important to do a lot of experimentation if you want better results, and if you still can't get good results after trying many things, perhaps you will need a more complicated face recognition algorithm than PCA (Eigen faces), such as 3D Face Recognition or Active Appearance Models, mentioned below.

CHAPTER THREE

RESEARCH METHODOLOGY

3 Introduction

This chapter presents the framework that the researchers used to achieve the objectives of this study. It discusses the research method used, the data collection methods and analysis techniques that were deployed to analyze the data, design, implement, and test the system.

3.1 Research Method

This section specifies the fact finding strategies and tools used to develop the system. This phase will involve an interaction with different security teams to study the existing video surveillance systems. Understand the requirements of the new system or use other existing systems to gather requirements. I will interact with different stakeholders of video security surveillance systems, the end users, and security agencies to gather inputs, and relevant content that will be used for development. We shall study system analysis and design of other application systems and collect information in comparison with our current system through carrying out interviews and observation.

3.2 Requirements collection methods

The researchers used the following techniques to gather data:

3.2.1 Document review

Annual, quarterly and monthly reports plus data collection forms were reviewed to enable the researchers understand how the processes are handled and how they relate to each other. This technique gave the researchers an opportunity to analyze the current or existing system.

3.2.2 Interviews

Interview techniques were used to gain more information about the system which enabled us to acquire requirements for the current system. This involved both structured and unstructured verbal communication between interviewers and the respondents, during which data was obtained. Interviews have a greater amount of flexibility, there is clarification of the problem and they form a basis of developing questionnaires. This method though is time consuming and expensive.

3.3 Data Analysis

This involved inspecting, cleaning and transforming data with the goal of highlighting useful information, suggesting conclusions, and supporting decision making. The researchers used MS word to analyse data and come up with conclusions upon which the system's requirements were determined.

3.4 System Analysis and Design

The researchers used Structured System Analysis and Design approach (Indulska, 2008) to understand, describe system developments, and define user needs specific to user views. This helped to show relationships between different functions of the system and how they relate to each other. Microsoft Visio was be used to construct the Context Diagram and Data Flow Diagram. The context diagram was be used to show system boundary, users with their inputs to and outputs from the system. Data flow diagrams were used to graphically show all the main requirements for the system (Indulska, 2008).

3.5 System Design Approach

The researchers used prototyping design approach (Mcclendon, 2012) to design and develop the solution of this study, as it uses multiple iterations of requirements gathering, analysis, design, development, and testing of prototype as shown in the figure below (Todd, 2013). After completion of each iteration, the result is analyzed by the researchers. Their response creates the next level of requirements and defines the next iteration and this goes on until the operational system is ready. The researchers chose this approach because according to Todd (2013), it suits rapidly changing or poorly understood requirements, so it suits a project in which the development areas are not well known to the developers, provides the end-user with clear signs of progress and gives the end-user a sense of control over the project's progress.

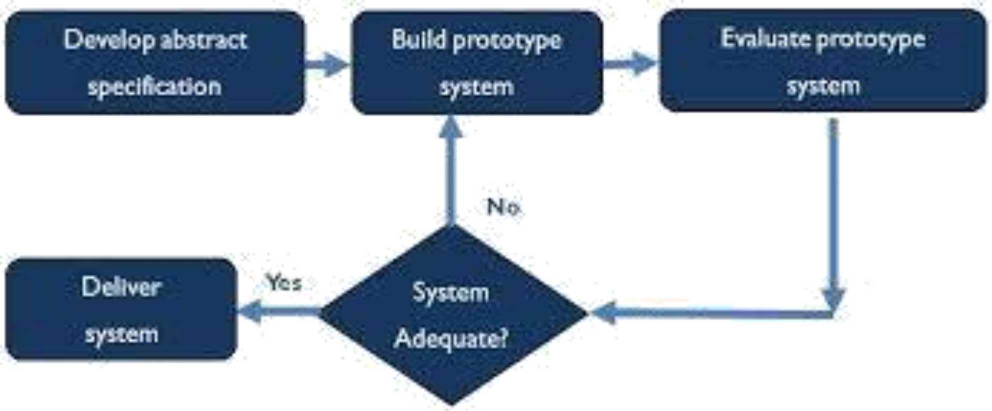


Figure 9: Prototyping Model (Source: Todd, 2013)

CHAPTER FOUR

SYSTEM ANALYSIS AND DESIGN

4 Introduction

This chapter presents how the study of the existing system was done, analysis of the requirements collected from interviews, document reviews and questionnaires and design of the system.

4.1 Requirement Specification

This section defines the functional and non-functional requirements of the artificially intelligent face recognition and Identification System. Requirements are the descriptions of the system services, processes and constraints that are generated during the requirements engineering process.

4.1.1 Functional Requirement

Functional requirements are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations ((Satzinger, 2009). The functional requirements of the Electronic Ticketing System are categorized based on the different system users as described below according to the data collected and analyzed during system study, findings showed that the different system users should be able to accomplish the following functions:

4.1.1.1 System Administrator

i. The System Administrator should login to access functionality of the system and the system should allow him/her to access only functionality of his/her privileges.
ii. The System Administrator should manage users who access this system : register ,view and update security user privileges
iii. The System Administrator should manage the criminal face database: view , delete and update.
iv. The System Administrator should provide support for the system
v. The System Administrator should update the criminal face database in the system

vi. The System Administrator should monitor the system for any improper functionality
vii. The System Administrator should process support tickets escalated by the user
viii. The System Administrator should generate reports from the system

4.1.1.2 Client User/Agent

i. The Customer should create an account to access functionality of the system and the system should allow him/her to access only functionality of his/her privileges.
ii. Security agents/users should be able to view real-time video surveillance directly from the system
iii. Client users should be able to train the system to detect criminal faces directly from the system.
iv. Security Agents should be able to view the wanted criminal database.

4.1.2 Non-Functional Requirements

Non-functional system requirements are those requirements which specify the criteria to be used to judge the operation of the application rather than specify its behaviors (Satzinger, 2009). The system was designed to fulfill some constraints and these are categorized based on the level on which they apply during the user's interaction with the Electronic Ticketing System.

i. Performance: The system should have a quick response time and should be able to handle several users concurrently.
ii. Ease of use: the system should be easy to understand and use by all users.
iii. Reliability: the system should have little or no down time in order to the real-time functionality of the system.
iv. The security agencies should be informed of any errors and invalid input of face data before or after submission.
v. The system should enable encryption of the criminal database in the database for investigation purposes.

4.1.3 Systems Requirements

The system requires minimum specifications of both hardware and software to operate optimally. The hardware minimum specifications include: a processor speed of 2.4 GHz or higher, computer memory of 4GB or higher, 10GB of Hard Disk Space. Minimum software requirements include: Any operating system (Windows, Mac OS, Linux, etc.), webserver software: sqlite3, Python Virtual Environment and OpenCv library.

4.2 Systems Design

4.2.1 Architectural design of the System

The face recognition is the hardest algorithm because it has many steps before it start the real recognition. A face must be detected to increase the possibility of recognition and speed up the process by choosing one location in the image. To detect a face, two steps must be done before the recognition. The first step is to resize the image to standard size (determined by the administrator), apply some filter to increase the quality, and convert the image into a compatible form. Next, go to detection face, such that the image required to recognize is uploaded to the database with a Markup Language (YAML) file to detect a face, and finally, go to recognition step. In recognition, the extracted face will be compared with training faces when they uploading to memory and extract face features by a recognition algorithm. Any operating system (OS) has multiple ways to deal with a process for different structures. Some process has a single thread and other has multithreads architecture (threads can run in a simultaneous manner). Generally the system Architecture and design is under three sections that perform basic functionality; User Interface, Controller, View and Model as shown on the next page in Figure 10.

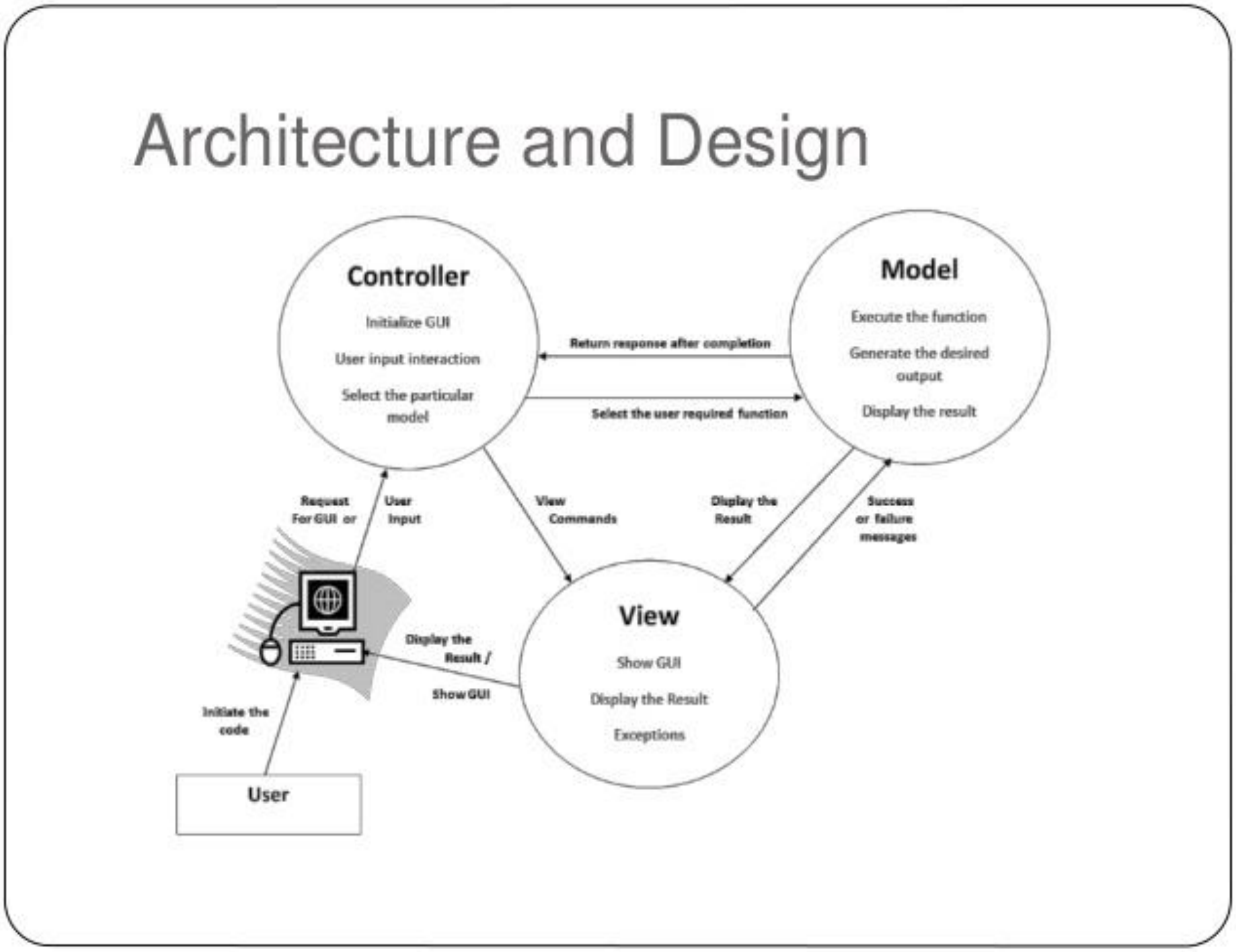
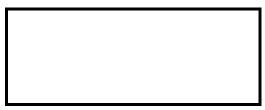
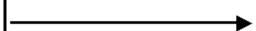


Figure 10: System Architecture & Design

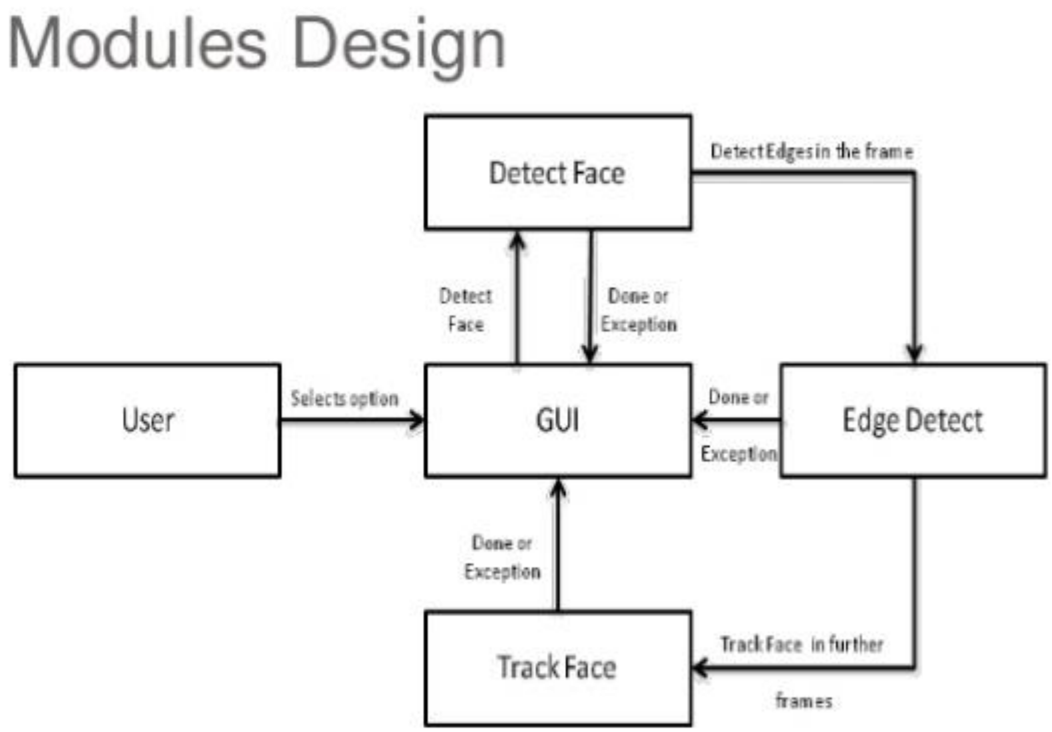
4.2.2 Context diagram of the system

According to Gane & Sarson (2014), a context diagram is a diagram that defines the boundary of the system, or part of a system, and its environment, showing the entities that interact with it as shown in figure. Below are the symbols used in modeling the Context Diagram (Level 0) for the artificially intelligent face detection and identification System.

Name	**Symbol**	**Description**
External entity		External entities are outside the system but they either supply input data into the system or use data from the system as output.
Data flow		Data flows represent movement of data. They show the passage of data in the system and are represented by lines joining system components.
System		Shows the system which the users (entities) interact with by either inputting data or getting it from the System

4.2.3 Data flow diagram of the system

Gane & Sarson (2014) define a data flow diagram (DFD) as a graphical representation of the flow of data through an information system modeling its process aspects. It is therefore used as a step to create an overview of the system by showing the kind of information that will be input to and output from the system, where the data will come from and go to, and where the data will be stored. The data flow diagram of the system is shown in figure below.



4.3 Face detection

First stage was creating a face detection model using Haar-cascade algorithm. Although, training requires 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 identified and eye cascades were incorporated to obtain stable face detection. The flowchart of the detection system can be seen in figure 8. Face ad eye classifier objects are created using classifier class in OpenCV through the cv2.CascadeClassifier () and loading the respective XML files. A camera object is created using the cv2.videoCapture () to capture images.

By using the CascadeClassifter.detectMultiScale () object of various sizes are matched and location is returned. Using the location data, the face is cropped for further verification. Eye cascade is used to verify there are two eyes in the cropped face. If satisfied a marker is placed around the face to illustrate a face is detected in the location

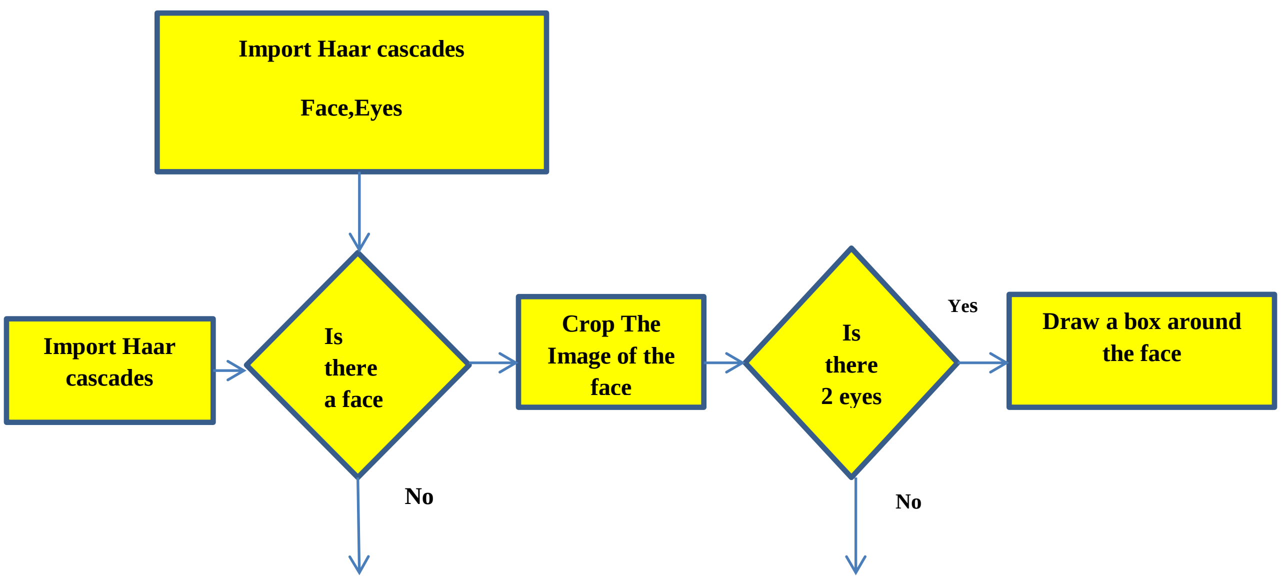


Figure 11: The face must pass all stages of Cascades to be included .

4.3.1 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 face recognition:

a) Collecting face from the National ID Database

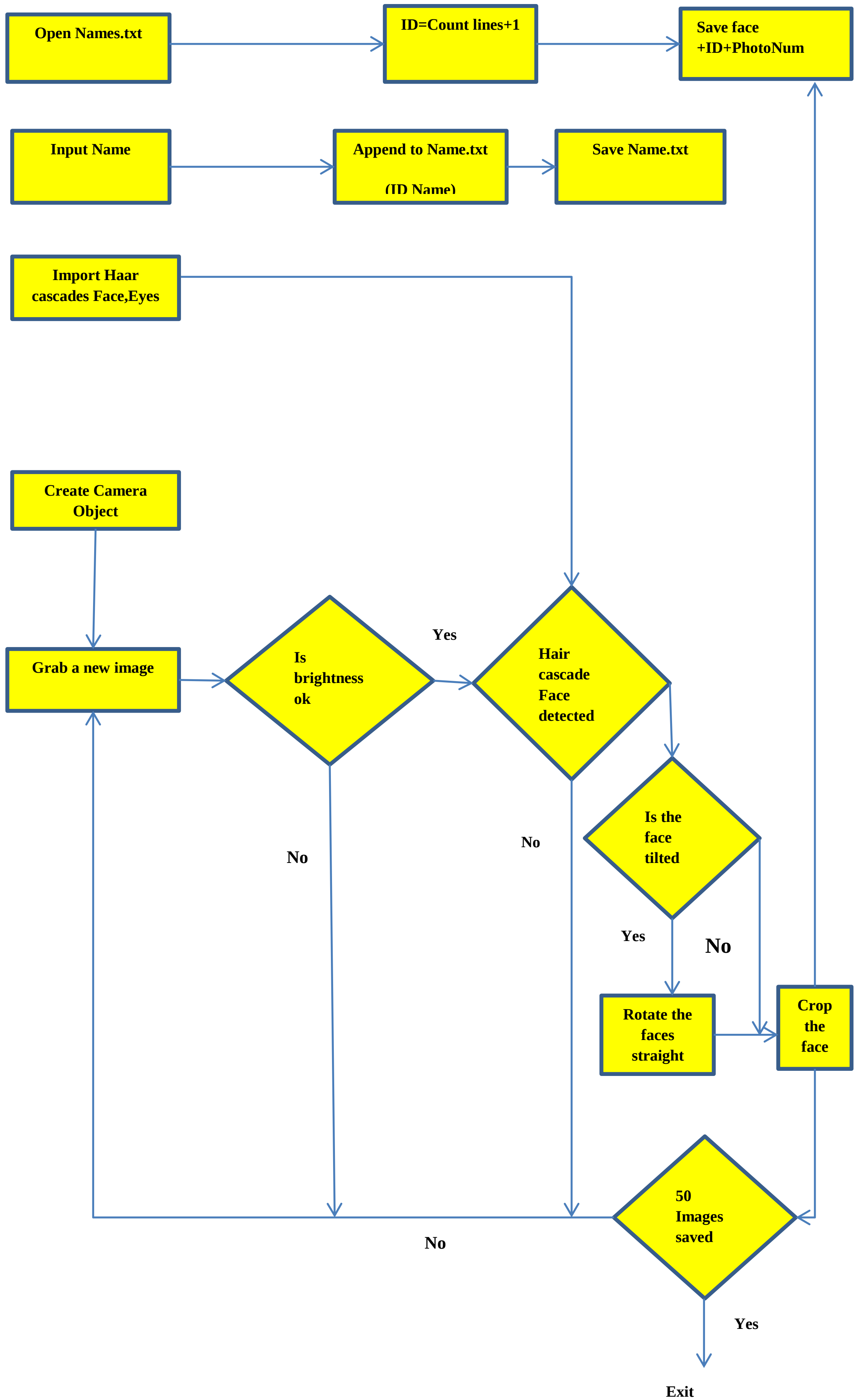
b) Extracting unique features, classifying them and storing in XML files

c) Matching features of an input image to the features in the saved XML files and predict identity

4.3.2 Generating the image dataset

Collecting classification images is usually done manually using a photo editing software to crop and resize photos. Furthermore, PCA and LDA require the same number of pixels in all the images for the correct operation. This time consuming and laborious task is automated through an application to collected 50 images with different expressions. The application detects suitable expressions between 300ms, straightens any existing tilt and save them. The flow chart for the application is shown in figure 12.

Figure 12: The flow chart of the image collection


Open Names.txt
ID=Count lines+1
Save face +ID+PhotoNum
Input Name
Append to Name.txt (ID Name)
Save Name.txt
Import Haar cascades Face,Eyes
Create Camera Object
Grab a new image
Is brightness ok
Yes
Hair cascade Face detected
No
No
Is the face tilted
Yes
No
Rotate the faces straight
Crop the face
50 Images saved
No
Yes
Exit

The Application starts with a request for a name to be entered to be stored with the ID in a text file. The face detection system starts the first half. However, before the capturing begins, the application checks for the brightness levels and will capture only if the face is well illuminated. Furthermore, after the face is detection, the position of the eyes is analyzed. 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 filename to be identified later. A loop runs this program until 30 viable images are collected from the person. This application made data collection efficient.

4.3.3 Training the Classifiers

OpenCV enables the creation of CML files 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 recognizer class. Each recognizer can take in parameters that are described below:

4.3.3.1 cv2.face.createEigenFaceRecognizer ()

i. Takes in the number of components for the PCA for creating Eigenfaces. OpenCV documentation mentions 80 can provide satisfactory capabilities.

ii. Takes in the threshold in recognizing faces. If the distance to the likeliest Eigenface is above this threshold, the function will return a-1, which can be used state the face is unrecognizable.

a) **cv2.face.createFisherfaceRecognizer ()**

i. The first argument is the number of components for the LDA for the creation of Fisher faces OpenCV mentions it to be kept 0 if uncertain.

ii. Similar to Eigenface threshold, -1 if the threshold is passed

b) **cv2.face.createLBPHFaceRecognizer ()**

i.The radius from the Centre pixel to build the local binary pattern.

ii.The number of simple points to build the pattern. Having a considerable number will slow down the computer.

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

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

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

Recognizer objects are created and images are imported, resized, converted into numpy arrays and stored in a vector. By using FaceRecognizer.train (NumpyImage, ID) all three of the objects are trained. It must be noted that resizing the image were requires only for Eigenface and Fishface, not for LBPH. Next, the configuration model is saved as a XML file using FaceRecognizer. Dave (FileName). In this project, all three are trained and saved through one application for convenience. The flow chart for the trainer is shown in figure 11. As Below;

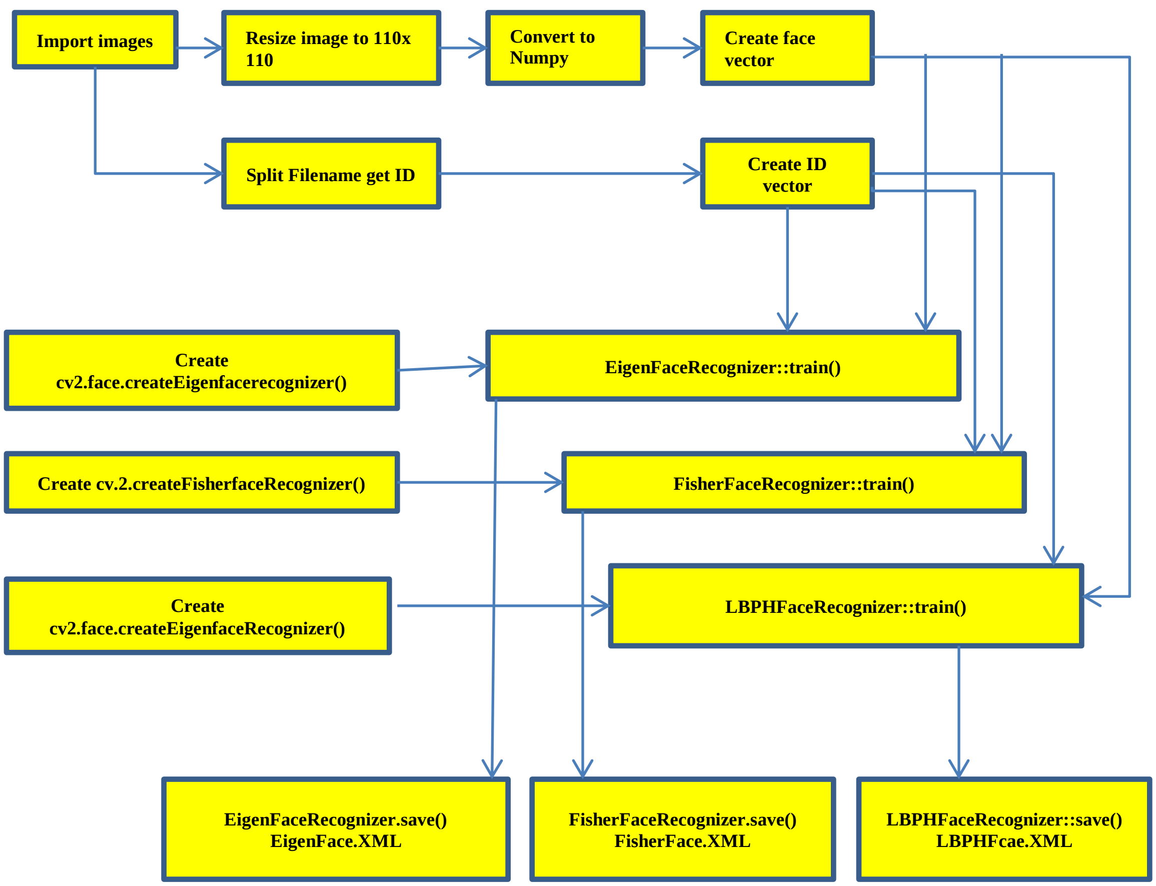


Figure 13: Flow chart of the training Application

The face recognizer object is created using the desired parameters. Face detector is used to detect faces in the image, cropped and transferred to be recognized. 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 confidence. The process is same for all algorithms and if the confidence his higher than the set threshold, ID is -1. Finally, names from the text file with IDs are used to display the name and confidence on the screen. If the ID is -1, the application will print unknown face without the confidence level. The flow chart for the application is shown in figure 12.

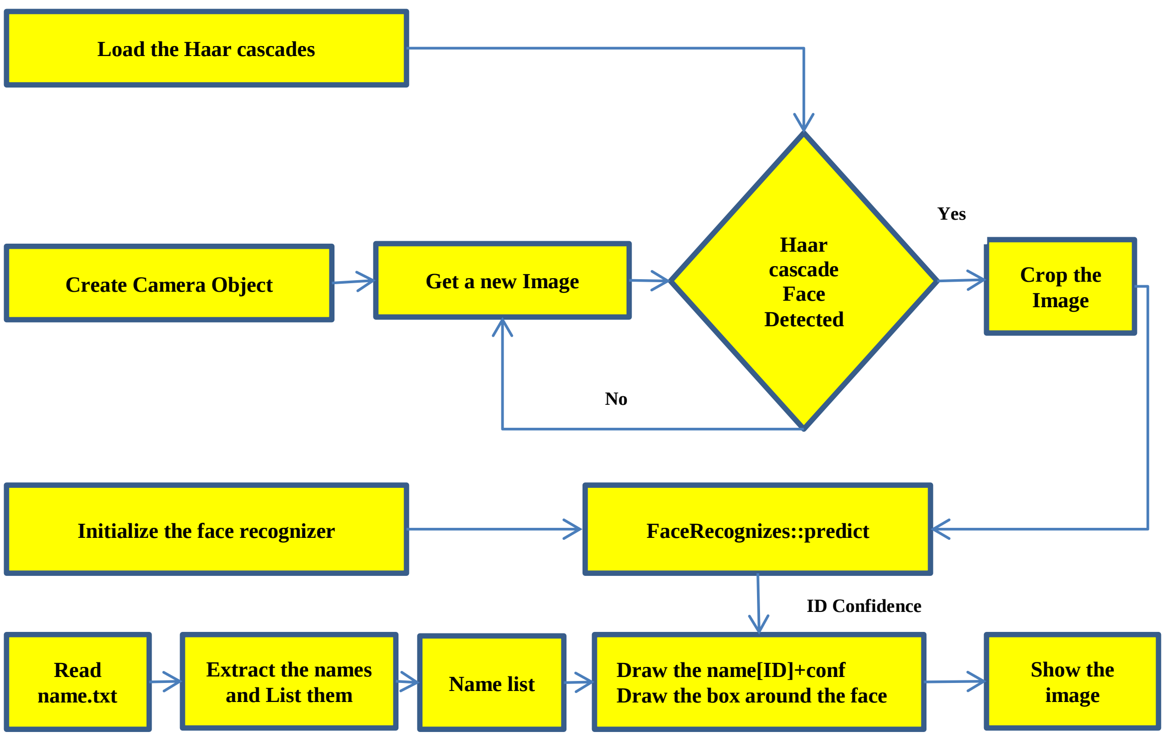


Figure 14: Flow Chart Of the Face Recognition Application

4.4 System Testing

Testing was done after the development phase. This was done in two ways that is: unit testing and system testing, integration testing and acceptance testing. Unit testing was carried out on individual modules of the system to ensure that they are fully functional and accept only valid data. This was done by submitting some data and generating some reports from the unit being tested to ensure that it functions as required. The success of each individual unit enabled the researchers to carryout system testing. System testing was aimed at ensuring that modules are compatible and that they can be integrated to form a complete working system.

4.4.1 Unit Testing:

Each module/ section was tested and the purpose of the unit testing was to verify the correctness of each module making sure individual parts of the system were functioning as expected.

a) Face Detection is tested by using a sample training image or a real time video feed from the camera. We confirm that the unit/section is functional when a rectangle is drawn around the face indicating the system detects a face present.
b) For testing the Face Recognizer, we confirm that recognition is correct when the system identifies the training model face of the criminal by name when displayed in front of the camera. The trained model is extracted from the dataset using the os module and the string operations from the name of the dataset images folder.
c) The Database is tested against its intended functionality of deleting, adding and updating of records

4.4.2 Integration Testing:

Integration testing is done after the individual modules are integrated as a group. So, integration testing is focused on checking whether modules are communicating with each other or not after implementation.

4.4.3 System Testing:

This involves evaluating the system as a whole to find errors and verify if it met specifications. This included first time end to end testing of the application as a complete software product before it is deployed into the target environment.

4.4.4 Acceptance Testing:

Acceptance testing is performed by the users to help verify if the system will meet the user requirements. User acceptance is be performed by the end users to certify the system performs with intended requirements agreed upon.

CHAPTER FIVE

SYSTEM DEVELOPMENT

5 Introduction

This chapter presents the tools and development technologies used to come up with system for this research and the system interfaces for the system developed.

5.1 System Development

The system was developed using Python Programming Language, OpenCv library and IDLE IDE. Sqlite3 was used to create and connect relational database tables, QT Designer was used to create the graphics user interface (GUI), PyQT5 library was used to provide interactive and functionality to the user interfaces. YAML file was used to store trained face data into application.

5.1.1 Python

An Interpreted, high-level, general-purpose, object-oriented programming language that emphasizes code readability with its use of white space supporting multiple programming paradigms, including procedural and functional programming.

5.1.2 OpenCv

A library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel and was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.

5.1.3 IDLE

Integrated development and learning environment is a simple IDE(integrated development environment) for python. It includes multi-window text editor with syntax highlighting, auto completion, smart indent with an integrated debugger.

5.2 System Development

5.2.1 Importing the required modules

The Modules required to perform the facial recognition are cv2, os, image module and numpy. Cv2 is the OpenCV module and contains the functions for face detection and recognition. OS will be used to maneuver with image and directory names. First, we use this module to extract the image names in the database directory and then from these names individual number is extracted, which is used as a label for the face in that image. Since, the dataset images are in gif format and as of now, OpenCV does not support gif format, Image module from PIL is used to read the image in gray scale format. Numpy arrays are used to store the images.

5.2.2 Loading the face detection Cascade

To Load the face detection cascade the first step is to detect the face in each image. Once we get the region of interest containing the face in the image, we use it for training the recognizer. For the purpose of face detection, we will use the Haar Cascade provided by OpenCV. The Haar cascades that come with OpenCV are located in the directory of OpenCV installation. Haar cascade frontal face default.xml is used for detecting the face. Cascade is loaded using the cv2 Cascade Classifier function which takes the path to the cascade xml file. If the xml file is in the current working directory, then relative path is used.

5.2.3 Creating the Face Recognizer Object

The next step involves creating the face recognizer object. The face recognizer object has functions like FaceRecognizer.train() to train the recognizer and FaceRecognizer.predict () to recognize a face. OpenCV currently provides Eigen face Recognizer, Fisher face Recognizer and Local Binary Patterns Histograms (LBPH) Face Recognizer. We have used LBPH recognizer because Real life isn’t perfect. We simply can’t guarantee perfect light settings in your images or 10 different images of a person. LBPH focus on extracting local features from images. The idea is to not look at the whole image as a high-dimensional vector but describe only local features of an object. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighborhood. LBP operator is robust against monotonic gray scale transformations.

5.2.4 Preparing the training set and performing the training

To create the function to prepare the training set, we will define a function that takes the absolute path to the image database as input argument and returns tuple of 2 lists, one containing the detected faces and the other containing the corresponding label for that face. For example, if the i^{th} index in the list of faces represents the 4th individual in the database, then the corresponding i^{th} location in the list of labels has value equal to 4.Train function, It requires 2 arguments, the features which in this case are the images of faces and the corresponding labels assigned to these faces which in this case are the individual number that we extracted from the image names.

5.3 System Implementation

This section presents some of the system interfaces that were designed for the system and their brief explanations.

5.3.1 Detection page

This page allows the user to test/train a criminal's image against the real-time detection functionality of the system as a way of checking whether the availed image of the criminal can be detected. If the system detects the availed training model/image, then the user can proceed to adding the criminal's details to the database which is done by adding their Name, Gender & Crime by clicking Add to Database.

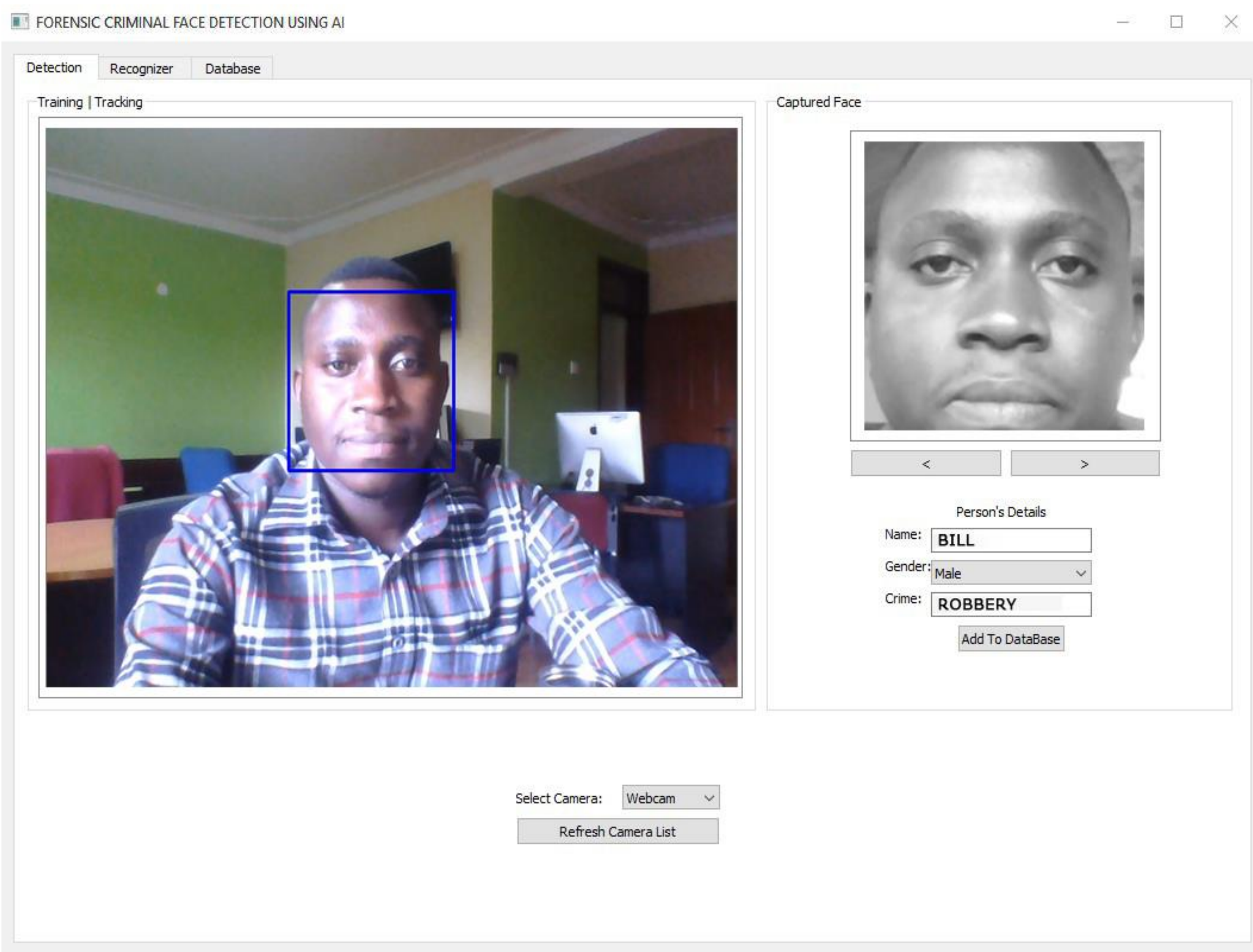


Figure 15: System Detection Page

5.3.2 Recognition Page

This tab has a live video surveillance section which captures & displays video frames from the camera in real time. Face detection is enabled at the backend of each frame to detect and recognize & identify criminals in the captured real time video. The Real-Time Detection section on the right indicates the faces & number of criminals present at the scene using confidence and accuracy algorithm on the backend, records the date, time and location of capture. Upon recognition & Identification, the system automatically generates & sends an email to concerned authorities notifying them of the criminal's presence in a particular location.

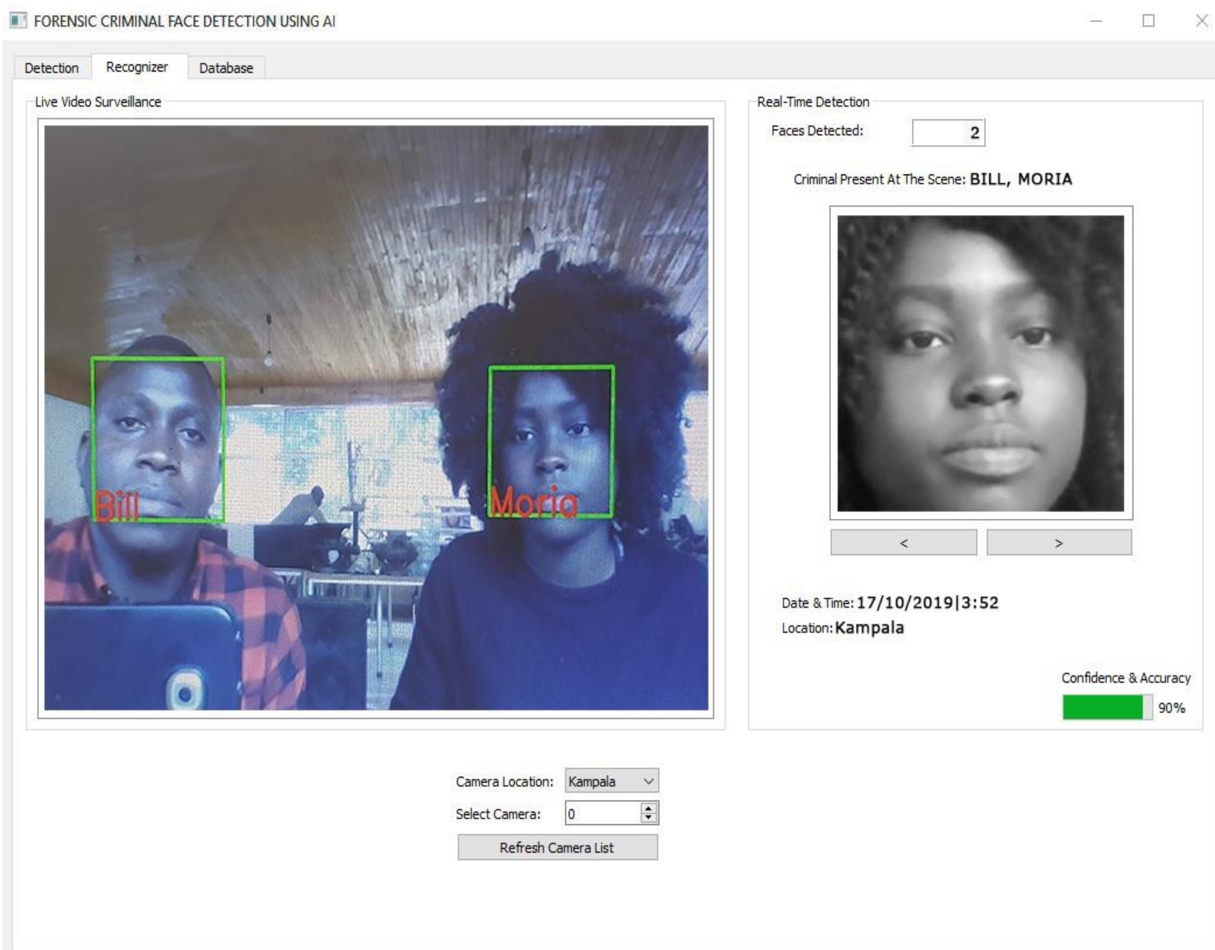


Figure 16: System Recognition Page

5.3.3 Database Page

This page/tab displays a criminal case file from the database in a table which is generated by the detection tab/page when the Add To Database button is clicked. It has a table which displays rows which include; Criminal ID (CID), Criminal's Name, Crime Committed, Gender & Action which deletes the criminals Profile/details from the database. The right section of the same window (Update DB) is where a criminal's image is uploaded & details of name, crime and gender are added/updated in the database by clicking the Add To Database button.

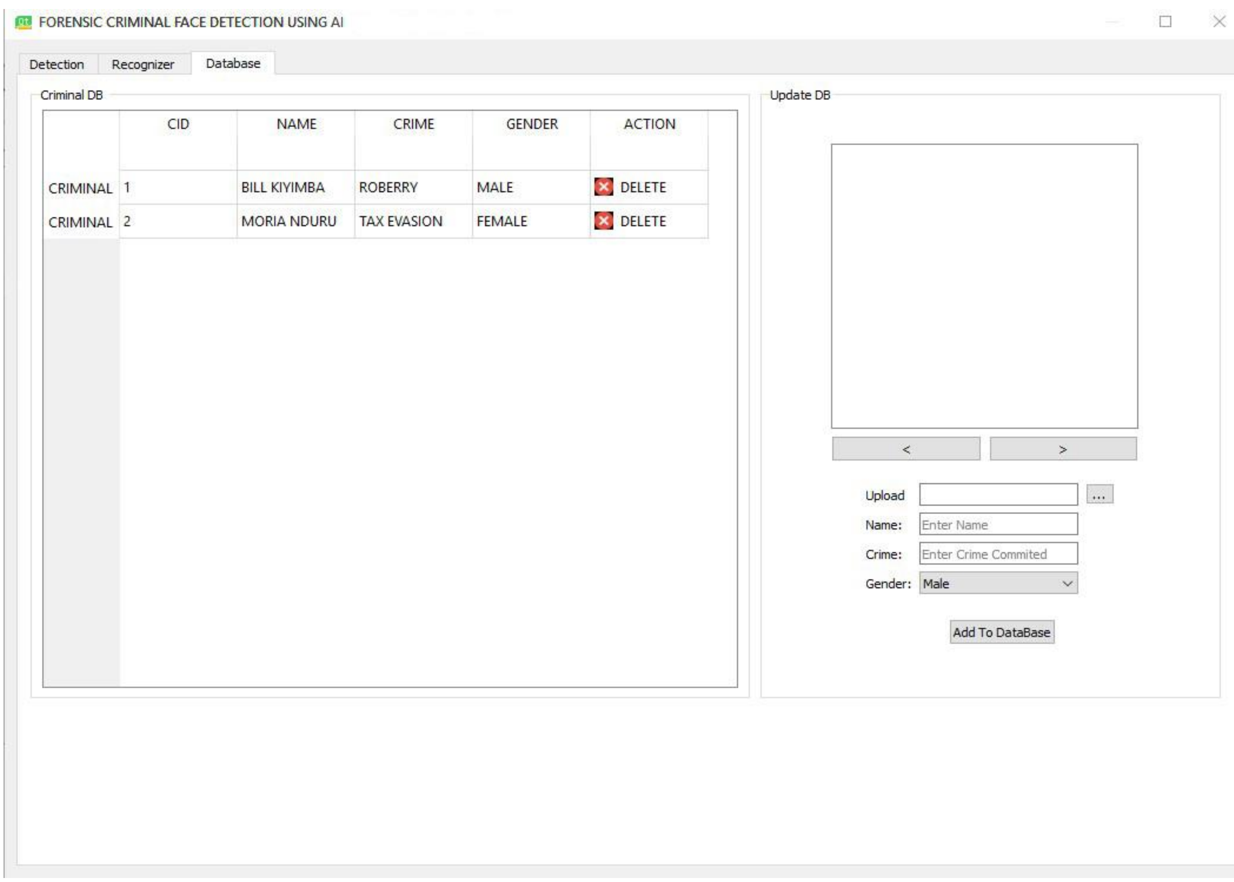


Figure 17: System Database View & Update Page

CHAPTER SIX

CONCLUSION AND RECOMMENDATIONS

6 Introduction

This chapter presents recommendations and conclusion of the research. It also includes the summary of the limitations while carrying out this research.

6.1 Conclusions

In this project, we were able to detect and recognize faces of the criminals in an image and in a video stream obtained from a camera in real time. We have used Haar feature-based cascade classifiers in OpenCV approach for face detection. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Also, we have used Local Binary Patterns Histograms (LBPH) for face recognition. Several advantages of this algorithm are: Efficient feature selection, Scale and location invariant detector, instead of scaling the image itself, we scale the features such a generic detection scheme can be trained for detection of other types of objects (e.g. cars, sign boards, number plates etc.). LBPH recognizer can recognize faces in different lighting conditions with high accuracy. Also, LBPH can recognize efficiently even if single training image is used for each person. Our application has some disadvantages like: Detector is most effective only on frontal images of faces; it can hardly cope with 45° face rotation both around the vertical and horizontal axis.

6.2 Recommendations

6.2.1 Client training

The researchers recommend that all system users should be given appropriate training on how the system works and also plan refresher trainings to keep reviewing the expertise of the users in regard to using the system.

6.2.2 Future works

Face recognition system is designed, implemented and tested. Test results show that system has acceptable performance. On the other hand, system has some future works for improvement and implementation on various video surveillance projects. Future works will be stated in the order of algorithm. First future work can be applied on camera device to improve imaging conditions. Sony camera that is used in thesis, can communicate with computer. Camera configurations can be changed via computer and these changes can improve imaging conditions. Exposure values can be fixed to capture all frames with same brightness value / similar histogram. Also, fixing white balance value can improve performance of skin segmentation which will lead to eliminate non-skin objects. Maybe, white balance correction section may not be needed any more. For later implementations, pan, tilt and zoom actuators can be controlled. Camera is controlled via remote controller in the test of thesis. Then skin color modelling can be improved. In the thesis work, some conditions are used to describe skin color. On the other hand, broad skin color modelling can be achieved by use of statistical modelling. Dark skins, skins under shadow or bright light can be modelled and more skin region segmentation can be achieved. Skin color segmentation is an important step for algorithm of system. If more correct skin regions are segmented, more faces can be detected. Also, instead of RGB, YCbCr skin color modelling with statistical model can be performed, since Cb and Cr channels values are not sensitive to light changes.

On the other hand, some improvements can be applied on facial feature extraction section in face detection part. Computational volume is the biggest with respect to other sections in the algorithm. Computations of facial feature extraction can be reduced. Other point is that to calculate eye orientation, which will be used to reorient face candidate and extract horizontally oriented face image. This operation will decrease working limitations of detection part.

6.2.3 IT infrastructure and System Access

This includes system administration (installing, maintaining, troubleshooting and providing security for server hardware and software) as well as long term maintenance to ensure useful, sustainable, scalable collections and services. The researchers also recommend that the system should be accessed at any time in order to achieve the intended purpose of real-time .

6.2.4 Good feedback mechanism

The researchers recommend that there should a proper feedback mechanism so that system users can give feedback and suggestions about the system, as this will help to improve the system.

6.3 Limitations faced while carrying out the Research

Some limitations of the designed system are determined after the experiments;

a) Skin color l: Skin color is segmented at skin segmentation stage but it will affect results of face candidate. Most of experiments with skin color like show that, face and cloth segments are merged. The software does not recognize them as two separate body regions. Thus, facial feature extraction operation can be resulted, as candidate is not a face or wrong face image extraction. Presence of object on face: Glasses on the face may affect facial feature extraction results. If glasses have reflection, LoG filter cannot perform well on eye region. Also, sunglasses will cover eye region and eyes cannot be detected based on the proposed algorithm.

b) Contrast of face candidate image: Contrast value of image affects results of filter. Less contrast image has fewer edges on the image. Thus, facial component could not be visible after filtered. Therefore, face image cannot be extracted if candidate contains face.

c) System working range: System can detect and recognize the person if person standing range is between 50 cm to 180 cm. This range is acquired with property of 3.1 mm focal length and 768x576 pixels resolution. Thus, working range can be changed by camera property.

d) Skin color range: RGB skin color segmentation work well in the range of light tone to dark tone. Head pose: Frontal head poses can be detected and extracted correctly. Small amount of roll and yaw rotation can be acceptable for the system

REFERENCES

(n.d.). Retrieved from https://www.researchgate.net/publication/265026957_DEVELOPMENT_OF_A_FACE_RECOGNITION_SYSTEM

Face Recognition Homepag. (2019). Retrieved from http://www.face-rec.org/algorithms/

Brunelli, R. P. (1993). *Face Recognition: Features versus templates.* IEEE Transaction on Pattern Analysis and Machine Intelligence .

DAVID H, H. T. (2004). *Brain and Visual Perception: The Story of a 25-year Collaboration.* Oxford Scholarship.

Dong chen He, L. W. (1990). *Texture unit, texture spectrum, and texture analysis.* IEEE Transactions on Geoscience and Remote Sensing.

Emami, S. (2010). *Face Detection and Recognition using OpenCV*. Retrieved from www.shervinemami.info: http://shervinemami.info/faceRecognition.html

Hewitt, R. (2010). *Seeing with OpenCV, Article*. Retrieved from www.cognotics.com: http://www.cognotics.com/opencv/servo_2007_series/part_1/index.html

Hongpe, Z. (2018). *Chinese facial ID tech to land in Africa.* Global Times.

https://www.researchgate.net/publication/265026957_DEVELOPMENT_OF_A_FACE_RECOGNITION_SYSTEM. (n.d.). Retrieved from https://www.researchgate.net/publication/265026957_DEVELOPMENT_OF_A_FACE_RECOGNITION_SYSTEM: https://www.researchgate.net/publication/265026957_DEVELOPMENT_OF_A_FACE_RECOGNITION_SYSTEM

Paul Viola, M. J. (2001). *Robust Real-Time Object Detection.* International Journal of Computer Vision .

PEI LI, P. J. (2019). *FaceRecognitioninLowQualityImages:ASurvey.*

Peng Wu, H. Z. (2019). Some Analysis and Research of the AdaBoost Algorithm. *International Conference on Intelligent Computing and Information Science* (pp. 1-5). Intelligent Computing and Information Science.

R.Rajalakshmi, M. (2013). *A Novel Approach to Face Recognition with Pose and Illumination Variation Using Support Vector Machine as Classifier.* International Journal of Innovative Technology and Exploring Engineering (IJITEE) .

Shailesh Wadhankar, S. S. (2018). *Real Face Detection and Recognition: The Live Experiment* . International Journal of Computer Applications (0975 – 8887) .

Sonia Ohlyan, S. S. (2013). *A Survey On Various Problems & Challenges In Face Recognition* . International Journal of Engineering Research & Technology (IJERT) .

T. X. Han, S. Y. (2009). An hog-lbp human detector with partial occlusion handling. *12th International Conference on Computer Vision* (pp. 32-39). IEEE.

TechTarget. (2019). *facial recognition*. Retrieved from www.searchenterpriseai.techtarget.com: https://searchenterpriseai.techtarget.com/definition/facial-recognition

Tom Wilson, M. M. (2019). *Uganda confirms use of Huawei facial recognition cameras*. Retrieved from Financial Times: https://www.ft.com/content/e20580de-c35f-11e9-a8e9-296ca66511c9

Viola, P. J. (2001). *Rapid object detection using boosted cascade of simple features.* IEEE Conference on Computer Vision and Pattern Recognition.

Wikipedia. (2019). */Cascading_classifiers.* Retrieved from www.wikipedia.org/wiki: https://en.wikipedia.org/wiki

Wikipedia. (2019). *Three-dimensional face recognition,*. Retrieved from www.en.wikipedia.org: http://en.wikipedia.org/wiki/Three- dimensional_face_recognition

APPENDIX

Interview guide

INTERNATIONAL UNIVERSITY OF EAST AFRICA

Interview guide for gathering information about Operations at the National Network of CCTV Expansion Police Project

The information provided will help to know the challenges of the current system and also get new requirements for designing the artificially intelligent forensic surveillance System. The information you provide will be kept confidential.

1 Demographic information

a. Position held...

b. Experience...

2 Which details do you normally capture from the CCTVs, Accidents, Traffic or Criminals?

3 What is the process is collecting this data currently?

4 Any measures to ensure that the data collected is valid?

5 In your opinion, are there any challenges related to the data collection process?

6 If YES, what are these challenges?

7 What is the current process of identification of criminals during investigations?

8 In your opinion, are there any challenges related to this process?

9 If YES, what are these challenges?

10 Which reports are normally generated from the CCTV footage?

11 In your opinion, are there any challenges related to the report generation process?

12 If YES, what are these challenges?

13 If you were to make recommendations to help improve the current system processes, which changes would you suggest?

Thank you for your Time.