AUTOMATED ATTENDANCE SYSTEM

B. Tech Major-Project Report

Submitted in partial fulfilment for the award of the Degree of

Bachelor of Technology in EC-Engineering

By

SAHIL SHARMA	*14BTD5060244*
ARPIT SHARMA	*14BTD5060208*
SHILPA KAPOOR	*14BTD5060248*
NAVNEET SINGH	*14BTD5060228*

Under the guidance of

Mr. AKSHAY KANWAR

Department of EC Engineering

Rajiv Gandhi government Engineering College Nagrota Bagwan

CERTIFICATE

We hereby certify that the work which is being presented in project work II report entitled "AUTOMATED ATTENDANCE SYSTEM" in partial fulfilment of the requirements for the award of the B.Tech. submitted to department of Electronics & Communication Engineering of Rajiv Gandhi Government Engineering College Kangra at Nagrota Bagwan is a record of project work carried out during a period from January 2019 to May 2019 under the supervision of Mr. Akshay Kanwar.

Name & Roll No.	Sahil Sharma 14BTD5060244	Arpit Sharma 14BTD5060208	Shilpa Kapoor 14BTD5060248	Navneet Singh 14BTD5060228
Signature				

This is to certify that the above statement made by the candidates is correct to the best of our knowledge.

Mr. Akshay kanwar
Asst. Prof. E&CED

Dr. P.P Sharma
HOD, E&CED

ACKNOWLEDGEMENT

On the submission of our major project work report on "AUTOMATED ATTENDANCE SYSTEM", we would like to express my indebted gratitude and special thanks to our guide Mr. Akshay Kanwar, Department of Electronics and Communication Engineering who in spite of being extraordinarily busy, spare time for guidance and keep us on the correct path and allowing us to carry out our work in the four months. We truthfully appreciate and value his admired supervision and support from the start to the end of this project. I am obliged to him for having helped me shape the trouble and providing insights towards the way out.

we would like to offer my special thanks to Prof. P.P Sharma, Head of Department Electronics and Communication Engineering for providing us a chance to work out on this project. He has been great source of motivation to us and We thank him from the core of our heart.

Last but not the least we would like to thank each and every person who is involved directly or indirectly to make this project successful.

SAHIL SHARMA	**ARPIT SHARMA**	**SHILPA KAPOOR**	**NAVNEET SINGH**
14BTD5060244	*14BTD5060208*	*14BTD5060248*	*14BTD5060228*

Contents

Table of figures 5
List of Abbreviation 6
Abstract........ 7
1. INTRODUCTION 8
1.1 Flow Chart 9
2. Literature Review 10
2.1 Digital Image Processing. 10
2.2 Human Perception 10
2.3 Machine Vision Applications 10
2.4 Image Representation in a Digital Computer. 11
2.5 Steps in Digital Image Processing. 11
2.6 Definition of Terms and History 12
2.6.1 Face Detection 12
2.6.2 Face Recognition 12
2.6.3 Difference between Face Detection and Face Recognition 13
2.7 Face Detection 13
2.8 Haar – Cascades 13
2.9 How the Haar – like Features Work 15
2.10 Cascaded Classifier 16
2.11 Integral Image 16
3. Methodology and Design 18
3.1 General Overview 18
3.1.1 Training Set Manager Sub System 19
3.1.2 Face Recognizer Sub System 19
3.2 System Architecture 20
3.3 Functions of the two Sub –Systems 20
4. Viola- Jones Algorithm for Face Detection 22
4.1 The Scale Invariant Detector 22
4.2 The Modified AdaBoost Algorithm 24
4.3 The Cascaded Classifier 24
5. Technology and Tools 27

5.1 Python Introduction 27
5.1.1 What can Python do? 27
5.1.2 Why Python? 27
5.2 Python libraries 28
5.2.1 OpenCV 28
5.2.2 OpenCV-Python 28
5.2.3 NumPy 29
5.2.4 Tkinter 30
5.2.5 CSV 30
5.2.6 Pandas 30
5.3 Computer Vision Libraries 31
5.3.1 Theano 31
5.3.2 Caffe 31
5.3.3 TensorFlow 32
5.4 Requirements to Run the Application: 32
5.4.1 Software Requirements 32
5.5 Hardware Requirements 33
6. GUI CONSTRUCTING 35
7. CODE-IMPLEMENTATION 36
7.1 Implementation of dataset creation 36
7.2 Algorithm for Creating Dataset 37
7.3 Algorithm for Trainer 37
7.4 Implementation of Recognizer and Attendance 38
8. LIMITATIONS 40
9. Conclusion and Recommendation 41
10. FUTURE SCOPE 42
11. REFERENCES 43

Table of figures

Figure 1 Students waiting in a que for biometric attendance 8
Figure 2 Flowchart of attendance system 9
Figure 3 A diagram showing the steps in digital image processing 12
Figure 4 Haar like Features 13
Figure 5 Haar-like features with different sizes and orientation 14
Figure 6 How the Haar like feature of figure used to scale the eyes 15
Figure 7 several classifiers combined to enhance face detection 16
Figure 8 Pixel Coordinates of an integral image 17
Figure 9 Values of the integral Image on a rectangle 17
Figure 10 Series Of Components 18
Figure 11 The logical design of the Desktop Module Subsystems 20
Figure 12 A block diagram showing functions of the components 21
Figure 13 The integral image 22
Figure 14 Sum calculation 23
Figure 15 The different types of features 23
Figure 16 The cascade classifier 25
Figure 17 Hardware Requirements 34
Figure 18 GUI Home page 35
Figure 19 taking images for dataset 36
Figure 20 marking attendance 1 38
Figure 21 Marking attendance 2 39

List of Abbreviation

OpenCV	Open Source Computer Vision Library
API	Application Programming Interface
GPU	Graphical Processing Units
NumPy	Numerical Python
CSV	Comma Separated Values
GUI	Graphical User Interface
SciPy	Scientific Python
MATLAB	Matrix Laboratory
PlotLib	Plotting Library
HCL	Hardware Compatibility List

Abstract

Uniqueness or individuality of an individual is his face. In this project face of an individual is used for the purpose of attendance making automatically. Attendance of the student is very important for every college, universities and school. Conventional methodology for taking attendance is by calling the name or roll number of the student and the attendance is recorded. Time consumption for this purpose is an important point of concern. Assume that the duration for one subject is around 60 minutes or 1 hour & to record attendance takes 5 to 10 minutes. For every tutor this is consumption of time. To stay away from these losses, an automatic process is used in this project which is based on image processing.

In this project face detection and face recognition is used. Face detection is used to locate the position of face region and face recognition is used for marking the understudy's attendance. The database of all the students in the class is stored and when the face of the individual student matches with one of the faces stored in the database then the attendance is recorded.

The main goal and objective of this automated attendance system of face detection and recognition is to present face recognition in real time environment, to see and mark the attendance of their students and employees on a daily basis to keep track of their presence. The system will mark and record the attendance in any environment. This system is automated and user can capture video and accordingly attendance will be marked, improving the accuracy to great extent and finally the attendance report will be generated.

1. INTRODUCTION

Attendance is prime important for both the teacher and student of an educational organization. It is very important to keep record of the attendance. The problem arises when we think about the traditional process of taking attendance in class room. Calling name or roll number of the student for attendance is not only a problem of time consumption but also it needs energy. An automatic attendance system can solve all above problems.

There are some automatic attendances making system which are currently used by much institution. One of such system is biometric technique. Although it is automatic and a step ahead of traditional method it fails to meet the time constraint. The student has to wait in queue for giving attendance, which is time taking.

This project introduces an involuntary attendance marking system, devoid of any kind of interference with the normal teaching procedure. The system can be also implemented during exam sessions or in other teaching activities where attendance is highly essential. This system eliminates classical student identification such as calling name of the student, or checking respective identification cards of the student, which can not only interfere with the ongoing teaching process, but also can be stressful for students during examination sessions.

FIGURE 1 STUDENTS WAITING IN A QUE FOR BIOMETRIC ATTENDANCE

1.1 Flow Chart

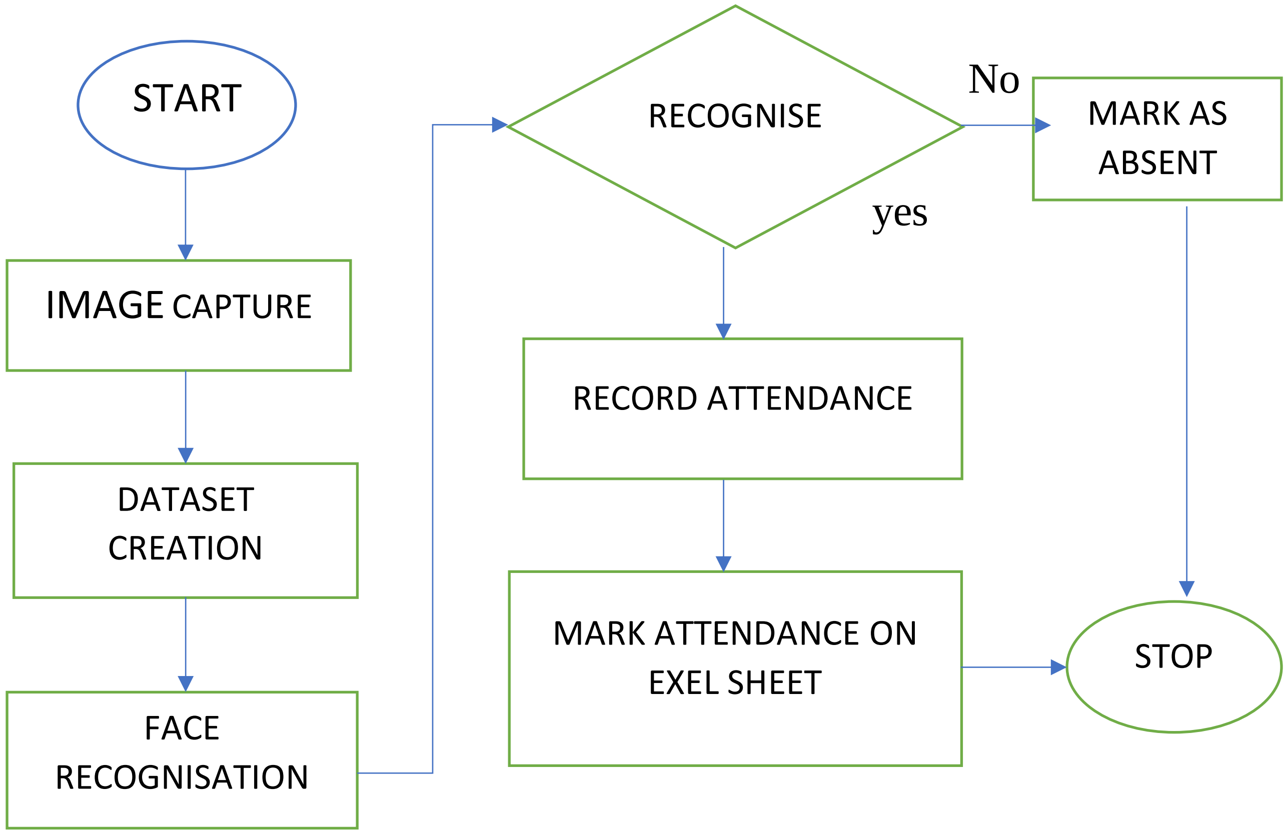


FIGURE 2 FLOWCHART OF ATTENDANCE SYSTEM

2. Literature Review

2.1 Digital Image Processing.

Digital Image Processing is the processing of images which are digital in nature by a digital computer. Digital image processing techniques are motivated by three major applications mainly:

- Improvement of pictorial information for human perception
- Image processing for autonomous machine application
- Efficient storage and transmission.

2.2 Human Perception

This application employs methods capable of enhancing pictorial information for human interpretation and analysis. Typical applications include; noise filtering, content enhancement mainly contrast enhancement or deblurring and remote sensing.

2.3 Machine Vision Applications

In this, the interest is on the procedures for extraction of image information suitable for computer processing. Typical applications include;

- Industrial machine vision for product assembly and inspection.
- Automated target detection and tracking.
- Finger print recognition.
- Machine processing of aerial and satellite imagery for weather prediction and crop assessment.

Facial detection and recognition fall within the machine vision application of digital image processing.

2.4 Image Representation in a Digital Computer.

An image is a 2-Dimensional light intensity function

$$\mathbf{f}\ (\mathbf{x}, \mathbf{y}) = \mathbf{r}\ (\mathbf{x}, \mathbf{y}) \times \mathbf{i}\ (\mathbf{x}, \mathbf{y})$$

$\mathbf{r}\ (\mathbf{x}, \mathbf{y})$ is the reflectivity of the surface of the corresponding image point.

$\mathbf{i}\ (\mathbf{x}, \mathbf{y})$ Represents the intensity of the incident light.

A digital image **f (x, y)** is discretized both in spatial co-ordinates by grids and in brightness by quantization. Effectively, the image can be represented as a matrix whose row, column indices specify a point in the image and the element value identifies gray level value at that point. These elements are referred to as pixels or pels.

Typically following image processing applications, the image size which is used is $\mathbf{256} \times \mathbf{256}$, elements, $\mathbf{640} \times \mathbf{480}$ pels or $\mathbf{1024} \times \mathbf{1024}$ pixels. Quantization of these matrix pixels is done at 8 bits for black and white images and 24 bits for coloured images (because of the three colour planes Red, Green and Blue each at 8 bits).

2.5 Steps in Digital Image Processing.

Digital image processing involves the following basic tasks;

- Image Acquisition - An imaging sensor and the capability to digitize the signal produced by the sensor.
- Preprocessing – Enhances the image quality, filtering, contrast enhancement etc.
- Segmentation – Partitions an input image into constituent parts of objects.
- Description/feature Selection – extracts the description of image objects suitable for further computer processing.
- Recognition and Interpretation – Assigning a label to the object based on the information provided by its descriptor. Interpretation assigns meaning to a set of labelled objects.

- Knowledge Base – This helps for efficient processing as well as inter module cooperation.

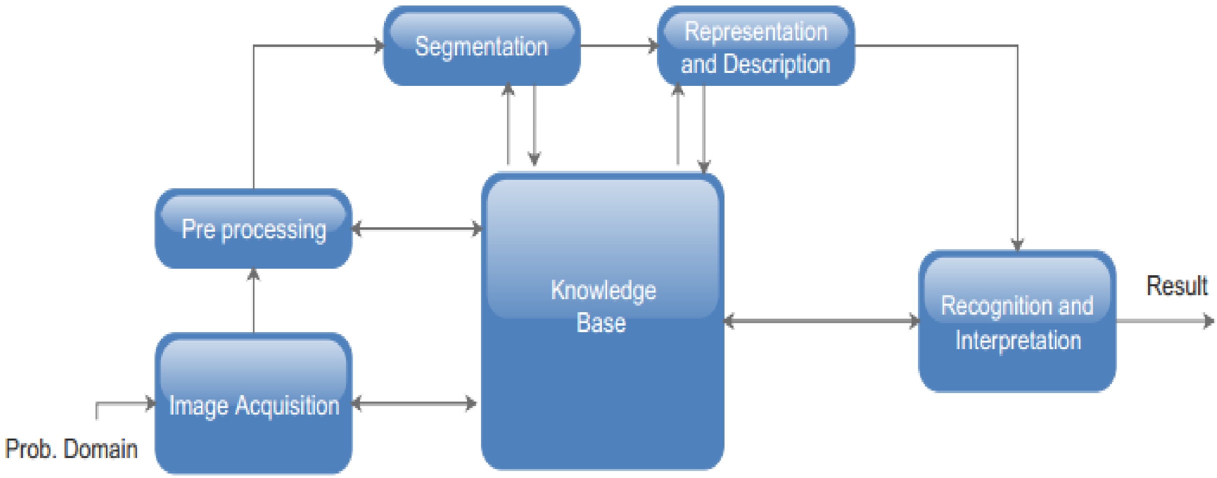


FIGURE 3 A DIAGRAM SHOWING THE STEPS IN DIGITAL IMAGE PROCESSING

2.6 Definition of Terms and History

2.6.1 Face Detection

Face detection is the process of identifying and locating all the present faces in a single image or video regardless of their position, scale, orientation, age and expression. Furthermore, the detection should be irrespective of extraneous illumination conditions and the image and video content.

2.6.2 Face Recognition

Face Recognition is a visual pattern recognition problem, where the face, represented as a three-dimensional object that is subject to varying illumination, pose and other factors, needs to be identified based on acquired images.

Face Recognition is therefore simply the task of identifying an already detected face as a known or unknown face and in more advanced cases telling exactly whose face it is.

2.6.3 Difference between Face Detection and Face Recognition

Face detection answers the question, where is the face? It identifies an object as a "face" and locates it in the input image. Face Recognition on the other hand answers the question who is this? Or whose face is it? It decides if the detected face is someone known or unknown based on the database of faces it uses to validate this input image.

2.7 Face Detection

A face Detector has to tell whether an image of arbitrary size contains a human face and if so, where it is. Face detection can be performed based on several cues: skin color (for faces in color images and videos, motion (for faces in videos), facial/head shape, facial appearance or a combination of these parameters. Most face detection algorithms are appearance based without using other cues.

An input image is scanned at all possible locations and scales by a sub window. Face detection is posed as classifying the pattern in the sub window either as a face or a non-face. The face/non-face classifier is learned from face and non-face training examples using statistical learning methods.

Most modern algorithms are based on the Viola Jones object detection framework, which is based on Haar-Cascades.

2.8 Haar – Cascades

Haar like features are rectangular patterns in data. A cascade is a series of "Haar-like features" that are combined to form a classifier. A Haar wavelet is a mathematical function that produces square wave output.

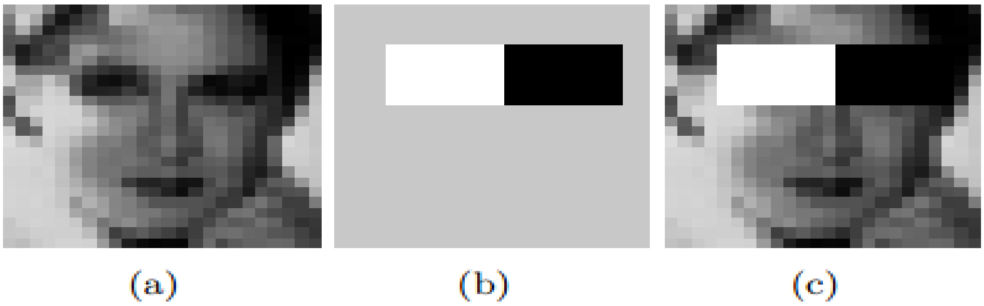


FIGURE 4 HAAR LIKE FEATURES

Figure 4 shows Haar like features, the background of a template like (b) is painted gray to highlight the pattern's support. Only those pixels marked in black or white are used when the corresponding feature is calculated.

Since no objective distribution can describe the actual prior probability for a given image to have a face, the algorithm must minimize both the false negative and false positive rates in order to achieve an acceptable performance. This then requires an accurate numerical description of what sets human faces apart from other objects. Characteristics that define a face can be extracted from the images with a remarkable committee learning algorithm called Adaboost. Adaboost (Adaptive boost) relies on a committee of weak classifiers that combine to form a strong one through a voting mechanism. A classifier is weak if, in general, it cannot meet a predefined classification target in error terms. The operational algorithm to be used must also work with a reasonable computational budget. Such techniques as the integral image and attentional cascades have made the Viola-Jones algorithm highly efficient: fed with a real time image sequence generated from a standard webcam or camera, it performs well on a standard PC.

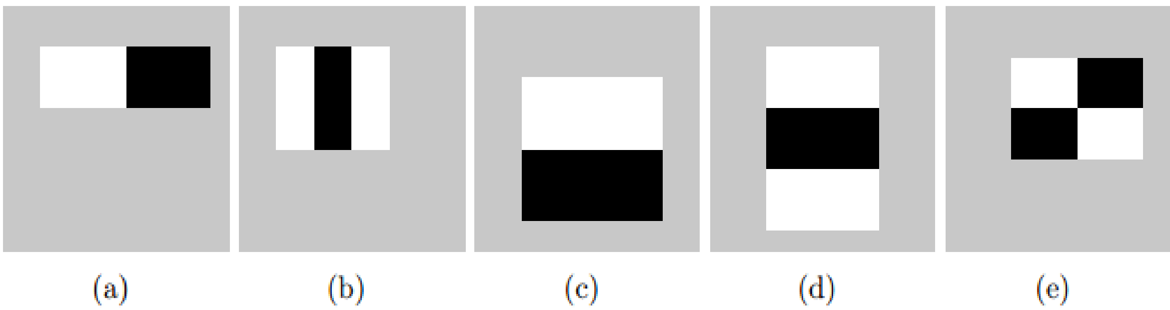


FIGURE 5 HAAR-LIKE FEATURES WITH DIFFERENT SIZES AND ORIENTATION

The size and position of a pattern's support can vary provided its black and white rectangles have the same dimension, border each other and keep their relative positions. Thanks to this constraint, the number of features one can draw from an image is somewhat manageable: a 24 × 24 image, for instance, has 43200, 27600, 43200, 27600 and 20736 features of category (a), (b), (c), (d) and (e) respectively as shown in figure 5, hence 162336 features in all.

In practice, five patterns are considered. The derived features are assumed to hold all the information needed to characterize a face. Since faces are large and regular by nature, the use of Haar-like patterns seems justified.

2.9 How the Haar – like Features Work

A scale is chosen for the features say 24×24 pixels. This is then slid across the image. The average pixel values under the white area and the black area are then computed. If the difference between the areas is above some threshold then the feature matches.

In face detection, since the eyes are of different colour tone from the nose, the Haar feature (b) from Figure 5 can be scaled to fit that area as shown below,

FIGURE 6 HOW THE HAAR LIKE FEATURE OF FIGURE USED TO SCALE THE EYES

One Haar feature is however not enough as there are several features that could match it (like the zip drive and white areas at the background of the image of figure 6). A single classifier therefore isn't enough to match all the features of a face, it is called a "weak classifier."

Haar cascades, the basis of Viola Jones detection framework therefore consists of a series of weak classifiers whose accuracy is at least 50% correct. If an area passes a single classifier, it moves to the next weak classifier and so on, otherwise, the area does not match.

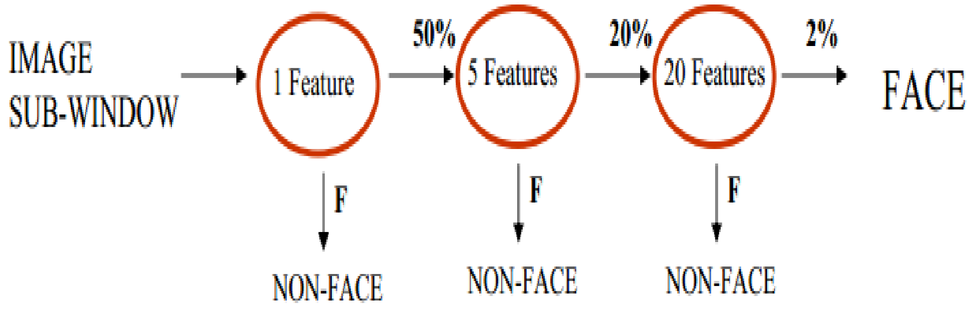


FIGURE 7 SEVERAL CLASSIFIERS COMBINED TO ENHANCE FACE DETECTION

2.10 Cascaded Classifier

From figure 7, a 1 feature classifier achieves 100% face detection rate and about 50% false positive rate. A 5-feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative). A 20-feature classifier achieves 100% detection rate with 10% false positive rate (2% cumulative). Combining several weak classifiers improves the accuracy of detection.

A training algorithm called Adaboost, short for adaptive boosting, which had no application before Haar cascades, was utilized to combine a series of weak classifiers in to a strong classifier. Adaboost tries out multiple weak classifiers over several rounds, selecting the best weak classifier in each round and combining the best weak classifier to create a strong classifier. Adaboost can use classifiers that are consistently wrong by reversing their decision. In the design and development, it can take weeks of processing time to determine the final cascade sequence.

After the final cascade had been constructed, there was a need for a way to quickly compute the Haar features i.e. compute the differences in the two areas. The integral image was instrumental in this.

2.11 Integral Image

The Integral image also known as the “summed area table” developed in 1984 came in to widespread use in 2001 with the Haar cascades. A summed area table is created in a single pass. This makes the Haar cascades fast, since the sum of any region in the image can be computed using a single formula.

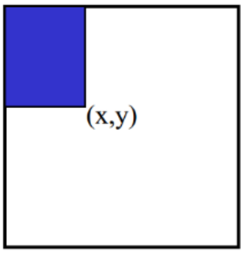


FIGURE 8 PIXEL COORDINATES OF AN INTEGRAL IMAGE

The integral image computes a value at each pixel (x, y) as is shown in figure 8, that is the sum of the pixel values above and to the left of (x, y), inclusive. This can quickly be computed in one pass through the image.

Let A, B, C D be the values of the integral image at the corners of a rectangle as shown in figure

The sum of original image values within the rectangle can be computed.

$$Sum=A-B-C+D$$

Only three additions are required for any size of rectangle. This face detection approach minimizes computation time while achieving high detection accuracy. It is now used in many areas of computer vision.

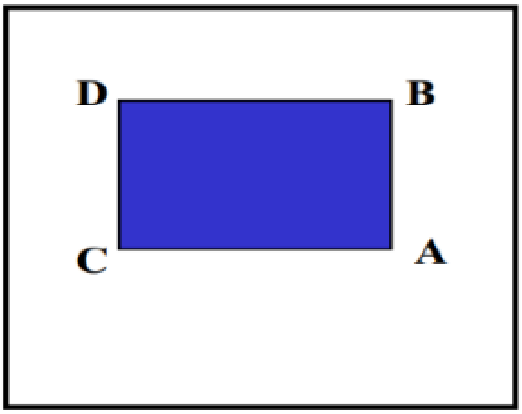


FIGURE 9 VALUES OF THE INTEGRAL IMAGE ON A RECTANGLE

3. Methodology and Design

3.1 General Overview

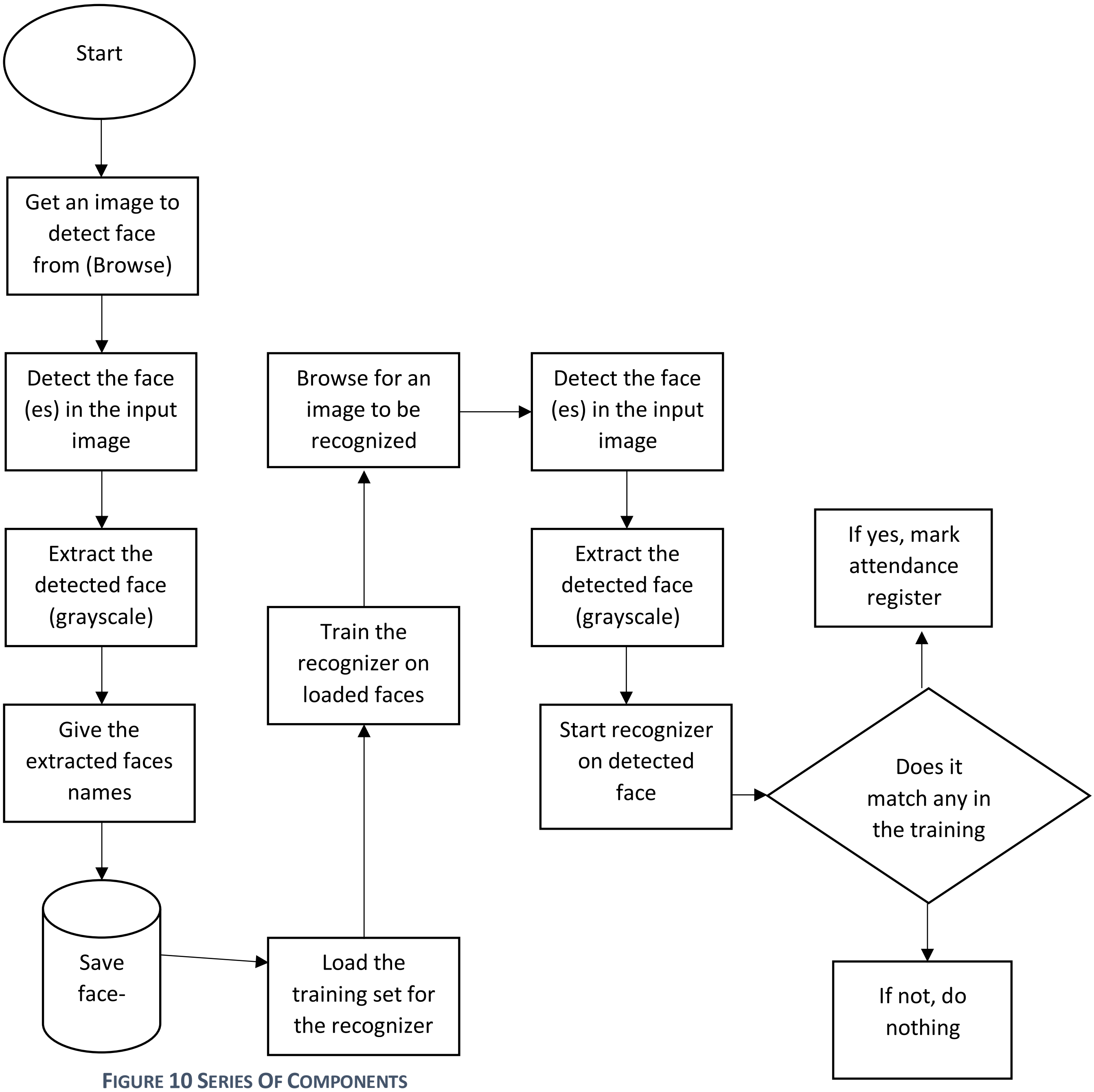


FIGURE 10 SERIES OF COMPONENTS

From Figure 10, it can be observed that most of the components utilized are similar;(the Image acquisition component for browsing for input images, the face detector and the faces database for storing the face label pairs) only that they are employed at the different stages of the face recognition process.

3.1.1 Training Set Manager Sub System

The logical design of the training set management sub-system is going to consist of an image acquisition component, a face detection component and a training set management component. Together, these components interact with the faces database in order to manage the training set.

These are going to be implemented in a windows application form.

3.1.2 Face Recognizer Sub System

The logical design of the Face Recognizer will consist of the image acquisition component, face recognizer and face detection component all working with the faces database.

In this the image acquisition, and face detection component are the same as those in the Training set manager sub system as the functionality is the same. The only difference is the face recognizer component and its user interface controls.

This will load the training set again so that it trains the recognizer on the faces added and show the calculated eigenfaces and average face. It should then show the recognized face in a picture box.

3.2 System Architecture

The figure below shows the logical design and implementation of the three desktop subsystems.

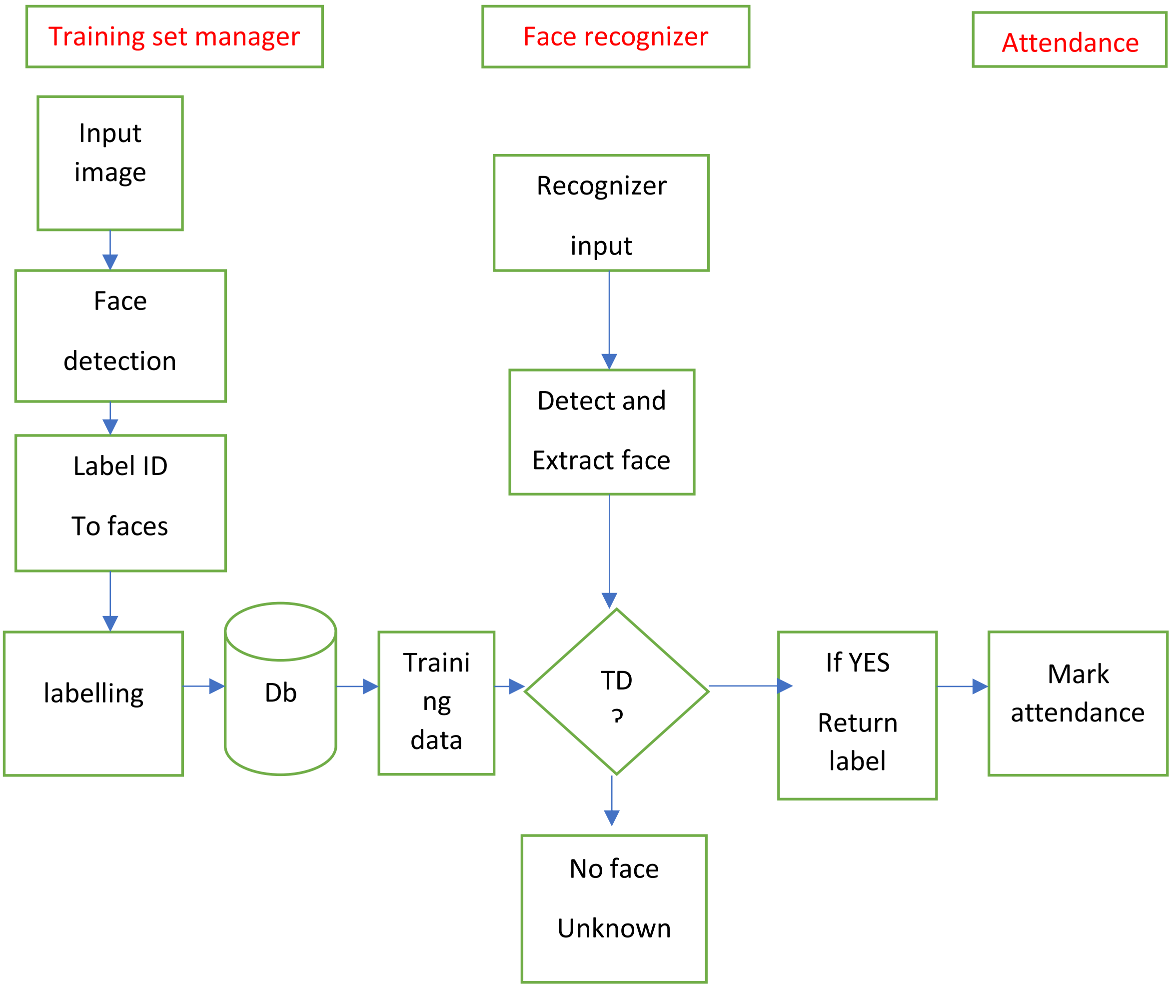


FIGURE 11 THE LOGICAL DESIGN OF THE DESKTOP MODULE SUBSYSTEMS

3.3 Functions of the two Sub –Systems

The functionalities of the components are depicted in the block diagrams of figure 11. The face recognizer system will consist of two major components i.e. the training set manager and the face recognizer. These two components will share the Faces database, the image acquisition and the face detector components; as they are common in their functionality.

We will therefore partition the system in to two subsystems and have their detailed logical designs to be implemented.

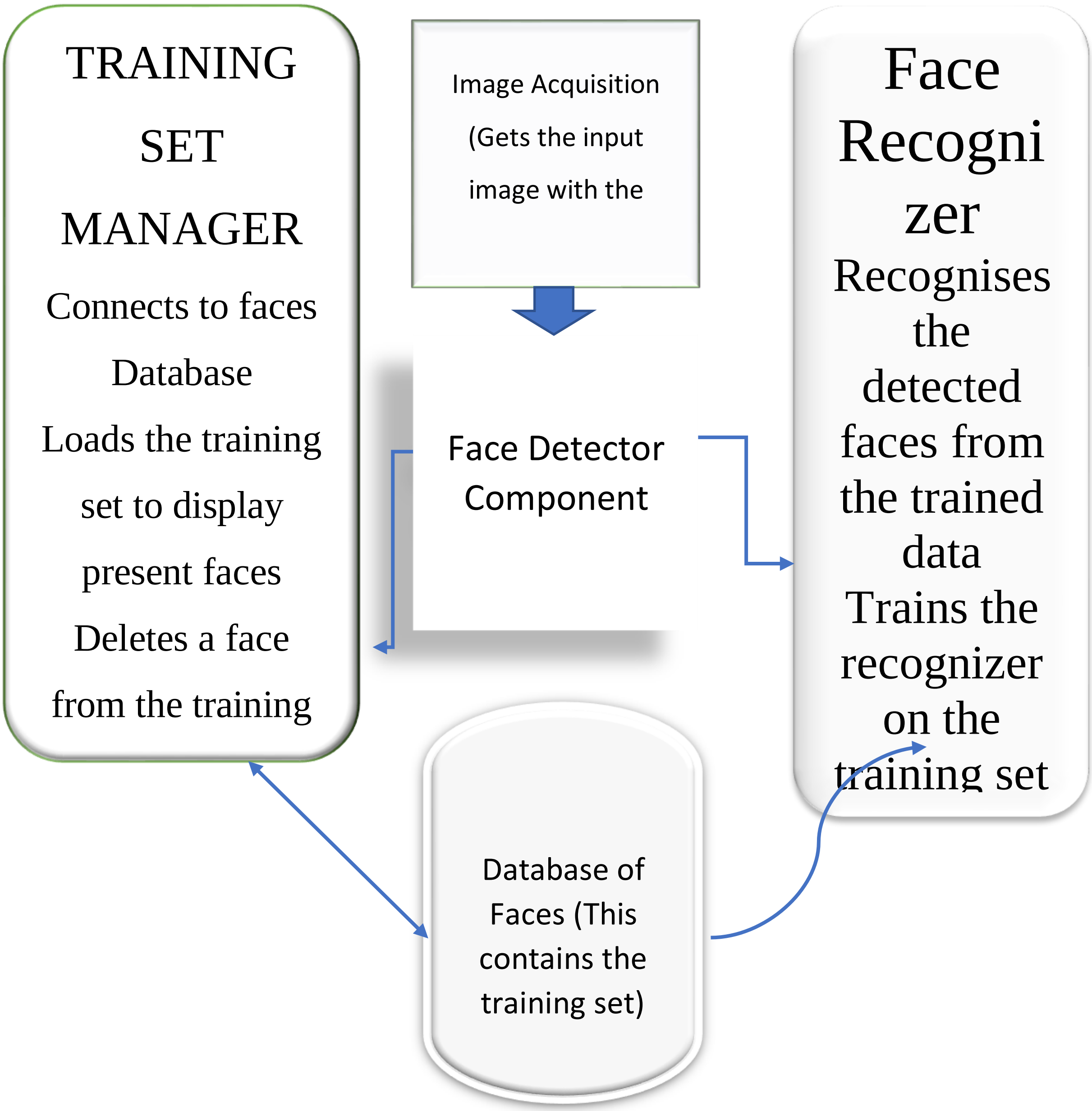


FIGURE 12 A BLOCK DIAGRAM SHOWING FUNCTIONS OF THE COMPONENTS

4. Viola- Jones Algorithm for Face Detection

In 2004 an article by Paul Viola and Michael J. Jones titled "Robust Real-Time Face Detection" was publish in the International Journal of Computer Vision. The algorithm presented in this article has been so successful that today it is very close to being the de facto standard for solving face detection tasks. This success is mainly attributed to the relative simplicity, the fast execution and the remarkable performance of the algorithm.

4.1 The Scale Invariant Detector

The first step of the Viola-Jones face detection algorithm is to turn the input image into an integral image. This is done by making each pixel equal to the entire sum of all pixels above and to the left of the concerned pixel. This is demonstrated in Figure:

1	1	1
1	1	1
1	1	1

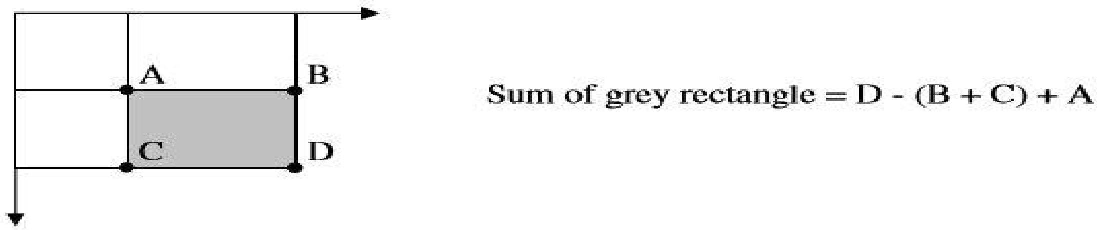
Input image

1	2	3
2	4	6
3	6	9

Integral image

FIGURE 13 THE INTEGRAL IMAGE

This allows for the calculation of the sum of all pixels inside any given rectangle using only four values. These values are the pixels in the integral image that coincide with the corners of the rectangle in the input image. This is demonstrated in Figure 13.



Since both rectangle B and C include rectangle A, the sum of A has to be added to the calculation.

FIGURE 14 SUM CALCULATION

It has now been demonstrated how the sum of pixels within rectangles of arbitrary size can be calculated in constant time. The Viola-Jones face detector analyses a given sub-window using features consisting of two or more rectangles. The different types of features are shown in

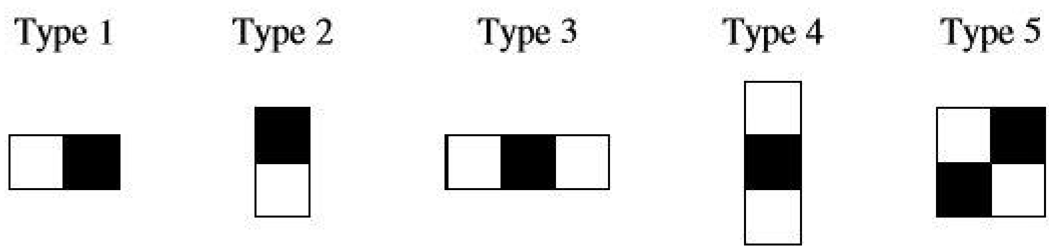


FIGURE 15 THE DIFFERENT TYPES OF FEATURES

Figure 15.

Each feature results in a single value which is calculated by subtracting the sum of the white rectangle(s) from the sum of the black rectangle(s).

Viola-Jones have empirically found that a detector with a base resolution of 24*24 pixels gives satisfactory results. When allowing for all possible sizes and positions of the features in Figure a total of approximately 160.000 different features can then be constructed. Thus, the number of possible features vastly outnumbers the 576 pixels contained in the detector at base resolution. These features may seem overly simple to perform such an advanced task as face detection, but what the features lack in complexity they most certainly have in computational efficiency.

One could understand the features as the computer's way of perceiving an input image. The hope being that some features will yield large values when on top of a face. Of course, operations could also be carried out directly on the raw pixels, but the variation due to different pose and individual characteristics would be expected to hamper this approach. The goal is now to smartly construct a mesh of features capable of detecting faces and this is the topic of the next section.

4.2 The Modified AdaBoost Algorithm

AdaBoost is a machine learning boosting algorithm capable of constructing a strong classifier through a weighted combination of weak classifiers. (A weak classifier classifies correctly in only a little bit more than half the cases.) To match this terminology to the presented theory each feature is considered to be a potential weak classifier.

An important part of the modified AdaBoost algorithm is the determination of the best feature, polarity and threshold. There seems to be no smart solution to this problem and Viola-Jones suggest a simple brute force method. This means that the determination of each new weak classifier involves evaluating each feature on all the training examples in order to find the best performing feature. This is expected to be the most time-consuming part of the training procedure.

The best performing feature is chosen based on the weighted error it produces. This weighted error is a function of the weights belonging to the training examples. As seen in Figure 15 part 4, the weight of a correctly classified example is decreased and the weight of a misclassified example is kept constant. As a result, it is more ‘expensive’ for the second feature (in the final classifier) to misclassify an example also misclassified by the first feature, than an example classified correctly. An alternative interpretation is that the second feature is forced to focus harder on the examples misclassified by the first. The point being that the weights are a vital part of the mechanics of the AdaBoost algorithm. With the integral image, the computationally efficient features and the modified AdaBoost algorithm in place it seems like the face detector is ready for implementation, but Viola-Jones have one more ace up the sleeve.

4.3 The Cascaded Classifier

The basic principle of the Viola-Jones face detection algorithm is to scan the detector many times through the same image – each time with a new size. Even if an image should contain one or more faces it is obvious that an excessive large amount of the evaluated sub-windows

would still be negatives (non-faces). This realization leads to a different formulation of the problem:
Instead of finding faces, the algorithm should discard non-faces.

The thought behind this statement is that it is faster to discard a non-face than to find a face. With this in mind a detector consisting of only one (strong) classifier suddenly seems inefficient since the evaluation time is constant no matter the input. Hence the need for a cascaded classifier arises.

The cascaded classifier is composed of stages each containing a strong classifier. The job of each stage is to determine whether a given sub-window is definitely not a face or maybe a face. When a sub-window is classified to be a non-face by a given stage it is immediately discarded. Conversely a sub-window classified as a maybe-face is passed on to the next stage in the cascade. It follows that the more stages a given sub-window passes, the higher the chance the sub-window actually contains a face. The concept is illustrated with two stages in Figure 16.

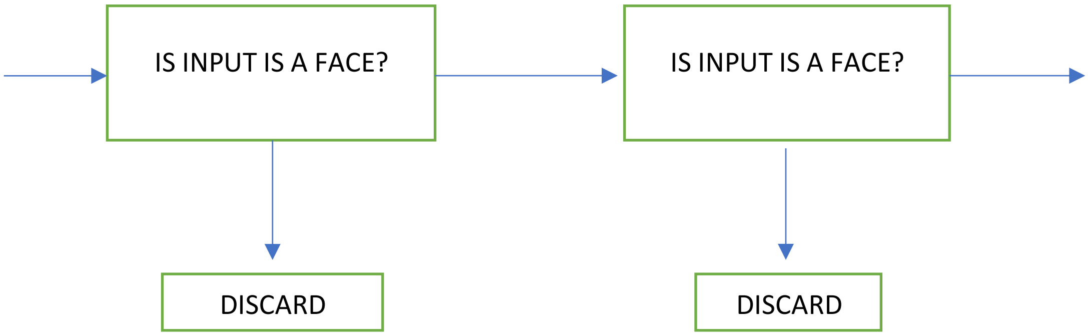


FIGURE 16 THE CASCADE CLASSIFIER

In a single stage classifier, one would normally accept false negatives in order to reduce the false positive rate. However, for the first stages in the staged classifier false positives are not considered to be a problem since the succeeding stages are expected to sort them out.

Therefore, Viola-Jones prescribe the acceptance of many false positives in the initial stages. Consequently, the number of false negatives in the final staged classifier is expected to be very small.

Viola-Jones also refer to the cascaded classifier as an attentional cascade. This name implies that more attention (computing power) is directed towards the regions of the image suspected to contain faces. It follows that when training a given stage, say n, the negative examples should of course be false negatives generated by stage n-1.

5. Technology and Tools

To implement this project, we need to use some technology and tools available, so here we introduce to tools we used:

5.1 Python Introduction

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

5.1.1 What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

5.1.2 Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

- Python can be treated in a procedural way, an object-orientated way or a functional way.

5.2 Python libraries

5.2.1 OpenCV

OpenCV was started at Intel in 1999 by Gary Bradsky and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel's Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle who won 2005 DARPA Grand Challenge. Later its active development continued under the support of Willow Garage, with Gary Bradsky and Vadim Pisarevsky leading the project. Right now, OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day-by-day.

Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations.

OpenCV-Python is the Python API of OpenCV. It combines the best qualities of OpenCV C++ API and Python language.

5.2.2 OpenCV-Python

Python is a general-purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation.

And the support of NumPy makes the task easier. NumPy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from NumPy arrays. So whatever operations you can do in NumPy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports NumPy can be used with this.

So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

5.2.3 NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. Explaining the basics of NumPy such as its architecture and environment. An introduction to Matplotlib is also provided.

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

5.2.3.1 Operations using NumPy

Using NumPy, a developer can perform the following operations −

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

5.2.3.2 NumPy – A Replacement for MATLAB

NumPy is often used along with packages like **SciPy** (Scientific Python) and **Mat−plotlib** (plotting library). This combination is widely used as a replacement for MATLAB, a popular platform for technical computing. However, Python alternative to MATLAB is now seen as a more modern and complete programming language.

It is open source, which is an added advantage of NumPy.

5.2.4 Tkinter

Tkinter is a Python binding to the Tk GUI toolkit. Tk is the original GUI library for the Tcl language. Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. There are several other popular Python GUI toolkits. Most popular are wxPython, PyQt, and PyGTK.

5.2.5 CSV

Python has a vast library of modules that are included with its distribution. The csv module gives the Python programmer the ability to parse CSV (Comma Separated Values) files. A CSV file is a human readable text file where each line has a number of fields, separated by commas or some other delimiter.

You can think of each line as a row and each field as a column. The CSV format has no standard, but they are similar enough that the csv module will be able to read the vast majority of CSV files. You can also write CSV files using the csv module.

5.2.6 Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyse.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

5.2.6.1 Key Features of Pandas

- Fast and efficient Data Frame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and sub setting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

5.3 Computer Vision Libraries

5.3.1 Theano

- Maintained by Montréal University group.
- Pioneered the use of a computational graph.
- General machine learning tool - Use of Lasagne and Keras.
- Very popular in the research community, but not elsewhere.

5.3.2 Caffe

Caffe is a well-known and widely used machine-vision library that ported Matlab's implementation of fast convolutional nets to C and C++. Caffe is not intended for other deep-learning applications such as text, sound or time series data. Like other frameworks mentioned here, Caffe has chosen Python for its API. Caffe perform image classification with convolutional nets, which represent the state of the art. In contrast to Caffe, Deeplearning4j offers parallel GPU support for an arbitrary number of chips, as well as many, seemingly trivial, features that make deep learning run more smoothly on multiple GPU clusters in parallel. While it is widely cited in papers, Caffe is chiefly used as a source of pre-trained models hosted on its Model Zoo site.

5.3.3 TensorFlow

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.

TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

5.4 Requirements to Run the Application:

Supportive Operating Systems: The supported Operating Systems for client include: Windows 2010, windows 2008, windows 2007.

5.4.1 Software Requirements

The Software Requirements in this project include:

a. Python

b. OpenCV framework

c. MS Excel.

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 7 million. The library is used extensively in companies, research groups and by governmental bodies. As an asynchronous event driven framework.

5.5 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

Components	Minimum	Recommended
Processor	Intel Core i3-2100 2nd generation	Intel Core i7 5th generation
RAM	4GB	8GB
Camera	HD 720p Webcam	Full HD 1080p Webcam
Disk	128Gb	512Gb

FIGURE 17 HARDWARE REQUIREMENTS

6. GUI CONSTRUCTING

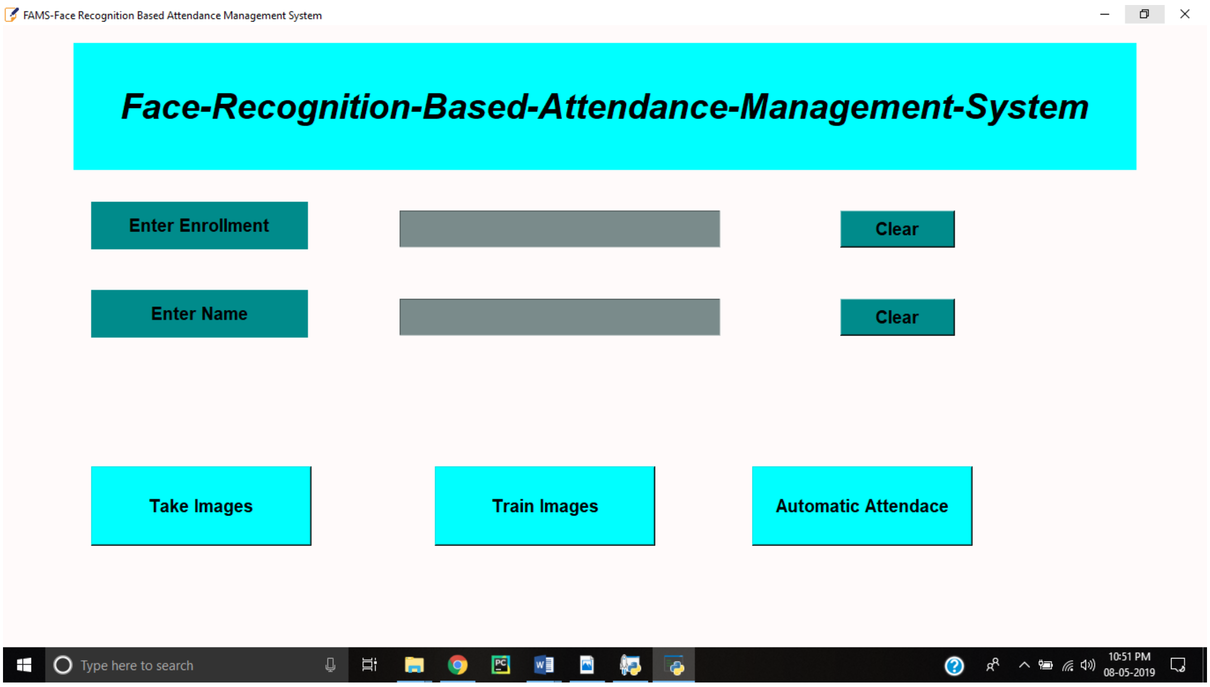


FIGURE 18 GUI HOME PAGE

Automated Attendance Management System has the following functions:

- Take Images - Clicking this function create dataset of a person and it clicks 200 images of a person.
- Train Images - After creating dataset this function trains the dataset images.
- Automatic Attendance - This function recognizes the person and marks its attendance in excel sheet.
- Enter enrollment- This function receives unique ID of person.
- Enter Name- This function receives name belongs to particular ID.

7. CODE-IMPLEMENTATION

Code for Face Detection Attendance System is implemented with python language and openCV library.

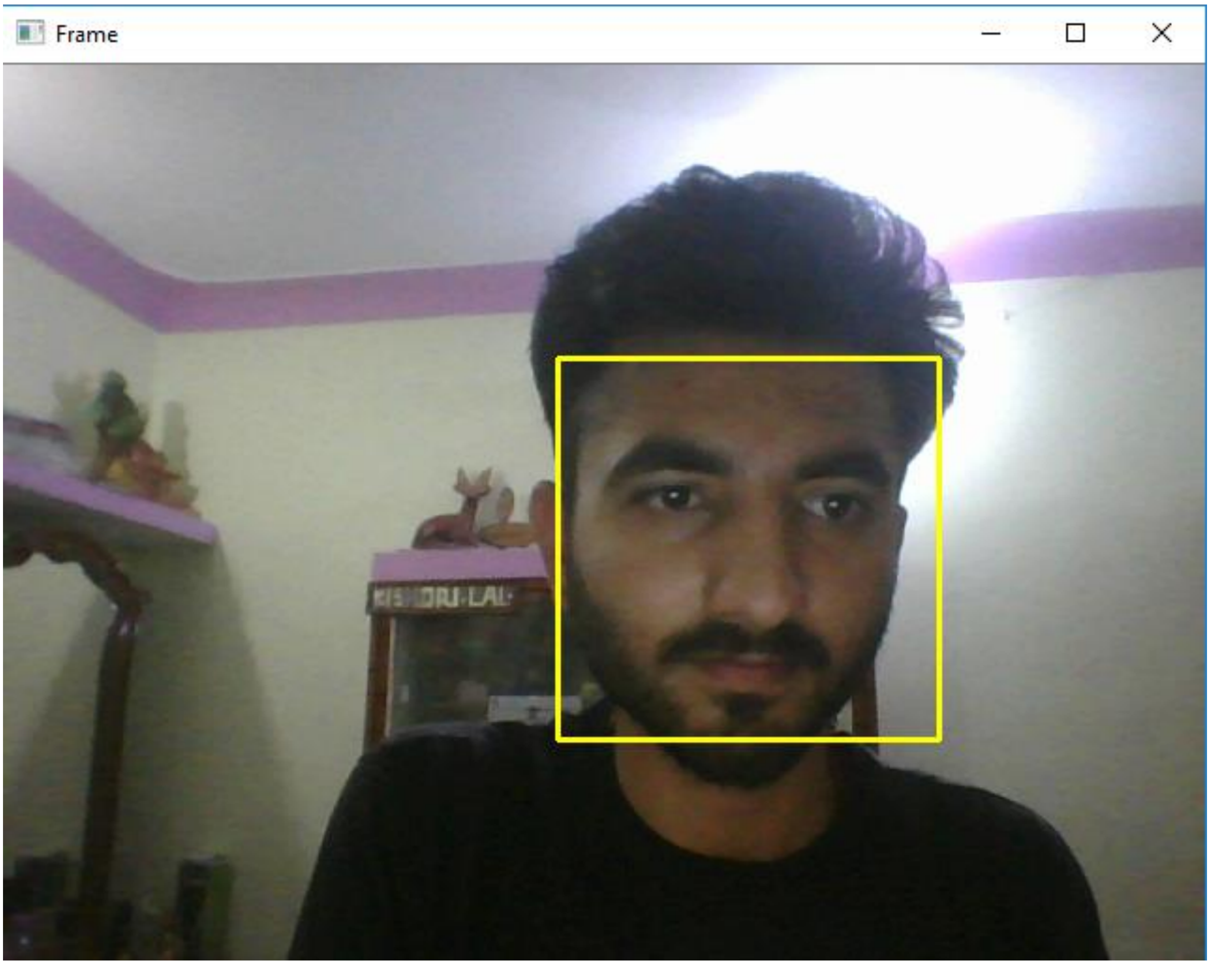


FIGURE 19 TAKING IMAGES FOR DATASET

7.1 Implementation of dataset creation

The dataset creation is first operation here in above figure it is detecting the face of a person in front of the camera. The program of data set creation will capture 200 images of the person with its unique Id and name entered in the respective fields.

7.2 Algorithm for Creating Dataset

```
#creating paths for dataset
def assure_path_exists(path):
dir = os.path.dirname(path)
if not os.path.exists(dir):
os.makedirs(dir)
face_id=input('enter your id')
# Start capturing video
vid_cam = cv2.VideoCapture(0)
# Detect object in video stream using Haarcascade Frontal Face
face_detector = cv2.CascadeClassifier('haarcascade_profileface.xml')
Ids.append(Id)
return faceSamples,Ids # Capture video frame
_, image_frame = vid_cam.read()
# Convert frame to grayscale
gray = cv2.cvtColor(image_frame, cv2.COLOR_BGR2GRAY)
# Detect frames of different sizes, list of faces rectangles
faces = face_detector.detectMultiScale(gray, 1.3, 5)
```

7.3 Algorithm for Trainer

```
def getImagesAndLabels(path):
#get the path of all the files in the folder
imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
#create empth face list faceSamples=[]
#create empty ID list
Ids=[]
#now looping through all the image paths and loading the Ids and the images
for imagePath in imagePaths:
```

```
#loading the image and converting it to gray scale
pilImage=Image.open(imagePath).convert('L')
#Now we are converting the PIL image into numpy array
imageNp=np.array(pilImage,'uint8')
#getting the Id from the image
Id=int(os.path.split(imagePath)[-1].split(".")[1])
# extract the face from the training image sample
faces=detector.detectMultiScale(imageNp)
#If a face is there then append that in the list as well as Id of it
for (x,y,w,h) in faces:
faceSamples.append(imageNp[y:y+h,x:x+w])
```

7.4 Implementation of Recognizer and Attendance

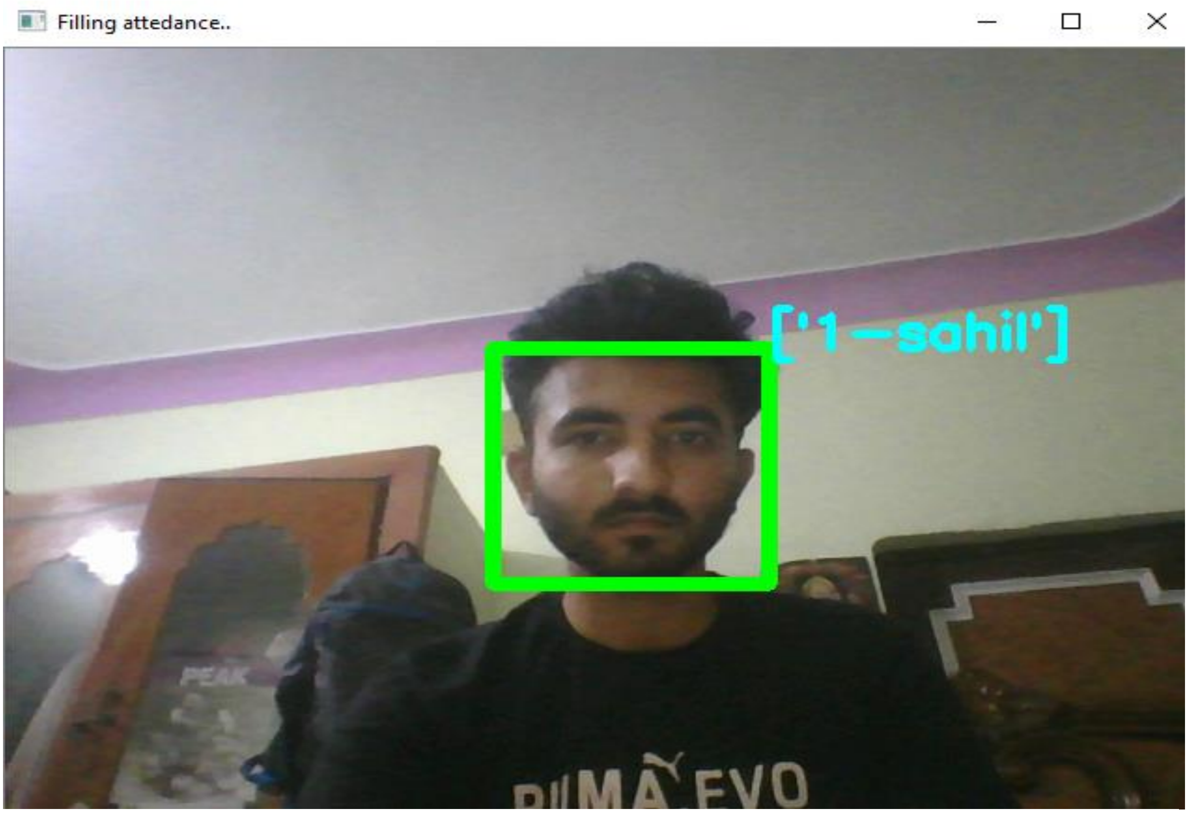


FIGURE 20 MARKING ATTENDANCE 1

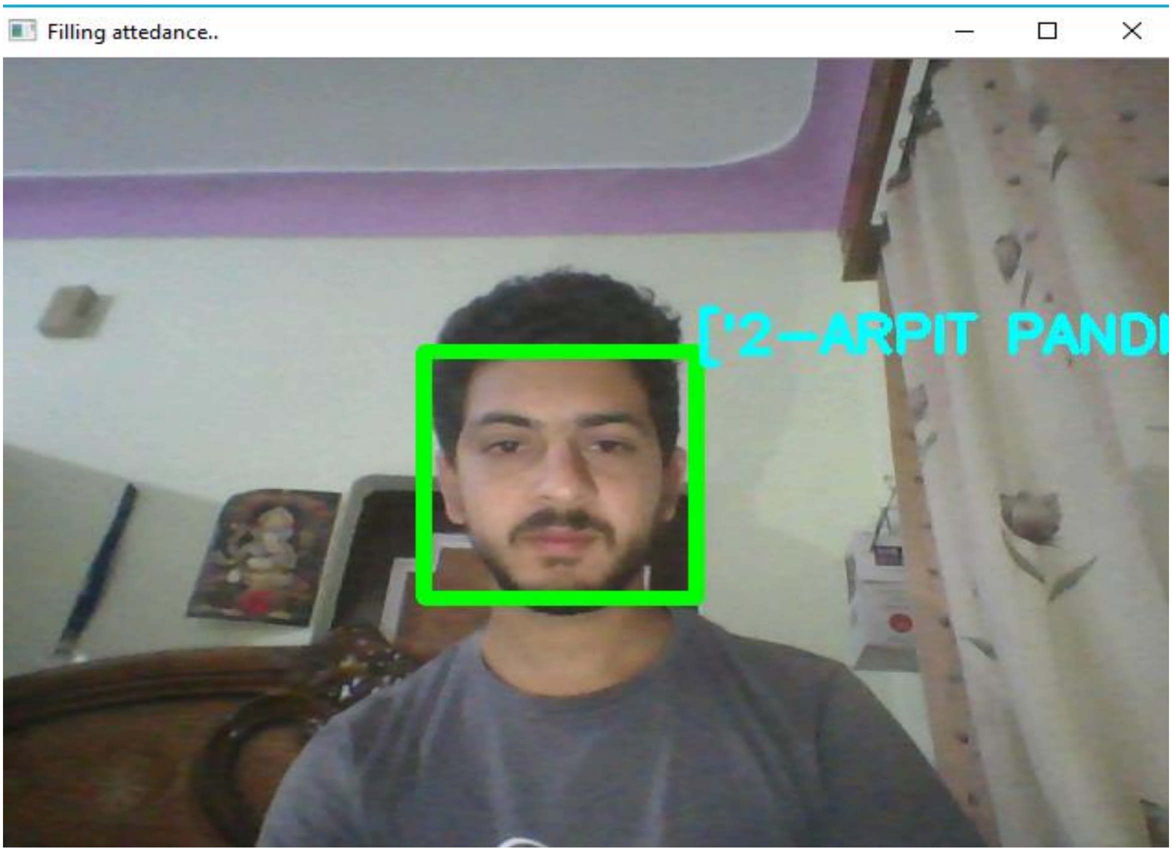


FIGURE 21 MARKING ATTENDANCE 2

8. LIMITATIONS

- System faces recognition issues if the light conditions are poor.

  If lighting conditions are poor than it affects the recognition as it recognizes wrong and sometimes don’t recognize in poor lighting conditions, so while creating dataset and recognizing it needs proper lighting.

- Webcam and processor with high specification is required as system consumes a lot of resources.

  As this project is doing image processing, so to run it smoothly high specification is required and graphic card is must to run it smoothly otherwise it will be too slow and may hang during its execution.

9. Conclusion and Recommendation

It can be concluded that a reliable, secure, fast and an efficient class attendance management system has been developed replacing a manual and unreliable system. This face detection and recognition system will save time, reduce the amount of work done by the administration and replace the stationery material currently in use with already existent electronic equipment.

There is no need for specialized hardware for installing the system as it only uses a computer and a camera. The camera plays a crucial role in the working of the system hence the image quality and performance of the camera in real time scenario must be tested especially if the system is operated from a live camera feed.

The system can also be used in permission-based systems and secure access authentication (restricted facilities) for access management, home video surveillance systems for personal security or law enforcement.

The major threat to the system is Spoofing. For future enhancements, anti- spoofing techniques like eye blink detection could be utilized to differentiate live from static images in the case where face detection is made from captured images from the classroom. From the overall efficiency of the system i.e. 83.1% human intervention could be called upon to make the system fool proof. A module could thus be included which lists all the unidentified faces and the lecturer is able to manually correct them.

Future work could also include adding several well-structured attendance registers for each class and the capability to generate monthly attendance reports and automatically email them to the appropriate staff for review.

10. FUTURE SCOPE

The research work has implemented a face recognition system by using PCA which is eigenvector based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. By implementing PCA the proposed Face Recognition System supplies the user with a lower-dimensional picture, a "shadow" of this object when viewed from its most informative viewpoint.

The algorithm has been tested with multiple students in the scene and also captured faces at different angles in the scene. The algorithm delivers quite good results but there is a room to improve the algorithm performance in case of large number of students and also in case of faces captured in a dark environment, so proposed system can be extended in the future to cover this aspect. The efficiency of the algorithm also can be increased further so there is also a room for future work in this area. This system can be enhanced further in terms of achieving more efficiency by ease of analysis of patterns in the data.

11. REFERENCES

- Technical Paper on â_ Image Based Intelligent Attendance Logging System, IEEE Technical Paper by Hary Oktavianto, Hsu Gee-Sern Chung Sheng-Luen, June 30-July 2, 2012, Dalian,China.
- Z. Zhou, X. Chen, Y. â“C. Chung, Z. He, T. X. Han, and J. M. Keller, "Activity analysis,summarization, and visualization for indoor human activity monitoring,â IEEE Transactions on Circuits and Systems for Video Technology, Volume 18, Issue 11, pp. 1489-1498, 2008.
- OpenCV. Available: http://sourceforge.net/projects/ opencvlibrary/.
- Activity zones for context-aware computing, Lecture Notes in Computer Science, Volume 2864/2003, pp. 90-106, 2003.
- International Journal of Innovative Research in Computer and communication Engineering(Vol.3,Issue 8,Aug 2015)(Automated Attendance System Using Face Recognition.)
- International Journal of Computer Applications (0975-8887) Volume 98-No.20,July 2014(Adjacency Matrix based Face Recognition Approach.)
- International Journal of Information Technology and Knowledge Management July-December 2012, Volume 5, No. 2, pp. 361-3639(Face Recognition Techniques: Classification and Comparisons.)
- Face Recognition using Line Edge Map Vipin Kumar and Mohit Mehta Department of Electronics and communication, Punjab College of engineering and technology, India Accepted 01 January 2014, Available online 10 January 2014, Vol.2 (Jan/Feb 2014 issue)
- International Journal of Advanced Computer Research (ISSN (Print): 2249-7277 ISSN (Online): 2277-7970) Volume-4 Number-4 Issue-17 December-2014 939 Study of Face Recognition Techniques Sangeeta Kaushik1*, R. B. Dubey2 and Abhimanyu Madan3