A PRELIMINARY REPORT ON

REAL TIME FACE RECOGNITION AUTOMATIC STUDENTS ATTENDENCE MANAGEMENT

Project report in partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology

In

Computer Science And Technology

Submitted By

Saumyajit Das	Enrollment No. 12023002022134
Rig Ghosh	Enrollment No. 12023002022239
Sunanda Patra	Enrollment No. 12023002022241
Sreyan Datta	Enrollment No. 12023002022157
Sania Khandakar	Enrollment No. 12023002022238

Under the guidance of

Prof. Anirban Ganguly

Department of Computer Science and Technology & Computer Science and Information Technology (CST & CSIT)

UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160.

CERTIFICATE

This is to certify that the project titled **A PRELIMINARY REPORT ON REAL TIME FACE RECOGNITION AUTOMATIC STUDENTS ATTENDENCE MANAGEMENT** submitted by **Saumyajit Das(University Roll No.53),Rig Ghosh(University Roll No. 75)**,**Sunanda Patra(University Roll No. 77), Sania Khandakar(University Roll No.74) and Sreyan Datta(University Roll No. 75)** students of **UNIVERSITY OF ENGINEERING and MANAGEMENT, KOLKATA,** in partial fulfillment of requirement for the degree of Bachelor of Computer Science and Technology, is a bonafide work carried out by them under the supervision and guidance of Prof. Anirban Ganguly during 3rd Semester of academic session of 2023-27. The content of this report has not been submitted to any other university or institute. I am glad to inform that the work is entirely original and its performance is found to be quite satisfactory.

_____

Signature of Guide

_____

Signature of Head of the Department

ACKNOWLEDGEMENT

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide **Prof. Anirban Ganguly** of the Department of CST & CSIT, UEM, Kolkata, for his/her wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Saumyajit Das

Rig Ghosh

Sunanda Patra

Sreyan Datta

Sania Khandakar

TABLE OF CONTENTS

INTRODUCTION

Introduction:

The purpose of the attendance monitoring system using face recognition is to ease the attendance process which consumes lot of time and efforts , it is a convenient and easy way for students and teacher. The system will capture the images of the students and using face recognition algorithm mark the attendance in the sheet. This way the class-teacher will get their attendance marked without actually spending time in traditional attendance marking.

The identification process to determine the presence of a person in a room or building is currently one of the routine security activities. Every person who will enter a room or building must go through several authentication processes first, that later these information's can be used to monitor every single activity in the room for a security purpose. Authentication process that is being used to identify the presence of a person in a room or building still vary. The process varies from writing a name and signatures in the attendance list, using an identity card, or using biometric methods authentication as fingerprint or face scanner.

Motivation:

Nowadays many educational institutes are using a manual monitoring system and most of the time they accidentally loss their attendance sheet so that they cannot properly monitor the attendance of their students .Therefore it is important to design software which will help these institutes to mark the attendance of the students by face recognition which will save their time.

Problem definition:

Automatic Face recognition attendance management using Machine Learning

A real-world student attendance system which recognizes face of student and attendance of the respective student will be marked automatically on excel sheet.

LITERATURE SURVEY

2. Literature survey :

Table 1.LITERATURE SURVEY

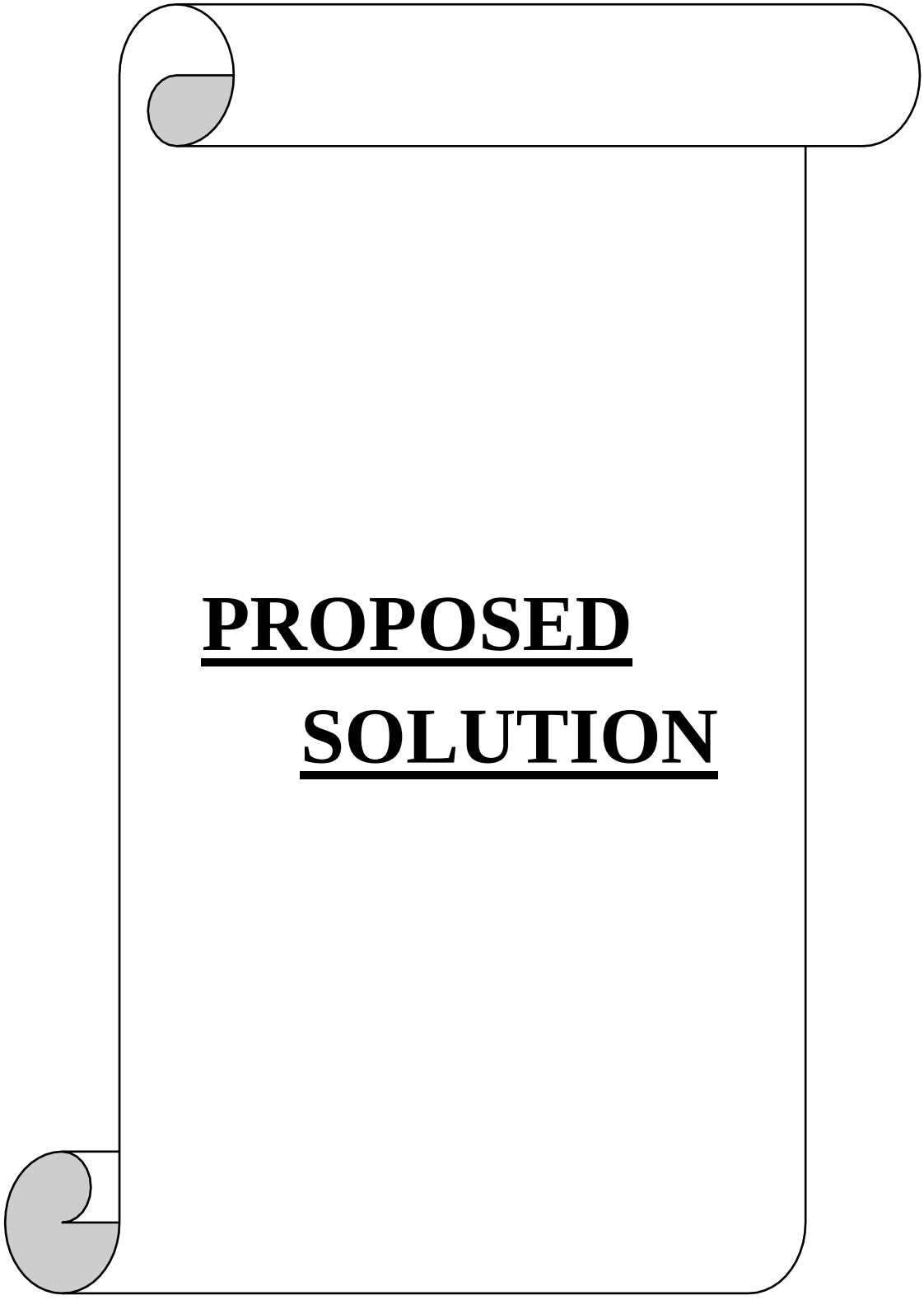
Sr. No.	Paper Name/Author	Publication Year	Methodology	Conclusion
1.	Monica C., Nithya.R, Prarthana.M, Sonika.S.V, Dr.M.Ramakrishna	2017	The design is expressed in sufficient detail so as to enable all the developers to understand the underlying architecture of Attendance system.	The Existing system is a manual entry for the Admin and also Faculty. Here the attendance will be carried out in the hand written registers. Maintaining the records for the Faculty is a tedious job
2.	Abdoulrahmaine Mohammad, Mohammad Elmi Hassan, Muslim Musa	2018	In this we study it capable of eliminating time wasted during manual collection of attendance and for the educational administration	The new system has been designed as per the user requirements so as to fulfill almost all them. -User friendly -Report Generation -Less paper work

PROBLEM STATEMENTS

3.1 Problem definition:

Automatic Face recognition attendance management using Machine Learning

A real-world student attendance system which recognizes face of student and attendance of the respective student will be marked automatically on excel sheet.



PROPOSED SOLUTION

4.Proposed Solution

4.1 INTRODUCTION

4.1.1 Project Scope

The scope of the system is to reduce the time of the teacher as well as student which they wasted by doing traditional attendance.

4.1.2 User Classes and Characteristics

Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.

4.1.3 Assumptions and Dependencies

This document will provide a general description of our project, including user

requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project such as interface, functional requirements and performance requirement

FUNCTIONAL REQUIREMENTS

Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail.

4.2 EXTERNAL INTERFACE REQUIREMENTS

4.2.1 User Interfaces

The user interface for the software shall be compatible to any Android version by which user can access to the system. The user interface shall be implemented using any tool or software package like Android Studio, MYSQL etc.

4.2.2 Hardware Interfaces

Since the application must run over the internet, the hardware shall require to connect internet to the hardware which is android device for the system.

4.2.3 Software Interfaces

This system is a Single-user, multi-tasking environment. It enables the user to interact with the server and attain interact with the server to show the animal information also leaves a record in the inbuilt database. It uses Java and android as the front-end programming tool and MySQL as the back-end application tool.

4.2.4 Communication Interfaces

The e-store system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

NON-FUNCTIONAL REQUIREMENTS:

4.2.5 Performance Requirements

- System can produce results faster on 2GB/4GB of RAM.
- It may take LESS time for peak loads at main node.
- The system will be available 100% of the time. Once there is a fatal error, the system will provide understandable feedback to the user.

4.2.6 Safety and Security Requirements

- The system is designed in modules where errors can be detected and fixed easily.

4.2.7 Software Quality Attributes

- **Reliability:** The Client machine will change the status of data indicating successful data transmission.
- **Usability:** The application should be easy to use through interactive interface.
- **Maintainability:** The system will be developed using the standard software development conventions to help in easy review and redesigning of the system.
- **Support ability:** The system will be able to support to transfer different types of SQL queries.

Portability: This software is portable to any system with the requirements specified. There must also be a server where the database can be set-up.

SYSTEM REQUIREMENTS:

4.2.8 Software Requirements Platform :

Operating System : Windows OS

Platform: Android Studio

Programming Language : PYTHON

4.2.9 Hardware Requirements

Processor: INTEL Pentium 4 Processor Core

Hard Disk: 40 GB (min)

RAM: 256 MB or higher

4.3 Basic flowchart diagram

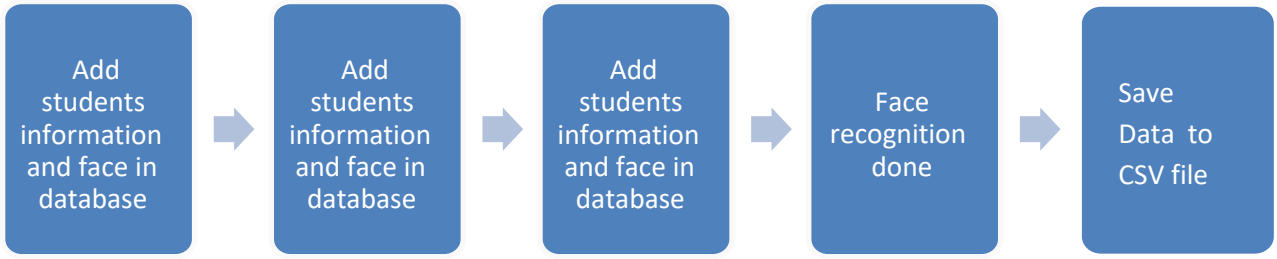
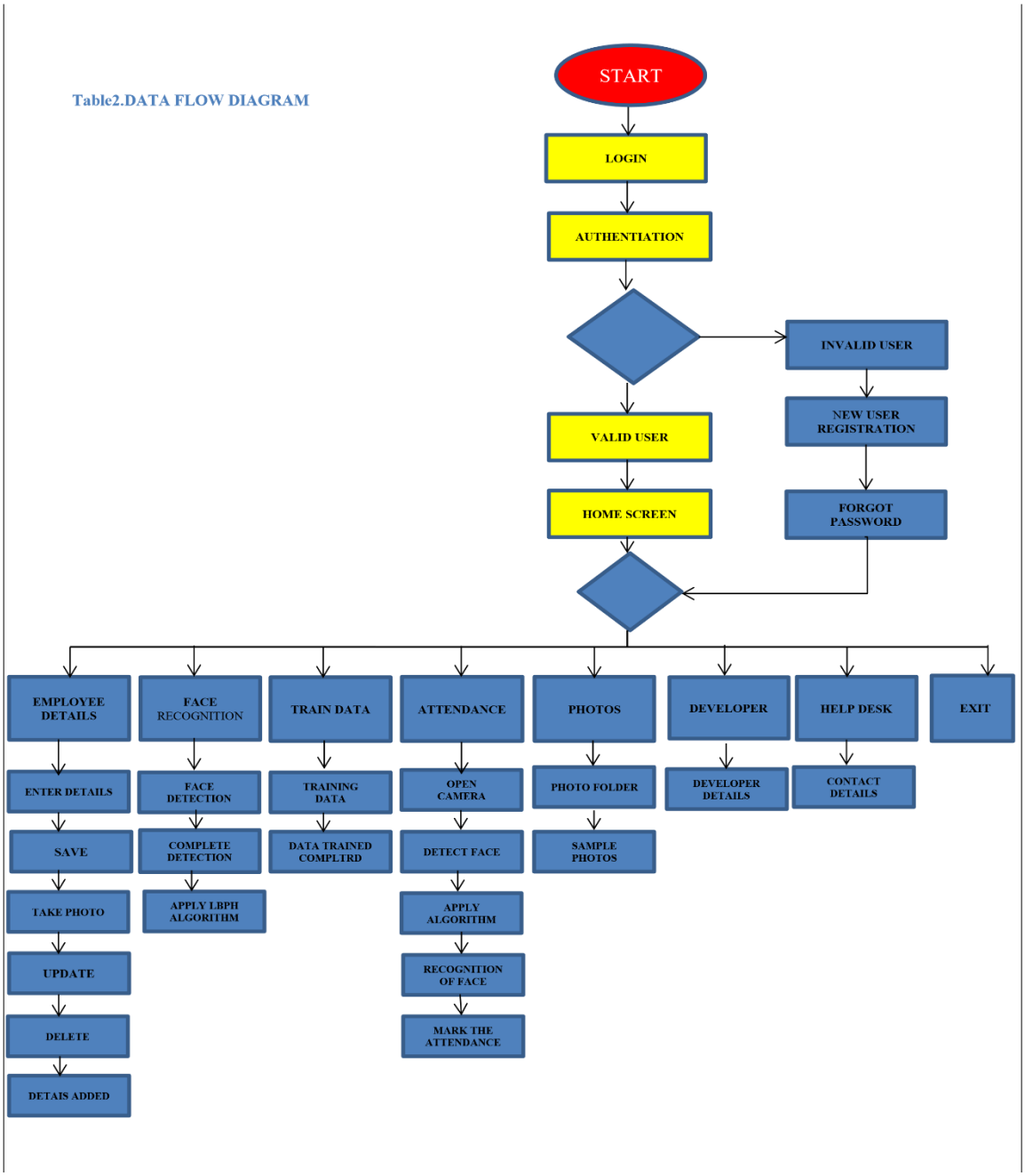


fig 1 .Flow chart

Table2.DATA FLOW DIAGRAM



SYSTEM DESIGN

5.1 **HAAR CASCADE ALGORITHM**

The core basis for Haar classifier object detection is the Haar-like features. These features, rather than using the intensity values of a pixel, use the change in contrast values between adjacent rectangular groups of pixels. The contrast variances between the pixel groups are used to determine relative light and dark areas. Two or three adjacent groups with a relative contrast variance form a Haar-like feature. Haar-like features as shown in figure are used to detect an image. Haar features can easily be scaled by increasing or decreasing the size of the pixel group being examined. This allows features to be used to detect objects of various sizes. The cascading of the classifiers allows only the sub-images with the highest probability to be analyzed for all Haar-features that distinguish an object. It also allows one to vary the accuracy of a classifier. One can increase both the false alarm rate and positive hit rate by decreasing the number of stages. The inverse of this is also true. Viola and Jones were able to achieve a 90% accuracy rate for the detection of a human face using only 100 simple features. Detecting human facial features, such as the mouth, eyes, and nose require that Haar classifier cascades first are trained. In order to train the classifiers, this gentle AdaBoost algorithm and Haar feature algorithms must be implemented. Fortunately, Intel developed an open-source library devoted to easing the implementation of computer vision related programs called Open Computer Vision Library (OpenCV). The OpenCV library is designed to be used in conjunction with applications that pertain to the field of HCI, robotics, biometrics, image processing, and other areas where visualization is important and includes an implementation of Haar classifier detection and training. Thus, with help of this algorithm system will detect the person's face in the video. Face of the person gets Green Square as an indication of detection process. As soon as the face gets detected user can paused the video and enters the data of detected person such as person's name, address, profession, criminal record if any. If the detected person has criminal record, then it can be defined as suspect. Check box option is given in the system where user can tick whether the person is suspect on not. This is the working of first module in which sample video is browsed and face is detected.

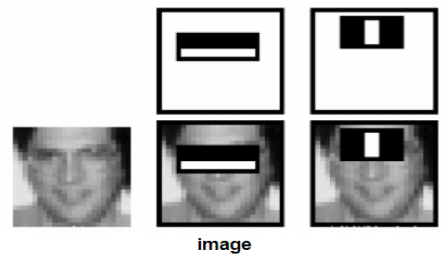


Fig 3. Haar Features

5.2 Local Binary Pattern Histogram (LBPH):

5.2.1.Introduction to LBPH algorithm

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameter's **radius** and **neighbors.**

5.1 Applying the LBP operation:

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameter's **radius** and **neighbors**.
The image below shows this procedure:

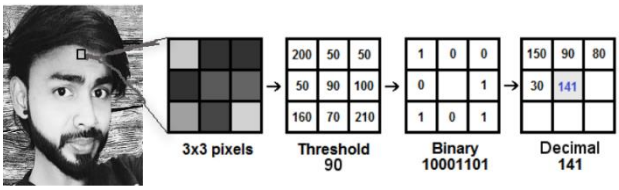


Fig 4. **Applying the LBP operation**

5.2. Performing the face recognition:

- In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.

- So, to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.

- We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: Euclidean distance, chi-square, absolute value, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

- So, the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a '**confidence**' measurement. **Note**: don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.

- We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

5.2.3 TRAINING THE ALGORITHM:

First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output.

Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.

5.2.4 APPLYING LBH OPERATIONS

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameter's **radius** and **neighbors**.

IMPORTANT POINTS

- Suppose we have a facial image in grayscale.
- We can get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).
- Then, we need to take the central value of the matrix to be used as the threshold.
- This value will be used to define the new values from the 8 neighbors.
  For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.

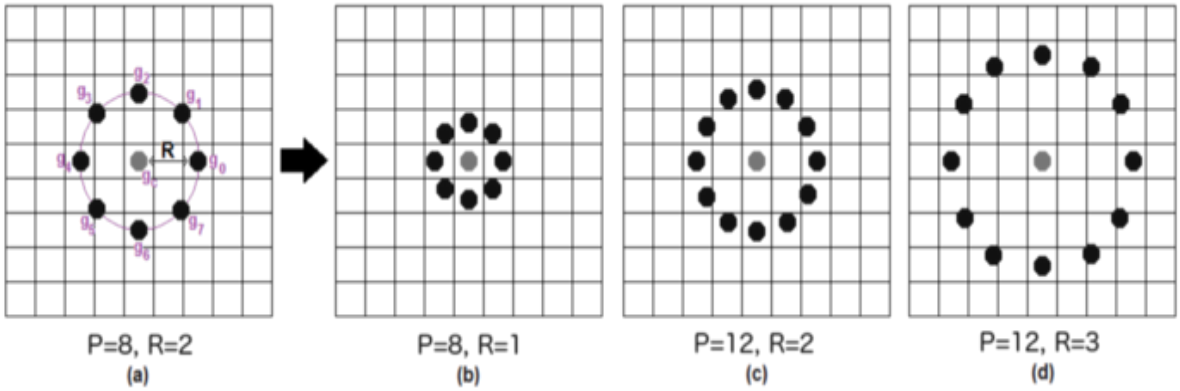
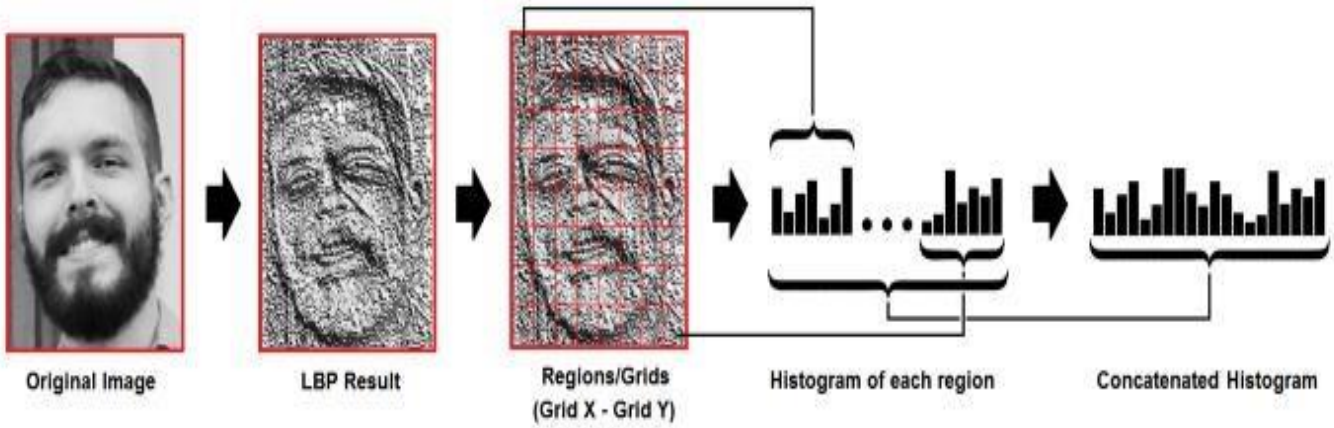


Fig.6 Radius of central pixel

5.4 EXTRACTING THE HISTOGRAM

Now, using the image generated in the last step, we can use the **Grid X** and **Grid Y** parameters to divide the image into multiple grids, as can be seen in the following image



Based on the image above, we can extract the histogram of each region as follows:

- As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.

- Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

No table of figures entries found.

In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.

So, to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.

We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: **Euclidean distance**, **chi-square**, **absolute value**, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

So, the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a '**confidence**' measurement. **Note**: don't be fooled

- 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.

- We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

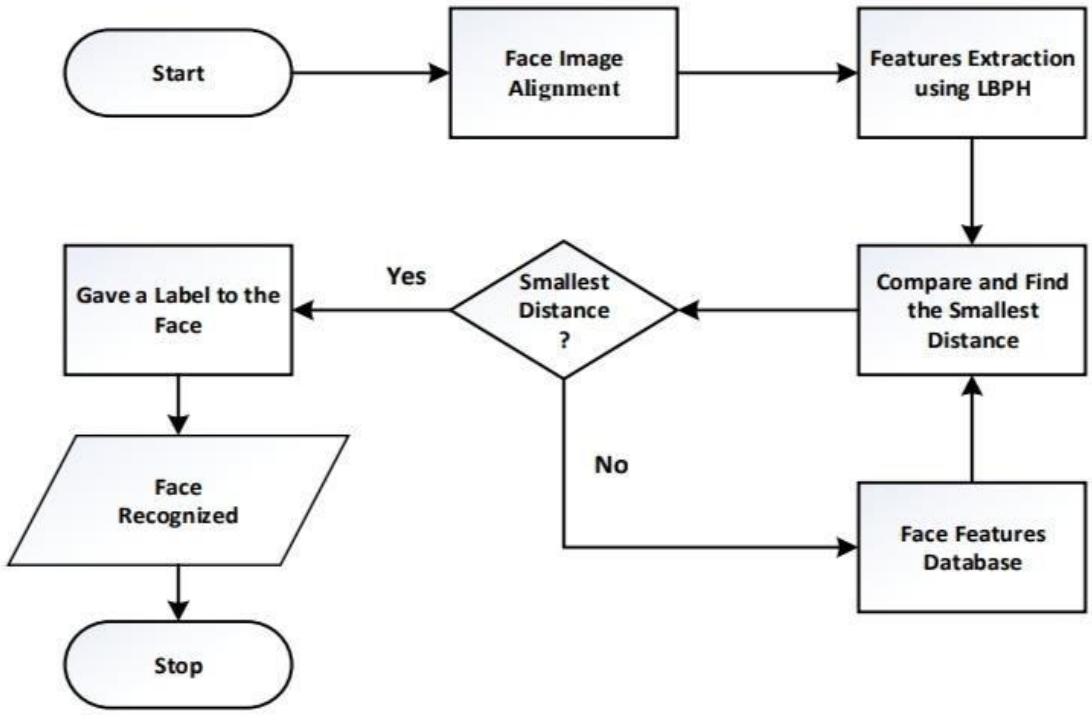


fig 8. Face alignment and feature extraction

CONCLUSION AND FUTURE WORK

6.OTHER SPECIFICATIONS

6.1 Advantages:

1. It is trouble-free to use.
2. It is a relatively fast approach to enter attendance.
3. Is highly reliable, approximate result from user .
4. Best user Interface .
5. Can obtain accuracy up to 85 percent.

6.2Limitations:

1. While training there generates nearly 100 of copies of sample image.
2. While dealing with high volume of data system required the powerful processor which is more costly

6.3Applications:

1. It is very useful for educational institutes to get attendance easily.
2. We can get attendance of students as well as teachers without doing conventional attendance

6.4 CONCLUSION AND FUTURE WORK

Conclusion:

- The Attendance Management System is developed using Machine Learning meets the objectives of the system which it has been developed. The system has reached a steady state where all bugs have been eliminated. The system is operated at a high level of efficiency. The system solves the problem. It was intended to solve as requirement specification.
- The system can recognize and identify the face well with an accuracy of 85 %, at a face distance 40 cm from the camera with adequate lighting.

Future Work:

We have planned to create the application for multiple college campuses and for the multiple schools.

Bibliography

7. Bibliography:

1) G. Satyanarayana Reddy , Rallabandi Srinivasu, Srikanth Reddy Rikkula , Vuda Sreenivasa Rao," Management Information System To Help Managers For Providing Decision Making In An Organization", International Journal of Reviews in Computing.

2) Learning OpenCV –Computer Vision with the OpenCV Library O'Reilly Publication

3) http://www.wordpress.org/

4) http://www.academia.edu/

5) http://www.stackoverflow.com/

6) http://www.iproject.com