IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-6, June 2019 E-ISSN: 2347-2693

Automatic Attendance Recording Using Face Recognition

Anjana Pandey[1*], Kanika Gupta[2], Shivang Agrawal[3]

[1,2,3]Department of IT, University Institute of Technology, RGPV, Bhopal, India

**Corresponding Author: anjanapandey@rgtu.net, Tel.: +91 9407278953*

DOI: https://doi.org/10.26438/ijcse/v7i6.266275 | Available online at: www.ijcseonline.org

Accepted: 11/Jun/2019, Published: 30/Jun/2019

Abstract- Face Recognition is an application in the computer vision industry .Many algorithms has been developed in order to achieve facial recognition with most accuracy. Face recognition is less sturdy than other biometric scanning systems like fingerprint and retina scanning. This report describes the face detection and recognition minor-project developed for the attendance system of the department where a student can view his attendance on the dates present and teacher can check the attendance and save it in pdf format. This project uses the technologies available in the OpenComputerVision (OpenCV) library in java interface and methods to implement them using . This project is accomplished using Haar-Cascades for face detection and Local binary pattern histograms(LBPH) for face recognition. The process is represented together with flow charts for every stage of the system. Next, the results are shown using screen-shots.

Keywords- Facial Recognition, Local Binary Patterns Histograms, Facial detection, Haar features, Computer Vision, OpenCV

I. INTRODUCTION

The primitive method of attendance system in school and colleges required the efforts of both student as well as lecturers where the attendance was taken manually. Nowadays instead of using this method a more advanced method like biometric scanning is used to mark the attendance of students. One such system is facial detection and recognition. This method reduces the efforts from human since the manual work is minimized.

This process involves the detection of the faces and then the recognition based on the faces recorded in the databases.

Face detection means recognition of the position of the face in the given frame. In face detection the features like nose, ears, hair, mouth etc. Some of the face detection techniques are Viola Jones , LBP, eigenfaces etc. The detected face is cropped and all this image face recognition is applied.

This can be done using Haar features, machine learning and many others. The recognition process needs a well trained data set and the recognition is done using the comparison of the received image to the images stored in this database. and award him with the attendance. First we discuss about the previous work conducted on this topic, then our proposed methodology followed by a result analysis.

Section I contains the introduction. Section II contain the related work Section III hardware and software requirements Section IV explains the methodology with Section V contain flow chart, Section VI describes results and discussion

Section VII contain the important code and Section VIII concludes research work with references.

II. RELATED WORK

Face Detection: Face detection locates the faces in a frame and extract the features of the faces from the face recognition algorithm

Face Recognition: the image that is obtained from the detection is already a cropped and what into a grayscale image. The basic role of phase recognition is to retrieve the characteristics from the image[1]

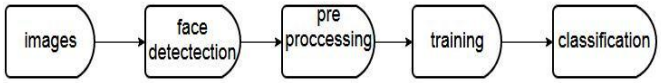


Figure 1. Classifier process

This report propagates through a step by step building of the system that will be able to capture, detect and then recognize the student. The face recognition systems can operate basically in two modes:

1. Verification or authentication of a facial image: the input image is compared to the image user in the data set present.

2. Identification or facial recognition: This input image is compared with all the images of the user and Returns the idea that is related to that user.

In this chapter, previous work based on the face detection and reorganization will be described along with their flaws to give an idea about the history and the approaches.

Description of the previous work.
Face recognition begin in 1977 with the first automatic System being introduced by kanade the feature vector of human Face. In 1983 Sirovich and Kirby news the PCA principal component analysis for the extraction of features in 1991 Eigenfaces was developed in 1994 local binary pattern analysis was developed and later improved by using histograms[2].

1996 fisherface was developed using linear discriminant analysis. There was an issue in Eigen method where for different illumination condition which was tackled by fisherface.

Viola and Jones introduced a new method for face detection using Haar cascade. There are different types of face recognition algorithms[3], for example:

Eigenfaces (1991)
Local Binary Patterns Histograms (LBPH) (1996)
Fisherfaces (1997)
Scale Invariant Feature Transform (SIFT) (1999)
Speed Up Robust Features (SURF) (2006)

1.Eigenfaces face recognizer
In eigenfaces the difference among the various features is considered as vectors known as eigenvectors. This vectors forms the base for the difference between the faces. Then the linear grouping of the eigenvalues is performed. Every face is then measured as a linear grouping of the eigenfaces. The recognizer extracts principal elements and trains itself. It keeps a record of which eigenvector belong to which person. Therefore when a new image is introduced to the algorithm the same process is followed.

1.Extract the principal components from the new picture.
2.Compare those features with the list of elements stored during training.
3. Find the ones with the best match.
4.Return the 'person' label associated with that best match component.

2.Fisherfaces face recognizer
the principal element from all the images that are stored for people combined .Doing this it doesn't focus on the feature that discriminate one individual from another.

Fisherface focuses on that principal component that will differentiate one person from other. Due to this the individuals component becomes more prominent over the other.

Fisherfaces (Jaiswal &al., 2011)(Morizet &al., 2006)(Belhumeur, Hespanha, & Kriegman, 1997) [4]also uses a holistic approach. This algorithm is a modification of Eigen-faces, thus also uses Principal Components Analysis. The main

modification is that Fisherfaces takes into consideration classes. As it has been said previously, Eigenfaces does not make the difference between two pictures from different classes during the training part. Each picture was affected by the total average. Fisherfaces uses the method Linear Discriminant Analysis in order to make the difference between two pictures from a different class. The aim is to minimize the variation in the class.

In fisherface algorithm if you have trained algorithm in well illuminated space, and the result is to be found in bad illuminated space the algorithm may lead to wrong results.

III. HARDWARE AND SOFTWARE REQUIREMENT

The following tools were used in the design of the prototype:
Java derby database: this database is the mostly used database for various java applications and also web applications amongst others. The derby database was choice used by the researcher for saving information about system users and user details. database is an open source and relatively simple to use.

Opencv: It is a computer vision library, which is equipped with various libraries to perform image-processing techniques. . It provides algorithms namely the Eigen Face, Fisher Faces well as LBPH (Local Binary Pattern Histograms) algorithm, which was implemented in this research. Furthermore, OpenCV have machine-learning algorithms that were used to train the classifier.

Web Camera: this was used for capturing images of individuals for testing the prototype
Java Programming Language: Java is the most secure language with inbuilt library and codes. It is also easy for performing OpenCV bindings.
Netbeans: it is the ide which makes the importing and performance easy and quick.

This algorithm reduces the problem of illumination in eigenfaces. This is the improved version of last one .In eigenfaces it is difficult to discriminate one individual from another because Eigenfaces find

IV. METHODOLOGY

In order to overcome the drawbacks of the previous algorithm a more efficient algorithm is used in this project. LBPH or Local Binary

Patterns histograms. The technologies available in the Open-Computer-Vision (OpenCV) library and methodology to implement them using Java are used.
For face detection, Haar-Cascades were used and for face recognition Local binary pattern histograms (LBPH) were used.
The methodology is represented together with flow charts for every stage of the system. Next, the results are shown using screen

Algorithm.
Local Binary Pattern Histogram
Local binary patterns were proposed as classifiers in computer vision and in 1990 By Li Wang [5]

For feature encoding, the image is divided into cells (4 x 4 pixels). Using either a clockwise or counterclockwise direction the values of surrounding pixels are computed and compared to the value of the centre pixel. The value of intensity of each number is compared to that of the pixel depending on the difference calculated whether higher or lower value 0 or 1 is assigned to that location the result is an 8 bit value to the cell. The advantage of this technique is that even if the luminosity of the image is different at different locations the result will be same as before[6] .

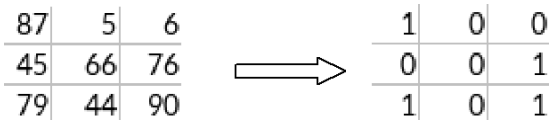


Figure 2:local binary pattern histogram generating 8 bit number

Histograms are used to find the frequency of the values process faster .By analyzing the cell result , edges can be detected as and when the value changes .Feature vectors are obtained by computing the value of all the cells and concatenating them with histograms . Images are classified with the help of the ID attached. The input images are classified on the same process and compared with the data set and the histogram obtained. Threshold level is set and with that threshold it can be easily identified if the faces are known or unknown[7][8][9]

1.**Parameters**: the LBPH uses 4 parameters:

- **Radius**: radius is used to build a local circular binary pattern and represent the radius around the centre pixel which is usually set to 1 neighbor
- **Neighbors**: the number of points required to build the circular local binary pattern is known as neighbors the more is the simple points included more is the computational cost this is usually set to 8

- **Grid X**: Numbers of cells in the horizontal direction of the neighbor the more number of cells are present the finer is the Grid. It is usually set to 8.

- **Grid Y**: the number of cells in the vertical direction. The grid will be finer if more number of vertical directed cells is present; this value is usually set to 8.

2. Training the Algorithm: to train the algorithm we first need to create a data set of the facial images of the people we want to recognize .We need to also set an ID for each person so that the algorithm will use this information to recognize the image that is send as an Input and give an output. Image of the person must have the same Id .After this LBPH computational steps are proceeded.

3. Applying the LBP operation: The first step in the computational steps is the creation of an intermediate image. This image describes the original image in a better way with the features and the facial characteristics highlighted. Here the concept of sliding window based on the parameters like **radius** and **neighbors** are used

Here several small steps are involved :

- We have a facial image in grayscale.
- This image is converted into a window of 3 X 3 pixel containing the intensity of each pixel (0~255).
- Then the centre value is treated as threshold. It is used to find the values of 8 neighbors. This value will be used to define the new values from the 8 neighbors.
- For each neighbor we find a value based on the threshold value if the value is higher than threshold then we assign one if it is lower than that value zero is assigned.
- Now, The Matrix now contains only binary value.
- Then, this binary value is converted to decimal value and set it to the central value of the matrix.
- At the end of this procedure there is new image which represent the original image in a better way by highlighting the characteristics.

Note: LBP procedure was expanded to use a different number of radius and neighbors, it is called Circular LBP.

4. Extracting the Histograms: Now the histogram is extracted from the image generated in the last year we use grid X and Y parameters to divide the image into multiple grid.
We can extract the histogram of each region as follows:

- As we have an image in grayscale each histogram will contain only 256 positions that is from 0 to 255
- Then, then the creation of final histogram is started where we can donate 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.

5. Performing the face recognition: In this step, algorithm is trained already. Histogram is created which are used to represent the image in the data set. For each new input a new histogram is created that 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 different approaches can be used to calculate the difference between the histograms.

- So this algorithm returns the ID of the image that has the closest histogram. The algorithm also returns the calculated distance which is termed as '**confidence**' measurement
- In order to estimate if the algorithm has correctly recognise the image the threshold and the confidence are used. If the confidence value is lower than that of the threshold defined then we can assume algorithm has been successfully completed

V. FLOW DAIGRAM

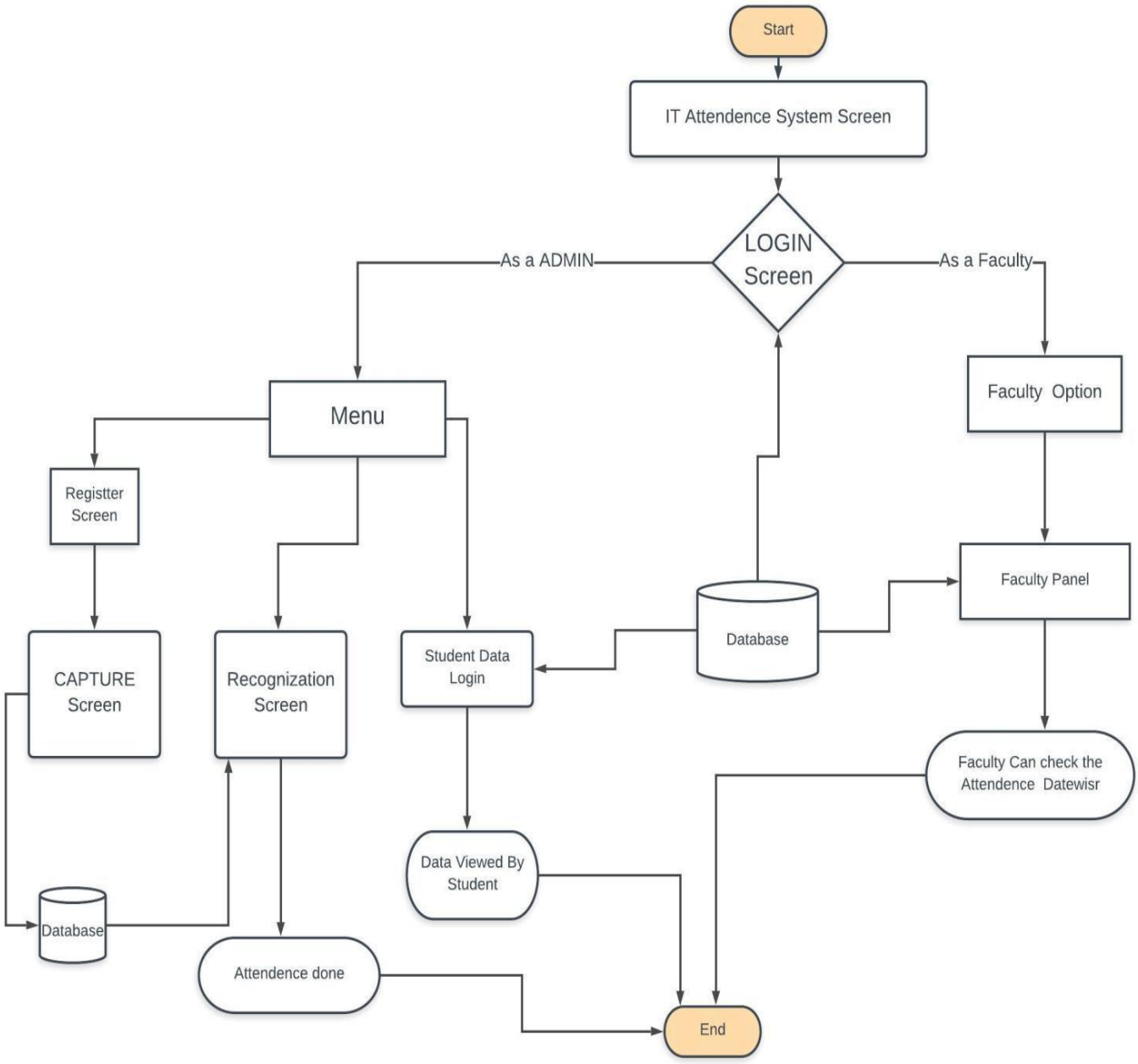


Figure 3: flowchart of the application

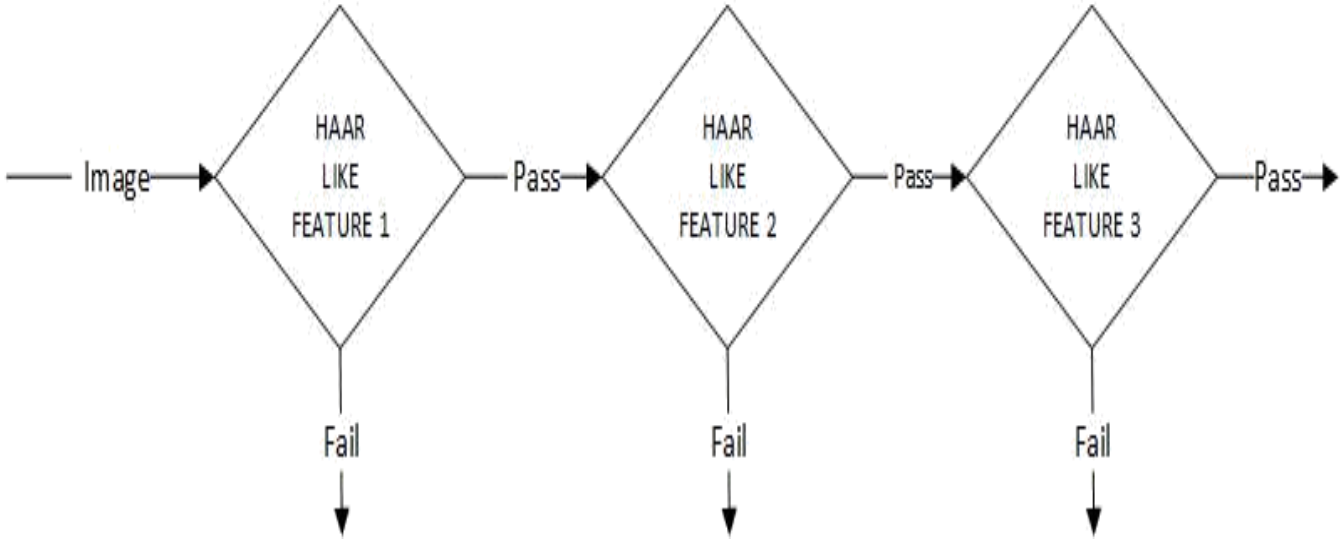


Figure 4: Flow chart of haar feature detection using voila jones

VI. RESULTS

FIRST FRAME

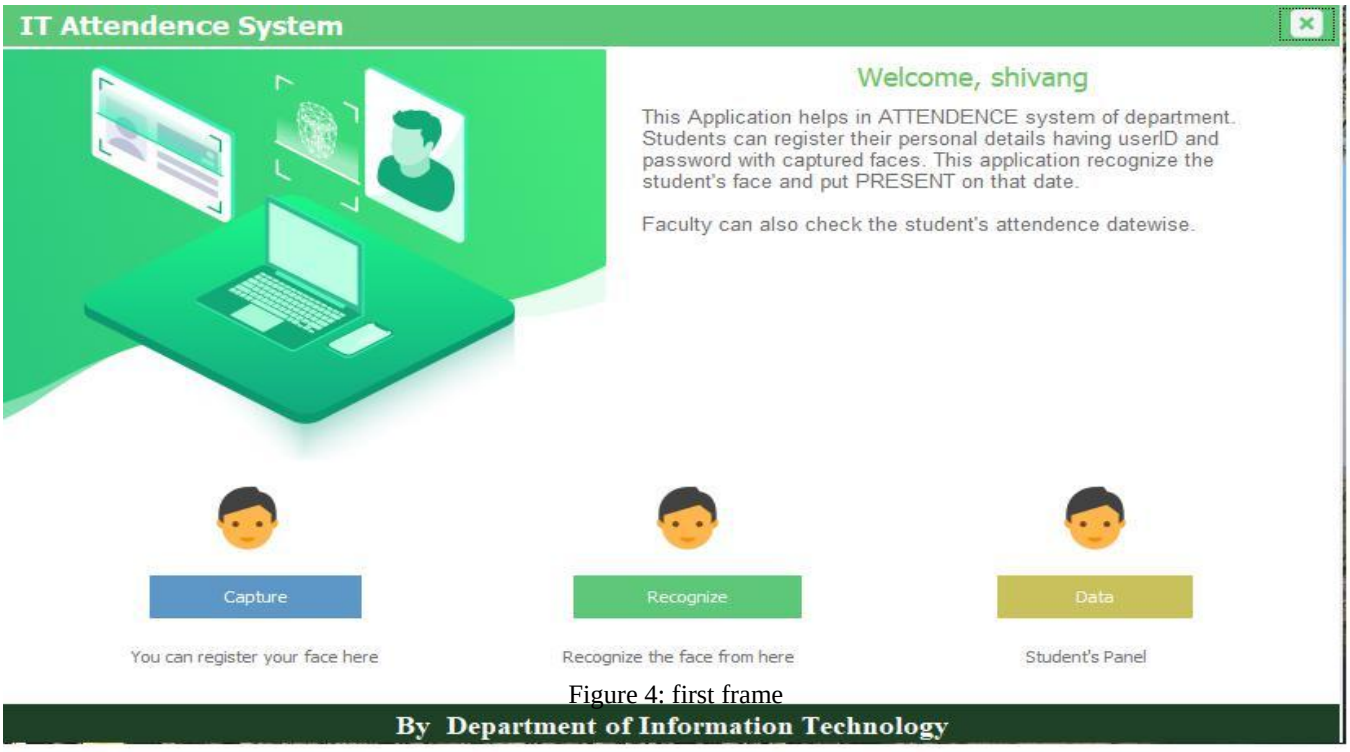
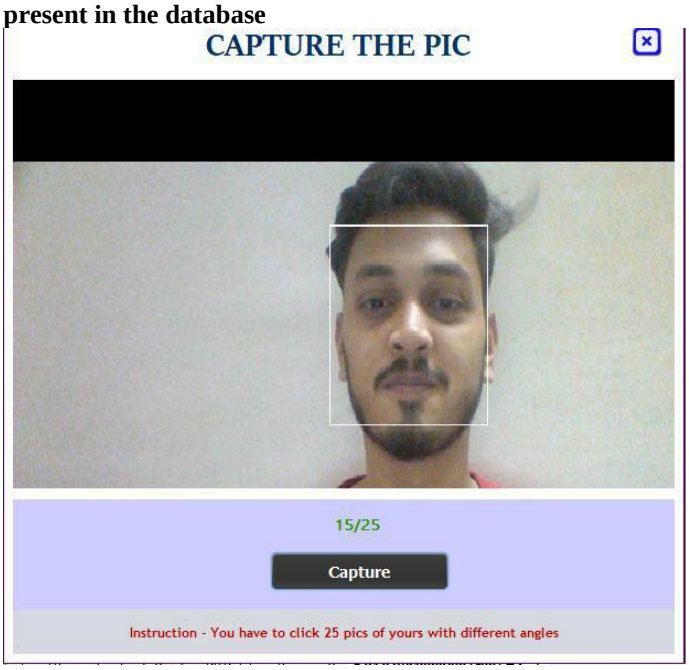
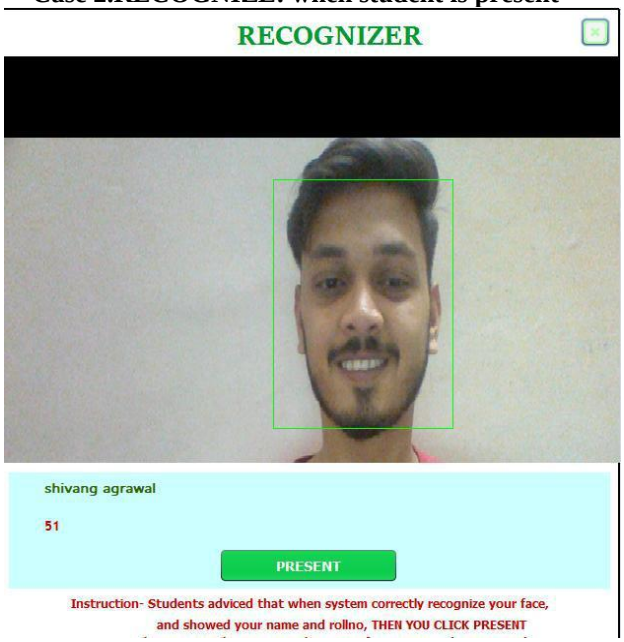


Figure 4: first frame

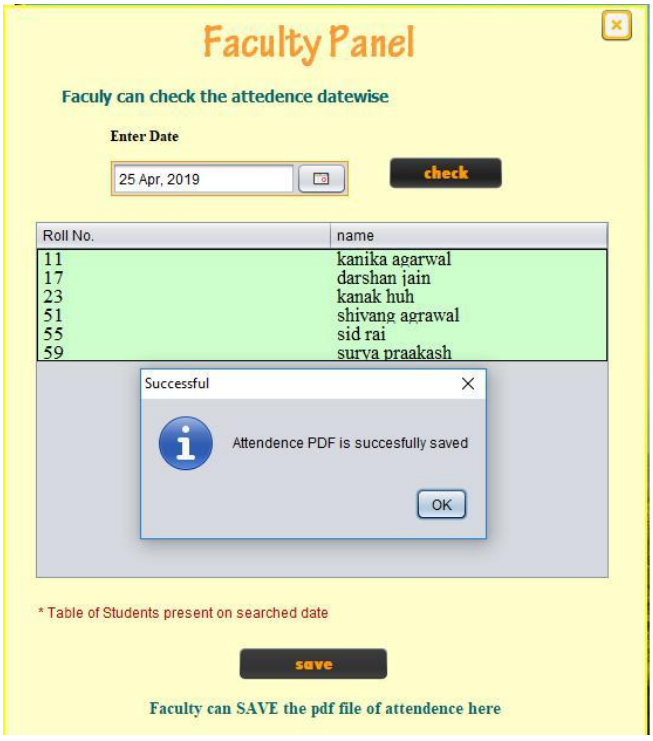
Case 1.CAPTURE: when the student is not present in the database



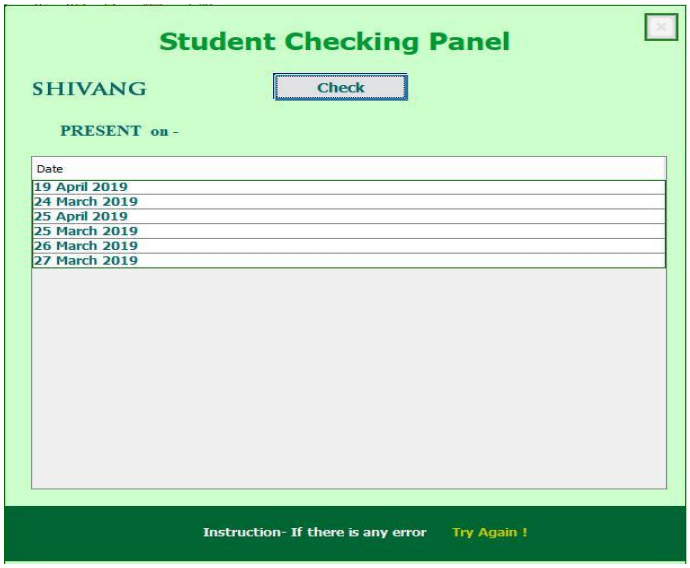
Case 2.RECOGNIZE: when student is present



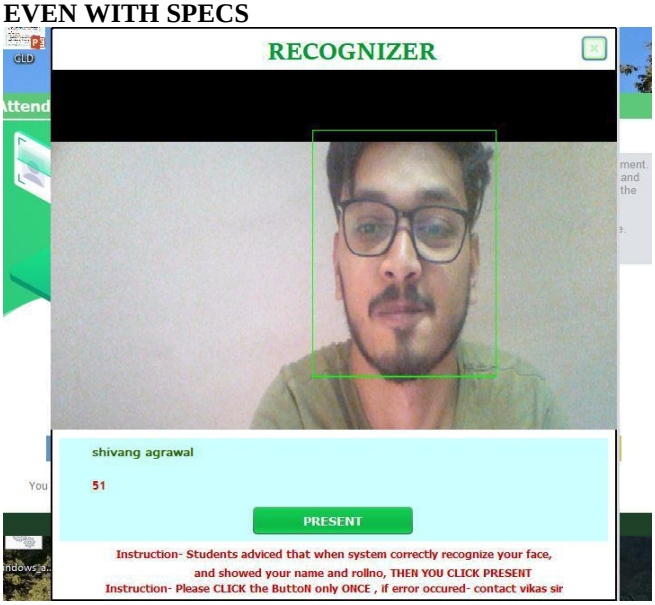
CASE:TEACHER CAN VIEW THE ATTENDENCE AND SAVE IT



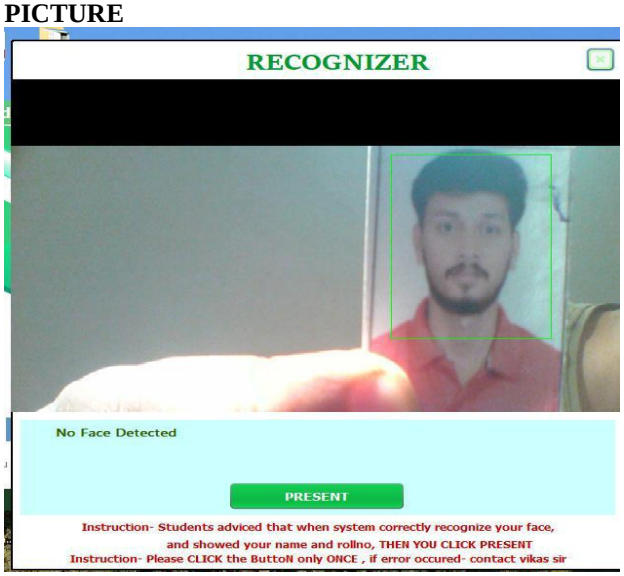
CASE:STUDENT CAN SAVE THE ATTENDENCE ON A PARTICULAR DATE



CASE: RECOGNIZITION SUCCESSFUL EVEN WITH SPECS



CASE: RECOGNIZATION FAILS IF TRIED WITH PICTURE



IMPORTANT CODE:

1. Image acquisition

```
webSource = new opencv_videoio.VideoCapture(0);
       myThread = new Capture.DaemonThread();
       Thread t = new Thread(myThread);
```

2. Conversion to grayscale

```
opencv_core.Mat imageGray = new opencv_core.Mat();
 cvtColor(imageColor, imageGray, COLOR_BGRA2GRAY);
```

3. Formation of rectangle around face

```
opencv_core.RectVector detectedFaces = new opencv_core.RectVector();
                        cascade.detectMultiScale(imageColor, detectedFaces, 1.1, 1, 0, new
opencv_core.Size(150, 150), new opencv_core.Size(150, 150));
```

4. Creation of classifier

```
 opencv_objdetect.CascadeClassifier cascade = new opencv_objdetect.CascadeClassifier("C:\\photos
\\haarcascade_frontalface_alt.xml");
    opencv_face.FaceRecognizer recognizer = opencv_face.LBPHFaceRecognizer.create();
```


5. Database connection

```
private final String driver = "org.apache.derby.jdbc.ClientDriver";
    private final String root = "jdbc:derby://localhost:1527/face";
    private final String user = "1234";
    private final String pass = "1234";

    public void connects() {
        try {
            System.setProperty("jdbc.Driver", driver);
            conn = DriverManager.getConnection(root, user, pass);
            System.out.println("OK!");
        } catch (SQLException e) {
            System.out.println("Error: " + e);
        }
```

6. Capturing of 25 images and creating file

```
   if (saveButton.getModel().isPressed()) {
  if (sample <= numSamples) {
   String cropped = "C:\\photos\\person." + ic
  imwrite(cropped, face);
  counterLabel.setText(String.valueOf(sample)
  sample++;
   }
   if (sample > 25) {
   insertDatabase();
   System.out.println("File Generated");
  stopCamera();
  }}
```

VII. CONCLUSION

Thus in the end we have successfully implement the attendance system using face recognition and achieved the following objectives:

1. The dataset of students is constructed using capture window and the classifier will be trained based on these images. These images are converted to grayscale and then fed to classifier.

2. Student is recognized by comparing their images to the dataset. The id is returned with accuracy within 30%.

3. The system then match and mark present of detected student's face cropped from an image from webcam to those present in database on the system.

4. Students and teacher have the facility to view and save the pdf of the attendance on particular dates.

Henceforth we have developed a system which can capture, store and recognize the student face and mark the attendance digitally .We can further develop our work towards the direction of a better recognition using better accuracy.

ACKNOWLEDGMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Dr. Anjana Pandey ma'am for providing me an opportunity to do the project work in face recognition and giving us all support and guidance which made me complete the project duly. I am extremely thankful to her for providing such a nice support and guidance, although she had busy schedule managing the corporate affairs.

REFERENCES

[1] Ayushi gupta, Ekta Sharma, Neha sachan and neha Tiwari, "Door Lock System through Face Recognition Using MATLAB", IJSRCSE (International Journal of Scientific Research in Computer Science andEngineering), 2013
[2] Face Recognition Based on HOG and Fast PCA Algorithm Xiang-Yu Li(&) and Zhen-Xian Li
[3] Dinalankara, Lahiru. (2017). Face Detection & Face Recognition Using Open Computer Vision Classifies.
[4] Study of Various Face Detection Methods , Ms.Varsha Gupta1 , Mr. Dipesh Sharma2,ijarcce vol.3
[5] Prabhjot Singh1 and Anjana Sharma "Face Recognition Using Principal Component Analysis in MATLAB" ", IJSRCSE (International Journal of Scientific Research in Computer Science andEngineering), vol 3,issue 1 2014
[6]. Kelvin Salton do Prado "Face Recognition: Understanding LBPH Algorithm" towardsdatascience.com
[7] Nicolas Delbiaggio "A comparison of facial recognition's algorithms " Hagga-Helia university of applied science
[8] Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen. "Face recognition with local binary patterns." Computer vision-eccv 2004 (2004): 469–481.
[9] Matti Pietikäinen (2010) Local Binary Patterns. Scholarpedia, 5(3):9775

ABOUT AUTHORS

Dr. Anjana Pandey, received her MCA degree from GGDU university bilaspur, M.Tech. in Computer Science from Banansthali Vidyapith Rajasthan. Ph.D. in Computer Application in 2011 from MANIT , Bhopal, Madhya Pradesh, India. Currently she is working as an Assistant Professor, Department of Information Technology, University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (Madhya Pradesh), India. She has published more than 30 Research Papers in various International & National Journals & Conferences, including SCIE Journals & Scopus Journals. she is guiding 06 Students in PhD program. International Journals and conferences

Mr .Shivang Agrawal is pursuing bachelor of engineering in information technology from University Institute of Technology, RGPV , Bhopal.

Ms .Kanika Gupta is pursuing bachelor of engineering in information technology from University Institute of Technology, RGPV ,

