LBPH FACE RECOGNITION BASED ATTENDANCE MARKING MODEL

A Project Report of Capstone Project - 2

Submitted by

AJAY TIWARI

16SCSE101762

in partial fulfilment for the award of the degree

of

Bachelor of Technology

IN

Computer Science Engineering

School of Computer Science and Engineering

Under the Supervision of

Mr. Soumalya Ghosh

Assistant Professor

May 2020

GALGOTIAS UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **"VISUAL PROCESSING"** is the bonafide work of **"AJAY TIWARI (1613101066)"** who carried out the project work under my supervision.

SIGNATURE OF HEAD

Dr. MUNISH SHABARWAL,

PhD (Management), PhD (CS)

Professor & Dean

School of Computer Science & Engineering

SIGNATURE OF SUPERVISOR

Mr. SOUMALYA GHOSH,

Assistant Professor

School of Computer Science & Engineering

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LBPH FACE RECOGNITION BASED ATTENDANCE MARKING MODEL

1. Abstract

Attendance marking has constantly been a tiresome task for every organisation in past years. But with the evolution of technology there has been huge enhancement in this field as well. There have been development of thumb impression and iris scanner based attendance systems but they have their own drawbacks. The recent breakthrough in the attendance marking technique is the face recognition based attendance marking model (FRBAMM). Earlier, teachers wasted their precious time in cross verification of the attendance. They had to check if someone's attendance has been missed. But this system will resolve all their manual complexities [8]. This model turns out to be highly accurate and amazingly fast in its processing. It is a great advancement as this system automatically recognises the face of student and updates the attendance in real time scenarios too.

2. Introduction

Attendance is the most fundamental and integral part of an education institution. Daily records have to be maintained for each student in order to measure their reports and tasks. In [2], the authors have emphasised that the FRBAMM can be really handy for this purpose and it works in real time and marks attendance for every student present in the class by recognising each face simultaneously. This technique can easily overcome the problem of proxy attendance in the classes. As explained by Hubel and Wiesel [1], recognising different faces can be very easy for human beings because of the nerve cells in our body but it can be too hard to handle for the machines. These machines are trained on huge data sets for extracting the basic and significant features. Some of those features include eyebrows, eyes, ears, nose, lips, etc from the pictures. The Attendance System Using Face Recognition and Class Monitoring System [3], describes the process of uploading the dataset for every student by differentiating their facial properties and reduces the manual work. The FCBAMM works in two phases. In the initial phase it spots all the faces from the provided input. Finally, it analogizes the faces with the registered faces for identification. The attendance is updated for the concerned students.

Automatic Attendance System Using Face Recognition [4] delineates the use of various algorithms such as Viola-Jones and PCA for the attempt of face recognition. In the described model, images were captured at the beginning and ending of the lectures and if the student is recognised in both the pictures then their attendance would be updated. Class Room Attendance System Using Facial Recognition System [5], displays us a new 3D model approach to recognise and identify any person in the classroom. These analytical researches would help us in recognising the students from a live feed recording or any captured image and mark their attendance.

3. Related Work

All over the globe, billions of people have to mark their attendance in innumerable institutions and almost every institution has the technology of their own to do the same task. Earlier it was generally done manually but with the passage of time technology advanced and we ascertained some new methods to perform this task with ease. Some of the models developed were RFID, fingerprint scanner, iris scanner, voice pattern, etc.

RFID based attendance system [6] explains a way to mark the attendance of the students. This RFID based model uses the radio frequency waves to complete the task. The students need to place their respective ID cards on the machine which then records the attendance for every student by reading the different frequencies from the student's ID card. This could turn out to be very tedious task to map numerous students.

The next technique used to mark the attendance is the iris scanning system as discussed in A design and implementation of a wireless iris recognition attendance marking technique [7]. In this type of system every student needs to stand in front of the iris scanner which deciphers the information present in the unique patterns from their retina blood vessels. This technique could be very expensive to handle for some organizations. Moreover, it is very complex way to perform such an easy task. This technique too doesn't allow you to automatically mark the attendance of anybody in real time without them doing any task.

The fingerprint scanning technique is still very widely used all over the world in various organisations. But it requires the person to manually punch the machine in order to mark their attendance. It doesn't allow you to automatically detect the person and mark their attendance. This leads to wastage of time of all the beings whether they are students or teachers of any college or the employees of any organisation. This technique can also be considered unhygienic as everyone present in that area have to touch the same machine in order to mark their attendance and it could pass on harmful germs and diseases from one person to another.

As we briefly discussed about some of the current attendance marking system being used in the society. We come to a final conclusion that none of these methods could automatically mark the attendance of any person. In every system there is one or another task that the user has to do in order to have their attendance marked. But this technique of FRBAMM could turn all the tables around as there is totally no need of the user to do anything at all. They just have to come to their daily schedule and utilize all their time for a better cause instead of standing against any machine to mark their attendance.

4. Motivation

The main motive behind developing this type of system for marking the attendance of the students was to save the time and energy of the teachers as well as students. In every lecture, a lot of time is wasted in the process to mark the attendance and the manual recoding could be very tedious task. If this system automatically marks and updates the attendance of the students attending the class in real time then every single bit of time could be consumed for some positive outcome. With the help of this system we could easily eliminate any probability of proxy attendance of any absent student in the class. The whole process of attendance marking could be efficient and true without any heavy manual task to be done by any person.

5. Proposed Model

The current system proposed in FRBAMM operates in the following steps:

5.1 Image Capturing

Every student needs to register their face in the database once in order to verify and mark their attendance during the lectures. Students stand in front of the camera while several snapshots for each individual are recorded. The cropped images of the face are stored in different files respectively.

5.2 Face Detection

Once all the students have recorded their data in the system then we can have real time attendance updating in the spreadsheet. The system captures the faces from the live stream video of the classes from the security cameras and then maps each individual face is cross checked with the registered snapshots of the students. Finally the attendance of the faces that match with the database are marked as present in the spreadsheet file.

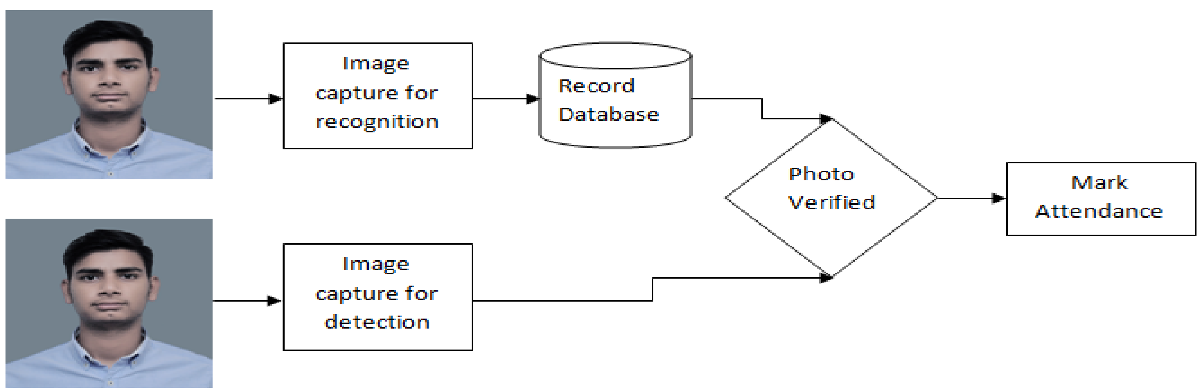

Figure 1: Process for FRBAMM

All the work done on the FRBAMM only involved local access to the data. Using the new model proposed here, with the use of cloud technology we could have remote access to all the data required and there wouldn't be any need of any sort of high end processing system at the physical location. We could store all of our dataset information on a cloud server, where we can have real processing to be done. The live camera feed could be verified directly from our cloud storage and the attendance sheet could be updated simultaneously at the same location as well. With this technique we could mark attendance of any physical location without being present there and can also see their reports from remote location as we see fit.

6. Experimental Setup

This experiment proved to be very interesting tasks as their needed several volunteers to carry out this process successfully and measure its effectiveness. To keep the process a balanced and fair one the volunteers were taken from different backgrounds. Some of them were from the total technical background whereas some other was total novice. A total of six volunteers were taken for carrying out the experiment effectively. The following tasks were carried out in the process:

- Capture image of the person for registration
- Pre-processing of data captured
- Training the model on the created dataset
- Real time capturing of faces from the camera to detect presence.
- Face detection from the camera
- Mark attendance (if present)

Initially, each individual needs to stand against the camera where the system detects the face in the frame. The required part of the picture is cropped and saved as the dataset. Each individual had at least sixty frames for recognition to help the system in better understanding. The face registration process had a variety of data in order to provide an efficient model to detect faces. The model was tested in various different scenarios where the physical appearance of the subjects was regularly changed but it didn't have any major affect on the accuracy of this model. This different facial expressions posed by the subjects were easily tackled by the system. Some issues were faced due to the angle of presentation of the face towards the camera. But repeated training and testing could enhance its accuracy. The real time effect helps this system to be highly efficient as the attendance could be updated automatically on the very next go when the face is matched if it is missed on the first attempt.

LBPH Face Recognition Algorithm

Local Binary Patterns Histogram (LBPH) face recognition algorithm is based upon the local binary operator that is an excellent text descriptor. This algorithm works by labelling the pixels of the image by thresholding the neighbours of each pixel and then converting the resultant into binary format. It is the blend of Local Binary Pattern (LBP) and Histograms of Oriented Gradients (HOG) which makes it highly efficient as it represents the images using a plain data vector.

- Parameters: LBPH has 4 parameters as follows:

  (a) **Radius** is used to build the circular local binary pattern and commonly has value 1.

  (b) **Neighbours** depict the number of sample points that is used to build the circular local binary pattern. More the number of points, higher is the computational cost. It is ordinarily set to 8.

  (c) **Grid X** refers to the number of cells in horizontal direction. More the number of cells, finer are the grid and higher will be the dimensionality of the resultant feature vector. The value is generally set to 8.

  (d) **Grid Y** represents the number of cells in vertical direction. More the number of cells finer is the grid and higher is the dimensionality of the resultant feature vector. It usually has the value of 8.

- Steps involved in this algorithm are as follows:

(a) Dataset Creation

(b) Face Acquisition

(c) Feature Extraction

(d) Classification

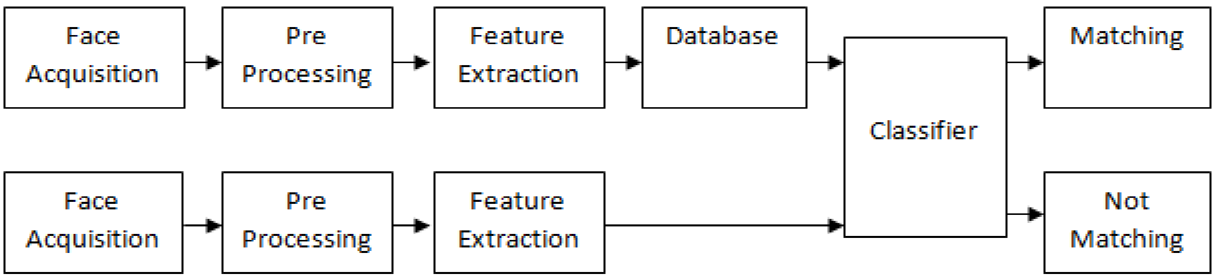

Figure 2: Working of LBPH algorithm

Suppose we have the image of dimensions M*N. We divide the image into small regions of same width and height (let it be m*m).

Figure 3: Region creation for LBP

Then we use the local binary operator for all the regions of the image. This operator is defines in the window of size 3*3.

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

(X_c, Y_c) = Central Pixel with intensity of I_c and I_p will be the intensity of the neighbouring pixel.

We use the value of median pixel as the threshold; it then compares the pixel with its closest 8 pixels n the function. The central value is set to 1 if the neighbour's value is greater than or equal to 1, else the central value is set as 0. Finally, all the 8 binary values of the neighbours is obtained. We then get a 8-bit binary number by combining these values which is then translated to the decimal number for the sake of simplicity and convenience. The resulting decimal number is referred as the Pixel LBP value which ranges from 0-255.

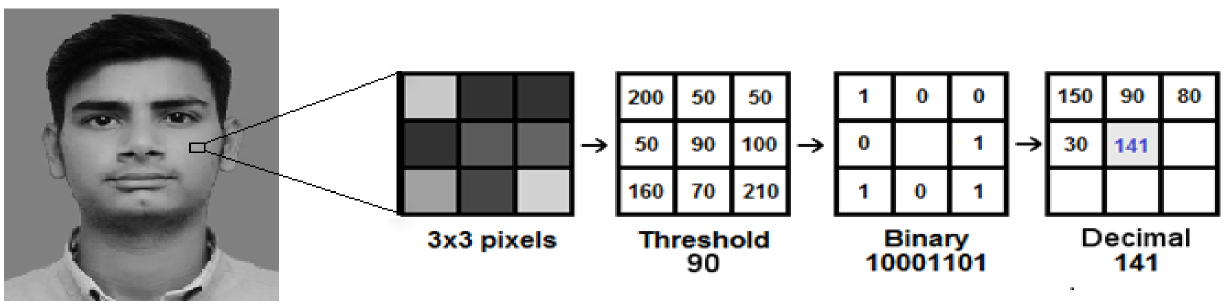

Figure 4: Converting image into Pixel LBP value

But as the fixed neighbourhood fails to encode the details varying in scale therefore this algorithm uses different number of neighbours and radius for better outcome and hence called circular LBP.

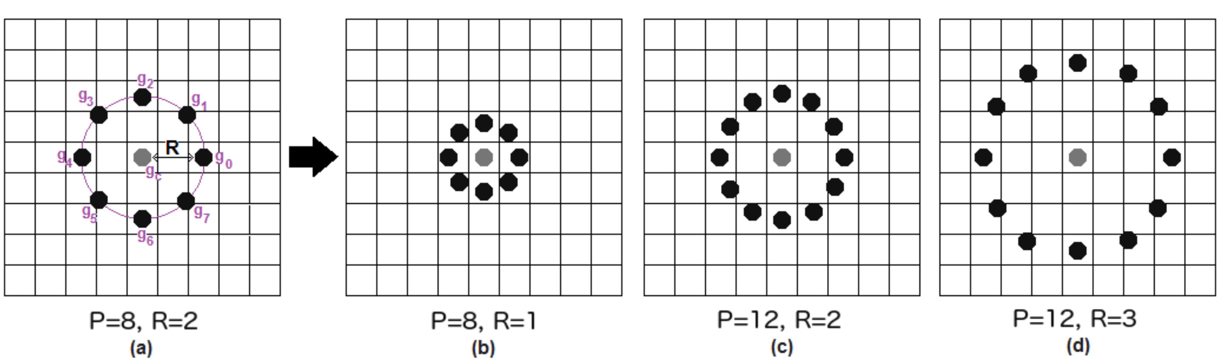

Figure 5: Representation of neighbouring points for varying radius

For any point given as (X_c,Y_c), position of the neighbouring point (X_p,Y_p), where p belongs to P is calculated using the formula:

$$x_p = x_c + R\cos\left(\frac{2\pi p}{P}\right)$$
$$y_p = y_c - R\sin\left(\frac{2\pi p}{P}\right)$$

Here, R refers to the circle's radius and P represents the number of the sample points.

If any coordinate of the point on the circle does not correspond to the image coordinates, it is then interpolated familiarly using the bilinear interpolation as follows:

$$f(x,y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

The LBP operator is well-conditioned against monotonic gray scale transformations.

After the generation of LBP value, the histogram for the region is created with the help of counting the number of matching LBP values in that given region. Similarly, when we have finished with the creation of histograms for all the regions, they are merged together to form a single histogram which is called the feature vector of the image.

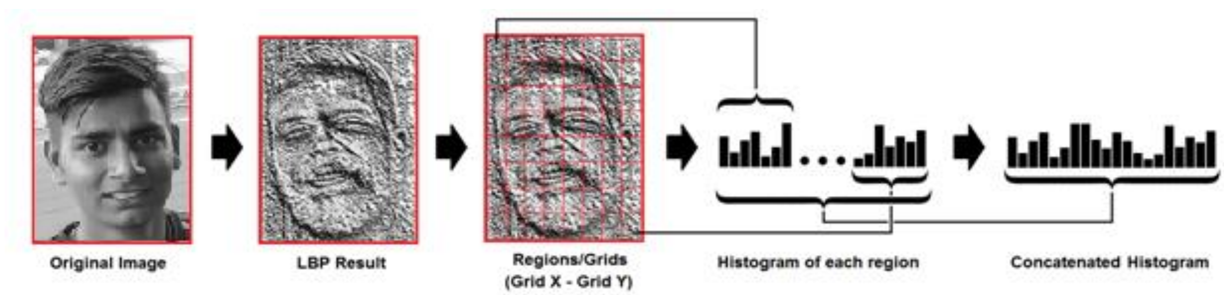

Figure 6: Extraction of histograms from image

Finally, to get our result, the histograms of the test image and the images present in the database are compared and the closest matching histogram image is returned. This could be done using various methods such as Euclidean distance, absolute value, chi-Square method, etc. The formula for carrying out such comparisons using Euclidean distance is as follows:

$$d(a,b) = \sqrt{\sum_{i=1}^{n} |a_i - b_i|^2}$$

Therefore, we get the Id and name on the recognised image from the dataset if the test image is recognised.

7. Result

The attendance for numerous students could be marked successfully due to this technique. Several attempts were carried out on order to check the efficiency of the system. The outcome of this model was highly satisfactory. The system easily recognised and marked the attendance of registered students present in the class. This system could also identify and mark attendance of the students even if there is some change in their physical appearance with time to time.

In the first attempts we saw the system working perfectly fine. But as we moved on with the attempts, we saw some discrepancy in the results. These problems were due to the face angle as recorded in the camera of some of the novice subjects. In some of the attempts the camera was unable to detect the faces properly due to the improper lighting in the room. In some cases we also found few minor errors when the system falsely identified any specific individual who were absent from the class. There were also a few attempts when the system failed to recognise the student in normal circumstances. But finally we could see the effectiveness of the system in real time scenario.

Table I: Result for attendance marking for students (TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative)

STUDENTS	ATTEMPTS							
	A	B	C	D	E	F	G	H
AJAY	TP	TP	TP	TP	TP	TP	TP	TP
OM	TP	TP	TP	TP	TP	TP	TP	TP
KUMAR	TP	TP	TP	FN	TN	TP	TN	TP
SHUBHAM	TP	TP	TP	FP	TN	TP	TN	TP
ABHISHEK	TP	TP	TP	TN	TP	TP	TP	TP
VARUN	TP	TP	TP	TP	TP	TN	TP	TP

Terms:

True Positive (TP): case that the model predicts present and is actually present.

True Negative (TN): case that the model predicts absent and is actually absent.

False Positive (FN): case that the model predicts present but is actually absent.

False Negative (FN): case that the model predicts absent but is actually present.

Calculated Values:

Sensitivity= (TP/ (TP+FN))*100= (40/ (40+1)*100= (40/41)*100= 97.56

Specificity= (TN/ (TN+FP))*100= (6/ (6+1))*100= (6/7)*100= 85.71

Accuracy= ((TP+TN)/ (TP+TN+FP+FN))*100= ((40+6)/ (40+6+1+1))*100= (46/48)*100= 95.83

Efficiency= (Sensitivity+ Specificity+ Accuracy)/2= (97.56+85.71+95.83)/3= 279.1/3= 93.03

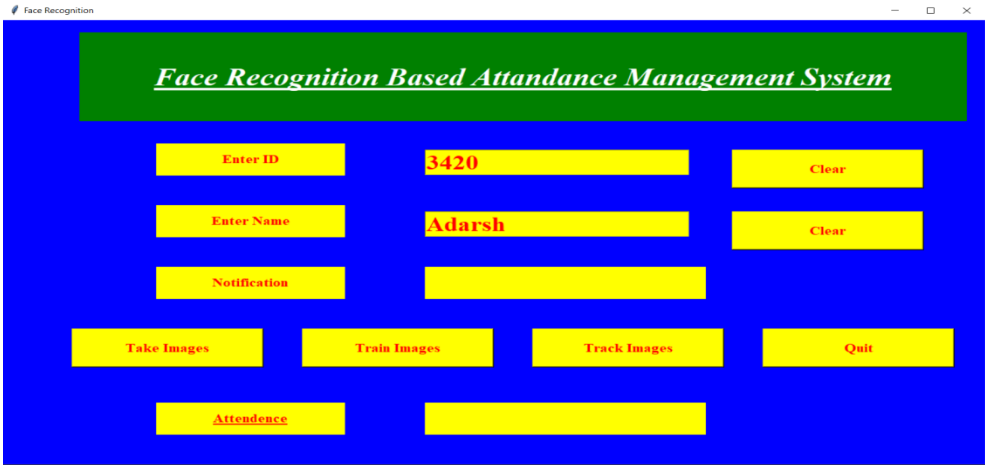

Figure 8 (a): Program Interface

Figure 8 (b): Image tracking process

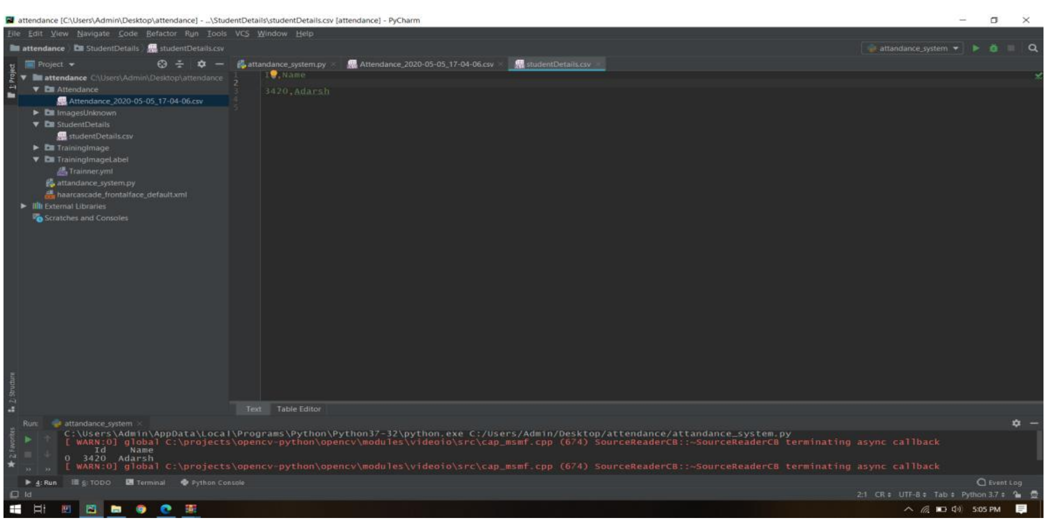

Figure 8 (c): Attendance Marking

8. Conclusion

This technique proves to be better than other attendance marking techniques such as RFID, Iris scanner, etc. as it is highly efficient and time saving process. However, we could come across some minor issues while recording the attendance which could occur due to the facial angle of the students while capturing photo or the reflection of light in the area. The main advancement that needs to be done in the improvement of the algorithm in order to detect faces in uneven lighting situations and increase the angle at which the face has the be reflected so that it could be detct to mark thr attendance. There could also be an addition where the system provides the information for the unknown person that hasn't been recognised by the system.

9. References

1. OpenCV2.4.13.5documentationhttps://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html
2. Shireesha Chintalapati, M.V. Raghunadh, "Automated Attendance Management System Based on Face Recognition Algorithms", 2013 IEEE International Conference on Computational Intelligence and Computing Research.
3. Arun Katara, Mr. Sudesh, V.Kolhe, Attendance System using Face Recognition and Class Monitoring System, IJRITCC, Volume: 5,Issues: 2 ,February 2017.
4. Ashish Choudhary, Abhishek Tripathi, Abhishek Bajaj, Mudit Rathi, and B.M Nandini, "Auttomatic Attendance System Using Face Recognition", International Journal of Modern Trends in Engineering and Research, Volume: 03, Issue: 04, ISSN (Online):23499745; ISSN (Print):2393-8161, April 2016.

5. Abhishek Jha, "Class room attendance system using facial recognition system", The International Journal of Mathematics, Science, Technology and Management (ISSN: 2319-8125) Volume: 2, Issue: 3,2014

6. T. Lim, S. Sim, and M. Mansor, "RFID based attendance system", Industrial Electronics Applications, 2009. ISIEA 2009. IEEE Symposium on, volume: 2, pages 778-782, IEEE 2009.

7. S. Kadry and K. Smaili, "A design and implementation of a wireless iris recognition attendance management system", Information Technology and control, volume: 36, no: 3, pages 323329, 2007.

8. Kalachugari Rohini, Sivaskandha Sanagala, Ravella Venkata Rathnam, Ch.Rajakishore Babu, "Face Recognition Based Attendance System For CMR College of Engineering and Technology", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-4S2 March, 2019

Code:

```
import tkinter as tk

from tkinter import Message,Text

import cv2,os

import shutil

import csv

import numpy as np

from PIL import Image,ImageTk

import pandas as pd

import datetime

import time

import tkinter.ttk as ttk

import tkinter.font as font



window=tk.Tk()
```

```
window.title("Face Recognition")

window.geometry('1280x720')

dialog_title='Quit'

dialog_text='Are you sure?'

window.configure(background='blue')

window.grid_rowconfigure(0,weight=1)

window.grid_columnconfigure(0,weight=1)



message=tk.Label(window, text="Face Recognition Based Attandance Management System",bg="Green",fg="White",width=50,height=3,font=('times',30,'italic bold underline'))

message.place(x=100,y=20)

lbl=tk.Label(window,text="Enter ID", width=20,height=2,fg="red",bg="yellow",font=('times',15,'bold'))

lbl.place(x=200,y=200)

txt=tk.Entry(window,width=20,bg="yellow",fg="red",font=('times',25,'bold'))

txt.place(x=550,y=210)
```

```
lbl2=tk.Label(window,text="Enter
Name",width=20,fg="red",bg="yellow",height=2,font=('times',15,'bold'))

lbl2.place(x=200,y=300)

txt2=tk.Entry(window,width=20,bg="yellow",fg="red",font=('times',25,'bold'))

txt2.place(x=550,y=310)



lbl3=tk.Label(window,text="Notification",width=20,fg="red",bg="yellow",height=2,font=('ti
mes',15,'bold'))

lbl3.place(x=200,y=400)



message=tk.Label(window,text="",bg="yellow",fg="red",width=30,height=2,activebackgroun
d="yellow",font=('times',15,'bold'))

message.place(x=550,y=400)



lbl4=tk.Label(window,text="Attendence",width=20,fg="red",bg="yellow",height=2,font=('ti
mes',15,'bold underline'))
```

```
lbl4.place(x=200,y=620)



message2=tk.Label(window,text="",bg="yellow",fg="red",width=30,height=2,activeforegrou
nd="green",font=('times',15,'bold'))

message2.place(x=550,y=620)



def clear():

    txt.delete(0,'end')

    res=""

    message.configure(text=res)



def clear2():

    txt2.delete(0,'end')

    res=""

    message.configure(text=res)
```

```
def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        pass
    try:
        import unicodedata
        unicodedata.numeric(s)
        return True
    except (TypeError,ValueError):
        pass
    return False


def TakeImages():
```

```
    Id=(txt.get())

    name=(txt2.get())

    if(is_number(Id) and name.isalpha()):

        cam=cv2.VideoCapture(0)

        harcascadePath="haarcascade_frontalface_default.xml"

        detector=cv2.CascadeClassifier(harcascadePath)

        sampleNum=0

        while(True):

            ret,img=cam.read()

            gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

            faces=detector.detectMultiScale(gray,1.3,5)

            for(x,y,w,h) in faces:

                cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)

                sampleNum=sampleNum+1;

                cv2.imwrite("TrainingImage\ "+name +"."+ Id +'.'+str(sampleNum)+".jpg",
gray[y:y+h,x:x+w])
```

```
            cv2.imshow('frame',img)

        if cv2.waitKey(100) & 0xFF==ord('q'):

            break

        elif sampleNum>60:

            break

    cam.release()

    cv2.destroyAllWindows()

    res="Images Saved for ID:"+ Id +" Name: "+name

    row=[Id,name]

    with open('StudentDetails\studentDetails.csv','a+') as csvFile:

        writer=csv.writer(csvFile)

        writer.writerow(row)

    csvFile.close()

    message.configure(text=res)

else:
```

```
        if(is_number(Id)):

            res="Enter Alphabetical Name"

            message.configure(text=res)

        if(name.isalpha()):

            res="Enter numeric Id"

            message.configure(text=res)




def getImagesAndLabels(path):

    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]

    faces=[]

    Ids=[]


    for imagePath in imagePaths:

        pilImage=Image.open(imagePath).convert('L')
```

```
        imageNp=np.array(pilImage,'uint8')

        Id=int(os.path.split(imagePath)[-1].split(".")[1])

        faces.append(imageNp)

        Ids.append(Id)

    return faces,Ids



def TrainImages():

    recognizer = cv2.face.LBPHFaceRecognizer_create()

    harcascadePath="haarcascade_frontalface_default.xml"

    detector=cv2.CascadeClassifier(harcascadePath)

    faces,Id=getImagesAndLabels("TrainingImage")

    recognizer.train(faces,np.array(Id))

    recognizer.save("TrainingImageLabel\Trainner.yml")

    res="Image Trained"#+",".join(str(f)) for f in Id

    message.configure(text=res)
```

```
def TrackImages():

    recognizer = cv2.face.LBPHFaceRecognizer_create()
#cv2.face_LBPHFaceRecognizer.create()

    recognizer.read("TrainingImageLabel\Trainner.yml")

    harcascadePath="haarcascade_frontalface_default.xml"

    #detector=cv2.CascadeClassifier(harcascadePath)

    faceCascade=cv2.CascadeClassifier(harcascadePath)

    df=pd.read_csv("StudentDetails\studentDetails.csv")

    #cam=cv2.FONT_HERSHEY_SIMPLEX

    cam = cv2.VideoCapture(0)

    font = cv2.FONT_HERSHEY_SIMPLEX

    #fontScale = 1

    #fontColor = (255, 255, 255)

    col_names=['Id','Name','Date','Time']

    attendance=pd.DataFrame(columns=col_names)
```

```
while True:
    ret,img=cam.read()
    gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces=faceCascade.detectMultiScale(gray,1.3,5)
    for(x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        Id,conf=recognizer.predict(gray[y:y+h,x:x+w])
        if 'Id' in df and conf>=4 and conf <= 85:#if (conf<50):
            ts=time.time()
            date=datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
            timeStamp=datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            aa=df.loc[df['Id']==Id]['Name'].values
            tt=str(Id)+"-"+aa
            attendance.loc[len(attendance)]=[Id,aa,date,timeStamp]
            cv2.putText(img,str(tt),(x,y+h),font,1,(255,255,255),2)
```

```
    if(conf>85):

        Id='Unknown'

        tt=str(Id)

        noOfFile=len(os.listdir("ImagesUnknown"))+1

        cv2.imwrite("ImagesUnknown\Image"+str(noOfFile)+".jpg",img[y:y+h,x:x+w])

        cv2.putText(img,str(tt),(x,y+h),font,1,(255,255,255),2)

    else:

        Id='Unknown'

        tt=str(Id)

        cv2.putText(img,str(tt),(x,y+h),font,1,(255,255,255),2)

attendance=attendance.drop_duplicates(subset=['Id'],keep='first')

cv2.imshow('img',img)

#if (cv2.waitKey(100) & 0xFF==ord('q')):

#    break

if(cv2.waitKey(1)==ord('q')):
```

```
        break
    ts=time.time()
    date=datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
    timeStamp=datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
    Hour,Minute,Second=timeStamp.split(":")
    fileName="Attendance\Attendance_" +date+"_"+Hour+"-"+Minute+"-"+Second+".csv"
    attendance.to_csv(fileName,index=False)
    cam.release()
    cv2.destroyAllWindows()
    res=attendance
    message2.configure(text=res)



clearButton1=tk.Button(window,text="Clear",command=clear,fg="red",bg="yellow",width=20,height=2,activebackground="Red",font=('times',15,'bold'))
```

```
clearButton1.place(x=950,y=210)

clearButton2=tk.Button(window,text="Clear",command=clear2,fg="red",bg="yellow",width=
20,height=2,activebackground="Red",font=('times',15,'bold'))

clearButton2.place(x=950,y=310)



takeImg=tk.Button(window,text="Take
Images",command=TakeImages,fg="red",bg="yellow",width=20,height=2,activebackground=
"Red",font=('times',15,'bold'))

takeImg.place(x=90,y=500)



trainImg=tk.Button(window,text="Train
Images",command=TrainImages,fg="red",bg="yellow",width=20,height=2,activebackground
="Red",font=('times',15,'bold'))

trainImg.place(x=390,y=500)



trackImg=tk.Button(window,text="Track
Images",command=TrackImages,fg="red",bg="yellow",width=20,height=2,activebackground
="Red",font=('times',15,'bold'))
```

```
trackImg.place(x=690,y=500)

quitWindow=tk.Button(window,text="Quit",command=window.destroy,fg="red",bg="yellow
",width=20,height=2,activebackground="Red",font=('times',15,'bold'))

quitWindow.place(x=990,y=500)

#message=tk.Label(window, text="Face Recognition Based Attandance Management
System",bg="Green",fg="White",width=50,height=3,font=('times',30,'italic bold underline'))

window.mainloop()
```