A PROJECT REPORT

ON

"PROXYCAM: SMART ATTENDANCE SYSTEM"

Submitted to

MNNIT

In Partial Fulfilment of 4th Semester the Requirement for the Award of Mini Project I (SRS and Design)(CS34601)

MASTER OF COMPUTER APPLICATION

BY

SANCHIT SINGH	**2018CA04**
ATANU SEN	**2018CA87**
ABHISHEK RANJAN	**2018CA60**

UNDER THE GUIDANCE OF
PROF. SUNEETA AGARWAL

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY

PRAYAGRAJ, U.P - 211004

2018-2021

A PROJECT REPORT
ON

"PROXYCAM: SMART ATTENDANCE SYSTEM"

Submitted to

MNNIT

In Partial Fulfilment of 4th Semester the Requirement for the Award of Mini Project I (SRS and Design)(CS34601)

MASTER OF COMPUTER APPLICATION

BY

SANCHIT SINGH	**2018CA04**
ATANU SEN	**2018CA87**
ABHISHEK RANJAN	**2018CA60**

UNDER THE GUIDANCE OF
PROF. SUNEETA AGARWAL

CSED

MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY

PRAYAGRAJ, U.P - 211004

2018-2021

MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering
PRAYAGRAJ, U.P - 211004

CERTIFICATE

This is certify that the project entitled

“ProxyCam: Smart Attendance System“

submitted by

SANCHIT SINGH	**2018CA04**
ATANU SEN	**2018CA87**
ABHISHEK RANJAN	**2018CA60**

is a record of bonafide work carried out by them, in the Partial Fulfilment of 4th Semester the Requirement for the Award of Mini Project I (SRS and Design)(CS34601) at , MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY. This work is done during year 2020, under our guidance.

Date: / /

(Prof. Suneeta Agarwal)
Project Guide

(Dr. Dinesh Kumar)
Project Coordinator

Acknowledgements

We are profoundly grateful to **Prof.Suneeta Agarwal** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express deepest appreciation towards **Dr.Dinesh Kumar**, as, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

SANCHIT SINGH
ATANU SEN
ABHISHEK RANJAN

ABSTRACT

Face is the representation of one's identity. Hence, we have proposed an automated attendance system based on face recognition. Face recognition system is very useful in life applications especially in security and pandemic situation. In our proposed approach, firstly, video framing is performed by activating the camera through a userfriendly interface. The face ROI is detected and segmented from the video frame by using (name of)algorithm. In the pre-processing stage, scaling of the size of images is performed if necessary in order to prevent loss of information. The median filtering is applied to remove noise followed by conversion of colour images to grayscale images. After that, contrast-limited adaptive histogram equalization (CLAHE) is implemented on images to enhance the contrast of images. In face recognition stage, enhanced local binary pattern (LBP) and principal component analysis (PCA) is applied correspondingly in order to extract the features from facial images. In our proposed approach, the enhanced local binary pattern outperform the original LBP by reducing the illumination effect and increasing the recognition rate. Next, the features extracted from the test images are compared with the features extracted from the training images. The facial images are then classified and recognized based on the best result obtained from the combination of algorithm, enhanced LBP and PCA. Finally, the attendance of the recognized student will be marked and saved in the excel file. The student who is not registered will also be able to register on the spot and notification will be given if students sign in more than once. The average accuracy of recognition is 100 Yale face database when two images per person are trained.

Contents

List of Figures

Chapter 1

Introduction

The main objective of this project is to develop face recognition based automated attendance system. In order to achieve better performance, the test images and training images of this proposed approach are limited to frontal and upright facial images that consist of a single face only. The test images and training images have to be captured by using the same device to ensure no quality difference. In addition, the person have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

Chapter 2

Problem Statement

Traditional attendance marking technique is often facing a lot of trouble. The face recognition attendance system emphasizes its simplicity by eliminating classical attendance marking technique such as calling names and fingerprint sensor for attendance is not applicable in terms of Covid-19 . Thus, face recognition attendance system is proposed in order to replace the manual signing of the presence of person . Furthermore, the face recognition based automated attendance system able to overcome the problem of fraudulent approach and. One of the difficulties of facial identification is the identification between known and unknown images. found out that the training process for face recognition student attendance system is slow and time-consuming. In addition, that different lighting and head poses are often the problems that could degrade the performance of face recognition based student attendance system. Hence, there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will be the evaluation points of the performance.

Objective

The objective of this project is to develop face recognition based automated attendance system. Expected achievements in order to fulfill the objectives are:

- To detect the face segment from the video frame.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the identified person.

Chapter 3

Software Requirements Specification

3.1 Software

Python library such as NumPY, Matplotlib, and OpenCV is the base library of our proect.

- Install python language in Your machine
- 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.
- Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.
- OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source BSD license.
- Pycharm for writing code. PyCharm is an integrated development environment used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.

3.2 Hardware

- Raspberry Pi: The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse.

It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

- 5 MP Camera: The Camera Module can be used to take high-definition video, as well as stills photographs. It's easy to use for beginners, but has plenty to offer advanced users if you're looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion, and other video cleverness. You can also use the libraries we bundle with the camera to create effects.

3.3 Operating System

- Windows with python, you can use the Pycharm editor.
- Linux with python.
- Raspberry Pi OS for Windows or linux.

3.4 Minimum Requirement

- 4 GB of RAM.
- 1 GB of Harddisk.
- Windows or linux with python and OpenCV pre installed.
- At least 5 MP camera.

Chapter 4

Requirement Analysis

4.1 Software

Making of ProxyCam inter-related to many libraries.

4.2 OpenCV

- OpenCV: OpenCV was started at Intel in the year 1999 by Gary Bradsky. The first release came a little later in the year 2000. OpenCV essentially stands for Open Source Computer Vision Library. Although it is written in optimized C/C++, it has interfaces for Python and Java along with C++. OpenCV boasts of an active user base all over the world with its use increasing day by day due to the surge in computer vision applications.

4.3 Installation

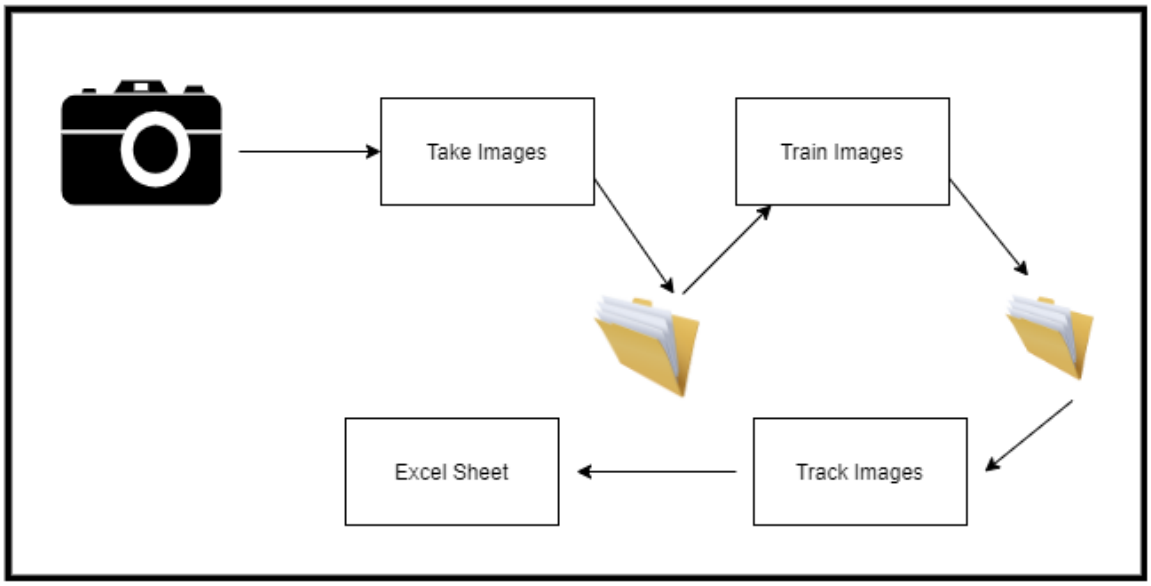
- OpenCV-Python supports all the leading platforms like Mac OS, Linux, and Windows. It can be installed in either of the following ways:
- run pip install opencv-python if you need only main modules
- run pip install opencv-contrib-python if you need both main and contrib modules.

Chapter 5

System Design

5.1 Methodology

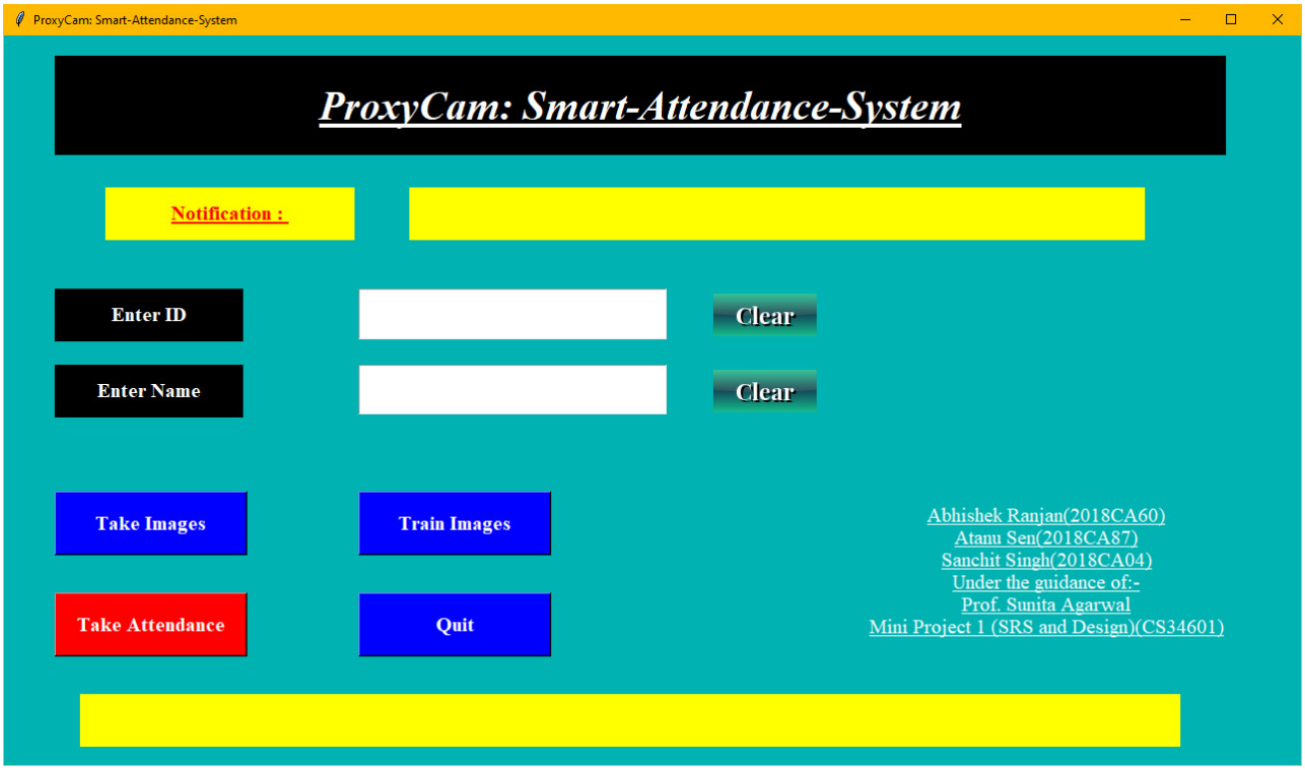
For making our software we are using 4 main function for perform our action.



- Basically we are taking image from Camera and our take images function will take 60 images and keep them in folder named TrainingImage.
- When will click on the train image button then function will call it will take all the data from TrainingImage folder and it makes a train.yml file with the help of Numpy Library and keep the trained data in TrainingImageLabel folder.
- When we click on the Track Images or Take attendance then it will open a window and start reconginzing face and it will match to train.yml file.
- What is train.yml file? train.yml file is a array of human face in which it store the value of face co ordinates and keep them in array.

- After clicking on Track images it maintain a excel sheet.

5.2 GRAPHICAL USER INTERFACE(GUI)



- For graphical user interface we are using Tkinter. What is tkinter? Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python. The name Tkinter comes from Tk interface.
- Four button associated with function named Take Images, Train Images, Take Attendance as Track Images and Quit button for closing the programme.

5.3 Functions

Take Images

```
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)
                #incrementing sample number
                sampleNum=sampleNum+1
                #saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage\ "+name +"."+Id +'.'+ str(sampleNum) + ".jpg", gray[y:y+h,x:x+w])
                #display the frame
                cv2.imshow('frame',img)
            #wait for 100 miliseconds
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            # break if the sample number is morethan 100
            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)
```

- In Take Images button actually we are taking 60 images of human face and keeping them in folder.
- Here we are using classifier, Classifier is an algorithm for face detection with co ordinates of face.
- For Algorithm we are converting our pictures in grayscale.

5.3.1 Train Images

```
def TrainImages():
    recognizer = cv2.face_LBPHFaceRecognizer.create()#recognizer = cv2.face.LBPHFaceRecog
    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 getImagesAndLabels(path):
    #get the path of all the files in the folder
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
    #print(imagePaths)

    #create empth face list
    faces=[]
    #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.append(imageNp)
        Ids.append(Id)
    return faces,Ids
```

- In train Images we have declared two function one function that is getImagesAndLabls is supporting function.
- When we call the TrainImages function it will recognize the human face all the co ordinates of the human face which is inside $haarcascade_frontalface_default.xml$ fetching and we are matching with all the value.

Track

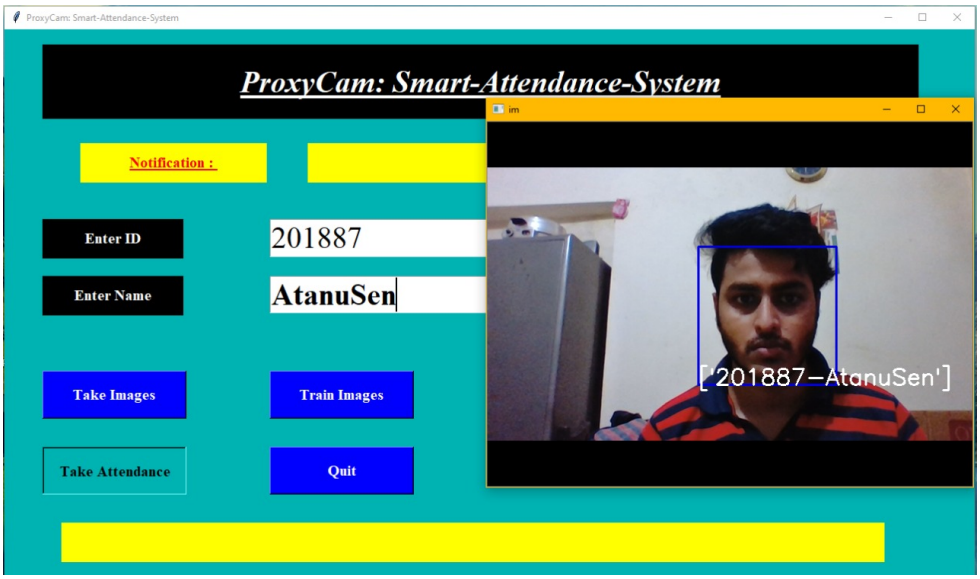
```
def TrackImages():
    recognizer = cv2.face.LBPHFaceRecognizer_create()#cv2.createLBPHFaceRecognizer()
    recognizer.read("TrainingImageLabel\Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);
    df=pd.read_csv("StudentDetails\StudentDetails.csv")
    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names =  ['Id','Name','Date','Time']
    attendance = pd.DataFrame(columns = col_names)
    while True:
        ret, im =cam.read()
        gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
        faces=faceCascade.detectMultiScale(gray, 1.2,5)
        for(x,y,w,h) in faces:
            cv2.rectangle(im,(x,y),(x+w,y+h),(225,0,0),2)
            Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
            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]

            else:
                Id='Unknown'
                tt=str(Id)
            if(conf > 75):
                noOfFile=len(os.listdir("ImagesUnknown"))+1
                cv2.imwrite("ImagesUnknown\Image"+str(noOfFile) + ".jpg", im[y:y+h,x:x+w])
            cv2.putText(im,str(tt),(x,y+h), font, 1,(255,255,255),2)
        attendance=attendance.drop_duplicates(subset=['Id'],keep='first')
        cv2.imshow('im',im)
        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()
    #print(attendance)
    res=attendance
    message2.configure(text= res)
```

- TrackImages is the core function of our project, It determines the human face and recongize them and keep them in excel sheet.

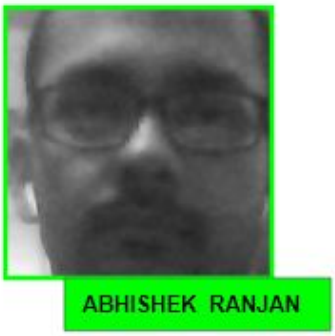
Chapter 6

System Testing

6.1 Run the Project



- Enter Id please Enter the id as number.
- Enter Name.
- Once you insert data for your name and Roll simply click the Take images and then after train images and then take attendance.



A	B	C
Abhishek		8:00 AM
Atanu		8:05 AM
Sanchit		8:06 AM
Prof. Suneeta Agarwal		10:00 AM
Dr. Dinesh Kumar		9:15 AM

- Once your system is recongized face you can convert it in pdf for future references.

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

ProxyCam is high demanding Technology in Market in terms COVID 19. It's has some bugs but we can fix it. Our system is reachable to all department, college and school, or in industries. Our system taking only 5 second to maintaine excel sheet of the institute's student and employee.

7.2 Future Scope

- Our system (ProxyCam) has very demanding for future because we can develop an android application and which linked to our system, whenever a person have COVID 19 Symptom, it will alert to all other member of dempartment.
- We want to add a thermal camera for taking body temparture of human and whenever a person has above 38 degree celcius then our system will beep.
- Once our system determine that a person have more than normal temparature then with data we will send to android app and it will message all the department members that this person have COVID 19 symptoms.

References

ReferencesDownload Raspberry Pi OS
https://www.raspberrypi.org/downloads/
ReferencesLearn Python
https://www.python.org/
ReferencesLearn OpenCV
https://www.learnopencv.com/
ReferencesHow to install OpenCV
https://docs.opencv.org/master/d5/de5/tutorial_py_setup_in_windows.html
ReferencesHow OpenCV and face detection works
https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/
ReferencesOur Project on GitHub
https://github.com/abhishekranjan7/ProxyCam