PROJECT PAPER

DS-630: machine learning

Professor: Dr. Lee obtained

HUMAN FACE RECOGNITION ATTENDANCE SYSTEM

GROUP MEMBER:

KHUSHALIBEN HARESHBHAI DIXIT

BIJNABEN A CHANDERA

THANUBOADHI NAVEEN REDDY

Introduction:

As we are making a system which can recognize face and match with its own database.

It will make the attendance system more authentic.

Our primary goal is to help different organizations to improve and organize the process of track and manage students or employees' attendance and absenteeism.

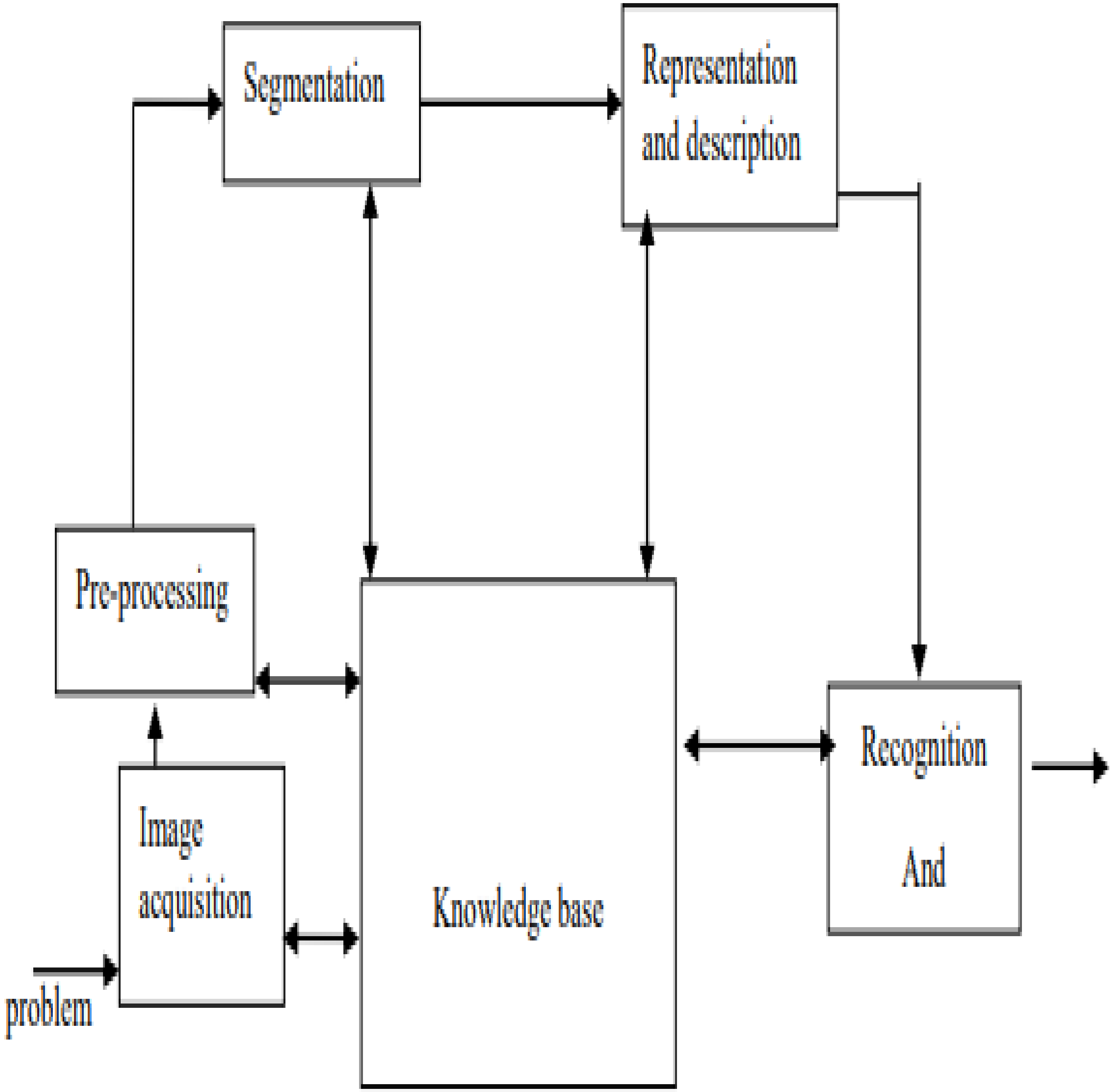
About:

Face detection is a type of computer vision technology that is ableto identify people's faces within digital images.

This is very easy for humans, but computers need precise instructions.

The images might contain many objects that aren't human faces,like buildings, cars, animals, and so on.

Plan Of Works:



Software Used:

Program Language: Python 3 With Its Libraries

Software:

Pycharm Edu Code

Libraries that we Used for this project:

appdirs

attrs

black

Click

cycler

demjson

kiwisolver

matplotlib

nose

numpy

opencv-contrib-python

pandas

Pillow

pyparsing

python-dateutil

pytz

six

toml

virtualenv

xlrd

xmltodict
yagmail
yapf

GUI (Graphical User Interface):

GUI provides a better experience in working with software.So, we
are providing our software with a GUI.
GUI will help even non-technical person to work with ease in oursoftware.

A nice GUI will help every user to get his/her work done

ADVANTAGES:

The system stores the faces that are detected and automatically marks attendance.

Ease of use is to recognize the faces in real time, using multiple face detection.
Multipurpose software can be used in different places.

LIMITATIONS:

The accuracy of the system is not 100%. It can only detect face from a limited distance.

FEATURES:

A Feature Is A Piece Of Information In An Image That Is Relevant ToSolving A Certain Problem.

It Could Be Something As Simple As A Single Pixel Value, Or MoreComplex Like Edges, Corners, And Shapes.

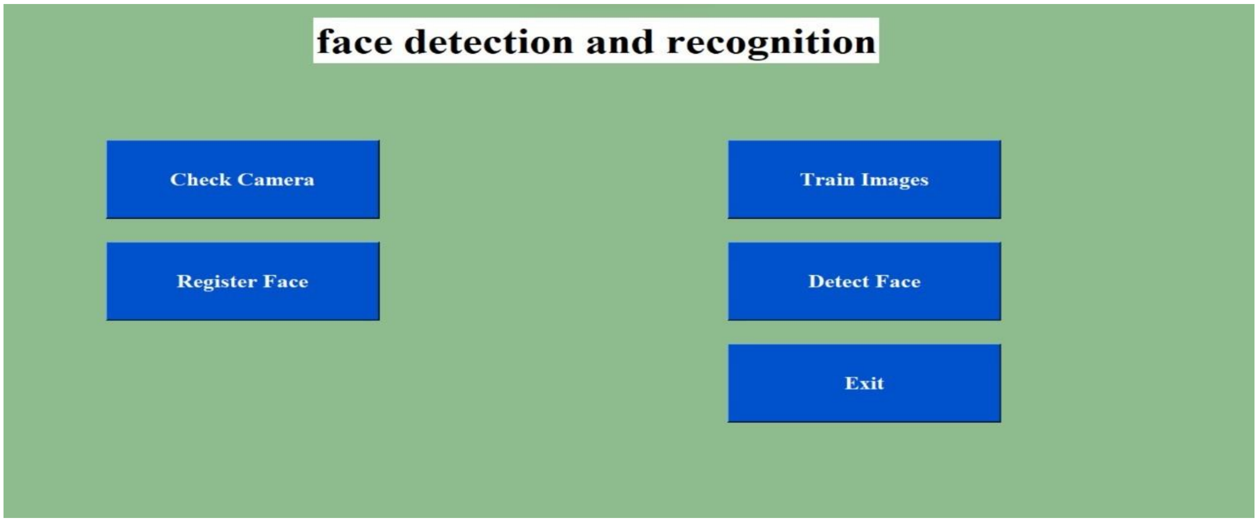
You Can Combine Multiple Simple Features Into A Complex Feature.

Applying Certain Operations To An Image Produces InformationThat Could Be Considered Features As Well.

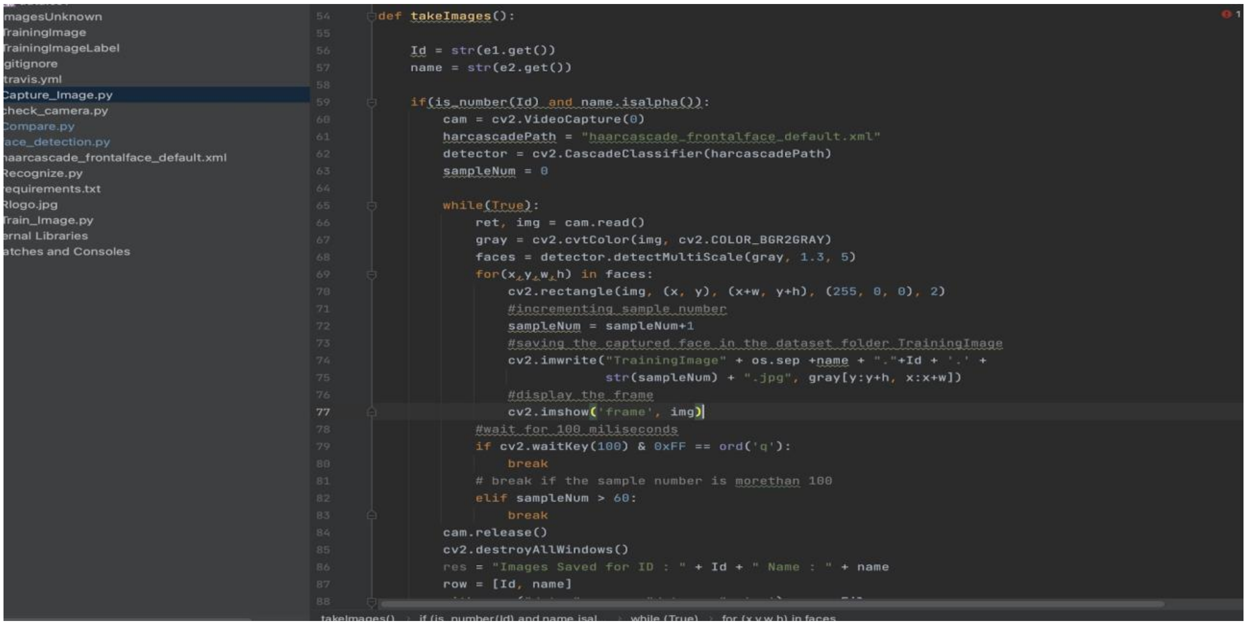
Computer Vision And Image Processing Have A Large Collection OfUseful Features And Feature Extracting Operations.

Basically, Any Inherent Or Derived Property Of An Image Could BeUsed As A Feature To Solve Tasks.

WHAT WE DO:



CAPTURE IMAGE



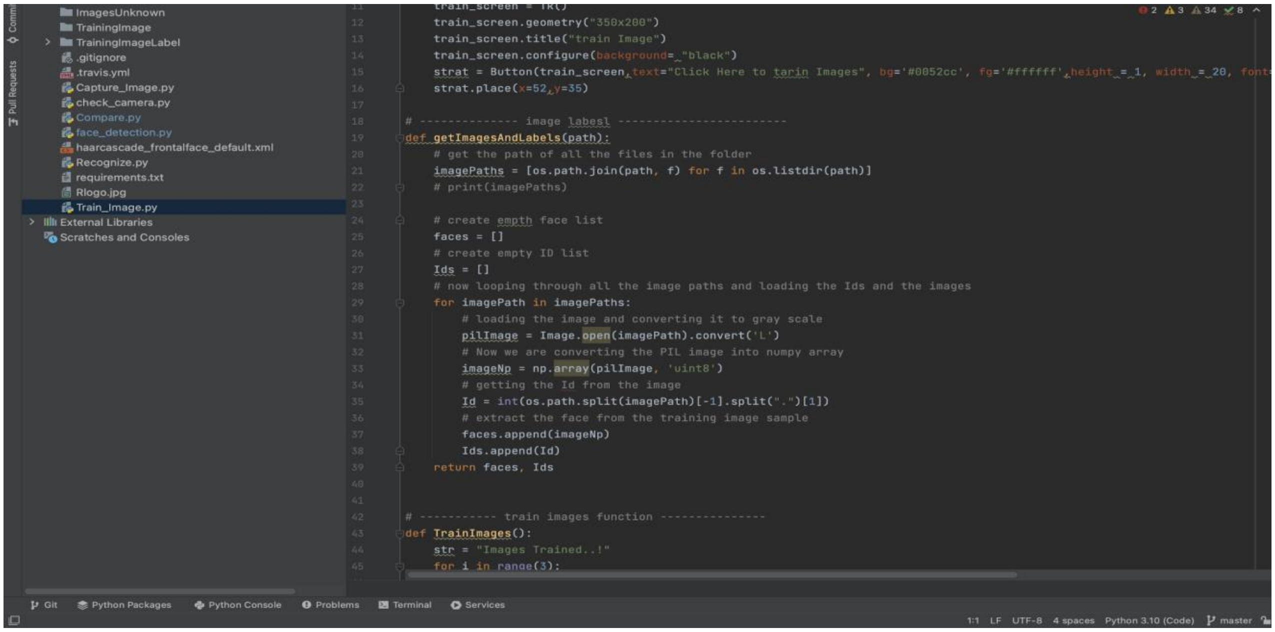
```python
def takeImages():

    Id = str(e1.get())
    name = str(e2.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" + os.sep +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]
```

TRAIN IMAGE



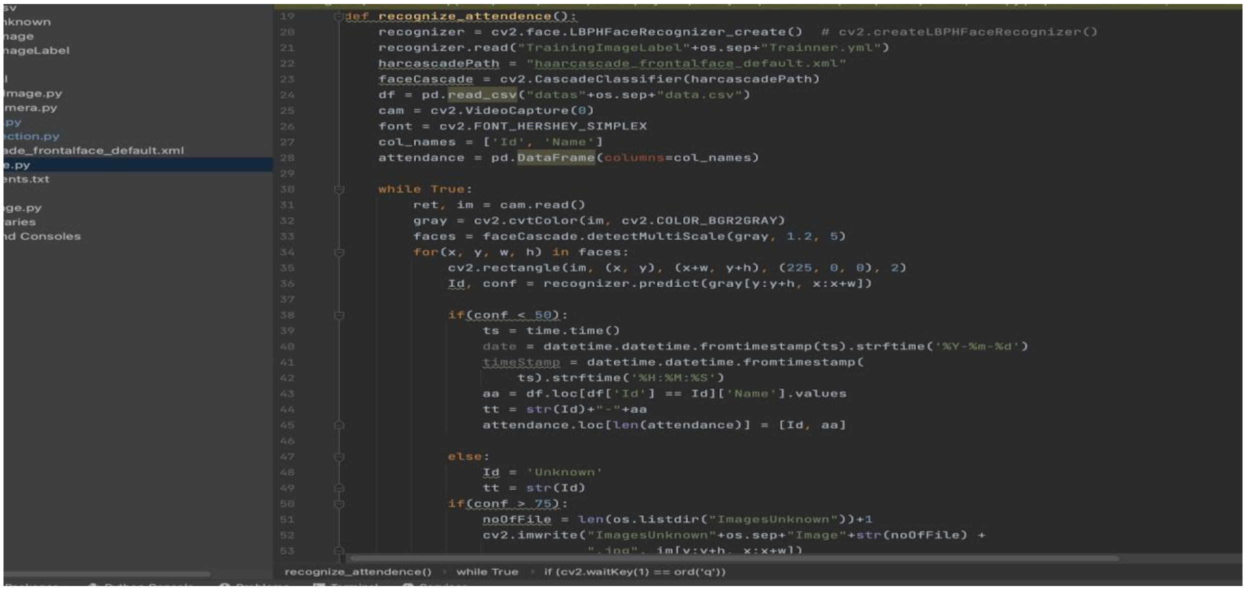
```python
    train_screen = Tk()
    train_screen.geometry("350x200")
    train_screen.title("train Image")
    train_screen.configure(background=_"black")
    strat = Button(train_screen,text="Click Here to tarin Images", bg='#0052cc', fg='#ffffff',height_=_1, width_=_20, font
    strat.place(x=52,y=35)

# ------------- image labesl -----------------------
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


# ----------- train images function ---------------
def TrainImages():
    str = "Images Trained..!"
    for i in range(3):
```

RECOGNIZE


sv
nknown
nage
nageLabel
l
Image.py
mera.py
.py
ection.py
ade_frontalface_default.xml
e.py
ents.txt
ge.py
aries
nd Consoles
recognize_attendence() › while True › if (cv2.waitKey(1) == ord('q'))

```python
def recognize_attendence():
    recognizer = cv2.face.LBPHFaceRecognizer_create()  # cv2.createLBPHFaceRecognizer()
    recognizer.read("TrainingImageLabel"+os.sep+"Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath)
    df = pd.read_csv("datas"+os.sep+"data.csv")
    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names = ['Id', 'Name']
    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]

            else:
                Id = 'Unknown'
                tt = str(Id)
            if(conf > 75):
                noOfFile = len(os.listdir("ImagesUnknown"))+1
                cv2.imwrite("ImagesUnknown"+os.sep+"Image"+str(noOfFile) +
                            ".jpg", im[y:y+h, x:x+w])
```

REPORT GENERATION:

Here We Created A Report Which Will Hold The Attendance Of The Students For Everyday Records.

This Report Will Have Information Of A Student Like Name & Id.

In This Report We Can Directly Get The List Of Students Who ArePresent And Absent.

RESULT ANALYSIS:

In Our Project We Have Been Working More The 120 Pictures. OurResult Of Our Project Percentages Is Almost 50-55%.

Though It Is Not Enough For This Little Dataset.

To Make Almost 98% Accuracy We Need To Use More PowerfulHardware And Also Need More Resources.

REFERENCE:

https://realpython.com/face-recognition-with-python/

https://developers.google.com/mediapipe/solutions/vision/face_detector/python

https://neptune.ai/blog/object-detection-algorithms-and-libraries

https://pypi.org/project/object-detector/