Face Detection and Recognition System

Project report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology in

Computer Science and Engineering

By

Gurupdesh Singh(171261)

Under the supervision

of

Dr. Monika Bharti

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CERTIFICATE

Candidate’s Declaration

I hereby declare that the work presented in this report entitled “ **Face Detection and Recognition System**” in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out under the supervision of **Dr.Monika Bharti** ,Department of Computer Science Engineering and Information Technology.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Gurupdesh Singh

Gurupdesh Singh (171261)

This is to certify that the above affirmation made by the candidate is true to the best of my knowledge.

Monika Bharti

Dr. Monika Bharti

ACKNOWLEDGEMENTS

I would like to take the opportunity to thank and express our deep sense of gratitude to mentor and project guide **Dr. Monika Bharti** for his immense support and valuable guidance without which it would not have been possible to reach at this stage of our major project.

I also obliged to all our faculty members for their valuable support in their respective fields which helped us in reaching at this stage of our project.

ABSTRACT

With every passing day, we are becoming more and more dependent upon technology to carry out even the most basic of our actions. Facial detection and Facial recognition help us in many ways, be it sorting of photos in our mobile phone gallery by recognizing pictures with their face in them or unlocking a phone by a mere glance to adding biometric information in the form of face images in the country’s unique ID database (Aadhaar) as an acceptable biometric input for verification.

This project lays out the basic terminology required to understand the implementation of Face Detection and Face Recognition using Intel’s Computer Vision library called ‘OpenCV’.

It also shows the practical implementation of the Face Detection and Face Recognition using OpenCV with Python embedding on both Windows as well as macOS platform. The aim of the project is to implement Facial Recognition on faces that the script can be trained for. The input is taken from a webcam and the recognized faces are displayed along with their name in real time.

This project can be implemented on a larger scale to develop a biometric attendance system which can save the time-consuming process of manual attendance system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

OpenCV	Open Computer Vision
Numpy	Numerical Python
PIL	Python imaging library
LDA	Linear Discriminant Analysis
LBPH	The Local Binary Pattern Histogram
XML	Extensible Markup Language

Chapter 1: Introdution

1.1 Introduction

A face recognition system could also be a technology which match a given install profile a personality's face from images and videos which it has or use it as a reference to map and identify against an info of faces. Researchers area unit presently developing multiple ways throughout that face recognition systems work. the foremost advanced face recognition methodology, that is to boot used to manifest users through ID verification services, works by pin pointing and mensuration countenance in image

While at first kind of laptop application, recognition system seen wider uses in recents times on smartphones and in alternative kinds of technol ogy, like artificial intelligence. as a result of computerised face recognition involves themeasuring of a human's physiological characteristics face recognition system area unit classified as bioscience. though the accuracy of face recognitions system as a biometric technology is a smaller amount than iris recognition and handprint recognition, it wide adopte because of its contactless and non-invasive method.Facial recognition systems area unit deployed in advanced human -computer interaction, video police work and automatic compartmentalisation of pictures.

We have a created a face recognition technology capable of identifying faces.

1.2 PROBLEM STATEMENT

“Facial Detection and Facial Recognition using Intel’s open source Computer Vision

Library (OpenCV) and Python dependency”

There are various scripts illustrated throughout the project that will have functionalities like detecting faces in static images, detecting faces in live feed using a webcam, capturing face images and storing them in the dataset, training of classifier for recognition and finally recognition of the trained faces.

All the scripts are written in python 3.6.5 and have been provided with documented code. This project lays out most of the useful tools and information for face detect ion and face recognition and can be of importance to people exploring facial recognition with OpenCV.

The project shows implementation of various algorithms and recognition approaches which will be discussed on later in the project report.

Face Recognition can be of importance in terms of security, organization, marketing, surveillance and robotics etc.

Face detection is able to very immensely improve surveillance efforts which can greatly help in tracking down of people with ill criminal record basically referring to criminals and terrorists who might be a vast threat to the security of the nation and the people collectively.The Personal security is also greatly exacerbated since there is nothing for hackers to steal or change, such as passwords.

1.3 OBJECTIVES

This project is created so as to study the various means of recognizing faces with more accuracy and reducing the error rates while recognition. The ideal condition for any recognition project is to reduce the intra class variance of features and increase the inter class variance of features to be detected or recognized.

Facial Recognition software is “Capable of uniquely identifying or verifying a person by comparing and analyzing patterns based on the person’s facial contours. It is mostly used for security purposes”.Many other areas of use.

Different recognizer approaches are used for recognition of faces. They are:

- Eigen Faces
- Fisher Faces
- Local Binary Pattern Histograms

1.4 METHODOLOGY

The Project utilizes various libraries of Python such as

1.4.1 OpenCV

“OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library”. The main purpose of this was common infrastructure for computer vision application and it was built specifically for such purposes not to mention it also accelerated the machine perciption inside the business product. “Being a BSD-licensed product, OpenCV makes it straightforward for businesses to utilize and modify the code”.

In total we can say that The library has about 2000 optimiz algorithm which is really insane, “These algorihms contain a comprehensive set which comprises of each classic and progressive laptop vision and machine learning algorithms. These algorithms area unit usually accustomed sight and acknowledge faces, determine objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, manufacture 3D purpose clouds from s tereo cameras, sew pictures along to produce a high resolution image of a full scene, realize similar pictures from an image info, take away red eyes from pictures taken exploitation flash, follow eye movements, acknowledge scenery and establish markers to overlay it with increased reality, etc”. The amazing thing about this library is that it has quite about forty seven thousand individuals of user community and calculable variety of downloads olympian eighteen million. The library is utilized extensively in corporations, analysis teams and by governmental bodies.

Along with well-established corporations like “Google, Yahoo, Microsoft, Intel, IBM,

Sony, Honda, Toyota” that use the library, there area unit several startups like “Applied Minds, VideoSurf, and Zeitera”, that create in depth use of OpenCV. OpenCV’s deployed wide array spans vary from sewing streetview pictures along, police work intrusions in police work videoin Israel, watching mine instrumentality in China, serving to robot navigated and devour objects at “Willow Garage, detection of natatorium drowning accidents in Europe, running interactive art in Espana and New York , checking runways for scrap in Turkey.

1.4.2 NumPy

“The Python programming language earlier wasn't originally designed for numerical computing as we know it to be , however it also attracted the attention of the scientific and engineering community early” . “In 1995 the interest (SIG) matrix-sig was based with the aim of shaping associate array computing package; among its members was Python designer and supporter Guido van Rossum, WHO extended Python's syntax (in explicit the compartmentalization syntax) to make array computing easier”.

“implementation of matrix packages was complete by Jim discoverer, then generalized further rationalization required by Jim Hugunin and known as Numeric (also diversely observed because the "Numerical Python extensions" or "NumPy").Hugunin, a collegian at the Masachusetts Institut of Technology (MIT), joine the Corporatio for National analysis Initiativ (CNRI) in 1997 to work on J Python, leaving Paul Dubois of Lawrence Livermore National Laboratory (LLNL) to need over as supporter. Others early contributor embrace David dAscher, Konred Hinsan and Traves Oliphent”.

“A new package known as Numarray was written as a additional versatile replacement for Numeric.Like Numeric, it too is currently deprecated.Numarray had quicker operations for large arrays, however was slower than Numeric on tiny ones, thus for a time each packages were utilised in parallel for varied use cases. The last version of Numeric (v24.2) was discharged on St Martin's Day 2005, whereas the last version of numarray (v1.5.2) was discharged on twenty four August 2006”.

There wa want urge Numerics into the Pythons customary library, however Guido van Rosum determined that code wasn't reparable inits state.

“In early 2005, NumPy developer Travis Oliphant needed to unify the community around one array package and ported Numarray's options to Numeric, cathartic the result as NumPy one.0 in 2006.This new project was a region of SciPy. To avoid putting in the large SciPy package simply to urge associate array object, t his new package was separated and known as NumPy. Support for Python three was other in

2011 with NumPy version one.5.0”.

pypy start developments on associate implementations of the numPy Api. It is not nevertheless absolutely compatable with numPys.

1.4.3 Pillow

The Python Imaging Library (PILLOW) or generally called as PIL is very useful for adding image processing capabilitiy to your Python interpreter which you have or have installed when working with Pyhon.

The main use of this library is that this library has a wide array of extensive file format support which allows it to run and store different files in different formats, an efficient representation which is further boosted by very powerful image process ing ability.The important point is that the image in librarie is meant for faster accessing to that date is stored during a few basic pixels formats.All this enable it to be a very commanding and powerful tool for image processing general, which is probably also the reason why it is used so heavily.

Let’s see a couple of possible things in which we can use the library.

Image Archive-The “Python Imaging Library” is rightin image archivalal and execution applications. you'll uses the libraryto make thumbnail, convert between file formats printf images etc.

From the information available to us currently about the current version is that it identifies and reads an outsized number of file formats. Write support is intentionally restricted to the foremost commonly used interchange and presentation formats.

“Image Display-The current release includes Tk PhotoImage and BitmapImage interfaces, also as a Windows DIB interface which will be used with PythonWin and other Windows-based toolkits. Many other GUI toolkits accompany some quite PIL support”.“For a purpose such as debugging there are a lot of functions like the show() method which saves a picture to disk, and callsit an external display utility.

Image ProcessingThe library contain basics image process functionalities, include point operation, filteres a group of built-in convolution kernel, and colours spacees conversion .Basically in short “Python Imaging library” is a free additional open source library which you have to install first by using the command such as pip so that you can use it to run various python functions in the module.It is important to note and keep in mind that all of these fucntions only work once you import this library and cannot be used if the library isnt imported or called upon.Henceforth why is it recommended to install this library beforehand inorder to properly use it.In regards to the main function of this library is to add functionality in opening, manipulating and storing of different file format images which can later be used to perform operations on accordingly to the needs of the programmer using the library.

1.4.4 FACE RECOGNITION

The “Face recognition” library in python is a library which helps in recognizing and manipulating the faces by using the programming language pythons or froms the comd line the simplest face recognitions libraries after importing the module and accessing the required functions.The “Face recognition” library was built using dlib’s “state-of-the-art face recognition” and was further enhanced and builts deep learning. The model have accurace of 99.37%.It is used to find faces in pictures

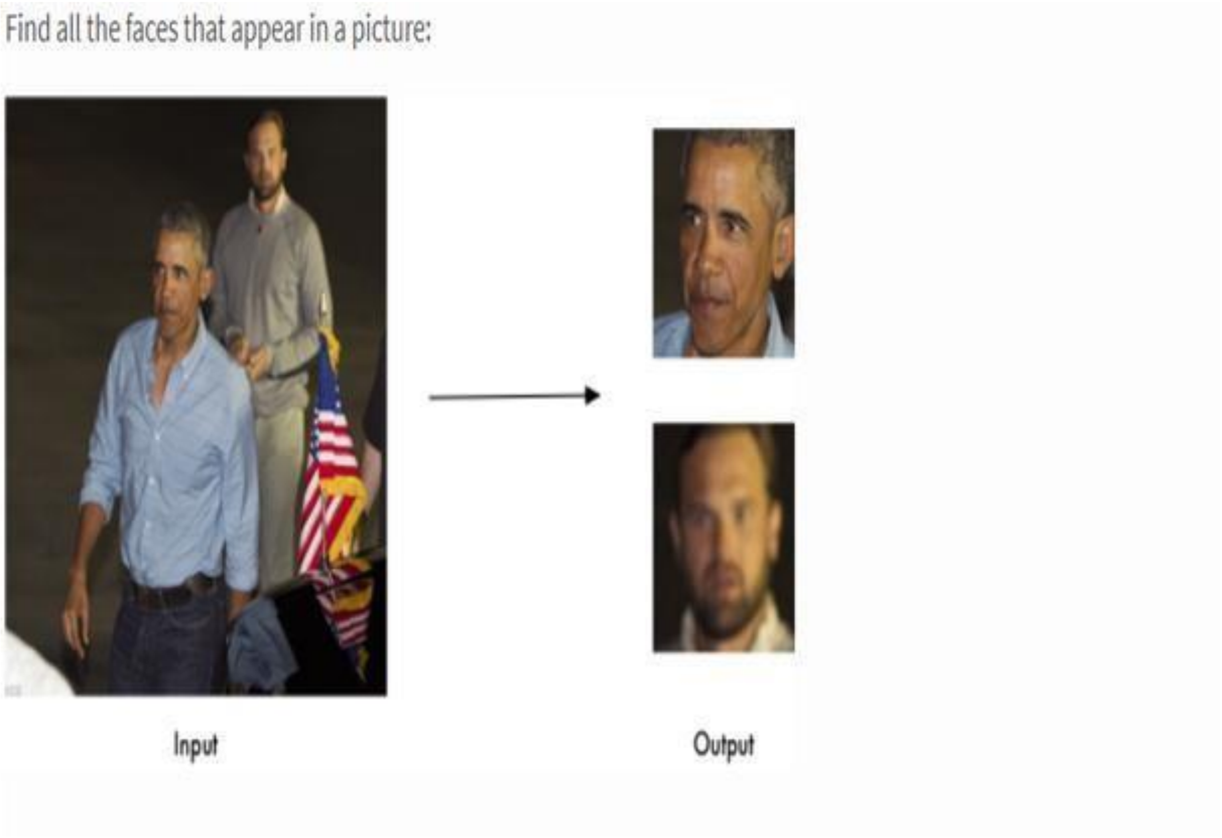


Figure 1.1: Finding all the faces in the picture

Find and according look for   facial features in the pictures

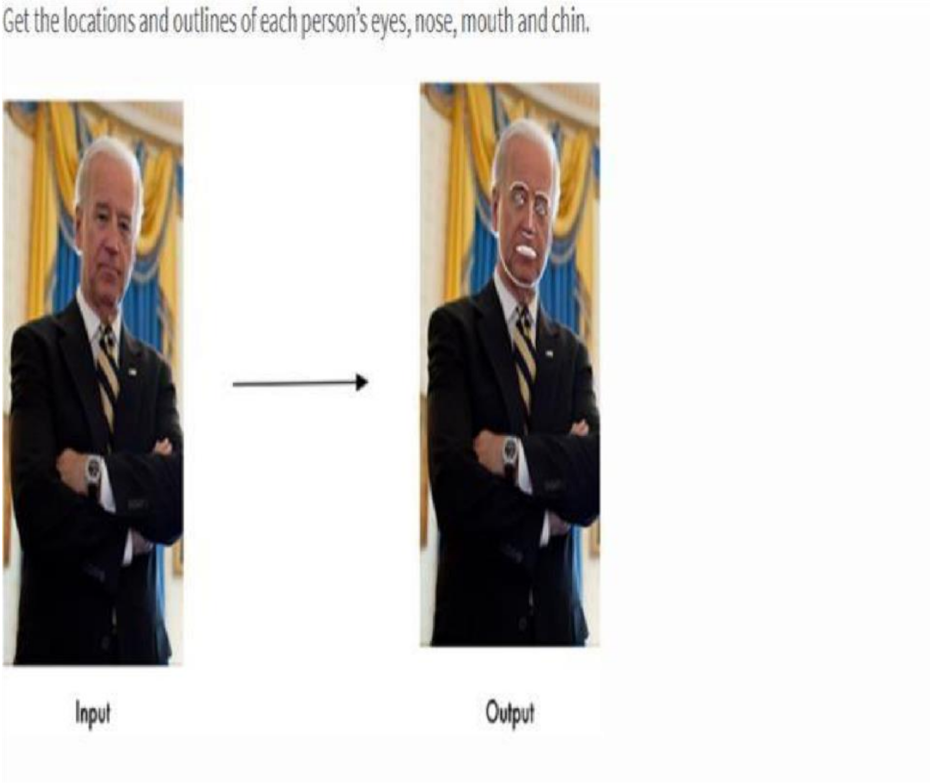

Get the locations and outlines of each person's eyes, nose, mouth and chin.
Input
Output

Figure 1.2:Getting the locations

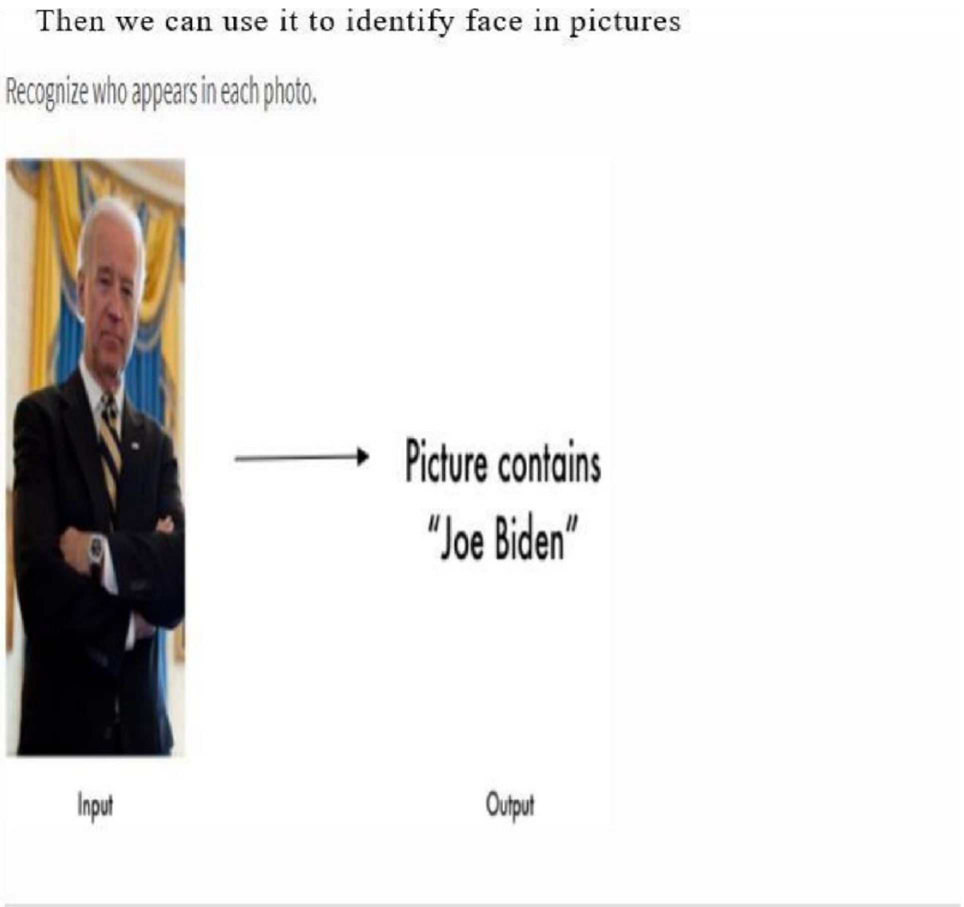

Then we can use it to identify face in pictures
Recognize who appears in each photo.
Picture contains "Joe Biden"
Input
Output

Figure 1.3: Recognizing who is in the Picture

1.4 Organization

Biometrics in simple terms basically refers to evaluating and measuring your Fingerprints,face features or any such human parameters that help to identify an individual.The study of Biometrics is very much important considering how bleak the security as a whole has become.Note that identification is only possible since each and every person has unique and distinct features which make it easier to identify individuals.

Keep in mind that the biometric feature which is use or are used must available in databases for all of individual in communities .before features or those features is use for other authentications . This is called enrolment.

Authentications is in one of these following forms

- Identificationss: This basically refers to matching an individual featured in againsts the record to verify whether his or her or simply put that if or not the records arethere in the database or not

- Verifications: “To check whether the person is who he/she is claiming to be”.In other words to check whether their claim of identity is true or not.In this case the other features of another person is matched only with the pre installed features of the person, they claim themselves to be .

Different types of Biometric are present:

There are another two present broad categories of biometrics:

1. Physiological Biometric
2. Behavioural BiometricPhysiological Biometric:

“As the name sounds out to be in this the physical traits of a person are measured greatly for identification and verification in this type of biometrics. The trait should be chosen such that it is unique among the general population, and no mater what is resistant to changes due to illness, aging, injury, etc”’.

Physiologicals Biometric Technique:

- handprint:handprints are one of the main methods to uniquely identify an individual.They are also regraded as one of the best methods for this sole purpose and many systems utilize this sort of recognition method. They have the uniques for every individual person and can be measured in variosus different way . Minutia-based measurement uses graphs which are there to match the ridges contained whereas imagebased measurments in finds similarities or the similar patterns inbetween the fingertips image and fingerprint images which are present or which have been uploaded in the database for the purpose of matching and uniquely identifying the identity of the person concerned. The security level is very high and use both in identifications and verifications. However, because to old age and different diseases/injury, handprints might gets alter.

in smartphons for verifications, in offices for other identifications.

- FacialRecognition: This basically refers to the set of the featured of face suppose distance between nose and the mouths or say the distance between the ear, length of whole face, skin color, are use In verifications and identifications in this regard. However there can be certain complications arising due to the complexity in aging, diesease,wearing sunglasses or any sort of face that disrupts to recognize those featrues and indirectly hampering the outcome greatly by playing a role in the low accuracy of the results.

- Iris and Retina: Not only just fingerprints the patterns found in say the iris and retina are also unique metrics which can be used to deicde or identify the concerned Device in analyses of retina is expensive and are use less common daily life. Disease cataract may hinder the pattern recognition of the iris and may cause discrepancies to occur

- Voice Recognition: The third kind of recognition that can be used is the voice recognition which is also very helpful.The pitches, voice modification, and tone, instead thing is takan into consideration and are taken in the dataset . Regarding the security of the system,it isnt necessarily a great choice in that regard since two different people can have same voice and henceforth the model will have to deal with a lot of problems and wich is why it isnt all that great and isnt used mostly in recognition. The accurace are hindered due to the present noise, or due to aging and illness.

- DNA: DNA might be the most unique kind of authentication ,since it is the most trusted metric for uniqely identifying an inidividual. Thus, securities are high and may be include in both identifications and verifications

Behavieral Biometric:

In this generally measure, which relate to the behaviour patterns of the individual and which can be a great source of auhentication

- Signature: “Signature might just be one of the most used metric for authentication . They are used to verify checks by matching the signature of the check against the signature present in the database. Signature tablets and special pens are used to compare the signatures”. timing requirement in writing the signature may be use in increase the accuracy. Signatur are mostly used in verifications.

- Keystrok Dynamic: This technique is used to measure the behavior of the person we are storing in database . Some of the characteristics take into account are:

1. Type speed-refers to how fast the person can type and matches that
2. Frequency of errors-what is the frequency of the errors committed
3. Duration of key depressions

CHAPTER- 2

LITERATURE SURVEY

2.1 Face Tracking

Face tracking refers to identifying the features which are then used to detect a Face In this case the example method includes the receiving or we can say that it gets first images and the second images user face who is being taken into consideration, where one or both of the images which were used to sort of look for a match have been granted a match by the facial recognition system which also proofs the correct working of the system. “The technique includes taking out a second sub- image coming from the second image, where the second sub-image includes a representation of the at least one corresponding facial landmark, detecting a facial gesture by determining whether a sufficient difference exists between the second sub - image and first sub-image to indicate the facial gesture, and determining, based on detecting the facial gesture, whether to deny authentication to the user with respect to accessing functionalities controlled by the computing” [1]

2.2 Mechanisms of human facial recognition

Basically what we see in this paper is that it presents an extension and a new way of perceptions of the theory of author for sapiens visual process information, in which method in extracting the second image of image second, where image of second includes a representations at least one corresponding of facial given landmarks. “In turn detecting a facial gesture by determining whether a sufficient difference exists between the second sub-image and first sub-image to indicate the facial gesture, and determining, based on detecting the facial gesture, whether to deny authentication to the user with respect to the human recognition system and same was applied”.Several indispensable techniques is implicat: encoding of visible photographs pattern of nural , detect features of facial, measurement standard, discount patterns of neural in dimensionality.

“The logical (computational) role suggested for the primary visual cortex has several components: size standardization, size reduction, and object extraction”. “The given result inprocessing of primary cortexs, it suggest,visual patternis encoding in suitable size instoraging. “(In this context, object extraction is the isolation of regions in the visual field having the same color, texture, or spatial extent.)”in topology map from cortex to retina, retina connection , primary visual cortex and geniculate bodies, and the cortex structure itselfs encode the visual patterns combined In this theory it is already illustrate the human graphically faces kind of primary stimulus. However, facial recognition is not limited to but Gestalt recognition to pertains of any class of familiar objects or scenes.

2.3 Eye Spacing Measurement for Facial Recognition

Few procedures to computerized facial consciousness has geometric size employe of attribute points of a sapien face. Space between eyes dimension has been recognized an essential stepin reaching the goal. Measurements in spacing has been made by mean of software of the hough radically change method in discovering occasion of a round form and ellipsoidial form which approx the iris’s perimeters and parameter of each of the sclera and the form the place under respective eyebrows. Both gradientx magnitudex anddirection of gradient used in handling noise contaminate the space feature. “Results of this application indicate that measurement of the spacing by detection of the iris is the most accurate of these three methods with measurement by detection of the position of the eyebrows the least accurate. However, measurement by detection of the eyebrows' position is the least constrained method. Application of these strategies has led to size of a attribute function of the human face with adequate accuracy to advantage later inclusion in a full bundle for computerized facial consciousness”..

2.4 A direct LDA algorithm for the high - dimensional data with applications of the recogniting face

“Linear discriminant analysis (LDA) has been successfully used as a dimensionality reduction technique to many classification problems, such as speech recognition, face recognition, and multimedia information retrieval”.The objectives of "nd projections A that maximized the ratio inbetween-class scatter against within - classes scatter

CHAPTER - 3

SYSTEM DEVELOPMENT

3.1 Recognizing the model by listing out the applications

- Face recognitions: Face recognitions system is a technologys that is capables in detecting the objects and verify the person in video and image from running video . There are m ultiple methods with which face recognitions system are working, , it works on c omparing selective face featured image of faces within database. It is als o describ the Biometrics AI applications based that can unique ide ntify a person of analysing patterns of person with different face structure and color. It can be use in surveillance camera, human computer interface and image databases manageme nt.

- Photography: Some recently image capturing camera device is used facial detecting technology for autofocusing. Facial detecting used in selecting the various regions that show in photos or slideshow and use a pan – and - scales Ken Burn effect.

- Marketings: Facial detections systen is gaining the interest in marketing. A webcamera may be integra ted into the television and detecting face that walks by.

3.2 Viola - Jones Algorithm

The Viola -

Jones algorithm is use everywhere now these days, mechanisms of detection of objects. The mainly the training of this algorithm is very slow, but have fast detection rate. Haar feature is used in This algorithm for filtering the feature.

The efficien of this algorithm viola jone is significantly increasing day by day generates the image integral.

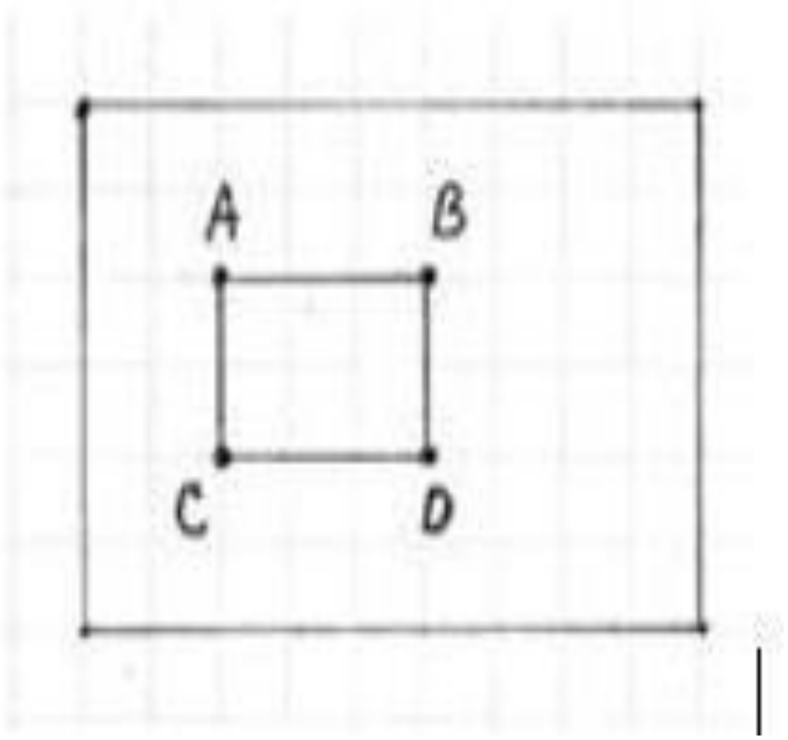


Figure 3.1: Recatangle used

A maximum and minimum window size i s chosen, and for each size a sliding step sizes arechosen. Then the detection window is mo ved across the image as follows:

1. Sets as minimum size window, and step sliding correspond to the size.

2. For the size window, window is slidded vertically and horizontally with the sam e step. At each and every steps, sets of N facial recognitions system filtes are applied every time. If thefilters give us the posit ive answer then the facial is detected.

3. If the window size is the maximum size stop the procedure. Otherwise increase t he size of the window and corresponding sliding step to the next chosen size and go to the step 2.Each face recognition filter (from the set of N filters) contains a set of cascade-connected classifiers. Each classifier looks at a rectangular subset of the detection wind ow and determines if it looks like a face. If it does, the next classifier is applied. If all clas sifiers give a positive answer, the filter gives a positive answer and the face is recognized . Otherwise the next filter in the set of N filters is run.

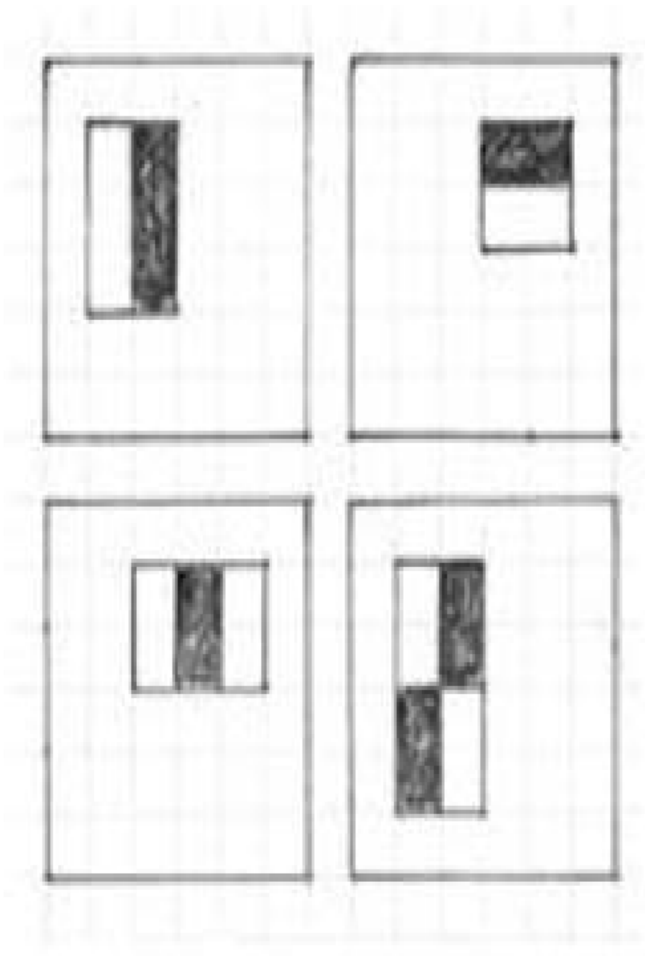


Figure 3.2:Classifiers

Object Detection using Haar feature-

based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection is using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade functi on is trained from a lot of positive and negative images. It is then used to detect objects in other images.

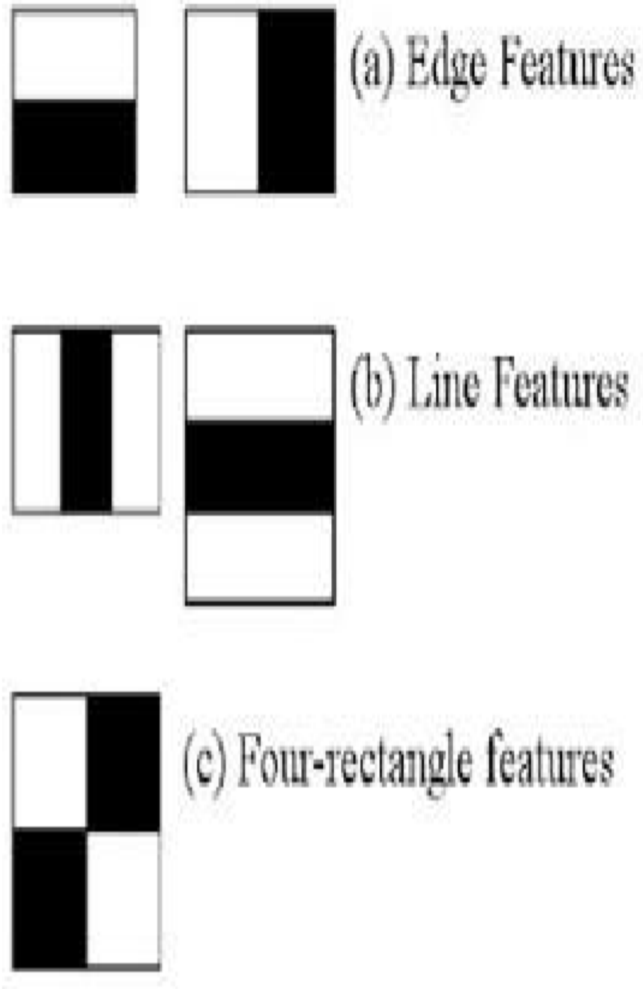


Figure 3.3

Now, all possible sizes and locations of each kernel are used to calculate lots of features.

(Just imagine how much computation it needs? Even a 24x24 window results over 16000 0 features). For each feature calculation, we need to find the sum of the pixels under whit e and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just fo ur pixels. Nice, isn't it? It makes things super-fast.

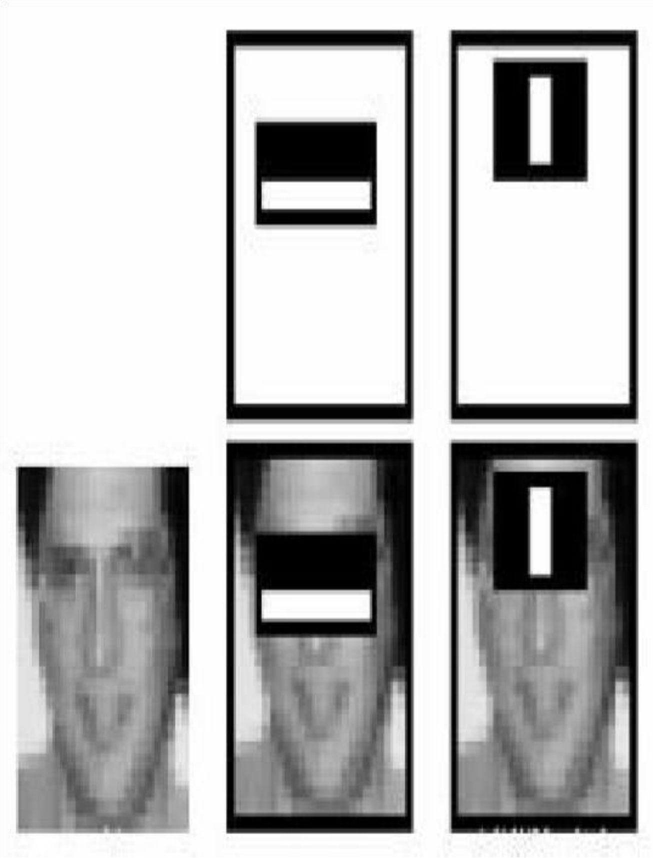
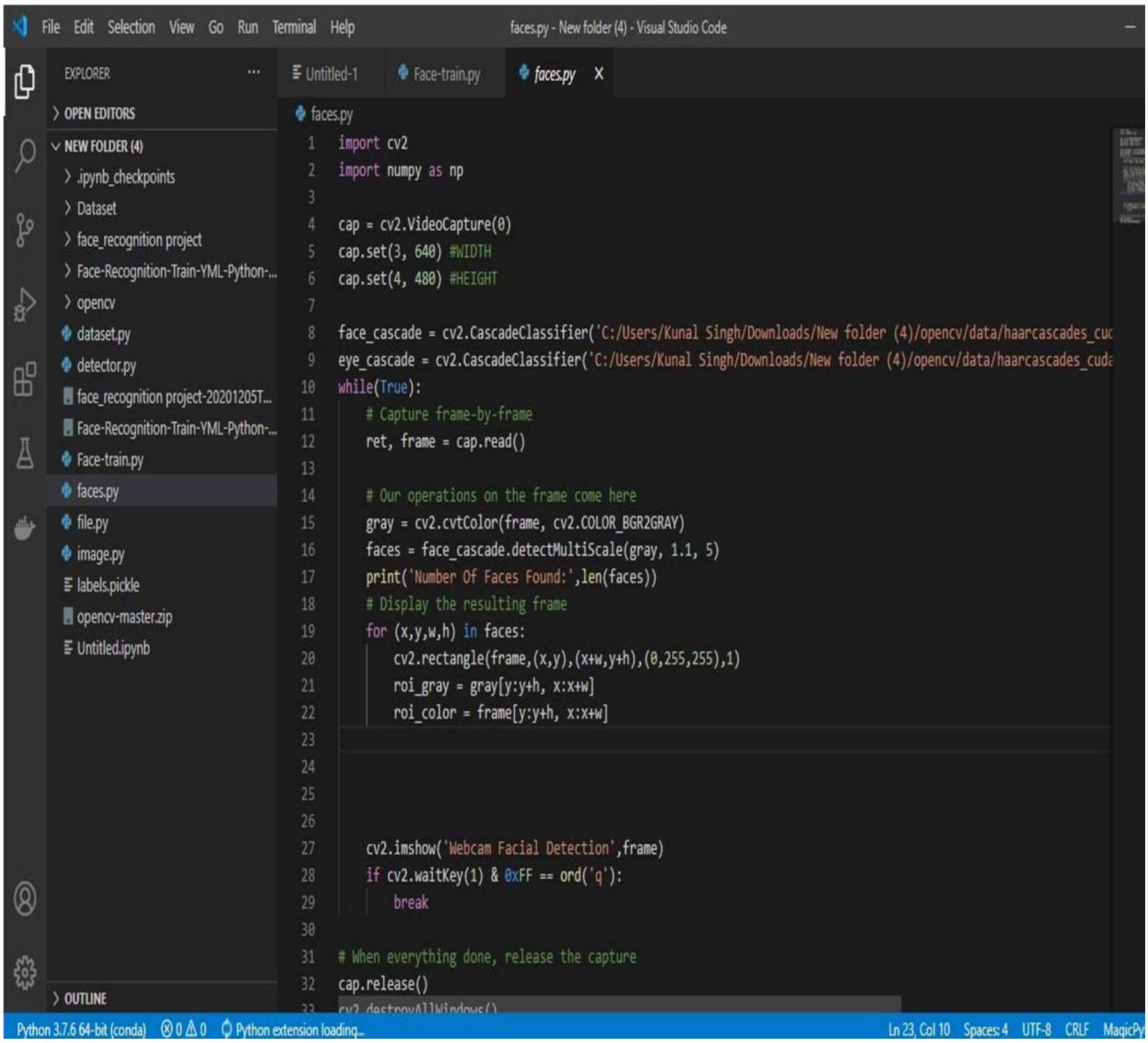


Figure 3.4



```python
import cv2
import numpy as np

cap = cv2.VideoCapture(0)
cap.set(3, 640) #WIDTH
cap.set(4, 480) #HEIGHT

face_cascade = cv2.CascadeClassifier('C:/Users/Kunal Singh/Downloads/New folder (4)/opencv/data/haarcascades_cuc
eye_cascade = cv2.CascadeClassifier('C:/Users/Kunal Singh/Downloads/New folder (4)/opencv/data/haarcascades_cuda
while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()

    # Our operations on the frame come here
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.1, 5)
    print('Number Of Faces Found:',len(faces))
    # Display the resulting frame
    for (x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,255),1)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]




    cv2.imshow('Webcam Facial Detection',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

Figure 3.5:Faces.py file

CHAPTER – 4
TRAINING AND TESTING

4.1 TRAINING IN OPENCV

In OpenCV, training refers to providing a recognizer algorithm with training data to learn from. The trainer uses the same algorithm (LBPH) to convert the images cells to histograms and then computes the value of cell which are included and concatenate on histogram, vectors feature may obtaine. Images is classifie by the process of the profile attached. Input images is being classify by processes and compares data which are stored in database and distances are obtaine by the system . By setting up a threshold, it can be identified if it is a known or unknown face.Eigenface and Fisherface compute the dominant features of the whole training set while LBPH analyses them individually.

To do so, firstly, a Dataset is created. You can either create your own dataset or start with one of the available face databases.

•Yale Face Database

•AT & T Face Database

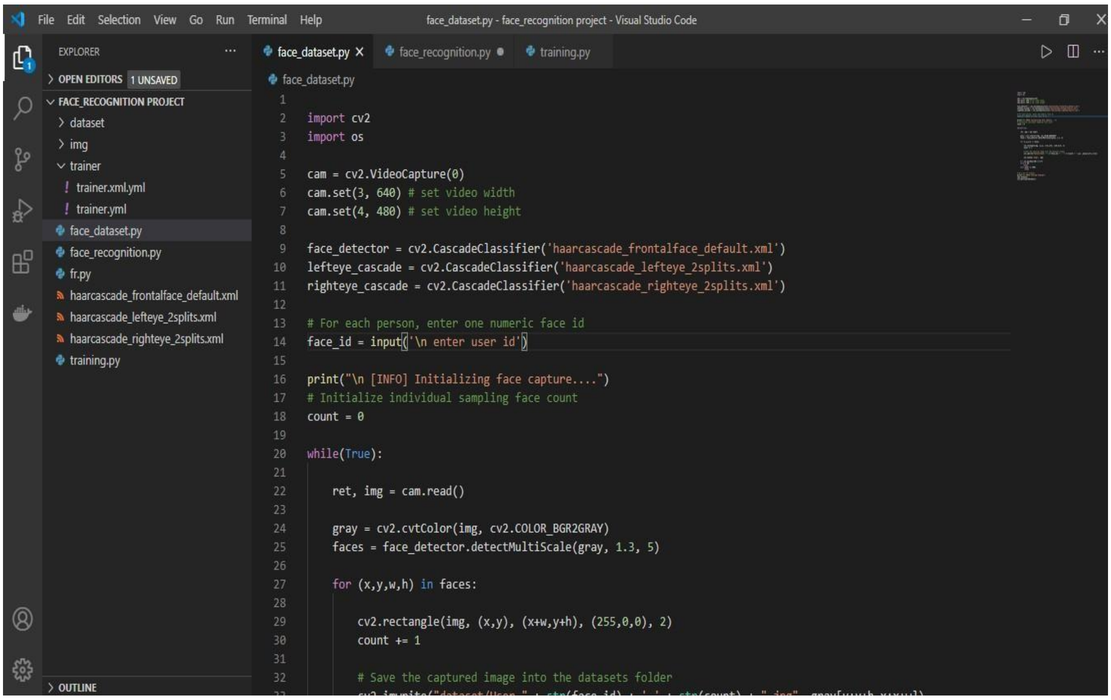
The .xml or .yml configuration file is made from the several features extracted from your dataset with the help of the FaceRecognizer Class and stored in the form of feature vectors.

4.4 CODE

Given below is the code for creating a .yml file, that is the configuration model that stores features extracted from datasets using the FaceRecognizer Class. It is stored in a folder named ‘recognizer’ under the name ‘training Data.yml’.

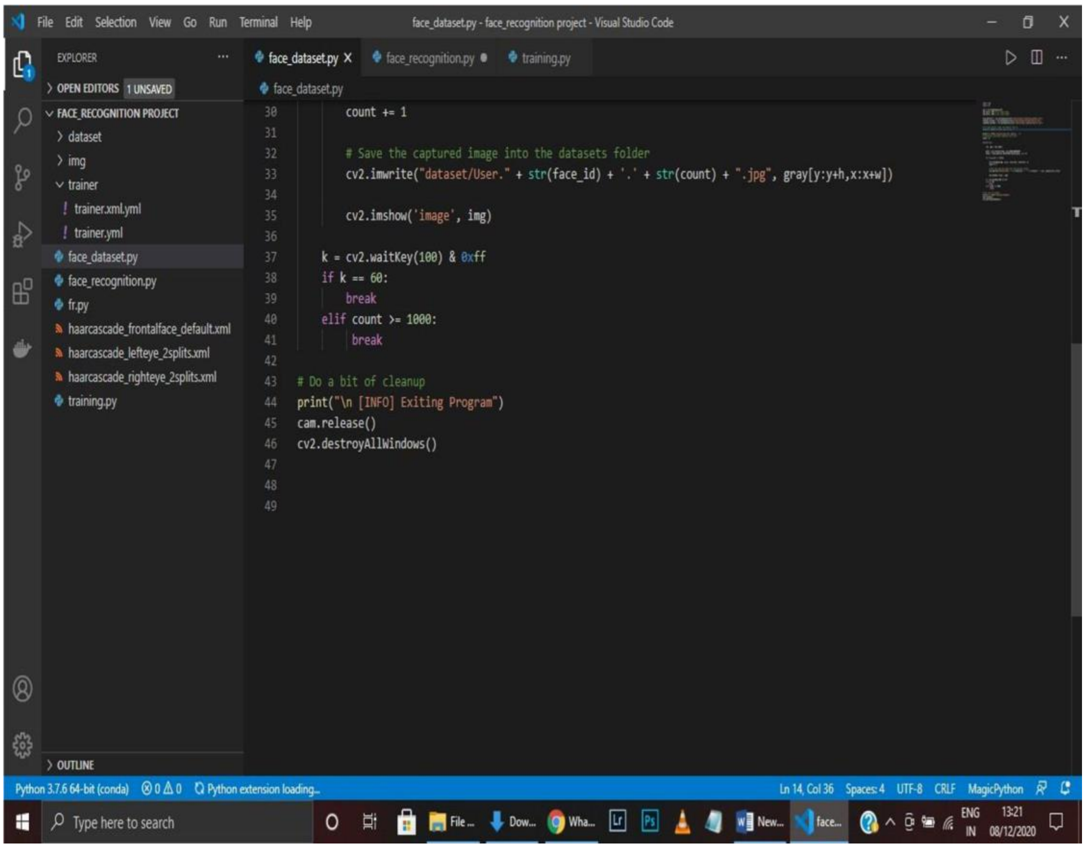
DATASET:

This is the code that will be used to create a dataset. It will turn the camera and take number of pictures for few seconds.Given below is the code for face_dataset.py


File Edit Selection View Go Run Terminal Help
face_dataset.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py

```python

import cv2
import os

cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video width
cam.set(4, 480) # set video height

face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
lefteye_cascade = cv2.CascadeClassifier('haarcascade_lefteye_2splits.xml')
righteye_cascade = cv2.CascadeClassifier('haarcascade_righteye_2splits.xml')

# For each person, enter one numeric face id
face_id = input('\n enter user id')

print("\n [INFO] Initializing face capture....")
# Initialize individual sampling face count
count = 0

while(True):

    ret, img = cam.read()

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_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)
        count += 1

        # Save the captured image into the datasets folder
```

Figure 4.1: Code snippet for the dataset

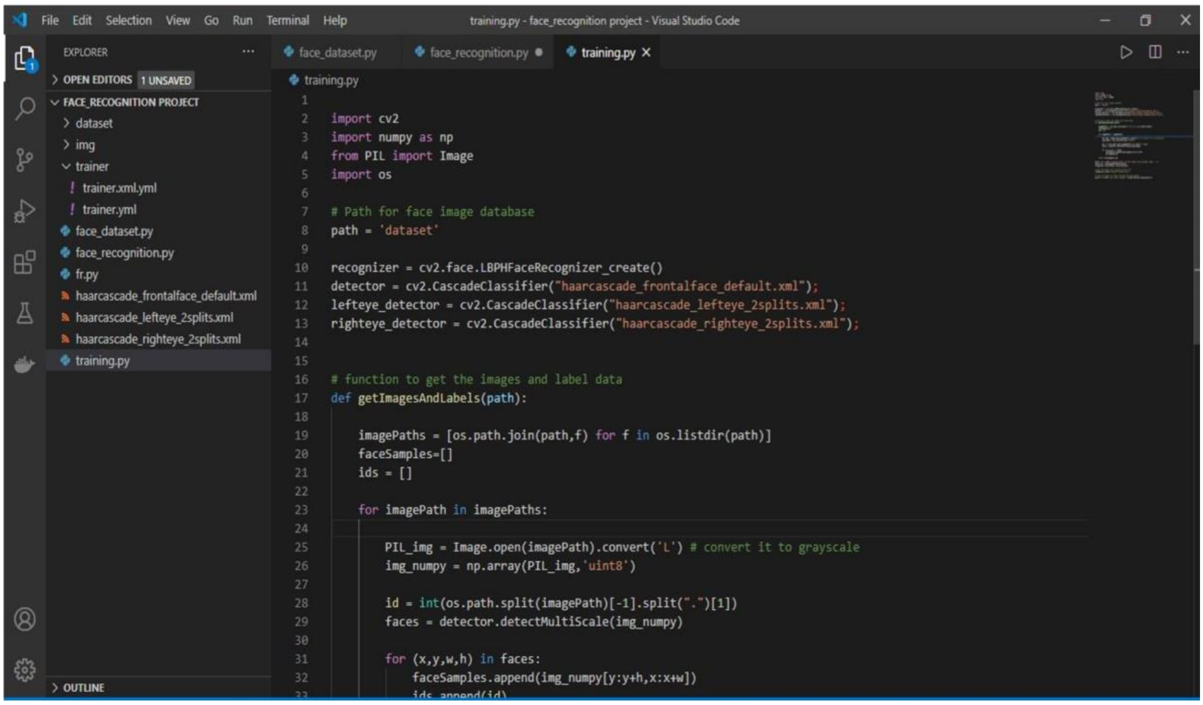

File Edit Selection View Go Run Terminal Help
face_dataset.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py
Python 3.7.6 64-bit (conda)
Python extension loading...
Ln 14, Col 36 Spaces: 4 UTF-8 CRLF MagicPython
Type here to search
ENG IN 13:21 08/12/2020

```
        count += 1

        # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])

        cv2.imshow('image', img)

    k = cv2.waitKey(100) & 0xff
    if k == 60:
        break
    elif count >= 1000:
        break

# Do a bit of cleanup
print("\n [INFO] Exiting Program")
cam.release()
cv2.destroyAllWindows()
```

Figure 4.2: Face_dataset.py

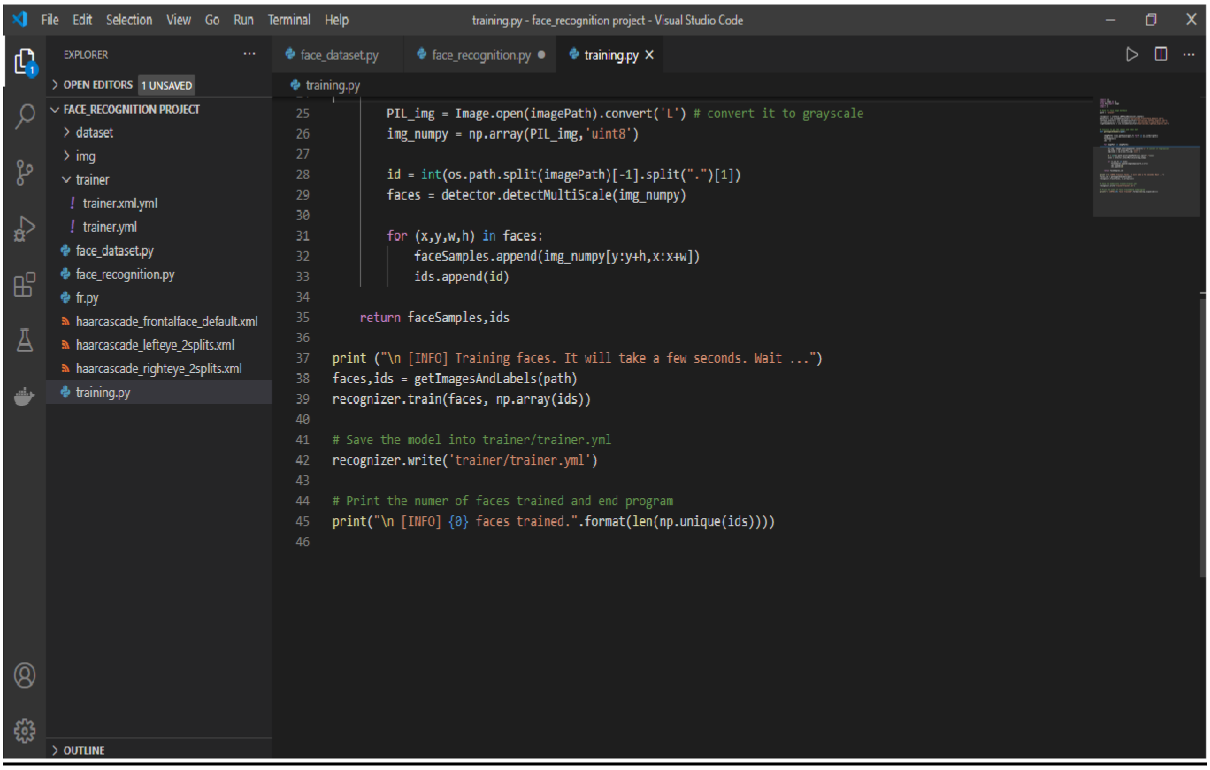
TRAINING:

This is the code that is going to be used to train and get the train.yml file


File Edit Selection View Go Run Terminal Help
training.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py

```python

import cv2
import numpy as np
from PIL import Image
import os

# Path for face image database
path = 'dataset'

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
lefteye_detector = cv2.CascadeClassifier("haarcascade_lefteye_2splits.xml");
righteye_detector = cv2.CascadeClassifier("haarcascade_righteye_2splits.xml");


# function to get the images and label data
def getImagesAndLabels(path):

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

    for imagePath in imagePaths:

        PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
        img_numpy = np.array(PIL_img,'uint8')

        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)

        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)
```

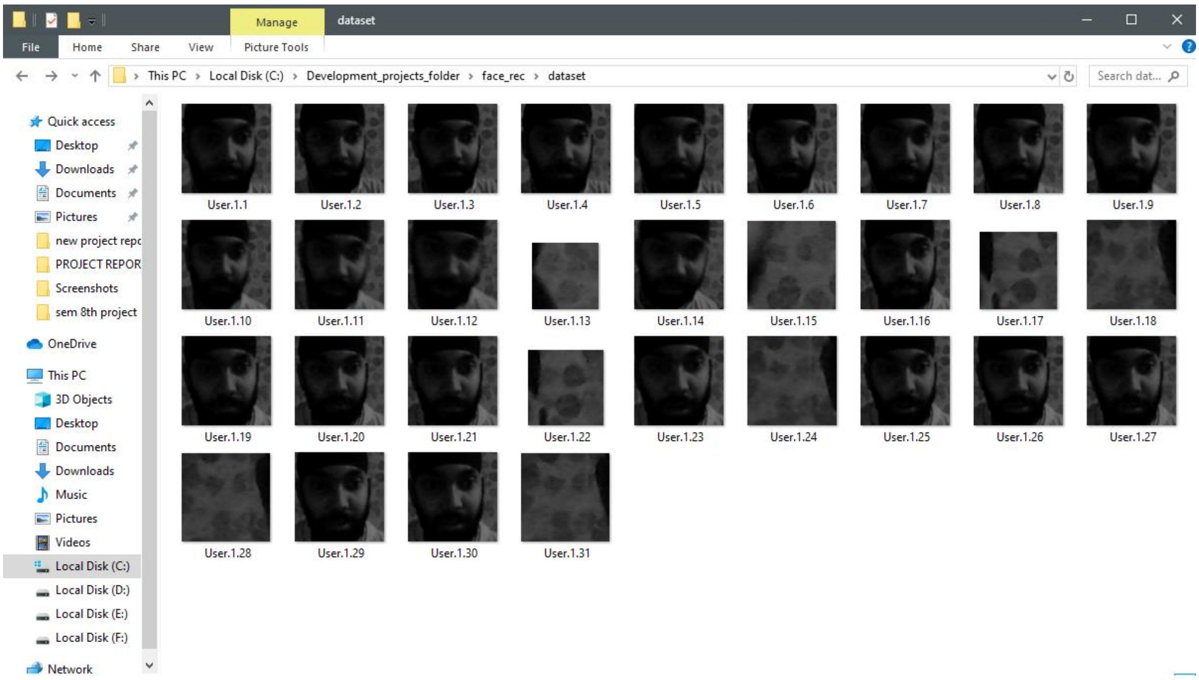
Figure 4.3: Training the dataset


File Edit Selection View Go Run Terminal Help
training.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py

```
        PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
        img_numpy = np.array(PIL_img,'uint8')

        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)

        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)

    return faceSamples,ids

print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))

# Save the model into trainer/trainer.yml
recognizer.write('trainer/trainer.yml')

# Print the numer of faces trained and end program
print("\n [INFO] {0} faces trained.".format(len(np.unique(ids))))
```

Figure 4.4

OUTPUT : After running dataset code we will get numberof pictures in the folder named dataset. Now therse photos will be used to train. The more the pics the greater the accuracy.



4.5 LBPH RECOGNIZER

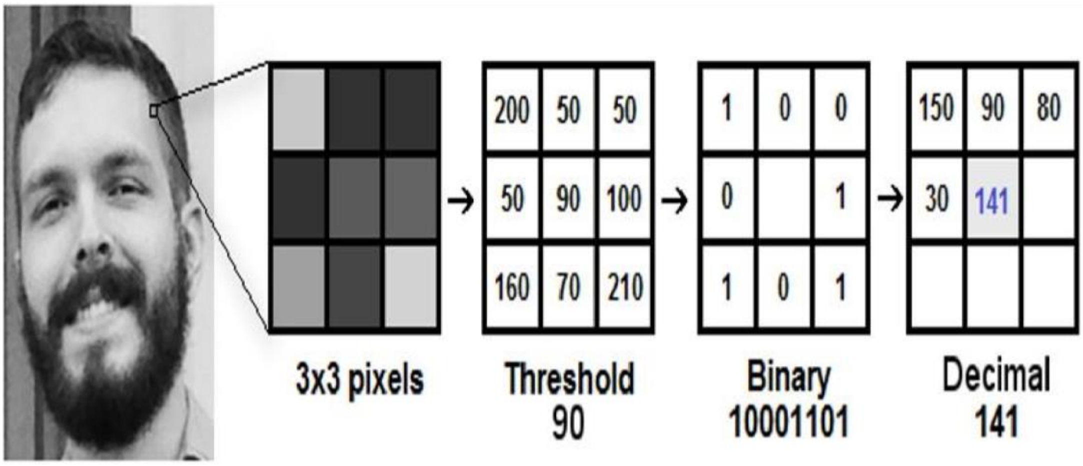
The approach that has been used in this project is, LBPH approach which uses the following algorithm to compute the feature vectors of the provided images in the dataset.

In this section, it is shown a step-by-step explanation of the LBPH algorithm:

1.First of all, we need to define the parameters (radius, neighbours, grid x and grid y) using the Parametersstructure from the lbph package. Then we need to call the Init function passing the structure with the parameters.

2. Secondly, we need to train the algorithm. To do that we just need to call the Train function passing a slice of images and a slice of labels by parameter. All images must have the same size.

3. The Train function will first check if all images have the same size. If at least one image has not the same size, the Train function will return an error and the algorithm will not be trained.



200	50	50
50	90	100
160	70	210

1	0	0
0		1
1	0	1

150	90	80
30	141	

Figure 4.6: referencing and assigning pixel value

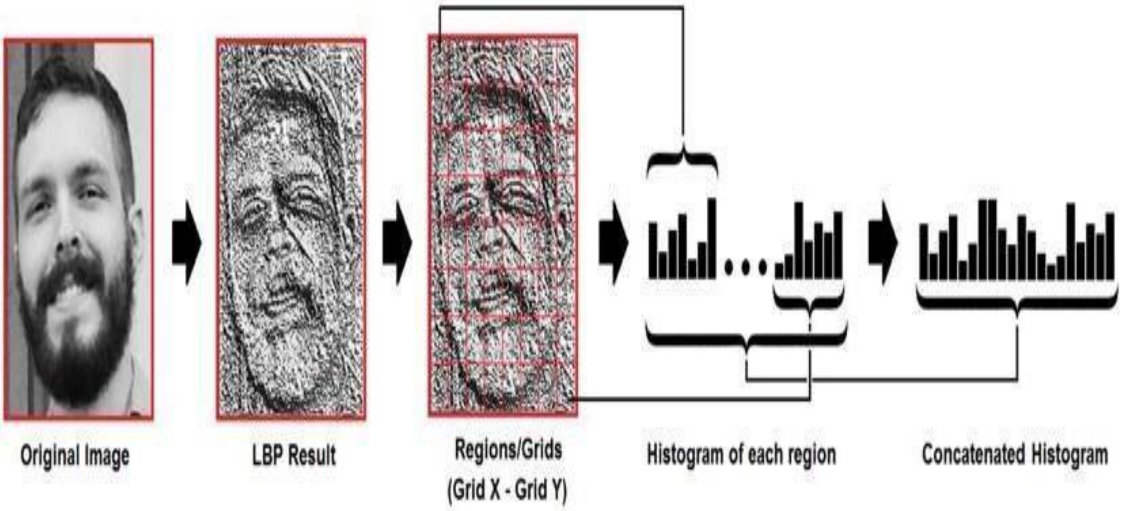


Figure 4.7: LBP result

The images, labels, and histograms are stored in a data structure so we can compare all of it to a new image in the Predict function.

The LBPH package provides the following metrics to compare the histograms: **Chi-Square** :

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

Equations 1 **Euclidan Distances** :

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

Equation 2

Normalize Euclidan Distances :

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

Equation 3

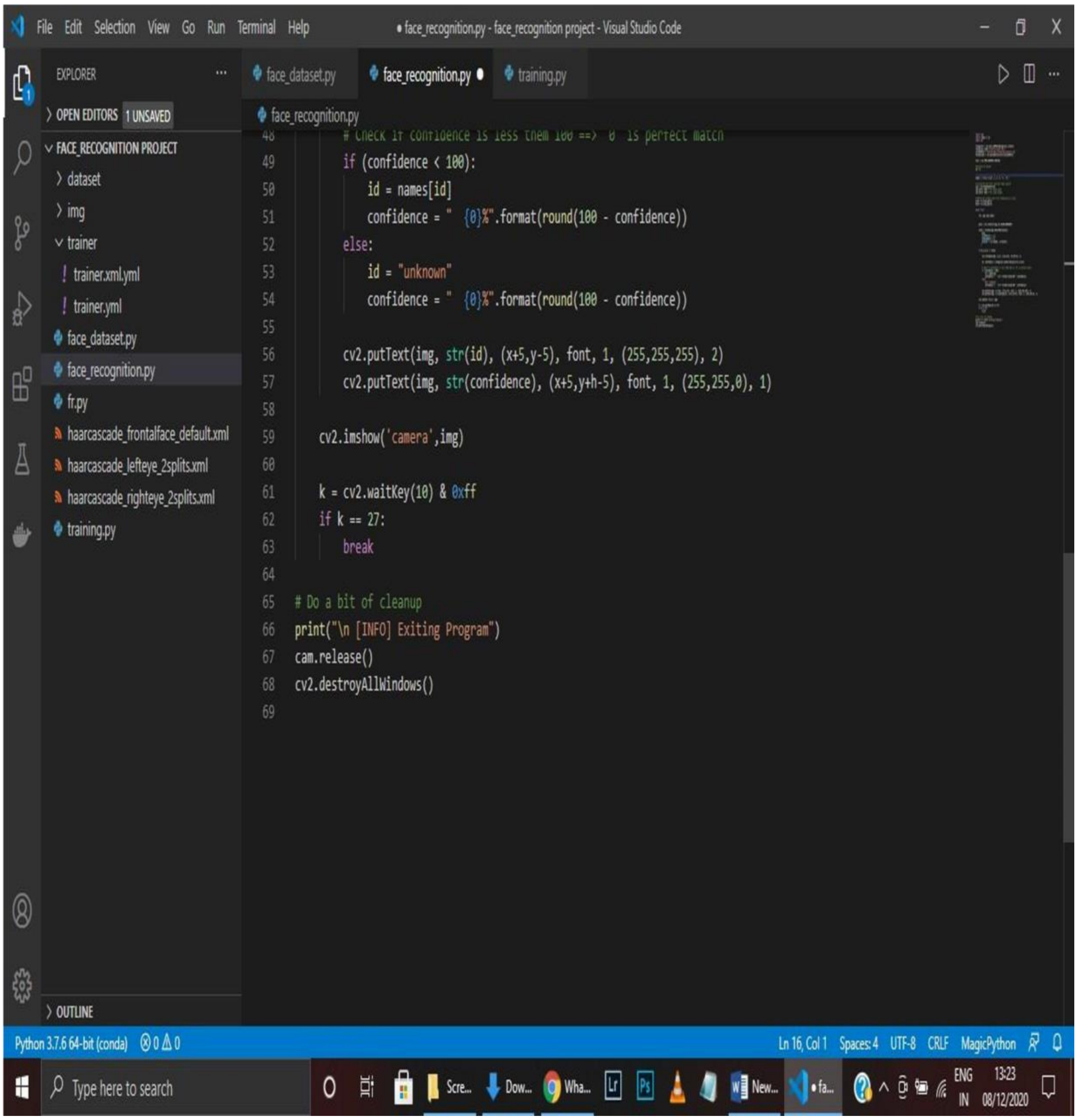
Absolute Values :

$$D = \sum_{i=1}^{n} |hist1_i - hist2_i|$$

Equation 4

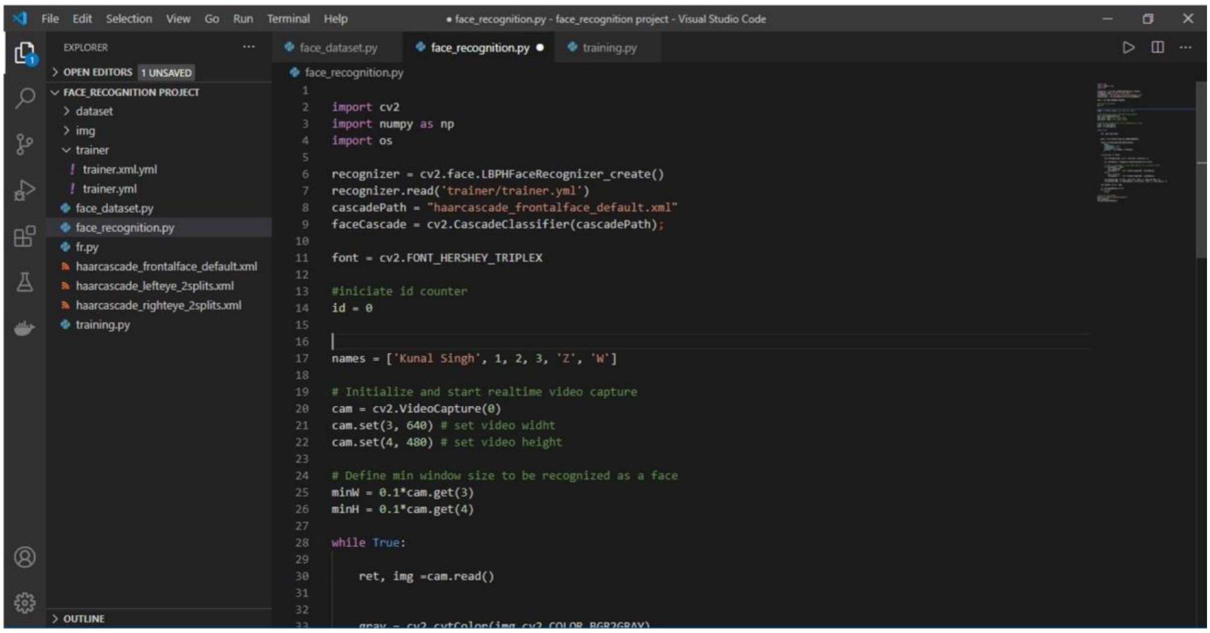
4.7 Code

Given below is the face recognition script that reads the data from the trainingData.yml file mentioned before and uses the .predict() function to pass a confidence value, recognize the face of the individual and display their names along with their face. The following script uses data that has been trained with images of the students working on this project


File Edit Selection View Go Run Terminal Help
• face_recognition.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
face_dataset.py
face_recognition.py
training.py
OUTLINE
Python 3.7.6 64-bit (conda) ⊗ 0 ⚠ 0
Ln 16, Col 1 Spaces: 4 UTF-8 CRLF MagicPython
Type here to search
ENG IN 13:23 08/12/2020

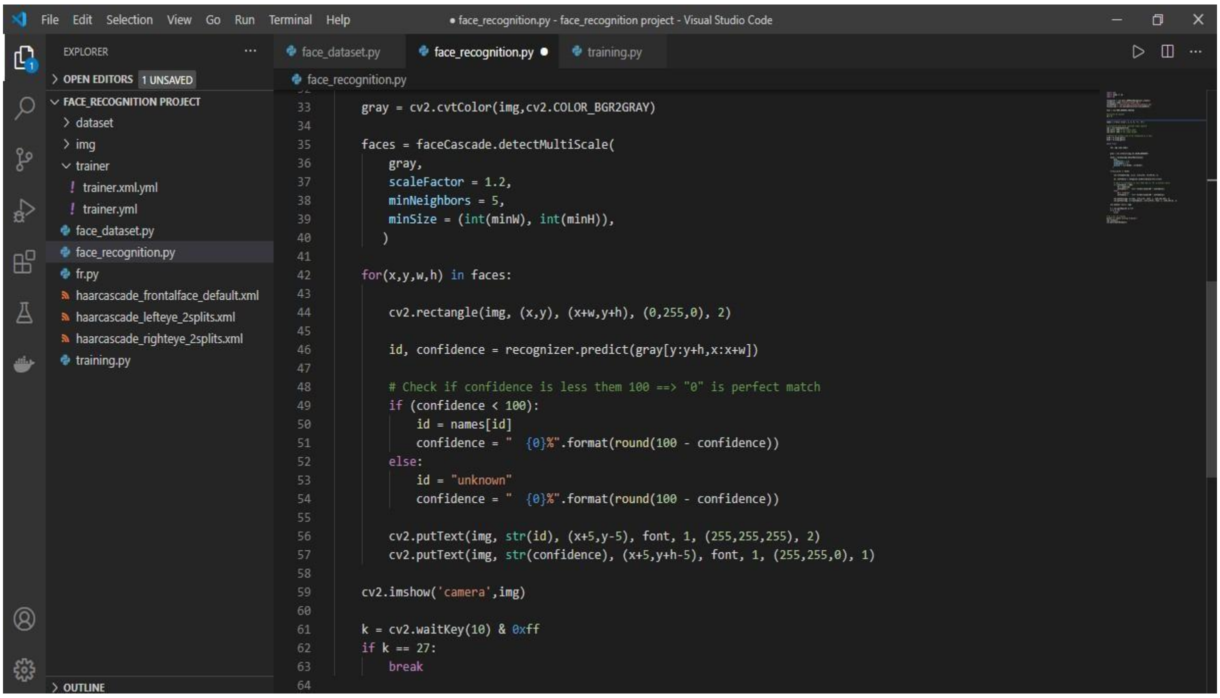
```
        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 100):
            id = names[id]
            confidence = "  {0}%".format(round(100 - confidence))
        else:
            id = "unknown"
            confidence = "  {0}%".format(round(100 - confidence))

        cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

    cv2.imshow('camera',img)

    k = cv2.waitKey(10) & 0xff
    if k == 27:
        break

# Do a bit of cleanup
print("\n [INFO] Exiting Program")
cam.release()
cv2.destroyAllWindows()
```

Figure 4.8: Functioning


File Edit Selection View Go Run Terminal Help
• face_recognition.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py

```python

import cv2
import numpy as np
import os

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT_HERSHEY_TRIPLEX

#iniciate id counter
id = 0


names = ['Kunal Singh', 1, 2, 3, 'Z', 'W']

# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video widht
cam.set(4, 480) # set video height

# Define min window size to be recognized as a face
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)

while True:

    ret, img =cam.read()


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

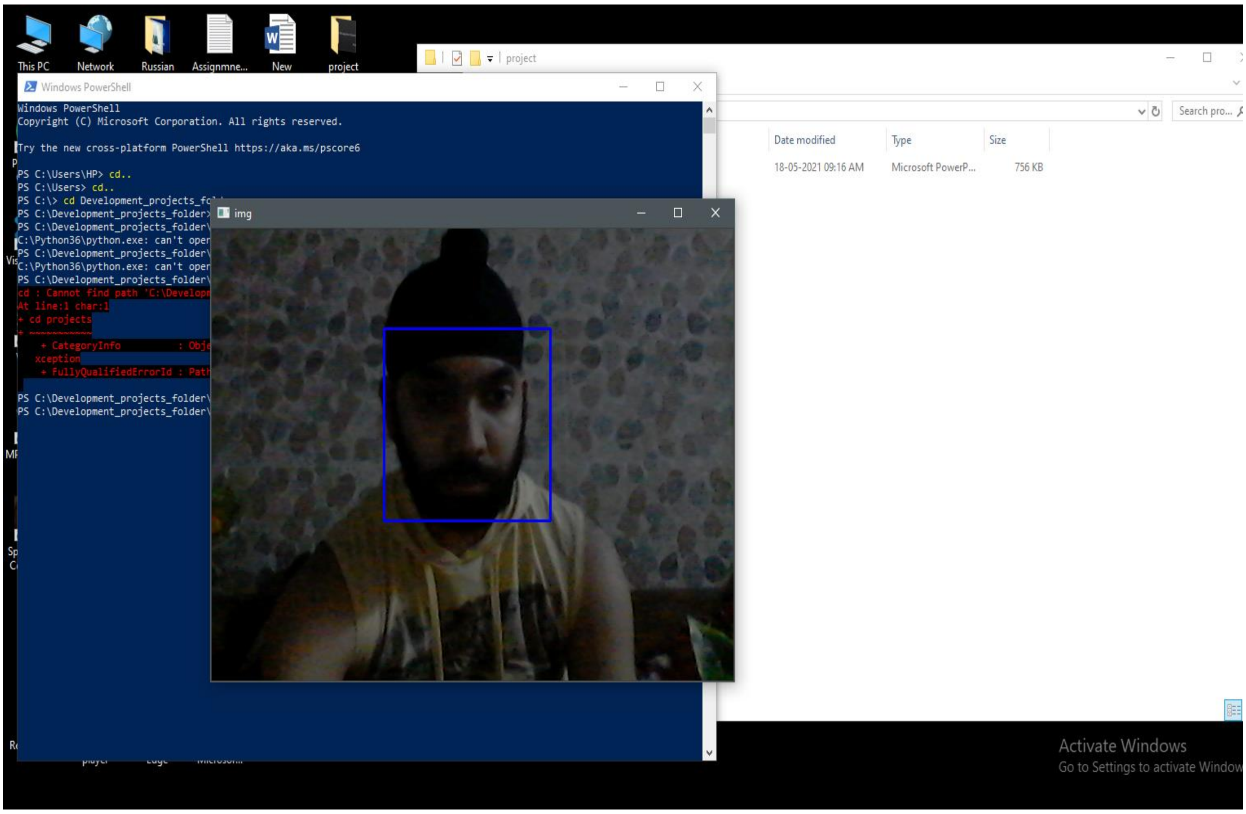
Figure 4.9: Other snippets


File Edit Selection View Go Run Terminal Help
• face_recognition.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py

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

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
       )

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

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

        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 100):
            id = names[id]
            confidence = "  {0}%".format(round(100 - confidence))
        else:
            id = "unknown"
            confidence = "  {0}%".format(round(100 - confidence))

        cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

    cv2.imshow('camera',img)

    k = cv2.waitKey(10) & 0xff
    if k == 27:
        break

```

Figure 4.10: Face_recognition.py

OUTPUT



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\HP> cd..
PS C:\Users> cd..
PS C:\> cd Development_projects_fo
PS C:\Development_projects_folder>
PS C:\Development_projects_folder\
C:\Python36\python.exe: can't oper
PS C:\Development_projects_folder\
C:\Python36\python.exe: can't oper
PS C:\Development_projects_folder\
cd : Cannot find path 'C:\Developm
At line:1 char:1
+ cd projects
+ ~~~~~~~~~~~
    + CategoryInfo          : Obje
   xception
    + FullyQualifiedErrorId : Path

PS C:\Development_projects_folder\
PS C:\Development_projects_folder\
```

4.9 APPLICATIONS

- Security: Face Recognition can help in developing security measures, that is unlocking of a safe using facial recognition.

- Attendance Systems: Face Recognition can be used to train a set of users in order to create and implement an automatic attendance system that recognizes the face of the individual and marks their attendance.

- Access: Face Detection can be used to access sensitive information like your bank account and it can also be used to authorize payments.

- Mobile Unlocking: This feature has taken the mobile phone industry by a storm and almost every smart phone manufacturing company has their flagship smartphones being unlocked using face recognition. Apple's FaceID is an excellent example.

- Law Enforcement: This is a rather interesting way of using face detection and face recognition as it can be used to assess the features of a suspect to see if they are being truthful in their statements or not.

- Healthcare: Face Recognition and Detection can be used in the healthcare sector to assess the illness of a patient by reading their facial features.

CHAPTER 5:CONCLUSION AND FUTURE SCOPE

5.1 Future Scope

- Government/ Identity Management: Governments all around the world are using face recognition systems to identify civilians. America has one of the largest face databases in the world, containing data of about 117 million people.

- Emotion & Sentiment Analysis: Face Detection and Recognition have brought us closer to the technology of automated psyche evaluation. As systems now a days can judge the precise emotions frame by frame in order to evaluate the psyche.

- Authentication systems: Various devices like mobile phones or even ATMs work using facial recognition, thus making getting access or verification quicker and hassle free.
- Full Automation: This technology helps us become fully automated as there is very little to zero amount of effort required for verification using facial recognition.

- High Accuracy: Face Detection and Recognition systems these days have developed very high accuracy and can be trained using very small data sets and the false acceptance rates have dropped down significantly.

5.2 Limitations

- Data Storage: Extensive data storage is required for creating, training and maintaining big face databases which is not always feasible.

- Computational Power: The requirement of computational power also increases with increase in the size of the database. This becomes financially out of bounds for smaller organizations.
- Camera Angle: The relative angle of the target’s face with the camera impacts the recognition rate drastically. These conditions may not always be suitable, therefore creating a major drawback.

5.3 Conclusion

Facial Detection and Recognition systems are gaining a lot of popularity these days. Most of the flagship smartphones of major mobile phone manufacturing companies use face recognition as the means to provide access to the user.

This project report explains the implementation of face detection and face recognition using OpenCV with Python and also lays out the basic information that is needed to develop a face detection and face recognition software.The goal of increasing the accuracy of this project will always remain constant and new configurations and different algorithms will be tested to obtain better results. In this project, the approach we used was that of Local Binary Pattern Histograms that are a part of the FaceRecognizer Class of OpenCV.

References

[1] Schneiderman.United States of America Patent U.S. Patent No. 8,457,367, 2013.

[2] R. J. Baron, “Mechanisms of human facial recognition,” *International Journal of Man- Machine Studies.*

[3] M. Nixon, “"Eye Spacing Measurement for Facial Recognition",” *International Society for Optics and Photonics.,* vol. (Vol. 575), (19 December 1985).

[4] H. &. Y. J. Yu, “A direct LDA algorithm for high-dimensional data—with application to face recognition,” 2001.

final_report-converted.pdf

by

Submission date: 18-May-2021 10:21PM (UTC+0530)
Submission ID: 1588910628
File name: final_report-converted.pdf (1.67M)
Word count: 5146
Character count: 26960

Chapter 1: Introdution

1.1 Introduction

A face recognition system could also be a technology which match a given install profile a personality's face from images and videos which it has or use it as a reference to map and identify against an info of faces. Researchers area unit presently developing multiple ways throughout that face recognition systems work. the foremost advanced face recognition methodology, that is to boot used to manifest users through ID verification services, works by pin pointing and mensuration countenance in image

While at first kind of laptop application, recognition system seen wider uses in recents times on smartphones and in alternative kinds of technol ogy, like artificial intelligence. as a result of computerised face recognition involves themeasuring of a human's physiological characteristics face recognition system area unit classified as bioscience. though the accuracy of face recognitions [20] system as a biometric technology is a smaller amount than iris recognition and handprint recognition, it wide adopte because of its contactless and non-invasive method.[17] Facial recognition systems area unit deployed in advanced human -computer interaction, video police work and automatic compartmentalisation of pictures.

We have a created a face recognition technology capable of identifying faces.

1.2 PROBLEM STATEMENT

“Facial Detection and Facial Recognition using Intel’s open source Computer Vision Library (OpenCV) and Python dependency”

There are various scripts illustrated throughout the project that will have functionalities like detecting faces in static images, detecting faces in live feed using a webcam, capturing face images and storing them in the dataset, training of classifier for recognition and finally recognition of the trained faces.

All the scripts are written in python 3.6.5 and have been provided with documented code. This project lays out most of the useful tools and information for face detect ion and face recognition and can be of importance to people exploring facial recognition with OpenCV.

The project shows implementation of various algorithms and recognition approaches which will be discussed on later in the project report.

Face Recognition can be of importance in terms of security, organization, marketing, surveillance and robotics etc.

Face detection is able to very immensely improve surveillance efforts which can greatly help in tracking down of people with ill criminal record basically referring to criminals and terrorists who might be a vast threat to the security of the nation and the people collectively.The Personal security is also greatly exacerbated since there is nothing for hackers to steal or change, such as passwords.

1.3 OBJECTIVES

This project is created so as to study the various means of recognizing faces with more accuracy and reducing the error rates while recognition. The ideal condition for any recognition project is to reduce the intra class variance of features and increase the inter class variance of features to be detected or recognized.

Facial Recognition software is “Capable of uniquely identifying or verifying a person by comparing and analyzing patterns based on the person’s facial contours. It is mostly used for security purposes”.Many other areas of use.

Different recognizer approaches are used for recognition of faces. They are:

- Eigen Faces
- Fisher Faces
- Local Binary Pattern Histograms

1.4 METHODOLOGY

The Project utilizes various libraries of Python such as

1.4.1 OpenCV

“OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library”. The main purpose of this was common infrastructure for computer vision application and it was built specifically for such purposes not to mention it also accelerated the machine perciption inside the business product. “Being a BSD-licensed product, OpenCV makes it straightforward for businesses to utilize and modify the code”.

In total we can say that The library has about 2000 optimiz algorithm which is really insane, “These algorihms contain a comprehensive set which comprises of each classic and progressive laptop vision and machine learning algorithms. These algorithms area unit usually accustomed sight and acknowledge faces, determine objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, manufacture 3D purpose clouds from s tereo cameras, sew pictures along to produce a high resolution image of a full scene, realize similar pictures from an image info, take away red eyes from pictures taken exploitation flash, follow eye movements, acknowledge scenery and establish markers to overlay it with increased reality, etc”. The amazing thing about this library is that it has quite about forty seven thousand individuals of user community and calculable variety of downloads olympian eighteen million. The library is utilized extensively in corporations, analysis teams and by governmental bodies.

Along with well-established corporations like “Google, Yahoo, Microsoft, Intel, IBM,

Sony, Honda, Toyota” that use the library, there area unit several startups like “Applied Minds, VideoSurf, and Zeitera”, that create in depth use of OpenCV. OpenCV’s deployed wide array spans vary from sewing streetview pictures along, police work intrusions in police work videoin Israel, watching mine instrumentality in China, serving to robot navigated and devour 5 objects at “Willow Garage, detection of natatorium drowning accidents in Europe, running interactive art in Espana and New York , checking runways for scrap in Turkey.

1.4.2 NumPy

“The Python programming language earlier wasn't originally designed for numerical computing as we know it to be , however it also attracted the attention of the scientific and engineering community early” . “In 1995 the interest (SIG) matrix-sig was based with the aim of shaping associate array computing package; among its members was Python designer and supporter Guido van Rossum, WHO extended Python's syntax (in explicit the compartmentalization syntax) to make array computing easier”.

6 “implementation of matrix packages was complete by Jim discoverer, then generalized further rationalization required by Jim Hugunin and known as Numeric (also diversely observed because the "Numerical Python extensions" or "NumPy").Hugunin, a collegian at the Masachusetts Institut of Technology (MIT), joine the Corporatio for National analysis Initiativ (CNRI) in 1997 to work on J Python, leaving Paul Dubois of Lawrence Livermore National Laboratory (LLNL) to need over as supporter. Others early contributor embrace David dAscher, Konred Hinsan and Traves Oliphent”.

“A new package known as Numarray was written as a additional versatile replacement for Numeric.Like Numeric, it too is currently deprecated.Numarray had quicker operations for large arrays, however was slower than Numeric on tiny ones, thus for a time each packages were utilised in parallel for varied use cases. The last version of Numeric (v24.2) was discharged on St Martin's Day 2005, whereas the last version of numarray (v1.5.2) was discharged on twenty four August 2006”.

There wa want urge Numerics into the Pythons customary library, however Guido van Rosum determined that code wasn't reparable inits state.

“In early 2005, NumPy developer Travis Oliphant needed to unify the community around one array package and ported Numarray's options to Numeric, cathartic the result as NumPy one.0 in 2006.This new project was a region of SciPy. To avoid putting in the large SciPy package simply to urge associate array object, t his new package was separated and known as NumPy. Support for Python three was other in

2011 with NumPy version one.5.0”.

pypy start developments on associate implementations of the numPy Api. It is not nevertheless absolutely compatable with numPys.

1.4.3 Pillow

The Python Imaging Library (PILLOW) or generally called as PIL is very useful for adding image processing capabilitiy to your Python interpreter which you have or have installed when working with Pyhon.

The main use of this library is that this library has a wide array of extensive file format support which allows it to run and store different files in different formats, an efficient representation which is further boosted by very powerful image process ing ability.The important point is that the image in librarie is meant for faster accessing to that date is stored during a few basic pixels formats.All this enable it to be a very commanding and powerful tool for image processing general, which is probably also the reason why it is used so heavily.

Let’s see a couple of possible things in which we can use the [14] library.

Image Archive-The “Python Imaging Library” is rightin image archivalal and execution applications. you'll uses the libraryto make thumbnail, [14] convert between file formats printf images etc.

From the information available to us currently about the current version is that it [11] identifies and reads an outsized number of file formats. Write support is intentionally restricted to the foremost commonly used interchange and presentation formats.

“Image Display-The current release includes Tk PhotoImage and BitmapImage interfaces, also as a Windows DIB interface which will be used with PythonWin and other Windows-based toolkits. Many other GUI toolkits accompany some quite PIL support”.“For a purpose such as debugging there are a lot of functions like the [10] show() method which saves a picture to disk, and callsit an external display utility.

Image ProcessingThe library contain basics image process functionalities, include point operation, filteres a group of built-in convolution kernel, and colours spacees conversion .Basically in short “Python Imaging library” is a free additional open source library which you have to install first by using the command such as pip so that you can use it to run various python functions in the module.It is important to note and keep in mind that all of these fucntions only work once you import this library and cannot be used if the library isnt imported or called upon.Henceforth why is it recommended to install this library beforehand inorder to properly use it.In regards to the main function of this library is to add functionality in opening, manipulating and storing of different file format images which can later be used to perform operations on accordingly to the needs of the programmer using the library.

1.4.4 FACE RECOGNITION

The “Face recognition” library in python is a library which helps in recognizing and manipulating the faces by using the programming language pythons or froms the comd line the simplest face recognitions libraries after importing the module and accessing the required functions.The “Face recognition” library was built using dlib’s “state-of-the-art face recognition” and was further enhanced and builts deep learning. The model have accurace of 99.37%.It is used to find faces in pictures

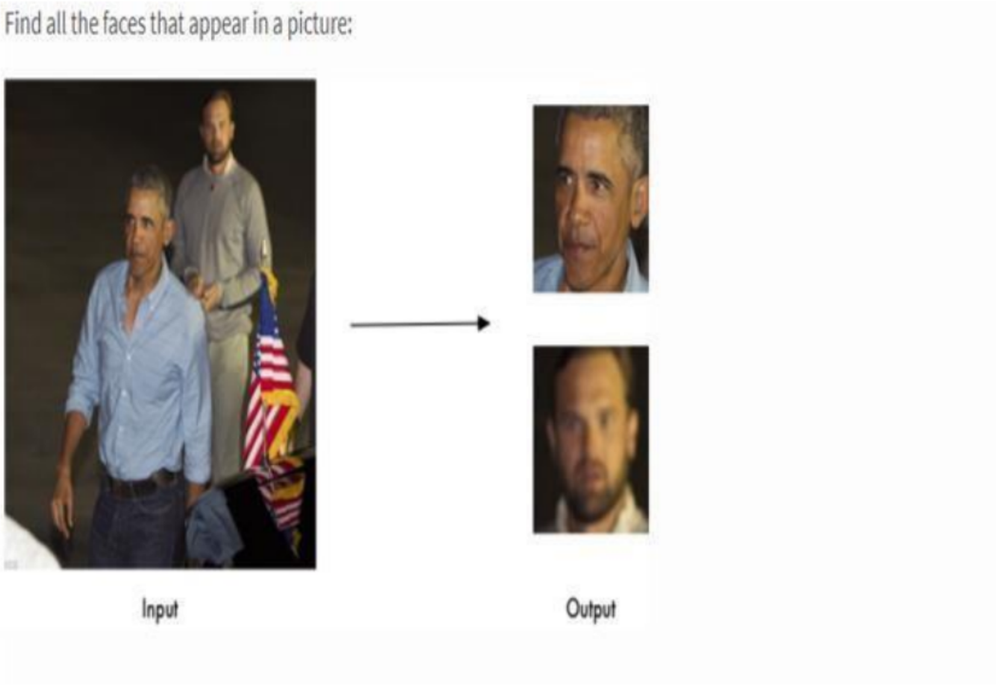


Figure 1.1: Finding all the faces in the picture

Find and according look for   facial features in the pictures

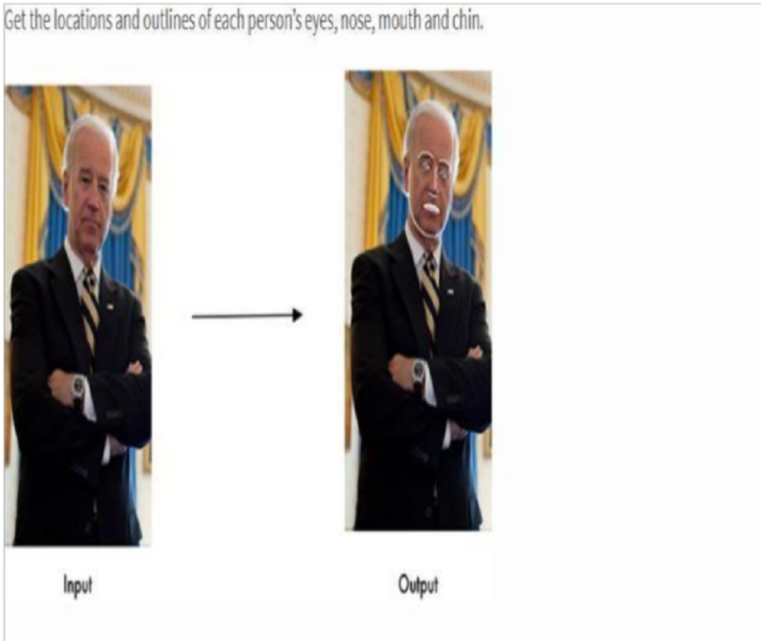
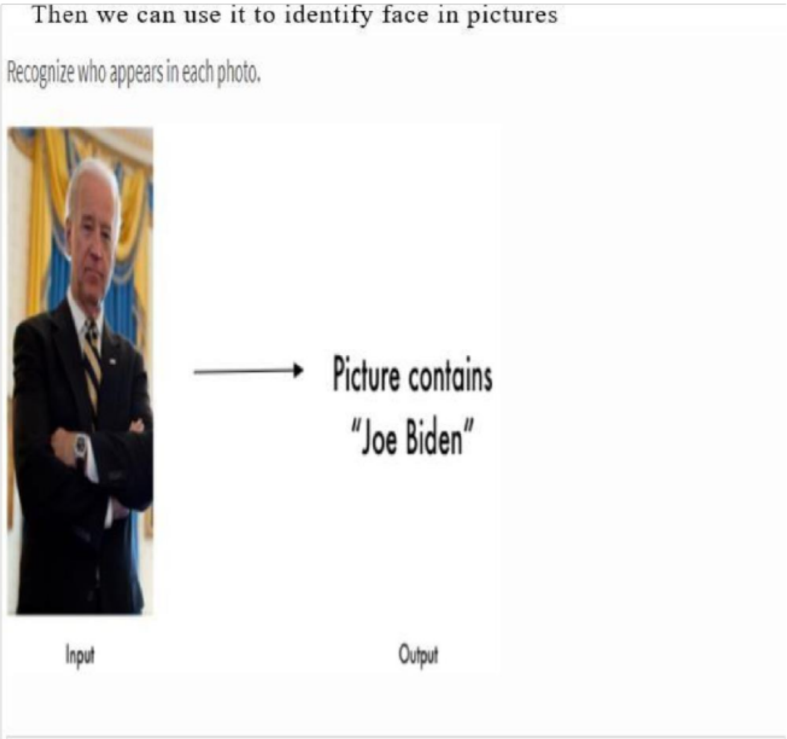

Get the locations and outlines of each person's eyes, nose, mouth and chin.
Input
Output

Figure 1.2:Getting the locations



Then we can use it to identify face in pictures

Recognize who appears in each photo.

Figure 1.3: Recognizing who is in the Picture

1.4 Organization

Biometrics in simple terms basically refers to evaluating and measuring your Fingerprints,face features or any such human parameters that help to identify an individual.The study of Biometrics is very much important considering how bleak the security as a whole has become.Note that identification is only possible since each and every person has unique and distinct features which make it easier to identify individuals.

Keep in mind that the biometric feature which is use or are used must available in databases for all of individual in communities .before features or those features is use for other authentications . This is called enrolment.

Authentications is in one of these following forms

- Identificationss: This basically refers to matching an individual featured in againsts the record to verify whether his or her or simply put that if or not the records arethere in the database or not

- Verifications: “To check whether the person is who he/she is claiming to be”.In other words to check whether their claim of identity is true or not.[4]In this case the other features of another person is matched only with the pre installed features of the person, they claim themselves to be .

Different types of Biometric are present:

There are another two present broad categories of biometrics:

1. Physiological Biometric
2. Behavioural BiometricPhysiological Biometric:

“As the name sounds out to be in this the physical traits of a person are measured greatly for identification and verification in this type of biometrics. The trait should be chosen such that it is unique among the general population, and no mater what is resistant to changes due to illness, aging, injury, etc”’.

Physiologicals Biometric Technique:

- handprint:handprints are one of the main methods to uniquely identify an individual.They are also regraded as one of the best methods for this sole purpose and many systems utilize this sort of recognition method. They have the uniques for every individual person and can be measured in variosus different way . Minutia-[4]based measurement uses graphs which are there to match the ridges contained whereas imagebased measurments in finds similarities or the similar patterns inbetween the fingertips image and fingerprint images which are present or which have been uploaded in the database for the purpose of matching and uniquely identifying the identity of the person concerned. The security level is very high and use both in identifications and verifications. However, because to old age and different diseases/injury, handprints might gets alter.

in smartphons for verifications, in offices for other identifications.

- FacialRecognition: This basically refers to the set of the featured of face suppose distance between nose and the mouths or say the distance between the ear, length of whole face, skin color, are use In verifications and identifications in this regard. However there can be certain complications arising due to the complexity in aging, diesease,wearing sunglasses or any sort of face that disrupts to recognize those featrues and indirectly hampering the outcome greatly by playing a role in the low accuracy of the results.

- Iris and Retina: Not only just fingerprints the patterns found in say the iris and retina are also unique metrics which can be used to deicde or identify the concerned Device in analyses of retina is expensive and are use less common daily life. Disease cataract may hinder the pattern recognition of the iris and may cause discrepancies to occur

- Voice Recognition: The third kind of recognition that can be used is the voice recognition which is also very helpful.The pitches, voice modification, and tone, instead thing is takan into consideration and are taken in the dataset . Regarding the security of the system,it isnt necessarily a great choice in that regard since two different people can have same voice and henceforth the model will have to deal with a lot of problems and wich is why it isnt all that great and isnt used mostly in recognition. The accurace are hindered due to the present noise, or due to aging and illness.

- DNA: DNA might be the most unique kind of authentication ,since it is the most trusted metric for uniqely identifying an inidividual. Thus, securities are high and may be include in both identifications and verifications

Behavieral Biometric:

In this generally measure, which relate to the behaviour patterns of the individual and which can be a great source of auhentication

- Signature: “Signature might just be one of the most used metric for authentication . They are used to verify checks by matching the signature of the check against the signature present in the database. Signature tablets and special pens are used to compare the signatures”. timing requirement in writing the signature may be use in increase the accuracy. Signatur are mostly used in verifications.

- Keystrok Dynamic: This technique is used to measure the behavior of the person we are storing in database . Some of the characteristics take into account are:

1. Type speed-refers to how fast the person can type and matches that
2. Frequency of errors-what is the frequency of the errors committed
3. Duration of key depressions

CHAPTER- 2

LITERATURE SURVEY

2.1 Face Tracking

Face tracking refers to identifying the features which are then used to detect a Face In this case the example method includes the receiving or we can say that it gets first images and the second images user face who is being taken into consideration, where one or both of the images which were used to sort of look for a match have been granted a match by the facial recognition system which also proofs the correct working of the system. "The technique includes taking out a second sub- image coming from the second image, where the second sub-image includes a representation of the at least one corresponding facial landmark, detecting a facial gesture by determining whether a sufficient difference exists between the second sub - image and first sub-image to indicate the facial gesture, and determining, based on detecting the facial gesture, whether to deny authentication to the user with respect to accessing functionalities controlled by the computing" [1]

2.2 Mechanisms of human facial recognition

Basically what we see in this paper is that it presents an extension and a new way of perceptions of the theory of author for sapiens visual process information, in which method in extracting the second image of image second, where image of second includes a representations at least one corresponding of facial given landmarks. “In turn detecting a facial gesture by determining whether a sufficient difference exists between the second sub-image and first sub-image to indicate the facial gesture, and determining, based on detecting the facial gesture, whether to deny authentication to the user with respect to the human recognition system and same was applied”.Several indispensable techniques is implicat: encoding of visible photographs pattern of nural , detect features of facial, measurement standard, discount patterns of neural in dimensionality.

“The logical (computational) role suggested for the primary visual cortex has several components: size standardization, size reduction, and object extraction”. “[9]The given result inprocessing of primary cortexs, it suggest,visual patternis encoding in suitable size instoraging. “(In this context, object extraction is the isolation of regions in the visual field having the same color, texture, or spatial extent.)”[9]in topology map from cortex to retina, retina connection , [19]primary visual cortex and geniculate bodies, and the cortex structure itselfs [19]encode the visual patterns combined In this theory it is already illustrate the human graphically faces kind of [9]primary stimulus. However, facial recognition is not limited to but Gestalt recognition to pertains of any class of familiar objects or scenes.

2.3 Eye Spacing Measurement for Facial Recognition

Few procedures to computerized facial consciousness has geometric size employe of attribute points of a sapien face. Space between eyes dimension has been recognized an essential stepin reaching the goal. Measurements in spacing has been made by mean of software of the hough radically change method in discovering occasion of a round form and ellipsoidial form which approx the iris’s perimeters and parameter of each of the sclera and the form the place under respective eyebrows. Both gradientx magnitudex anddirection of gradient used in handling noise contaminate the space feature. “Results of this application indicate that measurement of the spacing by detection of the iris is the most accurate of these three methods with measurement by detection of the position of the eyebrows the least accurate. However, measurement by detection of the eyebrows' position is the least constrained method. Application of these strategies has led to size of a attribute function of the human face with adequate accuracy to advantage later inclusion in a full bundle for computerized facial consciousness”..

2.4 A direct LDA algorithm for the high - dimensional data with applications of the recogniting face

“Linear discriminant analysis (LDA) has been successfully used as a dimensionality reduction technique to many classification problems, such as speech recognition, face recognition, and multimedia information retrieval”.The objectives of "nd projections A that maximized the ratio inbetween-class scatter against within - classes scatter

CHAPTER - 3

SYSTEM DEVELOPMENT

3.1 Recognizing the model by listing out the applications

- Face recognitions: Face recognitions system is a technologys that is capables in detecting the objects and verify the person in video and image from running video . There are m ultiple methods with which face recognitions system are working, , it works on c omparing selective face featured image of faces within database. It is als o describ the Biometrics AI applications based that can unique ide ntify a person of analysing patterns of person with different face structure and color. It can be use in surveillance camera, human computer interface and image databases manageme nt.

- Photography: Some recently image capturing camera device is used facial detecting technology for autofocusing. Facial detecting used in selecting the various regions that show in photos or slideshow and use a pan – and - scales Ken Burn effect.

- Marketings: Facial detections systen is gaining the interest in marketing. A webcamera may be integra ted into the television and detecting face that walks by.

3.2 Viola - Jones Algorithm

The Viola -

Jones algorithm is use everywhere now these days, mechanisms of detection of objects. The mainly the training of this algorithm is very slow, but have fast detection rate. Haar feature is used in This algorithm for filtering the feature.

The efficien of this algorithm viola jone is significantly increasing day by day generates the image integral.

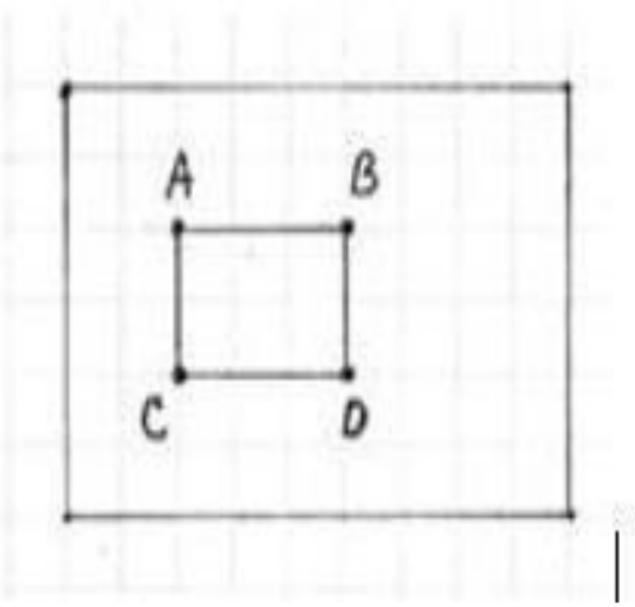


Figure 3.1: Recatangle used

A maximum and minimum window size i s chosen, and for each size a sliding step sizes arechosen. Then the detection window is mo ved across the image as follows:

1. Sets as minimum size window, and step sliding correspond to the size.

2. For the size window, window is slidded vertically and horizontally with the sam e step. At each and every steps, sets of N facial recognitions system filtes are applied every time. If thefilters give us the posit ive answer then the facial is detected.

3. If the window size is the maximum size stop the procedure. Otherwise increase t he size of the window and corresponding sliding step to the next chosen size and go to the step 2.Each face recognition filter (from the set of N filters) contains a set of cascade-connected classifiers. Each classifier looks at a rectangular subset of the detection window and determines if it looks like a face. If it does, the next classifier is applied. If all classifiers give a positive answer, the filter gives a positive answer and the face is recognized . Otherwise the next filter in the set of N filters is run.

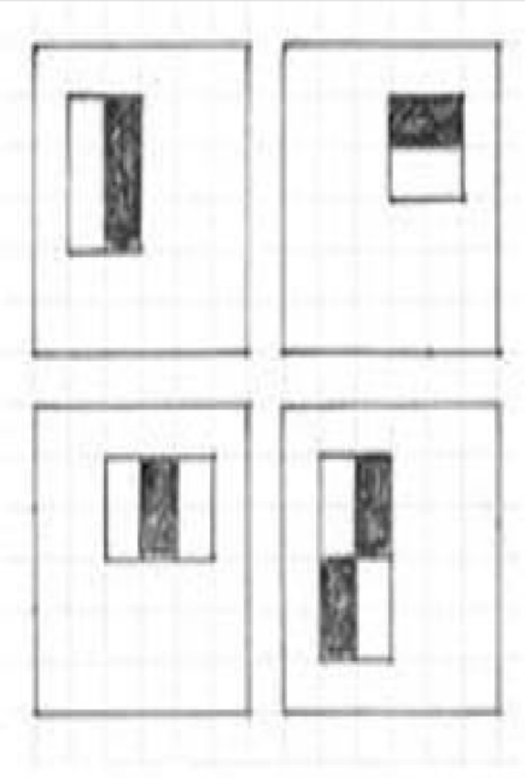


Figure 3.2:Classifiers

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection is using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade functi on is trained from a lot of positive and negative images. It is then used to detect objects in other images.

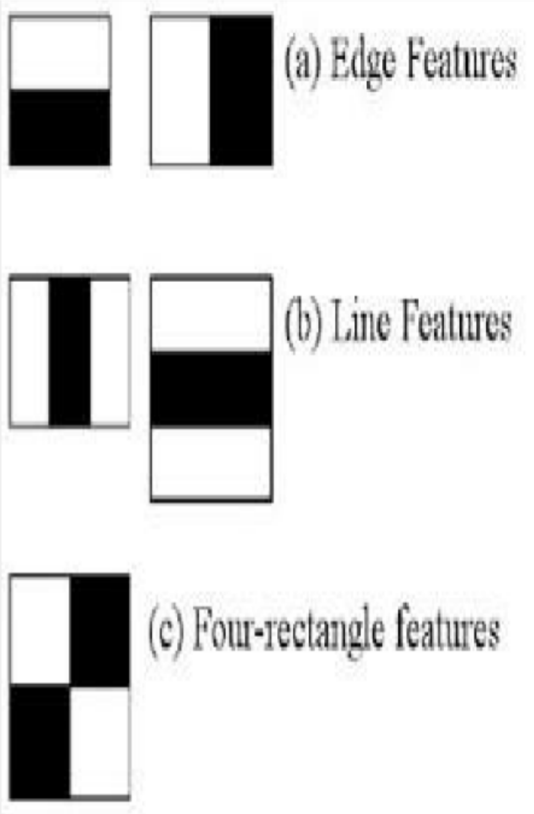


Figure 3.3

2

Now, all possible sizes and locations of each kernel are used to calculate lots of features.

(Just imagine how much computation it needs? Even a 24x24 window results over 16000 0 features). For each feature calculation, we need to find the sum of the pixels under whit e and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just fo ur pixels. Nice, isn't it? It makes things super-fast.

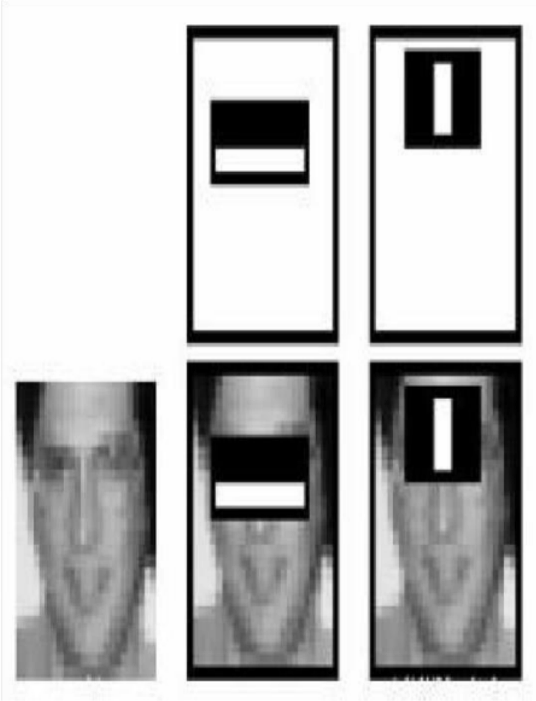
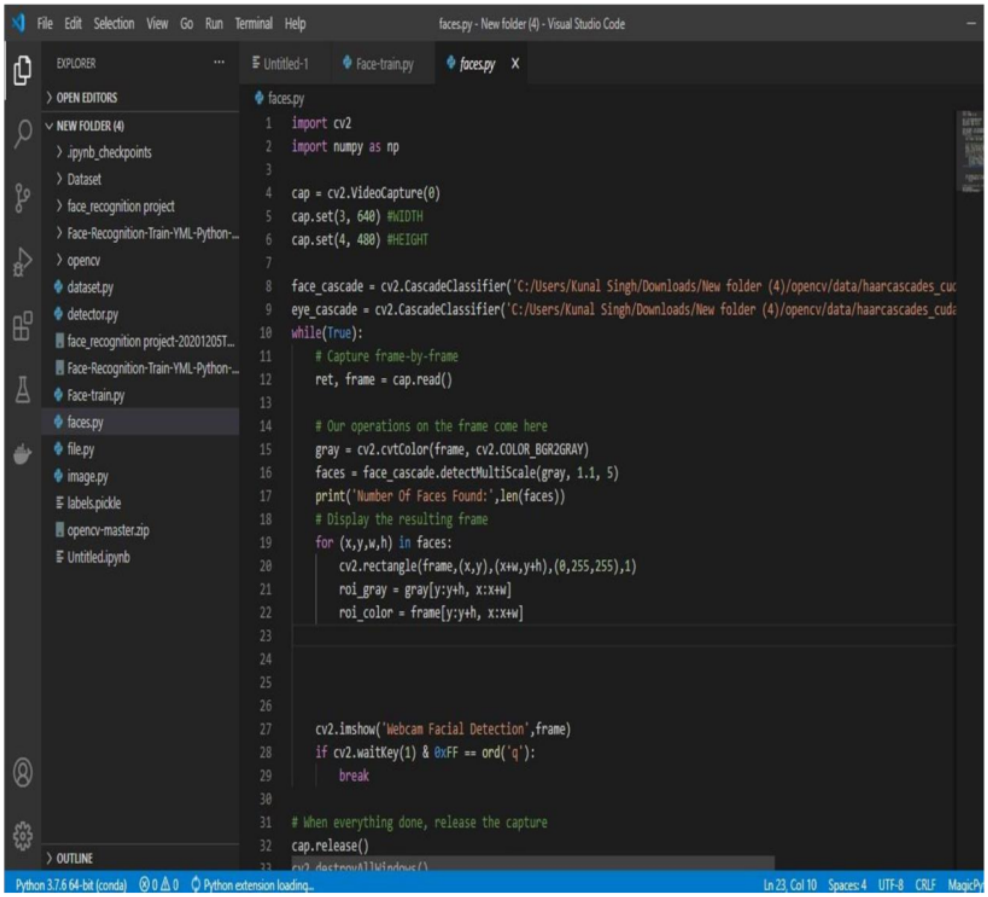


Figure 3.4



File Edit Selection View Go Run Terminal Help
faces.py - New folder (4) - Visual Studio Code
EXPLORER
OPEN EDITORS
NEW FOLDER (4)
.ipynb_checkpoints
Dataset
face_recognition project
Face-Recognition-Train-YML-Python-...
opencv
dataset.py
detector.py
face_recognition project-20201205T...
Face-Recognition-Train-YML-Python-...
Face-train.py
faces.py
file.py
image.py
labels.pickle
opencv-master.zip
Untitled.ipynb
OUTLINE
Untitled-1
Face-train.py
faces.py
Python 3.7.6 64-bit (conda)
Python extension loading...
Ln 23, Col 10 Spaces: 4 UTF-8 CRLF MagicPy

```
import cv2
import numpy as np

cap = cv2.VideoCapture(0)
cap.set(3, 640) #WIDTH
cap.set(4, 480) #HEIGHT

face_cascade = cv2.CascadeClassifier('C:/Users/Kunal Singh/Downloads/New folder (4)/opencv/data/haarcascades_cuc
eye_cascade = cv2.CascadeClassifier('C:/Users/Kunal Singh/Downloads/New folder (4)/opencv/data/haarcascades_cuda
while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()

    # Our operations on the frame come here
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.1, 5)
    print('Number Of Faces Found:',len(faces))
    # Display the resulting frame
    for (x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,255),1)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]




    cv2.imshow('Webcam Facial Detection',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

Figure 3.5:Faces.py file

CHAPTER – 4
TRAINING AND TESTING

4.1 TRAINING IN OPENCV

In OpenCV, training refers to providing a recognizer algorithm with training data to learn from. The trainer uses the same algorithm (LBPH) to convert the images cells to histograms and then computes the value of cell which are included and concatenate on histogram, vectors feature may obtaine. Images is classifie by the process of the profile attached. Input images is being classify by processes and compares data which are stored in database and distances are obtaine by the system . By setting up a threshold, it can be identified if it is a known or unknown face.Eigenface and Fisherface compute the dominant features of the whole training set while LBPH analyses them individually.

To do so, firstly, a Dataset is created. You can either create your own dataset or start with one of the available face databases.

- Yale Face Database
- AT & T Face Database

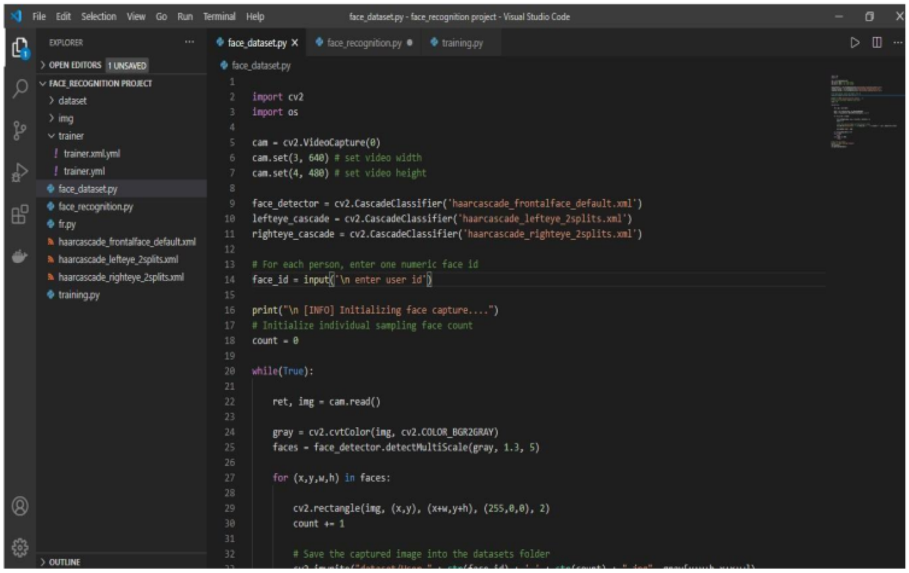
The .xml or .yml configuration file is made from the several features extracted from your dataset with the help of the FaceRecognizer Class and stored in the form of feature vectors.

4.4 CODE

Given below is the code for creating a .yml file, that is the configuration model that stores features extracted from datasets using the FaceRecognizer Class. It is stored in a folder named ‘recognizer’ under the name ‘training Data.yml’.

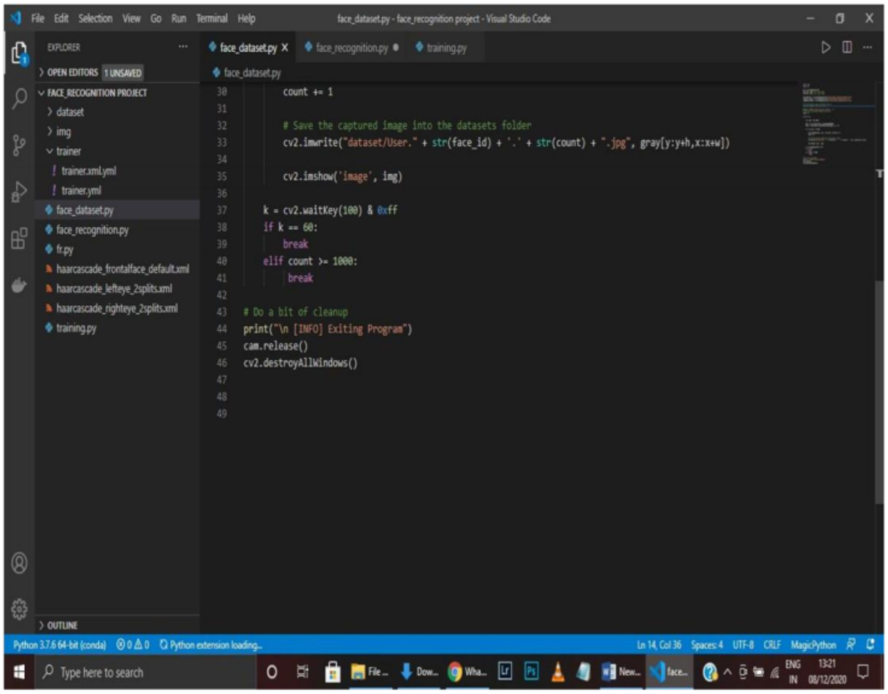
DATASET:

This is the code that will be used to create a dataset. It will turn the camera and take number of pictures for few seconds.Given below is the code for face_dataset.py


File Edit Selection View Go Run Terminal Help
face_dataset.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py

```python

import cv2
import os

cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video width
cam.set(4, 480) # set video height

face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
lefteye_cascade = cv2.CascadeClassifier('haarcascade_lefteye_2splits.xml')
righteye_cascade = cv2.CascadeClassifier('haarcascade_righteye_2splits.xml')

# For each person, enter one numeric face id
face_id = input('\n enter user id')

print("\n [INFO] Initializing face capture....")
# Initialize individual sampling face count
count = 0

while(True):

    ret, img = cam.read()

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_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)
        count += 1

        # Save the captured image into the datasets folder
```

Figure 4.1: Code snippet for the dataset

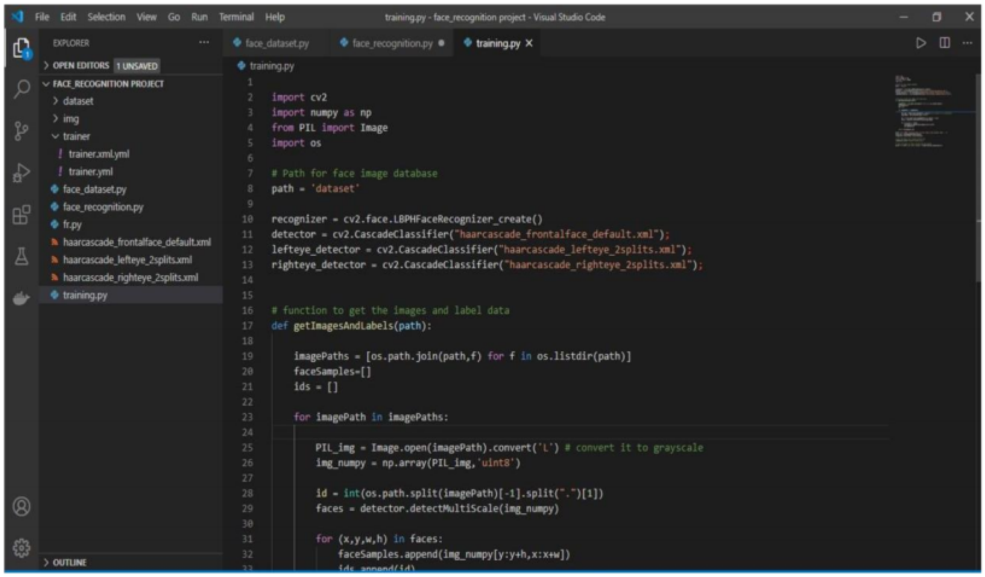

File Edit Selection View Go Run Terminal Help
face_dataset.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py
Python 3.7.6 64-bit (conda)
Python extension loading...
Ln 14, Col 36 Spaces: 4 UTF-8 CRLF MagicPython
Type here to search
ENG IN
13:21
08/12/2020

```
        count += 1

        # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])

        cv2.imshow('image', img)

    k = cv2.waitKey(100) & 0xff
    if k == 60:
        break
    elif count >= 1000:
        break

# Do a bit of cleanup
print("\n [INFO] Exiting Program")
cam.release()
cv2.destroyAllWindows()
```

Figure 4.2: Face_dataset.py

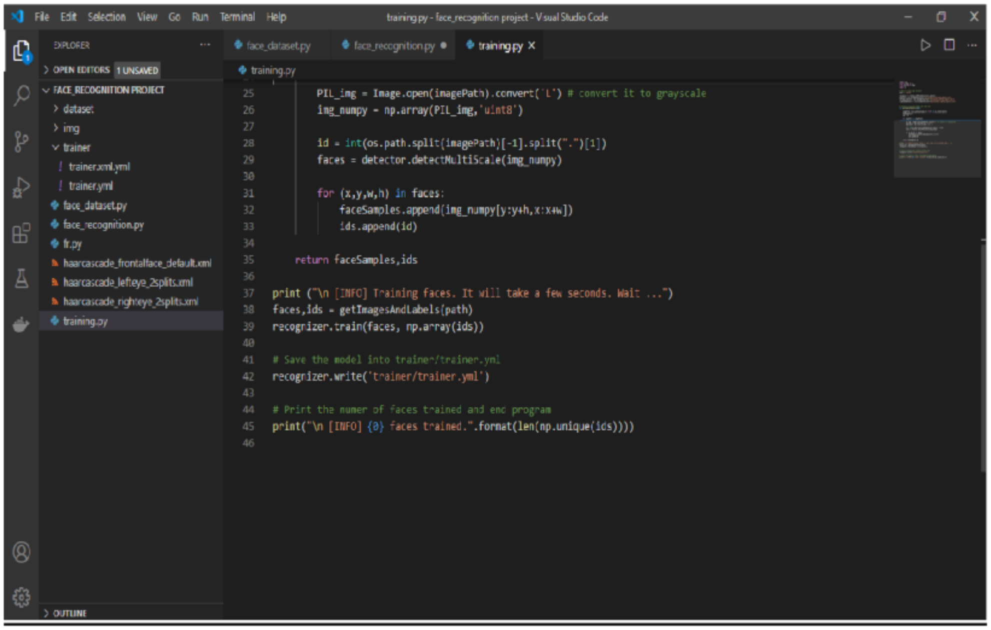
TRAINING:

This is the code that is going to be used to train and get the train.yml file


File Edit Selection View Go Run Terminal Help
training.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py

```python

import cv2
import numpy as np
from PIL import Image
import os

# Path for face image database
path = 'dataset'

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
lefteye_detector = cv2.CascadeClassifier("haarcascade_lefteye_2splits.xml");
righteye_detector = cv2.CascadeClassifier("haarcascade_righteye_2splits.xml");


# function to get the images and label data
def getImagesAndLabels(path):

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

    for imagePath in imagePaths:

        PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
        img_numpy = np.array(PIL_img,'uint8')

        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)

        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)
```

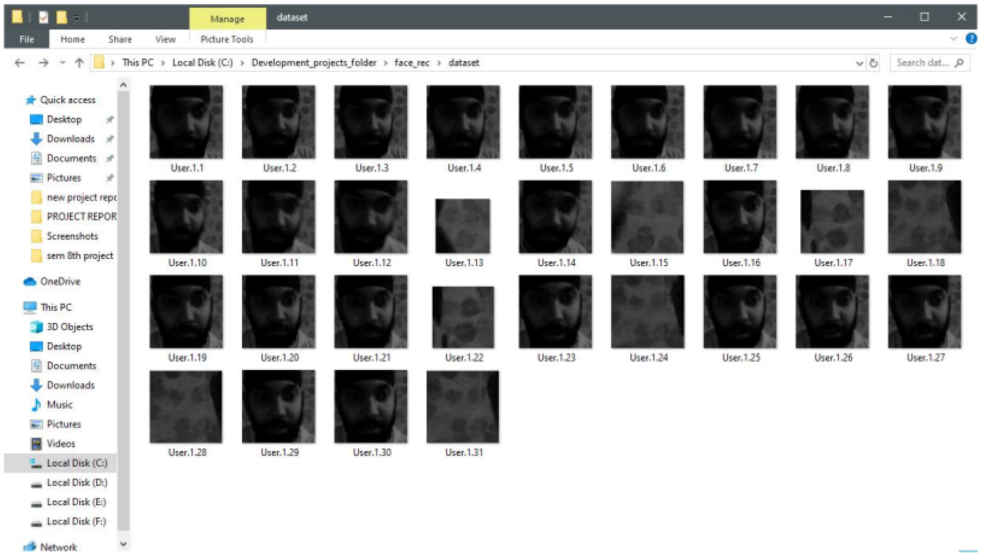
Figure 4.3: Training the dataset


File Edit Selection View Go Run Terminal Help
training.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py

```
        PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
        img_numpy = np.array(PIL_img,'uint8')

        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)

        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)

    return faceSamples,ids

print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))

# Save the model into trainer/trainer.yml
recognizer.write('trainer/trainer.yml')

# Print the numer of faces trained and end program
print("\n [INFO] {0} faces trained.".format(len(np.unique(ids))))
```

Figure 4.4

OUTPUT : After running dataset code we will get numberof pictures in the folder named dataset. Now therse photos will be used to train. The more the pics the greater the accuracy.



4.5 LBPH RECOGNIZER

The approach that has been used in this project is, LBPH approach which uses the following algorithm to compute the feature vectors of the provided images in the dataset.

In this section, it is shown a step-by-step explanation of the LBPH algorithm:

1.First of all, we need to define the parameters (radius, neighbours, grid x and grid y) using the Parametersstructure from the lbph package. Then we need to call the Init function passing the structure with the parameters.

2. Secondly, we need to train the algorithm. To do that we just need to call the Train function passing a slice of images and a slice of labels by parameter. All images must have the same size.

3. The Train function will first check if all images have the same size. If at least one image has not the same size, the Train function will return an error and the algorithm will not be trained.

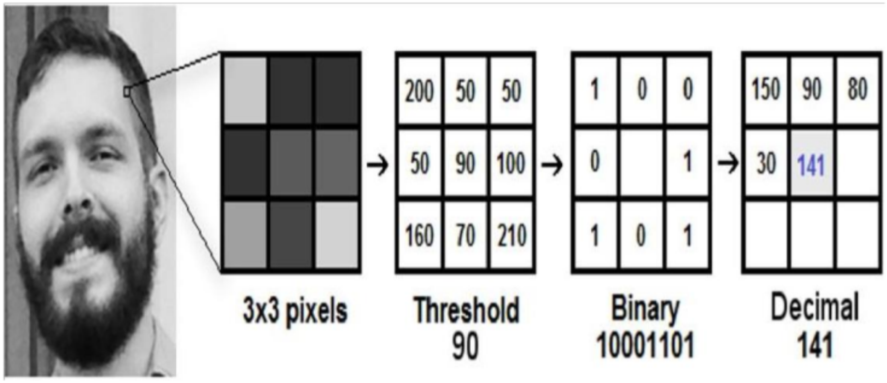


Figure 4.6: referencing and assigning pixel value

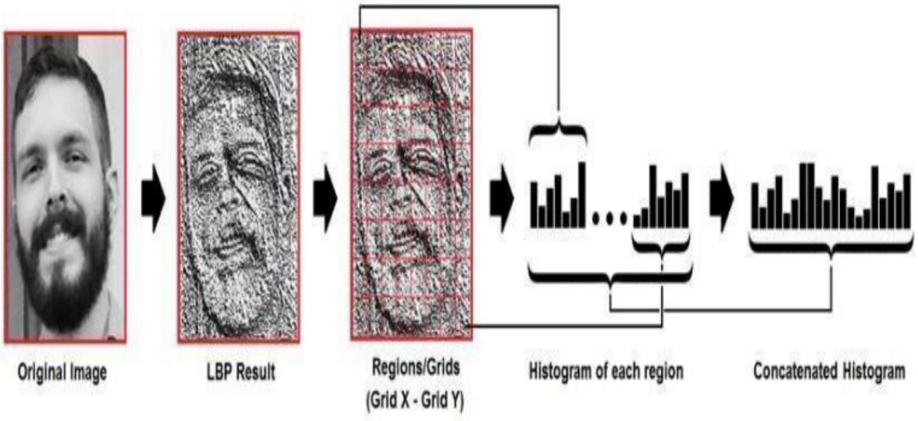


Figure 4.7: LBP result

The images, labels, and histograms are stored in a data structure so we can compare all of it to a new image in the Predict function.

The LBPH package provides the following metrics to compare the histograms: **Chi-Square** :

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

Equations 1 **Euclidan**

Distances :

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

Equation 2

Normalize Euclidan Distances :

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

Equation 3

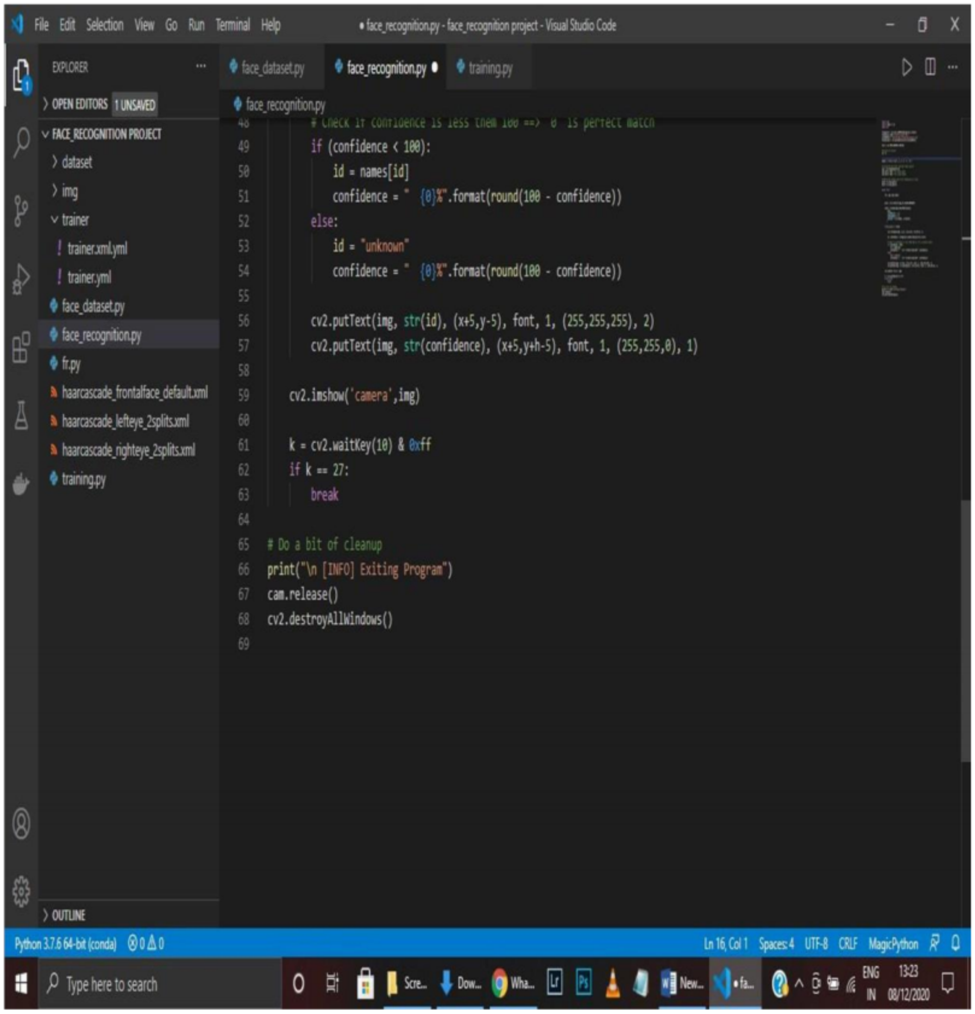
Absolute Values :

$$D = \sum_{i=1}^{n} |hist1_i - hist2_i|$$

Equation 4

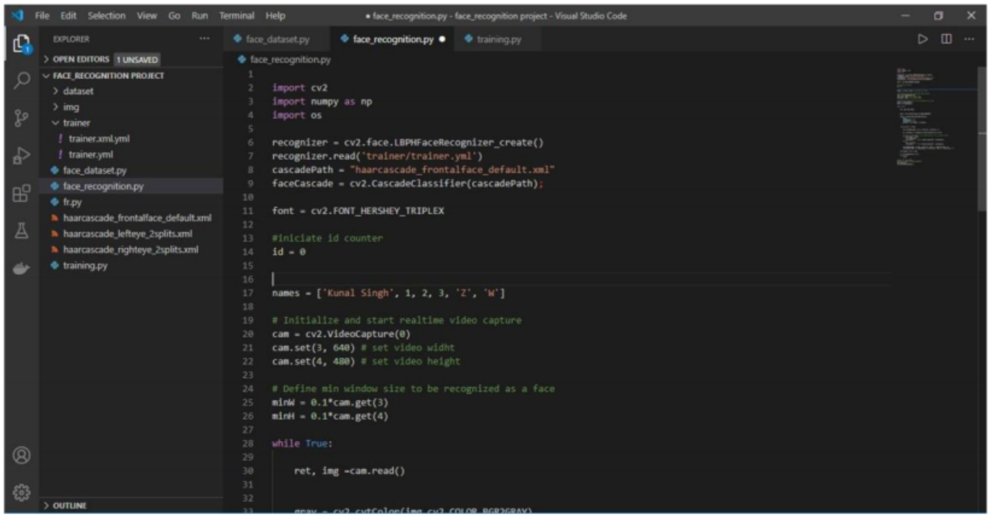
4.7 Code

Given below is the face recognition script that reads the data from the trainingData.yml file mentioned before and uses the .predict() function to pass a confidence value, recognize the face of the individual and display their names along with their face. The following script uses data that has been trained with images of the students working on this project


File Edit Selection View Go Run Terminal Help
• face_recognition.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py
Python 3.7.6 64-bit (conda)
Ln 16, Col 1 Spaces: 4 UTF-8 CRLF MagicPython
Type here to search
ENG IN 13:23 08/12/2020

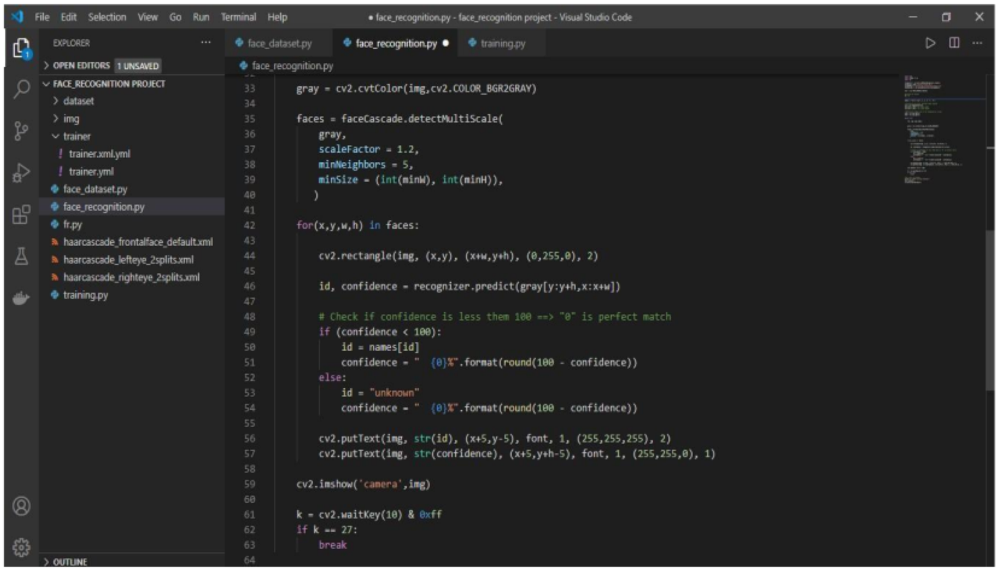
```
        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 100):
            id = names[id]
            confidence = "  {0}%".format(round(100 - confidence))
        else:
            id = "unknown"
            confidence = "  {0}%".format(round(100 - confidence))

        cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

    cv2.imshow('camera',img)

    k = cv2.waitKey(10) & 0xff
    if k == 27:
        break

# Do a bit of cleanup
print("\n [INFO] Exiting Program")
cam.release()
cv2.destroyAllWindows()
```

Figure 4.8: Functioning


File Edit Selection View Go Run Terminal Help
• face_recognition.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py

```python

import cv2
import numpy as np
import os

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT_HERSHEY_TRIPLEX

#iniciate id counter
id = 0


names = ['Kunal Singh', 1, 2, 3, 'Z', 'W']

# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video widht
cam.set(4, 480) # set video height

# Define min window size to be recognized as a face
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)

while True:

    ret, img =cam.read()


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

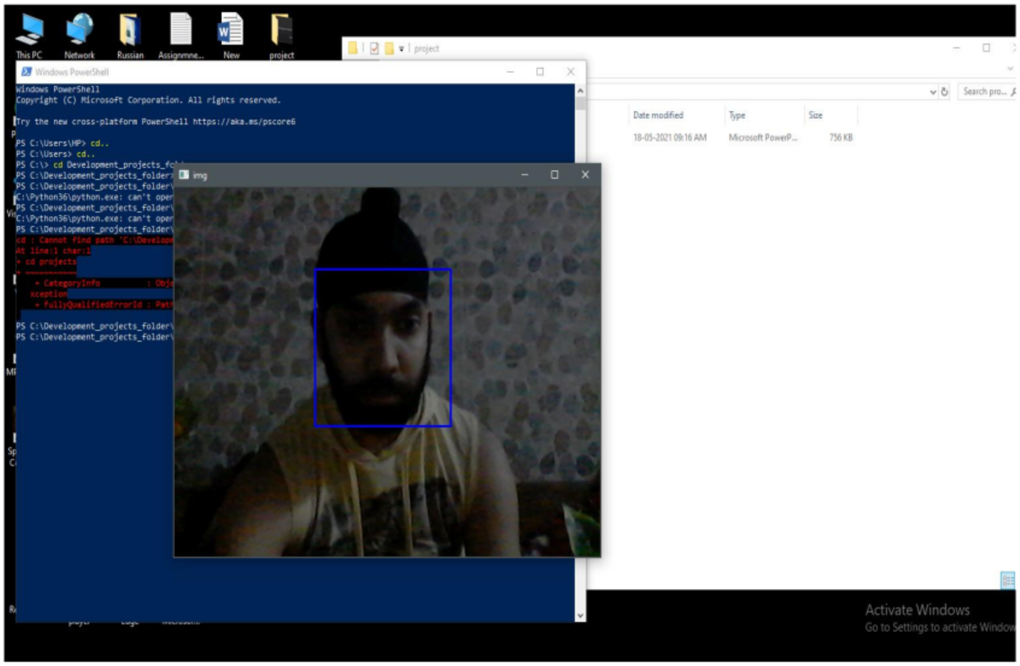
Figure 4.9: Other snippets


File Edit Selection View Go Run Terminal Help
• face_recognition.py - face_recognition project - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
FACE_RECOGNITION PROJECT
dataset
img
trainer
trainer.xml.yml
trainer.yml
face_dataset.py
face_recognition.py
fr.py
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
training.py
OUTLINE
face_dataset.py
face_recognition.py
training.py

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

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
       )

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

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

        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 100):
            id = names[id]
            confidence = "  {0}%".format(round(100 - confidence))
        else:
            id = "unknown"
            confidence = "  {0}%".format(round(100 - confidence))

        cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

    cv2.imshow('camera',img)

    k = cv2.waitKey(10) & 0xff
    if k == 27:
        break

```

Figure 4.10: Face_recognition.py

OUTPUT



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\HP> cd..
PS C:\Users> cd..
PS C:\> cd Development_projects_fo
PS C:\Development_projects_folder>
PS C:\Development_projects_folder\
C:\Python36\python.exe: can't oper
PS C:\Development_projects_folder\
C:\Python36\python.exe: can't oper
PS C:\Development_projects_folder\
cd : Cannot find path 'C:\Develop
At line:1 char:1
+ cd projects
+ ~~~~~~~~~~~
    + CategoryInfo          : Obje
   xception
    + FullyQualifiedErrorId : Path

PS C:\Development_projects_folder\
PS C:\Development_projects_folder\
```

Date modified	Type	Size
18-05-2021 09:16 AM	Microsoft PowerP...	756 KB

4.9 APPLICATIONS

- Security: Face Recognition can help in developing security measures, that is unlocking of a safe using facial recognition.

- Attendance Systems: Face Recognition can be used to train a set of users in order to create and implement an automatic attendance system that recognizes the face of the individual and marks their attendance.

- Access: Face Detection can be used to access sensitive information like your bank account and it can also be used to authorize payments.

- Mobile Unlocking: This feature has taken the mobile phone industry by a storm and almost every smart phone manufacturing company has their flagship smartphones being unlocked using face recognition. Apple’s FaceID is an excellent example.

- Law Enforcement: This is a rather interesting way of using face detection and face recognition as it can be used to assess the features of a suspect to see if they are being truthful in their statements or not.

- Healthcare: Face Recognition and Detection can be used in the healthcare sector to assess the illness of a patient by reading their facial features.

CHAPTER 5:CONCLUSION AND FUTURE SCOPE

5.1 Future Scope

- Government/ Identity Management: Governments all around the world are using face recognition systems to identify civilians. America has [12] one of the largest face databases in the world, containing data of about 117 million people.

- Emotion & Sentiment Analysis: Face Detection and Recognition have brought us closer to the technology of automated psyche evaluation. As systems now a days can judge the precise emotions frame by frame in order to evaluate the psyche.

- Authentication systems: Various devices like mobile phones or even ATMs work using facial recognition, thus making getting access or verification quicker and hassle free.
- Full Automation: This technology helps us become fully automated as there is very little to zero amount of effort required for verification using facial recognition.

- High Accuracy: Face Detection and Recognition systems these days have developed very high accuracy and can be trained using very small data sets and the false acceptance rates have dropped down significantly.

5.2 Limitations

- Data Storage: Extensive data storage is required for creating, training and maintaining big face databases which is not always feasible.

- Computational Power: The requirement of computational power also increases with increase in the size of the database. This becomes financially out of bounds for smaller organizations.
- Camera Angle: The relative angle of the target's face with the camera impacts the recognition rate drastically. These conditions may not always be suitable, therefore creating a major drawback.

5.3 Conclusion

Facial Detection and Recognition systems are gaining a lot of popularity these days. Most of the flagship smartphones of major mobile phone manufacturing companies use face recognition as the means to provide access to the user.

This project report explains the implementation of face detection and face recognition using OpenCV with Python and also lays out the basic information that is needed to develop a face detection and face recognition software.The goal of increasing the accuracy of this project will always remain constant and new configurations and different algorithms will be tested to obtain better results. In this project, the approach we used was that of Local Binary Pattern Histograms that are a part of the FaceRecognizer Class of OpenCV.

References

[1] Schneiderman.United States of America Patent U.S. Patent No. 8,457,367, 2013.

[2] R. J. Baron, “Mechanisms of human facial recognition,” *International Journal of Man- Machine Studies.*

[3] M. Nixon, “"Eye Spacing Measurement for Facial Recognition",” *International Society for Optics and Photonics.,* vol. (Vol. 575), (19 December 1985).

[4] H. &. Y. J. Yu, “A direct LDA algorithm for high-dimensional data—with application to face recognition,” 2001.

final_report-converted.pdf

ORIGINALITY REPORT

14%	13%	5%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijtrd.com Internet Source	3%
2	www.docs.opencv.org Internet Source	2%
3	Submitted to CSU, Los Angeles Student Paper	1%
4	www.geeksforgeeks.org Internet Source	1%
5	opencv.org Internet Source	1%
6	en.wikipedia.org Internet Source	1%
7	www.ijarse.com Internet Source	1%
8	Submitted to Higher Education Commission Pakistan Student Paper	1%
9	www.sciencedirect.com Internet Source	1%

10	softwarelist.us Internet Source	<1 %
11	wareseeker.com Internet Source	<1 %
12	drrajivdesaimd.com Internet Source	<1 %
13	www.irjet.net Internet Source	<1 %
14	pillow.readthedocs.io Internet Source	<1 %
15	www.vocal.com Internet Source	<1 %
16	Submitted to Sogang University Student Paper	<1 %
17	Submitted to Birkbeck College Student Paper	<1 %
18	Submitted to DIT university Student Paper	<1 %
19	Robert J. Baron. "Mechanisms of human facial recognition", International Journal of Man-Machine Studies, 1981 Publication	<1 %
20	Submitted to University of Petroleum and Energy Studies Student Paper	<1 %

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: …17/05/2021……

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report [✓ B.Tech Project Report] Paper

Name: GURUPDESH SINGH **Department:** CSE **Enrolment No** 171261

Contact No 8968020888 **E-mail.** gurupdeshsingh14@gmail.com

Name of the Supervisor: Dr Monika Bharti

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): FACE DETECTION AND RECOGNITION

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

– Total No. of Pages = 49
– Total No. of Preliminary pages =10
– Total No. of pages accommodate bibliography/references =1

Gurupdesh Singh

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 14 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Monika Bharti

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String	14%	Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com