A REPORT OF PROJECT

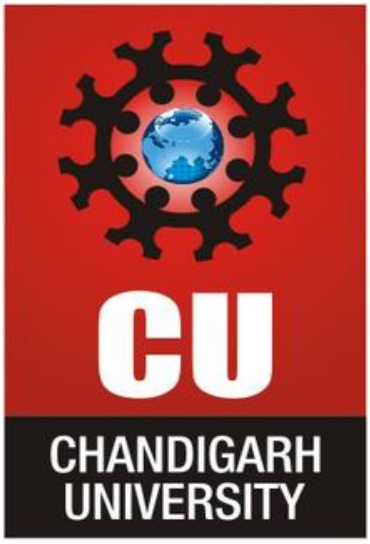
ON

FACIAL RECOGNITION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

(Computer Science & Engineering)



SUBMITTED BY:

HARSH, VEERESH

16BCS3054, 16BCS3059

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY GHARUAN, MOHALI

(2018-2019)

ACKNOWLEDGEMENT

The completion of this training work could not have been possible without continued & dedicated efforts & guidance of large number of faculty & staff members of the institute. I express my gratitude to all of them. The acknowledgement however will be incomplete without specific mention as follows

I wish to acknowledge my deep gratitude to **Mr. Suman Sarkar**, Assistant Professor at **AIT, Chandigarh University** for his cooperation and guidance.

Finally, I would like to say that I am indebted to my parents for everything that they have done for me. All of this would have been impossible without their constant support.

_____

(Mr. Suman Sarkar, Assistant Prof. AIT)

ABSTRACT

Project Objectives

The objective of the project is to develop a system which allows users to get their faces recognised. It allows facial recognition of your friends, family, and colleagues.

A face recognition can have up to 6000 classifiers all of which have to be watched for a face to be recognized but therein lies a problem for face detection, the algorithm starts at the top left of a picture and moves down across small blocks of data, looking at each block, constantly asking, “Is this a face?” Since there are 6000 or more tests, per block we might have millions of calculations to do, which will grind our computer to a halt.

A simple application allows you to detect your face and then the face shown would get saved so that when you show that face again it would remember it and show that name again.

Approach

When we were first given this project, we met to determine how we were to carry out the task assigned to us. We drew up a time-line, discussed about the programming language to use to carry out the task, how the GUI would look like and also to make sure that we understood what was assigned to us. We finally settled for Python as our programming language. We got more information on what we were to do and set about completing the task, making use of the new ideas taught in class.

INDEX

FACE RECOGNITION

What is Face Recognition?

Facial recognition is a category of biometric software that maps an individual's facial features mathematically and stores the data as a face print. The software uses deep learning algorithms to compare a live capture or digital image to the stored face print in order to verify an individual's identity.

High-quality cameras in mobile devices have made facial recognition a viable option for authentication as well as identification. Apple’s iPhone X, for example, includes Face ID technology that lets users unlock their phones with a face print mapped by the phone's camera. The phone's software, which is designed with 3-D modelling to resist being spoofed by photos or masks, captures and compares over 30,000 variables. As of this writing, Face ID can be used to authenticate purchases with Apple Pay and in the iTunes Store, App Store and iBooks Store. Apple encrypts and stores face print data in the cloud, but authentication takes place directly on the device.

Developers can use Amazon Recognition, an image analysis service that's part of the Amazon AI suite, to add facial recognition and analysis features to an application. Google provides a similar capability with its Google Cloud Vision API. The technology, which uses machine learning to detect, match and identify faces, is being used in a wide variety of ways, including entertainment and marketing. The Kinect motion gaming system, for example, uses facial recognition to differentiate among players. Smart advertisements in airports are now able to identify the gender, ethnicity and approximate age of a passer-by and target the advertisement to the person's demographic.

Facebook uses facial recognition software to tag individuals in photographs. Each time an individual is tagged in a photograph, the software stores mapping information about that person’s facial

characteristics. Once enough data has been collected, the software can use that information to identify a specific individual's face when it appears in a new photograph. To protect people's privacy, a feature called Photo Review notifies the Facebook member who has been identified.

Currently, there are no laws that specifically protect an individual's biometric data. Facial recognition systems are currently being studied or deployed for airport security and it's estimated that more than half the United States population has already had their face print captured. According the Department of Homeland Security, the only way to avoid having biometric information collected when traveling internationally is to refrain from traveling. The General Data Protection Regulation (GDPR) for European Member States does address biometric data.

TOOLS USED

a) Python IDLE

Python has many characteristics that have contributed to its popularity:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

1.1 **Python Platform:**

Python is an object oriented, interpreted, high-level programming language. It's high-level built in data structures, combined with dynamic typing and dynamic binding, make it easy for Rapid Application Development, as well as for scripting language connecting the already existing components together.

Python is simple, easy to learn syntax and works for readability and therefore reduces costs for program maintenance. It shows growth for modules and packages. It has the objective for modularity and code reusability. The interpreter and standard library are available and fairly distributed.

b) OPEN CV

OpenCV is machine learning software library and an open source computer vision. OpenCV is built to provide an easy structure for computer vision applications and to fasten the use of machine vision in the commercial products. Being a licensed product, OpenCV is making it easy for commerce to use and modify the code.

The library has like more than 2500 optimized algorithms, including set of classic and state-of-art computer vision and machine learning algorithms. They can be used to detect and recognize faces, objects, classify human actions in videos, camera movements, track moving objects, 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has greater than 47 thousands of people of user community and estimated number of downloads is exceeding 14 million. This library is utilised greatly in companies, research organisations.

LANGUAGES USED

1. **Python:**
   1.1. Version 3.7.X
   1.2. Latest PIP version

2. **XML:**
   2.1. For Cascading
   2.2. For Array storage

3. **YML:**
   3.1. For Training model
   3.2. For storing trained model results.

WORK UNDERTAKEN

PROJECT: Implementing OpenCV to make facial recognition system using a training based model.

HARDWARE REQUIREMENTS:

1.1. PC with monitor, keyboard and mouse.

1.2. CPU

1.2.1. 4GB RAM with Core i3 or above

1.2.2. 2.4 GHz Processor or above.

SOFTWARE REQUIREMENTS:

2.1. Python

2.1.1. 3.0.X or higher

2.1.2. Pip 18 configured

2.1.3. Opencv installed

OBJECTIVE:

This project will serve the following objectives:

- To detect new faces
- To recognise already shown faces
- To maintain records of old faces

ABOUT PROJECT:

A face recognition can have up to 6000 classifiers all of which have to be watched for a face to be recognized but therein lies a problem for face detection, the algorithm starts at the top left of a picture and moves down across small blocks of data, looking at each block, constantly asking, “Is this a face?” Since there are 6000 or more tests, per block we might have millions of calculations to do, which will grind our computer to a halt.

RELATED WORK:

Image acquisition and imaging conditions:

Generally, in consumer digital imaging, face recognition must contend with uncontrolled lighting conditions, large pose variations, facial expressions, makeup, changes in facial hair, ageing and partial occlusions without forgetting that the human face is not a unique rigid object. Similarly, in scenarios such as visual surveillance, videos are often acquired in uncontrolled situations or from moving cameras.

When the image is formed, factors such as lighting (spectra, source distribution and intensity) and camera characteristics (sensor response and lenses) affect to some degree the appearance of the human face. Illumination variations can also do this because of skin reflectance properties and because of the internal camera control.

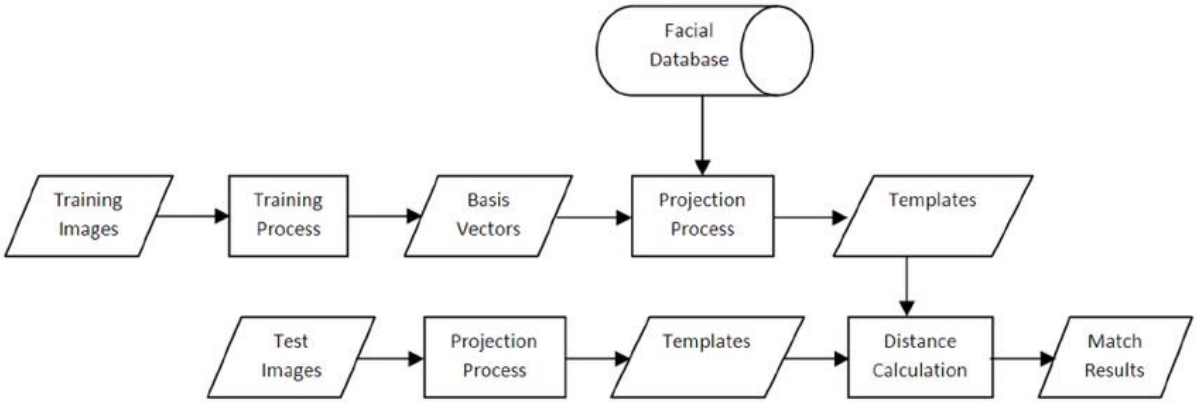
The problem also persists when we change the pose of the person that is to be detected. Generally the recognition software is used to look through a video recorded by a camera and mostly the places are heavily populated which means that people with same side face cuts are hard to differentiate between.

Benchmark dataset:

Systematic data collection and evaluation of face recognition systems on a fair basis are also considered a kind of challenges and problems that are related to research in the face recognition field. Collecting enough data to train the software to recognize face correctly is a time taking and tough process. Then training it to identify 2 faces at a time is also matter of resources.

The dataset is in the form of images and contains all the front and side views of people. This also involves variations such as wearing different cloths, covering the face partially with scarf, etc. The model is also trained against objects to which there is no result as expected. Similar looking masks have also been tested and the results are no match as expected.

DIAGRAM:



1. Working of the project using flowchart

The following diagram tells us how the program works.

MODULES:

1. **Training:** This module includes the part where we train the AI to recognize faces. The working is as follows:

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

Intially we have imported all the libraries that we are going to use later. ‘import os’ and ‘import cv2’ are directly importing opencv and operating system library. ‘import numpy as np’ indicated that we have imported numpy and we will be refering it as ‘np’ during usage. ‘from PIL import Image’ indicates that Image directory is imported from PIL library.

```
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
image_dir = os.path.join(BASE_DIR, "images")
```

In the above, BASE_DIR is an instance of os.path.dirname() which gives the name of the base directory. Similarly image_dir is an instance of os.path.join() that gives the directory name.

```
face_cascade = cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_alt2.xml')
recognizer = cv2.face.LBPHFaceRecognizer_create()
```

Here we have again defined instance of cv2.CascadeClassifier() which defines the path to haarcascade that we are going to use. 'haarcascade' are .xml files that defines the deepcuts and array values.

Similarly, cv2.face.LBPHFaceRecognizer_create() will define an instance of module.

```
pil_image = Image.open(path).convert("L")
```

This will convert the image to grayscale.

```
size=(550,550)
final_image = pil_image.resize(size, Image.ANTIALIAS)
```

This will set all the size of every image as 550x550. This enables us to keep all the images to a specific standard. This enables a more precise detection.

```
image_array = np.array(final_image, "uint8")
```

This will convert the image to numpy array.

```
recognizer.train(x_train, np.array(y_labels))
recognizer.save("trainner.yml")
```

We save the trained model into a 'trainner.yml' file. This file will be used to recognize faces once we start working with face recognition program.

2. **Face Recognition:** This module will use the 'trainner.yml' file that we have created using the Trainer module.

```
import numpy as np
import cv2
import pickle
```

We start by importing opencv module, cv2, pickle to use labels and numpy as np.

```
face_cascade = cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_alt2.xml')
```

We use the cv2.CascadeClassifier () here as well to import the haarcascade file. This will enable us to identify the face by recognizing deep cuts.

```
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read("trainner.yml")
```

Again cv2.face.LBPHfaceRecognizer_create () is an instance for recognizer module and recognizer.read () enables us to import trainner.yml file.

```
cap=cv2.VideoCapture(0)
```

We start the camera service and store the instance of all camera window related variable in variable ‘cap’.

```
ret, frame =cap.read()
```

The above lets us store all the frame values of the camera window. The ‘ret’ variable helps in capturing frame by frame video.

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.5, minNeighbors=5)
```

This will convert the image to grayscale and detect face using haarcascade that we previously uploaded such that if a pixel is selected to be in face then it should have at least 5 neighbor pixels to be in face as well.

```
roi_gray=gray[y:y+h, x:x+w]
roi_color=frame[y:y+h, x:x+w]
```

This defines our region of interest in gray as well as colour frames.

```
id_, conf = recognizer.predict(roi_gray)
if conf>=45 and conf<=85:
    print(id_)
    print(labels[id_])
    font = cv2.FONT_HERSHEY_SIMPLEX
    name = labels[id_]
    color = (255,255,255)
    stroke = 2
    cv2.putText(frame,name,(x,y),font,1,color,stroke, cv2.LINE_AA)
```

The ‘conf’ variable here defines our confidence with which our face is being detected. The labels would be displayed by the program only when confidence level is a minimum of 45.

```
color=(255,0,0) #BGR
stroke=2
width=x+w
height=y+h
cv2.rectangle(frame,(x,y),(width,height),color,stroke)
```

This will draw a rectangle on the sides of the faces that are detected.

```
cv2.imshow('frame',frame)
if cv2.waitKey(20) & 0xFF == ord('q'):
    break
```

This is to keep displaying the frame until we press ‘q’ key to quit the resulting window.

```
cap.release()
cv2.destroyAllWindows()
```

Finally we release all the memory and destroy all the variables.

RESULT AND DISCUSSION:

Even the best system developed has some flaws or others. There always exists scope of further improvement in the system. The effect of implementations of new computerized system is found remarkable.

The following are the major improvements of the new application.

- A faster and more organised application or software has been made
- All the operations are carried automatically preventing a lot of manual work.
- Additional checks have also been incorporated into the system to avoid duplications of data as far as possible.

SCOPE FOR FURTHER IMPROVEMENT:

Every project whether large or small has some limitations no matter however diligently developed. In some cases limitations is small while in other cases they may be broad also. The new system has got some limitations. Major areas where modifications can be done are as follows:

- My system is not online so further it can be improved.
- The security is limited so some additional arrangement could be made to provide more security to the system.

CONCLUSION:

This was our project about “Face Recognition”. Development of this system took a lot of efforts from us. Though every task is never said to be perfect in this development field even more improvement may be possible in this system. We learned so many things and gained a lot of knowledge about development field. We hope this will prove fruitful to us.

So we have successfully managed to make a system that recognises faces by the use of deep cut remembrance. We have also integrated database to identify the various profiles related to them.

REFERENCES

- **Udemy:** www.udemy.com
- **Google:** www.google.co.in
- **Python :** www.realpython.com
- **Tutorials point:** www.tutorialpoint.com
- **Opencv:** www.opencv.org
- **YouTube:** www.youtube.com
- **Research:**

  www.researchgate.net/publication/271584966_Face_Recognition_Challenges_Achievements_and_Future_Directions