

**SUMMER INDUSTRIAL TRAINING 2019**

**A Project Report**

**in partial fulfillment for the award of degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**Computer Science & Engineering**

**Under the guidence of**

***Subhadeep Chakroborty***

**Project Carried Out At**

**Name of The Students**

**Sk Rehan Ahmed**

**Hasibul Mondal**

**Maruf Hassan**

**Shabnam Parween**

**Arisha Aftab**

**Samadur Khan**

**College Name:- Aliah University**

**ACKNOWLEDGEMENT**

**We would like to express our gratitude to *Subhadeep Chakroborty* ,my project advisor for providing me an opportunity to work on this project, which gave me a great exposure to develop the skills towards Android Application Development. I thank him for his generous advice and encouragement which helped us to complete the project successfully.**

**Lastly, I would like to thank the entire faculty and staff of our Webskitters Academy.**

**Submitted By:- Submitted To:-**

**GROUP IV Subhadeep Chakroborty**

**Sk Rehan Ahamed**

**Shabnam Parween**

**Maruf Hassan**

**Hasibul Mandal**

**Arisha Aftab**

**Samadur Khan**

**Contents**

* **Introduction**
* **What is Machine Learning ?...**
* **Machine Learning Methods…**
* **Applications Of Machine Learning…**
* **How Does Machine Learning Work ?...**
* **Machine Learning using Python**
* **What is Python ?...**
* **What can Python Do?...**
* **Why We Use Python For Machine Learning?...**
* **Features Of Python for ML…**
* **Project (Face-Recognition)**
* **Objective**
* **Library Used**
* **Face-Recognition Module**
* **OpenCv module**
* **Math Module**
* **Numpy module**
* **Project Code**
* **Project Output**

**INTRODUCTION**

**What is Machine Learning ?**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development on the computer programs that can acess data and use it

Learn for themselves.

**Machine Learning Methods**

Machine learning algorithms are often categorized as supervised or unsupervised.

* **Supervised machine learning algorithms :** Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
* **Unsupervised machine learning algorithms :** Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
* Also Two categories are there which are basically mix with Supervised machine learning and Unsupervised machine learning algorithms :
* **Semi-supervised machine learning algorithms**
* **Reinforcement machine learning algorithms**

**Applications of Machine Learning**

The value of machine learning technology has been recognized by companies across several industries that deal with huge volumes of data. By leveraging insights obtained from this data, companies are able work in an efficient manner to control costs as well as get an edge over their competitors. This is how some sectors / domains are implementing machine learning -

* **Financial Services**
* **Marketing and Sales**
* **Government**
* **Healthcare**
* **Transportation**
* **Oil and Gas**

**How Does Machine Learning Work?**

To get the maximum value from big data, businesses must know exactly how to pair the right algorithm with a particular tool or process and build machine learning models based on iterative learning processes. Some of the key machine learning algorithms are -

* **Random forests**
* **Neural networks**
* **Discovery of sequence and associations**
* **Decision trees**
* **Mapping of nearest neighbor**
* **Supporting vector machines**
* **Boosting and bagging gradient**
* **Self organizing maps**
* **Multivariate adaptive regression**

**Machine Learning Using Python**

**What is Python?**

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

**It is used for:**

⦁ **Web development**

**⦁ Software development**

**⦁ Mathematics**

**⦁ System scripting**

**⦁ Computations and Analysis**

**What can Python do?**

⦁ **Python can be used on a server to create web applications.**

**⦁ Python can be used alongside software to create workflows.**

**⦁ Python can connect to database systems. It can also read and modify files.**

**⦁ Python can be used to handle big data and perform complex mathematics.**

**⦁ Python can be used for rapid prototyping, or for production-ready software Development**

**Why We Use Python For Machine Learning ?**

Python is a widely used high-level programming language for general-purpose programming. Apart from being open source programming language, python is a great object-oriented, interpreted, and interactive programming language. Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level dynamic data types, and dynamic typing. There are interfaces to many system calls and libraries, as well as to various windowing systems. New built-in modules are easily written in C or C++ (or other languages, depending on the chosen implementation). Python is also usable as an extension language for applications written in other languages that need easy-to-use scripting or automation interfaces.

Python is widely considered as the preferred language for teaching and learning Ml (Machine Learning). Few simple reasons are:

* **It’s simple to learn. As compared to c, c++ and Java the syntax is simpler and Python also consists of a lot of code libraries for ease of use.**
* **Though it is slower than some of the other languages, the data handling capacity is great.**
* **Open Source! – Python along with R is gaining momentum and popularity in the Analytics domain since both of these languages are open source.**
* **Capability of interacting with almost all the third party languages and platforms.**

**Features of Python for ML :**

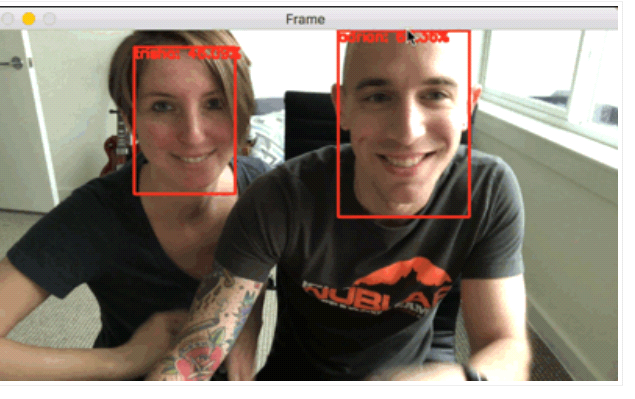
* **NumPy Package**
* **Pandas Package**
* **Matplotlib**
* **Seaborn**
* **Sklearn**

All these in built packages of Python are advancely used for Machine Learning which makes Python unique from others.

**ABOUT PROJECT**

**FACE-RECOGNITION**

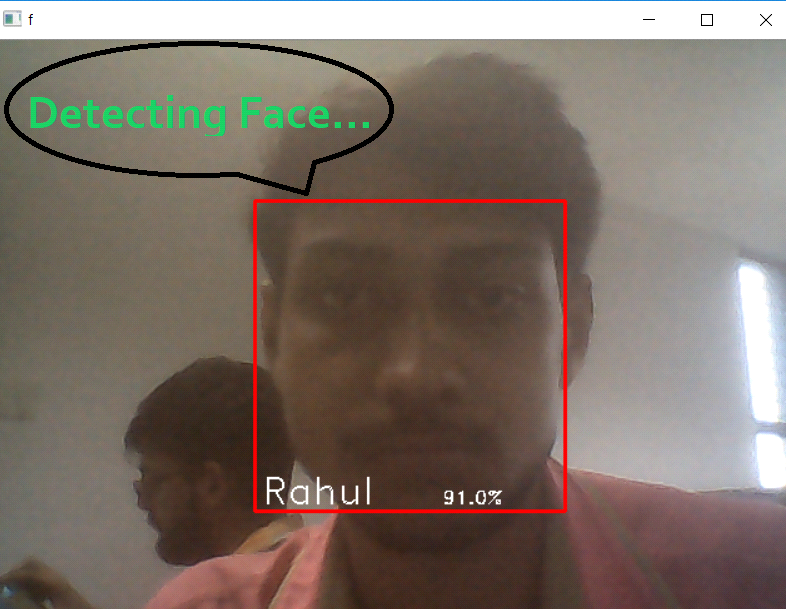
**OBJECTIVE: Objective is to recognize a face using Python from a webcam by putting a box and get the matching percentage if the face is known to the database.**



**Library Used :**

**Face-Recognition Module :** This is a module by which we get the features of face and detect a face from a picture or frame(in case of Video). Through this module we get each pixel point of an image in a 3-D matrix for colored image and 1-D matrix for a black and white image through the Face\_Encoding method of this module . And Its Face\_Locations object also helps us to detect the co-ordinate of an object with respect to the Frame.

**Why We use Face-Recognition Module ?**



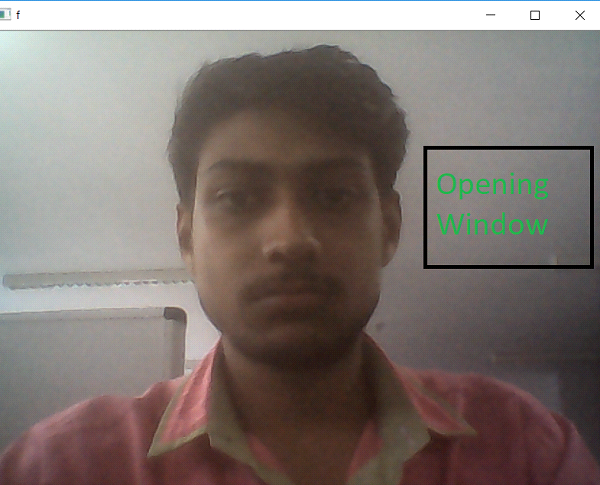
This module is so Powerful that it is very easy to recognize the face with a great accuracy of 99.38 % . The **D-Library(dlib)** has a big contribution for it . To use the **face-recognition** module we will have to install the dlib for windows. Basically dlib is a library of **C++**  and to use it we will also have to install some C++ modules.

And for dlib, accuracy also increased in face-recognition.

**OpenCv module :** OpenCv is open source module for getting graphical interface and free for commercial use. Its an highly optimized module and most of real time applications nowadays are using OpenCv. It supports on Python, C++ , Java e.t.c . Mainly to get the Graphical Interface so that we can capture an image or a frame from a video and to do lots of more jobs , OpenCv helps us to do this

There are lots of work which are done by the OpenCv are listed :

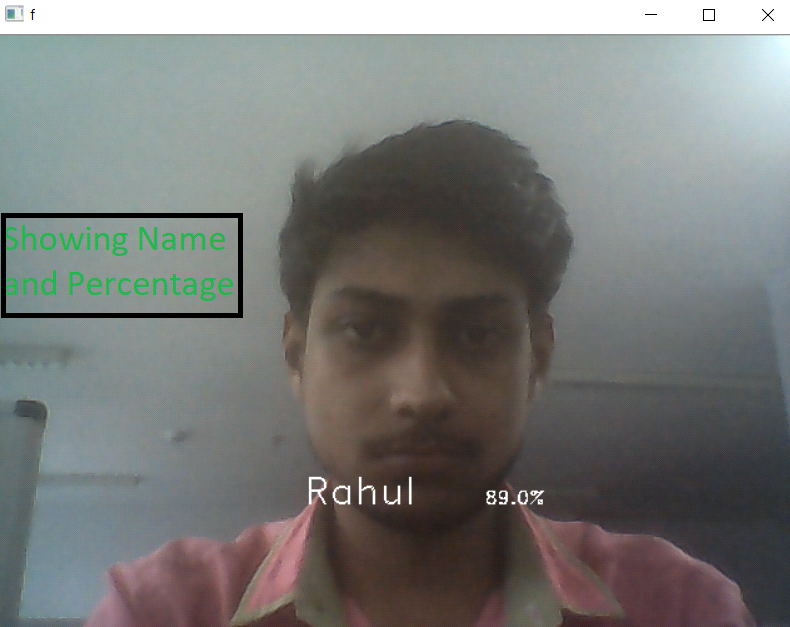
* Its **VideoCapture** method helps us to open a window which will capture a video.



* Its rectangle method helps us to draw a rectangle over the face.



* Using OpenCv we are putting text on the detected face and also put the percentage of the face matching if the face is stored on the database.Otherwise It will show Unknown .



**Math Module :** Math module basically use for a calculation. You see on the output that there is a percentage on one corner of the rectangle that is actually the matching between your current face with the stored face of yours . The encoding of the frame where the face is detected and the stored image , between the both only face portion encoding we will try to match and make a probability of that how much portion of that face is matched. This Job actually is done by Math Module.

**NumPy Module :** Numpy module is used here for basically to deal with multi-dimensional array . For a colored image when we will get the encondings of an image then the encoding stored into a n-D array that can’t be done without **numpy n-D array .** Because of that we import the numPy array.

**Project Code**

import cv2

import face\_recognition

import math

import numpy as np

video=cv2.VideoCapture(0) #Method to create video Capture Object.It #Will Trigger the Camera.

# 0 - to use build in camera,1,2... - to use external camera.

#VIdeoCapture(0)- it reads the first image/frame of the video.

# load a sample pic and learn how to recognize it.

shabnam\_image = face\_recognition.load\_image\_file("F:\\python project\\Known faces\\Shabnam.jpg")# load the jpg file numpy array

shabnam\_face\_encoding = face\_recognition.face\_encodings(shabnam\_image)[0]# list.

# face encoding of each face ,it return a list of encoding because a #image may contain more than one face.

#Histogram of Oriented Gradients

hasibul\_image = face\_recognition.load\_image\_file("F:\python project\Known faces\\Hasibul.jpg")

hasibul\_face\_encoding = face\_recognition.face\_encodings(hasibul\_image)[0]

rehan\_image = face\_recognition.load\_image\_file("F:\\python project\\Known faces\\Rehan.jpg")

rehan\_face\_encoding = face\_recognition.face\_encodings(rehan\_image)[0]

maruf\_image = face\_recognition.load\_image\_file("F:\\python project\\Known faces\\Maruf.jpg")

maruf\_face\_encoding = face\_recognition.face\_encodings(maruf\_image)[0]

Arisha\_image = face\_recognition.load\_image\_file("F:\\python project\\Known faces\\arisha.jpg")

Arisha\_face\_encoding = face\_recognition.face\_encodings(Arisha\_image)[0]

#create array of know faces and encodings

known\_face\_encodings = [

maruf\_face\_encoding,

rehan\_face\_encoding,

hasibul\_face\_encoding,

shabnam\_face\_encoding,

Arisha\_face\_encoding,

]

known\_face\_names = [

"maruf",

"rehan",

"Hasibul",

"Shabnam",

"Arisha",

]

while True:

# grab a single frame of video.

face\_names=[]

lval=[]

rit,frame=video.read()

#rit- boolean type object to detect if python is able to read the # #videoCapture object.(true / False)

#frame- numpy array ...represent the first image that the video #Captures.

small\_frame = cv2.resize(frame, (0, 0), fx=1, fy=1)

rgb\_small\_frame=small\_frame[:,:,:: -1]

#rezise frame for faster face recogniton processing.

face\_locations=face\_recognition.face\_locations(rgb\_small\_frame)

#Find all the faces in the image using the default HOG-based model.

face\_encodings = face\_recognition.face\_encodings(rgb\_small\_frame, face\_locations)

# find the faces and face enoding in the current current frame of the #video

for face\_encoding in face\_encodings:

matches = face\_recognition.compare\_faces(known\_face\_encodings, face\_encoding)

# checked for match with known faces .

#matches is a array of true /false

if True in matches:

first\_match\_index = matches.index(True)

name = known\_face\_names[first\_match\_index]

#calculating the face\_distances.

face\_distances = face\_recognition.face\_distance(known\_face\_encodings, face\_encoding)

face\_match\_threshold=0.6

for face\_distance in face\_distances:

if face\_distance < face\_match\_threshold:

range = face\_match\_threshold

linear\_val = 1.0 - (face\_distance / (range \* 2.0))

l=round((linear\_val + ((1.0 - linear\_val) \* math.pow((linear\_val - 0.5) \* 2, 0.2)))\*100)

l=str(l)

else:

name="Unknown"

l="Null

lval.append(l)

face\_names.append(name)

# display the result

for (top, right, bottom, left), name,l in zip(face\_locations, face\_names,lval):

top \*= 4

right \*= 4

bottom \*= 4

left \*= 4

#draw a box

cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

#put name of the face and percentage

font = cv2.FONT\_HERSHEY\_DUPLEX

cv2.putText(frame, name, (left + 6, bottom - 6), font, 0.50, (255, 255, 255), 1)

cv2.putText(frame, l+"%", (left + 80, bottom - 6), font, 0.50, (255, 255, 255)

cv2.imshow("capture",frame)# captures the first frame/image of the #video.

key=cv2.waitKey(1) # generate a new frame after every 1 milliseconds.

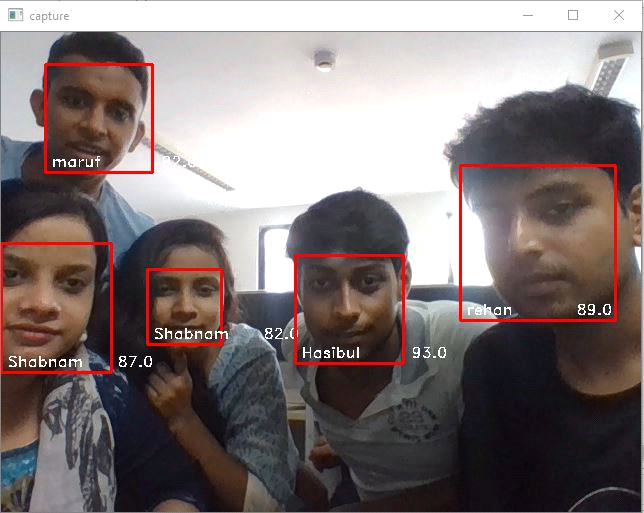
if key == ord("q"): # once we enter 'q' window will be destroyed

break

video.release()# release the camera in some milliseconds.

cv2.destroyAllWindows()# destroy all the windows according to waitKey #parameter.

**Project Output**



***------------------The End----------------***