DELHI TECHNOLOGICAL UNIVERSITY

PROJECT REPORT

TOPIC

FACE RECOGNITION SYSTEM.

(USING K-NN ALGORITHM)

CWS PROJECT

ALGORITHM DESIGN & ANALYSIS
(IT-208)

SUBMITTED BY :-

ISHAAN JAGGI (2K19-IT-062)

UNDER THE GUIDANCE OF – MR. RAHUL GUPTA

DEPARTMENT OF INFORMATION TECHNOLOGY
(DTU)

CONTENTS :-

DEPARTMENT OF INFORMATION TECHNOLOGY
(DTU)

CERTIFICATE

This is to certify that ***Ishaan Jaggi*** (2K19-IT-062) ; BTech [IT] ; Second year student of ***Delhi Technological University*** has done a Project work (MTE) on the **Topic** : *FACE RECOGNITION SYSTEM USING **K-NN ALGORITHM*** under the guidance of ***Mr. Rahul Gupta*** (Asst. Prof.) during the time period of ***Jan-May 2021***.

Regards :-

Mr. Rahul Gupta
(Asst. Prof.)

ACKNOWLEDGMENT

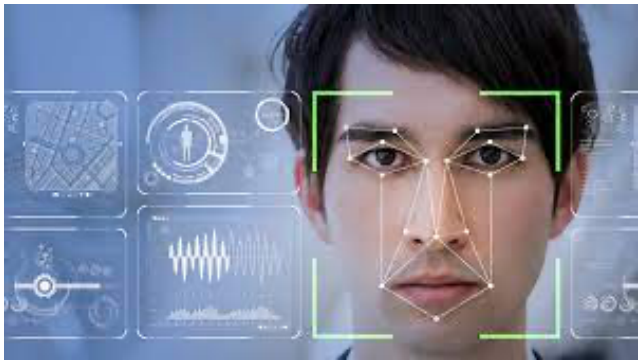
I would like to convey my heartfelt thanks to our supervisor, **Mr. Rahul Gupta** for his ingenious ideas, tremendous help and cooperation. I am extremely grateful to my friends who gave valuable suggestions and guidance for completion of my project.

The cooperation and healthy criticism came handy and useful with them. Finally, I would like to thank all of the above mentioned people once again.

Ishaan Jaggi
(2K19-IT-062)

ABSTRACT

Face Recognition is becoming a new trend in the ***security authentication systems.*** Modern FR systems can even detect, if the person is real (live) or not while doing face recognition, preventing the systems being hacked by showing the picture of a real person. All these successful face recognition systems are the results of recent advancements in the field of computer vision, which is backed by powerful deep learning algorithms. Let us explore one of such algorithms and see how we can implement a real time face recognition system using ***KNN algorithm***.



Face recognition can be done in ***two*** ways. Imagine you are building a face recognition system for an enterprise. One way of doing this is by training a neural network model, which can classify faces accurately. As you know for a classifier to be trained well, it needs millions of input data. Collecting that many images of employees, is not feasible. So this method seldom works. The best way of solving this problem is by opting one-shot learning technique. One-shot learning aims to learn information about object categories from one, or only a few, training images. The model still needs to be trained on millions of data, but the dataset can be any, but of the same domain. The pre-trained model that we are going to use in ***Haarcascade with Opencv***.

INTRODUCTION

Machine Learning is a subfield of Artificial Intelligence. The patterns in headway of AI relates its prosperity to the calculation abilities of the current CPUs and GPUs which can run enormous datasets and give us results. The advanced mobile phones of today, without an exemption have a selfie camera, which for the most part has a **rectangular** limit around the face. The selfie camera has certain highlights, similar to it might show the age of the individual, the gender of the individual. The precision of the highlights showed by the selfie camera depends on a few components.

Like light, point, excellence mode empowered and so forth The advanced cell industry, utilizes a pre-trained model to perceive the gender and age of the individual. The pre-prepared model employments profound figuring out how to distinguish highlights utilizing which ***it predicts the gender and age of the individual in view of recognizing and breaking down facial highlights***.

Biometric go up against affirmation, additionally called Automatic Face Recognition (AFR), is a particularly engaging biometric approach, since it revolves around a comparable identifier that individuals use essentially to remember one individual from another: their "faces". One of its essential destinations is the understanding of the baffling human visual system what's more, the learning of how individuals address faces with the ultimate objective to isolate assorted characters with **high accuracy**.

PROBLEM STATEMENT

The ***Problem Statement*** is to perceive the face that is inserted into the model by means of webcam. The picture will be inserted to the pre-prepared model through a webcam. This endeavour was done with this mind boggling "Open Source Computer Vision Library", the ***OpenCV***.

On this instructional exercise, we will focus on ***Python***. OpenCV was expected for computational capability and with a strong focus on persistent applications. Along these lines, it's ideal for progressing face affirmation using a camera. *We need to make this framework by utilizing* ***K-Nearest Neighbour Algorithm***.

SUPERVISED v/s UN-SUPERVISED LEARNING

1. A ***Supervised machine learning*** algorithm is one that relies on labelled input data to learn a function that produces an appropriate output when given new unlabelled or new data.

2. *Supervised learning* allows collecting data and produce data output from ***previous*** experiences. It helps to optimize performance criteria with the help of experience. Supervised machine learning helps to solve various types of real-world computation problems.

3. **Disadvantage** : Training for supervised learning needs a lot of computation time. So, it consumes a lot of time.

4. ***Example*** - Imagine a computer is a child, we are its supervisor (e.g. parent, guardian, or teacher), and we want the child (computer) to learn what a Lion looks like. We will show the child several different pictures, some of which are lions and the rest could be pictures of anything (cats, dogs, etc**). When we see a lion**, we shout "lion!" When it's not a lion, we shout "no, not lion!" After doing this several times with the child, we show them a picture and ask "lion?" and they will correctly (most of the time) say "lion!" or "no, not lion!" depending on what the picture is. ***That is Supervised machine learning***.

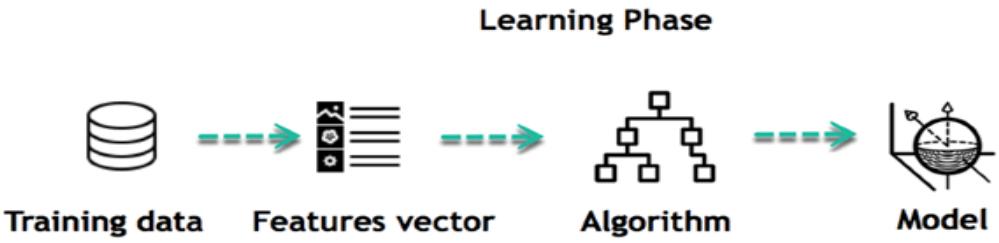
5. ***Un-Supervised Learning*** is an AI procedure where the clients **don't have to oversee** the model. All things considered, it permits the model to deal with its own to find examples and data that was already undetected. It essentially manages the unlabelled information.

6. Unsupervised learning algorithms are used to group cases based on similar attributes, or naturally occurring trends, patterns, or relationships in the data. These models also are referred to as self-organizing maps. Unsupervised models include ***clustering techniques*** and self-organizing maps.

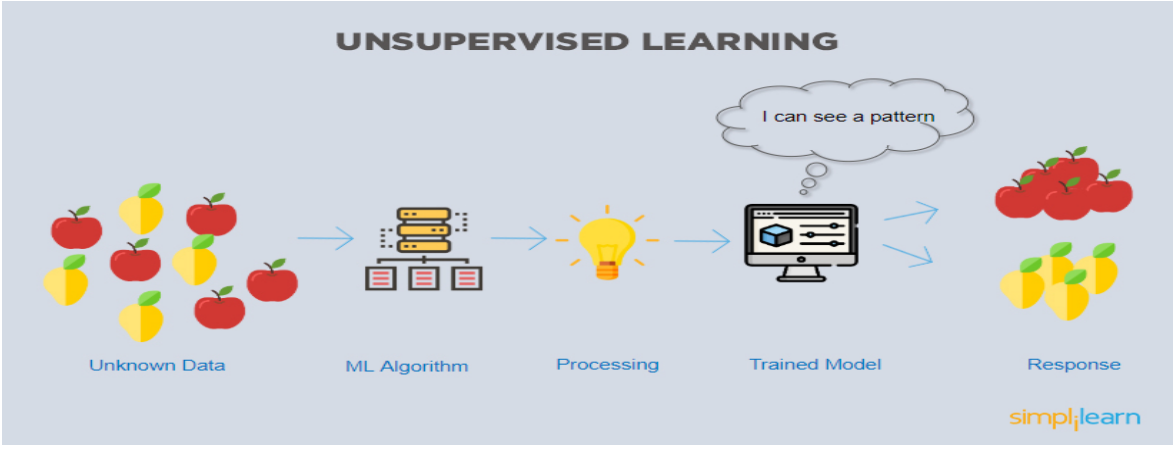
7. ***Advantage*** - ***Unsupervised learning*** is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labelled input data.

DIFFERENCE B/W SUPERVISED & UN-SUPERVISED LEARNING :-

SUPERVISED :-



UNSUPERVISED :-



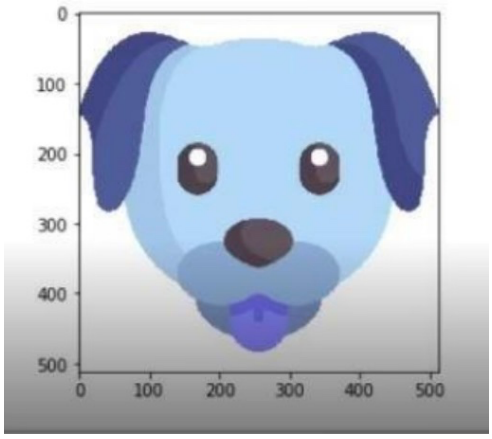
ABOUT OPENCV

OPENCV is a ***library*** is helpful while we need to work with pictures. It is dispatched as ***cv2***. Essentially we'll utilize this library to peruse and compose pictures and to include a video transfer moreover. Thus, we should perceive how we can peruse a picture utilizing Opencv.

How Opencv Reads an Image :-

```
import cv2
Img=cv2.imread("OcBasics/dog.png")
 from matplotlib.importpyplot as plt
plt.imshow(img)
```

OPENCV EXAMPLE :-

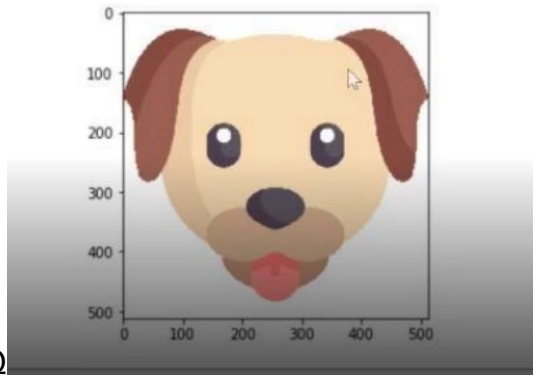


BGR Image of Dog

When opencv peruses a picture, **it is a RGB picture**. Hence, ***each picture has 3 channels***. Thus, when we read a picture utilizing opencv, it naturally peruses these channels as **BGR**. It feels that the main channel is the blue channels. It occurs of course and it attempts to consider the to be as BGR. ***Fundamentally, the red and the blue channels get traded. Any place there was red you can see blue. Thus, we need to switch between the shading spaces***. Opencv gives us one strategy, utilizing which we can see switch between two shading spaces. We need to change from BGR to RGB colourspace.

```
import cv2
img=cv2.imread("OcBasics/dog.png")
newImg=cv2.cvtcolor(img, cv2.COLOR_BGR2RGB)
```

NOTE - The **BOLD** part defines the constant. It is a constant in the library, mapped to some integer when opencv sees that conversion we've chosen. This is a notation to denote a conversion. It is a constant that tells opencv, that what type of conversion we want to do.



RGB Image of Dog

Finallly, we have converted our image into a new image, **that is from BGR to RGB.**

METHOD TO READ IMAGES :-

We will use opencv to read and display images

```
import cv2
img=cv2.imread("dog.png")
cv2.imshow("Dog Image",img)
cv2.waitkey(0)
cv2.destroyAllWindows()
```

NOTE - ***waitkey(0)*** signifies that after waiting for infinite time the terminal, will be exited.

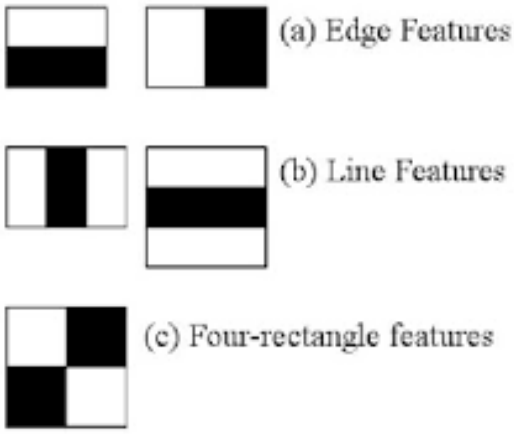
ABOUT HaarCascade CLASSIFIER

1. It is an **AI based approach** where a course work is set up from a significant proportion of positive and negative pictures. It is then used to perceive inquiries in various pictures.

2. It is a pre-prepared model on a ton of ***facial information*** and it is utilized to arrange faces.

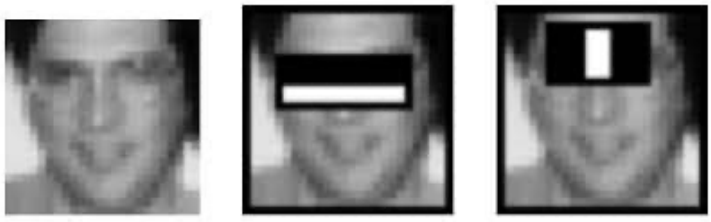
3. It was made by ***Rainer Leinhart***. It utilizes adaboost to identify if a given article is a face. It observes the most pertinent ***highlights like eyes, nose, lips, eyebrows*** and so on.

4. **Features** :-



5. Here we will chip away at face ID. From the outset, the estimation needs an impressive proportion of positive (pictures of appearances) and negative (pictures without faces) to set up the classifier. By then we need to eliminate features from it. For this, ***Haar features*** showed up in underneath picture are used. They are similarly as our convolutional piece. **Every part is a lone regard procured by taking away all out of pixels under white square shape from total of pixels under dull square-shape**. The little squares and square shapes emphasise throughout the image to discover the highlights for co-ordinating and perceiving the picture.

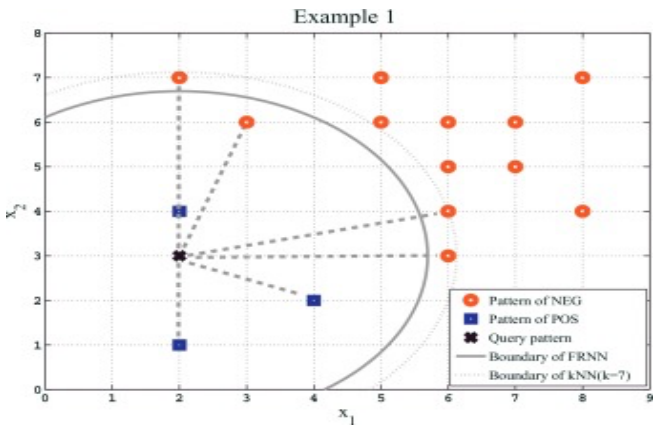
6. ***Real-life example*** :-



ALGORITHM USED IN PROJECT

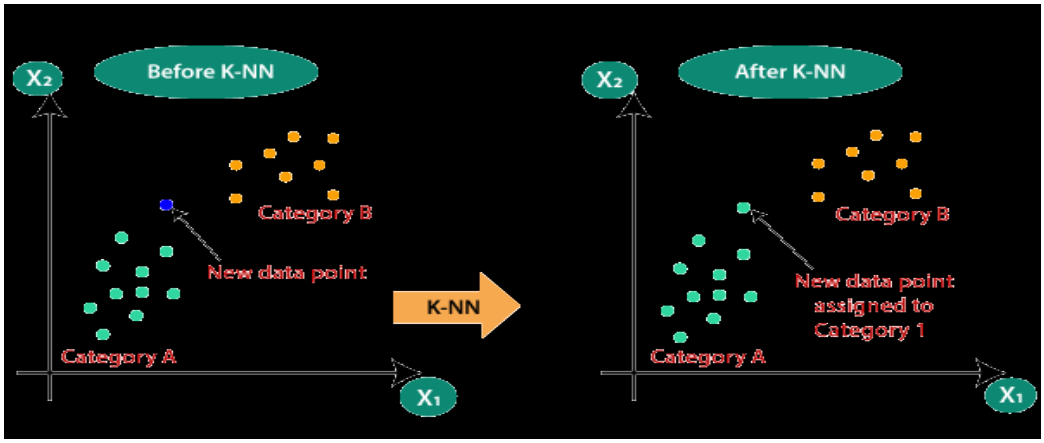
K-NN ALGORITHM IS USED IN THIS PROJECT.

- ✓ K-Nearest Neighbour (K-NN) is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- ✓ K-NN is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). It has been used in statistical estimation and pattern recognition .
- ✓ K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- ✓ It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

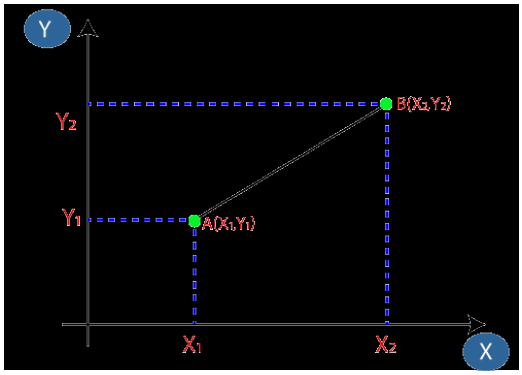


HOW K-NN ALGORITHM WORKS

- ***Step-1*:** Select the number 'K', where 'K' is the no. of neighbours of the new data.
- ***Step-2***: Calculate the ***Euclidean distance*** of K number of neighbours.
- ***Step-3***: Take the K nearest neighbours as per the calculated Euclidean distance.
- ***Step-4***: Among these K neighbours, count the number of the data points in each category.



- ***Step-5***: Assign the new data points to that category for which the number of neighbour is maximum.
- ***Step-6***: *The new data point got its category and our model is ready.*
- ***Step-7:*** Euclidean Distance = [(X2-X1)^2] + [(Y2-Y1)^2].



APPLICATIONS OF K-NN ALGORITHM

- KNN can be used for **Recommendation Systems**. Although in the real world, more sophisticated algorithms are used for the recommendation system. KNN is not suitable for high dimensional data, but it is an excellent baseline approach for the systems. Many companies make a personalized recommendation for its consumers, such as Netflix, Amazon, YouTube, and many more.

- KNN can search for ***semantically similar documents***. Each document is considered as a vector. If documents are close to each other, that means the documents contain identical topics.

- KNN can be effectively used in ***detecting outliers***. One such example is Credit Card fraud detection.

WORKING OF PROJECT

Producing Training Data: The accompanying advances are followed to create preparing information :-

A. Write a Python Script that catches pictures from your webcam video transfer

B. Extracts all Faces from the picture outline (utilizing haar files)

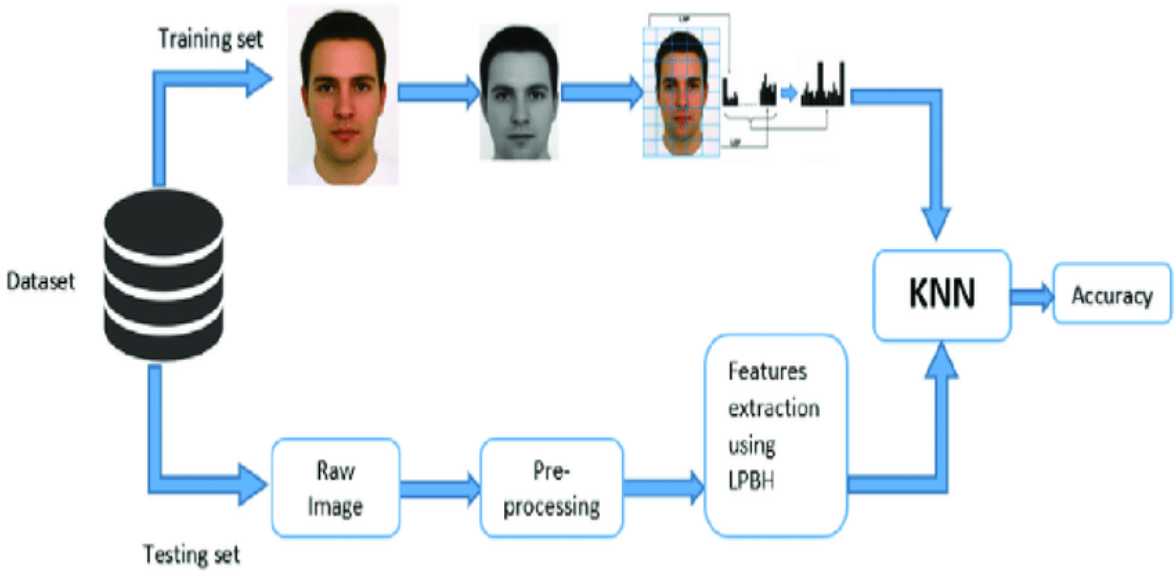
C. Stores the Face data into numpy exhibits ,

1. Peruse and show video transfer, catch pictures.

2. Identify Faces and show bouncing box (haar course) .

3. Straighten the biggest face image(gray scale) and save in a numpy exhibit.

4. Rehash the above for various individuals to create preparing information.



Building The Face Classifier - Perceive Faces utilizing the arrangement calculation — ***KNN*** :-

A. load the preparation information (Numpy varieties of the multitude of people).

   x- values are put away in the Numpy clusters.

   y- values we need to allot for every individual.

B. Peruse a video transfer utilizing Opencv .
C. remove faces out of it .
D. use ***K-NN*** to discover the expectation of face (int) .
E. map the anticipated id to name of the client.
F. Show the expectations on the screen — jumping box and name.

PROJECT CODE (PYTHON)

✓ FOR TRAINING DATA :-

```
import cv2
import os
import numpy as np

cap = cv2.VideoCapture(0)
classifier =
cv2.CascadeClassifier("C:\\Users\\jaggi\\OneDrive\\Documents\\Face_D
etection-master-20201029T135414Z-001\\Face_Detection-
master\\haar.xml")  # xml with face proportions

name = input("Enter your name: ")
count = int(input("No. of pics: "))
images = []

while True:
    retval, image = cap.read()
    if retval:
        faces = classifier.detectMultiScale(image)
        cv2.imshow("cropped", image)
        if len(faces) > 0:
            sorted_faces = sorted(faces, key=lambda item: item[2] *
item[3])

            face1 = sorted_faces[-1]

            x, y, w, h = face1

            cut_1 = image[y:y + h+200, x:x + w+200]
            cut_1 = cv2.resize(cut_1, (100, 100))
            cv2.imshow("cropped", cut_1)

    key = cv2.waitKey(1)

    if key == ord('q'):
        break

    if key == ord('c'):
        while count >= 0:
            img = cv2.cvtColor(cut_1, cv2.COLOR_BGR2GRAY)
            images.append(img.flatten())
            count -= 1
            print(count)
```

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

X = np.array(images)
y = np.full((X.shape[0], 1), name)

data = np.hstack([y, X])  # first column has name of whose image is
in the other rest of the columns (just like in
# knn) 784 features of image

if os.path.exists("faces.npy"):     # to update the file with new
image data.
    old = np.load("faces.npy")
    data = np.vstack([old, data])

np.save("faces.npy", data)
```

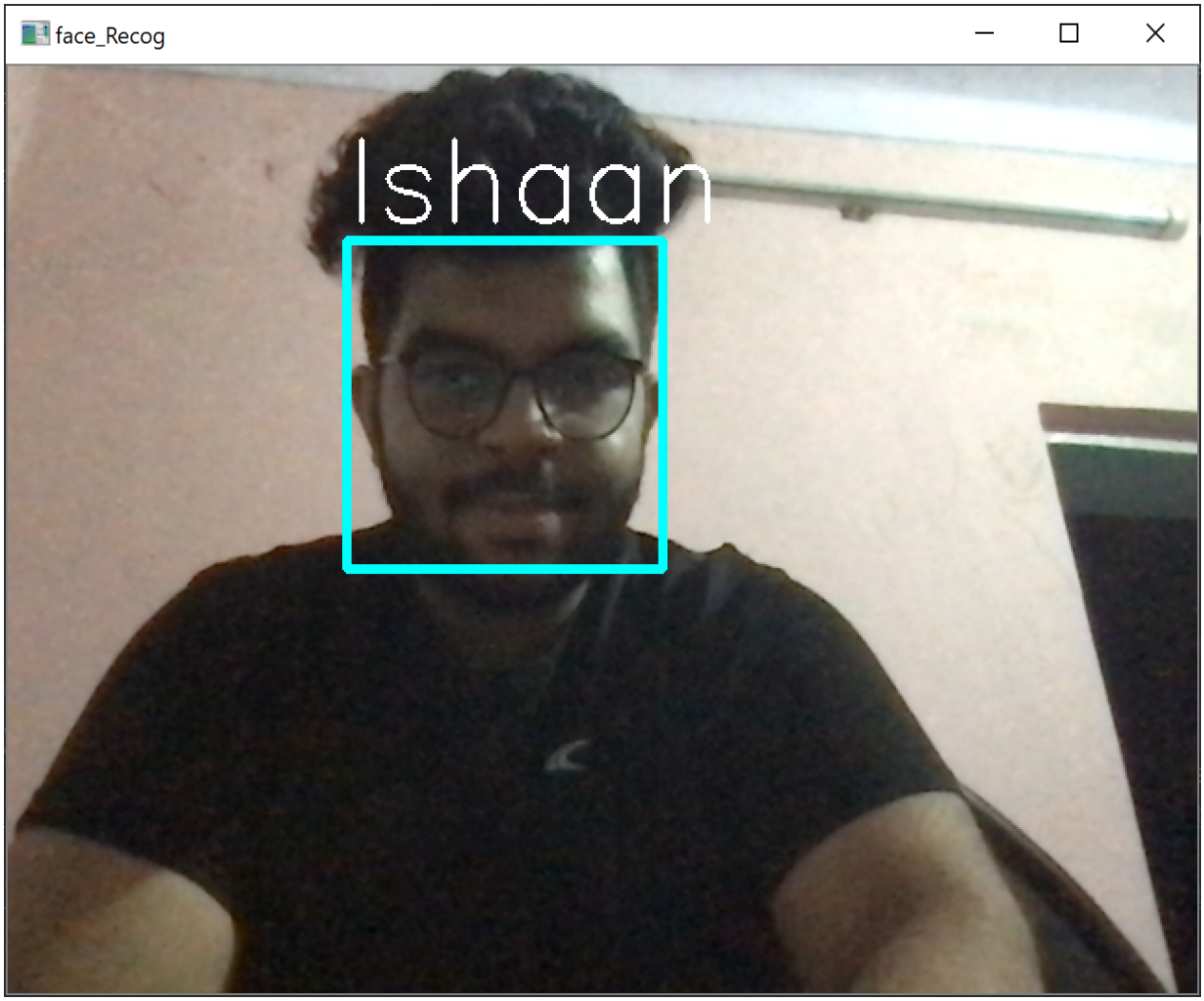
✓ **FOR TESTING DATA** :-

```
import cv2
import numpy as np
import os

from sklearn.neighbors import KNeighborsClassifier

cap = cv2.VideoCapture(0)
classifier =
cv2.CascadeClassifier("C:\\Users\\jaggi\\OneDrive\\Documents\\Face_D
etection-master-20201029T135414Z-001\\Face_Detection-
master\\haar.xml")  # xml with face proportions
data = np.load("faces.npy")

X = data[:, 1:]
y = data[:, 0]

model = KNeighborsClassifier()
model.fit(X, y)
```

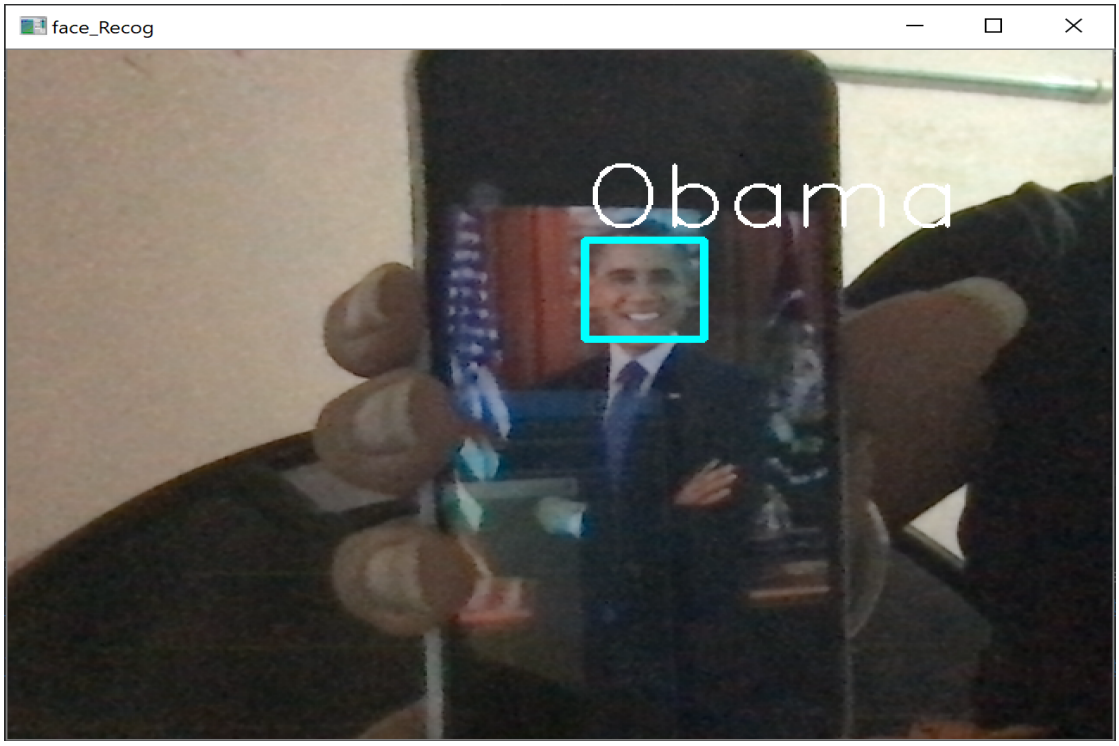
```
while True:
    retval, image = cap.read()
    if retval:
        faces = classifier.detectMultiScale(image)
        if len(faces)>0:
            for face in faces:
                x ,y ,w ,h = face

                cut = image[y:y + h, x:x + w]

                resized = cv2.resize(cut, (100, 100))

                y_test = cv2.cvtColor(resized, cv2.COLOR_BGR2GRAY) #
because the face_detect_train has trained photos
                # with gray scale
                y_test = y_test.flatten()

                output = model.predict([y_test])[0] # list of
outputs are there so only one has to be the title of
                # rectangle

                cv2.rectangle(image, (x, y), (x+w, y+h), (255, 255,
0), 3)
                cv2.putText(image, str(output), (x, y - 10),
cv2.FONT_HERSHEY_SIMPLEX, 2, (255, 255, 255), 2)

        cv2.imshow("face_Recog", image)

        key = cv2.waitKey(1)

        if key == ord('q'):
            break
cap.release()

cv2.destroyWindow()
```

RESULTS & PERFORMANCE ANALYSIS

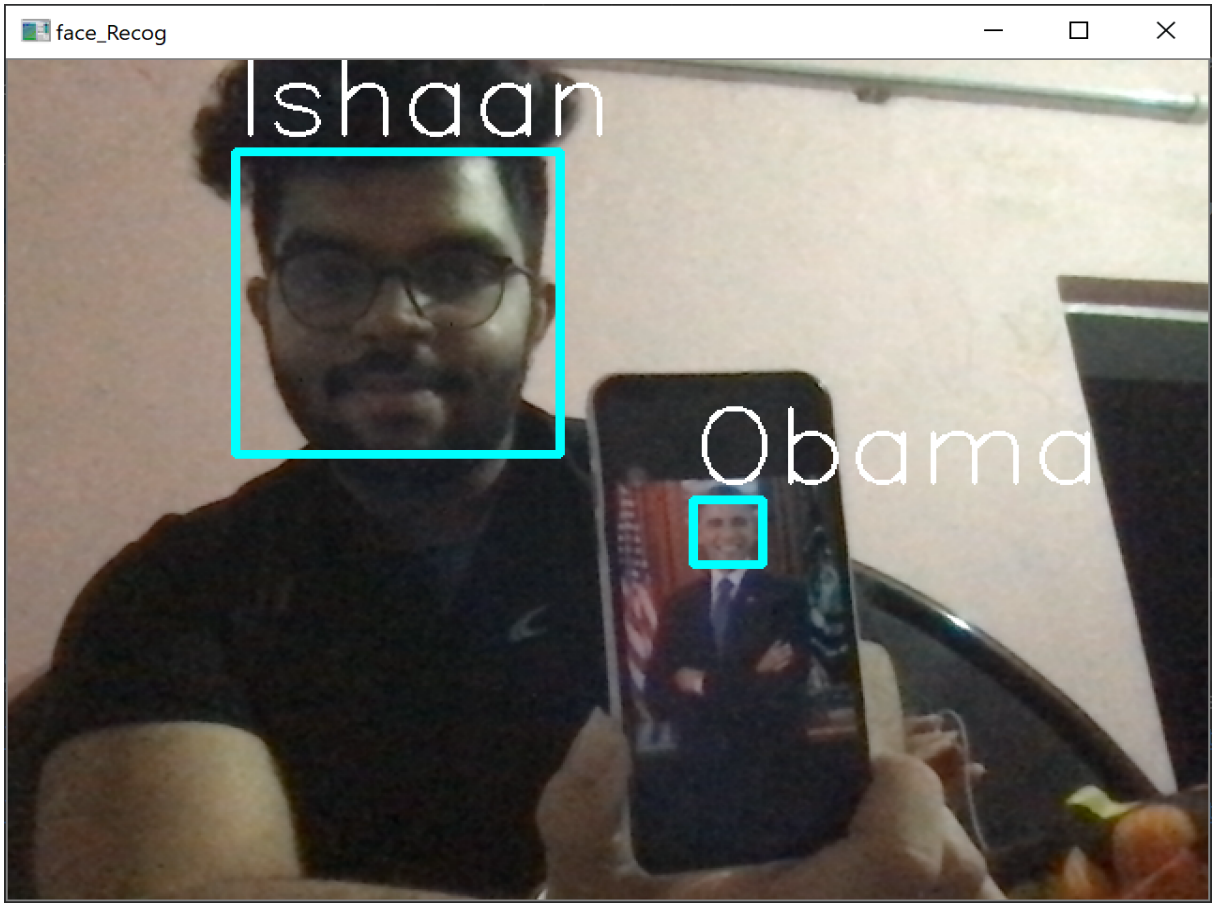
1. ***SINGLE FACE*** :-



|
| (OBAMA'S FACE)
\/



2. ***MULTIPLE FACES*** :-



CONCLUSION

My project is on ***Face Recognition System Using K-NN Algorithm***. I successfully completed it. I take this opportunity to express our sense of indebtedness and gratitude to all those people who helped me in completing this project. I am immensely grateful to our esteemed project-guide ***Mr. Rahul Gupta*** (Asst. Prof.) and other faculties for their supervision and guidance without which this work would not have been possible. This project has contributed a lot to our knowledge that has proved to be a value addition for us.

After preparing and testing the ***Face Recognition System***, we **conclude** that by using this System arrangement we can ***use these systems as a security purpose in our devices such as smartphones, Laptops etc*** so that people can securely use their devices & if by-chance their device lost somewhere So, any stranger person cannot be able to unlock the device & misuse it.

We can also prepare this system to fix it in many photo vaults & important files vault. By using this system we ***can solve*** Real-World problems in an efficient manner.

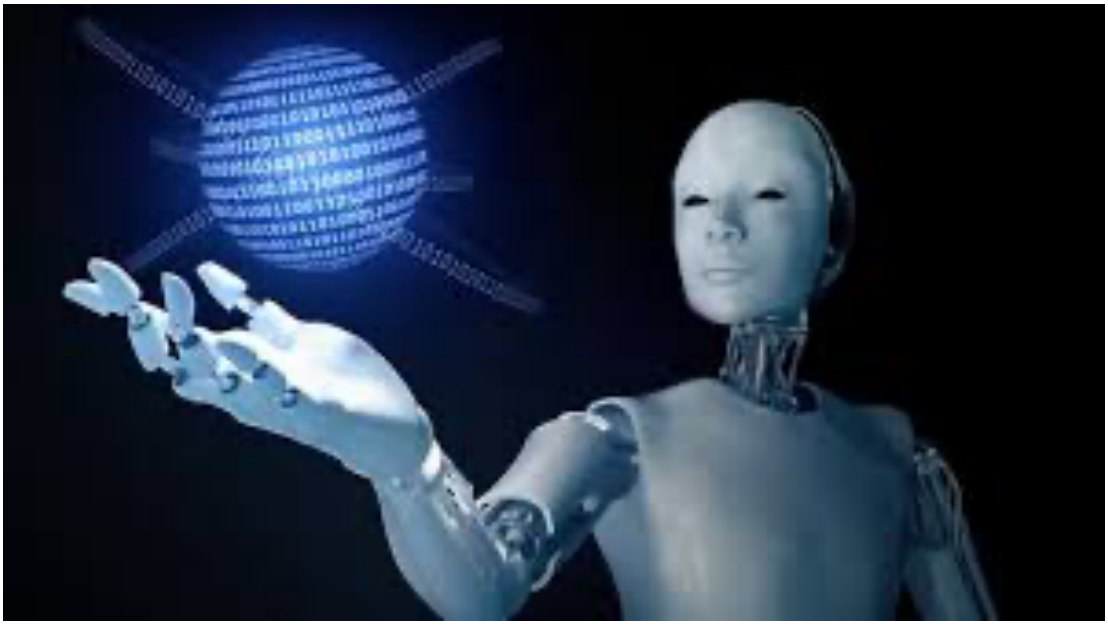
NOTE – The ***Time*** & ***Space Complexity*** of this System (using K-NN) is :-

1. ***O(k*d*n)*** → where,
   k = no. of neighbours considered.
   d = Data Dimensionality.
   n = no. of points in training dataset.

2. ***O(1)*** → Space Complexity.

FUTURE SCOPE

- The eventual fate of facial acknowledgment innovation is brilliant. ***Forecasters think that this innovation*** is required to develop at an imposing rate and will produce enormous incomes in the coming years. ***Security and observations*** are the significant sections which will be profoundly impacted.



- Different regions that are presently greeting it wholeheartedly are private enterprises, public structures, and ***schools***. It is assessed that it will likewise be embraced by retailers and banking frameworks in coming a long time to keep misrepresentation in charge/Mastercard buys and installment particularly the ones that are on the web.

- This innovation would fill in the provisos of to a great extent common insufficient secret key framework. Over the long haul, **robots utilizing facial acknowledgment** innovation may likewise come to raid. They can be useful in finishing the assignments that are unreasonable or hard for people to finish.

BIBLIOGRAPHY

- ✓ https://www.techsciresearch.com/blog/future-of-facial-recognition-technology/105.html
- ✓ https://medium.com/analytics-vidhya/face-recognition-using-knn-open-cv-9376e7517c9f
- ✓ https://www.researchgate.net/figure/Classical-recognition-system-using-the-KNN-classifier-method_fig7_324421394
- ✓ https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/
- ✓ https://en.wikipedia.org/wiki/OpenCV
- ✓ https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html
- ✓ https://sebastianraschka.com/pdf/lecture-notes/stat479fs18/02_knn_notes.pdf
- ✓ https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

(THE END)