



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Name:- Sanket Bhostekar

Roll No - 40

Batch C

Aim: To perform face detection on video.

Objective:

1. Performing face recognition.
2. Generating the data for face recognition.
3. Recognizing faces.
4. Preparing the training data.
5. Loading the data and recognizing faces.

Theory:

Generating the Data for Face Recognition:

Face recognition relies on having a dataset of faces to learn from. This dataset typically consists of images or video frames containing faces of individuals. To generate the data for face recognition, we capture video frames or collect images of individuals under various lighting conditions, angles, and expressions. This diverse dataset helps improve the accuracy of the recognition system.

Recognizing Faces:

Face recognition is the process of identifying and verifying individuals by analyzing their facial features. This involves comparing the features extracted from an input face to those in a database. Various algorithms, such as Eigenfaces, LBPH (Local Binary Pattern Histograms), or deep learning-based approaches like CNNs (Convolutional Neural Networks), can be used for face recognition.

Preparing the Training Data:



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Before recognizing faces, it's crucial to preprocess and format the training data correctly. This includes resizing images, normalizing pixel values, and extracting relevant facial features. Additionally, labeling each face with the corresponding individual's identity is essential for supervised learning.

Loading the Data and Recognizing Faces:

To recognize faces in a video, we start by capturing frames from the video stream. These frames are then processed to detect and recognize faces using the trained recognition model. The model compares the features of the detected face with the features of the individuals in the training dataset to make a match or identify the person.

Code:

```
import numpy as np
import cv2
from google.colab.patches import cv2_imshow

cap = cv2.VideoCapture('/content/IPS.mp4')
face_cascade = cv2.CascadeClassifier('/content/haarcascade_frontalface_default.xml')
#eye_cascade = cv2.CascadeClassifier('/content/haarcascade_eye.xml')

while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 5)
        roi_gray = gray[y:y+w, x:x+w]
        roi_color = frame[y:y+h, x:x+w]
```



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

```
#eyes = eye_cascade.detectMultiScale(roi_gray, 1.3, 5)

#for (ex, ey, ew, eh) in eyes:

    #cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 5)

cv2_imshow(frame)

if cv2.waitKey(1) == ord('q'):

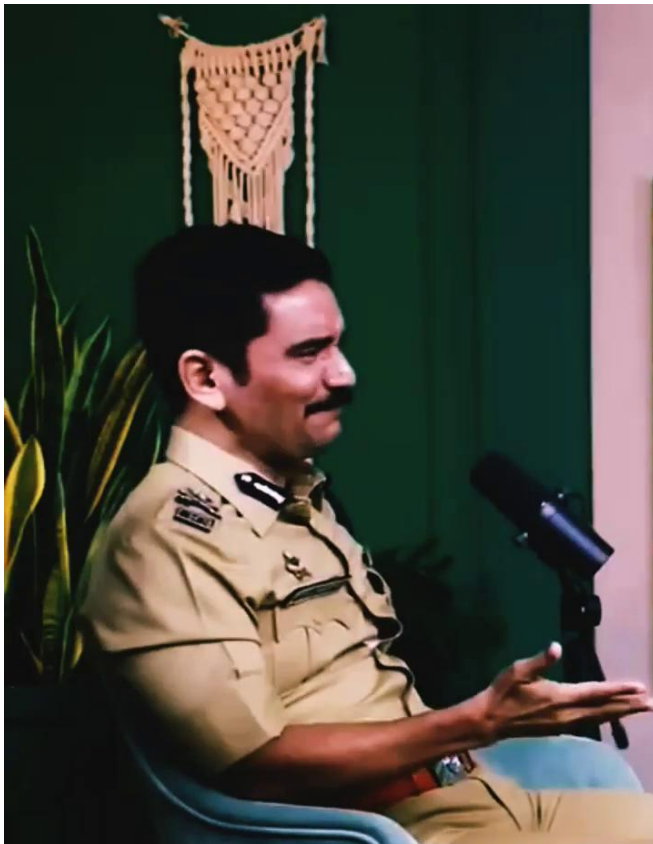
    break

cap.release()

cv2.destroyAllWindows()
```

Output:

Input Video -

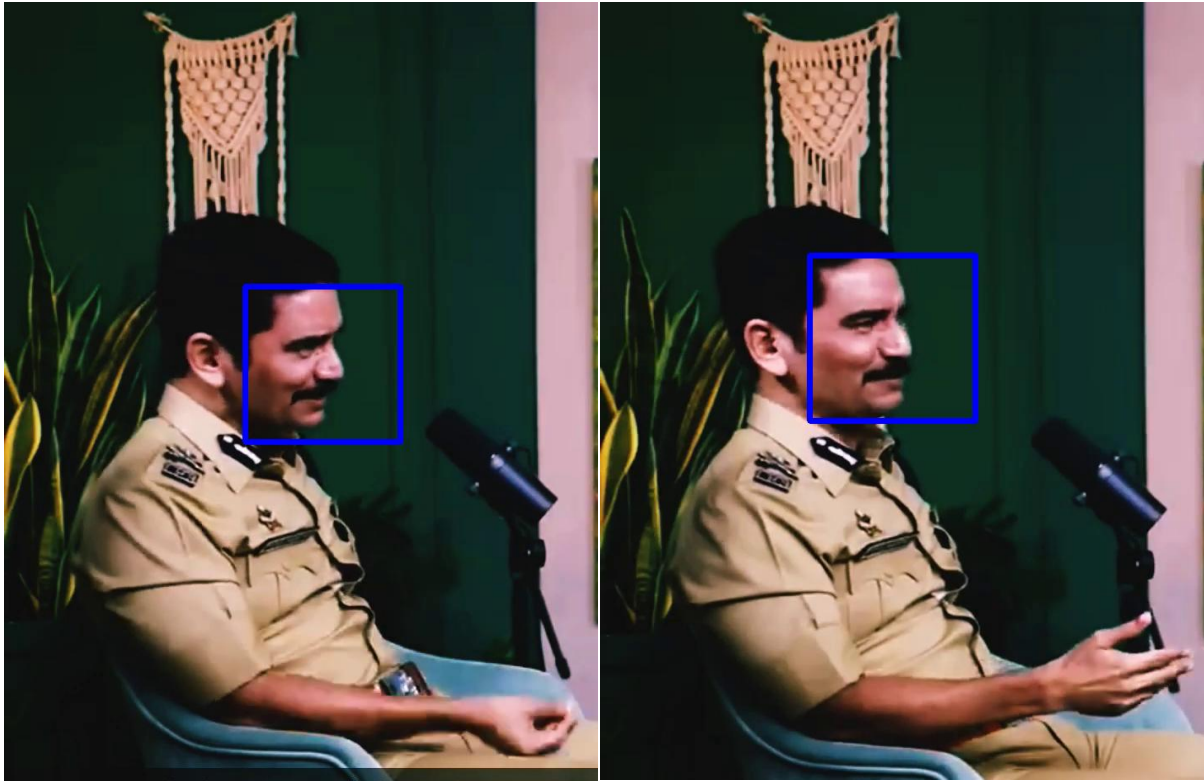




Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Output Video -



Conclusion:

This experiment delves into the practical aspects of face detection and recognition in videos. It highlights the importance of a well-structured dataset for training, various algorithms for recognition, and the critical steps involved in preprocessing and analyzing video frames. By the end of this experiment, we will gain valuable insights into the capabilities and limitations of face recognition systems in real-world applications.