Name: Divya Patil

Roll No: 08

Experiment No: 07

Aim: To perform Face detection on Video

Objective: Performing face recognition Generating the data for face recognition. Recognizing faces preparing the training data Loading the data and recognizing faces.

Theory:

1. Generating the data for face recognition:

Generating data for face recognition typically involves collecting a dataset of facial images from various sources, including photographs or video frames. This dataset should encompass a wide range of individuals, expressions, lighting conditions, and backgrounds to ensure robust model training. Data augmentation techniques such as rotation, scaling, and adding noise may also be applied to increase dataset diversity and improve model generalization.

2. Recognizing faces:

Recognizing faces refers to the automated process of identifying and distinguishing human faces within images or videos. This involves analyzing facial features, such as eyes, nose, and mouth, to match them with known individuals or categorize them based on attributes like age, gender, or emotions. Facial recognition technology is widely used for security, authentication, and personalization purposes, but it also raises privacy and ethical concerns.

3. Preparing the training data:

Preparing training data involves several key steps. First, data collection involves gathering a diverse set of images or videos relevant to the task. Data annotation then labels these images with ground truth information, like object bounding boxes or semantic segmentation masks. Data augmentation is used to artificially increase the dataset's size and diversity by applying transformations like rotation or scaling. Finally, data splitting divides the dataset into training, validation, and test sets to train and evaluate machine learning models effectively. These steps are crucial for building robust computer vision models.

4. Loading the data and recognizing faces:

Loading the data and recognizing faces involves the process of acquiring and preprocessing image or video data, followed by applying facial detection and recognition algorithms. This typically includes tasks like data ingestion, image resizing, feature extraction, and the use of deep learning models to identify and classify faces within the data, enabling applications like facial authentication, emotion detection, or person identification.

Code and Output:

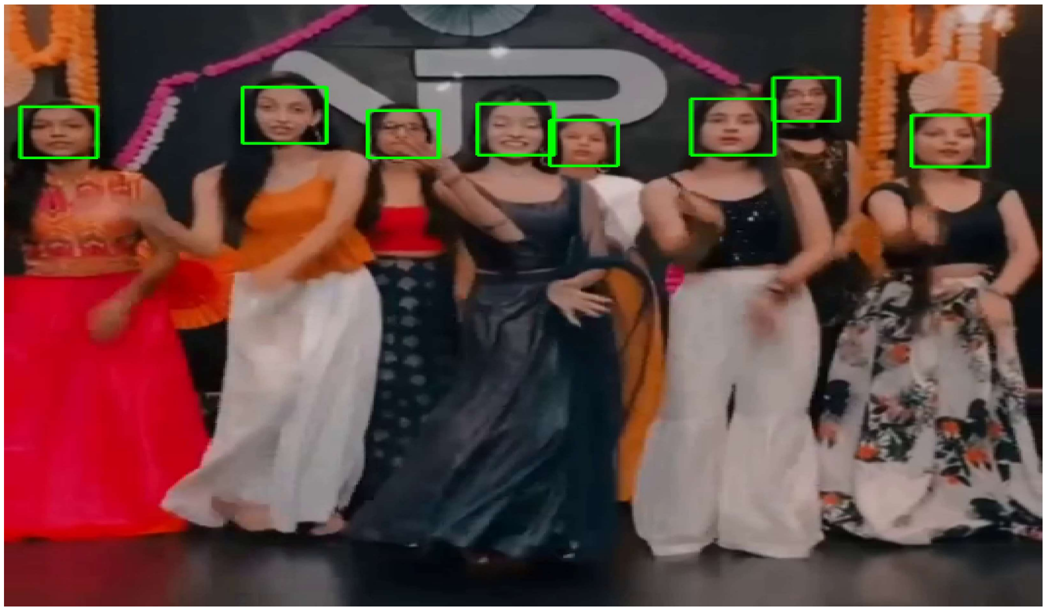
Input:

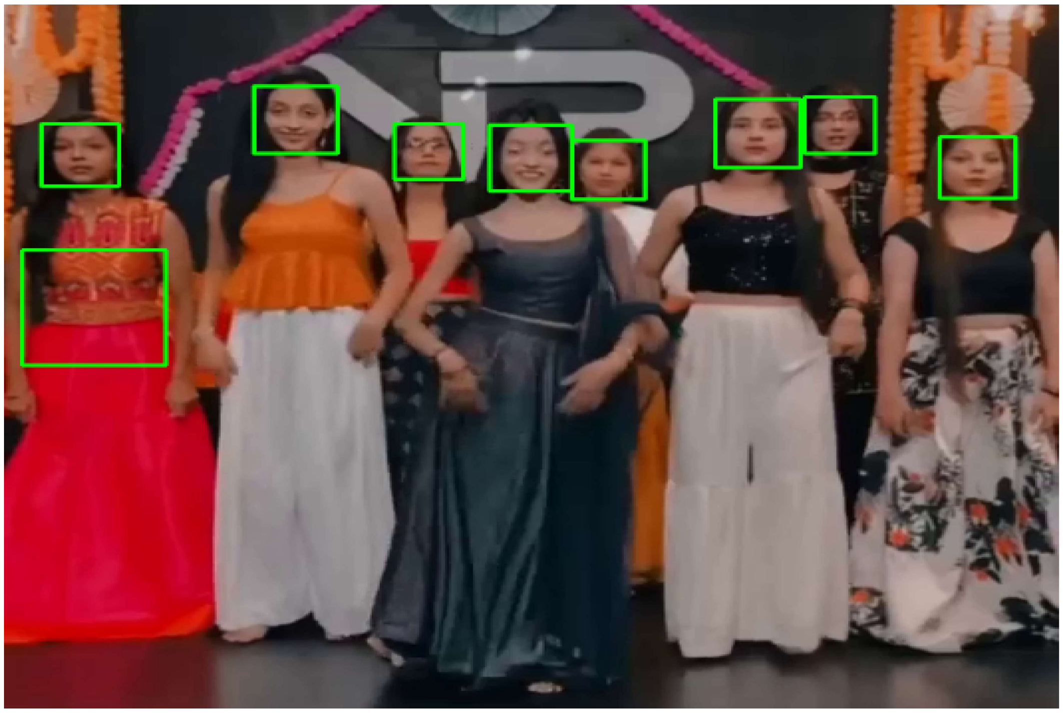
video.mp4

```
import cv2

import datetime

from google.colab.patches import cv2_imshow

face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')

cap = cv2.VideoCapture('/content/video.mp4')

while True:

 ret, frame = cap.read()

 if not ret:

  break

 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

 faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3,

minNeighbors=5, minSize=(30, 30))

 timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

 cv2.putText(frame, timestamp, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

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

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

 cv2_imshow(frame)

 if cv2.waitKey(1) & 0xFF == ord('q'):

  break

cap.release()

cv2.destroyAllWindows()
```

Output:





Conclusion :

The code successfully identifies and tracks faces within the video frames, Face detection in videos means finding and following faces as they appear in different frames, allowing us to track facial expressions and movements in real-time. Face recognition relies on having data about people's faces, helping us identify them accurately. Preparing the data for training is crucial to create strong recognition models. Loading and processing this data helps us recognize faces effectively, useful in various applications like security and tagging people on social media.