Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Name - Polomi Adak

Roll no - 20

Experiment: 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:

```
!pip install opencv-python

import cv2

import datetime

from google.colab.patches import cv2_imshow

face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +

'haarcascade_frontalface_default.xml')

video_path = ‘video.mp4’

cap = cv2.VideoCapture(video_path)


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
```

Input:

video.mp4 file

Output:

Conclusion :

Performing face detection on video involves identifying and tracking faces across multiple frames, enabling real-time analysis of facial movements and expressions. Face recognition relies on generating and organizing data about individuals, making it possible to match and identify faces accurately. Preparing training data is crucial for developing robust recognition models. Loading and processing this data allows for the efficient recognition of faces in various applications, from security systems to social media tagging. Together, these steps contribute to the development and deployment of effective facial recognition systems.