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 :

Generating the data for face recognition: It involves the following steps:

- Data Collection: Gather video footage with the faces to be recognized.
- Video Preprocessing: Extract frames, resize, standardize, and reduce noise.
- Face Detection: Use a model to identify and extract faces from frames.
- Face Alignment (Optional): Align faces to a standardized pose.
- Data Annotation: Label detected faces with identities.
- Data Augmentation (Optional): Apply data augmentation for diversity.
- Data Splitting: Divide data into training, validation, and test sets.
- Data Storage: Organize and store preprocessed data.
- Train a Model: Use deep learning to train a face recognition model.
- Model Evaluation: Assess the model's performance on a validation set.
- Testing and Deployment: Test and deploy the model.
- Inference on New Videos: Apply the model to recognize faces in new video data, considering privacy and ethical concerns.

Recognizing faces:involves the following key steps:

- Extract frames from the video.
- Detect faces in each frame using algorithms or models.
- Optionally, track faces across frames for continuity.
- Apply face recognition to identify individuals using known face databases.
- Set a similarity threshold for matching faces.
- Annotate and visualize recognized faces.
- Handle privacy and ethical considerations.
- Evaluate system performance.
- Optimize for real-time processing.
- Deploy the system in relevant applications while adhering to privacy and ethical standards.

Preparing the training data: involves these key steps:

- Collect diverse video footage with faces.
- Extract frames, standardize, and reduce noise.
- Annotate frames with face bounding boxes.
- Optionally, apply data augmentation for diversity.
- Split data into training, validation, and test sets.
- Organize and format data for training.
- Ensure balanced positive and negative examples.
- Review and ensure annotation accuracy.
- Normalize pixel values and perform model-specific preprocessing.
- Implement a data loading pipeline for model training.
- Optionally, apply data augmentation during training.
- Train the face detection model using the prepared data.

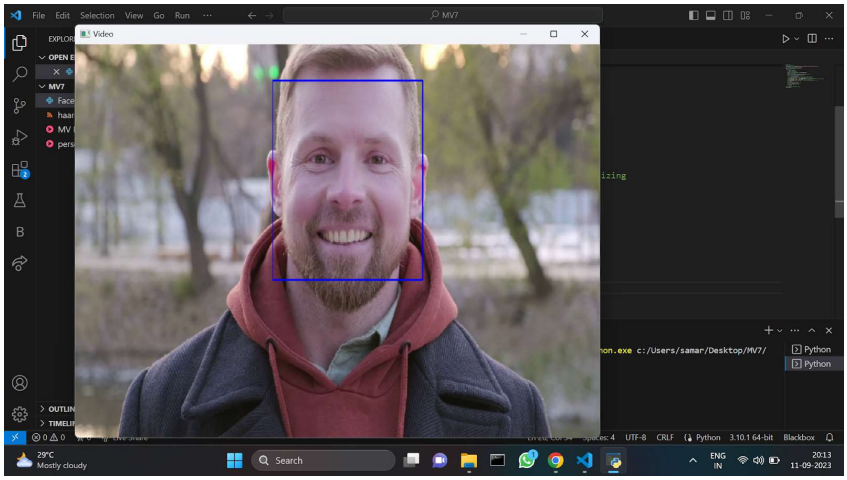
Loading the data and recognizing faces:

- Load the video and extract frames.
- Apply face detection to locate faces in frames.
- Use a trained face recognition model to identify faces.
- Compare detected faces to a database of known individuals.
- Apply a threshold to decide on face matches.
- Optionally, annotate and visualize recognized faces.
- Address privacy and ethical considerations.
- Monitor and optimize system performance.
- Deploy the system for real-time or specific applications.
- Implement post-processing for complex scenarios.

Code:-

```
import cv2
# Load the cascade
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
# To capture video from existing video.
cap = cv2.VideoCapture('person.mp4')
while True:
    # Read the frame
    _, img = cap.read()
    # Convert to grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # Detect the faces
    faces = face_cascade.detectMultiScale(gray, 1.1, 4)
    # Draw the rectangle around each face
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
    # Create a named window with a custom size and position
    cv2.namedWindow('Video', cv2.WINDOW_NORMAL)  # Use WINDOW_NORMAL to allow
resizing
    cv2.resizeWindow('Video', 800, 600)  # Set the window size
    cv2.moveWindow('Video', 100, 30)    # Set the window position (x, y)
    # Display
    cv2.imshow('Video', img)
    # Stop if escape key is pressed
    k = cv2.waitKey(30) & 0xff
    if k==27:
        break
# Release the VideoCapture object
cap.release()
```

Output:-



https://youtu.be/wybCtrWOrtg

Conclusion:-

Face detection in a video involves locating and identifying faces within video frames. This process includes extracting frames, applying face detection algorithms or models, and potentially utilizing face recognition for identity verification. Preparing a diverse training dataset is critical, as it ensures the model's accuracy. Addressing privacy and ethical concerns, monitoring system performance, and optimizing for real-time deployment are important considerations. Ultimately, face detection in videos is vital for applications such as surveillance and access control, with ongoing advancements in deep learning and computer vision techniques enhancing its capabilities.