Experiment No.7
To perform Face detection on Video
Date of Performance:
Date of Submission:

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.

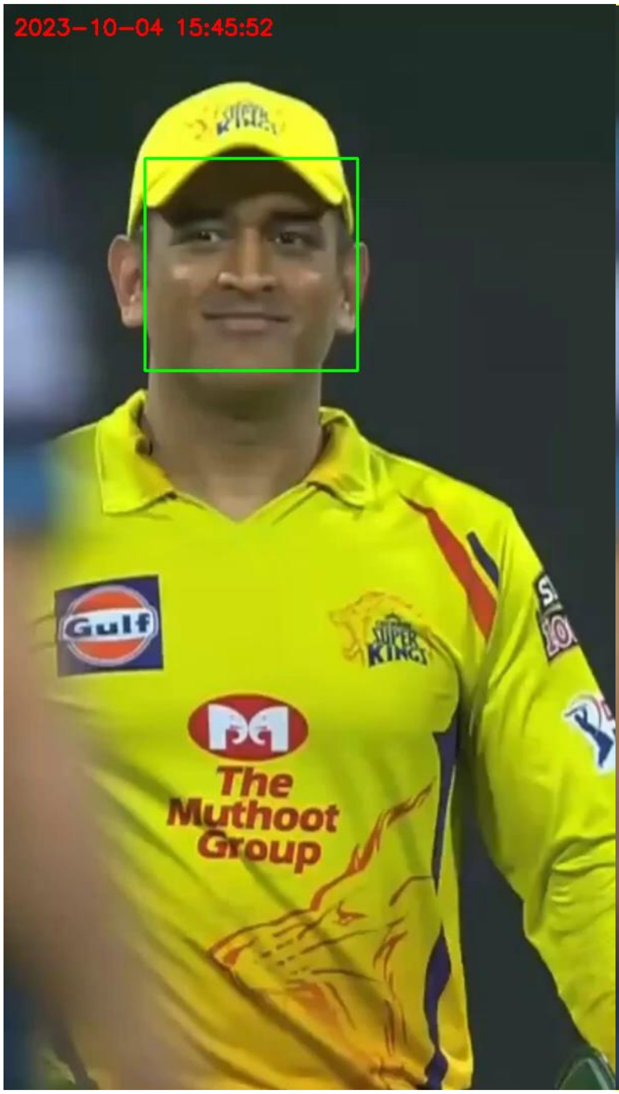
Loading the data and recognizing faces:

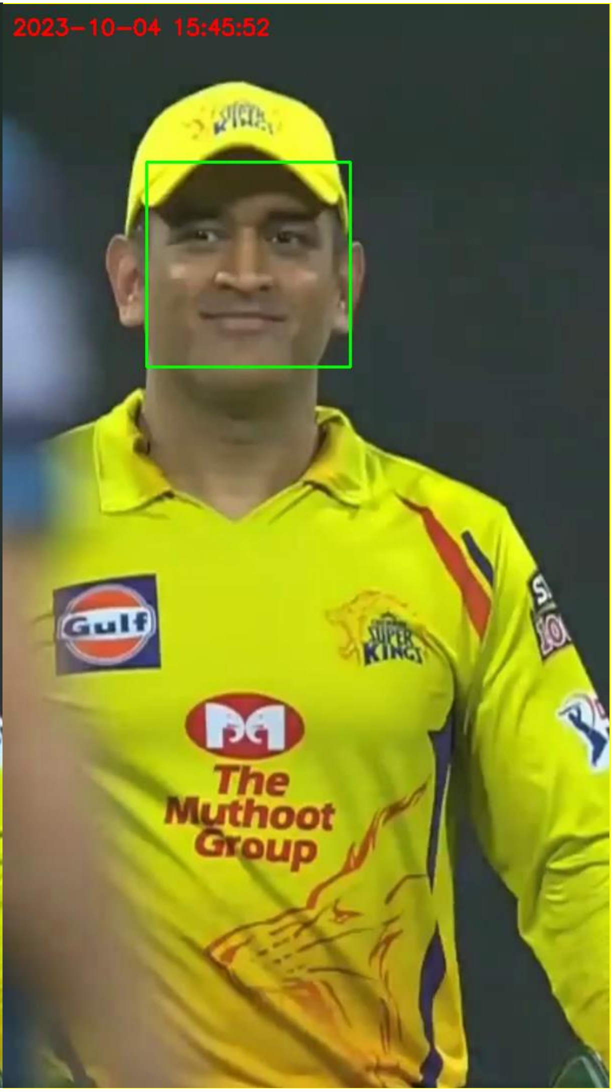
- Load the video and extract frames.
- Apply face detection to locate faces in frames.
- Use a trained face recognition model to identify faces.
- Compare detected faces to a database of known individuals.
- Apply a threshold to decide on face matches.
- Optionally, annotate and visualize recognized 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
import datetime
from google.colab.patches import cv2_imshow

face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture('/content/Ms_dhoni.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:

In conclusion, performing face detection and recognition on video is a multifaceted process crucial for various applications, including security and surveillance, personalization, and more. It involves the stages of data generation, face recognition, and training data preparation. Detecting and recognizing faces accurately is essential for the success of such systems. While advancements in deep learning and computer vision have made significant strides in improving face detection and recognition accuracy, challenges still exist, such as handling varying lighting conditions, angles, and occlusions. However, with ongoing research and development, the capabilities of these systems are constantly evolving, promising even more robust and efficient solutions in the future.