Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

School: .... Campus: ....

Academic Year: .... Subject Name: .... Subject Code: ....

Semester: .... Program: .... Branch: .... Specialization: ....

Date: ....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement :

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

1. Object detection and tracking using center based algorithm

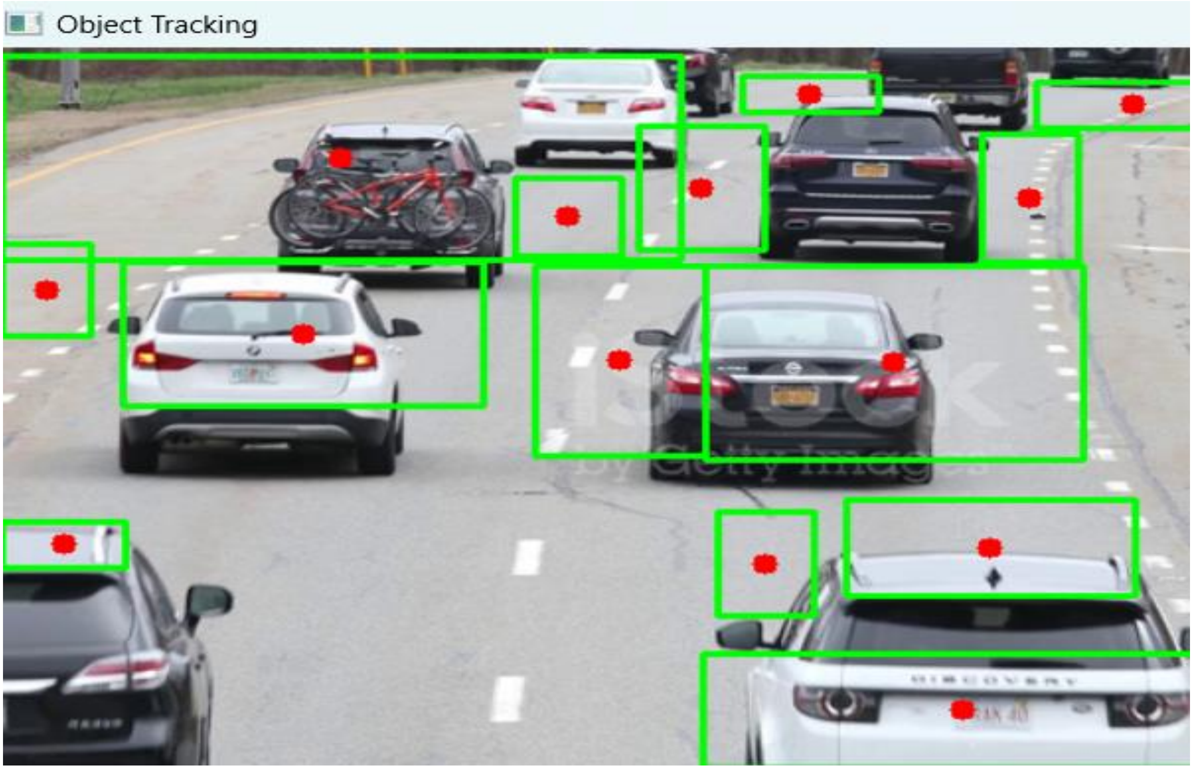
✱As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Testing Phase: Compilation of Code (error detection)

1) *Object detection and tracking using center based algorithm*

```
import cv2

# Define the minimum area to consider for an object
MIN_AREA = 500

# Function to detect objects in a frame
def detect_objects(frame):
    # Convert frame to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # Apply GaussianBlur to reduce noise
    blurred = cv2.GaussianBlur(gray, (11, 11), 0)
    # Apply thresholding to get binary image
    _, thresh = cv2.threshold(blurred, 200, 255, cv2.THRESH_BINARY)
    # Find contours in the binary image
    contours, _ = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    # Initialize list to store object bounding boxes
    bounding_boxes = []
    # Iterate through contours
    for contour in contours:
        # Compute the area of contour
        area = cv2.contourArea(contour)
        # If area is greater than minimum area, consider it as an object
        if area > MIN_AREA:
            # Get the bounding box coordinates
            x, y, w, h = cv2.boundingRect(contour)
            bounding_boxes.append((x, y, x + w, y + h))  # Store bounding box coordinates
    return bounding_boxes

# Function to track objects using their center points
def track_objects(bounding_boxes, frame):
    # Iterate through bounding boxes
    for box in bounding_boxes:
        # Calculate the center of the bounding box
        center_x = (box[0] + box[2]) // 2
        center_y = (box[1] + box[3]) // 2
        # Draw rectangle around the object
        cv2.rectangle(frame, (box[0], box[1]), (box[2], box[3]), (0, 255, 0), 2)
        # Draw circle at the center of the object
        cv2.circle(frame, (center_x, center_y), 5, (0, 0, 255), -1)

# Open video capture
cap = cv2.VideoCapture(r"C:\\Users\\HP\\Downloads\\car_-_2165 (360p).mp4")

# Loop through the video frames
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # Detect objects in the frame
    bounding_boxes = detect_objects(frame)

    # Track objects using their center points
    track_objects(bounding_boxes, frame)

    # Display the frame
    cv2.imshow('Object Tracking', frame)

    # Break the loop when 'q' is pressed
    if cv2.waitKey(25) & 0xFF == ord('q'):
        break

# Release video capture and close all windows
cap.release()
cv2.destroyAllWindows()
```

* Implementation Phase: Final Output (no error)



*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

From above experiment we learnt about object detection and tracking using center-based algorithm

Object detection and tracking using a center-based algorithm starts by identifying objects in an image or video frame using advanced models like YOLO or SSD. Once objects are detected, their centroids, or central points, are computed. These centroids serve as unique identifiers for each object. As frames progress, the algorithm continuously updates these centroids to track the movement of objects across frames. By comparing centroids between frames, the algorithm determines the trajectory and speed of each object. This approach is efficient and computationally lightweight, making it suitable for real-time applications like surveillance, autonomous vehicles, and human-computer interaction.

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Signature of the Faculty:

Regn. No. :