

## **EXERCISE NO. 7**

### **SSD NETWORK FOR A SELF-DRIVING CAR APPLICATION**

#### **AIM:**

To train a SSD network for a self-driving car application.

#### **ALGORITHM:**

1. Import the necessary libraries.
2. Load the pre-trained SSD model and configuration file.
3. Define the object classes relevant to self-driving cars.
4. Load the video file.
5. Start video capture with the initialisation of time-tracking.
6. Capture a video frame every 2 seconds until the timer ends.
7. Show the last captured frame with detection.

#### **PROGRAM:**

```
import cv2
import numpy as np
import time

prototxt_path = ".../MobileNetSSD_deploy.prototxt"
caffemodel_path = ".../MobileNetSSD_deploy.caffemodel"
net = cv2.dnn.readNetFromCaffe(prototxt_path, caffemodel_path)

CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat", "bottle", "bus",
           "car", "cat", "chair", "cow", "diningtable", "dog", "horse", "motorbike",
           "person", "pottedplant", "sheep", "sofa", "train", "tvmonitor"]
RELEVANT_CLASSES = {"car", "bus", "motorbike", "person", "bicycle", "traffic light"}

video_source = ".../FREE STOCK FOOTAGE - Heavy traffic.mp4"
cap = cv2.VideoCapture(video_source)

if not cap.isOpened():
    print("Error: Could not open video.")
    exit()

last_frame = None
```

```

last_capture_time = time.time()

while True:
    ret, frame = cap.read()
    if not ret:
        break

    current_time = time.time()

    if current_time - last_capture_time >= 2:
        (h, w) = frame.shape[:2]
        blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)), 0.007843, (300, 300), 127.5)
        net.setInput(blob)
        detections = net.forward()

        for i in range(detections.shape[2]):
            confidence = detections[0, 0, i, 2]
            if confidence > 0.4:
                idx = int(detections[0, 0, i, 1])
                label = CLASSES[idx]

                if label in RELEVANT_CLASSES:
                    box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
                    (startX, startY, endX, endY) = box.astype("int")

                    cv2.rectangle(frame, (startX, startY), (endX, endY), (0, 255, 0), 2)
                    text = f"{label}: {confidence * 100:.1f}%"
                    cv2.putText(frame, text, (startX, startY - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,
255, 0), 2)

                    print(f"Detected: {label} with {confidence * 100:.1f}% confidence")

        cv2.imshow("Self-Driving Car Object Detection", frame)
        last_frame = frame.copy()
        last_capture_time = current_time

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

```

```
break
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```

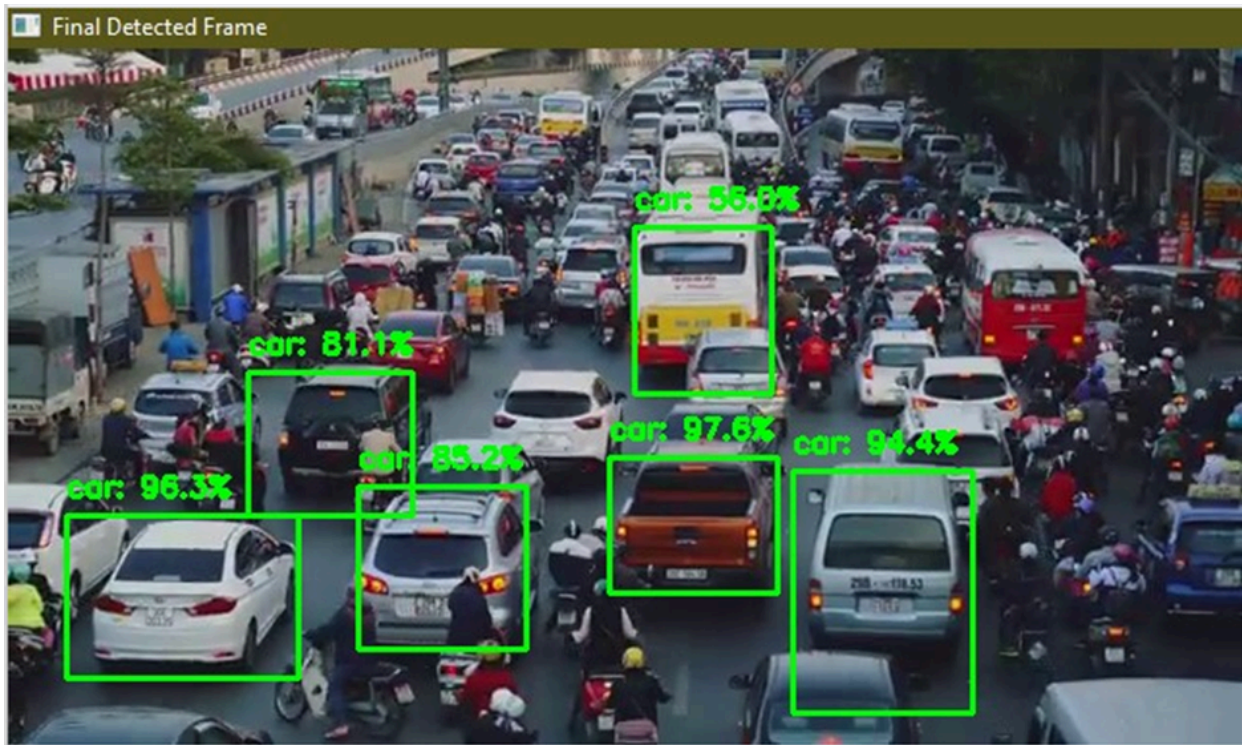
```
if last_frame is not None:
```

```
    cv2.imshow("Final Detected Frame", last_frame)
```

```
    cv2.waitKey(0)
```

```
    cv2.destroyAllWindows()
```

## OUTPUT:



## RESULT:

Thus the program has been successfully implemented and verified.