IISER Project Report

Submitted By-Tanisha Basu
Submitted To-Dr.Kuntal Roy

TOPIC-AUTONOMOUS VEHICLE

Autonomous vehicles, often referred to as self-driving cars, represent a groundbreaking advancement in transportation technology. These vehicles are capable of navigating and operating without direct human intervention, relying on a combination of sensors, cameras, radar, and artificial intelligence to perceive their environment, make decisions, and drive to their destinations. The development of autonomous vehicles promises to transform the automotive industry by enhancing safety, increasing efficiency, and reducing the need for human drivers.As technology continues to evolve, autonomous vehicles are poised to revolutionize the way we travel, offering potential benefits such as reduced traffic congestion, lower emissions, and greater accessibility for individuals unable to drive.

USE OF ML IN AUTONOMOUS VEHICLES-

Machine learning (ML) plays a pivotal role in the development and functionality of autonomous vehicles. It is crucial for several reasons:

1.Object Detection-
ML algorithms analyze data from cameras and LIDAR to identify and classify objects like pedestrians, other vehicles, traffic signs, and lane marking

2. Decision Making and Planning
ML is used to predict the behavior of otherroad users, which helps in making informed decisions about speed, lane changes, and turns.

3.Anomaly Detection and Safety
ML algorithms can detect anomalies in the vehicle systems and predict potential failures before they occur, enhancing reliability and safety.

4.Traffic Management and Optimization
ML allows autonomous vehiclesto communicate with traffic management system.

Object Detection-

The dataset used for object detection that can be done in Autonomous vehicle contains verious medias of car that can be used for testing as well as training data.(as well as videos) .
Here the detection can be done with help of RNN and SSD that can carry out follows-

1.Faster R-CNN

a.Region Proposal Network (RPN): Generates region proposals where objects might be located.

b.Region of Interest (RoI) Pooling: Extracts fixed-size feature maps from these proposals.

c.Classification and Regression Heads: Classifies the object within each proposal and refines the bounding box coordinates.

2.Single Shot MultiBox Detector (SSD)

a.Convolutional Layers: Extract features from the input image.

b.Multi-scale Feature Maps: Predict objects at multiple scales.

c.Default Boxes: Uses predefined boxes of different aspect ratios to detect objects at various scales.

Faster R-CNN offers high accuracy through precise region proposals and detailed classification.
SSD provides real-time detection.

2.Traffic Signal Detection-

The dataset used for the traffic detection contains various cropped images of traffic signal that can be used for test and training purpose.

Approach used is CNN that helps to to identify and classify various traffic signs from input images, improving the accuracy and reliability of an autonomous vehicle's navigation system.

Explanation-

1.Input Layer: Processes images of traffic signs, typically standardized in size and format.

2.Convolutional Layers: Detect key features of traffic signs, such as shapes, colors, and patterns, crucial for distinguishing between different signs (e.g., stop sign vs. speed limit sign).

3.Pooling and Dropout Layers: Reduce the data complexity and enhance the generalization of the model, making it more robust to variations in traffic sign images.

Layers and Functions-

1.Conv2D Layers:

First Layer: Applies 64 filters of size 2x2 to the input image, enhancing feature extraction (like edges and textures) important for recognizing traffic signs.

2.Second Layer: Adds another set of 64 filters, refining the features identified in the previous layer.

3.Third Layer: Uses 128 filters of size 2x2 to capture more complex features and patterns.

BatchNormalization Layers: Normalizes the output from the convolutional layers, speeding up training and improving model stability.

3.MaxPooling2D Layers:
Reduces the spatial dimensions of the feature maps, helping to downsample the data and highlight the most important features.

4.Dropout Layer:
With a dropout rate of 0.1, this layer helps to prevent overfitting by randomly setting some of the activations to zero during trainin

3.Lane Detection-
CNN can help out to detect the lane as well boundaries .CNN architectures often include multi-scale analysis through techniques li
pooling layers or multi-resolution input. This allows the network to detect lane markings at different scales, such as both close-up
markings and distant lane boundaries visible on the horizon.
We used a similar approach for marking along the layers and boundaries that can help out to the autonomous vehicles

4.3D Object Detection-
Detecting 3D objects involves understanding their spatial layout and attributes in a given scene. Convolutional Neural Networks (CNNs) are a popular choice for this task due to their ability to learn hierarchical features from 2D images and their extensions to handle 3D data, such as volumetric convolutions or point cloud processing.

MOTION DETECTION-
Machine learning can be effective for motion detection, it's not the only approach. Traditional computer vision techniques like background subtraction or optical flow are also commonly used for motion detection tasks.

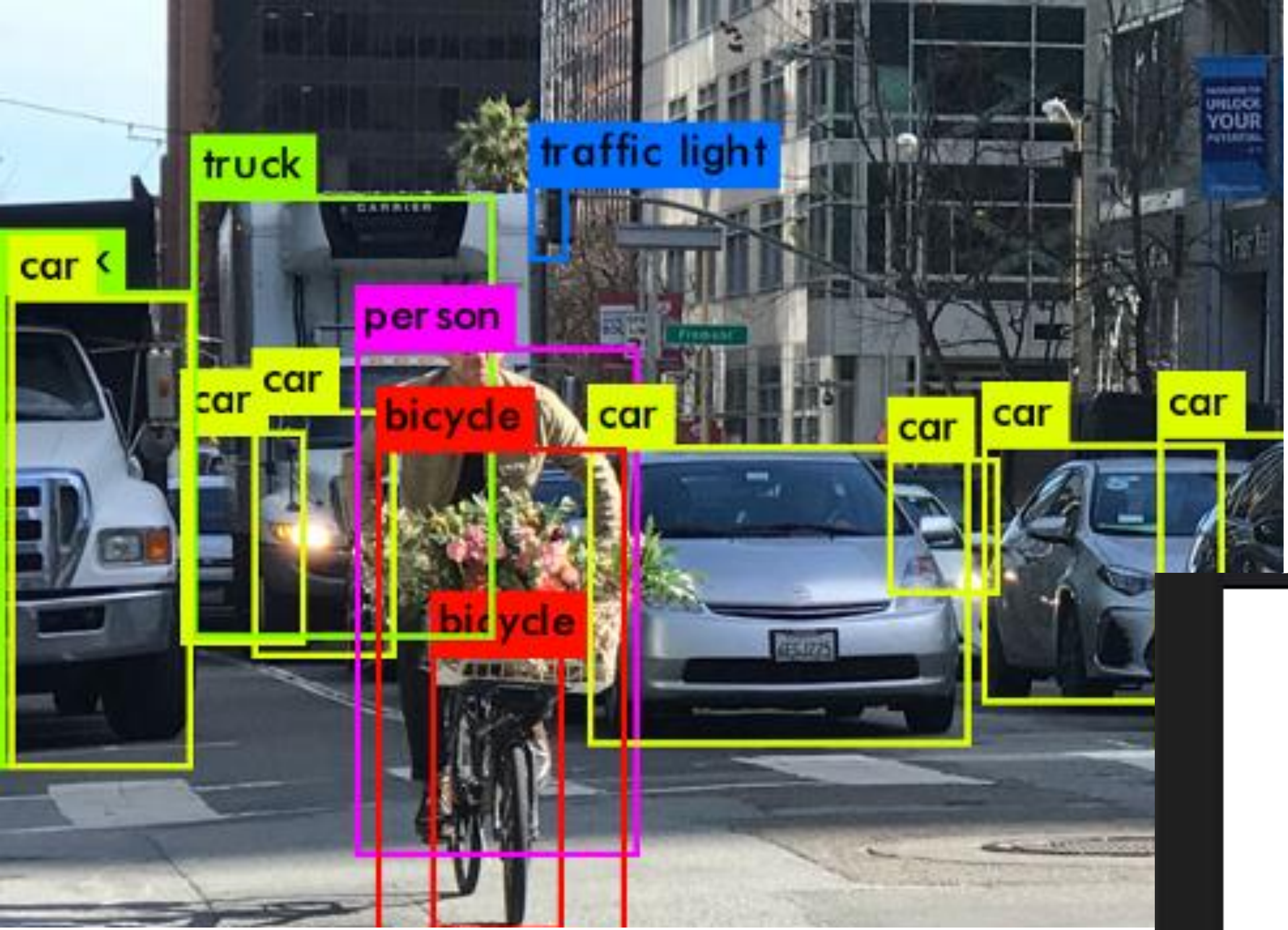
YOLOV DETECTION-

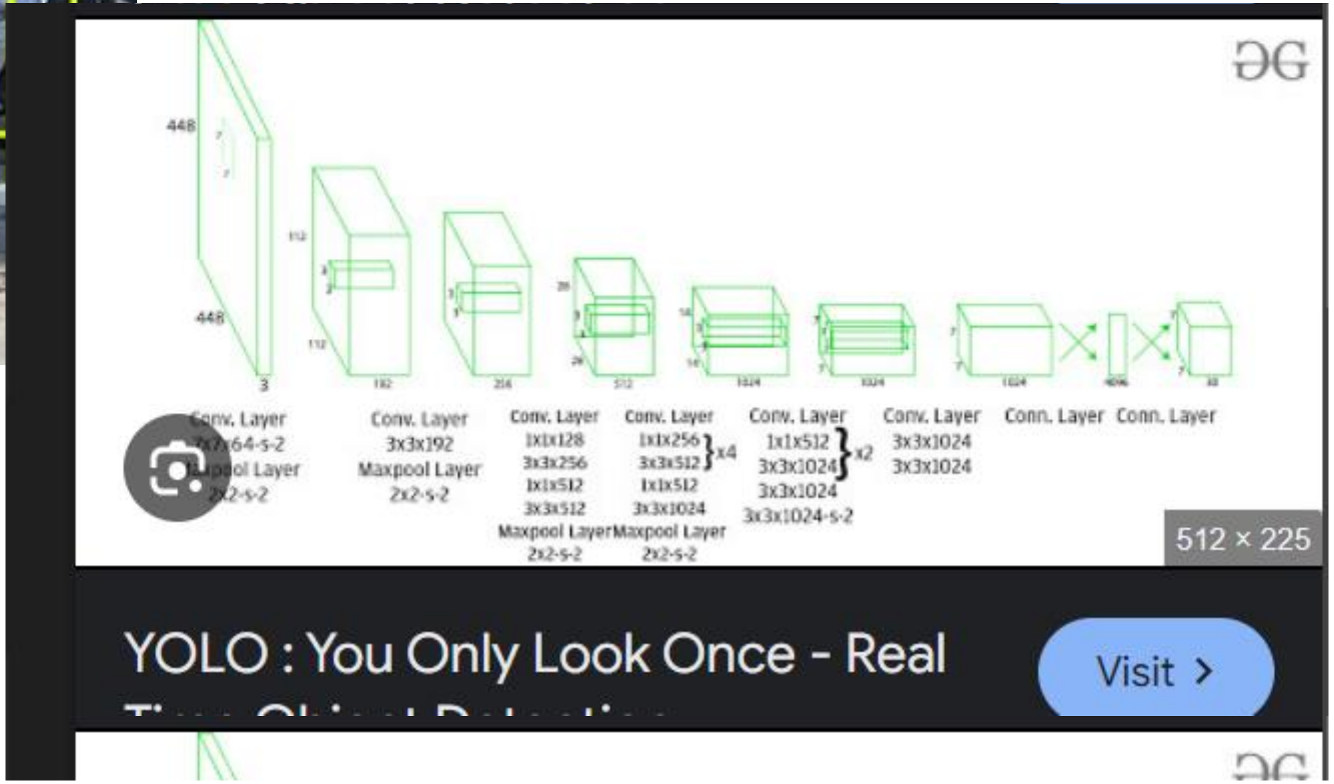
YOLOv (You Only Look Once version n) is a family of deep learning models used for object detection.

The key idea behind YOLO isto divide the input image into a grid and predict bounding boxes and probabilitiesfor each grid cell.YOLO (You Only Look Once) models can be integral to the functionality of autonomous vehicles by providing real-time object detection capabilities.

These models are adapt at recognizing and localizing various objects in a vehicle

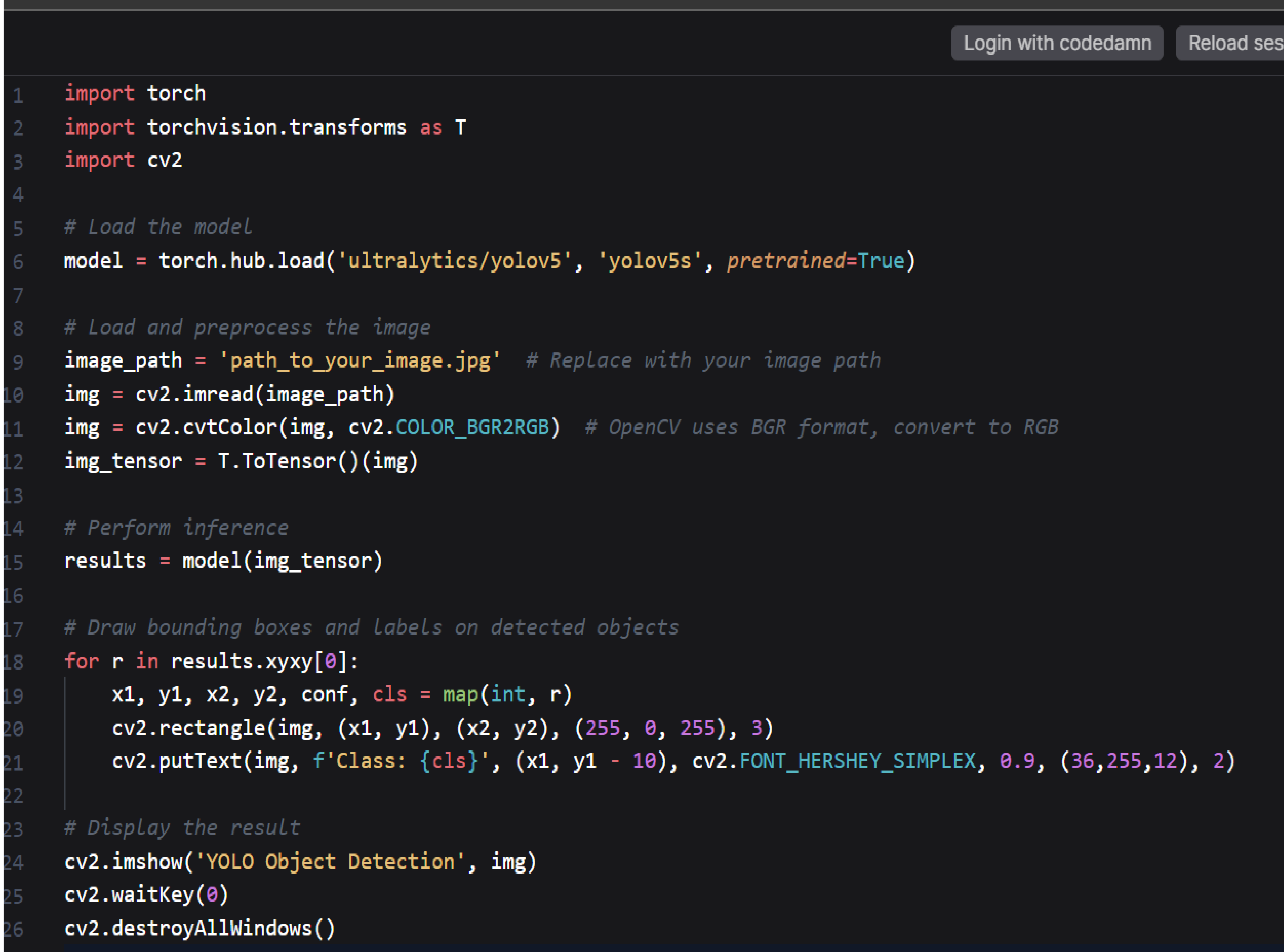
Versions of YOLOv are_x0002_YOLOv 1,YOLOv2,YOLOv3,YOLOv4, YOLOv5, YOLOv6, and YOLOv7,YOLOv8,YOLOv10 models.





YOLO : You Only Look Once - Real

Code Snippet(YOLOV)



```python
import torch
import torchvision.transforms as T
import cv2

# Load the model
model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)

# Load and preprocess the image
image_path = 'path_to_your_image.jpg'  # Replace with your image path
img = cv2.imread(image_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  # OpenCV uses BGR format, convert to RGB
img_tensor = T.ToTensor()(img)

# Perform inference
results = model(img_tensor)

# Draw bounding boxes and labels on detected objects
for r in results.xyxy[0]:
    x1, y1, x2, y2, conf, cls = map(int, r)
    cv2.rectangle(img, (x1, y1), (x2, y2), (255, 0, 255), 3)
    cv2.putText(img, f'Class: {cls}', (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36,255,12), 2)

# Display the result
cv2.imshow('YOLO Object Detection', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

STUDY ON RESULTS

ASPECT	CNN BASED DETECTION	YOLOV OBJECT DETECTION
Approach	Proposes Regions and classifies them	Predicts bounding boxes and classifies
Speed	Slower due to multiple passes	Faster due to single pass prediction
Performance	Typically slower	High real time performance
Accuracy	Can achieve high accuracy proposals	Generally good accuracy with speed
Object Detection Prediction	May require multiple scales for detection	Unified detection across scales
Application	Suitable for accurate	Ideal for autonomous drive,surveillance
Usage	Used for real time processing	Widely used for better accuracy and speed
Model Size	Larger due to multiple stages	Generally compact especially YOLOV4
Examples	R-CNN,FASTER R-CNN	yolov2,yolov3,yolov5

Thank You