**ROAD OBJECT DETECTION WITH DEEP LEARNING**

Team TRICORE

M Arun Kumar

Shaik Asif Pasha

M Pawan Kalyan

**Abstract :**

This paper presents a comprehensive review of road object detection using deep learning. It discusses the challenges associated with road object detection and the limitations of traditional computer vision techniques. The paper introduces deep learning architectures such as CNNs, Faster R-CNN, YOLO, and SSD, which have shown effectiveness in detecting road objects. It highlights the importance of large-scale annotated datasets for training deep learning models. The challenges of deep learning-based road object detection, including computational resources and real-time performance, are addressed. Recent advancements such as multi-modal sensor fusion and attention mechanisms are discussed. The paper concludes with a summary of key findings and future directions, emphasizing the need for improved robustness, real-time performance, and generalization capabilities. Road object detection using deep learning has the potential to enhance road safety and enable autonomous vehicles in the future.

**METHODOLOGY :**

DATA COLLECTION –

1. Gather a diverse dataset of road images or videos and label the objects of interest, like cars, pedestrians, and signs.

2. Split the dataset into training, validation, and testing sets. Use data augmentation techniques to enhance the model's performance.

**MODEL Used:**

**YOYO3**

YOLO stands for You Only Look Once. It is a real-time object recognition algorithm. It can classify and localize multiple objects in a single frame. YOLO is a very fast and accurate algorithm for its simpler network architecture.

**TRAINING**

Train the model using pre-trained weights on a large-scale dataset like ImageNet. Fine-tune optimization algorithms like SGD or Adam, and monitor its performance it with on the validation set.

Optimize the model's hyperparameters, such as learning rate, batch size, and regularization techniques, through methods like grid or random search.

Evaluate the trained model's performance on the testing set using metrics like mAP, precision, recall, and F1 score to measure its accuracy in detecting road objects.

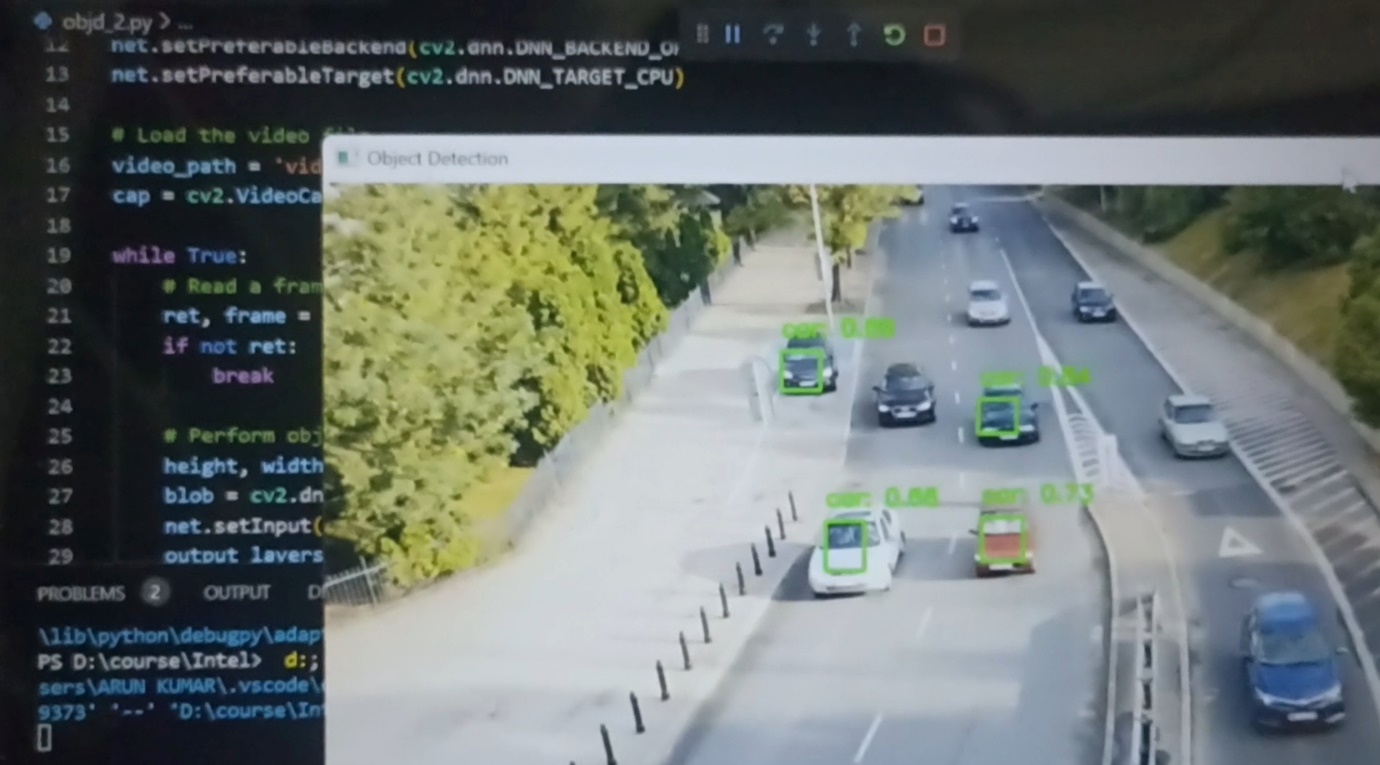
Apply post-processing techniques like non-maximum suppression (NMS) to eliminate duplicate or overlapping detections. Set confidence thresholds to filter out less certain detections

**LIBRARIES:**

1. Python – 3.x (We used python 3.8.8 in this project)  
2. OpenCV – 4.4.0

3. NumPy – 1.20.3  
4. YOLOv3 Pre-trained model weights and Config Files.

**OUTPUT SCREEN:**



**Reference:**

[**https://github.com/TilakD/2D-Road-Object-Detection**](https://github.com/TilakD/2D-Road-Object-Detection)

[**https://github.com/sourabbsridhar/road-object-detection-using-yolov4**](https://github.com/sourabbsridhar/road-object-detection-using-yolov4)

**Status:**

We are currently getting the output for the given problem statement as of now, in the future edition we will provide the advancements.