PROJECT REPORT ON

YOLOV9 FOR VEHICLE DETECTION AND TRAFFIC ANALYSIS

Submitted by

Jay Khinchi	234101020
Jaya Krishna	234101021
Raja Gopal	234101026
Chenna Kesava	234101061

Under The Guidance Of

Dr. Debanga Raj Neog
Mehta Family School of Data Science and
Artificial Intelligence, IIT Guwahati

Jan- April 2024

Image Processing with Machine Learning (DA526)

1. Problem Statement

The goal is to develop an efficient and accurate vehicle detection and traffic analysis system using YOLOv9 and YOLOv8. This system will enable real-time vehicle detection in traffic scenarios, providing insights such as vehicle counts, traffic flow analysis, and congestion monitoring.

2. Related Work

Real-time object detection has emerged as a critical component in numerous applications, spanning various fields such as autonomous vehicles, robotics, video surveillance, and augmented reality. Among the different object detection algorithms, the YOLO (You Only Look Once) framework has stood out for its remarkable balance of speed and accuracy, enabling the rapid and reliable identification of objects in images.

In addition to the YOLO framework, the field of object detection and image processing has developed several other notable methods. Techniques such as R-CNN (Region-based Convolutional Neural Networks) and its successors, Fast R-CNN and Faster R-CNN , have played a pivotal role in advancing the accuracy of object detection. These methods rely on a two-stage process, where selective search generates region proposals, and convolutional neural networks classify and refine these regions. Another significant approach is the Single-Shot MultiBox Detector (SSD), which, similar to YOLO, focuses on speed and efficiency by eliminating the need for a separate region proposal step. Additionally, methods like Mask R-NN have extended capabilities to instance segmentation, enabling precise object localization and pixel-level segmentation. These developments, alongside others such as RetinaNet and EfficientDet have collectively contributed to the diverse landscape of object detection algorithms. Each method presents unique tradeoffs between speed, accuracy, and complexity, catering to different application needs and computational constraints.

The name YOLO stands for "You Only Look Once," referring to the fact that it was able to accomplish the detection task with a single pass of the network, as opposed to previous approaches that either used sliding windows followed by a classifier that needed to run hundreds or thousands of times per image or the more advanced methods that divided the task into two-steps, where the first step detects possible regions with objects or regions proposals and the second step run a classifier on the proposals. Also, YOLO used a more straightforward output based only on regression to predict the detection outputs as opposed to Fast R-CNN that used two separate outputs, a classification for the probabilities and a regression for the box's coordinates.

YOLO models have been used in agriculture to detect and classify crops , pests, and diseases , assisting in precision agriculture techniques and automating farming processes. They have also been adapted for face detection tasks in biometrics, security, and facial recognition systems.

2.1 Object Detection Metrics and Non-Maximum Suppression (NMS):

The Average Precision (AP), traditionally called Mean Average Precision (mAP), is the commonly used metric for evaluating the performance of object detection models. It measures the average precision across all categories, providing a single value to compare different models.

The AP metric is based on precision-recall metrics, handling multiple object categories, and defining a positive prediction using Intersection over Union (IoU).

2.1.1 Intersection over Union: It is a measure to assess the quality of the predicted bounding boxes. IoU is the ratio of the intersection area to the union area of the predicted bounding box and the ground truth bounding box. It measures the overlap between the ground truth and predicted bounding boxes.

2.1.2 Non-Maximum Suppression (NMS) : It is a post-processing technique used in object detection algorithms to reduce the number of overlapping bounding boxes and improve the overall detection quality. Object detection algorithms typically generate multiple bounding boxes around the same object with different confidence scores. NMS filters out redundant and irrelevant bounding boxes, keeping only the most accurate ones. Algorithm 1 describes the procedure. Figure 2.1 shows the typical output of an object detection model containing multiple overlapping bounding boxes and the output after NMS.

Figure 2.1

3.Dataset

Car Counting dataset

Dataset consists of 7284 images of 8 classes name as: Bus, Pickup, Motor-Cycle, SUV, Sedan, Truck, Van, Background.
Dataset is splitted in three set as:

Train	**Validation**	**Test**
5537	1456	291

3.1 Sample Images from Dataset:

Figure 3.1.1

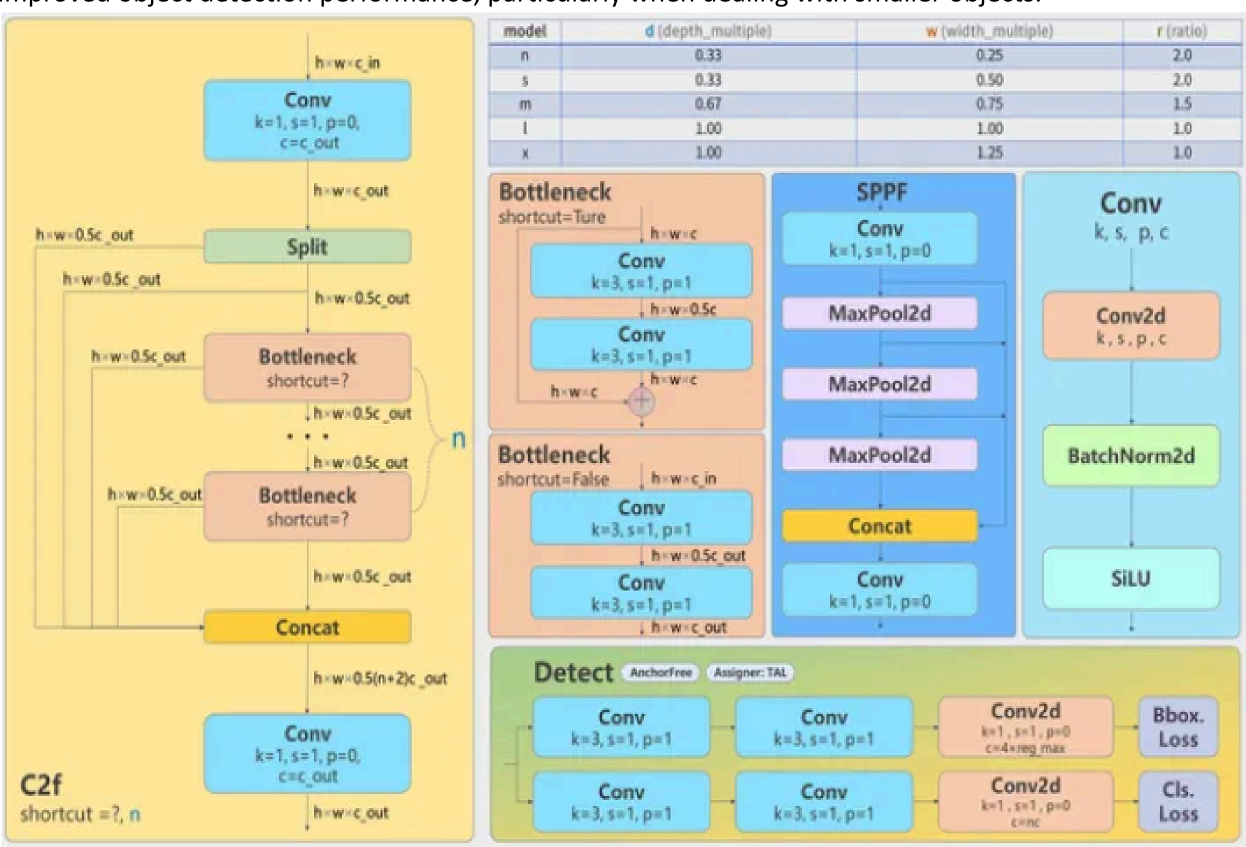
Figure 3.1.2

4.Method

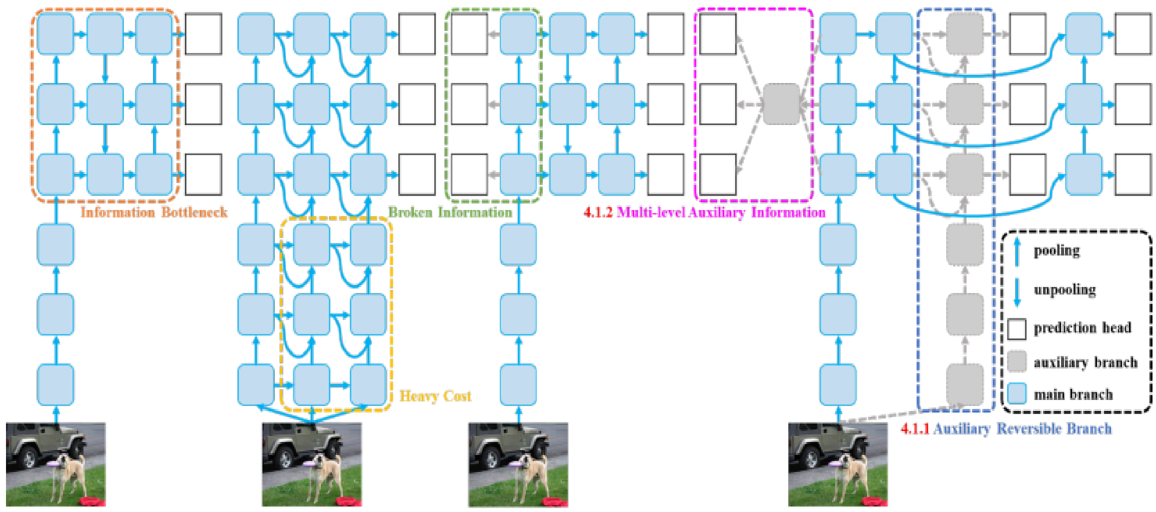
We use two architectures Yolov8 and Yolov9 , we have trained the dataset on yolov8 and yolov9 then we compare the results of both the methods some classes yolov8 missed to detect from an image but when we tried the same image with yolov9 and yolov9 able to detect those missed classes. Yolov9 performs faster than yolov8. During all the process we are getting double bounding boxes on some objects of input image to get rid of this we used the technique of non-max suppression: The intersection over union(IOU) method we set IOU as greater than or equal to 0.7. Also to classify the correct labels of the object we used a confidence score which we set 0.25. If for any bounding box of object confidence score less than 0.25 then we will not classify that object and ignore it.

4.1 Architecture of Yolov8:

YOLOv8 uses an anchor-free model with a decoupled head to independently process objectness, classification, and regression tasks. This design allows each branch to focus on its task and improves the model's overall accuracy. In the output layer of YOLOv8, they used the sigmoid function as the activation function for the objectness score, representing the probability that the bounding box contains an object. It uses the softmax function for the class probabilities, representing the objects' probabilities belonging to each possible class. YOLOv8 uses CIoU(Complete Intersection over Union) and DFL(Distribution Focal loss) functions for bounding box loss and binary cross-entropy for classification loss. These losses have improved object detection performance, particularly when dealing with smaller objects.



4.2 Architecture of Yolov9



(a) PAN [37] (b) RevCol [3] (c) Deep Supervision (d) Programmable Gradient Information

Information Bottleneck Principle:
The Information Bottleneck Principle reveals a fundamental challenge in deep learning: as data passes through successive layers of a network, the potential for information loss increases.

Reversible Functions:

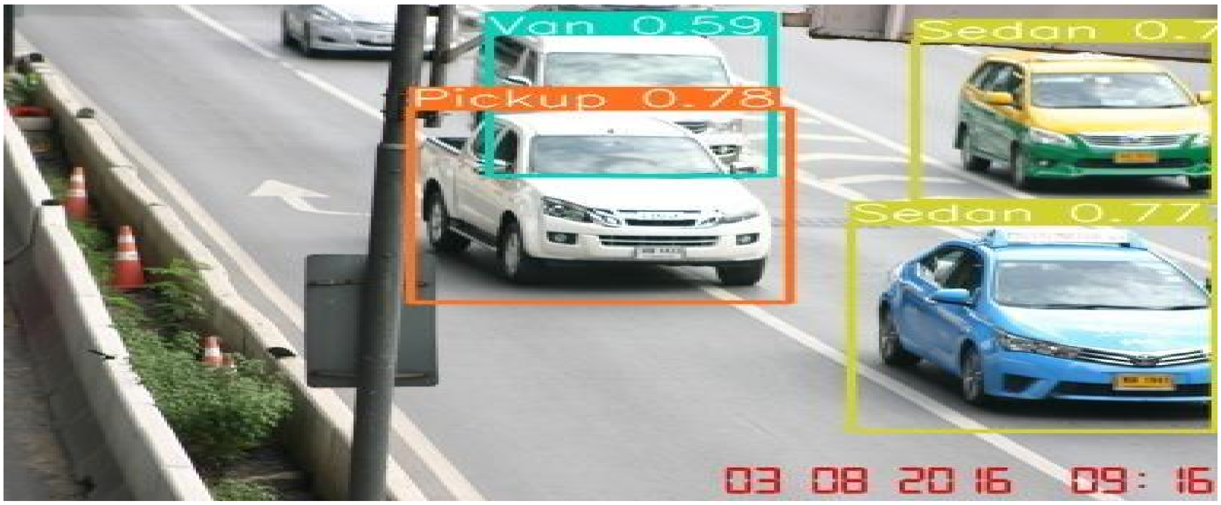
The concept of Reversible Functions is another cornerstone of YOLOv9's design. A function is deemed reversible if it can be inverted without any loss of information.

Programmable Gradient Information(PGI):

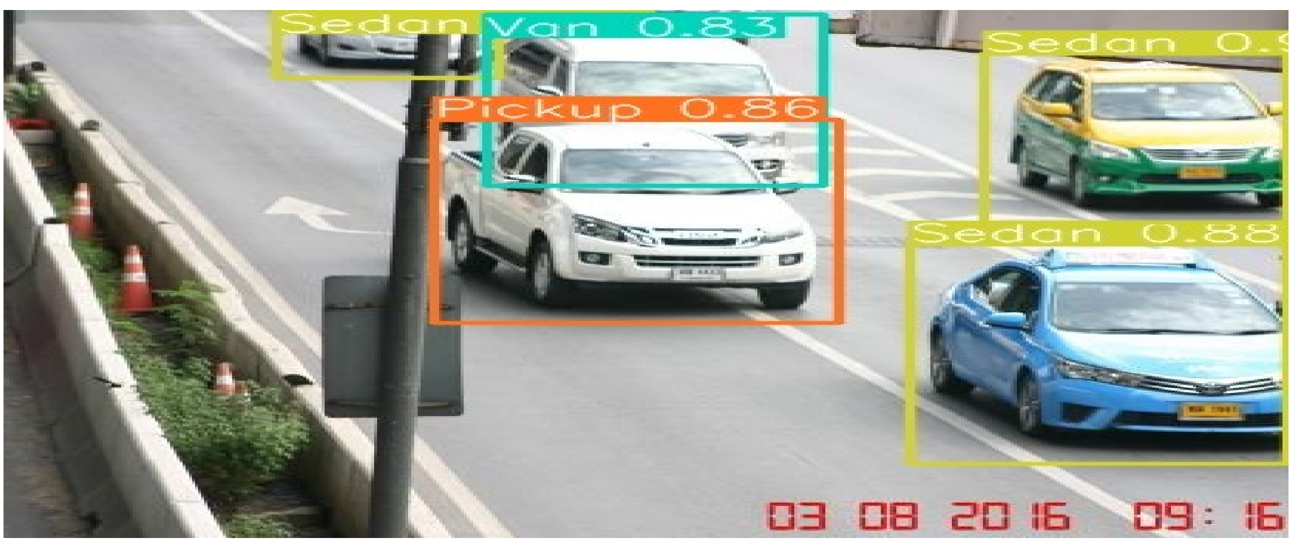
PGI is a novel concept introduced in YOLOv9 to combat the information bottleneck problem, ensuring the preservation of essential data across deep network layers. This allows for the generation of reliable gradients, facilitating accurate model updates and improving the overall detection performance.

5.Experiments and Results

In this experiment we trained the dataset on both the models yolov8 and yolov9 and compared the performance and Yolov9 performs better than yolov8.



Yolov8



Yolov9

The confidence scores of Yolov9 is higher than Yolov8 , as we can see in both the images the rightmost car in the image is predicted as a sedan. The confidence score of prediction in Yolov8 is 0.77 whereas in Yolov9 it is 0.88 and similarly for all the classified vehicles**.**

5.1 Confusion Matrix Yolov8

Predicted

Bus	20					3	1	20
motorcycle		73						26
pickup			2696	96	114	33	21	834
Suv			50	372	81		8	222
sedan			63	94	2637		13	814
Truck	11		39		3	1353	6	537
Van	2		10	18			320	118
Background	4	15	128	29	158	70	19	0
	Bus	Motorc ycle	Pickup	Suv	Sedan	Truck	van	background

Actual

5.2 Confusion Matrix Yolov9

Bus	29					1	1	9
Motorcycle		76						44
Pickup			2715	73	72	28	20	594
Suv			46	434	112		10	206
Sedan			31	68	2608		7	594
Truck	3		26		2	1313	2	346
Van	2		10	18		1	332	68
Background	3	12	165	31	202	118	16	0
	Bus	**Motor cycle**	**Pickup**	**Suv**	**Sedan**	**Truck**	**van**	**background**

Predicted

Actual

6. Conclusion (with real world use case's)

- Traffic Flow Analysis: Use the "Car Counting" model to investigate traffic congestion on highways, identify peak travel hours, or optimize traffic lights timings.
- Transportation Planning: Use it in the assessment of potential road upgrade/expansion by counting the number of specific vehicle types over time.
- Parking Management: Employ it in counting and monitoring available spaces in a lot by identifying and subtracting currently occupied spots (depending on vehicle size/type).
- Security and Law Enforcement: Use the model in identifying unusual or suspicious activity by tracking the frequency and type of vehicles in particular areas at various times.
- Retail Studies: Use it to analyze customer visits in a store parking lot by identifying and counting the types of vehicles at different intervals, potentially understanding the store's peak hours or client habits.

Results(video and Weights)

Drive Link of Results and Weights

Github link:

https://github.com/j2810/IPML-DA-526--Project-Yolov9-yolov8-for-Vehicle-Detection-and-Traffic-Analysis.git

References

- https://docs.ultralytics.com/models/yolov8/
- https://arxiv.org/abs/2402.13616
- https://arxiv.org/abs/2401.17270
- https://universe.roboflow.com/cc-pintel/car-counting/dataset/1