Modal Selection

Model selection is a critical process of model training. The most important factors for considerations are as follows:

- Real Time Object Detection Performance
- Simple Expandable Design
- Handle diverse data
- Good performance in Object detection
- Easily available pre-trained models for faster fine tuning
- Multi-class support in object detection

Considering all the above factors, I have chosen YOLOv5, as per pretrained models it has outperform major object detection. It is a good choice for the **BDD100K dataset** due to several reasons mention below:

- High efficiency in real-time object detection.
- Architecture strikes a good balance between speed and accuracy, making it suitable for processing a vast number of images from the dataset quickly without significant compromises in performance.
- Architecture with better anchor box generation and model tuning. This can be beneficial when dealing with the variability in object shapes and sizes in the BDD100K dataset, which consists of images taken from different perspectives, lighting conditions, and environments.
- Handle large-scale datasets efficiently.
- Strong in detecting small and medium-sized objects, which aligns well with the task of vehicle detection in the BDD100K dataset, where the vehicles can vary greatly in size, pose, and occlusion.
- Multi-class detector and works well when detecting multiple object types in a single frame, making it an ideal choice for BDD100K's varied object annotations.

Architecture Design YOLOv5

The architecture of YOLOv5 is designed to be efficient and fast in object detection tasks. Below is an overview of the key components of the YOLOv5 architecture:

1. Input Layer (Image Preprocessing)

- The input layer of YOLOv5 processes normalised images of size **640x640** (or other sizes based on the configuration) before feeding them into the network.
- **Augmentation:** YOLOv5 uses various data augmentation techniques like flipping, rotation, and color jitter to improve robustness and generalization on diverse datasets.

2. Backbone (Feature Extraction)

YOLOv5 uses a modified version of the **CSPDarknet** architecture, which is efficient and lightweight. The backbone in YOLOv5 is designed to handle multi-scale feature extraction, which is crucial for detecting objects of various sizes.

- **CSPDarknet53 Backbone:** It improves the flow of gradients and reduces computation while maintaining accuracy. The backbone applies multiple convolutional layers to detect hierarchical features at different scales.

3. Neck (Feature Aggregation)

The main component of the neck in YOLOv5 is **PANet (Path Aggregation Network)**, which is used to gather information from both deep and shallow layers of the network to help improve feature fusion.

- **PANet:** PANet enhances the feature fusion capabilities by allowing for better utilization of low-level, high-resolution features. This enables better localization, especially for small objects. The features from different levels of the backbone are passed through PANet for multi-scale feature aggregation.
- **FPN (Feature Pyramid Network):** In some versions of YOLOv5, a Feature Pyramid Network (FPN) is used in conjunction with PANet for better multi-scale feature extraction. FPN helps the model perform well on both large and small objects by using multi-scale feature maps.

4. Head (Prediction Layer)

The **Head** layer of YOLOv5 is responsible for generating the final object predictions, including as follows:

- Bounding Box Prediction
- Objectness Score
- Class Scores

The head consists of convolutional layers that are responsible for this final prediction output, where the number of channels in the output corresponds to the number of bounding boxes (each having 4 values for coordinates, 1 for objectness, and the number of class labels).

5. Output Layer (Post-processing)

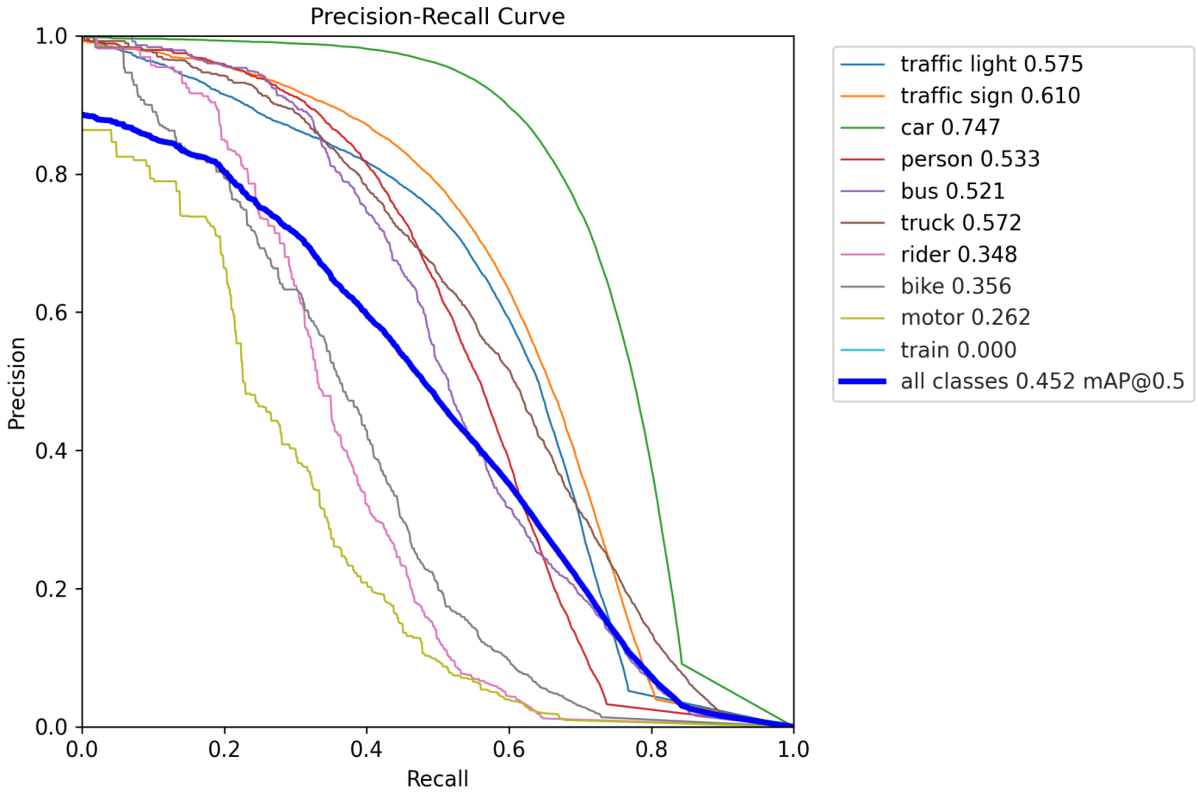
The output layer involves post-processing the raw predictions made by the head. This includes:

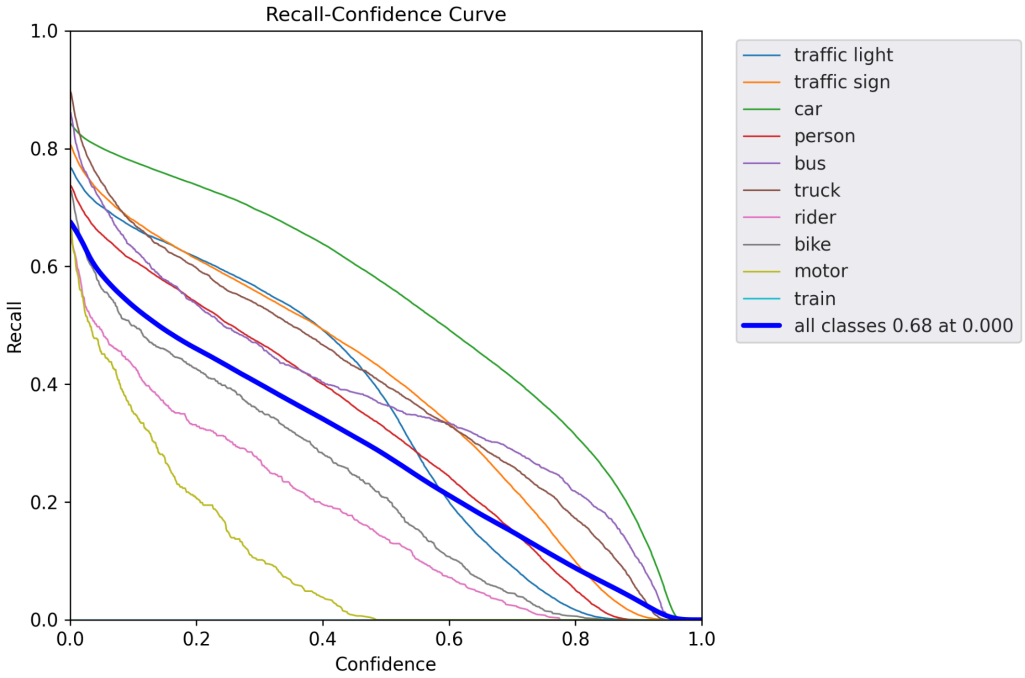
- **Non-Maximum Suppression (NMS):** After detecting multiple bounding boxes for the same object, NMS is used to eliminate redundant boxes and retain the most confident one.
- **Thresholding:** Predictions below a certain confidence threshold (e.g., 0.5) are discarded.
- **Final Predictions:** The model outputs the final bounding boxes with class labels and confidence scores for each object detected in the image.

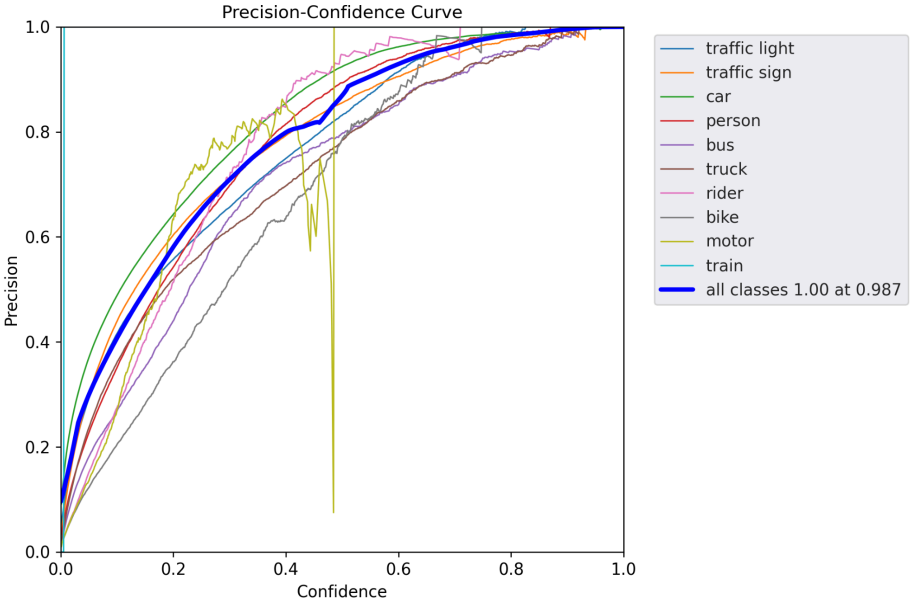
In conclusion, YOLOv5 architecture is optimized for both speed and accuracy, making it ideal for real-time object detection tasks, especially in scenarios like autonomous driving, surveillance, and robotics.

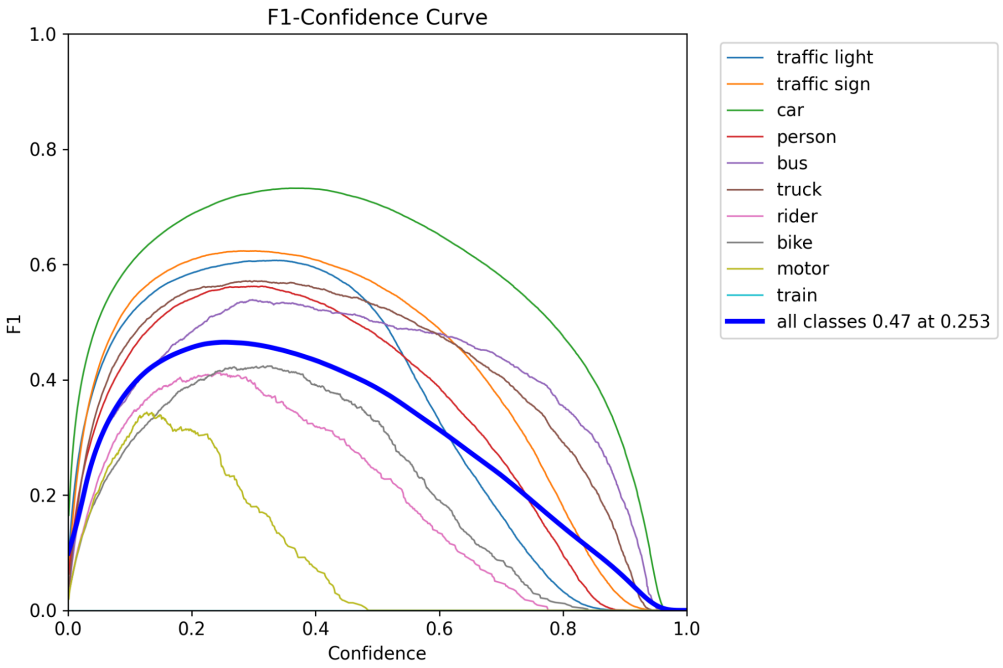
Modal Training

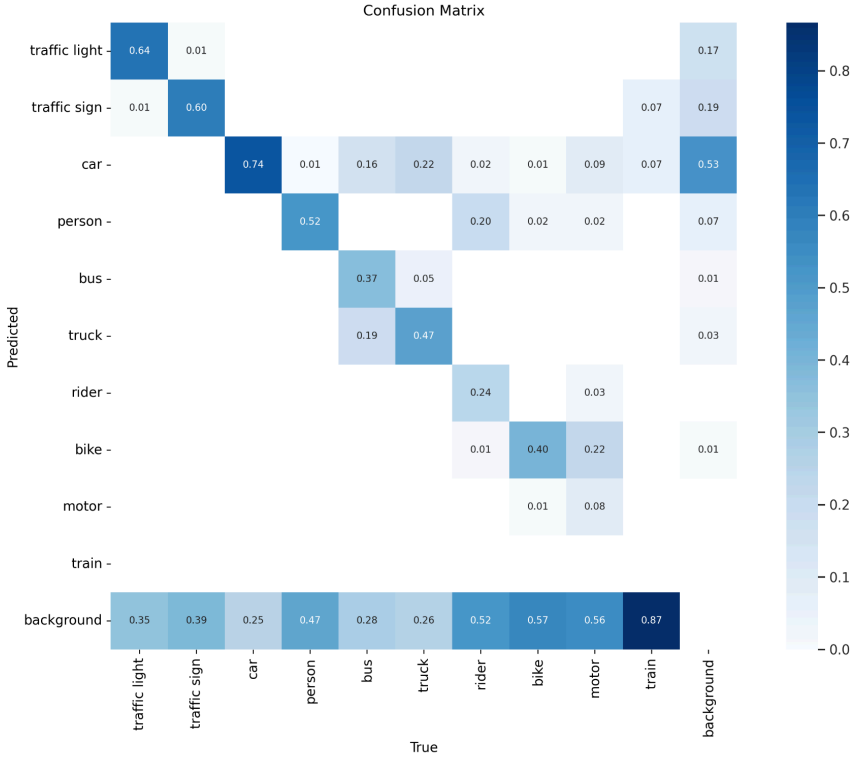
The selected model is YOLOv5 Large, consisting of 360 layers. It was trained on a dataset of approximately 70,000 images for 10 epochs and validated using a separate validation set of 10,000 images. Below are some key plots from the model's training:




Recall-Confidence Curve
Recall
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
traffic light
traffic sign
car
person
bus
truck
rider
bike
motor
train
all classes 0.68 at 0.000


Precision-Confidence Curve
Precision
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
traffic light
traffic sign
car
person
bus
truck
rider
bike
motor
train
all classes 1.00 at 0.987


F1-Confidence Curve
F1
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
traffic light
traffic sign
car
person
bus
truck
rider
bike
motor
train
all classes 0.47 at 0.253


Confusion Matrix
Predicted
True
traffic light
traffic sign
car
person
bus
truck
rider
bike
motor
train
background
0.64 0.01 0.17
0.01 0.60 0.07 0.19
0.74 0.01 0.16 0.22 0.02 0.01 0.09 0.07 0.53
0.52 0.20 0.02 0.02 0.07
0.37 0.05 0.01
0.19 0.47 0.03
0.24 0.03
0.01 0.40 0.22 0.01
0.01 0.08
0.35 0.39 0.25 0.47 0.28 0.26 0.52 0.57 0.56 0.87
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Images with Ground Truth

Images with Trained model predictions