# Computer Vision

## Course Project Report

**Student :** Khemchand | Khemchand.r-25@scds.saiuniversity.edu.in

**Title of the project:** Car Parts Detection

**CV Category:** Object Detection

### 1. Introduction

The objective of this project is to develop an object detection model to identify and localize car parts from images accurately. The primary goal is to utilize the YOLO v8 detector to predict bounding boxes, classify car parts, and achieve a high detection accuracy measured through mAP metrics. This project emphasizes practical applications such as enhancing vehicle maintenance and inspection processes.

### 2. Dataset

The dataset used for this project consists of annotated images of various car parts, divided into training, validation, and testing sets. The details are:

- **Training set**: Used for model training.
- **Validation set**: Employed during training to evaluate the model's performance and avoid overfitting.
- **Testing set**: Comprises 80 images with 617 instances of car parts, used to measure the final performance of the trained model.

The dataset contains 18 categories of car parts, such as back_bumper, back_glass, front_bumper, and wheel, among others. Each category is labelled with bounding boxes and class IDs.

### 3. Method

The YOLO v8 object detector, known for its speed and accuracy, is implemented for this task. Key features of YOLO v8 include:

- **Real-time inference capability**: YOLO v8 offers optimized inference with high accuracy.
- **Anchor-free detection**: It improves the localization of objects, particularly for small or overlapping regions.
- **PyTorch integration**: This makes it easier to customize the model and training pipeline.

The model was trained for 100 epochs, optimizing the parameters such as box loss, classification loss, and dfl (distribution focal loss). The best-performing model weights were saved for evaluation.

.

**4. Results and Analysis**

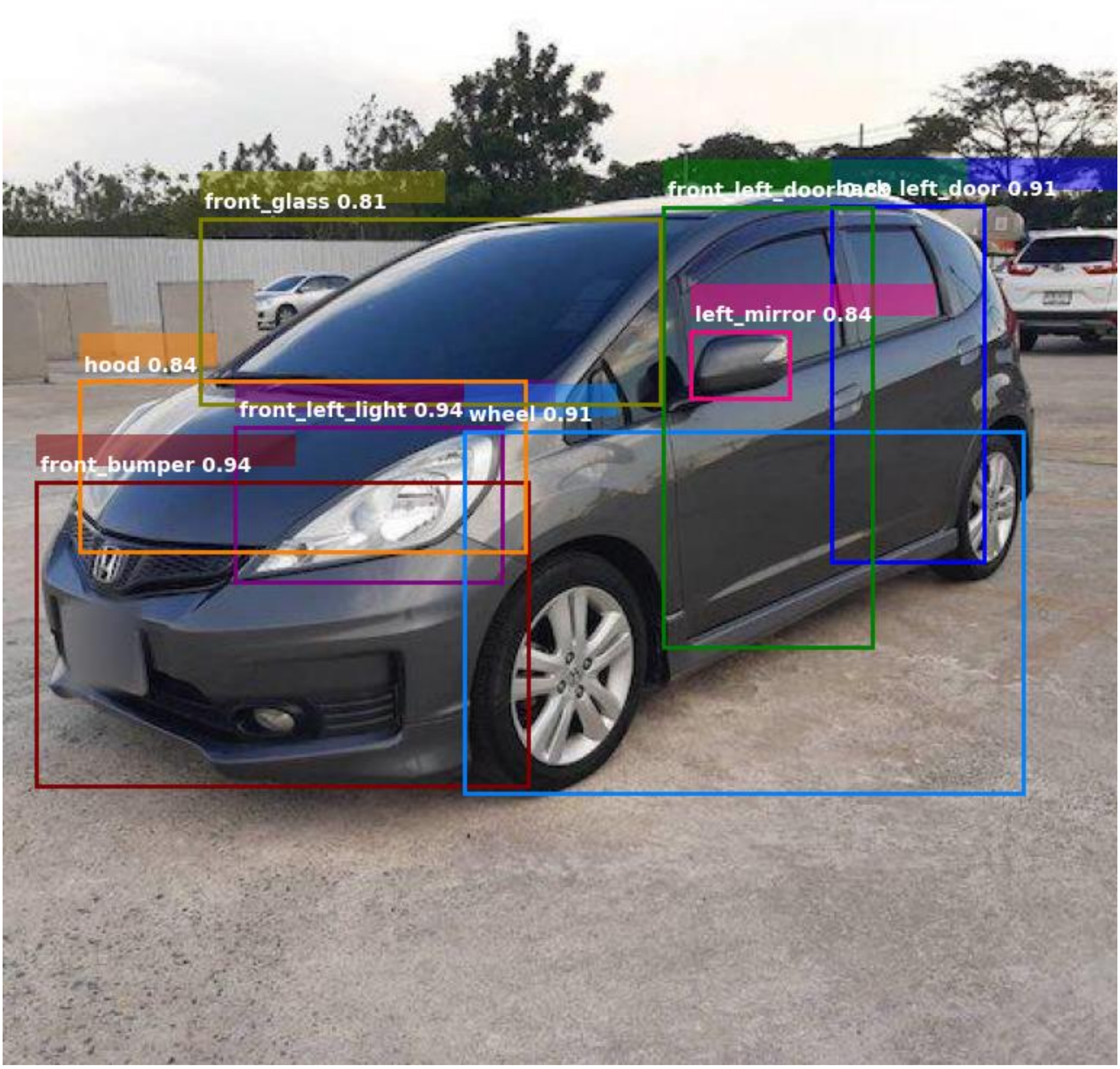
1. What is the MaP@[IoU=50] and MaP@[IoU=75] for the testing dataset?

   The model's performance is evaluated using mAP (Mean Average Precision) metrics. The results obtained are:
   **mAP@[IoU=50]**: 0.739
   **mAP@[IoU=50:95]**: 0.669
   These metrics indicate a strong detection performance, particularly for IoU thresholds of 50%. However, challenges remain for higher IoU thresholds.

2. Show the visualization of the predictions for any two random samples from the testing dataset. They must contain the boundingbox, class name and the confidence.

   Three random test images were visualized with predictions, including bounding boxes, class names, and confidence scores. The visualizations demonstrate the model's ability to detect and localize multiple car parts accurately, with minimal overlapping of bounding box texts.

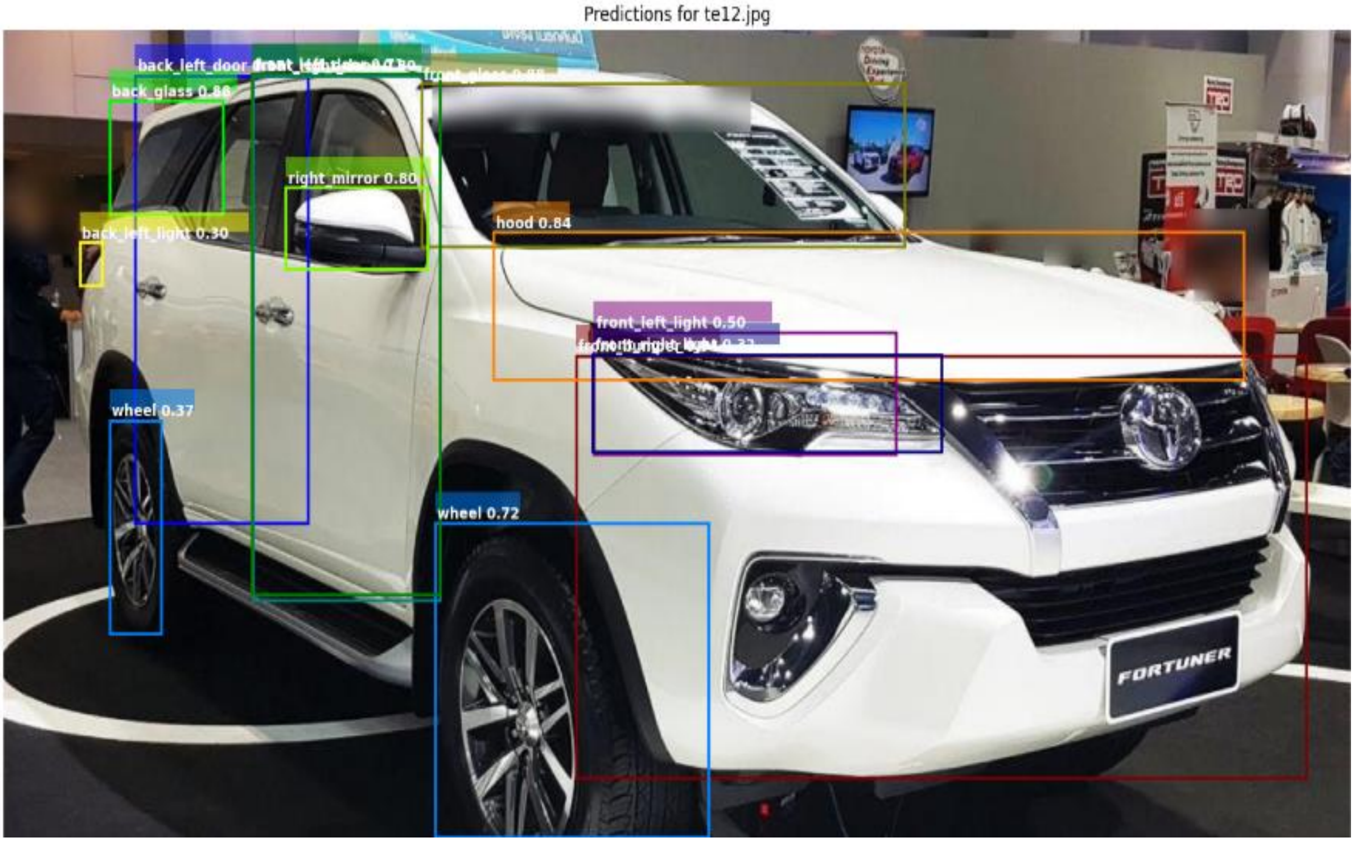
Predictions for car6.jpg



```
image 1/1 /content/car_parts_dataset/test/images/te43.jpg: 480x640 1 front_bumper, 1 front_glass, 2 front_right_lights, 1 hood, 2 left_mirrors, 1 right_mirror, 14.0ms
Speed: 3.9ms preprocess, 14.0ms inference, 1.6ms postprocess per image at shape (1, 3, 480, 640)
```

Predictions for te43.jpg


Predictions for te12.jpg



3. In your model, find out which category is very difficult to correctly localize? You must devise your own strategy to find out answer for this and explain it in details.

   In my analysis of localization difficulty, certain classes were harder to detect accurately. The devised strategy based on detection confidence and error rates helped me in identifying the most challenging classes which are:

   **front_left_light(9)**: Difficulty Ratio = 1.00
   **back_right_door(4)**: Difficulty Ratio = 1.00

   These classes often exhibited lower confidence scores and mis localized bounding boxes, possibly due to occlusions or insufficient training examples.

4. Can you suggest a solution to this problem, i.e, how to improve the detection accuracy for that class which is difficult to localize?

   To improve detection accuracy for challenging classes like front_left_light and back_right_door, there are some suggestions that are:

- **Data Augmentation**: to increase the number and diversity of samples for these classes through data augmentation techniques, such as rotations, lighting adjustments, and occlusion simulations.
- **Class-specific Fine-tuning**: to apply transfer learning to fine-tune the model specifically on underperforming classes.
- **Ensemble Methods**: to combine YOLO v8 with other detection models to improve robustness for difficult classes.
- **Adjusting loss weights**: to modify the class weights in the loss function to prioritize accurate detection of challenging classes.

## 5. Conclusion

This project helped me to draw insights into the application of YOLO v8 for car parts detection. While the model achieved high overall accuracy, it highlighted the importance of dataset quality and class-specific challenges. Implementing the above suggested solutions could further enhance the performance of the model.

## 6. Discussions and Collaborations (if any)

During this project, I took help from **Ishaan** to resolve issue related to the directory structure, and he advised me to split the training data into training and validation sets too. Additionally, I referred to AI models like ChatGPT and Gemini to obtain code snippets and debugging tips. But the overall code implementation and analysis was carried out independently.

----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------

**THANK YOU**