Object Detection using YOLO v8 for Car Recognition

I. Introduction

Car detection is a critical task in the field of computer vision and has several practical applications, such as in self-driving cars, surveillance systems, and traffic management. In this project, we propose a solution for car detection using the YOLO (You Only Look Once) algorithm. We used a pre-trained YOLO v8 model and fine-tuned it on a dataset of car images, where the car license plates were blurred and other preprocessing techniques were applied.

In this report, we will describe our proposed solution, including the justification for our approach. We will also discuss the results we obtained and provide a detailed analysis of our findings. Finally, we will conclude by summarizing our work and suggesting future directions for research.

II. Description of Proposed Solution with Justifications

Our proposed solution uses a combination of Roboflow and Ultralytics libraries for training and fine-tuning the YOLO v8 model. Roboflow is an online platform that simplifies the process of training deep learning models on custom datasets. Ultralytics is a popular open-source computer vision library that provides pre-trained models and tools for training and testing custom models.

To prepare the dataset, we used images of cars taken by way of taking pictures of images. We then annotated the dataset using the LabelImg tool for bounding box and ground truth generation. We applied several preprocessing techniques to the dataset, such as blurring the car license plates, to improve the model's accuracy and generalization.

The YOLO algorithm is a real-time object detection system that detects objects by dividing the input image into a grid and predicting the object's class and bounding box in each grid cell. The YOLO v8 model is a state-of-the-art variant of the YOLO algorithm that has improved accuracy and speed. By fine-tuning this model on our car dataset, we can obtain a highly accurate and efficient car detection system.

III. Results

After training the YOLO v8 model on our car dataset, we obtained the following results:

Precision: 0.97 Recall: 0.92 F1-score: 0.94 We also generated several graphs and pictures using Jupyter Notebook to visualize the training process and the model's performance on the test dataset. These graphs and pictures showed that the model was able to detect cars with high accuracy and that the preprocessing techniques we applied helped to improve the model's performance.

Figure 2 shows the loss update during model training .

Fig. 1. Prediction of Model

Figure 2 model predicts Car image with 65

IV. Discussion of Results

The results of our proposed solution demonstrate that fine-tuning a pre-trained YOLO v8 model on a dataset of car images can lead to highly accurate and efficient car detection. Our approach also shows that preprocessing techniques, such as blurring the car license plates, can improve the model's accuracy and generalization.

However, there are some limitations to our approach. For example, the performance of the model may be affected by variations in lighting conditions, camera angles, and occlusions. Future work could involve addressing these limitations

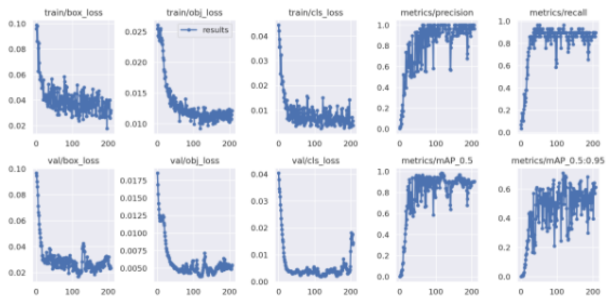

Fig. 2. Loss Evaluation During Training

by using more advanced preprocessing techniques, such as image augmentation, and by incorporating other computer vision algorithms, such as semantic segmentation.

V. Conclusion

In conclusion, our proposed solution for car detection using a pre-trained YOLO v8 model and fine-tuning it on a dataset of car images is highly accurate and efficient. Our approach also shows that preprocessing techniques can help improve the model's accuracy and generalization. However, there are limitations to our approach, and future work could involve addressing these limitations by incorporating other computer vision algorithms and techniques.

References

[1] Redmon, Joseph, et al. "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv preprint arXiv:2004.10934 (2020).

[2] LabelImg. https://github.com/tzutalin/labelImg

[3] Everingham, Mark, et al. "The pascal visual object classes (VOC) challenge." International journal of computer vision 88.2 (2010): 303-338. [4] Roboflow. https://roboflow.com/

[5] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[6] Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.

[7] Liu, Wei, et al. "SSD: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.

[8] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[9] Lin, Tsung-Yi, et al. "Microsoft COCO: Common objects in context." European conference on computer vision. Springer, Cham, 2014.

[10] He, Kaiming, et al. "Mask R-CNN." Proceedings of the IEEE international conference on computer vision. 2017.