License Plate Detection and Recognition

- Kirti Deepak Kshirsagar
- Prathiksha Mohan Raje Urs
- Shrushti Agarwal

GitHub Link: https://github.com/kirti-kshirsagar/License-Plate-Detection-and-Recognition

Introduction:

The primary challenge our project addresses is the accurate detection and recognition of vehicle license plates in various conditions. This problem is significant due to its wide range of applications in traffic management, law enforcement, parking automation, and toll collection. Efficient license plate recognition can enhance security, streamline traffic flow, and automate processes that currently require manual intervention.

This problem is intriguing as it intersects advanced computer vision techniques and practical real-world applications. By solving this, we can improve traffic monitoring, enhance security measures, and assist in automating vehicular management tasks. It finds use in smart city infrastructures, at border crossings, in parking lots, and by law enforcement agencies for monitoring and security purposes.

The approach involves using YOLOv8 for vehicle and license plate detection, followed by EasyOCR for optical character recognition on the detected plates. YOLOv8, being a robust and fast object detection model, is suitable for real-time detection in various lighting and angle conditions. EasyOCR complements this by extracting alphanumeric characters from the plates accurately.

Experiment Setup:

The dataset underpinning our project, sourced from the "License Plate Detector Image Dataset" on Roboflow, comprises vehicle images showcasing number plates. The dataset is meticulously organized into three distinct folders—train, test, and validation, and comprises a total of 395 images.

Our training data accounts for 70% of the dataset, with 20% allocated for validation and 10% for testing. These images have undergone preprocessing, including auto-orientation adjustments and resizing to dimensions of 640x640, ensuring standardized inputs for training.

The models used for training are YOLOv8n and YOLOv8x and are trained for 100 epochs. The whole project has been executed on T4 GPU using Google Colab.

The primary objective is to accurately detect and recognize vehicle number plates from various images. This involves two main processes: number plate detection and number plate recognition. Initially, YOLOv8 is employed for the detection part, where it identifies the vehicle's number plate from an image. YOLOv8 is chosen for its efficiency and accuracy in object detection tasks. Once the number plate is detected, EasyOCR comes into play for the recognition part. This tool is responsible for accurately reading and extracting the alphanumeric characters present on the number plate.

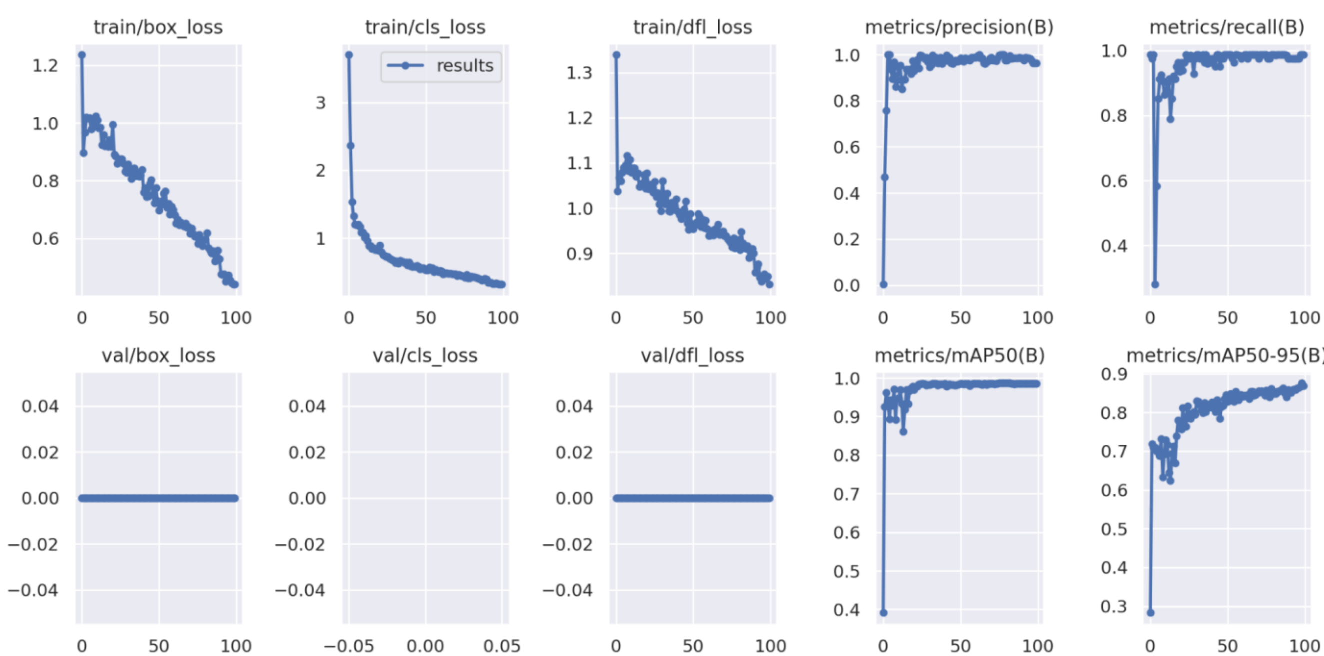
YOLOv8 employs a deep CNN as its backbone. This network structure is designed to extract features from the input images efficiently. CNNs are known for their ability to learn spatial hierarchies of features, which is crucial for detecting objects like license plates in various sizes and orientations. YOLOv8 utilizes anchor boxes to predict the bounding boxes for the objects (license plates in this case). These anchor boxes are predefined and are adjusted during training to fit the shapes and sizes of the target objects in the dataset. Each anchor box is associated with an objectness score indicating the likelihood that the box contains an object of interest. Along with the objectness score, the model also predicts the class probabilities for the detected objects. In the context of license plate detection, the primary class would be the vehicle number plates.

After detecting the license plates using YOLOv8, EasyOCR is used for optical character recognition (OCR). EasyOCR is a tool that converts images of text into machine-readable text, essential for extracting information from license plates i.e. character recognition.

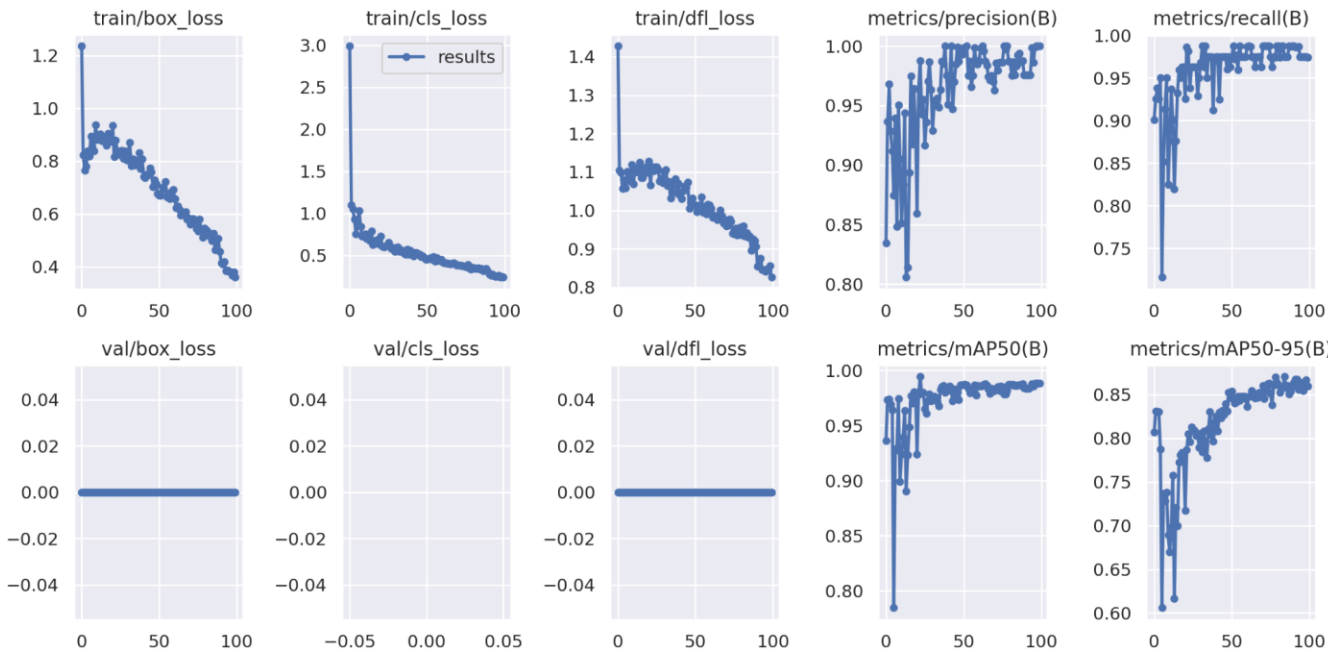
Experiment Results:

We have performed using two YOLOv8 models i.e. YOLOv8n and YOLOv8x. The YOLOv8n is the smallest and fastest variant, but it is also the least accurate. The YOLOv8x is the largest and slowest variant, but it is also the most accurate. So we have trained both of the models on the vehicle number plate dataset as mentioned above, for 100 epochs at first.

- When we used the YOLOv8n model for training we got the following results-



- When we used the YOLOv8x model for training we got the following results-



Analysis -

➔ Loss Metrics: The training loss for box, classification, and direction/rotation, all show a downward trend as training progresses, which suggests the model is learning and improving its ability to predict with each epoch. The validation losses appear to be low and stable, indicating good generalization to unseen data.

➔ Precision and Recall: The precision and recall graphs are relatively high, indicating the model's good performance in identifying relevant instances (precision) and the proportion of actual positives identified correctly (recall).

➔ Mean Average Precision (mAP): The mAP at IoU threshold 0.50 (mAP50) and across IoU thresholds from 0.50 to 0.95 (mAP50-95) are both high. This suggests that the model is consistent in detecting objects accurately across different intersections over union (IoU) thresholds. The high mAP values indicate that the model is reliable in its predictions.

➔ Stability Over Time: The metrics remain stable or improve without significant volatility as epochs increase, which implies that the training process is stable and the model is not experiencing high variance or overfitting issues.

- When these models were applied on the dataset for prediction, we got the output as follows-

Fig (a)

Fig(b)

Figure (a) comes from when we train the dataset using the YOLOv8n model whereas Figure(b) comes from when we train the dataset using the YOLOv8x model.
As we can see the difference in the prediction, we can infer that YOLOv8x performs better than the YOLOv8n model.
Our project also works on videos. YOLOv8 is an advanced deep-learning model for object detection and operates on videos by processing individual frames to detect and identify objects. Videos are composed of a series of frames (images). The first step in using YOLOv8 on videos is to extract these frames. Each frame is then processed individually, allowing the model to detect objects in each frame as it would in a static image. For each extracted frame, YOLOv8 performs object detection. It scans the entire image and identifies objects based on the trained categories. YOLOv8 is known for its speed and accuracy, which is crucial in video processing where real-time or near-real-time detection is often required.
We have included the video results before performing object detection and recognition on it and after the processing has been completed on the GitHub.

Improvements:

- Although YOLOv8 provides high accuracy in object detection, it can be further improved for varying lighting conditions, plate obfuscation, and diverse plate designs across different regions. Advanced algorithms or hybrid models may enhance accuracy.
- While EasyOCR is effective, its accuracy in character recognition can vary based on font styles, sizes, and image quality. Implementing more robust OCR techniques or fine-tuning the model for specific license plate fonts and styles could improve performance.

- For applications requiring real-time analysis, such as traffic monitoring and law enforcement, the speed of detection and recognition can be critical. Optimizing these models for faster processing without sacrificing accuracy is a key area for improvement.
- The current systems may struggle with angled, distant, or partially obscured plates. Developing algorithms that can accurately detect and interpret plates under these conditions is necessary.
- We could also work on a larger dataset and train for more epochs, but we faced computational limitations.

Conclusion:

In summary, this project has successfully demonstrated the viability and effectiveness of applying deep learning techniques for the detection and recognition of vehicle license plates. Through meticulous analysis and testing on various vehicle images, including challenging scenarios, the system has shown a high level of accuracy in locating and extracting license plate information. This achievement underscores the potential of deep learning in solving complex real-world problems, such as automatic license plate recognition. While the system performed well on the dataset, including front images of national license plates, future work could involve expanding the dataset to include more diverse and challenging scenarios, enhancing the robustness of the system. This project not only contributes significantly to the field of computer vision and automated vehicle monitoring but also lays a foundation for further research and development in this domain.

References:

[1] Ultralytics YOLOv8 - https://docs.ultralytics.com/models/yolov8/

[2] Pyimagesearch - https://pyimagesearch.com/2020/09/14/getting-started-with-easyocr-for-optical-character-recognition/

[3] Roboflow - https://public.roboflow.com/