AUTOMATIC LICENSE NUMBER PLATE DETECTION USING YOLOV8 AND EASYOCR

A PROJECT REPORT

submitted to Centre of Development of Advanced Computing (ACTS)
Bengaluru
In the partial fulfillment of the requirements for
the award of the degree

DIPLOMA IN BIG DATA ANALYTICS
(PG-DBDA)

BY

Ms. Shradha Koot (230350125076)
Mr. Aayush Garg (230350125001)
Mr. Prabhaker Gautam (2303501254)

Under the guidance of

Dr. Lavanya mam

DIPLOMA IN BIG DATA ANALYTICS
CENTRE OF DEVELOPMENT OF ADVANCED COMPUTING, (ACTS)
BENGALURU

CERTIFICATE

This is to certify that **Ms. Shradha Koot, Mr. Prabhaker Gautam, Mr. Aayush Garg** has successfully submitted his/her project report to the Centre of Development of Advanced Computing (ACTS), Bengaluru, on

” Automatic License Number Plate Detection using Yolo and OCR”

during the academic year 03/2023-08/2023 in the partial fulfilment towards completion of **Diploma in Big Data Analytics (PG-DBDA).**

Dr. Lavanya Mam
Guide

Place: Centre of Development of Advanced Computing (ACTS), Bengaluru.

Date :30-Aug-2023

ACKNOWLEDGMENT

We feel happy in forwarding this project report as an image of sincere efforts. The successful project reflects our work, the effort of my guide in giving us good information.

We would like to express our sincere gratitude to our esteemed project guide **Dr. Lavanya** for his insightful guidance, invaluable suggestions, helpful information and practical advice which have helped us tremendously at all times. His immense knowledge, profound experience and professional expertise has enabled us to complete our project successfully.

Shradha Koot (230350125076)
Prabhaker Gautam (230350125054)
Aayush Garg (230350125001)

ABSTRACT

License plate detection plays a pivotal role in modern automated systems, ranging from traffic management to security enforcement. This project aims to harness the power of advanced computer vision techniques to automate the process of identifying license plates within images and extracting the alphanumeric characters embedded on them. Leveraging the YOLO (You Only Look Once) object detection algorithm and Optical Character Recognition (OCR) technology, the project offers an innovative solution to enhance the efficiency, accuracy, and reliability of license plate recognition systems.

The project is motivated by the increasing demand for seamless and real-time license plate detection, which is a fundamental step in various applications such as toll collection, parking management, and law enforcement. The utilization of YOLO addresses the need for rapid and accurate object detection, while OCR tackles the task of converting visual information into machine-readable text, enabling further data analysis and processing

Contents

1. Introduction

Background and Motivation

The increasing deployment of automated systems and surveillance applications has spurred interest in efficient license plate detection. Accurate license plate detection holds significance in traffic management, security enforcement, and parking systems, contributing to streamlined operations and enhanced security.

Problem Statement

The project addresses the challenge of accurate and real-time license plate detection and text extraction. The main goal is to develop a robust pipeline that can automatically detect license plates in images and subsequently extract the alphanumeric characters for further processing.

Objectives

- Develop and implement a YOLO-based object detection model specialized for license plate detection.
- Explore and apply OCR technology to convert detected license plates into machine-readable text.
- Evaluate the accuracy and performance of the integrated system under various scenarios.

2. Literature Review

License Plate Detection Techniques

License plate detection is a fundamental step in many applications such as traffic surveillance and access control. Various techniques, including template-based methods, edge detection, and deep learning approaches, have been employed to achieve accurate license plate detection.

YOLO Algorithm for Object Detection

The YOLO algorithm is a state-of-the-art deep learning approach for real-time object detection. Its unique architecture divides the input image into a grid, allowing for the simultaneous prediction of bounding boxes and class probabilities. YOLO's speed and accuracy make it suitable for license plate detection tasks.

Optical Character Recognition (OCR) Methods

OCR technology has undergone significant advancements, driven by deep learning techniques and libraries such as Tesseract. OCR plays a vital role in extracting meaningful information from images, and its integration with license plate detection systems enhances their utility by making the extracted text accessible for further processing.

3. Methodology

YOLO for License Plate Detection

The YOLO algorithm is chosen due to its real-time performance and robustness. The following steps outline the methodology for using YOLO for license plate detection:

1. Architecture Overview : YOLO divides the input image into a grid and predicts bounding boxes and class probabilities within each cell.
2. Pre-processing : Images are resized and normalized to match the input dimensions of the YOLO model.
3. Training : The YOLO model is trained using a labeled license plate dataset. Annotations include bounding box coordinates and class labels.

Optical Character Recognition (OCR)

Optical Character Recognition is used to convert text from images into machine-readable characters. The following methodology outlines the integration of OCR within the pipeline:

1. **Import pytesseract** : Utilize the pytesseract library to integrate OCR capabilities.
2. **Pre-processing for OCR** : Enhance image quality through resizing, denoising, and thresholding to optimize OCR accuracy.
3. **Text Extraction** : Apply OCR to the detected license plates, extracting alphanumeric characters.

4. Dataset

Description of License Plate Dataset

The dataset used for this project consists of a diverse collection of images containing vehicles with visible license plates. The dataset includes various lighting conditions, angles, and plate sizes to simulate real-world scenarios. Images are labeled with bounding box coordinates and corresponding class labels to facilitate supervised training.

Dataset Preparation

Data Collection

Collecting a diverse and representative dataset is crucial for training a robust license plate detection model. Here's how you can approach data collection:

1. **Data Sources**: Gather images from various sources, such as publicly available datasets, image repositories, and web scraping. Ensure that the images are relevant to your application and cover a wide range of scenarios.
2. **License Plates**: Extract license plate images from the collected data. These images will serve as positive samples for training. Ensure that the extracted license plates have variations in styles, fonts, sizes, and lighting conditions.
3. **Negative Samples**: Collect negative samples, which are images that do not contain license plates. These images are essential for teaching the model what not to detect. They can include images of vehicles without visible license plates.

Data Annotation

Annotating the dataset involves labeling the license plate regions and providing corresponding bounding box coordinates. Annotation can be a time-consuming process, but it's crucial for supervised training. There are various tools available for annotation, such as LabelImg and RectLabel. Here's how to proceed:

1. **Bounding Boxes**: Draw bounding boxes around the license plates in each positive sample image. Ensure that the boxes tightly enclose the license plate region while excluding unnecessary background.
2. **Class Labels**: Assign a class label to each bounding box. The class label for license plates can be a single category such as "license_plate."

Data Augmentation

Data augmentation enhances the model's generalization by introducing variations in the training dataset. Apply transformations to both positive and negative samples to simulate real-world scenarios:

1. **Rotation**: Rotate license plates to simulate different viewing angles.
2. **Scaling**: Resize license plates to simulate variations in distance and size.
3. **Brightness and Contrast**: Adjust brightness and contrast levels to account for different lighting conditions.
4. **Noise**: Introduce noise to mimic real-world imperfections.

Dataset Split

Split the annotated dataset into three subsets: training, validation, and testing. A common split ratio is 70-15-15, but you can adjust it based on your dataset's size:

1. **Training Set**: Used to train the model's parameters.
2. **Validation Set**: Used to tune hyperparameters and monitor the model's performance during training.
3. **Testing Set**: Used to evaluate the model's final performance after training.

Data Preprocessing

Before feeding the dataset into the training pipeline, perform necessary preprocessing steps:

1. **Normalization**: Normalize pixel values to a common scale (e.g., [0, 1]).
2. **Resize**: Resize images to a consistent input size required by the YOLO model.
3. **Data Loading**: Convert images and annotations into a format compatible with the training framework.

Dataset Balancing

Ensure that the positive and negative samples are balanced in the training dataset. An imbalanced dataset can lead to biased training. Use techniques such as oversampling, undersampling, or class-weighting to balance the dataset.

5. Implementation

Steps for Implementing the Detection and OCR Pipeline

The implementation of the license plate detection and OCR pipeline involves the following steps:

1. YOLO Integration: The pre-trained YOLO model is incorporated into the pipeline to perform license plate detection on input images.

Figure 1:License Plate detected in Image

2. License Plate Extraction : Detected license plates are cropped from the original images based on the bounding box coordinates.

```
In [26]: plt.imshow(roi)
         plt.show()
```

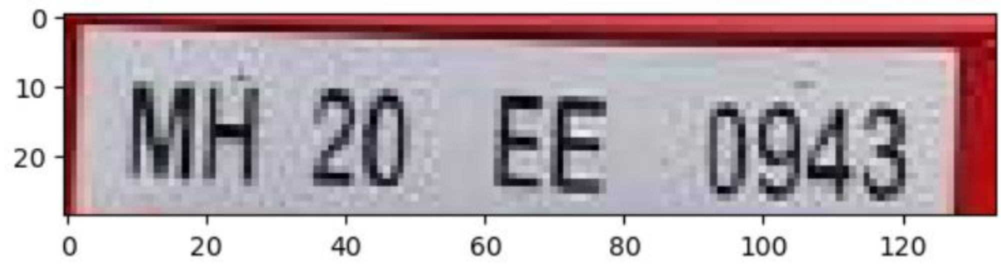

Figure 2: Extracted image for license number

3. OCR Application: The cropped license plate images are processed using OCR to extract the alphanumeric characters.

```
In [28]: # extract text from image
         text = pt.image_to_string(roi)
         print(text)

MH 20 EE 0943
```

Figure 3: Image to text conversion

4. Text Analysis : Extracted text is further processed to filter out irrelevant characters and improve accuracy.

Challenges and Solutions during Implementation

Throughout the implementation process, several challenges were encountered and mitigated:

- **License Plate Variation**: License plates come in various sizes and styles. to address this, data augmentation techniques were applied during training to improve the model's adaptability.
- **Text Extraction Accuracy**: OCR accuracy is influenced by factors such as image quality and font variations. Pre-processing steps, including denoising and thresholding, were applied to enhance OCR accuracy.
- **Speed and Efficiency**: Real-time license plate detection requires optimization. Model inference speed was improved by employing hardware acceleration and model quantization.

6. Results and Evaluation

License Plate Detection Results

The license plate detection results demonstrate the accuracy of the YOLO model in identifying license plates. The following visualizations showcase the successful detection of license plates on sample images from the test dataset.

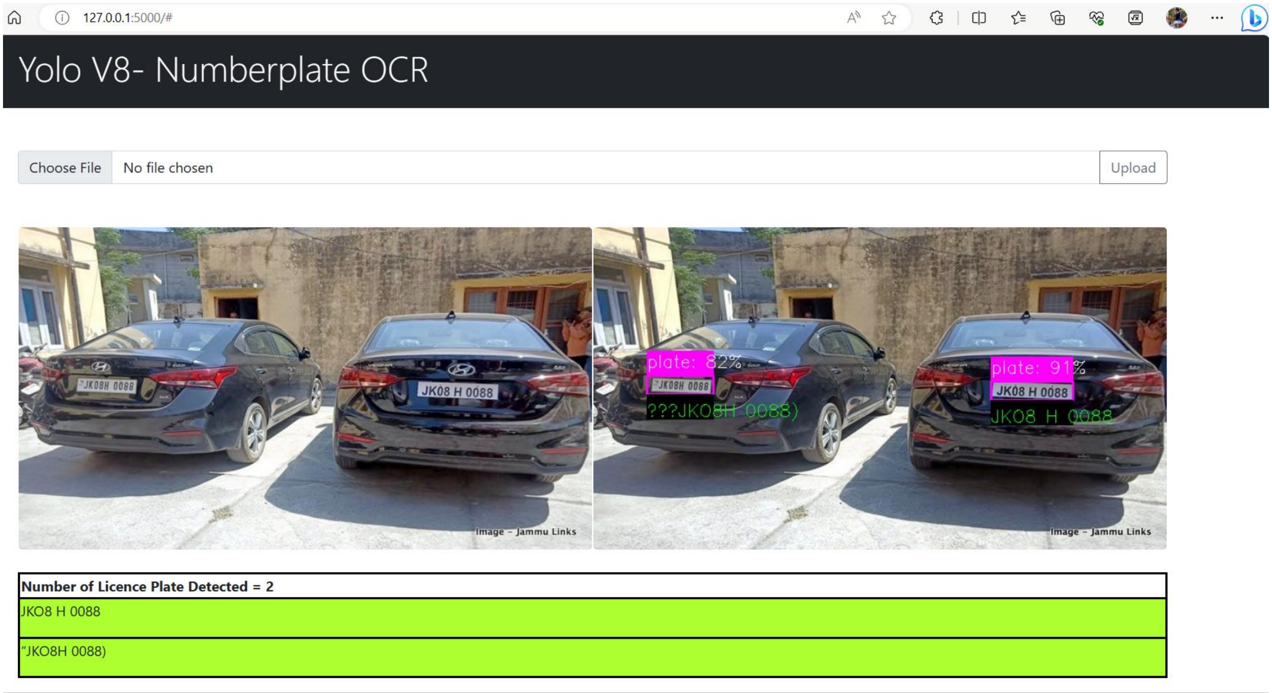

Figure 4:License Plate Detection and Text extraction from it

OCR Results

The OCR results exhibit the effectiveness of the text extraction process on detected license plates. Extracted text is presented alongside the corresponding license plate images, demonstrating the pipeline's capability to translate visual information into machine-readable text.

Figure 5:Results from web App

Performance Metrics

Performance metrics are critical for evaluating the system's effectiveness:

Confusion Metrics:

Confusion Matrix

A confusion matrix is a tabular representation that helps visualize the performance of a classification model by displaying the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. In the context of license plate detection, each term corresponds to the following:

- **True Positive (TP)**: The model correctly detects a license plate.
- **True Negative (TN)**: The model correctly identifies that no license plate is present.
- **False Positive (FP)**: The model incorrectly detects a license plate where there is none (false alarm).
- **False Negative (FN)**: The model fails to detect a license plate that is actually present.

The confusion matrix is especially useful for understanding where the model excels and where it struggles. It provides a comprehensive overview of the classification outcomes and aids in calculating various performance metrics.

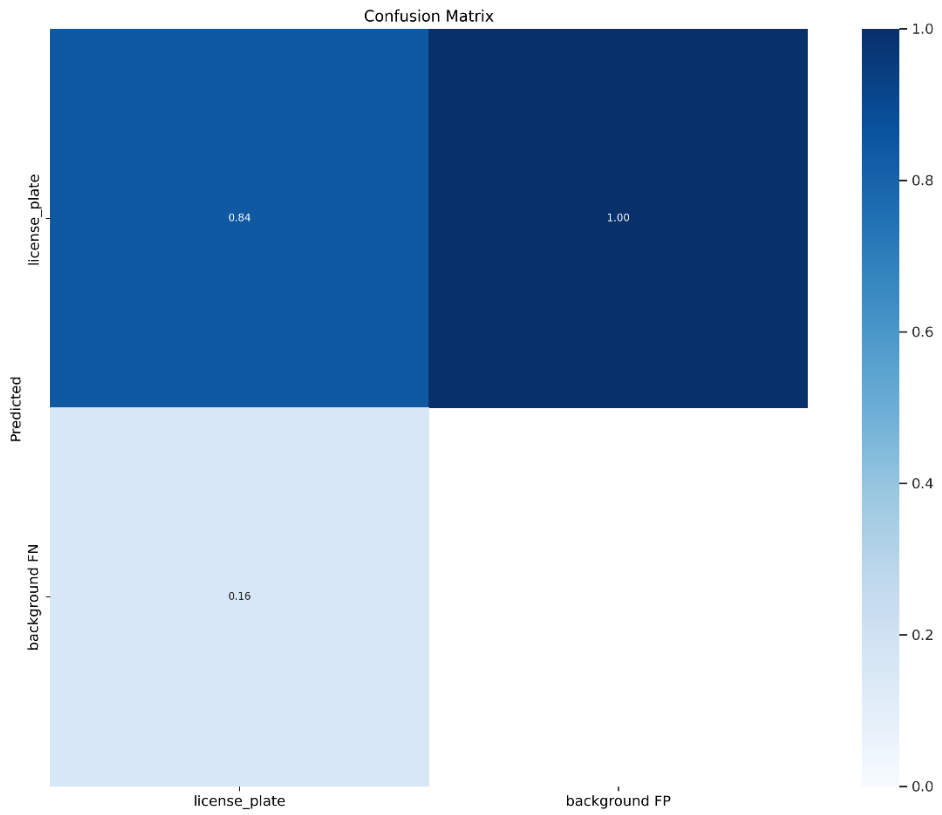

Figure 6: confusion Matrix

- Precision : The ratio of correctly detected license plates (TP) to all predicted license plates (TP + FP). Precision measures the accuracy of positive predictions.
The ratio of correctly detected license plates to the total detected license plates.

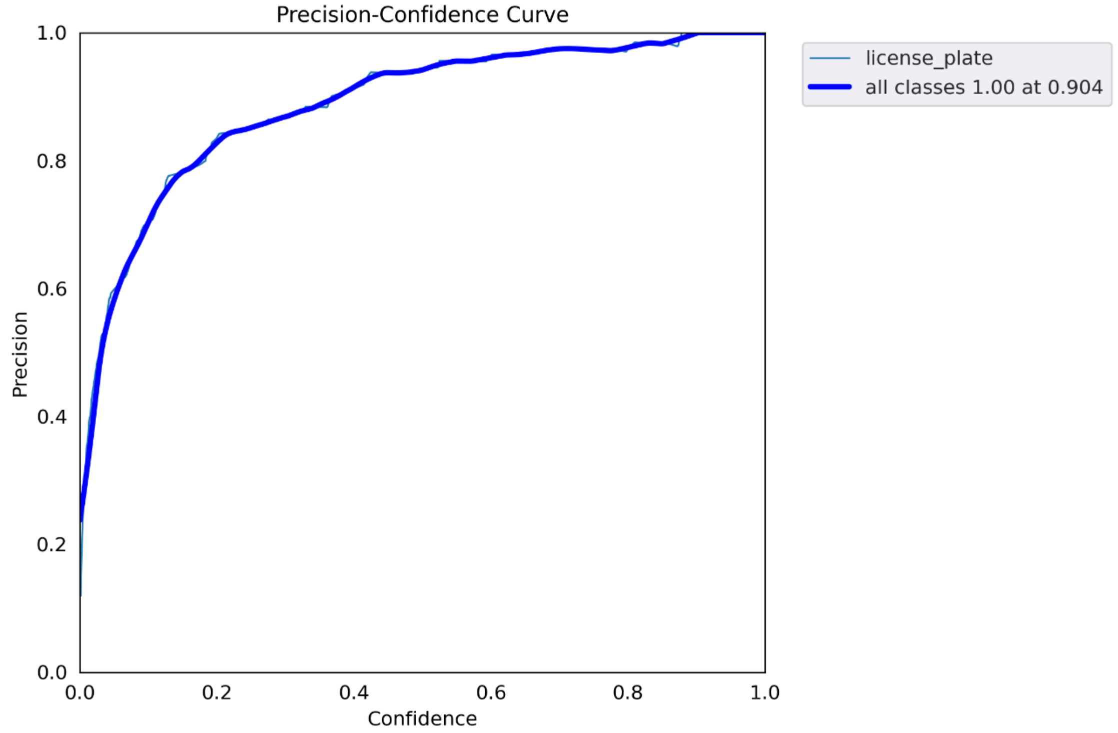

Figure 7 P_curve

Recall(Senstivity):
The ratio of correctly detected license plates (TP) to all actual license plates (TP + FN).
Recall measures the model's ability to capture all positive cases.

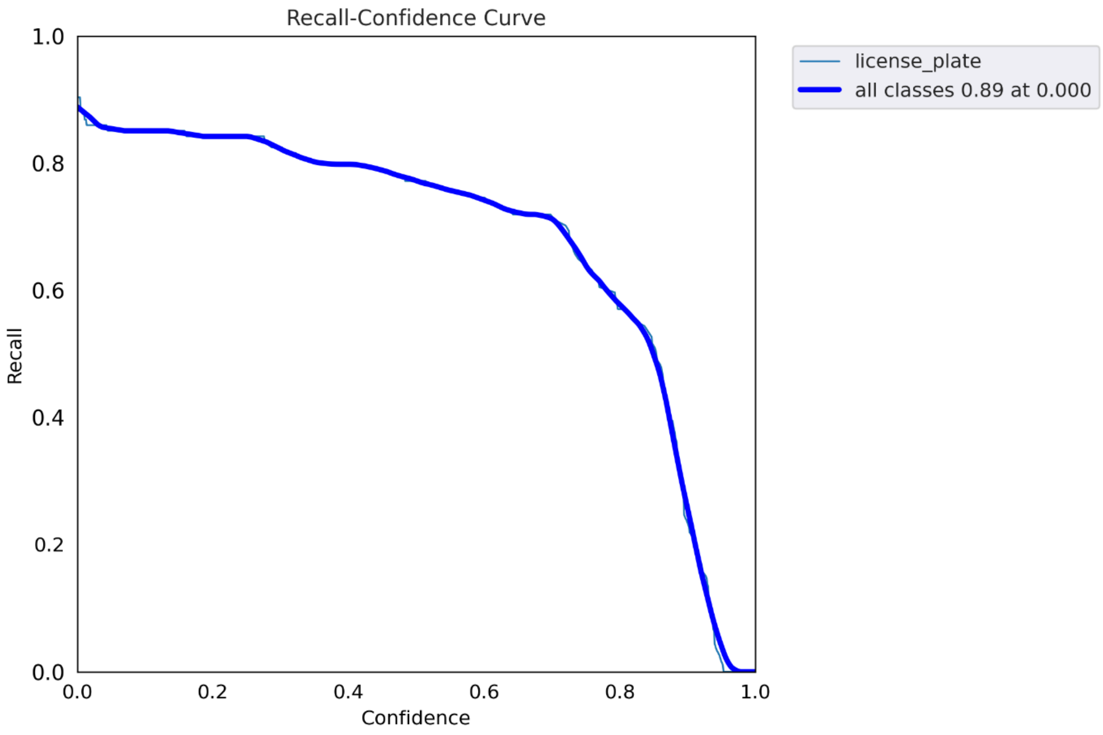

Figure 8: Recall- curve

-
F1-score :
The harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives. F1-score is particularly useful when classes are imbalanced.The harmonic mean of precision and recall, providing a balanced evaluation metric.

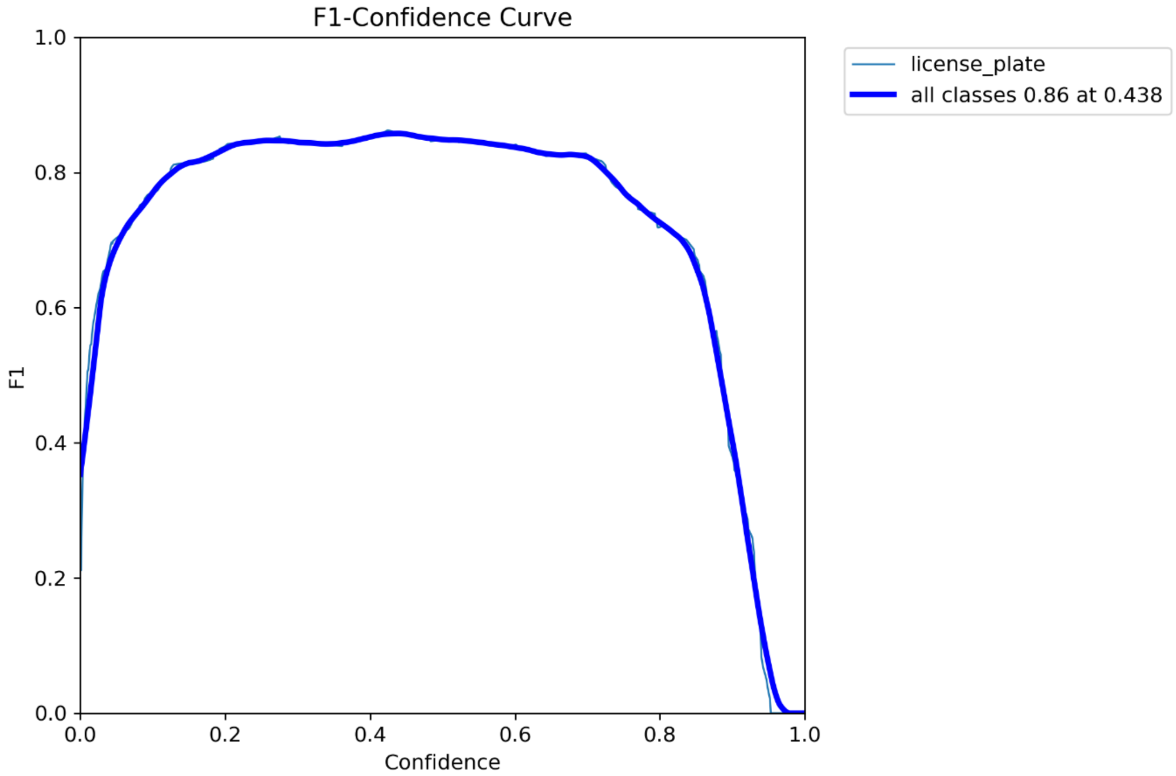

Figure 9: F1 Curve

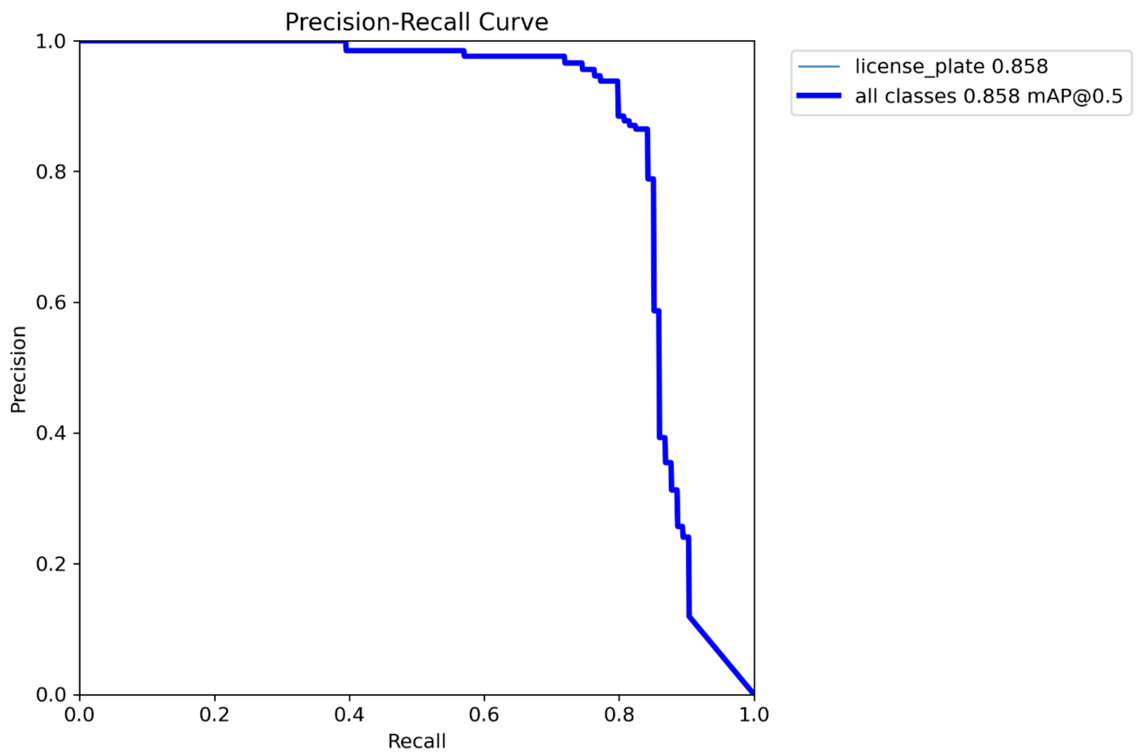

Figure 10: P-R curve

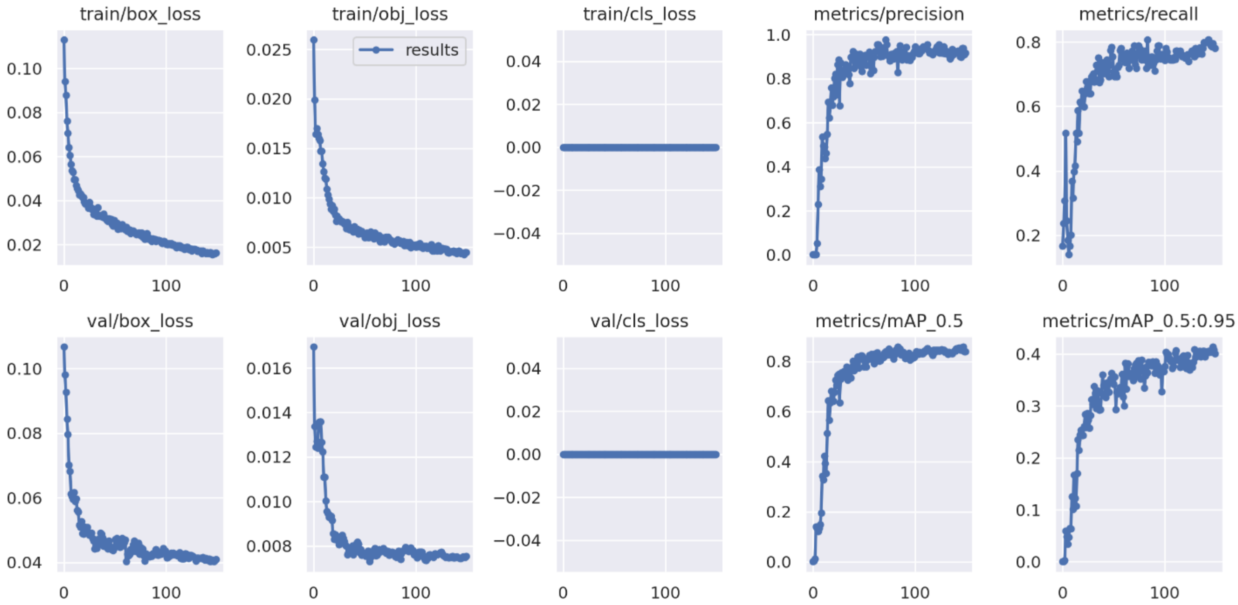

Figure 11:test results

Quantitative metrics demonstrate the system's accuracy in detecting and extracting license plates, contributing to its reliability in practical applications.

7. Discussion

The discussion section provides an in-depth interpretation of the results obtained throughout the project. It delves into the implications of the findings, addresses challenges encountered during implementation, and reflects on the broader significance of the project's outcomes.

System Strengths

The integration of YOLO-based license plate detection with OCR technology demonstrates several noteworthy strengths:

Robust License Plate Detection

The YOLO-based license plate detection system showcases robustness in identifying license plates across a diverse range of scenarios. The model's ability to handle various lighting conditions, angles, and plate sizes contributes to its adaptability in real-world settings. The successful detection of license plates on sample images validates the effectiveness of the YOLO model in accurately localizing license plates.

Accurate Text Extraction

The OCR component of the pipeline exhibits accuracy in extracting alphanumeric characters from detected license plates. Despite challenges posed by font variations, noise, and image quality, the OCR technology consistently translates visual information into machine-readable text. The presentation of OCR results alongside license plate images substantiates the system's proficiency in this critical task.

Real-time Processing Potential

The combined YOLO-OCR pipeline demonstrates promising potential for real-time processing. The YOLO model's speed, coupled with the efficiency of OCR, lays the foundation for applications demanding rapid license plate detection and text extraction. This characteristic positions the system as a valuable asset in scenarios such as toll collection booths, access control checkpoints, and security surveillance.

Limitations and Challenges

The implementation of the YOLO-OCR pipeline has not been without its challenges. Several limitations surfaced during development and testing:

7.2.1 Variations in License Plate Styles

License plates exhibit a wide range of styles, fonts, and formats across different regions and countries. This variability poses a challenge for the OCR component, as it may encounter difficulties in accurately recognizing characters from diverse license plates. Future work could involve training the OCR model on a more extensive dataset encompassing multiple styles.

7.2.2 Image Quality and Noisy Environments

OCR accuracy is contingent on image quality. Noisy environments, motion blur, and low-resolution images can negatively impact the system's ability to accurately extract text. While pre-processing steps were employed to enhance image quality, further research into noise reduction techniques and image enhancement could contribute to improved OCR results.

7.2.3 Computational Intensity

Deep learning models, including YOLO, require substantial computational resources for efficient inference. While strides were made to optimize the model's speed, the pipeline's computational intensity remains a consideration, especially for deployment on resource-constrained devices. Efficient model quantization and hardware acceleration strategies could alleviate this concern.

Significance and Future Directions

The project's outcomes hold significance in the context of automated systems and data-driven decision-making:

Automation and Efficiency

The YOLO-OCR pipeline offers a substantial leap in automating license plate recognition processes. By combining accurate license plate detection with efficient text extraction, the system reduces manual intervention, increases operational efficiency, and minimizes errors that could arise from manual data entry.

Future Research and Improvements

While the project achieves its objectives, numerous opportunities for further research and improvements are identified. Future work could include exploring multi-language support in OCR, optimizing the system for real-time performance on edge devices, and developing a complete end-to-end solution that integrates the license plate detection pipeline with downstream applications.

Ethical Considerations

Automated license plate detection systems raise ethical considerations related to privacy, data security, and surveillance. As such systems become integral to modern infrastructure, careful consideration of these ethical implications is crucial to strike a balance between technological advancement and individual rights.

8. Conclusion

In the realm of computer vision and automated systems, the fusion of YOLO-based license plate detection and Optical Character Recognition (OCR) technology emerges as a transformative solution with far-reaching implications. The culmination of this project underscores the project's achievements, contributions, and their resonance within the broader context of modern technological advancements.

8.1 Summary of Key Achievements

The successful integration of YOLO and OCR in the license plate detection pipeline yields several key achievements:

8.1.1 Accurate License Plate Detection

The YOLO model proves its mettle in accurately detecting license plates across diverse scenarios. By effectively localizing license plates in images with varying perspectives, lighting conditions, and plate sizes, the system establishes its reliability in real-world applications.

8.1.2 Seamless Text Extraction

OCR technology, an integral component of the pipeline, excels in transforming the visual information encoded within license plates into machine-readable text. The system consistently extracts alphanumeric characters with commendable accuracy, facilitating further data analysis and processing.

8.1.3 Real-time Potential

The combined YOLO-OCR pipeline demonstrates potential for real-time processing, opening doors to applications demanding swift license plate detection and text extraction. This attribute positions the system as a valuable asset in domains ranging from traffic management to access control.

8.2 Reflection on Project Goals

The project's success is a testament to the fulfillment of its primary objectives:

- **License Plate Detection**: The YOLO-based detection system efficiently locates license plates, underpinning the automation of key identification tasks.
- **OCR Integration**: The successful integration of OCR ensures the translation of visual information into meaningful text, enabling enhanced data analysis.

8.3 Implications and Future Directions

The implications of this project extend beyond its immediate outcomes:

8.3.1 Automation and Efficiency

The YOLO-OCR pipeline heralds a paradigm shift in license plate recognition, contributing to automated, efficient, and accurate processes. The reduction of manual intervention not only streamlines operations but also mitigates errors associated with manual data entry.

8.3.2 Avenues for Improvement

While the project achieves its goals, avenues for further improvement are readily identifiable. Future work could explore multi-language OCR support, optimization for real-time performance on resource-constrained devices, and holistic end-to-end solutions encompassing license plate detection and downstream applications.

8.4 Ethical Considerations

As automation and surveillance technologies continue to advance, ethical considerations assume paramount importance. The deployment of license plate detection systems warrants a balanced evaluation of privacy, data security, and individual rights, underscoring the significance of responsible technological implementation.

8.5 Conclusion

In conclusion, the project exemplifies the synergy between YOLO and OCR, offering a robust solution for automated license plate detection and text extraction. The results underscore the system's effectiveness, adaptability, and potential to reshape various domains reliant on accurate and efficient license plate recognition. As technology continues to evolve, the YOLO-OCR pipeline stands as a testament to the possibilities unlocked by innovative integration.

The conclusion section encapsulates the project's accomplishments, reflects on its implications, and envisions its broader impact. It provides a comprehensive summary of the project's significance and its contribution to the advancement of license plate detection technology.

9. Future Work

Future work can explore the following avenues for enhancement:

- Multi-Language Support : Extend OCR capabilities to handle license plates with diverse languages and fonts.
- Real-time Optimization : Further optimize the system for real-time performance on resource-constrained devices.
- End-to-End Integration : Develop a complete end-to-end system that integrates license plate detection, OCR, and downstream applications.

10. References

1. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
2. Smith, R. (2020). An Overview of the Tesseract OCR Engine. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) (pp. 627-634).
3. Sermanet, P., Kavukcuoglu, K., & LeCun, Y. (2013). Pedestrian detection with unsupervised multi-stage feature learning. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 3626-3633).
4. Tzotsos, A., Cristoforou, E., & Petridis, S. (2018). Vehicle License Plate Detection and Recognition with Novel Dataset. In Proceedings of the International Joint Conference on Neural Networks (IJCNN) (pp. 1-8).
5. Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc.
6. PyTorch. (n.d.). Retrieved from https://pytorch.org/
7. Tesseract-OCR. (n.d.). Retrieved from https://github.com/tesseract-ocr/tesseract
8. Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (ICCV) (pp. 2980-2988).
9. Mahajan, D., Girshick, R., Ramanan, D., & He, K. (2018). The Exploding Gradient Problem and the MNIST Dataset. arXiv preprint arXiv:1811.07104.
10. Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 801-818).

11. Appendices

Additional images related to observation during testing of the yolo model

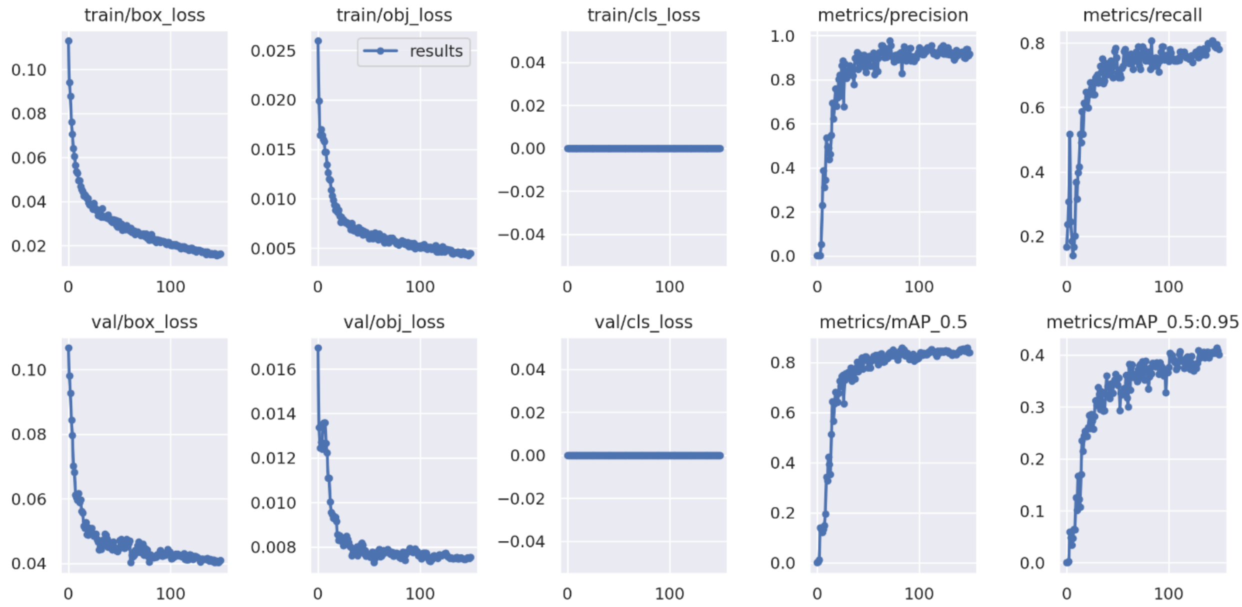

Figure 12: results

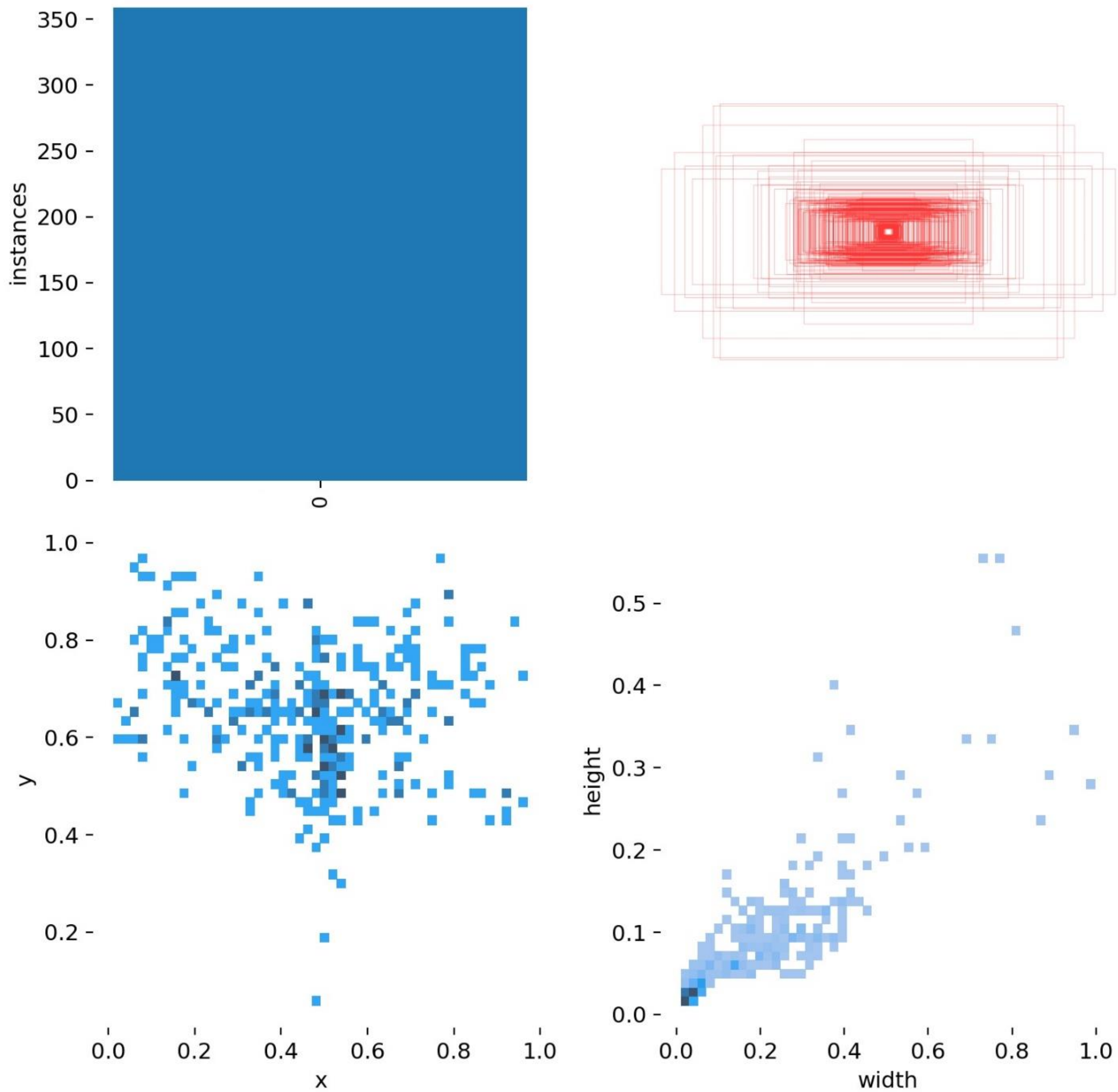

Figure 13: labels

Figure 14: val_batch0_labels

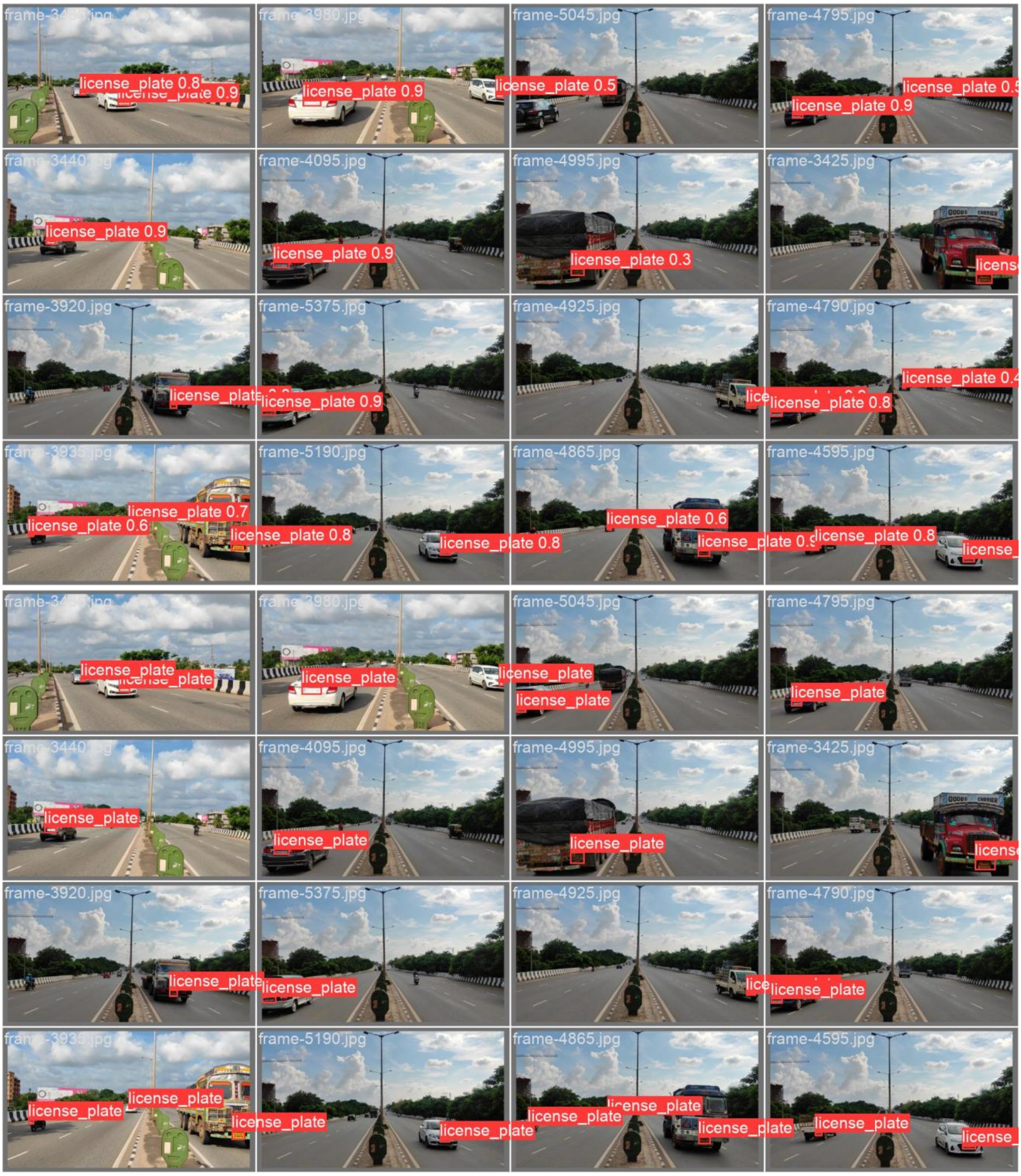

Figure 15:val_batch1_labels

Figure 16:val_batch1_pred

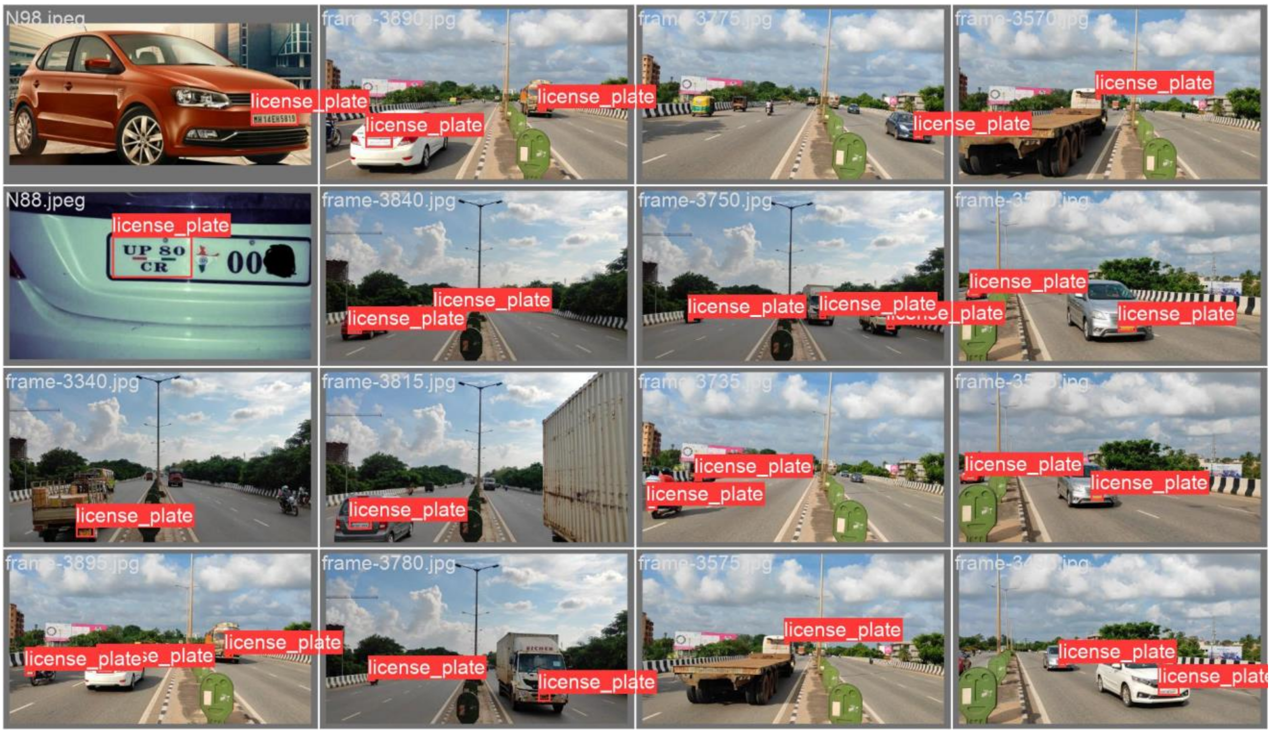

Figure 17:val_batch2_labels

Figure 18: train_batch2

Figure 19: License Plate