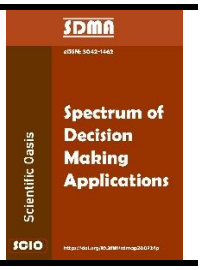
SCIENTIFIC OASIS

Spectrum of Decision Making and Applications

Journal homepage: www.dmap-journal.org
ISSN: 3042-1462



Estimation of Automatic License Plate Recognition Using Deep Learning Algorithms

Kattur Soundarapandian Ravichandran[1, *]

[1] Department of Mathematics, Amrita School of Physical Sciences, Amrita Vishwa Vidyapeetham, India

ARTICLE INFO

Article history:
Received 19 April 2024
Received in revised 20 September 2024
Accepted 3 October 2024
Available online 5 October 2024

Keywords:
Automatic License Plate Recognition, Preprocessing, Deep Learning, YOLO. Optical Character Recognition.

ABSTRACT

Automatic License Plate Recognition (ALPR) or Automatic Number Plate Recognition (ANPR) is the technology responsible for reading the license plates of a vehicle in an image or a video sequence using optical character recognition (OCR). With the latest advancements in deep learning and computer vision, these tasks can be done in a matter of milliseconds. Much work has been proposed in recent days for number plate identification through OCR using standard datasets available in open databases. However, in a real-time scenario, there are many external factors such as rain, heavy wind, and other conditions; the surface of the plates may be affected by sand or hair-like objects pasted on them. The resultant images are called noisy input images. In this paper, we apply various preprocessing techniques, such as noise removal algorithms and extended hair-removal algorithms, before applying deep learning algorithms, including the latest version of YOLO v8. Finally, it is concluded that the proposed model performs better than recently proposed state-of-the-art technologies.

1. Introduction

1.1 License plate recognition

An automated system called license plate recognition (LPR) uses optical character recognition (OCR) to read and interpret the characters on vehicle license plates. It uses cameras to take pictures of license plates, software algorithms to read and process those pictures, and databases to store and compare data on license plate numbers. Automating the process of reading license plates for multiple purposes is the primary goal of LPR. It has security and law enforcement uses, including locating stolen cars and detecting unregistered or underinsured vehicles. Access control, toll collection, and parking management systems are a few other applications for LPR (Figure 1). Recent advancements in LPR technology have substantially increased image quality, speed, and accuracy. It is now a vital tool for boosting productivity, security, and safety in a variety of industries and applications.

[*] *Corresponding author.*
E-mail address: ravichandran20962@gmail.com

https://doi.org/10.31181/sdmap21202512

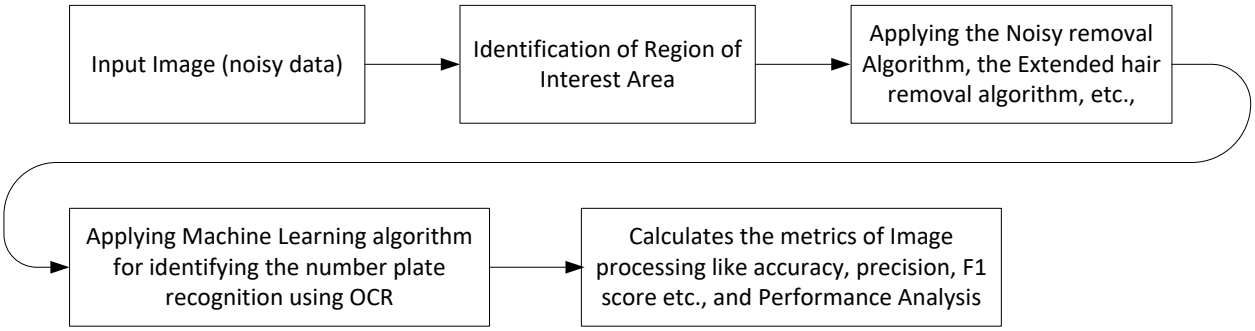


Fig. 1. Overview of the coarse-to-fine vehicle recognition procedure

Significant advancements have been made in LPR technology development over time. Hardware capabilities, OCR algorithms, and image-processing techniques have all advanced since the software's inception. Early LPR systems had character recognition accuracy, image quality, and lighting issues. However, LPR systems have substantially advanced with the development of computer vision and machine learning techniques. The use of cutting-edge technologies, such as neural networks and deep learning, has made license plate recognition more reliable and precise. Faster processing rates because of these developments enable real-time or almost real-time LPR applications. With improved efficiency and efficacy in license plate detection and data extraction, LPR technology has evolved into a crucial tool for many businesses (Table 1).

Table 1
Some applications of the LPR method

Source	Type of pre-processing technique used	Type of Deep Learning / Machine learning method used
Shashidhar *et al.,* [1]	Deblurring of the image using Weiner effect	You Only Look Once (YOLO) V3 for Region of Interest (ROI) detection and Convolutional Neural Network (CNN) for optical character recognition
Coetzee *et al.,* [2]	Niblack algorithm for thresholding	Neural network-based OCR system with character segmentation and dimension reduction technique.
Wang *et al.,* [3]	Image enhancement, image segmentation and localization	Faster R-CNN, YOLO, or SSD for object detection CNNs for Image Classification Deep SORT for Multi-Object Tracking CNNs and RNNs for License Plate Recognition
Roy and Ghoshal [4]	Pixel-based segmentation algorithm for alphanumeric character extraction from the license plate image.	Artificial Neural Network (ANN) for character recognition.
Fahmy [5]	Image processing techniques for extracting the location of characters in the number plate	BAM (Bidirectional Associative Memory) neural network for character recognition.
Selmi *et al.,* [6]	Adaptive thresholding, fine contours, geometric filtering	CNN model for LP detection and character recognition.
Zidouri and Deriche [7]	Vertical edge detection (Sobel filter), thickening mask, predefined conditions.	Neural networks (multilayer perceptrons) for character recognition.
Setiyono *et al.,* [8]	Data pre-processing is performed to enhance the quality of the number plate images, but the specific	YOLOv3 for object detection of number plates and character recognition, with Darknet-53 as the feature extractor

Source	Type of pre-processing technique used	Type of Deep Learning / Machine learning method used
	techniques used are not mentioned in the abstract.	
Shashirangana *et al.,* [9]	Image enhancement	CNNs, RNNs, Region-based CNN (R-CNN) and OCR
Baviskar *et al.,* [10]	Edge detection analysis is used to address obstacles in vehicle license plate detection	YOLO V5
Poorani *et al.,* [11]	Standard pre-processing techniques for preparing the car number plate dataset	YOLO V4 for Region of Interest (ROI) detection and Convolutional Neural Network (CNN) for optical character recognition
Sarfraz *et al.*, [12]	Vertical edge detection, filtering, matching, character extraction, normalization	Template matching, Hamming distance

1.2 You look only once (YOLO)

YOLO (You Only Look Once) is a well-known computer vision architecture that has transformed tasks involving object detection. In order to detect objects, YOLO uses a multi-layer convolutional network topology and a thorough methodology. YOLO scans the entire image once to find and identify items, in contrast to conventional methods that use sliding window techniques. This single-pass processing makes Real-time object detection possible with amazing speed and precision. Due to its design focus on regions of interest (ROI), YOLO needs help accurately recognizing small items. However, YOLO is a useful tool for further research or higher-resolution detection methods because of its success in locating ROIs. YOLO has been widely used in a variety of applications, such as license plate identification, due to its quickness and capacity for real-time object detection.

The combination of license plate recognition (LPR) systems and YOLO deep learning models offer a potent solution for precise and effective object detection and recognition. The accurate identification of license plates and other objects of interest can be made in real-time or nearly real-time by integrating YOLO into LPR systems. Prior to feeding the images into the YOLO model, pre-treatment methods were particularly created to deblur input images increase the image quality, and improve the sharpness and clarity of the pictures. This pre-processing step promotes more precise and trustworthy object recognition by minimizing the effects of blurriness.

This study also recommends investigating the possibilities of later versions of YOLO, like versions v5, v6, and v8, to enhance object detection performance. These iterations' review and analysis can shed light on the developments and improvements made in terms of precision, speed, and other important factors. The combination of LPR with YOLO expands the potential of license plate recognition systems by precisely and effectively detecting and recognizing objects using deep learning models and pre-processing methods.

2. Preprocessing techniques used

2.1 Blur removal

License Plate Recognition (LPR) is an essential technology used in various applications such as traffic surveillance, parking management, toll collection, and law enforcement. However, taking clear and accurate pictures of license plates can be difficult because of things like motion blur, defocus blur, and poor lighting. Blur removal techniques are used to make licence plate photos easier to read

in order to increase the performance of LPR systems. An overview of the blur removal techniques frequently employed in LPR projects is provided on this page.

2.1.1 Motion blur removal

Motion blur results when the camera or the item being watched moves during the exposure time. Several algorithms, including the following, have been developed to overcome this issue:

Deconvolution-based methods: These methods seek to recover the sharp licence plate image while estimating the motion blur kernel. In order to reconstruct the image, deconvolution techniques like Richardson-Lucy and Wiener deconvolution make use of the blur kernel.

Blind deconvolution: Blind deconvolution algorithms, in contrast to deconvolution-based techniques, simultaneously estimate the blur kernel and the latent image without being aware of the blur. This strategy can deliver superior outcomes despite being more difficult in complex motion blur circumstances.

2.1.2 Defocus blur removal

When the image of the captured license plate is not in focus, defocus blur happens. Defocus blur can be reduced by several methods:

i. Image restoration using point spread function (PSF) estimation: PSF estimation methods assess the defocus-induced blur kernel and restore the sharp image. Common techniques include focus-measure-based algorithms, depth estimation, and blind deconvolution.
ii. High-frequency emphasis: By enhancing the high-frequency elements of the image, this method makes up for the loss of sharpness brought on by defocus blur. Laplacian filtering and un-sharp masking are two common techniques for high-frequency emphasis.

2.1.3 Hybrid approaches

Several blur removal approaches combine defocus and motion blur reduction techniques to deal with challenging blurring situations. To restore licence plate photos, these hybrid systems frequently combine depth estimation, motion estimation, and deconvolution-based algorithms.

2.1.4 Deep learning-based approaches

Deep learning has recently demonstrated promise in a number of computer vision tasks, including blur reduction. To learn the correspondence between hazy and sharp photos of licence plates, convolutional neural networks (CNNs) can be trained. Even in difficult circumstances, these models are capable of properly removing fuzz from photos of licence plates.

Techniques for removing blur are essential for improving the accuracy and legibility of licence plate photographs in LPR projects. Researchers and developers continue to look for novel approaches to enhance the performance of LPR systems, ranging from conventional deconvolution-based techniques to deep learning strategies. LPR systems can increase the license plate recognition rate by using cutting-edge blur removal techniques, improving security, traffic management, and law enforcement effectiveness.

2.2 Salt and pepper noise removal

Reading license plate (LPR) systems are necessary for several purposes, including traffic monitoring, parking control, and law enforcement. However, a noise like salt and pepper noise frequently contaminates licence plate photographs taken in real-world situations. The clarity and legibility of licence plate photos are diminished by salt and pepper noise, which appears as erratically appearing white and black pixels. Effective salt and pepper noise removal techniques are used to

improve LPR system performance. An overview of popular techniques for salt and pepper noise removal in LPR applications is given on this page.

2.2.1 Median filtering

A common method for removing salt and pepper noise is median filtering. The median value of the pixels within a given neighbourhood is used in place of the pixel value. The licence plate image's edges and details are kept while the median filter successfully gets rid of isolated salt and pepper noise. The neighbourhood's (filter window's) size should be carefully selected to strike a compromise between edge preservation and noise reduction.

2.2.2 Adaptive median filtering

By adjusting the filter window size to the properties of the image, adaptive median filtering expands on the idea of median filtering. It begins with a tiny window and gradually enlarges until it contains pixels with comparable values. This adaptive method makes better noise removal possible while still protecting the licence plate image's fine details and edges.

2.2.3 Bilateral filtering

An effective non-linear edge-preserving filtering method for reducing noise in images is called bilateral filtering. When filtering, it considers both the spatial separation and intensity differential between pixels. Bilateral filtering significantly eliminates salt and pepper noise while maintaining the overall clarity of the licence plate image by preserving edges and particulars.

2.2.4 Non-local means denoising

The popular Non-local Means (NLM) denoising technique reduces noise by using redundancy in the image. The centre pixel is replaced with the weighted average of the surrounding pixels after comparing similar patches in the image. The licence plate image can be effectively cleaned up using NLM denoising while still retaining key information and structural elements.

2.2.5 Deep learning-based approaches

Convolutional neural networks (CNNs) have demonstrated considerable promise for image-denoising applications. CNN models may be taught to efficiently remove salt and pepper noise from licence plate photos by training on big datasets. These versions are appropriate for real-world LPR applications since they can adjust to various noise levels and variances.

The quality and readability of licence plate photographs in LPR projects can be severely diminished by salt and pepper noise. These noise removal techniques can help LPR systems become more accurate and reliable, which will increase the percentage of licence plate recognition. The best method to use will rely on the project's unique requirements, including computational capabilities, noise characteristics, and desired trade-offs between noise reduction and feature retention.

2.3 Hair removal

Licence Plate Recognition (LPR) devices are used in many fields, including law enforcement, parking management, and traffic surveillance. However, while taking pictures of licence plates in actual situations, undesirable objects like hair may make the plate difficult to see. Utilizing efficient hair removal procedures, LPR systems' accuracy and dependability are improved. Below are the typical hair removal techniques used in LPR projects:

2.3.1 Thresholding and morphological Operations

Thresholding and morphological operations provide the foundation of one of the most straightforward hair removal techniques. The procedure entails setting a threshold value and converting the licence plate image to binary representation. Hair sections often have a distinct colour or intensity from the characters on a licence plate, allowing for thresholding separation. By removing little hair-like structures while maintaining the primary elements of the licence plate, morphological techniques like erosion and dilation can be used to improve the hair-removal process further.

2.3.2 Image inpainting

Techniques for image inpainting can be used to fill in the areas of a licence plate image where hair is present. Inpainting methods estimate and reconstruct missing or damaged sections of an image using the surrounding data. The licence plate's visual appeal can be recovered using inpainting techniques created especially for hair removal, improving readability.

2.3.3 Edge detection and segmentation

Hair strands can frequently be recognized and segregated from a licence plate photograph thanks to their different edges. The edges of hair areas can be located using edge detection methods like the Canny edge detector or Sobel operator. The hair strands can then be separated from the background of the licence plate using segmentation techniques. As a result, the characters on the licence plate can be processed separately, increasing the accuracy of LPR systems.

2.3.4 Machine learning-based approaches

In a variety of computer vision applications, including hair removal, machine learning approaches, particularly deep learning, have displayed astounding ability. Convolutional neural networks (CNNs) can be trained on vast datasets, including images with and without hair to understand patterns and features particular to hair regions. These trained models can successfully detect and eliminate hair from photos of licence plates, improving readability and recognition accuracy all around.

2.3.5 Pre-processing with background subtraction

The background and other stationary elements of the licence plate image can be eliminated using background subtraction techniques, leaving only the moving licence plate and its text. This procedure can assist in lessening the effect that hair has on the final recognition outcome. You can use a variety of background removal algorithms, such as the Gaussian mixture model (GMM) or frame differencing, to remove the background from the licence plate image.

Hair removal is crucial for enhancing the readability of licence plate images in LPR projects, and using the right techniques that are adapted to the project's needs improves accuracy, allowing for higher recognition rates and more effective traffic management, security, and law enforcement.

3. YOLO architecture and methodologies

3.1 YOLO – v5

YOLO (You Only Look Once) is a popular object detection algorithm known for its real-time performance and high accuracy. YOLO version 5 advances previous iterations with enhanced speed and detecting skills. This article thoroughly analyzes the structure of YOLO version 5, emphasizing its essential elements and the improvements it makes to the object detection industry.

i. Backbone Network:

A convolutional neural network (CNN) serves as the structural foundation of YOLO version 5. The task of extracting features from the input image by the backbone network enables subsequent object

detection. The CSPDarknet53 design, which includes several convolutional layers, residual connections, and down sampling procedures, is updated for use in YOLO v5. The extraction of detailed hierarchical features from the input image is made easier by this design.

ii. Feature Pyramid Network (FPN):

A Feature Pyramid Network (FPN) is used in YOLO v5 to capture objects at various scales. By combining features from several network levels with various resolutions, FPN improves feature representation. This enhances detection performance by allowing the model to efficiently detect objects of various sizes.

iii. Neck:

In YOLO v5, a brand-new element known as the "neck" is added to help fill the space between the backbone and the detecting head. The neck is composed of extra convolutional layers and up sampling techniques to improve the feature representation and allow for more precise object localization.

iv. Detection Head:

The detection head is in charge of forecasting confidence scores, object classes, and bounding boxes. The detecting head in YOLO v5 is made up of several convolutional layers, followed by a number of output layers. These layers forecast bounding box coordinates, class probabilities, and confidence levels at various anchor sizes and aspect ratios.

v. Anchors:

Anchor boxes are used in YOLO v5 to make object localization easier. Anchor boxes are pre-defined boxes with various dimensions and aspect ratios that are used as markers when looking for things. The model modifies the anchor boxes for better alignment with the ground truth bounding boxes during training.

vi. Loss Function:

The loss function in YOLO v5 is a crucial component that guides the training process by measuring the errors between predicted and ground truth values. It combines three main components: localization loss, confidence loss, and class probability loss. The localization loss assesses how accurately the predicted bounding box coordinates match the ground truth coordinates. This helps the model improve its ability to localize objects in the image precisely. The confidence loss evaluates the correctness of object detection by measuring the discrepancy between predicted confidence scores and the ground truth. This encourages the model to assign higher confidence to accurately detected objects and lower confidence to false positives. Lastly, the class probability loss ensures the accurate classification of objects by calculating the difference between predicted class probabilities and the true class labels. By optimizing these components jointly, YOLO v5 is trained to detect and classify objects with improved accuracy and reliability effectively.

vii. Training and Data Augmentation:

Typically, YOLO v5 is trained using a sizable collection of tagged photos. To broaden the variety of training samples, data augmentation techniques, including random scaling, flipping, and translation, are used during training. This enhances the model's capacity to deal with object attitude, size, and appearance fluctuations.

viii. Inference and post-processing:

YOLO v5 processes an input image through the network during inference to generate predictions (Figure 2). Following that, non-maximum suppression (NMS) omits pointless bounding box detections and keeps only the most certain predictions. The residual bounding boxes, along with the class names and confidence scores they are connected with, are what remain after object detection.

To ensure precise and effective object recognition, the YOLO version 5 architecture incorporates a strong backbone network, Feature Pyramid Network (FPN), a neck component, and a detection

head. Yolo v5 offers increased performance in a variety of real-time object identification applications, from autonomous vehicles and surveillance systems to robotics and industrial automation, thanks to improvements in speed and detection skills.

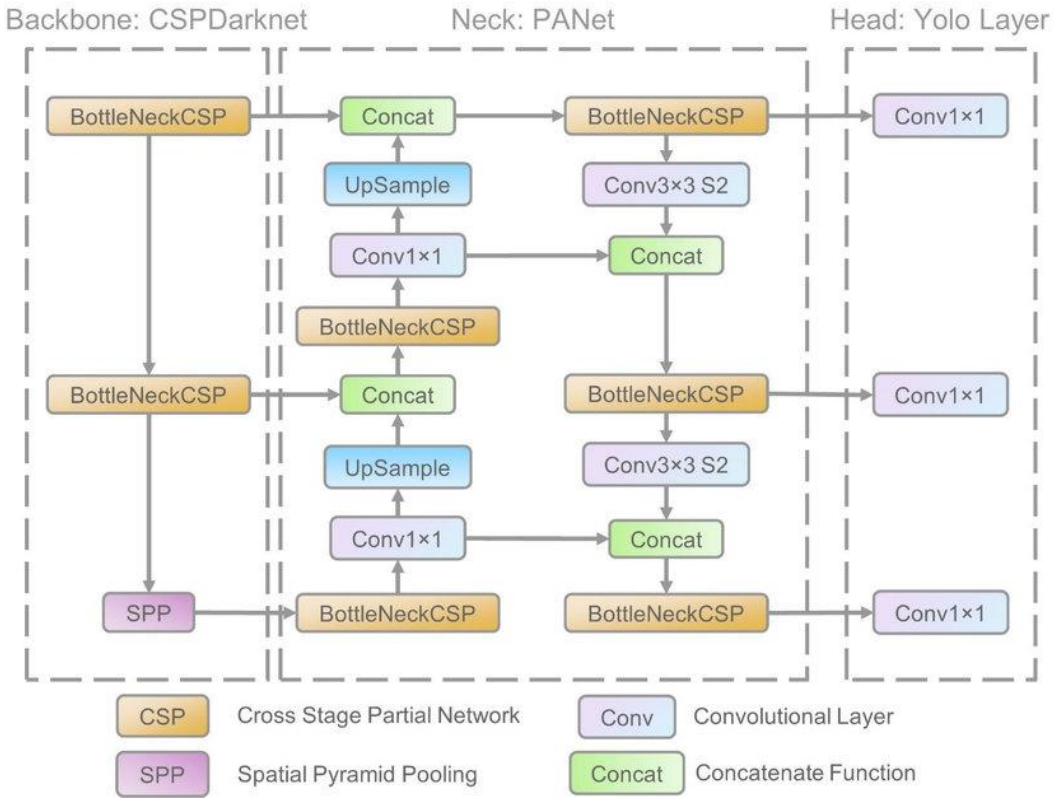


Fig. 2. Structure of YOLOv5

3.2 YOLO – v6

Yolo version 6 continues to improve on the popularity of its predecessors by adding new performance and detecting features. The enhancements that YOLO version 6 makes to the object detection field are highlighted below:

i. Backbone Network:

Similar to previous versions, YOLO v6 incorporates a powerful backbone network for feature extraction. It typically utilizes a deep convolutional neural network (CNN) architecture, such as Darknet or CSPDarknet, to capture meaningful features from input images. The backbone network plays a crucial role in extracting high-level representations that form the foundation for accurate object detection.

ii. Neck:

The enhanced neck component introduced in YOLO v6 is intended to boost feature fusion and information flow. This component focuses on combining features with various resolutions from various network levels. YOLO v6 employs cutting-edge fusion algorithms like Feature Pyramid Network (FPN) or Cross Stage Partial Network (CSPNet) to handle scale changes better and enhance object identification performance across various object sizes.

iii. Detection Head:

In YOLO v6, the detecting head is in charge of forecasting bounding box coordinates, object classes, and confidence ratings. The precision of this component has been improved, allowing for exact localization. To improve object identification, YOLO v6 includes innovative architectural options like enhanced anchor design and multi-scale prediction. The anchor design uses anchor clustering and anchor refinement algorithms to adapt to the dataset's distribution of items, improving localization capabilities.

iv. Loss Function:

YOLO v6 employs an optimized loss function combining various components to guide the training process effectively. The loss function typically includes localization loss, confidence loss, and class probability loss, similar to previous versions. However, YOLO v6 may introduce refinements or additional loss components tailored to specific objectives, such as improved localization precision or enhanced object class discrimination. By optimizing these loss components jointly, YOLO v6 trains the network to detect objects with high precision while ensuring robust classification accuracy.

v. Training Strategies and Data Augmentation:

Large datasets with annotated photos are used during YOLO v6 training to enable learning and generalization from a variety of object examples. The model is frequently optimized using sophisticated training techniques like stochastic gradient descent (SGD) with adaptive learning rate scheduling or cosine annealing. The training data is augmented using data augmentation techniques, such as random cropping, flipping, rotation, and color jittering, to increase its diversity and improve the model's capacity to handle differences in object look, posture, and scale.

vi. Inference and Post-processing:

YOLO v6 runs input photos through the trained network during inference to produce object predictions. The most certain predictions are normally kept while redundant bounding box detections are removed using non-maximum suppression (NMS). The residual bounding boxes, along with the class names and confidence scores they are connected with, are what remain after object detection.

By expanding on the advantages of its predecessors, YOLO version 6 represents a substantial improvement in the field of object detection (Figure 3). YOLO v6 seeks to improve accuracy, localization, and object categorization with its upgraded backbone network, improved neck component, improved detecting head, optimized loss function, and cutting-edge training methodologies. By utilizing these developments, YOLO v6 shows the capability to address practical issues and contribute to a variety of applications, such as robots, autonomous cars, and surveillance systems.

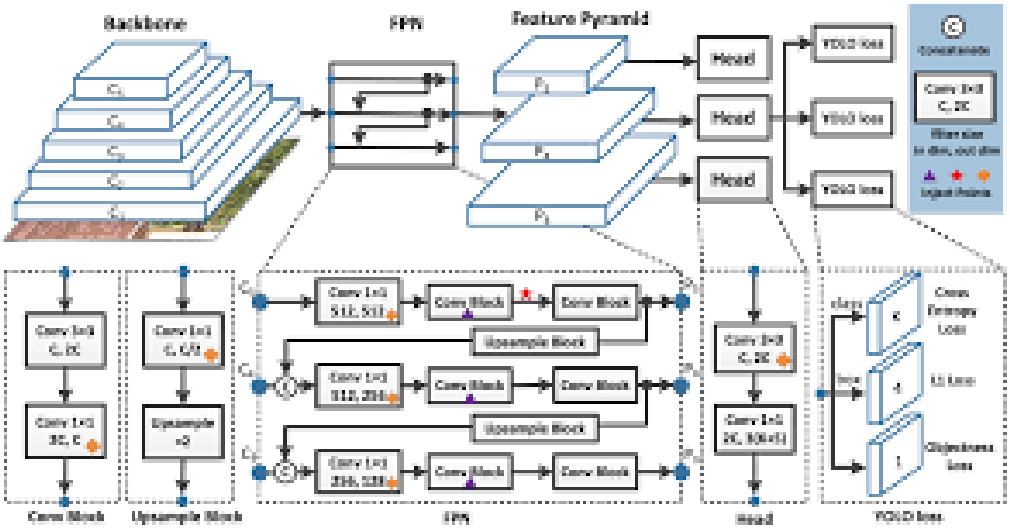


Fig. 3. Structure of YOLOv6

3.3 YOLO – v8

In the YOLO version 8 architecture, several updates and new convolutions have been introduced to enhance object detection performance, such as:

i. Backbone Changes:

The backbone of YOLOv8 underwent changes compared to its predecessor, YOLOv5. The introduction of C2f (Context Module) replaced C3 (CSPDarknet53 module). In C2f, the outputs from the Bottleneck (a combination of two 3x3 convolutions with residual connections) are combined. In contrast, in C3, only the output from the last Bottleneck was utilized. This change in the backbone architecture aims to improve feature representation and information flow within the network.

ii. Convolution Removal:

In YOLOv8, the elimination of two convolutions (#10 and #14 in the YOLOv5 configuration) indicates a refinement in the network structure to simplify computation and improve efficiency. Convolutions are mathematical operations that help the model analyse and understand different features in an image. By removing these specific convolutions, the YOLOv8 architecture streamlines the computation process, making it more efficient and faster. Simplifying the network structure in this way can lead to better performance in terms of speed and resource utilization, allowing the model to process images more quickly and effectively.

iii. Bottleneck Updates:

The only difference between the Bottleneck module in YOLOv8 and YOLOv5 is that the first convolution's kernel size was changed from 1x1 to 3x3. With this adjustment, YOLOv8 is brought into line with the ResNet block design that was put forth in 2015, potentially allowing for the use of the ResNet structure's feature extraction and representation advantages [14].

iv. Anchor-Free Detection:

Anchor-free detection is a significant change in YOLOv8 compared to earlier versions, which relied on the use of anchor boxes. In anchor-free detection, the model directly predicts the centre of an object rather than estimating the offset from a fixed anchor box. Anchor boxes are pre-defined boxes with specific heights and widths that are used to identify object classes with desired sizes and aspect ratios. These anchor boxes are chosen based on the objects' sizes in the training dataset and are placed across the image during detection. However, anchor-free detection in YOLOv8 eliminates the need for predetermined anchor boxes, making it more adaptable and effective. Instead of relying on fixed anchor box configurations, the model outputs probabilities and attributes for each potential object centre, enabling it to adjust and adapt to different object sizes and orientations dynamically. This flexibility allows for better detection accuracy and a more efficient object detection process.

v. Working with Anchor Boxes:

In previous versions of YOLO, such as YOLOv1 and YOLOv2, boundary box predictions relied on anchor boxes as fixed starting positions. These anchor boxes, with specific sizes and aspect ratios, were replicated across the image during detection based on the object sizes in the training dataset. However, in YOLOv8, the approach is more flexible and adaptive. Instead of using fixed anchor boxes, the network generates probabilities and properties for each tiled box. These properties include information about whether the box contains an object (background), the overlap between the predicted box and the ground truth (IoU), and the offsets needed to refine the box prediction. Based on these variables, the anchor boxes are adjusted accordingly. By dynamically adjusting the anchor boxes using these generated properties, YOLOv8 can better capture objects of various scales and aspect ratios, improving the accuracy and versatility of the object detection process.

A more advanced object detection model is YOLOv8. The model is now more intelligent due to modifications it has made to how it perceives and comprehends the world. It is more adaptable and effective now that it anticipates the centre of objects without the use of specific boxes. These improvements make the model more adept at identifying and comprehending things in images (Figure 4).

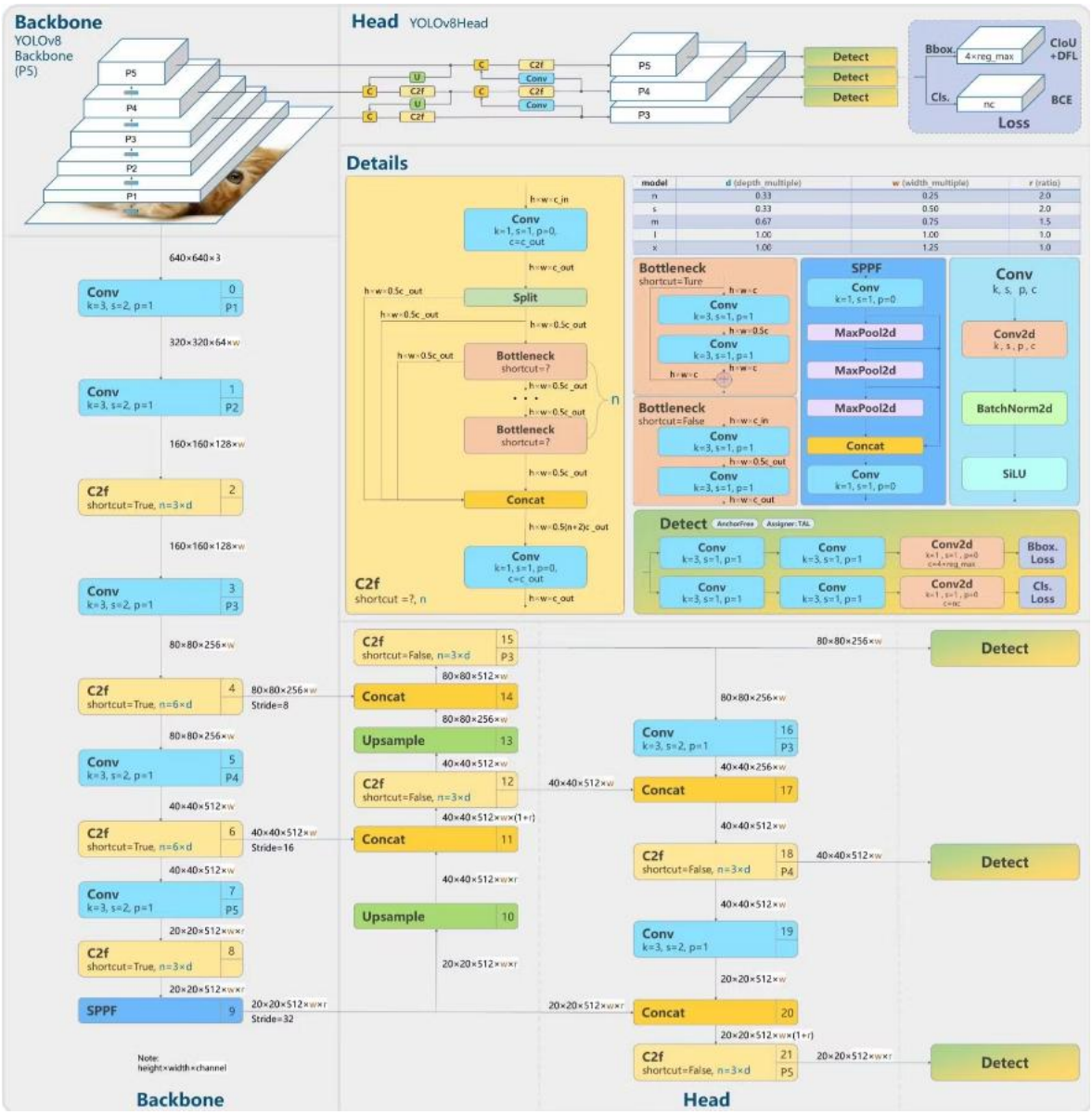


Fig. 4. Structure of YOLOv8 model

4. Results and discussions

The dataset for training and testing will be available in the Google-Open-Images datasets of vehicles under the folder /darknet/data. The dataset contains 1800 images in YOLO format, and it is used for our project. Apart from other machine learning algorithms, we have also implemented YOLO v5, v6, and v8 in this project. YOLO outperforms other machine learning algorithms [15-17], and the detailed discussions are given below: Figures 5 to 10 discuss the calculation of various metrics like accuracy and Loss curves, sample validation images, F1 confidential curve, Precision confidential curve, Recall confidential curve and precision-recall curve for YOLO v5. Similarly, various metrics for YOLO v6 are given from Figures 11 to 16, and the calculation of various metrics for YOLO v8 is given in Figures 17 to 21.

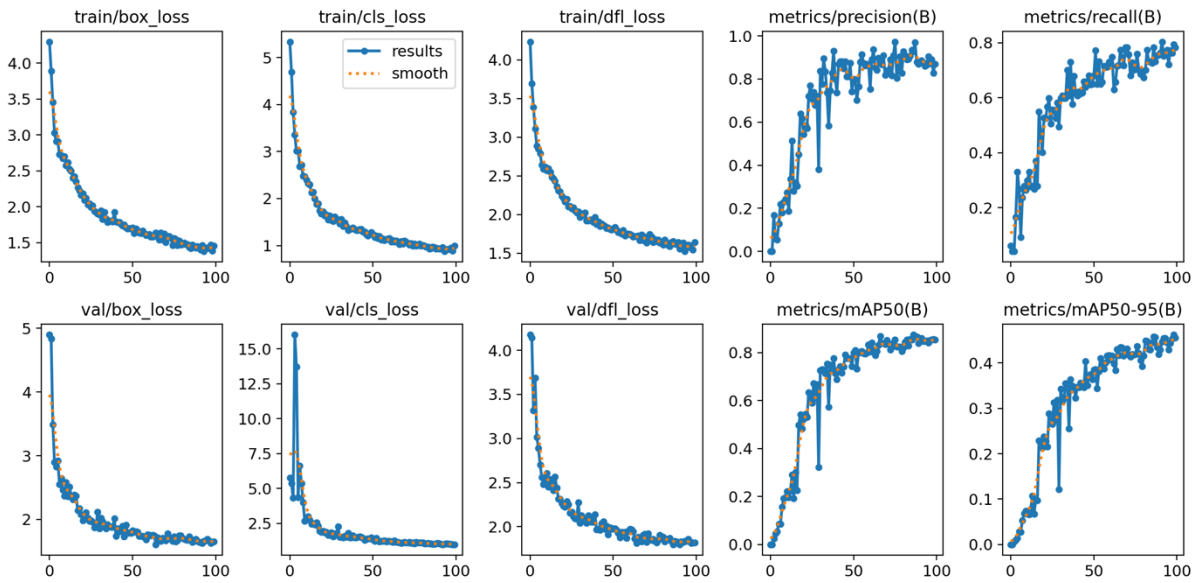


Fig. 5. YOLOv5 Accuracy & Loss

Fig. 6. Sample Validation predictions

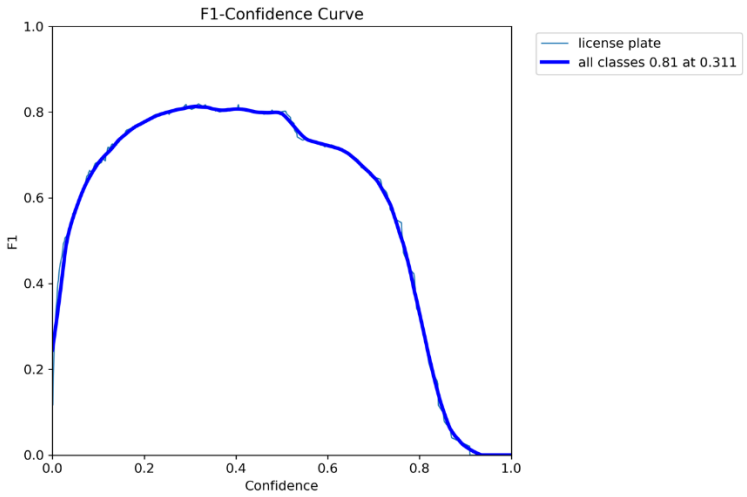


Fig. 7. YOLOv5 F1 Confidence curve

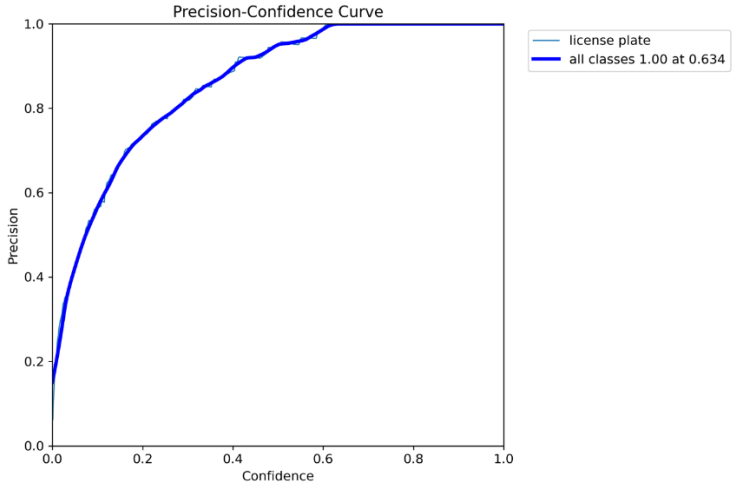


Fig. 8. YOLOv5 Precision Confidence curve

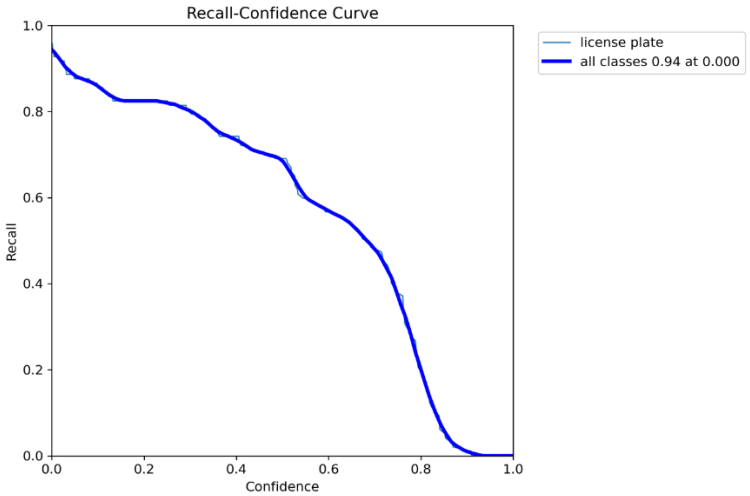


Fig. 9. YOLOv5 Recall Confidence curve

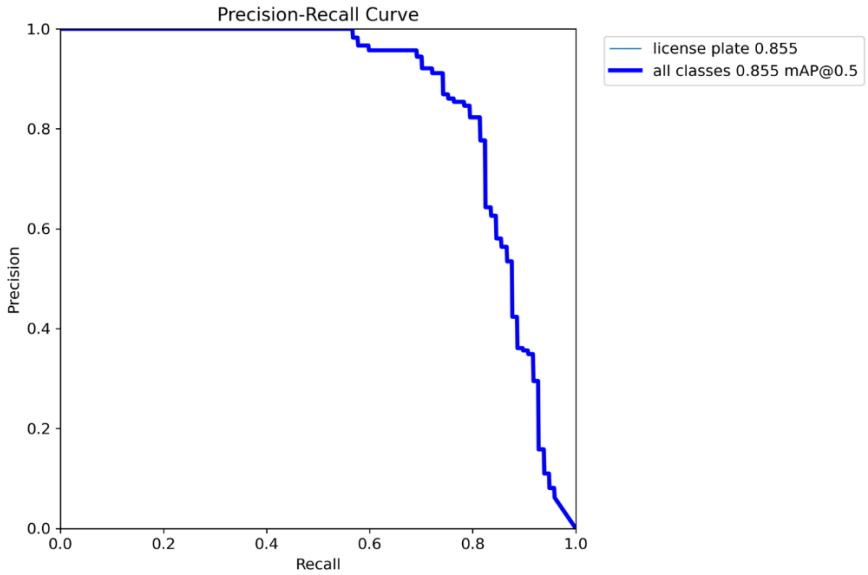


Fig. 10. YOLOv5 Precision-Recall curve

4.1. YOLOv6

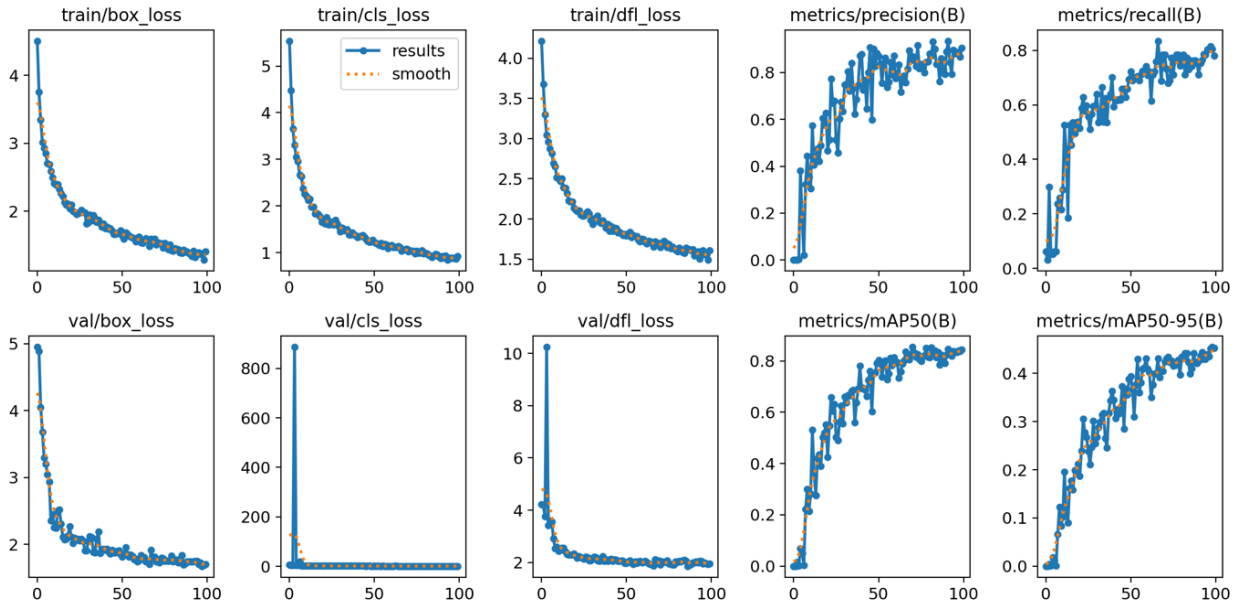


Fig. 11. YOLOv6 Accuracy & Loss

Fig. 12. Sample Validation predictions

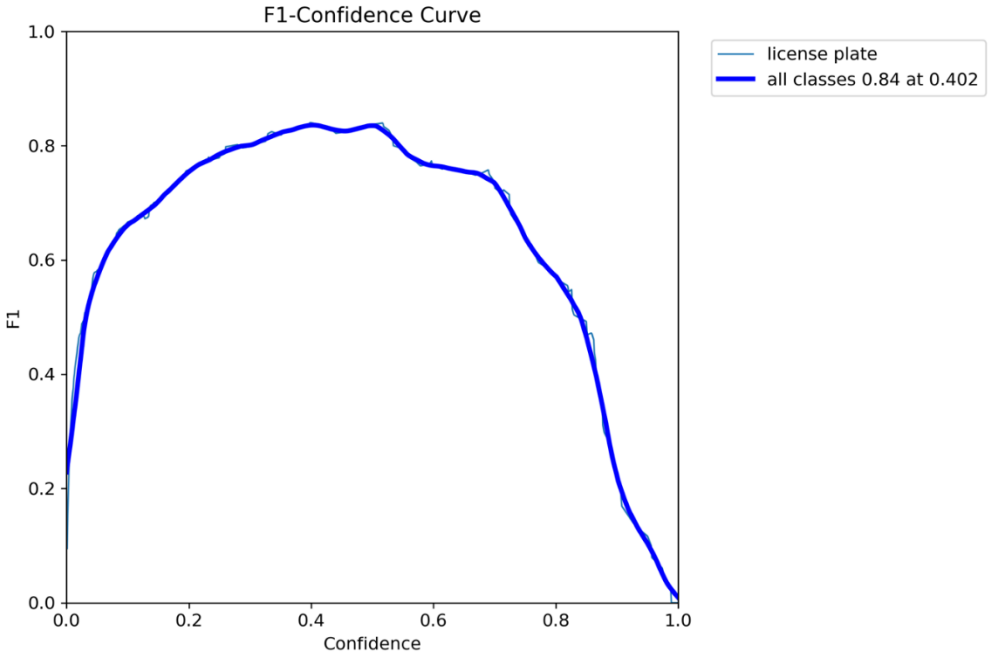


Fig. 13. YOLOv6 F1 Confidence curve

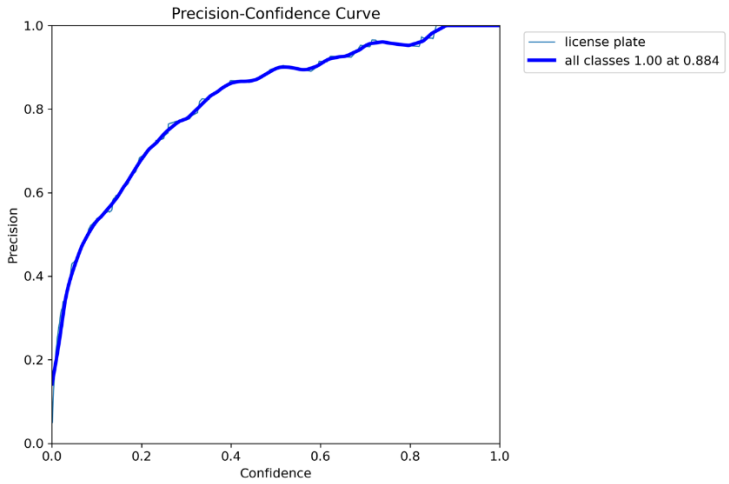


Fig. 14. YOLOv6 Precision Confidence curve

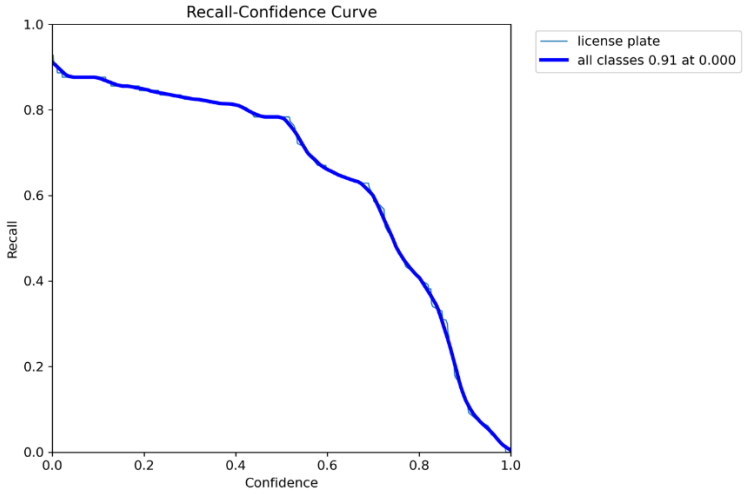


Fig. 15. YOLOv6 Recall Confidence curve

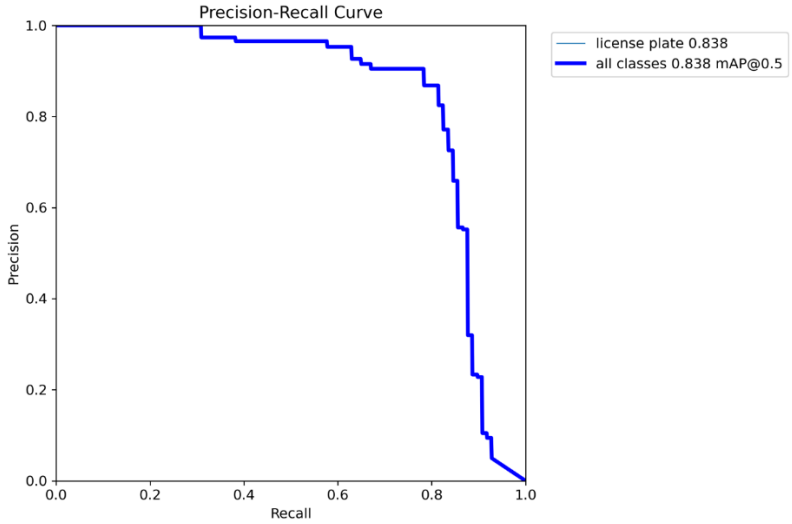


Fig. 16. YOLOv6 Precision-Recall curve

4.2 YOLOv8

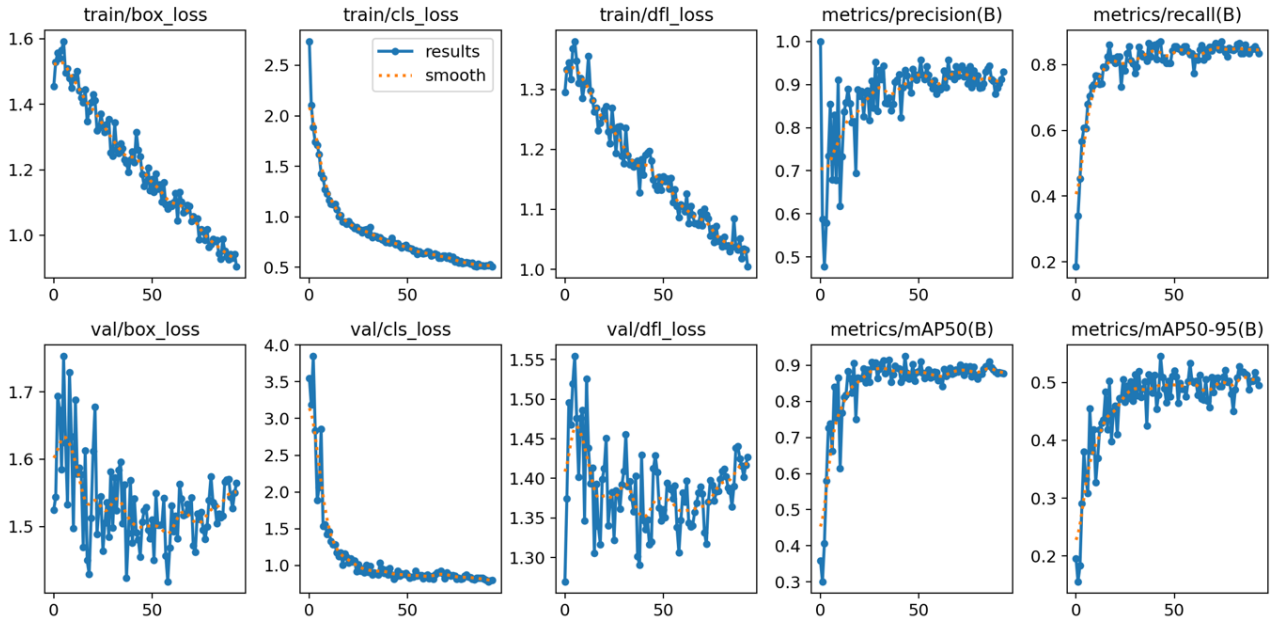


Fig. 17. YOLOv8 Accuracy & Loss

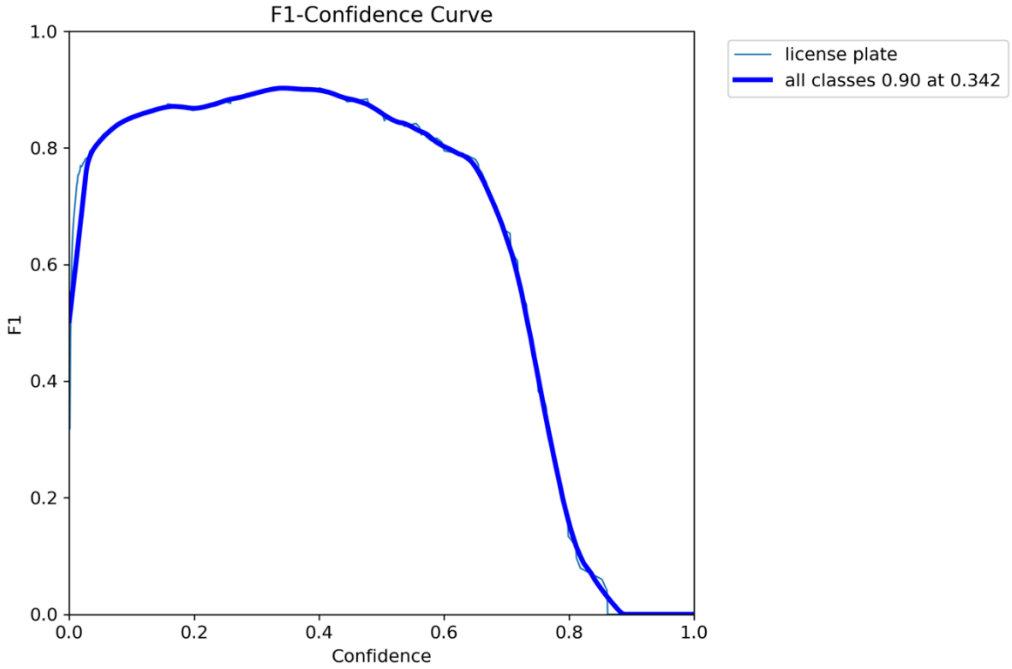


Fig. 18. YOLOv8 F1 Confidence curve

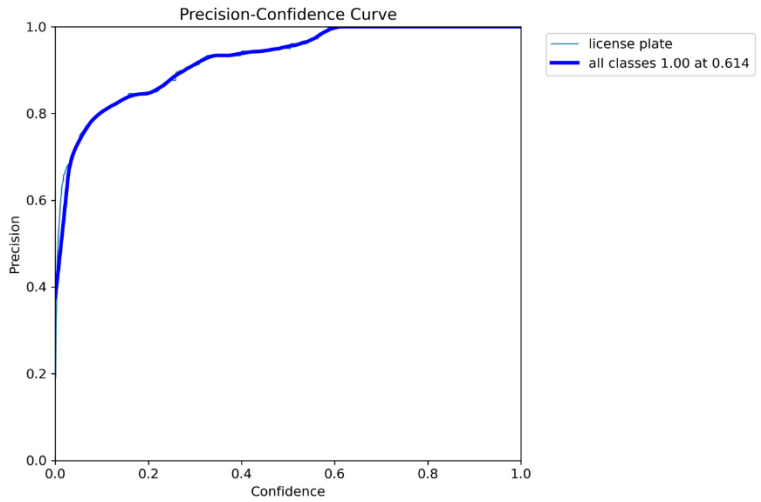


Fig. 19. YOLOv8 Precision Confidence curve

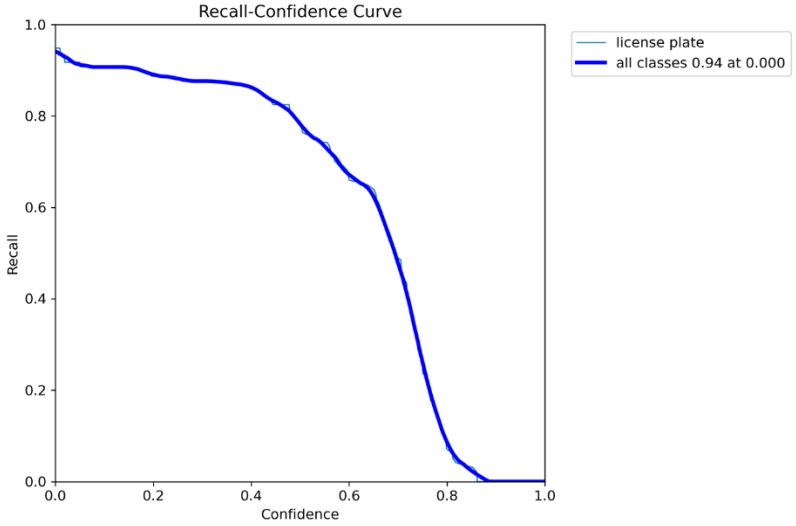


Fig. 20. YOLOv8 Recall Confidence curve

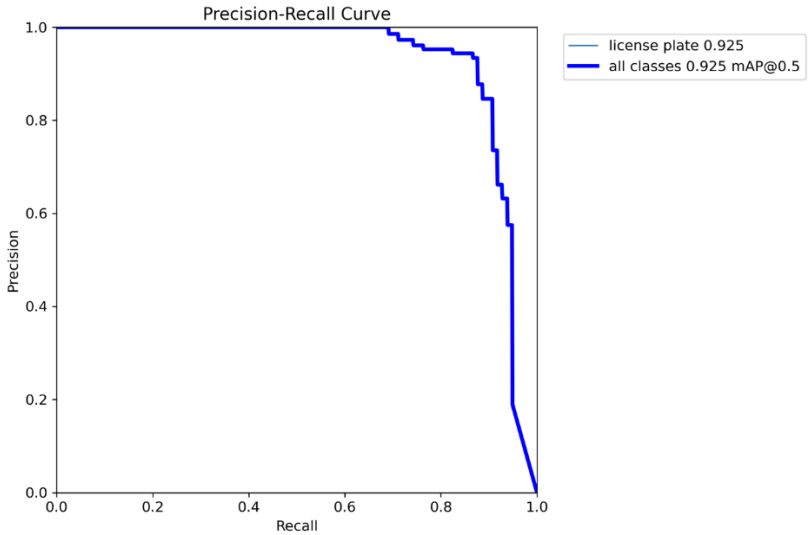


Fig. 21. YOLOv8 Precision-Recall curve

The performance analysis of the proposed machine learning algorithms, including YOLO v5, v6, and v8, is given in Table 2.

Table 2
Performance Analysis of Various Machine Learning Algorithms of number plate identification

S.No.	Machine Learning Algorithms	Accuracy
1	SVM	92.4
2	Convolutional Neural Network	95.8
3	YOLO v5	97.9
4	YOLO v6	93.5
5	YOLO v8	95.6

From the above table, YOLO v5 outperforms other machine-learning models, and it showed an accuracy of 97.8, which is better than the other state-of-methods proposed recently.

5. Conclusion

In this project, we have applied the pipeline of operations given in Figure 1 to determine the exact prediction of number plate using machine learning and deep learning algorithms before applying various pre-processing methods like noise removal algorithm, extended hair removal algorithm, etc. [18,19]. During the execution of the project, we found that the YOLO v5 outperforms the other deep learning and machine learning models. Finally, it is concluded that the YOLO v5 showed better performance than the other state-of-art techniques.

Acknowledgement

This research has not been funded by any grant.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Shashidhar, R., Manjunath, A. S., Kumar, R. S., Roopa, M., & Puneeth, S. B. (2021). Vehicle Number Plate Detection and Recognition using YOLO-V3 and OCR Method. In 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC) (pp. 1-5). IEEE. https://doi.org/10.1109/ICMNWC52512.2021.9688407

[2] Coetzee, C., Botha, C., & Weber, D. (1998). PC based number plate recognition system. In IEEE International Symposium on Industrial Electronics. Proceedings. ISIE'98 (Cat. No. 98TH8357) (Vol. 2, pp. 605-610). IEEE. https://doi.org/10.1109/ISIE.1998.711680

[3] Wang, H., Hou, J., & Chen, N. (2019). A survey of vehicle re-identification based on deep learning. IEEE Access, 7, 172443-172469. https://doi.org/10.1109/ACCESS.2019.2956172

[4] Roy, A., & Ghoshal, D. P. (2011). Number Plate Recognition for use in different countries using an improved segmentation. In 2011 2nd National Conference on Emerging Trends and Applications in Computer Science (pp. 1-5). IEEE. https://doi.org/10.1109/NCETACS.2011.5751407

[5] Fahmy, M. M. (1994). Automatic number-plate recognition: neural network approach. In Proceedings of VNIS'94-1994 Vehicle Navigation and Information Systems Conference (pp. 99-101). IEEE. https://doi.org/10.1109/VNIS.1994.396858

[6] Selmi, Z., Halima, M. B., & Alimi, A. M. (2017). Deep learning system for automatic license plate detection and recognition. In 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 1132-1138). IEEE. https://doi.org/10.1109/ICDAR.2017.187

[7] Zidouri, A., & Deriche, M. (2008). Recognition of Arabic license plates using NN. In 2008 First Workshops on Image Processing Theory, Tools and Applications (pp. 1-4). IEEE. https://doi.org/10.1109/IPTA.2008.4743757

[8] Setiyono, B., Amini, D. A., & Sulistyaningrum, D. R. (2021). Number plate recognition on vehicle using YOLO-Darknet. In Journal of Physics: Conference Series (Vol. 1821, No. 1, p. 012049). IOP Publishing. https://doi.org/10.1088/1742-6596/1821/1/0120 4

[9] Shashirangana, J., Padmasiri, H., Meedeniya, D., & Perera, C. (2020). Automated license plate recognition: a survey on methods and techniques. IEEE Access, 9, 11203-11225. https://doi.org/10.1109/ACCESS.2020.3047929

[10] Baviskar, D., Choudhary, S., Danve, R., Kinkar, K., & Patil, S. (2022). Auto Number Plate Recognition. In 2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST) (pp. 1-6). IEEE. https://doi.org/10.1109/AIST55798.2022.10064725

[11] Poorani, G., Krishnna, B. K., Raahul, T., & Kumar, P. P. (2022). Number Plate Detection Using YOLOV4 and Tesseract OCR. Journal of Pharmaceutical Negative Results, 130-136. https://doi.org/10.47750/pnr.2022.13.S03.021

[12] Sarfraz, M., Ahmed, M. J., & Ghazi, S. A. (2003). Saudi Arabian license plate recognition system. In 2003 International conference on geometric modeling and graphics, 2003. Proceedings (pp. 36-41). IEEE. https://doi.org/10.1109/GMAG.2003.1219663

[13] Ahmed, M. J., Sarfraz, M., Zidouri, A., & Al-Khatib, W. G. (2003). License plate recognition system. In 10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 (Vol. 2, pp. 898-901). IEEE. http://dx.doi.org/10.1109/ICECS.2003.1301932

[14] Phong, B. H. (2024). Classification of plant leaf diseases using deep neural networks in color and grayscale images. Journal of Decision Analytics and Intelligent Computing, 4(1), 99-110. https://doi.org/10.31181/10002052024p

[15] Chatterjee, V., Maitra, A., Ghosh, S., Banerjee, H., Puitandi, S., & Mukherjee, A. (2024). An efficient approach for breast cancer classification using machine learning. Journal of Decision Analytics and Intelligent Computing, 4(1), 32-46. https://doi.org/10.31181/jdaic10028012024c

[16] Nandan, M., & Ghosh, D. (2023). Pre-owned car price prediction by employing machine learning techniques. Journal of Decision Analytics and Intelligent Computing, 3(1), 167-184. https://doi.org/10.31181/jdaic10008102023n.

[17] Yazdi, A. K., & Komasi, H. (2024). Best Practice Performance of COVID-19 in America continent with Artificial Intelligence. Spectrum of Operational Research, 1(1), 1-12. https://doi.org/10.31181/sor1120241

[18] Sahoo, S. K., Choudhury, B. B., & Dhal, P. R. (2024). Exploring the Role of Robotics in Maritime Technology: Innovations, Challenges, and Future Prospects. Spectrum of Mechanical Engineering and Operational Research, 1(1), 159-176. https://orcid.org/0000-0002-8551-7353

[19] Younas, R., Haq, H. B. U., & Baig, M. D. (2024). A framework for extensive content-based image retrieval system incorporating relevance feedback and query suggestion. Spectrum of Operational Research, 1(1), 13-32. https://doi.org/10.31181/sor1120242