J-COSINE (Journal of Computer Science and Informatics Engineering)
Vol. 10, No. 1, June 2026 E-ISSN:2541-0806
Accredited Sinta-4 by RISTEKDIKTI Decree No. 79/E/KPT/2023 P-ISSN:2540-8895

Vehicle License Plate Number Detection with YOLO11

Lalu Mischa Khalqin Adhiguna*, Ario Yudo Husodo, Fitri Bimantoro
Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: f1d021101@student.unram.ac.id, [ario, bimo]@unram.ac.id

Corresponding Author

Abstract **Automatic License Plate Recognition (ALPR) systems are crucial for automating vehicle monitoring and access control, especially in parking management. However, many existing ALPR methods rely on earlier detection models and perform poorly under low-light or night-time conditions. This study addresses that issue by developing a license plate detection and recognition system using the You Only Look Once (YOLO11) algorithm and PaddleOCR (Optical Character Recognition) for character recognition. The objective is to evaluate the performance of this approach on Indonesian two-wheeled vehicles under varied lighting conditions. The dataset includes daylight and low-light images, divided into training, validation, and test sets. YOLO11 was trained with several epoch configurations, and the best model achieved 99.8% precision, 100% recall, 99.5% mAP@50, and 80.7% mAP@50–95. The model was then integrated with PaddleOCR to extract license plate text. Results show this combination is effective for real-time vehicle identification and can improve parking queue management.**

***Key words*: ALPR, YOLO, Detection, OCR, Plate.**

I. Introduction

A. Background

Vehicles are a secondary human need because they have a supporting role and support all daily human activities and productivity. Population growth is usually followed by an increase in the number of vehicles, as many people prefer to use private vehicles. The growth rate of the number of motorized vehicles in Indonesia has increased significantly from year to year. For example, the number of sedan cars increases by about 15% annually, while the growth in the number of motorcycles is even higher, exceeding 30% per year [1]. This increase in the use of private vehicles is not only due to the convenience factor, but also due to the relatively low cost of using them. In addition, along with the increasing density of vehicles, every vehicle owner is required to register their vehicle with the authorities so that the vehicle has an identification mark or a form of identification in the form of a license plate [2].

Because there are many types of vehicles on the market that are similar in terms of brand and type, it is necessary to have a distinguishing element in each vehicle. One part that functions as a differentiator is the vehicle license plate number, which is a unique identity for each vehicle [2]. A vehicle license plate is a form of official identification of a vehicle issued by an authorized agency as proof that the vehicle has obtained permission to be used on public roads [3]. Each license plate has a different serial number and is classified based on the province where the vehicle is registered in Indonesia. The letter code located before the number on the license plate indicates the province of vehicle registration, while the letter code located after the number indicates a more specific sub-region or region within the province [4].

License plates also have other very important roles in various real-life applications, such as automated parking systems, traffic management, and many others. In parking systems, license plates are used as proof of vehicle identity. Every vehicle that enters and exits the parking lot needs to be registered to prevent crime. In general, officers must enter vehicle information manually by recording license plates in a recording book. This process becomes less efficient, especially with the increasing number of vehicles and the limited number of human resources available to handle the registration [5].

Along with current technological advances, automatic license plate recognition systems can be developed through computer software. One of the rapidly growing technologies in this field is LPR or License Plate Recognition. This identification system is designed to reduce the use of time and energy in the process of recording vehicle identity data, which was previously still done manually [4]. In general, Automatic License Plate Recognition (ALPR) is a technology that utilizes a combination of specialized hardware and software to automatically detect, recognize, and record vehicle license plates through image processing. ALPR systems are able to identify vehicle license plates in real-time and present important information such as vehicle data, owner identity, and legal status associated with the vehicle [6].

The process to detect this utilizes object detection technology, one of the popular algorithms used is You Only Look Once (YOLO). This algorithm has an advantage in terms of detection speed because it is able to analyze the entire image in one process. YOLO is a deep learning technique designed to detect objects in an image efficiently. This approach enables real-time identification of multiple objects as it is able to draw a bounding box around each detected object. YOLO works by predicting the presence of objects using a convolutional neural network (CNN), which is able to calculate the class probability of each bounding box formed [7]. Convolutional Neural Network is a type of deep neural network and is often used in image data [8]. Not only that, there are also other technologies that can be used with the algorithm, namely optical character recognition (OCR) technology which is a method of extracting characters from a digital image that can be processed and further processed by digital computing [9].

In some cases in Indonesia, one of the studies conducted in Palembang said that the system generally still relies on officers who record vehicle data manually and provide tickets as proof of parking[10]. However, in practice, automated parking systems still have some disadvantages, one of the studies conducted in Yogyakarta also said that one of which is the time required to record vehicle numbers and open and close parking portals tends to be quite long. This condition can cause queues and congestion, especially during high vehicle volumes when vehicles will enter or exit through the parking portal [11]. In addition, research conducted in Medan also said that some areas already use a security system with barrier gates, but this system often causes long queues due to the time-consuming validation process, so motorists have to wait a long time on their vehicles. If the solution implemented is to increase the number of exits, then it automatically requires additional vehicle data validation terminals and additional labor to operate them. The problem will be even more complex if the system is implemented in a campus environment, which generally has limited access areas for vehicle entry and exit, as well as a limited number of parking attendants, for example only two people in charge of organizing and arranging untidy vehicles. Under these conditions, the use of an automated system that can reduce dependence on parking attendants becomes very important so that parking services are more efficient and do not cause jamming [12].

With increasing urbanization and the rapid growth of vehicle numbers in Indonesia, the need for efficient, real-time vehicle identification systems has become critical. Manual entry systems used in parking or traffic law enforcement are time-consuming, prone to human error, and impractical in high-volume environments. As the demand grows for intelligent transportation systems that operate around the clock, it is important to develop detection systems that function accurately in both daylight and low-light scenarios. A model that performs greatly across lighting conditions can significantly improve the automation of vehicle monitoring, parking systems, and existing laws.

B. Research Gap

Prior studies on Indonesian vehicle license plate detection have commonly focused on earlier YOLO architectures, such as YOLOv5 or YOLOv8, for example. These studies have mostly focused on ideal conditions, such as well lighting and high image quality. In real-world applications, however, such as in parking lots, traffic monitoring, and urban surveillance systems, lighting conditions can vary greatly including dim or night-time. Additionally, few studies have systematically evaluated the performance of the newest YOLO architecture, the YOLO11, in the context of license plate detection, particularly when paired with advanced OCR methods such as PaddleOCR. This leaves a gap in understanding how the most recent object detection models perform in challenging, non-ideal conditions for Indonesian license plates. Through this comparative evaluation, the study seeks to provide insights into the advantages offered by YOLO11 in terms of detection precision and environmental adaptability. Furthermore, by combining it with PaddleOCR, the research aims to improve the integration of the license plate recognition system.

This study has the potential to improve the accuracy of Indonesian vehicle license plate detection by selecting a new effective model architecture, as in this topic is YOLO11. Higher detection accuracy will support more reliable license plate recognition. By choosing a more efficient and accurate architecture, the detection process can be executed faster and with reduced computational resources, making it suitable for real-time applications in both bright and low-light conditions. Furthermore, the integration with PaddleOCR allows for precise character recognition, forming a comprehensive and optimized system for license plate analysis.

C. Novelty

This study contributes to the field by implementing and evaluating YOLO11, a newer and more efficient object detection algorithm, for detecting Indonesian vehicle license plates under varied lighting conditions. In contrast to prior studies that rely solely on YOLOv5 or YOLOv8, this research tests YOLO11's adaptability and precision on a mixed dataset of bright and low-light images. Furthermore, PaddleOCR is integrated into the system to extract plate text accurately. The novelty lies in the application of YOLO11 for the first time in this specific context, the use of lighting-varied data, and the practical integration aimed at improving parking management systems. This approach not only enhances detection accuracy but also explores how the model can be effectively deployed in embedded environments such as surveillance cameras or gate automation systems.

II. State of the Art and Related Research

A. State of The Art

The table that depicts the research collection is presented in TABLE I.

TABLE I. State of The Art

No.	Article	Model	Claimed Accuracy
1	[2]	YOLOv4	86.82%
2	[4]	YOLOv3	92.32%
3	[5]	YOLOv3	90% (F1 Score)
4	[7]	YOLOv5	84%
5	[9]	YOLOv5	92.38%
6	[13]	YOLOv5	70%
7	[14]	YOLOv8	87.1%
8	[15]	YOLOv8	99%
9	[16]	YOLOv8	98.64%
10	[17]	YOLOv5	98%
11	[18]	YOLOv4	96%
12	[19]	YOLOv4	90%
13	[20]	YOLOv4	92.4%
14	[21]	YOLOv7	84%
15	[22]	YOLOv8	98.7%
16	[23]	YOLOv8	90%
17	[24]	YOLOv8	98%
18	[25]	YOLOv8	95.3%
19	[26]	YOLOv5	83.08%
20	[27]	YOLOv8	99%

Table I shows various results from previous studies, the use of the YOLO algorithm for vehicle license plate detection has shown highly variable performance depending on the model version used and the test environment conditions. YOLOv3 and YOLOv4 show fairly high detection accuracy with values ranging from 86% to 96%, but there are performance discrepancies when tested under low-light conditions, such as in the YOLOv4 study which only achieved 60% accuracy under low-light conditions. YOLOv5, which is one of the most popular architectures, recorded accuracy results ranging from 70% to 98%, indicating that the performance of this model is highly influenced by data quality and the training process. Meanwhile, the YOLOv8 model showed consistently high performance with accuracy values ranging from 87.1% to 99%, and even reaching 98.7% in some studies. YOLOv7, although newer than YOLOv5, showed lower results in one study at 84%, indicating that a newer version of the model does not necessarily guarantee better performance without optimal settings and data.

In general, the trends from this data show that while many versions of YOLO have been widely used, their performance is still heavily influenced by factors such as lighting, vehicle data type, annotation quality, and the combination of post-detection methods such as OCR. In addition, there have not been many studies comparing the performance of recent YOLO versions such as YOLO11, especially in the context of varying lighting conditions in real environments. Therefore, this study seeks to fill the gap by evaluating the performance of YOLO11 in detecting Indonesian two-wheeler license plates under light and low-light conditions, and integrating it with PaddleOCR to improve the accuracy of character recognition.

B. Related Research

Based on explanation above, several related studies have used methods with the YOLO algorithm such as using YOLOv5 and YOLOv8 to improve the accuracy of vehicle license plate detection. The first study used YOLOv5 to detect Indonesian vehicle license plates and used Tesseract OCR to read vehicle license plate numbers. This study concludes that this method produces an average object detection accuracy of 92.38% in optimal lighting conditions (daytime) and the results of character extraction in license plates get an accuracy of 95.45% for detected text and 97.2% accuracy in the success rate of the program to determine the category of the license plate [9].

The second study used YOLOv5 to detect Indonesian vehicle license plates and used TesseractOCR for character extraction on vehicle license plate numbers. The study concluded that the method produced an average accuracy value of 85% depending on image conditions such as image quality, light, and angle. TesseractOCR's ability to read the characters on the license plate achieved an accuracy of 70% [13].

The third study used YOLOv8 to detect Indonesian vehicle license plates and used PaddleOCR for character extraction on vehicle license plate numbers. This study concluded that this method produced a precision score of 87.1% and a recall score of 85%. This study also concluded that the character recognition method on license plates also gives good results in recognizing characters on police license plates, including characters that are distorted or have low image quality [14].

The fourth study used YOLOv8 to detect Indonesian vehicle license plates and used EasyOCR to extract text from vehicle plates. This study concluded that the research conducted resulted in 99% for the accuracy of the detection model and 81% for recognition [15].

Overall, the studies mentioned above shows that each of the YOLO and OCR methods had different results. The comparison of the studies is expected to identify things that need to be improved for a more accurate analysis in detecting Indonesian vehicle license plates. However, there are still some research gaps that need to be addressed, one of which is the challenge posed by variations in lighting conditions, such as images taken during the day with bright lighting or at night with low lighting. In addition, many developed algorithms are still limited to ideal conditions and standardized license plate formats in certain regions, so their performance may degrade when applied to more complex real-world environments.

In this context, the use of newer algorithms such as YOLO11 becomes relevant to explore, given its ability to detect objects in real-time with high efficiency. This research aims to address the gap by testing the performance of the detection model on a dataset that includes diverse lighting conditions. By combining light and low-light data in the training and evaluation process, it is expected that the resulting model will be able to perform reliably in a variety of real-world situations, including on embedded system devices. This will open up wider application opportunities, ranging from traffic surveillance systems to automated law enforcement. In this research, the focus of application is on the parking management.

III. Research Methodology

A. Research Flow

The flowchart that explains the flow of the proposed research is presented in Fig. 1.

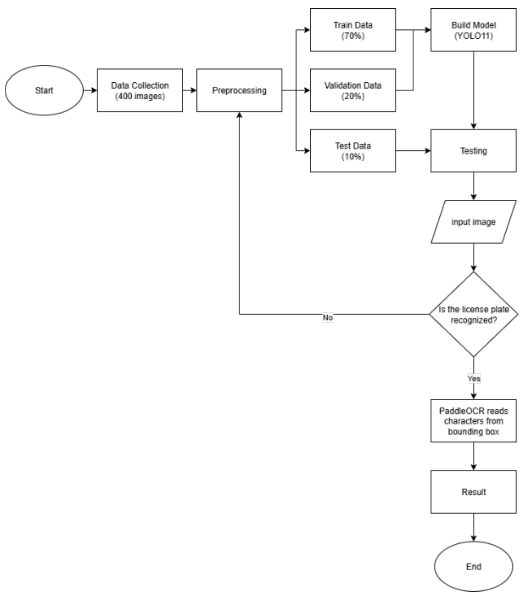


Fig. 1. Research Flow

The image above is the flowchart of overall process of a license plate detection and recognition system using YOLOv11 for object detection and PaddleOCR for character recognition. This research starts by collecting images of vehicles with visible license plates. Before using the images for training, preprocessing is applied. Then, the dataset is divided into three parts. The next step is to build and train the YOLO11 object detection model using the training and validation data. The model learns to detect license plates within vehicle images. After training, evaluate the model on the test data to check its performance (e.g., precision, recall, mAP50, and mAP50-95 scores). Once the model is trained, you input a new image into the system for detection and recognition. The YOLOv11 model processes the input image: If no, fallback to preprocessing step and repeat the model training. If yes, the detected license plate is passed on for text recognition. Then, PaddleOCR extracts and recognizes the characters from the detected license plate (bounding box provided by YOLO11). The system outputs the recognized license plate number as the final result. The process ends after recognition is completed.

B. Dataset

This study uses images totaling 400 images of two-wheeled motor vehicle license plates, 250 images of two-wheeled motor vehicle license plates in bright light taken from Kaggle and 150 images of two-wheeled motor vehicle license plates in low light condition at 20.00 to 22.00 at night, these images was taken personally using iPhone XR camera. All are processed with the help of Roboflow. On Fig. 2, the data annotation is done manually using one label named plate. Images that are used in this research consist of two-colored vehicle plates, white and black. One of the notations that portray the proposed dataset annotation is presented in Figure 2.

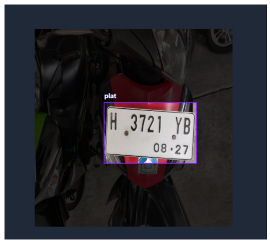


Fig. 2. Dataset Annotation

C. Data Split

The data splitting process is performed randomly to minimize bias during model training and to ensure a representative distribution across all subsets. This approach guarantees that each subset reflects the overall variation and characteristics present in the complete dataset. The table that represents how the dataset is split is presented in TABLE II.

TABLE II. Data Splitting

Subset	Value
Train	280 images
Validation	80 images
Test	40 images

In the data splitting stage, the dataset is divided into three different subsets, namely the training set, validation set, and testing set. The training set takes up the largest proportion, 70% of the overall data, consisting of 280 images. This subset is used as training data to teach the model to recognize patterns and features related to vehicle license plates. During the training process, the model learns the visual characteristics of the data in the training set to form an initial understanding.

The validation set includes about 20% of the total data, consisting of 80 images. This subset serves to evaluate the performance of the model during the training process. Through the validation set, it can be seen whether the model is overfitting (too well matched to the training data) or underfitting (less able to capture patterns). Therefore, this subset is very important in the model refinement process.

Meanwhile, the testing set is the smallest subset with a proportion of about 10% of the overall data, which is 40 images. This subset is used to test the final performance of the model independently after the training is completed. Evaluation using the Testing Set gives an idea of the extent to which the model is able to recognize new plate numbers that have never been seen before, thus reflecting the generalization ability of the model to real data.

D. Preprocessing

Preprocessing is an important step in preparing the dataset before the training process of the vehicle license plate recognition model is carried out. In this research, the preprocessing stage is carried out by applying two main techniques, which are auto-orient and resize. The image that portrays the preprocessing of the dataset is presented in Fig. 3.

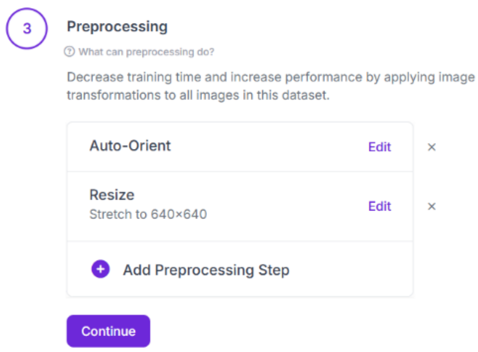


Fig. 3. Data Preprocessing

The auto-orient is used to align the orientation of vehicle license plate images so that all images have a uniform position. This adjustment is important because orientation differences in images can make it difficult for the model learning process to recognize patterns consistently. After that, a resize process is carried out which aims to change the size of the entire image to 640x640 pixels. This uniform size is adjusted to the standard in the YOLO documentation to ensure that the input data provided to the model is consistent, making it easier for the model to learn the visual features of the vehicle license plate efficiently [14].

E. Model Building

After the dataset has been processed, the next step is to train the model by initializing YOLO11 using pretrained weights as the starting point. This training process uses two subsets of data, the training set to train the model and the validation set to evaluate the performance of the model during the training process.

In this process, the "AdamW" optimizer is used, which was chosen for its ability to handle weight updates efficiently, especially in deep learning models. The learning rate is determined to set the speed at which the model's weights are adjusted during training. In addition, parameters such as momentum and weight decay are also used for optimization and to reduce the possibility of overfitting the model. The training process is performed in units of iteration called epochs. At each epoch, the model learns from the training data and is then evaluated using validation data to measure its performance.

Training is carried out iteratively so that the model is increasingly trained in recognizing patterns and features on vehicle license plates. In this study, the model was trained using three epoch configurations, which are 50, 100, and 150 epochs, with additional hyperparameters in the form of image size (imgsz) of 640 pixels and batch size of 16. This approach aims to evaluate the effect of the number of epochs on the final performance of the model.

F. Model Implementation and Testing

After obtaining training results from three epoch configurations, namely 50, 100, and 150 epochs, the best epoch configuration value will be used as a model that will be tested together with the use of PaddleOCR. This character extraction test on the vehicle license plate will use the bounding box that has been obtained from the detection results of the vehicle license plate from the best training model. If there is a bounding box detected from an image, then PaddleOCR will be used to read the characters on the vehicle license plate that already has a bounding box whose results will be displayed in text form.

IV. RESULTS AND DISCUSSION

This section will discuss the evaluation results of the YOLO11 model in detecting vehicle license plates using three epoch configurations. The model performance evaluation is performed based on four main metrics. Precision is used to measure the accuracy of the model in correctly identifying vehicle license plates by minimizing the number of false positives. Recall is used to describe how well the model captures all license plate objects contained in the image. The mean Average Precision (mAP@50) metric at the Intersection over Union (IoU) threshold of 50% to indicate the average detection accuracy of the model, is used as a metric to measure how well the model finds objects roughly. The mAP@50-95 metric at the Intersection over Union (IoU) threshold ranging from 50% to 95% is used to provide a more comprehensive picture of the model's ability to generalize or recognize objects of varying difficulty, meaning that the higher the value, the better the overall quality of the model. In addition, the analysis also focused on the effect of varying the number of epochs on model performance, both in terms of accuracy and efficiency. Tests also included simulations of real-world scenarios to assess the model's capabilities. These tests provide an in-depth look at the effectiveness of the model in accurately detecting vehicle license plates.

In training this model, several parameters are used, such as epochs configured into three parts, namely, 50, 100, and 150, the use of imgsz = 640 for an image size of 640x640 pixels, and size of batch = 16. The additional dataset is divided into 70%, 20%, and 10% respectively for training, validation, and test data respectively.

A. Dataset Preprocessing

The collected dataset will go through a processing stage to ensure that its structure is well-organized and ready to be used in model training. This process involves adjusting and optimizing the data to improve the accuracy and efficiency of the model in recognizing patterns. Results show that using an image resolution of 640x640 pixels consistently provides optimal performance in classification tasks. This resolution is considered to provide an ideal balance between accuracy and computational efficiency. It also retains important details in the image, supporting a more effective and accurate classification process. The table that portrays the dataset before and after preprocessing is presented in TABLE III.

TABLE III. DATASET BEFORE AND AFTER PREPROCESSING

Before	After
	
	
	
	
	

The dataset can be accessed at the given link. (universe.roboflow.com/nomor-plat-kendaraan/nomor-plat-kendaraan).

B. Model Training

The model will be trained using the YOLO11 algorithm to learn the previously prepared dataset. A total of 70% of the total dataset is used as training data, 20% is used for the validation process, and the rest 10% is used for the test. The table that represents the hyperparameter configuration is presented in TABLE IV.

TABLE IV. HYPERPARAMETER CONFIGURATION

Hyperparameter	Value
Optimizer	AdamW (Learning Rate = 0,002)
Epochs	50, 100, 150
Image Size	640 x 640
Batch	16

The table above is a selection of hyperparameter configurations used in model training. The use of AdamW optimizer with learning rate (0,002) provides stability and shorter time, especially on small datasets. The use of three epochs, which are 50, 100, and 150 as a comparison later to determine the best model from these three configurations, as well as the use of batches with a number of 16 so that the trained model is consistent and controllable.

C. Discussion and Analysis Results

At this stage, the model is trained to recognize patterns from the training data, then evaluated with validation data to test the ability of the model. The model is trained with three epoch configurations, which are 50, 100, and 150.

The visualization that portrays the training results of 50 epochs is presented in Fig. 4.

C.1. Training Results of 50 Epochs

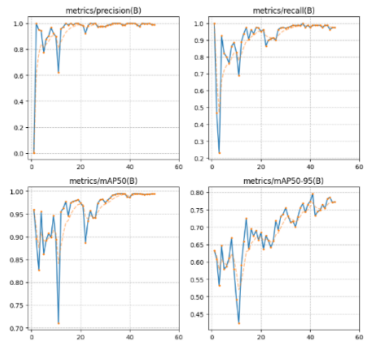


Fig. 4. Training result graphs with 50 epochs

The image above portrays the precision value, which was initially quite low, increased consistently until it reached 98.7%. Meanwhile, recall also shows a rapid spike and stabilizes at 97.4%. The mAP@50 value reached 99.4%, indicating a very high level of prediction accuracy. In addition, the mAP@50-95 value, which measures the accuracy of the model at various Intersection of Union (IoU) threshold levels, also gave good results with a value of 77.2%.

C.2. Training Results of 100 Epochs

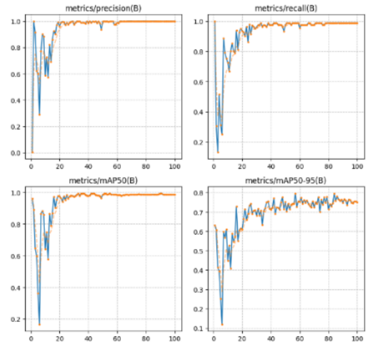


Fig. 5. Training result graphs with 100 epochs

The visualization that portrays the training results of 100 epochs is presented in Fig. 5. The precision value continues to rise until it reaches 99.9%, indicating the model's ability to recognize the target object with a very low error rate. Recall also increased and stopped at 98.7%, showing high sensitivity in detecting all relevant objects. On the mAP metric, the model recorded 98.7% at mAP@50 and 75% at mAP@50-95.

C.3. Training Results of 150 Epochs

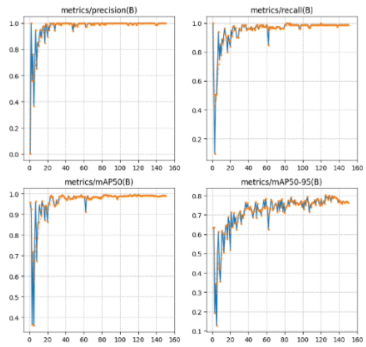


Fig. 6. Training result graphs with 150 epochs

The visualization that portrays the training results of 150 epochs is presented in Fig. 6. Overall, all metrics were highly consistent until the end of training. Precision peaked at 99.96%, which was the highest value of the three training epochs. Recall also performed highly at 98.7%, maintaining a balanced detection of all objects. This result recorded a value of 99.1% at mAP@50 and 76.3% at mAP@50-95.

The table that represents the training results of epoch variations is presented in TABLE V.

TABLE V. TRAINING RESULT OF EPOCH VARIATIONS

Epoch	Precision (%)	Recall (%)	mAP50 (%)	mAP50-95 (%)
50	98,73	97,42	99,40	77,23
100	99,93	98,75	98,68	75,02
150	99,96	98,75	99,06	76,28

The table above presents the evaluation results of three YOLO11 models trained with different number of epochs, namely 50, 100, and 150. The evaluation is based on four main metrics, namely precision, recall, mAP@50, and mAP@50-95, all of which are valued in percentage.

C.4. Analysis of The Training Results

Based on the results written in the table, the model trained for 50 epochs shows a very competitive performance, with a precision of 98.73% and a recall of 97.42%. These results show that the model is able to detect license plates accurately and consistently without many errors in the form of false positives and false negatives. The mAP@50 value that reaches 99.40% reflects the model's ability to identify objects with high overlap with the ground truth. Meanwhile, the mAP@50-95 value of 77.23% is the highest among all configurations, indicating the model's excellent spatial generalization ability even at stricter IoU thresholds. These results show that despite being trained in a relatively short time, the model is already highly optimized and efficient to be applied to the single-class detection task.

Increasing the number of epochs to 100 resulted in an almost perfect precision of 99.93%, and a recall that also increased to 98.75%. These results show that the model becomes very precise and makes very few errors in detecting license plates. However, the mAP@50 value decreased slightly to 98.68%, and mAP@50-95 dropped to 75.02%. This drop in mAP could be due to potential overfitting, where the model over-remembers the pattern of the training data and thus slightly degrades performance on tests with more varied conditions. Even so, this configuration still excels for applications that prioritize minimal false positives, such as security or automatic recognition systems that do not tolerate detection errors.

The model with 150 epochs of training recorded the highest precision of 99.96%, and recall remained at 98.75%, equivalent to the 100 epochs model. The mAP@50 value of 99.06% and mAP@50-95 of 76.28% place this model in an intermediate position between the other two configurations. Although it did not outperform the 50 epochs model in terms of spatial generalization, it still showed stability and consistency in performance after long training.

C.5. Testing Set Results

The results of the training model will be used to evaluate the test set that has previously been separated from the training data. This test set is used to test the generalization ability of the model to new data that has never been seen during the training or validation process. The evaluation is done using the best model from each epoch configuration (50, 100, and 150 epochs), which is the model with the highest performance based on the fitness value calculated from the combination of mAP@50 and mAP@50-95 metrics. This test produces key metric values such as precision, recall, mAP@50, and mAP@50-95, which become the basis for assessing how well the model can detect license plate objects in new images accurately and consistently. The results of this evaluation are then analyzed to determine the best model to be used in the character extraction stage using OCR.

The table that represents the test set results of epoch variations is presented in TABLE VI.

TABLE VI. TEST SET RESULTS OF EPOCH VARIATIONS

Epoch	Precision (%)	Recall (%)	mAP50 (%)	mAP50-95 (%)
50	99.81	100	99.50	80.73
100	99.93	100	99.50	79.21
150	99.96	100	99.50	78.58

The table above presents the test results of using three YOLO11 models trained with different number of epochs, namely 50, 100, and 150. The evaluation is based on four main metrics, namely precision, recall, mAP@50, and mAP@50-95, all of which are valued in percentage. Based on the evaluation of the test set, the performance results were obtained: the training model with 50 epochs achieved 99.81% precision, 100% recall, mAP@50 of 99.5%, and mAP@50-95 of 80.73%. While the model with 100 epochs resulted in precision 99.83%, recall 100%, mAP@50 of 99.5%, and mAP@@50-95 of 79.21%. The model with 150 epochs shows 99.84% precision, 100% recall, mAP@50 of 99.5%, and mAP@@50-95 of 78.58%. These results show that the three models have very high performance on precision, recall, and mAP@50 metrics. However, there is a slight decrease in the value of mAP@50-95 as the number of epochs increases, which indicates that even though the model continues to learn, the improvement in detection accuracy at various IoU scales does not increase significantly and even decreases slightly.

The effect of epoch variation on this test is that the model trained for 50 epochs produces the best performance with superior metrics on precision, recall, mAP@50, and mAP@50-95. This shows that the model has been able to learn quite well from the training data without overfitting. On the other hand, increasing the number of epochs to 100 and 150 does not result in significant performance improvement, and even tends to decrease at mAP@50-95, which indicates a decrease in the generalization ability of the model. Thus, it can be concluded that using too many epochs in this case can have a negative impact on the model's performance in identifying new data.

The above explanation can be concluded that the model with 50 epochs is chosen as the optimal model for the next step. This is because this configuration provides the best balance between high performance and training efficiency and produces the highest mAP@50-95 value compared to the other two configurations. In addition, using fewer epochs also reduces the risk of overfitting and saves training time and resources.

C.6. Result Discussion

Evaluation of models trained with different epoch configurations revealed some key findings regarding training efficiency, generalization, and performance stability. In previous research using similar methods with YOLOv8, it was found that all metrics experienced an increasing trend as the number of epochs during training increased. The precision value was initially slightly below 20%, then continued to increase progressively until it reached 96.6%. Recall also showed an increase as the number of epochs increased, with a final achievement of 96.4%. Meanwhile, the mAP@50 reached 99%. In the mAP@50-95 metric, the final value was recorded at 63.5%. These results show that increasing the number of epochs can contribute to the improvement of the model's performance in detecting and recognizing objects [22].

Another research that has been conducted with YOLOv8, with result as the mAP@50-95 value was recorded at 82.64%, reflecting the model's ability to detect objects with high accuracy at various IoU threshold levels. In addition, the mAP@50 metric reached 99.39%. The model's precision was recorded to be very high at 99.9%, meaning that false positive detection was minimal. Meanwhile, recall reached 98.5%, indicating the model's ability to capture almost all relevant objects [15].

Based on the training results, the model with 50 epochs already shows excellent detection capabilities, with 98.73% precision, 97.42% recall, 99.40% mAP@50, and the highest mAP@50-95 value of 77.23% among all training configurations. This shows that the model is able to effectively learn important features in a relatively short training duration. The high mAP@50-95 values also indicate better generalization across a wide range of IoU thresholds.

When the number of training epochs was increased to 100 and 150, the precision and recall values increased slightly, indicating that the model became more confident in identifying license plates. However, a downward trend of mAP@50-95 was observed at 75.02% for 100 epochs and 76.28% for 150 epochs, which implies that additional training leads to a loss of localization robustness. This decrease is consistent with potential overfitting, where the model memorizes the training pattern too specifically and loses its ability to generalize well to a wide range of spatial variations. Based on results above, those results can be achieved due to YOLO11's more advanced and efficient architecture as it performs better feature extraction and localization, resulting in a more accurate and stable bounding box at high IoU ranges.

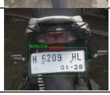
Further testing phases supported this analysis. All three models achieved near-perfect performance in terms of precision above 99.8% and 100% recall on the test images, indicating that the models maintained strong object recognition capabilities across the board. The model trained with 50 epochs again recorded the highest mAP@50-95 (80.73%), surpassing the 100 and 150 epoch models. This reinforces the previous observation that prolonged training does not improve and even slightly reduces the model's ability to generalize over a wider range of IoU thresholds.

C.7. PaddleOCR Testing

The model with 50 epochs is then used to detect the license plate area in the vehicle image, and the detection result in the form of a bounding box that becomes the input for PaddleOCR to extract the text characters from the license plate. This process aims to measure the effectiveness of the integration between the object detection model (YOLO) and the OCR system (PaddleOCR) in the automatic license plate recognition system scenario. To extract text information from the detected license plate area, the PaddleOCR, an optical character recognition system known for its speed and accuracy, as well as its ability to handle text in complex image conditions was used.

The table that represents the test set results with PaddleOCR utilized in daylight condition is presented in TABLE VII.

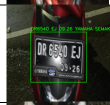
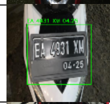
TABLE VII. PADDLEOCR RESULTS IN DAYLIGHT CONDITION

#	Picture	License Plate Number	OCR Result
1		H 6583 AFC	H6583 AFC
2		H 6602 HK	H 6602 HK
3		H 6629 NB	H 6629NB
4		H 6648 KK	H 6648 KK
5		H 6713 AAC	H6713 AAC
6		H 6725 EK	H 6725 EK
7		H 6774 KI	H6774KI
8		H 6823 DK	H 6823 DK
9		H 6912 QK	H6912 QK
10		H 6209 HL	H6209HL

The table above represents the test results of using the YOLO11 model and PaddleOCR integration in daylight condition. The results of this test can be concluded that the model can recognize vehicle license plates well and the use of PaddleOCR to read license plates is very accurate. The minor errors can be seen in character segmentation by OCR, or can be called less perfect in the context of post-processing formats.

The table that represents the test set results with PaddleOCR utilized in low-light condition is presented in TABLE VIII.

TABLE VIII. PADDLEOCR RESULTS IN LOW-LIGHT CONDITION

#	Picture	License Plate Number	OCR Result
1		DR 2086 NF	DR 2086 NF
2		DR 2217 CE	DR 2217CE
3		DR 6540 EJ	DR6540 EJ
4		EA 4931 XM	EA 4931 XM
5		DR 3656 ER	DR3656 ER
6		DR 2195 EU	DR2195 EU
7		DR 2652 HR	DR2652HR
8		DR 3545 UE	DR3545UE
9		DR 4155 MP	DR4155 MP
10		DR 5435 ET	DR5435.ET

The table above represents the test results of using the YOLO11 model and PaddleOCR integration in low-light condition. The results of this test can be concluded that the model can recognize vehicle license plates well and the use of PaddleOCR to read license plates is quite accurate. The minor errors can be seen in character segmentation by OCR, or can be called less perfect in the context of post-processing formats

Based on two conditions above, it is known that either in daylight or low-light conditions, the model can detect the license plate almost perfect and OCR can utilize the bounding box that was made by the model to read the number of the license plate. The license plate is still well detected by YOLO11, the character reading results by PaddleOCR show some minor discrepancies, such as the loss of spaces between characters, especially between initial letters and license plate numbers. The spacing issue is not a fatal flaw in the system, but rather a limitation in character segmentation by OCR. To overcome this, a regex validator approach can be used as a post-processing stage, by applying regular expression patterns based on the standard Indonesian license plate format (such as, AA 1234 BBB). This regex works by reordering the OCR results into the appropriate format, even if spaces are lost or not perfectly legible.

The picture that portrays the plate detection outside of the license plate standard is presented in Fig. 7.

Fig. 7. Plate detection outside of the license plate standard

From case of image above, the YOLO11 model demonstrated the ability to detect vehicle license plates despite “extreme” conditions such as variations that were not previously trained. This can be explained by the way CNN-based detection models such as YOLO work, which focuses more on spatial features in the form of shapes, lines, and characteristic patterns rather than specific colors. YOLO accepts input images in RGB format and processes them by emphasizing the edges, contours, and structure of the license plate object, thus being able to recognize plates with different background colors such as red plates or white plates with red writing. In addition, the detection process in YOLO does not rely on a specific color space such as grayscale or YCbCr, but rather maintains the image in 3 channels (RGB), so as long as the physical pattern of the license plate is still similar to the training data, the model can generalize well. These results show that a model trained with variations in plate shape and structure can still detect plates outside its training color distribution, demonstrating the good generalization level of the YOLO11 model.

V. Conclusion and Suggestion

A. Practical Implementation Roadmap

The table that represents the CCTV specifications is presented in TABLE IX.

TABLE IX. CCTV Specifications

No.	CCTV Brand	Resolution	Estimated Price	Link
1	Hikvision DS-2CE16D0T-EXIPF	1080p	Rp294.000	https://tk.tokopedia.com/ZSBeTnDV3
2.	TP-Link Tapo C500 IP Cam	1080p	Rp490.000	https://tk.tokopedia.com/ZSBeTVEFg
3.	Xiaomi Mi C 200 1080p	1080p	Rp349.000	https://tk.tokopedia.com/ZSBeTgbnW
4.	Bardi Smart PoE IP Cam	1080p	Rp825.000	https://tk.tokopedia.com/ZSBeT34by

Based on camera specification table above, in order to support the practical implementation of the YOLO11 and PaddleOCR-based vehicle license plate detection and reading system, several types of CCTV cameras that are commonly available in the Indonesian online marketplace were selected. The table above presents four variants of CCTV cameras with 1080p resolution, which is the recommended resolution to maintain the clarity of the license plate characters to support detection and OCR accuracy. The Hikvision DS-2CE16D0T-EXIPF camera was chosen because it is the most economical (est. Rp294.000), and has been proven to be widely used in security and surveillance systems. The TP-Link Tapo C500 and Xiaomi Mi C200 cameras offer wireless connectivity features and installation flexibility, are suitable for indoor use with easy installation, and are in the mid-range price range (est. Rp349.000). Meanwhile, the Bardi Smart PoE IP Cam camera is in a slightly higher class (est. Rp825.000), but is equipped with the Power over Ethernet (PoE) feature which increases connection stability and reliability for outdoor use.

All selected cameras support 1080p resolution which is ideal for maintaining license plate image quality in various lighting conditions. With affordable prices and technical specifications that support system needs, the combination of these cameras and edge devices such as Raspberry Pi or Jetson Nano [28] allows the development of an efficient, cost-effective ALPR system that can be widely implemented, both in building parking environments, housing, and MSMEs (Micro, Small, and Medium Enterprises).

B. Conclusion and Suggestion

The conclusion of this research shows that through training the YOLO11 model with various epochs (50, 100, and 150), the results show that the model with 50 epochs provides the most balanced results in terms of metrics. The model achieved of 4 main metrics, which are precision of 99.81%, a recall of 100%, 99.50% of mAP@50 and 80.73% of mAP@50-95, indicating a very high level of bounding box detection accuracy and strong generalization ability to various input conditions and reflecting a balance between the ability to recognize correct license plates and avoid detection errors. When compared to previous studies using similar evaluation metrics, it can be concluded that YOLO11 is competitive and produces good detection performance. In addition, YOLO11 offers the advantage of using fewer parameters than previous versions of the model, making it more efficient without sacrificing accuracy. The model is then used as a base model for the vehicle license plate detection results which are further processed by PaddleOCR for text extraction. From a limited test of ten image samples for each daylight and low-light conditions, the system showed a great ability to detect and read the characters on the license plate accurately.

It is recommended for future research to expand the type of the license number plate and increasing the number of images in the vehicle variation dataset that is not limited to two motorcycles. Integration in real-time systems is also needed to implement the system in real life. Exploration of other algorithms that have similar capabilities can also be used as a model comparison to measure a better model.

REFERENCES

[1] D. Pramesti, N. L. P. J. Andini, D. A. K. Raharjo, and A. D. Dwipayana, "Efektivitas Penggunaan Moda Transportasi Umum dengan Kendaraan Pribadi," *Indonesian Journal of Multidisciplinary on Social and Technology*, vol. 2, no. 1, pp. 6–16, Jan. 2024, doi: 10.31004/ijmst.v2i1.246.

[2] Y. A. Usen and C. Hayat, "Design and Build Vehicle Plate Detection System using You Only Look Once Method Based on Android," *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 4, pp. 807–818, Aug. 2023, doi: 10.52436/1.jutif.2023.4.4.791.

[3] M. R. Ali and A. Y. Husodo, "Pengenalan Plat Kendaraan Bermotor Menggunakan Metode Gradien Karakter dan BPNN (Backpropagation Neural Network) (Licences Recognition Using Gradient Character and Backpropagation Neural Network)." [Online]. Available: http://jcosine.if.unram.ac.id/

[4] I. H. Al amin and A. Aprilino, "Implementasi Algoritma YOLO dan TesseractOCR pada Sistem Deteksi Plat Nomor Otomatis," *Jurnal Teknoinfo*, vol. 16, no. 1, p. 54, Jan. 2022, doi: 10.33365/jti.v16i1.1522.

[5] K. Khotimah, M. Taqijjuddin Alawiy, and B. M. Basuki, "Pendeteksi Plat Nomor Kendaraan Bermotor Berbasis Algoritma YOLO (You Only Look Once) Menggunakan Kamera CCTV," *SCIENCE ELECTRO*, vol. 1, no. 8, 2023.

[6] W. Purwanto and M. Septiani, "Implementasi Automatic License Plate Recognition untuk mengurangi pelanggaran lalu lintas berbasis Artificial Intelligence," *Informatics for Educators and Professionals : Journal of Informatics*, vol. 8, no. 2, pp. 148–157, 2023.

[7] B. P. Nugroho, Y. Prihati, and S. T. Galih, "Implementation of YOLOv5 Algorithm in Vehicle License Plate Detection Application Design," *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 7, no. 3, 2024.

[8] F. Aulia, A. Harahap, R. Mardianson Sinaga, K. Arifin, and K. Saputra, "Implementasi Algoritma Convolutional Neural Network untuk Mendeteksi Penyakit Ginjal." [Online]. Available: http://jtika.if.unram.ac.id/index.php/JTIKA/

[9] Reezky Illmawati and Hustinawati, "YOLOv5 for Vehicle Plate Detection in DKI Jakarta," *Jurnal Ilmu Komputer dan Agri-Informatika*, vol. 10, no. 1, pp. 32–43, May 2023, doi: 10.29244/jika.10.1.32-43.

[10] N. Nanda, E. D. A. Eriene, T. A. Tesa Anggraini, and V. S. Vina Sumsari, "Rancang Bangun Sistem Parkir Otomatis Dengan Menggunakan Long Range RFID Reader Berbasis Arduino Uno," *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, vol. 4, no. 1, pp. 70–78, Jun. 2025, doi: 10.62712/juktisi.v4i1.333.

[11] W. Dewanto, A. Fathurrahman, and A. Bejo, "Evaluasi Platform Perangkat Keras Sistem Tertanam untuk Unit Kontrol Parkir Otomatis," vol. 12, Nov. 2023.

[12] S. R. Siregar, A. Azlan, and P. Pristiwanto, "Aplikasi Sistem Informasi Parkir Kampus Menggunakan Model Single Channel Single Phase Service," *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*, vol. 23, no. 1, p. 171, Feb. 2024, doi: 10.53513/jis.v23i1.9607.

[13] A. Meirza and N. R. Puteri, "Implementasi Metode YOLOv5 dan Tesseract OCR untuk Deteksi Plat Nomor Kendaraan," *Jurnal Ilmu Komputer dan Desain Komunikasi Visual*, vol. 9, no. 1, 2024.

[14] L. Satya, M. R. D. Septian, M. W. Sarjono, M. Cahyanti, and E. R. Swedia, "Sistem Pendeteksi Plat Nomor Polisi Kendaraan dengan Arsitektur YOLOv8," *Sebatik*, vol. 27, no. 2, pp. 753–761, Dec. 2023, doi: 10.46984/sebatik.v27i2.2374.

[15] Anthony, Herman, and A. Yulianto, "Pengembangan Sistem Pengenalan Plat Nomor Indonesia Menggunakan YOLOv8 dan EasyOCR," *Jurnal Ilmiah Komputasi*, vol. 23, no. 4, Dec. 2024, doi: 10.32409/jikstik.23.4.3659.

[16] L. Zheng You, B. Z. Jun, P. Ying Han, O. Shih Yin, K. Wee How, and H. F. San, "Integration of YOLOv8 and PaddleOCR for Car Plate Recognition System in Diverse Environments: Malaysia," 2025.

[17] T. Maulana and Erwin Harahap, "Deteksi Plat Nomor Kendaraan Menggunakan Algoritma YOLOv5 dengan Metode Convolutional Neural Network," *Jurnal Riset Matematika*, pp. 103–112, Dec. 2024, doi: 10.29313/jrm.v4i2.5060.

[18] S. Amanda Putri, G. Ramadhan, Z. Alwildan, I. Irwan, and R. Afriansyah, "Perbandingan Kinerja Algoritma YOLO Dan RCNN Pada Deteksi Plat Nomor Kendaraan," *Jurnal Inovasi Teknologi Terapan*, vol. 1, no. 1, pp. 145–154, Feb. 2023, doi: 10.33504/jitt.v1i1.30.

[19] I. Irfan, A. Patombongi, and C. Cakra, "Pelatihan dan Pengujian YOLO (You Only Look Once) untuk Mendeteksi Plat Kendaraan," *Simtek : jurnal sistem informasi dan teknik komputer*, vol. 8, no. 2, pp. 404–411, Oct. 2023, doi: 10.51876/simtek.v8i2.362.

[20] M. I. Wiawan, S. Nur, and W. S. Gumelar, "Prototipe Sistem Deteksi dan Pengenalan Plat Nomor Sepeda Motor di Indonesia Menggunakan YOLOv4," Nov. 2023, doi: https://doi.org/10.30999/abtek.v5i2.3029.

[21] S. Gunawan Zain and A. Ardilla, "Detection of Vehicle Type and License Plate with Convolutional Neural Network Model YOLOv7," *Jurnal Teknik Informatika (JUTIF)5.2.XX*, vol. 5,

no. 2, pp. 621–636, 2024, doi: 10.52436/1.jutif.2024.5.2.XX.

[22] F. Hidayat, N. Billy, N. R. Permana, and M. E. Hariady, "Penerapan You Only Look Once dan DeepSORT untuk Deteksi Plat Nomor Kendaraan," *Jurnal Telematika*, vol. 19, no. 2, Mar. 2025, doi: 10.61769/telematika.v20i2.676.

[23] G. Ramadhan, R. D. A. Khairiyah, S. Natania, and A. Harris, "Vehicle Police Number Detection using YOLOv8," *Media Journal of General Computer Science*, vol. 1, no. 2, pp. 62–70, Jun. 2024, doi: 10.62205/mjgcs.v1i2.25.

[24] D. Fitri Utaminingrum and R. Regasari Mardi Putri, "Purwarupa Sistem Klasifikasi Warna Plat Kendaraan Berbasis YOLOv8 pada Gerbang Masuk UB," Mar. 2024. [Online]. Available: http://j-ptiik.ub.ac.id

[25] L. A. Putra and Y. Yohannes, "Deteksi Plat Nomor Kendaraan Menggunakan Metode YOLOv8," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 12, no. 2, pp. 210–221, Jun. 2025, doi: 10.35957/jatisi.v12i2.11261.

[26] G. S. Susilo, D. Utami, and K. Putri, "Deteksi Objek dan Pengenalan Karakter Plat Nomor Kendaraan dengan Metode Deep Learning," *Indonesian Journal of Electronics and Instrumentation Systems (IJEIS*, vol. x, No.x, pp. 1–5, 2023, doi: 10.22146/ijeis.xxxx.

[27] H. Moussaoui, N. El Akkad, M. Benslimane, W. El-Shafai, A. Baihan, C. Hewage, and R. S. Rathore, "Enhancing Automated Vehicle Identification by Integrating YOLOv8 and OCR Techniques for High-Precision License Plate Detection and Recognition," *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-65272-1.

[28] M. R. Rais, F. Utaminingrum, and H. Fitriyah, "Sistem Pengenalan Plat Nomor Kendaraan untuk Akses Perumahan menggunakan YOLOv5 dan Pytesseract berbasis Jetson Nano," 2023. [Online]. Available: http://j-ptiik.ub.ac.id