Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.5, October 2025, pp. 2320~2328
e-ISSN: 2548-6861

YOLOV12 Based on Stationary Vehicle for License Plate Detection

The, Obed Danny Kurniawan [1], Eko Hari Rachmawanto [2*]
* Department of Informatics Engineering, Universitas Dian Nuswantoro, Semarang, Indonesia
eko.hari@dsn.dinus.ac.id [2]

Article Info

Article history:

Received 2025-08-28
Revised 2025-09-10
Accepted 2025-09-19

Keyword:

YOLOv12
Vehicle License plate
Recognize
Accuracy

ABSTRACT

The use of technology for vehicle license plate recognition in this modern era is increasingly developing in supporting the needs of more effective transportation system management. This research aims to design and implement a vehicle license plate recognition system with the YOLOv12 (You Only Look Once) algorithm. The use of the YOLOv12 algorithm in license plate recognition is due to its superiority in detecting and recognizing objects in real-time with high accuracy. This research method will involve collecting a dataset of vehicle license plates from various viewing angles, lighting conditions, license plate colors, and the shape of the license plate. These datasets are then used to train an adapted YOLOv12 model to detect and recognize characters on license plates. Tests are conducted by measuring the detection accuracy, processing speed, and robustness of the detection system to disturbances such as noise and variations in environmental conditions when detecting license plates. The results of the study shown that this system yielded accuracy rate of 97.5%, recall of 95.4%, precision of 96.7%, and is capable of recognizing characters on vehicle license plates with an accuracy rate of 88%, recall of 87%, and precision of 85.8%. The average processing time is 1 second per image on CPU and 20 seconds per image on GPU. The system's ability to detect vehicle license plates shows that the YOLOv12 algorithm can be used for large-scale vehicle license plate system implementation. The significance of these results lies in their potential application in various fields such as parking management systems, traffic management, and law enforcement, which can improve efficiency and safety.

This is an open access article under the CC–BY-SA license.

I. INTRODUCTION

Transportation has been an important necessity for humans since ancient times. With transportation, humans can carry out their activities efficiently[1]. Human activities can take many forms, such as economic, cultural, and social activities. Of course, the continuous development of transportation to date has been very beneficial to human life. However, the benefits enjoyed by humans cannot be separated from the negative impacts that occur. With the abundance of transportation available, traffic conditions have become increasingly congested, leading to many violations committed by humans, such as traffic light violations, hit-and-run incidents, and many more [2]. Transportation itself cannot be separated from license plates. License plates are a form of identification for transportation [3]. License plates are usually black, yellow, red, and white discs located on the front and rear of vehicles [4]. License plates also have characters on them, consisting of letters and numbers. The characters on the license plate indicate the type of vehicle and the province where the license plate is registered [5].

Indonesian license plates are rectangular and have a black/yellow/red/green background, depending on the function and use of the vehicle [6]. Private transportation has a black background color on the license plate, public transportation has a yellow background color on the license plate, government/agency transportation has a red background color on the license plate. And one more thing, transportation in the free trade zone has a green background color on the license plate [7]. Indonesian license plates consist of 1-2 letters at the front of the plate, which indicate where the vehicle is registered. The middle of the license plate consists of 1-4 numbers that reflect the vehicle's police number. The police numbers used range from 1-2999 and 8000-8999. The back of the license plate features a series of letters from A to Z, indicating the sequence of the police number in the middle. When the police number that has been established reaches the last position in the sequence, the next vehicle to register will receive a new police number sequence, and the series of letters at the back will change [8].

YOLO12 is the latest object detection model that introduces an attention-centric mechanism to achieve state-of-the-art accuracy while maintaining real-time inference speed. This model achieves an impressive accuracy of 40.6% (mAP) while processing images in just 1.64 milliseconds on an Nvidia T4 GPU, outperforming YOLO v10 and YOLO v11 [11]. The difference between yolov12 and previous yolo models is that yolov12 uses an attention-centric mechanism, which allows the model to focus on the most important parts of the image rather than processing everything evenly, resulting in better computational efficiency [13]. For vehicle license plate detection applications, YOLO12 is ideal due to its real-time processing capabilities and high accuracy.

Researchers Ranjan Sapkota and Manoj Karkee conducted research to detect apples using three models: YOLOv10, YOLOv11, and YOLOv12. From their research results, they obtained the highest precision, recall, and mAP50 scores in detecting clustered and multiple apples in a single image, with a precision of 0.916, recall of 0.969, and mAP50 of 0.978 using the YOLOv12n model. Besides precision, recall, and mAP50, the YOLOv12n model also ranked second in image processing speed at 5.6 ms [12].

Previous researchers by Islam et al was conducted research on license plate recognition in 2023 using an efficient method with 4 stages, namely preprocessing (improving image quality in different lighting conditions), license plate extraction, character segmentation on the plate, and character recognition. After preprocessing, morphological operations are performed to extract the license plate area. For character segmentation, the bounding box method is used, and character recognition is done by template matching. The test results show that this system is able to recognize license plate characters with high accuracy, which is 94.17% using MATLAB software [9]. Reda Al-batat et al in 2022 also conducted research to create a license plate recognition program using the YOLOv2 and YOLOv4 object detection algorithms. YOLOv2 was used for the initial stage of vehicle detection, while YOLOv4 was used for the advanced stage with various data augmentation techniques. The method was tested using five public datasets from different regions, and managed to achieve an average recognition accuracy of 90.3% with processing speed (FPS) remaining good on low-end GPUs [10].

In this study, the researcher aims to develop an existing license plate recognition system to be more accurate and efficient. To achieve this goal, the researcher applies an artificial intelligence-based system using the You Only Look Once (YOLO)v12 object detection algorithm. This algorithm is used to recognize the license plate area on a vehicle and detect the characters contained in the license plate. Overall, the operation of this recognition system is as follows: first, YOLOv12 detects the location of the license plate and marks it with a bounding box. Next, YOLOv12 focuses detection on the characters within the bounding box area of the license plate, while ignoring other characters outside that area.

II. METHOD

A. Research Framework

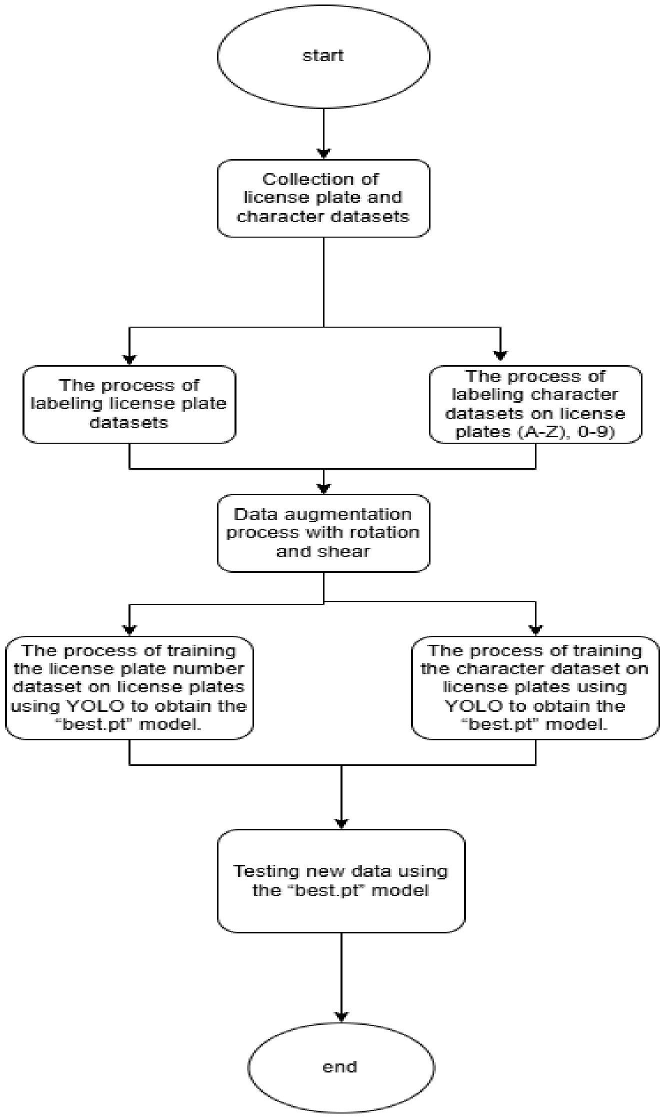


Figure 1. Research Framework

Based on Figure 1, the process begins with the initial stages:

1. Data Collection - The initial stage begins with collecting datasets consisting of vehicle license plate images and character datasets. This dataset serves as the main foundation for training the detection model that will be developed. The quality and quantity of the dataset at this stage greatly determines the final performance of the system to be built.
2. Parallel Labeling - The labeling process is carried out simultaneously for two different types of datasets: the license plate dataset and the character dataset. License

plate dataset labeling aims to identify the location of plates in images, while character dataset labeling (A-Z, 0-9) is used to identify each letter and number on the plate. This parallel process increases time efficiency in training data preparation.

3. Data Augmentation - The labeled datasets are then expanded using augmentation techniques such as rotation and shear to create more diverse data variations. This technique aims to improve the model's generalization ability in recognizing objects from various angles and conditions. Data augmentation also helps overcome overfitting problems by increasing the amount of training data.
4. Dual Model Training - Two YOLO models are trained simultaneously using the augmented datasets. The first model is trained to detect and localize license plates in vehicle images, while the second model is trained to recognize individual characters on license plates. These two models will work sequentially in the final system to provide comprehensive detection results.
5. Model Optimization - Both models are trained iteratively until they achieve the best performance, which is saved in the best.pt model file. This process involves parameter adjustment, loss function monitoring, and evaluation of metrics such as precision, recall, and mAP. The best model is selected based on validation performance to ensure optimal generalization capability.
6. System Testing - The best trained model is tested using new data that has never been seen during the training process. This stage aims to validate the overall system performance and ensure the model can work well under real-world conditions. The test results will determine whether the system is ready for implementation or needs further improvement.
7. Completion - The system has been completed and is ready for use after going through all testing and validation stages. The optimized model can be integrated into larger applications or systems for automatic license plate detection purposes. This system is ready for deployment and can be used for various applications such as automatic parking systems or traffic monitoring.

B. Collection of License Plate Number and Character Datasets

The researchers collected both types of datasets independently using a rear camera with a resolution of 50 megapixels and an aperture of f/1.8. The use of this high-specification camera was intended to obtain images with good sharpness and clarity, thereby facilitating the annotation process and improving the accuracy of the detection model to be trained. From this image acquisition process, the researchers successfully collected 4,000 images for the license plate dataset and 3,000 images for the character dataset. Figure 2 illustrates the number of initial datasets used in this study, which consists of two types of datasets, namely license plate datasets and character datasets.

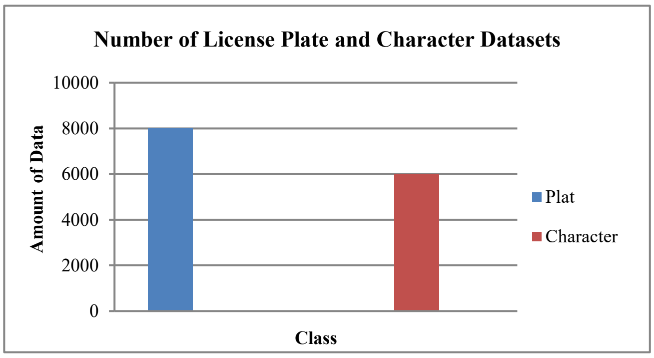


Figure 2. Number of Plate and Character Datasets

Researchers apply data augmentation techniques in an effort to increase the amount of available data. Data augmentation is the process of expanding a dataset with various modifications such as rotation, flipping, zooming, shifting, brightness adjustment, and other methods [14] These techniques aim to increase diversity and possibilities in training data, such as simulating varying real-world conditions in the field, including different shooting angles, uneven lighting, and differences in camera quality, so that the model can be more optimal in recognizing various types of data [15]. The researchers used an augmentation ratio of 1:2, where 1 represents the ratio of the initial dataset size and 2 represents the ratio of the dataset size after augmentation. After the augmentation process was completed, the number of datasets increased significantly. The license plate dataset increased from 4,000 to 8,000 images, while the character dataset increased from 3,000 to 6,000 images. This increase provides additional variation that can improve the generalization performance of the license plate character detection and classification model.

The dataset that has been collected and augmented is then divided into three main groups, namely training data, testing data, and validation data. This division follows the commonly used ratio in training artificial intelligence models, namely 70% for training data, 10% for testing data, and 20% for validation data. For the license plate dataset, which consists of a total of 8,000 images, this division results in 5,600 images for training data, 800 images for testing data, and 1,600 images for validation data. This division is intended to ensure that the model can be trained optimally, its accuracy tested, and validated to ensure that the model does not overfit the training data.

Meanwhile, the character dataset, which consists of a total of 6,000 images, was also divided using the same ratio. A total of 4,200 images were used as training data to train the model to recognize characters on license plates, 600 images were used as test data to measure the initial performance of the model, and the remaining 1,200 images were used for validation to assess the stability and ability of the model to recognize data it had never seen before. This division is a crucial foundation for building a reliable and accurate license plate detection and character recognition system. Through systematic data collection and processing methods, researchers ensure that the resulting model can be applied in various real-world conditions with high precision.

C. *YOLO (You Only Look Once) Algorithm Architecture*

YOLO (You Only Look Once) is an object detection algorithm developed by Redom et al in 2015. YOLO has developed rapidly over the years, reaching YOLOv12 in 2025 [16]. From Figure 3, YOLO works by dividing an image into an S x S grid, which is 3 x w3. After dividing the image into a grid, YOLO will check for the presence of objects in each grid. Each grid that has an object will be marked with a bounding box and confidence score (Pc) to determine how confident YOLO is in detecting the object. The bounding box prediction consists of five values: pc, bx, by, bh, and bw. Pc is a confidence value that indicates how confident the model is in detecting objects in the bounding box. Bx and by are the x and y coordinates of the center point of the bounding box on each grid that contains an object. Bh is the ratio of the height of the bounding box to the entire image. Bw is the ratio of the width of the bounding box to the entire image [17]. As YOLO evolves from version to version, the architecture of each version differs. These differences can be attributed to several factors, such as the addition of convolutional parameters, new modules, and new residuals.

Figure 3. How YOLO works [17]

Figure 4 shows the differences between the YOLOv12 architecture and its previous versions. These differences include the addition of the Area Attention (A2) module, residual *R-ELAN* in the *YOLOv12* architecture, flashAttention, and hybrid CNN-Transformer design [13]. Area Attention (A2) has the function of reducing the computational workload of YOLOv12 by 50% through spatial rearrangement. Residual R-ELAN has the function of optimizing large attention-based models with periodic residual designs and the latest feature aggregation methods. [12] With the addition of the A2 module and R-ELAN residual, the computational process using YOLOv12, such as model training, becomes faster with more optimal results.

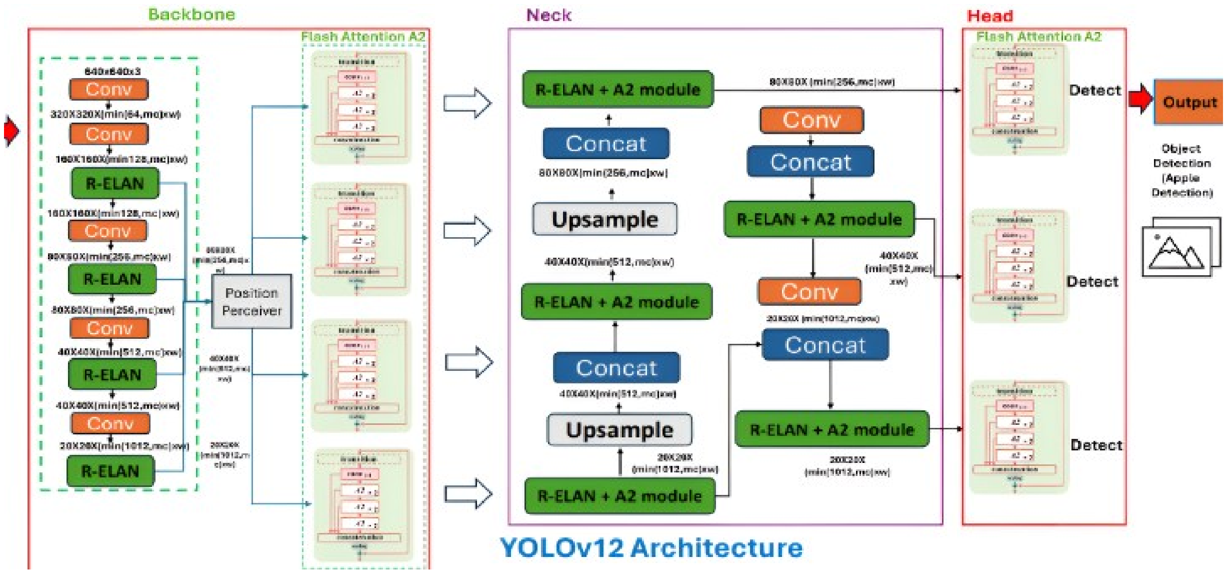


Figure 4. YOLOv12 Architecture [12]

D. *Evaluation of the YOLO (You Only Look Once) Matrix*

YOLO has several evaluation metrics for object detection. Here are some metrics that are often used by YOLO:

1) *Average Precision (AP):* This metric is often used by object detection algorithms to measure model accuracy. It calculates the area under the precision-recall curve for various thresholds. The formula for calculating AP is shown in the equation (1) [18].

$$AP = \sum_{i=1}^{n}(R_i - R_{\{i-1\}})Pi \tag{1}$$

2) *Intersection of Union (IoU):* This metric measures the overlap between the detection bounding box and the ground truth bounding box. The formula for calculating (IoU) is shown in the equation (2) [18].

$$IoU = \frac{The\ slice\ area}{The\ combined\ area} \quad (2)$$

3) *Precision (P):* This metric measures the accuracy of positive predictions generated by the model, i.e., the extent to which the model's predictions are correct compared to all positive predictions. The formula for calculating precision is shown in the equation (3) [19].

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{All\ prediction} \quad (3)$$

4) *Recall (R):* This metric measures the extent to which the model can detect correctly according to the ground truth in the dataset. The formula for calculating recall is shown in the equation (4) [20].

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{All\ Ground\ Truth} \quad (4)$$

E. Dataset Annotation/Labeling Process

Annotation is the process of assigning class names in the form of bounding boxes to objects that are to be detected. The annotation format for YOLOv12 is .txt, and .yaml [21]. The dataset annotation process is done manually by drawing a bounding box around the object to be detected using a bounding box annotation tool called Roboflow [22]. The annotation process using Roboflow is shown in Figure 5. The annotation process of providing bounding boxes around objects is essential when training models used for object detection. During the training process, the model can clearly recognize the class of the detected object based on the bounding box. In the license plate dataset, researchers performed the annotation process by assigning the class name "license plate" to the license plate dataset. Here, we performed the annotation process by assigning class names consisting of letters "A-Z" and numbers 0-9 to the dataset containing letters or numbers. The numbers of data per label is shown in Figure 6.

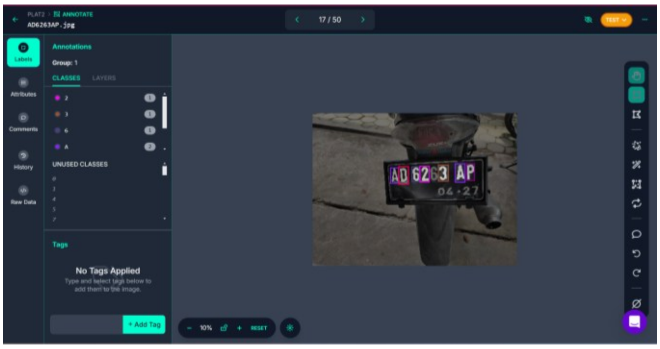


Figure 5. Annotation Process with Roboflow

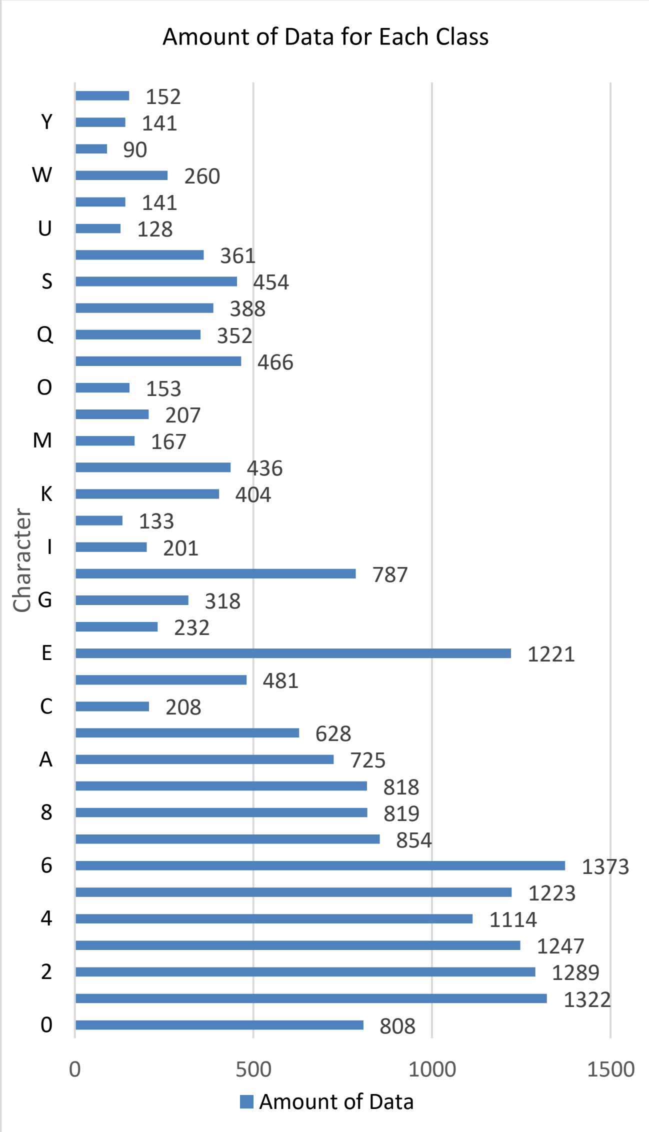


Figure 6. Amount of Data on Each Label

F. Data Augmentation Process

Data augmentation is a technique for expanding image datasets and increasing the diversity of images in the dataset. Image augmentation consists of several techniques such as image rotation, image flipping, image shearing, and image enlargement [15]. Here, we used image rotation and image shearing techniques. The rotation technique used was to rotate the image 25 degrees clockwise and 25 degrees counterclockwise. The shearing technique used involved a 45-degree horizontal rotation clockwise and counterclockwise, and a 30-degree vertical rotation clockwise and counterclockwise. The data augmentation techniques for rotation and shearing are illustrated in Figure 7.

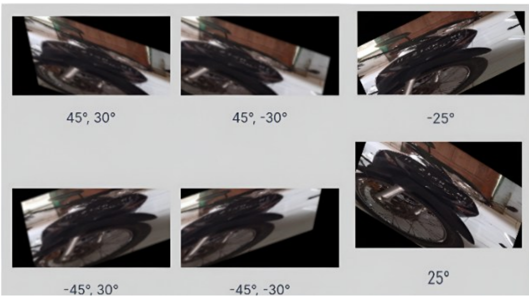


Figure 7. Data Augmentation Shear and Rotation

G. Training Dataset with YOLOv12

We used the YOLOv12 model to train the license plate and character dataset. Researchers will conduct training on the license plate and character dataset, which has been divided into 70% training data, 10% testing data, and 20% validation data. The dataset training process using YOLOv12 was carried out on Google Colab with a T4 GPU processor to support a fast-training process without taking up too much time. The dataset training process for license plate and character datasets uses the yolo12s.yaml model with 30 epochs, 16 batches, 640x640 image size, and the following total number of parameters. A summary of all parameters in the yolov12s architecture is that yolov12s uses 497 layers, 9,096,851 parameters, 9,096,835 gradients, and 19.6 GFLOPs.

TABLE 1
PROPOSED YOLO12 ARCHITECTURE

Layer	Previous layer	n (Repetition)	Params	Module	Arguments
0	-1	1	928	ultralytics.nn.modules.conv.Conv	[3, 32, 3, 2]
1	-1	1	9344	ultralytics.nn.modules.conv.Conv	[32, 64, 3, 2]
2	-1	1	26080	ultralytics.nn.modules.block.C3k2	[64, 128, 1, False, 0.25]
3	-1	1	37120	ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
4	-1	1	103360	ultralytics.nn.modules.block.C3k2	[128, 256, 1, False, 0.25]
5	-1	1	590336	ultralytics.nn.modules.block.C3k2	[256, 256, 3, 2]
6	-1	2	677120	ultralytics.nn.modules.conv.A2C2f	[256, 256, 2, True, 4]
7	-1	1	1180672	ultralytics.nn.modules.conv.Conv	[256, 512, 3, 2]
8	-1	2	2664960	ultralytics.nn.modules.block.A2C2f	[512, 512, 2, True, 1]
9	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
10	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
11	-1	1	345856	ultralytics.nn.modules.block.A2C2f	[768, 256, 1, False, -1]
12	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
13	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
14	-1	1	95104	ultralytics.nn.modules.block.A2C2f	[512, 128, 1, False, -1]
15	-1	1	147712	ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
16	[-1, 11]	1	0	ultralytics.nn.modules.conv.Concat	[1]
17	-1	1	296784	ultralytics.nn.modules.block.A2C2f	[384, 256, 1, False, -1]
18	-1	1	590336	ultralytics.nn.modules.conv.Conv	[256, 256, 3, 2]
19	[-1, 8]	1	0	ultralytics.nn.modules.conv.Concat	[1]
20	-1	1	1511424	ultralytics.nn.modules.block.C3k2	[768, 512, 1, True]
21	[14, 17, 20]	1	819795	ultralytics.nn.modules.head.Detect	[1, [128, 256, 512]

III. RESULTS AND DISCUSSION

In this section, the researcher will describe the results obtained from vehicle license plate recognition tested by the researcher using the YOLOv12 method, as well as analyze the data obtained during the research.

A. Analysis of MAP results from the license plate dataset

In this study, vehicle license plate character detection was performed using the YOLOv12 method. The mAP result was obtained by training the dataset with 30 epochs. The evaluation results of the model on the character dataset shown in Table 1 indicate overall satisfactory performance. With a total of 595 images evaluated, the model achieved a precision (Box P) of 0.858 and a recall (Box R) of 0.875, indicating the model's strong ability to accurately identify objects and capture most relevant examples. The mAP50 value of 0.88 indicates moderate detection performance at the IoU threshold of 0.5, while the mAP50-95 value of 0.596 reflects the model's stronger capability across various IoU levels, with higher average performance across different precision levels. The total number of examples evaluated, 4000, indicates that this dataset is sufficiently large to evaluate the model comprehensively. Overall, these results show that the model has solid performance for character detection tasks, although there is still room for improvement, particularly in the mAP50, which remains below 0.6.

The results of the license plate training dataset conducted by researchers showed excellent performance with 793 images and 860 license plate instances. The model achieved a precision of 96.7% (BOX P), which means it rarely gives false positives, and a recall of 95.4% (BOX R), which shows its ability to detect most license plates without missing any. The mAP50 value of 97.5% indicates very high detection accuracy at an IoU threshold of 0.5, while the mAP50-95 value of 79.8% shows good performance even with stricter localization requirements. Overall, this model is highly effective for license plate detection with high accuracy and reliability.wo

TABLE 2
MAP RESULTS FROM CHARACTER DATASET

Class	Images	Box (P)	Box (R)	mAP50	mAP50-95	Instance
all	595	0.858	0.87	0.88	0.596	4000

TABLE 3
MAP RESULTS FROM LICENSE PLATE DATASET

Class	Images	Box (P)	Box (R)	mAP50	mAP50-95	Instance
all	793	0.967	0.954	0.975	0.798	860

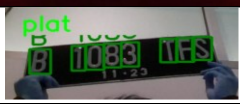
B. Analysis of bounding box prediction results

Each letter or number on the license plate is marked with a rectangular box to indicate the character's location. The color and label of the box indicate the character's identity and the model's confidence level in its prediction. Figure 8 shows the results of character detection on vehicle license plates using bounding boxes. The detected characters are E, 3, 7, 0, 8, T, and U, where this sequence of characters forms the vehicle license plate "E 3708 TU." The numbers above the boxes indicate the model's confidence level in predicting the character, for example, the character "0" has a confidence value of 0.82, meaning the model is 82% confident that the character is "0." These results indicate that the detection model is capable of working fairly well in recognizing characters on vehicle license plates. However, the accuracy of these predictions may vary, influenced by factors such as the distance of the license plate from the camera, the presence of noise around the plate, and the lighting conditions during the detection process. The image above shows the detection results on a vehicle license plate using a bounding box. Each license plate is detected correctly and in accordance with the license plate size but has a different confidence value. This shows that the detection model can work well enough to recognize the character of a license plate, however, the predicted value of a character can displayed vary because it is influenced by several factors such as the distance between the plate and the camera, the noise around the plate, and the brightness level when the camera detects a license plate.

Figure 8. Character prediction results with bounding boxex

Figure 8. Predicted license plate results with border box

TABLE 4
LICENSE PLATE RECOGNITION BASED ON YOLO12

Original Image	Result Image	Output	Status
		F1152BS	TRUE
		H1481UH	TRUE
		AD7467IC	TRUE
		B1566BMH	TRUE
		B1083TFS	TRUE
		H2872XP	TRUE
		AB1053FY	TRUE
		A7753K	FALSE

IV. CONCLUSION

The accuracy of vehicle license plate detection in Indonesia using the system is 97.5% on average, while the accuracy of character detection on license plates is 88%. These detection rates are very high for a prototype license plate recognition system if it is to be implemented in a vehicle parking system. These accuracy results confirm the effectiveness of the YOLOv12 algorithm in capturing data from vehicle license plates in Indonesia. The achieved accuracy is highly promising and has proven practical in detecting license plates. Through this research, information was obtained regarding several factors influencing the accuracy of vehicle license plate recognition in Indonesia using the YOLOv12 method.

These factors include the distance between the camera and the vehicle license plate, noise around the license plate during recognition, brightness during recognition, the position of the license plate (angle) during recognition, and others. The camera used in this study is the Acer Nitro 5 - AN515-57-921P laptop camera with video quality specifications of 720p 16:9 30fps and flicker reduction at 50 Hz. The license plate recognition system developed by the author is still a prototype that can only detect license plates in a stationary state using a laptop camera at a distance of less than 2 meters and under appropriate lighting conditions to ensure accurate detection. Therefore, for further development, this system can be enhanced to enable detection of moving vehicle license plates.

BIBLIOGRAPHY

[1] J. N. Njoku, C. I. Nwakanma, G. C. Amaizu, and D. S. Kim, "Prospects and challenges of Metaverse application in data-driven intelligent transportation systems," Jan. 01, 2023, *John Wiley and Sons Inc*. doi: 10.1049/itr2.12252.

[2] B. Ugbede Umar, O. M. Olaniyi, J. Agajo, and O. R. Isah, "Traffic Violation Detection System Using Image Processing," *Computer Engineering and Applications*, vol. 10, no. 2, 2021.

[3] H. Roßnagel, C. H. Schunck, L. Fritsch, and N. Gruschka, "Extraction and Accumulation of Identity Attributes from the Internet of Things," 2021. [Online]. Available: https://www.wigle.net

[4] U. Yousaf *et al.*, "A deep learning based approach for localization and recognition of pakistani vehicle license plates," *Sensors*, vol. 21, no. 22, Nov. 2021, doi: 10.3390/s21227696.

[5] M. M. Khan, M. U. Ilyas, I. R. Khan, S. M. Alshomrani, and S. Rahardja, "License Plate Recognition Methods Employing Neural Networks," *IEEE Access*, vol. 11, pp. 73613–73646, 2023, doi: 10.1109/ACCESS.2023.3254365.

[6] D. Satrya Utama and H. Adianto Mardijono, "Legal protection against licence plate businesses that produce licence plates for stolen vehicles," *International Journal of Social Sciences and Humanities*, vol. 2, no. 2, pp. 69–74, 2024, doi: 10.55681/ijssh.v2i2.1323.

[7] N. R. Adytia and G. P. Kusuma, "Indonesian license plate detection and identification using deep learning," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 7, pp. 1–7, Jul. 2021, doi: 10.46338/ijetae0721_01.

[8] M. C. Wijaya, "Research Of Indonesian License Plates Recognition On Moving Vehicles," *EUREKA, Physics and Engineering*, vol. 2022, no. 6, pp. 185–198, 2022, doi: 10.21303/2461-4262.2022.002424.

[9] D. Islam, T. Mahmud, and T. Chowdhury, "An efficient automated vehicle license plate recognition system under image processing," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, no. 2, pp. 1055–1062, Feb. 2023, doi: 10.11591/ijeecs.v29.i2.pp1055-1062.

[10] R. Al-batat, A. Angelopoulou, S. Premkumar, J. Hemanth, and E. Kapetanios, "An End-to-End Automated License Plate Recognition System Using YOLO Based Vehicle and License Plate Detection with Vehicle Classification," *Sensors*, vol. 22, no. 23, Dec. 2022, doi: 10.3390/s22239477.

[11] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors Latency (ms) MS COCO mAP (%)," 2025, doi: 10.0.

[12] R. Sapkota and M. Karkee, "Improved YOLOv12 with LLM-Generated Synthetic Data for Enhanced Apple Detection and Benchmarking Against YOLOv11 and YOLOv10," Feb. 2025, [Online]. Available: http://arxiv.org/abs/2503.00057

[13] R. Khanam and M. Hussain, "A Review of YOLOv12: Attention-Based Enhancements vs. Previous Versions," Apr. 2025, [Online]. Available: http://arxiv.org/abs/2504.11995

[14] K. Alomar, H. I. Aysel, and X. Cai, "Data Augmentation in Classification and Segmentation: A Survey and New Strategies," *J Imaging*, vol. 9, no. 2, Feb. 2023, doi: 10.3390/jimaging9020046.

[15] X. Hao, L. Liu, R. Yang, L. Yin, L. Zhang, and X. Li, "A Review of Data Augmentation Methods of Remote Sensing Image Target Recognition," Feb. 01, 2023, *MDPI*. doi: 10.3390/rs15030827.

[16] C. Kodithuwakku, "Vehicle Registration-Plate Detection with ML A Practical Approach," 2024.

[17] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Dec. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/make5040083.

[18] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, "Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection," Dec. 01, 2023, *Springer Science and Business Media B.V.* doi: 10.1007/s44196-023-00302-w.

[19] M. Mahasin and I. A. Dewi, "Comparison of CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0 Backbones on YOLO V4 as Object Detector," 2022, doi: 10.52088/ijesty.v1i4.291.

[20] Y. Chen, H. Xu, X. Zhang, P. Gao, Z. Xu, and X. Huang, "An object detection method for bayberry trees based on an improved YOLO algorithm," *Int J Digit Earth*, vol. 16, no. 1, pp. 781–805, 2023, doi: 10.1080/17538947.2023.2173318.

[21] D. Al-Turki *et al.*, "Human-in-the-Loop Learning With LLMs for Efficient RASE Tagging in Building Compliance Regulations," *IEEE Access*, vol. 12, pp. 185291–185306, 2024, doi: 10.1109/ACCESS.2024.3512434.

[22] B. Gouila, "Instance Segmentation for Rock Particle Quality Monitoring: Integration of Deep Learning for Machine Vision Application in the Aggregates Industry," 2023.