International Journal of Research and Review
[Indexed, Peer-Reviewed Journal]
Volume 12; Issue: 2; February 2025
Website: www.ijrrjournal.com
Research Paper *E-ISSN: 2349-9788; P-ISSN: 2454-2237*

Comparing YOLOv5 and YOLOv8 Performance in Vehicle License Plate Detection

Made Ayu Dusea Widyadara[1], Marga Asta Jaya Mulya[2]

[1]Information Technology, Faculty of Engineering and Computer Science, Nusantara PGRI University of Kediri, Indonesia,
[2]Electronics and Informatics Research Organization, Badan Riset Dan Inovasi Nasional (BRIN), Tangerang, Indonesia.

Corresponding Author: Made Ayu Dusea Widyadara

DOI: https://doi.org/10.52403/ijrr.20250202

ABSTRACT

The number of mobile vehicles on the roads in Indonesia is increasing every year. Therefore, it is essential to verify the identities of these vehicles for a variety of reasons, including locating stolen vehicles, enforcing traffic laws, managing car parks, and collecting tolls. Nevertheless, inspecting these vast numbers of vehicles manually is a challenging task. Motor vehicle number plate detection and recognition play a crucial role in intelligent transport systems. Generally, the detection and recognition of number plates on motor vehicles entail three main stages. Machine learning-based object detection, which encompasses a range of algorithms that can automatically identify and locate objects in images or videos, is the first stage. These models leverage multiple layers of processing units to extract intricate features from input data, thereby enhancing overall efficiency for object detection purposes. The YOLO algorithm is a popular object detection algorithm that can detect objects from images or videos in real-time using custom dataset. In this study, we directly compared YOLOv5 and YOLOv8 models which underwent equal training epochs, achieved stability, and utilized hyperparameters with an image size 640, 100 epochs, val 200, and batch 16. The YOLOv8 gets the best performance with almost 97.5% mAP and 69.4% mAP50-95.

Keywords: Plate Number, YOLOv5, YOLOv8, Object Detection, Custom Dataset

INTRODUCTION

Based on data from the Central Statistics Agency (BPS) on the development of the number of motorised vehicles by type in 2019-2021, the increase in the volume of mobile vehicles is increasing every year in Indonesia. Under these conditions, it is necessary to check the identity of these vehicles for various purposes, such as locating stolen vehicles, traffic law enforcement, managing car parks, and toll collection. However, checking these large numbers of moving vehicles is difficult to do manually. In the past few years, imaging technology has advanced significantly. Cameras are becoming more affordable, portable and high-quality compared to before. Also, computing power has also grown significantly.

Traffic congestion, breaches of regulations, and vehicle theft pose significant challenges to modern transportation and management systems. Several potential solutions have been suggested, such as intelligent traffic surveillance [1, 2], autonomous vehicles [3], as well as the automatic tracking and speed detection of vehicles [4]. Motor vehicle number plate detection and recognition represents a crucial aspect of intelligent transport systems. Generally, the object detection and recognition of motor vehicle number plates involve three primary phases. The initial phase is pre-processing, where in the captured image under goes processing for colour space conversion to grey scale, resizing, and noise elimination.

Object detection using machine learning models involves a collection of algorithms that can automatically identify and locate objects in images or videos. Such models leverage feature extraction, feature selection, and classification techniques to recognize objects in visual data. To train these models, labelled images are provided, where each object of interest is labelled with its corresponding class. The model then utilizes these labeled images to learn the specific features for each object class. Various machine learning models can be used to detect objects, such as support vector machines (SVM), decision trees, and random forests [5]. These models differentiate in their feature extraction and classification techniques, and their efficiency may vary depending on the specific task and data.

Deep learning models refer to a class of neural networks that can automatically identify and locate objects in images or videos. These models utilize multiple layers of processing units to extract complex features from the input data, which makes them efficient for object detection tasks. Training a deep learning model for object detection involves providing a large dataset of labeled images or videos to the model, with each object labeled by class and bounding-box coordinates. The model learns to identify and locate objects by minimizing a loss function that measures the difference between predicted and ground-truth labels and bounding boxes [6]. These models are used in applications, such as autonomous driving, surveillance, and robotics [7].

LITERATURE REVIEW

In our study, experimental investigations are presented for Faster R-CNN, SSD, and various versions of YOLO. Single Shot Multibox Detection (SSD) is a commonly utilized real-time algorithm for identifying objects in computer vision and deep learning. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg introduced it in (2016). The algorithm's strengths include its speedy object recognition and its simple and logical structure, which allows for a continuous stream of object detection [8]. Kang LLu ZMeng L et al (2023), YOLO detector based on fuzzy attention (YOLO-FA)[9]. Shi QLi CGuo B et al., (2022), the modified YOLO is characterized by large objects' sensitivity and faster detection speed, named "LF-YOLO" [10]. Gholamalinejad HKhosravi H (2021), using DWT instead of Max-pooling improved the recognition rate on the IRVD dataset [11]. Liao, Shi, Bai, Wang, & Liu (2017), enhances the text detection model TextBox by designing a coarse-to-fine detection method to minimise the effect of background noise [12]. There are comparisons that have been reviewed by previous researchers, namely YOLOv3 with SSD (Single Shot multi-box detector) [13], Faster R-CNN [14], and several other real-time deep learning algorithms. However, this research will compare between YOLOv5 and YOLOv8 to compare the detection of image objects on motor vehicle number plates.

YOLO (You Only Look Once) is a popular object detection algorithm which can detect objects in real-time from images or videos. There have been eight versions of YOLO developed so far, each with its own improvements and enhancements [15]. Figure 1 below presents the timeline documenting YOLO's development to the present day.

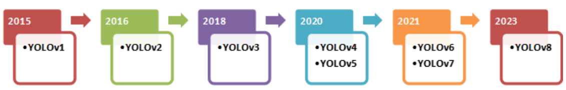


Figure 1. YOLO Timeline

YOLOv1 is the original version of YOLO, released in 2015, which can detect objects in real time but with limited accuracy and still cannot detect small objects. YOLOv2 was then released in 2016 with several improvements over the previous version, namely better accuracy, faster performance and the ability to detect smaller objects than the previous version. In 2018, YOLOv3 was released with improved accuracy and object detection speed, YOLOv3 has become one of the most widely used versions to date. YOLOv3 also introduced the concept of feature pyramids to better detect objects with different scales. In 2020, YOLOv4 was released with major improvements over YOLOv3, with better accuracy and speed. And there are new features such as scaled YOLOv4, which is able to detect smaller objects than previous versions. In the same year, YOLOv5 was released, but it was developed by a different team and used a different architecture from previous YOLO versions. The model presents a modern and adaptable architecture, diverging from its forerunners' Darknet-based methodology, while keeping true to the fundamentals of the YOLO series [16]. YOLO uses a single stage detector with a focus on speed and efficiency, but with slightly lower accuracy than YOLOv4.

YOLOv5 uses the PyTorch framework instead of the Darknet framework. On the other hand, YOLOv5 is different from previous releases. Where in YOLOv5 uses CSPDarknet53 as a backbone. This backbone solves the repetitive problem of gradient information in large backbones and integrates gradient changes into the feature map which reduces inference speed, improves accuracy, and reduces model size by reducing parameters.

It uses path aggregation network (PANet) as neck to boost the information flow. PANet adopts a new feature pyramid network (FPN) that includes several bottom ups and top down layers. This improves the propagation of low level features in the model. PANet improves the localization in lower layers, which enhances the localization accuracy of the object. The image is fed to CSPDarknet53 for feature extraction and again fed to PANet for feature fusion.

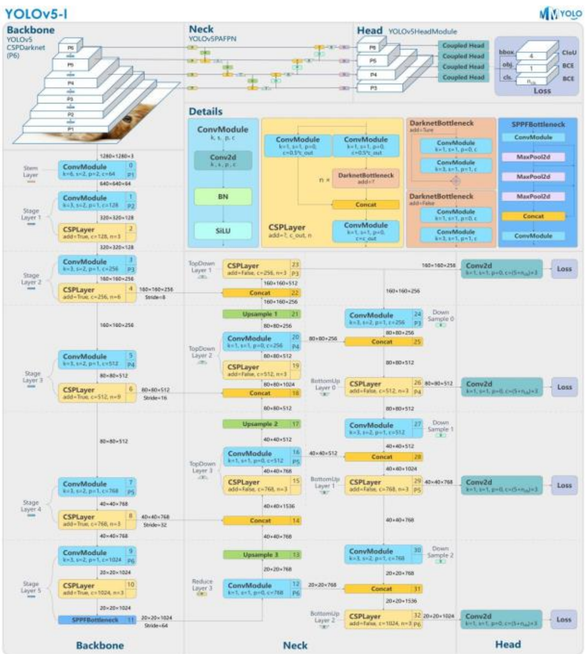


Figure 2. Structure of Yolov5 [17]

YOLO Nano is a lightweight version of YOLO designed for use in low power devices. YOLOv7 is an experimental version of YOLO that is not in widespread use, but has shown promise in improving the accuracy of the detector. There is an experimental version of YOLO that is not widely used, namely YOLOv7, but there are still improvements in object detection accuracy. YOLOv8 is an experimental version of YOLO currently under development, with no official release date announced. It is expected to further improve object recognition accuracy and speed.

YOLOv8 belongs to the family of YOLO models, using an innovative backbone architecture founded on EfficientNet, a series of convolutional neural networks that aim to achieve higher accuracy while using fewer parameters than conventional models. As a result of this, YOLOv8 operates more efficiently and quickly than certain other object detection models, all while retaining its precision.

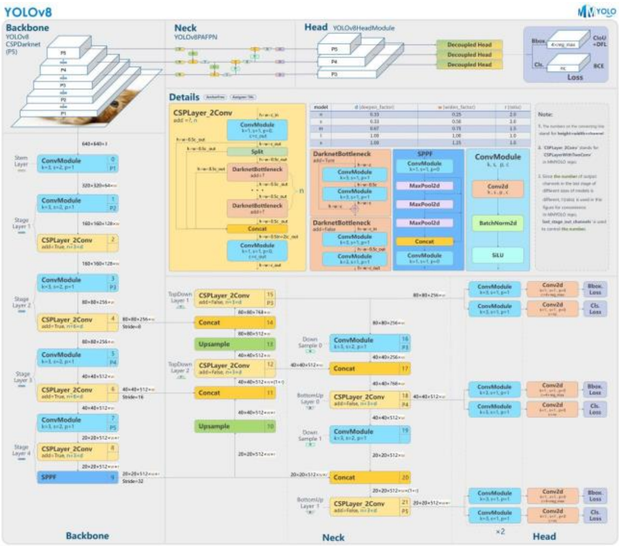


Figure 3. Structure of Yolov8 [17]

The architecture of YOLOv8 comprises of an input layer, backbone network, neck, head, and an output. The input layer of YOLOv8 receives an image as input and processes it by scaling it to a predetermined size. The backbone network is made up of a sequence of convolutional layers that extract features from the input image. YOLOv8 adopts the CSPDarknet53 backbone network which is an enhanced version of the one used in YOLOv7 - the Darknet53 network. The final predictions for object detection, comprising of the class labels, bounding box coordinates, and confidence scores, are generated by the output layer of YOLOv8. In YOLOv8, the computation of the bounding box loss is accomplished through the CIoU [18] and DFL loss functions, alongside the calculation of the classification loss via binary cross-entropy. The inclusion of these loss functions has exhibited improved object detection capabilities, particularly when handling smaller objects [19]. The image provided is first converted into an equal-length grid (S x S). Then, confidence scores are established for the "b" bounding boxes in each grid cell, as displayed in equation (1). [20]. Confidence is the probability that an object exists in each bounding box.

$$\text{Confidence (C)} = \text{P(object)} * IoU_{pred}^{target} \qquad (1)$$

The Intersection over Union (IoU) is the proportion of the intersecting area to the combined area of the predicted bounding box and the ground truth bounding box (refer to Figure 4). It quantifies the overlap between the predicted and actual bounding boxes.

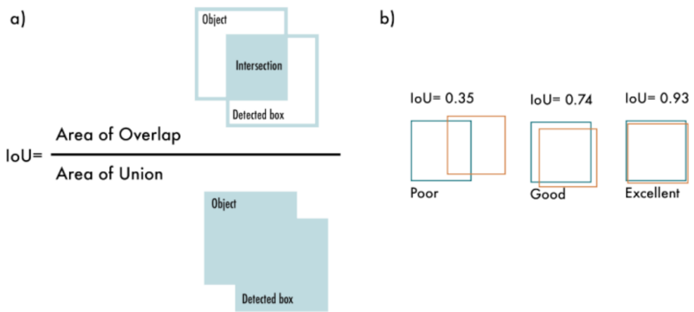


Figure 4. Intersection over Union (IoU). a) IoU is calculated by dividing the overlapping area of the two bounding boxes by their combined area; b) here are three different examples of IoU values for different locations of the boxes.[21]

YOLO predicts multiple bounding boxes per grid cell. The loss function is calculated by summing all the loss function results of the bounding box parameters, as shown in equation (2)

$$l_{box} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[(x_i \times \bar{x}_i)^2 + (y_i \times \bar{y}_i)^2 + \left(\sqrt{w_i} - \sqrt{\bar{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\bar{h}_i}\right)^2 \right]$$

$$l_{cls} = \sum_{i=0}^{S^2} l_{ij}^{obj} \sum_{C \epsilon classes} \left(p_i(c) - \bar{p}_i(c)\right)^2$$

$$l_{obj} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} (c_i - \bar{c}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{noobj} (c_i - \bar{c}_i)^2$$

(2)

YOLOv8 has the capability to be trained for various tasks related to object detection, which include but are not limited to recognizing motor vehicle number plates, tracking objects, and detecting pedestrians. Its usage is widespread in applications such as autonomous vehicles, surveillance systems, and robotics, where real-time object detection is critical. The YOLOv8 model is offered in three versions that correspond to three distinct image sizes, namely 224x224, 640x640, and 1280x1280, and are respectively named nano, small, and medium. The versions vary in the number of parameters and the duration of the training process [22].

Table 1. Comparison between structures of YOLOv5 and YOLOv8

	YOLOv5	**YOLOv8**
Neural Network Type	Fully convolution	Fully convolution
Backbone Feature Extractor	CSPDarknet53	Multiple Backbone: EfficientNet, ResNet, and CSPDarknet53
Loss Function	Binary cross entropy and Logits loss function	For the classification task, binary cross-entropy loss (BCE Loss) For the predicted box bounding regression task, distribution focal loss (DFL) and CIoU
Neck	Path Aggregation Network (PANet)	Path Aggregation Network (PANet)
Head	YOLO Layer	decoupled head structure: with separate branches for object classification and prediction bounding box regression

MATERIALS & METHODS

To test the detection performance of our proposed improved model, we use precision, recall, mAP0.5, mAP0.5:0.95, number of model parameters, model size, and detection speed as evaluation metrics. The following parameters are used in the formulae for some of the above evaluation metrics: TP (predicted as a positive sample and actually as a positive sample as well), FP (predicted as a positive sample, though it is actually a negative sample), and FN (predicted as a negative sample, though it is actually a positive sample). Intersection over Union (IoU) represents the ratio of intersection and concatenation between the bounding box and the true box [23]. Precision is the ratio of the number of positive samples predicted by the model to the number of all detected samples and is calculated as shown in Equation (3):

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

Recall is the ratio of the number of positive samples correctly predicted by the model to the number of positive samples that actually appeared. Recall is calculated as shown in Equation (4):

$$\text{Recall}=\frac{TP}{TP+FN} \quad (4)$$

The average precision (AP) is equal to the area under the precision–recall curve and is calculated as shown in Equation (5):

$$\text{AP}=\int_0^1 Precision\,(Recall)d(Recall) \quad (5)$$

Mean average precision (mAP) is the result obtained by the weighted average of AP values of all sample categories, which is used to measure the detection performance of the model in all categories, and the formula is shown in Equation (6):

$$\text{mAP}=\frac{1}{N}\sum_{i=1}^{N} AP_i \quad (6)$$

The APi in Equation (6) denotes the AP value with category index value I , and N denotes the number of categories of the samples in the training dataset (in this paper, N is 10). mAP0.5 denotes the average accuracy when the IoU of the detection model is set to 0.5, and mAP0.5:0.95 denotes the average accuracy when the IoU of the detection model is set from 0.5 to 0.95 (with values taken at intervals of 0.5). The dataset used to train the YOLOv5 and YOLOv8 models was acquired from the Roboflow repositories [24]

The process of calibrating the weights of a YOLO model to facilitate vehicle and license plate detection tasks was accomplished through training and validation. During the training phase, a specific subset of labeled images is employed to fine-tune the network's weights and minimize discrepancies between its predictions and the ground truth labels. A separate validation set assessed the network's performance on images not included in the training dataset to avoid over fitting. This measure prevents the network from memorizing the training set and promotes generalization to unseen data. Subsequently, an independent set of images is reserved for the testing pro-cess, enabling the objective evaluation of the model's overall performance and ability to detect vehicles and license plates accurately.

After the model training is complete, the resulting weights encapsulate the knowledge that the network has gained throughout the training process. These weights can be saved and employed in future training exercises using the same learning transfer methodology. This approach empowers new models to undergo faster and more accurate training by leveraging the network's previously acquired knowledge. To verify the accurate detection of license plates by the models, a thorough assessment was carried out using a varied collection of images. This assessment enabled a direct comparison between the YOLOv5 and YOLOv8 models, which had undergone an equal number of training epochs and had achieved stability in their precision and loss metrics. YOLOv5 is simpler to use, but YOLOv8 is faster and more precise. However, YOLOv8 is a better option for applications that need real-time object detection. Your selection of a model should be based on your specific application needs [25].

Figure 3, it was observed that the YOLOv8 model out performed the YOLOv5 model in detecting objects (such as vehicles and license plates) within the test images.

The dataset comprises 1360 JPG image files with bounding box annotations which depict car license plates. This research use YOLOv5 and YOLOv8 with hyperparameter configuration here is an input size of 640 x 640, 100 epochs, val 200 and batch size 16.

RESULT

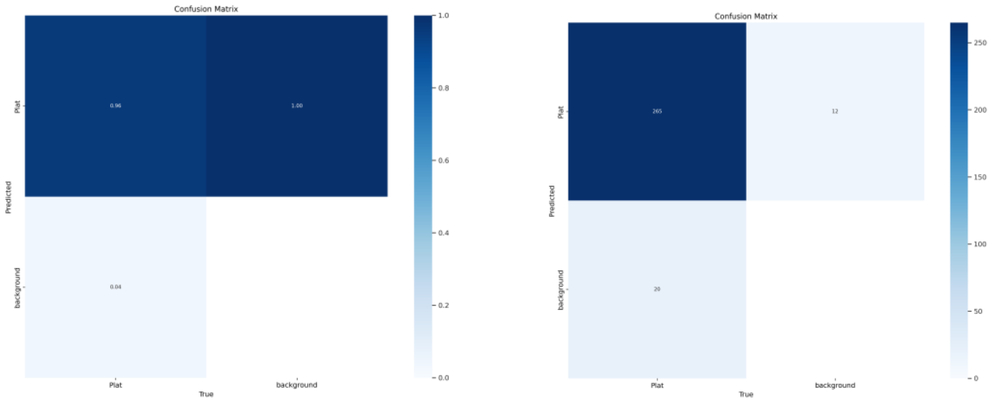
The respective performance of the YOLOv5 and YOLOv8 models regarding their classification task of identifying vehicles and license plates was evaluated based on their respective confusion matrices. The confusion matrices of the model were created while validating the system, as illustrated in Figure 5, where the 0 class is presented.

(a) YOLOv5

(b)YOLOv8

Figure 6. Demonstration of YOLO algorithm's detection tasks on a sample image validation for YOLOv5 and YOLOv8

Figure 7 shows the YOLOv5 and YOLOv8 matrix diagrams, demonstrating their improvements. These changes have contributed to a remarkable progress in mAP@0.5/%, with accurate positive readings of 0.96 in YOLOv5 and 0.98 in YOLOv8. This confirms that YOLOv8 is indeed more effective and presents a better model of precision and robustness. However, when these true positive values are rounded up, both have the same value.



(a) Yolov5 **(b) Yolov8**

Figure 7. Confusion Matrix Diagram

Based on the presented measurements in Table 2, the resulting metrics attained are accuracy, precision, recall, and F1-score. The balanced precision and recall of YOLOv5 and YOLOv8 models lead to a high F1 score.

To obtain the accuracy, precision, and recall metrics of the confusion matrix presented in Table 2, one can calculate them by applying equations (3,4) and the equation stated below.

$$\text{Accuracy} = TP / (\text{Total Dataset}) \quad (7)$$

Table 2. Results of comparing YOLOv5 and YOLOv8 algorithms for emergency landing spot detection

	YOLOv5	**YOLOv8**
Precision Recall (mAP 0.5)	0.989	0.99
Precision	0.96	0.98
Recall	0.959	0.926
F1-Score	0.95	0.95
Accuracy	48%	89%
mAP0.5:0.95	0.699	0.709
speed	1.84it/s	1.46it/s

In this experiment, the YOLOv8 gets the best performance. With nearly 99% mAP and 70.9% mAP50-95 and have better speed.

DISCUSSION

The findings from this study demonstrate the superiority of the YOLOv8 model in comparison to the YOLOv5 model with regard to vehicle license plate detection. YOLOv8 achieved higher precision and mAP metrics, particularly in mAP@0.5 and mAP@0.5:0.95, which underscores its improved detection accuracy and robustness. This enhancement can be ascribed to several pivotal advancements in YOLOv8's architecture, including its innovative CSPDarknet53 backbone, the utilisation of EfficientNet for feature extraction, and the incorporation of sophisticated loss functions such as Distribution Focal Loss (DFL) and Complete Intersection over Union (CIoU). These features empower YOLOv8 to manage smaller and more intricate objects with greater efficiency in comparison to its predecessors. Additionally, the decoupled head structure of YOLOv8, which is designed with separate branches for object classification and bounding box regression, contributes significantly to its precision and adaptability across diverse datasets. Another critical advantage of YOLOv8 lies in its capability to optimize real-time object detection tasks. The study observed that YOLOv8, while maintaining a slightly higher computational load, offers faster inference speeds and greater accuracy, making it a suitable choice for applications such as traffic management, autonomous vehicles, and surveillance systems. Furthermore, YOLOv8's upgraded dataloader mosaic feature and anchor-free detection system enhance its training and generalisation capabilities, allowing it to perform better with fewer training epochs. The aforementioned attributes suggest that YOLOv8 represents a significant evolution in object detection algorithms, aligning with recent advancements in deep learning and computer vision research.

It is imperative to acknowledge that the selection between YOLOv5 and YOLOv8 should be meticulously aligned with the specific requirements of the intended application. For tasks characterised by lower computational complexity and less stringent performance demands, YOLOv5 remains a viable option due to its straightforward implementation and reduced resource consumption. Conversely, YOLOv8 emerges as the optimal choice for applications necessitating superior accuracy, enhanced processing speed, and the capacity to adapt to complex datasets. Future research could focus on integrating YOLOv8 with optical character recognition (OCR) systems to establish a comprehensive end-to-end framework for license plate recognition, with the potential to enhance operational efficiency in vehicle identification and traffic law enforcement, thereby contributing to the development of intelligent transportation systems. Furthermore, subsequent investigations could prioritize optimizing YOLOv8 for deployment on edge devices, ensuring that the model delivers real-time performance even in environments with constrained computational resources.

CONCLUSION

The models were trained until they attained stability while utilizing the same set of images for performance evaluation. Analysis of the confusion matrix indicated that the YOLOv8 model exhibited marginally better results than YOLOv5. Furthermore, YOLOv8's training time was less than YOLOv5 for the given scenario. Using a low number of epochs, YOLOv8 may yield better accuracy, precision, recall, and F1-Score values in comparison to YOLOv5 because of upgrades to the dataloader mosaic feature and anchor-free detection system in the YOLOv8 algorithm resulting in better model performance. Future research will involve the integration of a second character recognition model for license plates, forming an end-to-end system. The detection and licence plate character recognition tasks are crucial for the system to access vehicle records.

Declaration by Authors

Acknowledgement: None
Source of Funding: None
Conflict of Interest: The authors declare no conflict of interest.

REFERENCES

1. Palaiahnakote Shivakumara. CNN-RNN based method for license plate recognition, CAAI Trans. Intellig. Technol. 2018; 3 : 169-175.
2. Marriwala, C. Tripathi. S. Jain, D. Kumar (Eds.), Natural Language processing implementation for sentiment analysis on tweets, Mobile Radio Communications and 5G Networks. Lecture Notes in Networks and Systems. Springer. Singapore. 2023; vol. 588
3. Weihong Wang, Jiaoyang Tu. Research on License Plate Recognition algorithms based on deep learning in complex environment IEEE Access. 2020; 8: 91661-91675
4. Mishra, S. Choubey, A. Choubey, N. Yogeesh, J. Durga Prasad Rao, P. William. Data extraction approach using natural language processing for sentiment analysis. International Conference on Automation. Computing and Renewable Systems (ICACRS). Pudukkottai. India 2022; vol. 2022: 970-972
5. Guleria, P., Naga Srinivasu, P., Ahmed, S., Almusallam, N., Alarfaj, F.K. XAI framework for cardiovascular disease prediction using classification techniques. Electronics. 2022; 11(24): 4086.
6. Sirisha, U., Praveen, S.P., Srinivasu, P.N. et al. Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection. Int J Comput Intell Syst 16. 2023; 126.

7. Atefeh Hemmati, Amir Masoud Rahmani. The Internet of Autonomous Things applications: A taxonomy, technologies, and future directions, Internet of Things. 2022; 20: 100635.
8. Liu W, Anguelov D, Erhan D, et al, Ssd: Single shot multibox detector. In: Computer Vision--ECCV 2016: 14th European Conference. Amsterdam. The Netherlands.2016; Proceedings Part I 14: 21–37
9. Kang LLu ZMeng L et al. YOLO-FA: Type-1 fuzzy attention based YOLO detector for vehicle detection. Expert Systems with Applications. 2024; 121209: 237
10. Shi QLi CGuo B et al. Manipulator-based autonomous inspections at road checkpoints: Application of faster YOLO for detecting large objects. Defence Technology. 2022; 18(6): 937-951
11. Gholamalinejad HKhosravi H. Vehicle classification using a real-time convolutional structure based on DWT pooling layer and SE blocks. Expert Systems with Applications. 2021; 115420: 183
12. Liao M., Shi B., Bai X., Wang X., Liu W. TextBoxes: a fast text detector with a single deep neural network Proceedings of the thirty-first AAAI conference on artificial intelligence. AAAI Press. San Francisco. California. USA. 2017; 4161-4167,
13. Viraktamath, D.S.; Navalgi, P.; Neelopant, A. Comparison of YOLOv3 and SSD Algorithms. Int. J. Eng. Res. Technol. 2021; 10: 1156–1160.
14. Benjdira, B.; Khursheed, T.; Koubaa, A.; Ammar, A.; Ouni, K. Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3. In Proceedings of the 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), Muscat, Oman, 5–7 February 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
15. Jiang P, Ergu D, Liu F et al, A review of yolo algorithm developments. Procedia Comput Sci 199. 2022;1066–1073.
16. Jocher G, Stoken A, Borovec J et al, Ultralytics/yolov5: v3. 0. Zenodo, 2020.
17. https://github.com/open-mmlab/mmyolo/tree/main/configs/, view: November, 2023.
18. Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren. Distance-iou loss: Faster and better learning for bounding box regression. in Proceedings of the AAAI conference on artificial intelligence. 2020; 34: 12993–13000.
19. Terven J, Cordova-Esparza D. A comprehensive review of YOLO: from YOLOv1 to YOLOv8 and beyond. 2023; 1–33.
20. Thuan, D. Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detection algorithm. 2021.
21. Terven, J., & Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. arXiv preprint. arXiv. 2023; 2304.00501.
22. Lou H, Duan X, Guo J et al. DC-YOLOv8: small-size object detection algorithm based on camera sensor. Electron. 2023; 12:1–14.
23. Wang, G.; Chen, Y.; An, P.; Hong, H.; Hu, J.; Huang, T. UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. Sensors. 2023; 23: 7190
24. Easy Apk, “Plat nomor indo”. (view 10 Nov, 2024). Available: https://universe.roboflow.com/easy-apk-trhqj/plat-nomor-indo/dataset/1/images
25. YOLOv8 vs. YOLOv5: Choosing the Best Object Detection Model (view Nov, 2024). Available: https://www.augmentedstartups.com/blog/yolov8-vs-yolov5-choosing-the-best-object-detection-model

How to cite this article: Made Ayu Dusea Widyadara, Marga Asta Jaya Mulya. Comparing YOLOv5 and YOLOv8 performance in vehicle license plate detection. *International Journal of Research and Review*. 2025; 12(2): 8-17. DOI: *https://doi.org/10.52403/ijrr.20250202*

******