SITEKNIK
Sistem Informasi, Teknik dan Teknologi Terapan
E-ISSN: 3032-3991 P-ISSN: 3090-1626
Vol. 2. No. 3 July 2025. Pages. 249-259

Comparison Of Efficientnet And Yolov8 Algorithms In Motor Vehicle Classification

Ferian Fauzi Abdulloh[1]✉, Favian Afrheza Fattah[2], Devi Wulandari[3], and Ali Mustopa[4]

ferian@amikom.ac.id[1], favianafrheza@students.amikom.ac.id[2], devi@amikom.ac.id3, ali.m@amikom.ac.id[4]
[12] Informatics, Computer Science, Universitas Amikom Yogyakarta, Indonesia
[34] Informatics, Computer Science, Universitas Amikom Yogyakarta, Indonesia

Keywords: Vehicles, classification, EfficientNet, YOLOv8, accuracy, precision, recall, F1-score.

Submitted: 14/07/2025
Revised: 26/07/2025
Accepted: 29/07/2025

Abstract

The YOLOv8 accuracy curve highlights clear overfitting. As shown in the graph, the model reaches 100% training accuracy from the first epoch and remains flat, indicating it memorized the training data. However, validation accuracy lags behind, fluctuating between 90% and 92% without significant improvement. This discrepancy between training and validation performance suggests that YOLOv8 struggles to generalize to unseen data. The issue likely stems from its architecture, which is optimized for object detection tasks that prioritize object localization over feature extraction for classification. When repurposed for classification, YOLOv8 may not extract the nuanced visual patterns needed to differentiate similar classes, such as trucks and buses. Consequently, although YOLOv8 performs well on the training set, its classification accuracy in real-world scenarios is limited. Addressing this may require architectural adjustments, stronger regularization, or more diverse training data to enhance the model's generalization for pure classification tasks.

Corresponding Author:
Ferian Fauzi Abdulloh
Informatics, Computer Science, Universitas Amikom Yogyakarta, Indonesia
University Address: Jl. Ring Road Utara, Condong Catur Depok sleman Yogyakarta
Email: ferian@amikom.ac.id

INTRODUCTION

Artificial Intelligence (AI) has emerged as a transformative force across a wide range of sectors, generating growing interest due to its adaptability and potential to solve complex real-world problems. From its applications in industries such as healthcare and finance to its growing role in education—including Islamic religious instruction—AI technologies continue to reshape traditional systems (Humaeroh, 2023). Among the most

dynamic areas within AI is computer vision, particularly image processing, which enables machines to interpret and analyze visual data. A prominent use case within this field is object classification, such as identifying different types of motor vehicles like cars, motorcycles, buses, and trucks. Accurate vehicle classification plays a crucial role in enhancing transportation safety, traffic regulation, and law enforcement. It enables real-time vehicle monitoring and provides valuable data for optimizing traffic systems and supporting urban infrastructure planning (Dwiyanto et al., n.d.; Ilal Mahdi, 2022).

This study focuses on evaluating and comparing the performance of two deep learning algorithms in the task of vehicle classification: EfficientNet and YOLOv8. EfficientNet is a convolutional neural network (CNN) architecture known for its scalability and classification accuracy, as demonstrated in prior studies, such as identifying poisonous mushrooms with over 84% accuracy (Muhammad Wildan Mauludy, 2024). Recent work has further validated EfficientNet's strength in various classification contexts, including environmental monitoring and object recognition (Van-Thanh Hoang, 2021). On the other hand, YOLOv8—an evolution of the widely adopted YOLO (You Only Look Once) framework—is primarily designed for object detection but has shown promise in classification tasks due to its speed and high detection accuracy (Nur et al., 2023). Its ability to process visual data efficiently in real time makes it a compelling candidate for comparison. YOLOv8 has been improved over earlier versions by reducing latency and file size while enhancing accuracy using anchor-free techniques (Yanto et al., 2023). By analyzing both models on a standardized motor vehicle dataset, this research seeks to explore their respective strengths and limitations. The ultimate goal is to generate insights that can inform the development of more robust, efficient, and scalable image classification systems, particularly within the context of intelligent transportation and surveillance technologies (Zhang et al., 2022).

RESEARCH METHODS

This study explored the implementation and performance evaluation of two prominent deep learning architectures, YOLOv8 and EfficientNet, in the task of motor vehicle classification. The dataset, which includes images of various vehicle types such as cars, motorcycles, buses, and trucks, was sourced from online repositories including Kaggle, as well as public datasets and road surveillance archives. Each image in the dataset presents a front-facing view of a vehicle, ensuring consistency in visual perspective. To facilitate effective training and evaluation, the dataset was divided into three subsets: 70% for training, 20% for validation, and 10% for testing, a common practice to ensure balanced learning and unbiased performance evaluation (Alex Krizhevsky, 2012; Jia Deng, 2009)To structure the research process in figure 1, the CRISP-DM (Cross Industry Standard Process for Data Mining) framework was adopted. This methodology consists of five main phases—Business Understanding, Data Understanding, Data Preparation, Modeling, and Evaluation—and has been widely used for data science projects since its publication by Daimler Chrysler, SPSS, and NCR in 1999 (Cahyo Prianto, 2019).

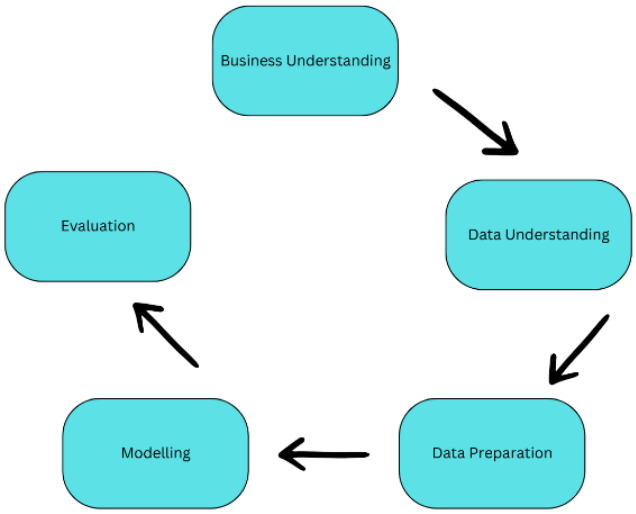

Figure 1. Title written below Image (middle)

Businness Understanding

The motivation behind this research stems from real-world urban transportation challenges. Many city governments are grappling with the issue of heavy trucks traversing restricted zones, leading to traffic congestion, accelerated road damage, and elevated safety risks. In response, there is growing interest in deploying intelligent systems that can automatically identify and classify vehicles using roadside camera feeds. The primary goal is to detect heavy trucks violating road restrictions and enable timely enforcement through automated alerts or fines. Accurate classification of trucks versus other vehicles—such as motorcycles, cars, and buses—would allow the system to focus enforcement efforts precisely on violators. Such AI-powered monitoring solutions are increasingly viewed as vital components of smart city traffic infrastructure (Buda et al., 2018; Muhammad Wildan Mauludy, 2024; Van-Thanh Hoang, 2021).

Data Understanding

The dataset used in this research comprises 4,000 labeled vehicle images, representing the four key categories: truck, motorcycle, car, and bus. The images were collected from varied sources including traffic camera footage, publicly available image datasets, and manually curated road scene photographs. The images feature diverse conditions—different lighting, backgrounds, and angles—to simulate real-world traffic scenarios. This variety is crucial for ensuring the trained model can generalize well across various operational environments (Alzubaidi et al., 2021; Buda et al., 2018; Li Liu, 2020).

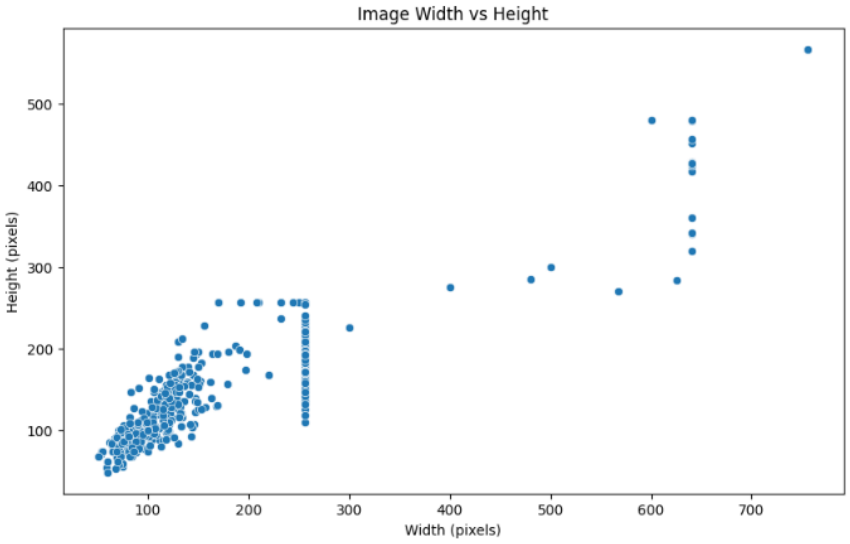

Figure 2. Image Dimensions in the Dataset

Figure 2 presented a scatter plot illustrating the distribution of image dimensions. The x-axis represents width, and the y-axis denotes height in pixels. Most images fall within a resolution range of 50–200 pixels in width and 50–300 pixels in height, reflecting that the dataset primarily contains low-resolution inputs. Interestingly, a vertical line at approximately 220 pixels wide suggests a standardized image width used across many samples, likely due to partial preprocessing. A few outliers with much larger dimensions (over 700×500 pixels) highlight inconsistencies in image resolution, emphasizing the need for uniform preprocessing prior to training (Connor Shorten, 2019; Mingxing Tan, 2019).

Figure 3. Variance Image

Data Preparation

To ensure optimal model performance, a systematic data preparation phase was conducted. Initially, all 4,000 images were reviewed to eliminate low-quality samples, such as blurry visuals, poorly lit images, or incorrectly labeled instances. After cleaning, the remaining images were resized to a fixed resolution of 224×224 pixels, a widely

accepted input size for convolutional neural networks, including EfficientNet and YOLO-based models. This resizing step ensured uniformity and compatibility with the architectures used (Muhammad Wildan Mauludy, 2024; Van-Thanh Hoang, 2021). In addition, pixel values were normalized to a range of 0 to 1, standard practice in deep learning to improve convergence during training (Chollet, 2016). Following normalization, the dataset was divided into training, validation, and testing sets using a 70:20:10 split, with an even distribution of each vehicle category across the sets. This approach helped to prevent data imbalance issues, which could otherwise skew the model's predictions (Buda et al., 2018; Muhammad Wildan Mauludy, 2024)The completed preprocessing pipeline resulted in a well-balanced and standardized dataset, establishing a strong foundation for accurate and reliable model training.

RESULTS AND DISCUSSION

This study compared two state-of-the-art deep learning algorithms—EfficientNet and YOLOv8—using a balanced dataset of motor vehicle images (cars, motorcycles, trucks, buses), captured from a consistent front-facing angle. The models were trained on identical data splits with the same data augmentation strategies, optimizer settings, and learning rates to ensure a fair and meaningful comparison.

The EfficientNetB0 model was trained for 25 epochs. The loss curves show a clear and consistent downward trend for both training and validation loss, indicating that the model successfully learned from the data while avoiding significant o(Buda et al., 2018; Muhammad Wildan Mauludy, 2024)verfitting. By the end of training, both losses approached near-zero values, suggesting excellent convergence. The train and validation loss curves shown in the figure 4 illustrate the model's learning progress over 25 epochs. Initially, the train loss starts very high, around 1.5, and decreases sharply within the first few epochs, indicating that the model rapidly learns patterns in the training data. Meanwhile, the validation loss begins at a lower value of approximately 0.2 and gradually declines alongside the train loss, reflecting improvements in the model's performance on unseen data. As training progresses into the middle epochs, both losses continue to decrease and move closer together, suggesting that the model effectively generalizes beyond the training set. By the final epochs, both curves level off near zero without significant divergence, indicating convergence and minimal signs of overfitting. Overall, the steady downward trend and the close alignment of the train and validation loss curves demonstrate that the model has successfully learned to generalize well to new data.

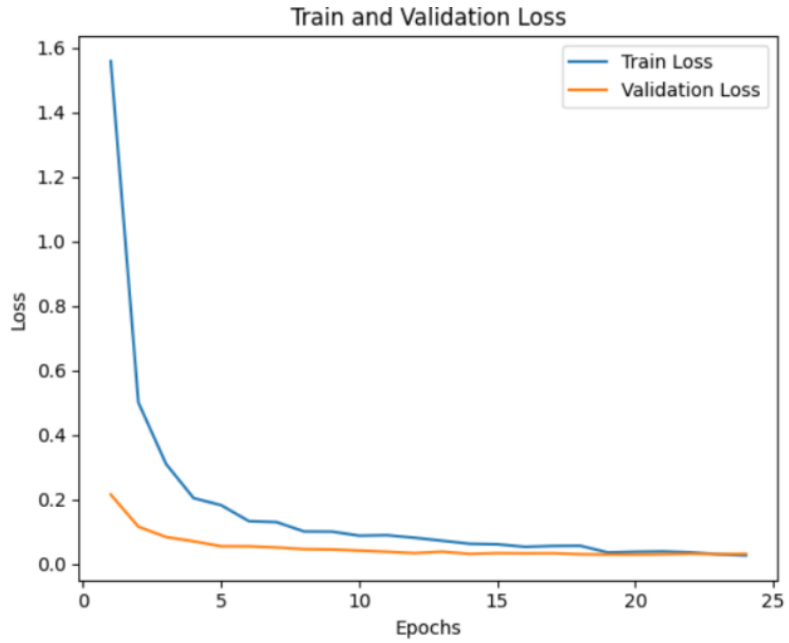

Figure 4. EfficientNetB0 Loss Curves

The accuracy curves in figure 5 reflected steady progress throughout training. The training accuracy consistently increased during the epochs, eventually surpassing 99% toward the final stages of training, indicating that the model was effectively learning from the dataset. Meanwhile, the validation accuracy also demonstrated a positive upward trend, gradually improving and stabilizing around 98% by the end of training. This steady rise in both training and validation accuracy suggests that the model achieved strong generalization performance without significant overfitting. These curves together indicate that EfficientNet was highly effective at generalizing from training data to unseen validation data.

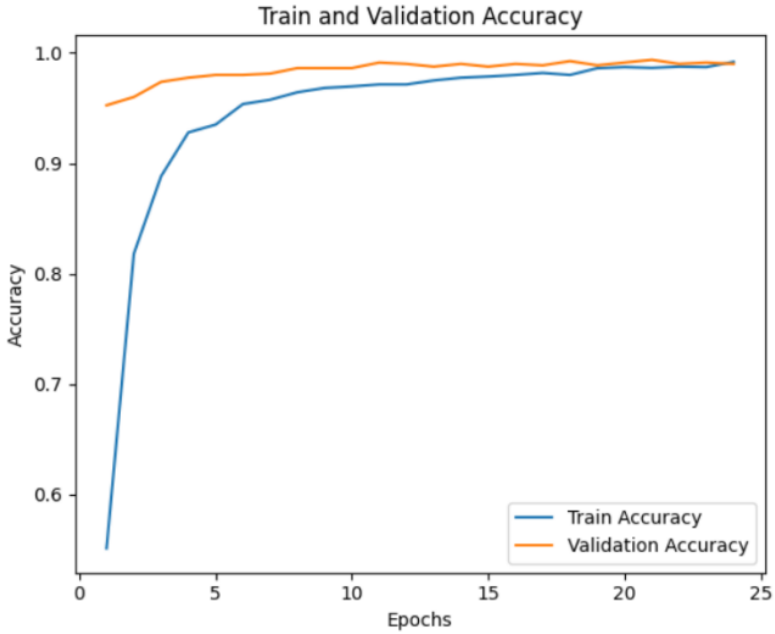

Figure 5. EfficientNetB0 Accuracy Curves

The summarizes the evaluation metrics in figure 6 for each vehicle class—Bus, Car, Motorcycle, and Truck—using precision, recall, and F1-score, along with the number of images (support) for each class. The results show that the model achieved perfect precision (1.000) across most classes, with a slightly lower precision of 0.9796 for Trucks, suggesting occasional misclassifications involving trucks. Recall values were consistently high for all classes, ranging from 0.9889 for Cars to perfect recall (1.000) for Motorcycles and Trucks, indicating the model's strong ability to correctly identify most true instances. F1-scores, which balance precision and recall, were also excellent across all categories, with the lowest being 0.9897 for Trucks and perfect scores for Motorcycles. Overall accuracy reached 99.5%, reflecting outstanding overall performance. The macro average—which gives equal weight to each class—shows precision, recall, and F1-scores all above 0.99, while the weighted average—which accounts for the number of samples in each class—produces similarly high scores. These results demonstrate that the model achieved remarkable consistency and reliability across different vehicle types, with only minor weaknesses in distinguishing trucks, indicating a strong capability for real-world vehicle classification tasks.

	precision	recall	f1-score	support
Bus	1.000000	0.989362	0.994652	94.000
Car	1.000000	0.988889	0.994413	90.000
Motorcycle	1.000000	1.000000	1.000000	120.000
Truck	0.979592	1.000000	0.989691	96.000
accuracy	0.995000	0.995000	0.995000	0.995
macro avg	0.994898	0.994563	0.994689	400.000
weighted avg	0.995102	0.995000	0.995012	400.000

Figure 6. EfficientNetB0 Evaluation Metrics

The YOLOv8 model was also trained for 25 epochs, and in contrast to the EfficientNet model, its loss curves revealed a clear divergence between training and validation performance. Throughout the training process, the training loss exhibited a significant and steady decline, rapidly approaching zero within the first few epochs. This sharp decrease indicates that YOLOv8 effectively memorized the training data, learning to predict the correct classes with near-perfect accuracy on the examples it had already seen. However, the validation loss told a different story: instead of following a similar downward trend, it remained stubbornly high, stagnating around 0.8 throughout the entire 25 epochs. This persistent gap between the decreasing training loss and the flat validation loss suggests that the model struggled to generalize its learning to new, unseen data. The plateau in validation loss indicates that while the model was exceptionally good at fitting the training set, it overfitted to those examples, capturing patterns specific to the training images rather than learning features that would generalize well. This behavior reflects a core challenge in deep learning: achieving a balance between memorizing the training data and developing the ability to perform accurately on unfamiliar inputs. The disparity between the two loss curves highlights the need for improved regularization, enhanced data augmentation, or a more diverse dataset to enable YOLOv8 to generalize effectively beyond the training data.

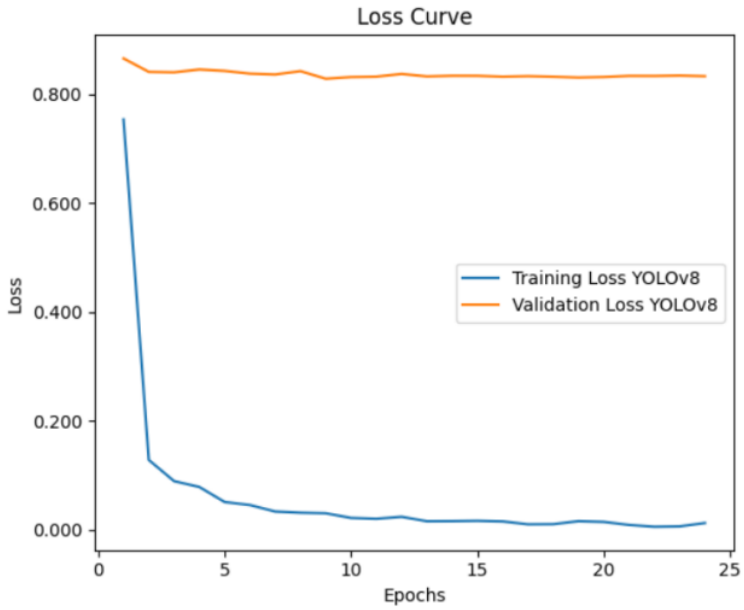

Figure 7. YOLOv8 Loss Curves

The chart in figure 7 illustrated the loss curves for YOLOv8 over 25 training epochs, where the blue line represents the training loss and the orange line represents the validation loss. The training loss starts relatively high, around 0.75, but decreases sharply within the first few epochs, approaching near zero by approximately epoch 10, which indicates that YOLOv8 is effectively memorizing the training data and achieving near-perfect accuracy on it. In contrast, the validation loss begins at an even higher value of about 0.85 and remains almost flat throughout the entire training process, showing minimal improvement. This persistent high validation loss suggests that the model struggles to generalize its learning to new, unseen data, signaling significant overfitting. The stark discrepancy between the two curves—marked by the training loss decreasing almost to zero while the validation loss stagnates—highlights YOLOv8's inability to maintain similar performance on validation images, posing a challenge for its real-world application. Overall, these trends demonstrate that although YOLOv8 excels at fitting the training set, it fails to generalize effectively, emphasizing the importance of implementing strategies such as regularization, expanding the dataset with more diverse examples, or applying advanced data augmentation techniques to improve validation performance and model robustness.

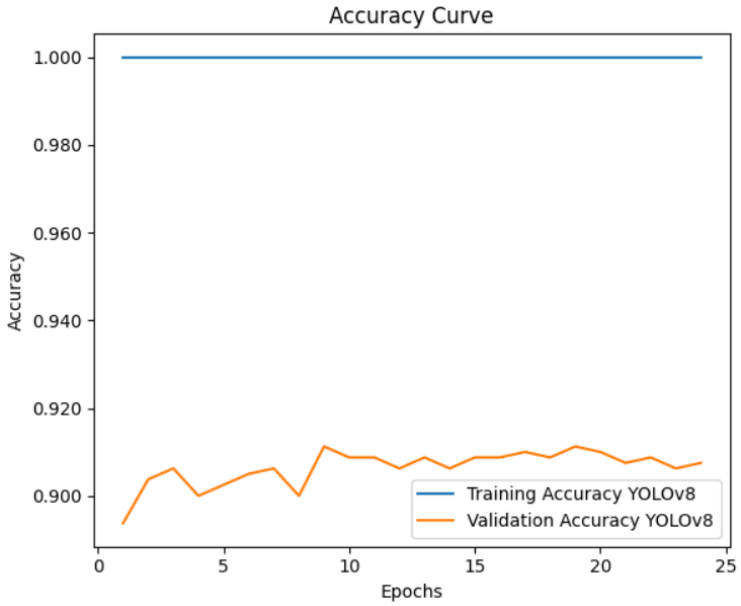

Figure 8. YOLOv8 Accuracy Curves

This chart in figure 8 presented the accuracy curves for the YOLOv8 model over 25 training epochs, where the blue line represents training accuracy and the orange line represents validation accuracy. The training accuracy immediately reaches 100% in the very first epoch and remains perfectly flat at that level throughout the entire training process, indicating that the model memorized the training data without any errors. While this perfect training performance might initially appear impressive, it is actually a clear indicator of overfitting, as it suggests the model is learning specific details from the training set rather than generalizable features. Meanwhile, the validation accuracy starts around 90% and shows slight fluctuations during training, eventually stabilizing between 91% and 92%. The relatively modest and inconsistent improvement in validation accuracy, combined with its notable gap from the perfect training accuracy, highlights the model's limited ability to generalize to unseen data. Overall, this pattern demonstrates that although YOLOv8 is highly effective at fitting the training set, it struggles to achieve comparable accuracy on new data, emphasizing the importance of implementing techniques such as regularization, data augmentation, or expanding the dataset to enhance the model's generalization capabilities.

	precision	recall	f1-score	support
Bus	0.645000	1.000000	0.784195	129.00000
Car	1.000000	0.980392	0.990099	204.00000
Motorcycle	1.000000	0.947867	0.973236	211.00000
Truck	1.000000	0.781250	0.877193	256.00000
accuracy	0.911250	0.911250	0.911250	0.91125
macro avg	0.911250	0.927377	0.906181	800.00000
weighted avg	0.942756	0.911250	0.916319	800.00000

Figure 6. YOLOv8 Evaluation Metrics

The summarizes the evaluation metrics in figure 9 presented the classification performance metrics—precision, recall, and F1-score—for each vehicle class: Bus, Car, Motorcycle, and Truck, along with the number of images (support) per class. The results show considerable variability across categories. The model achieved perfect precision (1.000) for Cars, Motorcycles, and Trucks, meaning that when it predicted these classes, it was almost always correct. However, the precision for Buses was much lower at 0.645, indicating frequent misclassifications where other vehicles were incorrectly labeled as buses. Recall values were highest for Buses (1.000), showing that nearly all true bus instances were detected, but much lower for Trucks at 0.781, suggesting the model missed many trucks. F1-scores, which reflect the balance between precision and recall, ranged widely, with the lowest at 0.784 for Buses and the highest at 0.990 for Cars. The overall accuracy for the model was 91.1%, indicating that it correctly classified most images but still struggled with certain classes—particularly Buses and Trucks. The macro average metrics, which treat each class equally, and the weighted average metrics, which account for class size, both reflected these disparities, with F1-scores of around 0.906 and 0.916, respectively. These results suggest that while the model performed well on Cars and Motorcycles, it had difficulty accurately distinguishing Buses and Trucks, pointing to areas for improvement in handling classes with more visual similarity or imbalance.

CONCLUSIONS AND SUGGESTIONS

Conclusion

This research addressed the challenge of accurately classifying different types of motor vehicles—cars, motorcycles, buses, and trucks—using image-based artificial intelligence models. The study focused on comparing two popular deep learning algorithms: EfficientNet, a model optimized for image classification, and YOLOv8, a leading object detection framework that can also perform classification tasks. To ensure fairness and reliability, both models were trained on the same dataset, which consisted of frontal images of vehicles collected from various online sources. A consistent preprocessing pipeline, identical training settings, and evaluation metrics such as accuracy, precision, recall, and F1-score were used to comprehensively assess each model's performance.

The results demonstrated that EfficientNet achieved a better balance between training and validation performance, with both losses steadily decreasing and evaluation metrics averaging above 97% across all vehicle categories. This indicates EfficientNet's strong generalization capabilities for classification tasks. Conversely, YOLOv8 showed excellent training accuracy but suffered from higher validation loss and more fluctuation in precision and recall, suggesting potential overfitting and reduced generalization to

unseen data. These findings highlight the importance of selecting a model architecture suited to the target task: EfficientNet's design aligns well with pure classification problems, while YOLOv8's strengths lie primarily in real-time object detection. In general, the research confirms that EfficientNet is a more effective solution for high-accuracy vehicle classification, providing a solid foundation for applications in traffic monitoring, automated tolling, and transportation safety systems..

Suggestion

Building on these results, future research could explore optimizing YOLOv8 specifically for classification tasks by fine-tuning its architecture or loss functions. Since YOLOv8 is inherently designed for detection, modifying its training pipeline—such as integrating specialized classification heads or experimenting with hybrid architectures—might help improve its generalization and close the performance gap with classification-focused models like EfficientNet. Additionally, exploring techniques like transfer learning with domain-specific datasets or applying advanced regularization strategies could further enhance YOLOv8's ability to classify vehicles accurately without overfitting.

Moreover, subsequent studies could expand the scope of this research by incorporating more diverse datasets, including images taken from various angles, under different lighting, or in challenging weather conditions. Introducing more vehicle categories, such as vans or bicycles, would also test the scalability and adaptability of both algorithms. Finally, evaluating these models in real-time video streams or integrating them into intelligent transportation systems could provide valuable insights into their practical deployment, ensuring robust performance in dynamic, real-world environments.

REFERENCE

Dwiyanto, R., Widodo, D. W., & Kasih, P. (2022). Implementasi Metode You Only Look Once (YOLOv5) Untuk Klasifikasi Kendaraan Pada CCTV Kabupaten Tulungagung. arXiv preprint.

Humaeroh, E. (2023). Islamic Religious Education Learning and Trends in the Use of Artificial Intelligence. Asian Journal of Islamic Studies.

Mahdi, I., Muchtar, K., Arnia, F., & Ernita, T. (2022). Substraksi Latar Menggunakan Nilai Mean Untuk Klasifikasi Kendaraan Bergerak Berbasis Deep Learning. Jurnal Rekayasa Elektrika, 18(2).

Mauludy, M. W., Rulyana, D., & Hardjianto, M. (2024). Deteksi Jamur Beracun dengan Algoritma Convolutional Neural Network dan Arsitektur EfficientNet-B0. Jurnal Media Informatika Budidarma.

Nur, M., et al. (2023). Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan YOLO V8. Conference on AI and Health Technology.

Hoang, V.-T., & Jo, H. (2021). Practical Analysis on Architecture of EfficientNet. International Journal of Advanced Computer Science and Applications.

Yanto, F. A., & Irmawati, I. (2023). YOLO-V8 Peningkatan Algoritma untuk Deteksi Pemakaian Masker Wajah. Jurnal Mahasiswa Teknik Informatika.

Zhang, Y., et al. (2022). Comparative Analysis of CNN Architectures for Vehicle Classification. Applied Sciences, 12(18).

Alzubaidi, L., et al. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1), 53.

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in CNNs. Neural Networks, 106, 249-259.

Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Deng, J., et al. (2009). ImageNet: A large-scale hierarchical image database. CVPR.

Dwiyanto, R., Widodo, D. W., & Kasih, P. (2022). Implementasi Metode You Only Look Once (YOLOv5) Untuk Klasifikasi Kendaraan Pada CCTV Kabupaten Tulungagung.

Hoang, V.-T., & Jo, H. (2021). Practical Analysis on Architecture of EfficientNet. International Journal of Advanced Computer Science and Applications.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems.

Liu, W., et al. (2020). Deep learning for generic object detection: A survey. International Journal of Computer Vision, 128(2), 261–318.

Mahdi, I., Muchtar, K., Arnia, F., & Ernita, T. (2022). Substraksi Latar Menggunakan Nilai Mean Untuk Klasifikasi Kendaraan Bergerak Berbasis Deep Learning. Jurnal Rekayasa Elektrika, 18(2).

Mauludy, M. W., Rulyana, D., & Hardjianto, M. (2024). Deteksi Jamur Beracun dengan Algoritma CNN dan Arsitektur EfficientNet-B0. Jurnal Media Informatika Budidarma.

Prianto, C., & Harani, N. S. (2019). The data mining analysis to determine the priorities of families who receiving assistance. Journal of Physics: Conference Series.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 60.

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. International Conference on Machine Learning (ICML).

Wildan, M., et al. (2023). Optimization of Vehicle Image Classification using YOLOv8 and Custom CNN. Conference on Applied AI Systems.