Implementation of YOLOv5 Algorithm for Classification and Detection of Car Types Using Deep Learning

Muhammad Rifqi Rajwa Syafirly[1], Ravi Rangga Wahyu Firmansyah[2], Ahmad Nurdiansyah[3], Agis Adienia Haqi Megana[4], I Made Wirawan[5]

Program Studi Teknik Elektro,
Fakultas Teknik,
Universitas Negeri Malang, Malang
[1]muhammad.rifqi.2105366@students.um.ac.id, [2]ravi.rangga.2105366@students.um.ac.id, [3]ahmad.nurdiansyah.21005366@students.ac.um.id, [4]agis.adienia.2105366@students.um.ac.id, [5]made.wirawan.ft@um.ac.id

Abstrak

Perkembangan teknologi *computer vision* dan *deep learning* telah membuka peluang baru dalam klasifikasi kendaraan, namun menghadapi tantangan dalam kompleksitas desain dan variasi visual mobil. Penelitian ini berfokus pada implementasi algoritma YOLO (*You Only Look Once*) V5 untuk mengklasifikasikan jenis mobil, yang bertujuan untuk mengembangkan model yang akurat, kuat, dan efektif dalam membedakan kategori mobil. *Dataset* yang digunakan dalam penelitian ini, yang bersumber dari Kaggle, terdiri atas 1.000 gambar pada empat kategori mobil (*convertible*, *double cabin*, *sport*, dan *van*), dengan 80% data digunakan untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Metodologinya mencakup prapemrosesan data, pelatihan model YOLOv5, dan evaluasi performa menggunakan metrik, seperti akurasi, presisi, perolehan, dan skor *F1*. Hasilnya menunjukkan bahwa model tersebut dapat mengklasifikasikan jenis mobil dengan *accuracy* 96%, *precision* 96,6%, *recall* 97%, dan *F1-score* 97%, sehingga menunjukkan performa yang baik untuk setiap kategori mobil.

Kata kunci: Klasifikasi, YOLOv5, *Computer vision*, Deteksi, Mobil.

Abstract

The development of computer vision and deep learning technologies has opened new opportunities in vehicle classification, but faces challenges in design complexity and visual variation of cars. This research focuses on implementing the YOLO (You Only Look Once) V5 algorithm to classify car types, aiming to develop a model that is accurate, robust, and effective in differentiating between car categories. The dataset used in this research, sourced from Kaggle, consists of 1,000 images across four car categories (convertible, double cabin, sport, and van), with 80% of the data used for training, 10% for validation, and 10% for training. The methodology includes data preprocessing, training the YOLOv5 model, and evaluating the performance using metrics such as accuracy, precision, recall, and F1-score. The findings show that the model can classify car types with 96% accuracy, 96.6% precision, 97% recall, and 97% F1-score, demonstrating reliable performance for each car category.

Keywords: Classification, YOLOv5, Computer vision, Detection, Car.

Received: 21 Februari 2025 Revised: 3 Mei 2025 Accepted: 5 Mei 2025

1. Introduction

In urban areas, the number of private vehicles, especially cars, has grown rapidly. This trend has triggered several challenges that require immediate attention. People are more inclined to use private vehicles rather than public transport, leading to increased traffic congestion and higher carbon emissions. However, this growth is not accompanied by sufficient knowledge regarding the types of cars that suit different needs. As a result, many people face difficulties in selecting the right vehicle, which can ultimately affect fuel efficiency, comfort, and safety. The need for smart solutions to help people identify and select the appropriate car type is becoming more critical, particularly with the rising demand for efficiency and sustainability in the transportation sector. Furthermore, car counting is necessary to monitor traffic volume on urban roads, making car type detection essential. Therefore, classifying car types through object detection is necessary.

Object detection is a branch of computer vision, which is a field focused on how computers can automatically analyze and interpret elements within images [1]. A key application of this technology is vehicle recognition, which serves various purposes, such as traffic management, security surveillance, and smart parking system development. The speed of detection is a crucial factor in real-time object detection. Previous research analyzed the performance of various YOLO architecture variants for vehicle detection from traffic images in Bangladesh. Findings showed that YOLOv5x outperformed YOLOv3 and YOLOv5s in mAP and accuracy [2]. Another study compared one-stage object detection algorithms, including YOLOv5, for detecting road-level objects. The experiment results indicated that YOLOv5l achieved the highest accuracy with an mAP@0.5 of 0.593, while MobileNetv2 FPN-lite had the fastest inference time [3]. Research has also explored video-based vehicle speed and license plate detection using YOLOv5-DeepSORT and HyperLPR, aiming to develop a high-accuracy speed tracking application [4]. This study takes a different approach by combining YOLOv5 with OpenCV for more effective and accurate vehicle classification. The dataset used reflects real-world conditions, such as varying lighting and viewing angles. Data collection and labeling were managed via Roboflow to enhance dataset organization and model accuracy. Additionally, Google Colab was employed for model training, ensuring efficient data processing. The focus is not only on vehicle detection but also on vehicle classification. Unlike still images, videos contain over 24 frames per second, so if the detection process is too slow, delays will occur in processing each frame, resulting in a choppy or disrupted video feed [5]. Object detection is divided into two main types, namely soft detection and hard detection. Soft detection simply identifies the presence of an object, while hard detection not only identifies the presence but also locates the object within the image. Object detection is critical for extracting information from images, enhancing data collection, supporting the labeling process, and systematically categorizing objects. This technology forms the basis for many applications, including pattern recognition, surveillance, and the development of autonomous systems [6]. Convolutional Neural Networks (CNN) are a type of neural network that is specifically designed for image data and are frequently used in image processing. CNNs share similarities with Artificial Neural Networks (ANN), particularly in the classification process, where input images are processed and categorized into different groups. However, CNNs differ from ANNs due to their architecture, which includes layers optimized for extracting features from images. CNNs are often used for object detection by creating bounding boxes around objects in images [7]. Subsequently, CNN performs classification to identify the object category within the bounding box. This method makes CNNs highly effective in various applications, including object recognition and image analysis.

In this study, YOLOv5 (You Only Look Once) is integrated with OpenCV to effectively classify vehicle types. YOLOv5 is a deep learning algorithm well-known for its fast and accurate real-time object detection capabilities. Unlike traditional detection methods, which typically use separate systems for detection and classification, YOLOv5 utilizes a single neural network to detect and classify objects simultaneously. It predicts bounding boxes for the objects and their respective class probabilities, making it highly efficient for applications requiring real-time detection [8]. This allows YOLO to process images much faster and with greater efficiency. The real-time detection capability of YOLOv5 makes it highly suitable for applications such as traffic monitoring and vehicle classification. As a result, YOLO is highly effective in applications that require quick and accurate detection, such as vehicle classification and surveillance [9]. The dataset used in this research is obtained from the Kaggle platform, consisting of images with varying lighting conditions and angles, thereby reflecting real-world situations. These variations ensure that the model can perform well under different environmental factors. Techniques like image rotation, scaling, flipping, and color adjustments were used to expand the training dataset artificially, which helps the model generalize better across different scenarios and avoid overfitting. By leveraging YOLOv5's advanced detection capabilities along with OpenCV's image processing tools, the study presents an effective approach for classifying vehicle types with high accuracy and speed in diverse conditions.

2. Research Method

The objective of this study is to assess the effectiveness of the YOLOv5 method in classifying various car types. The methodology applied in this study involves several steps, which are shown in Figure 1.

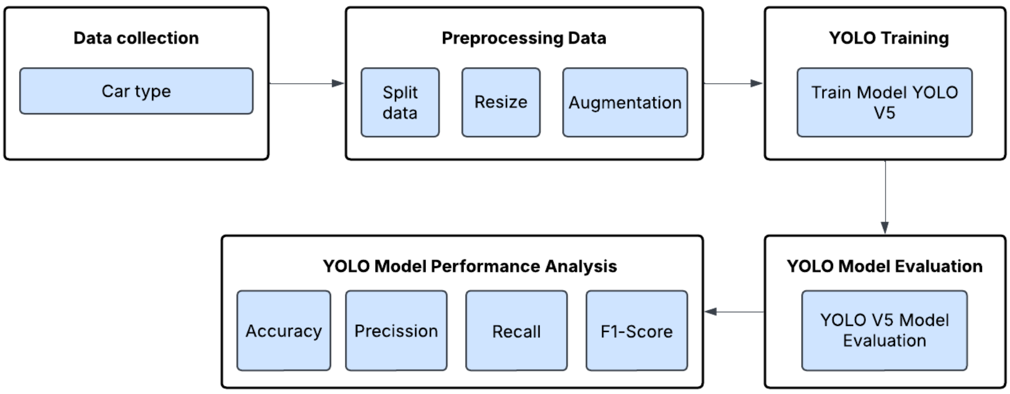

Figure 1. Research stages

2.1. Data Collection

This step involves gathering the necessary data to support the execution of this research. The dataset used in this study was obtained from the Kaggle platform and contains various types of cars [10]. These images form the basis of both the training and evaluation processes of the model. The dataset is divided into four primary categories: convertible cars, double cabins, sports cars, and vans. The images vary in terms of angles, lighting conditions, and perspectives, which ensures that the model is trained on realistic and diverse scenarios.

In this study, the YOLO (You Only Look Once) algorithm is applied to effectively classify car types. YOLO is a well-known and highly effective deep learning algorithm that is capable of performing both object detection and classification simultaneously in real-time. The images in the Kaggle dataset are pre-processed in RoboFlow, a platform used for preparing data and training the model. The images are standardized to a uniform size to maintain consistency, and a total of 1,000 images are included, with each car type represented by 250 images.

The training data is used to build and train the model, teaching it to identify the distinctive features of each car type. On the other hand, the validation data is used to assess the performance of the model and determine its classification accuracy. YOLO's ability to detect and classify car types quickly and accurately is the key focus of this study. During the validation process, the model's ability to generalize to unseen data is evaluated to ensure that it is not overfitting, and the model is capable of making accurate predictions on new, real-world images.

1. Double Cabin

   A sample image of the double cabin car type used in the training process can be seen in Figure 2.

Figure 2. Sample Image of a Double Cabin Car [11]

2. Van

   Sample image of the van car type used in the training process can be seen in Figure 3.

Figure 3. Sample Image of a Van Car [11]

3. Sport

   Sample image of the sport car type used in the training process can be seen in Figure 4.

Figure 4. Sample Image of a Sport Car [11]

4. Convertible
   Sample image of the convertible car type used in the training process can be seen in Figure 5.

Figure 5. Sample Image of a Convertible Car [11]

2.2. Data Preprocessing

This step involves data preprocessing before it can be utilized in the training and evaluation phases of YOLOv5. Image labeling is the first task in dataset processing, where each image is tagged with a label that holds information about the image. This labeling step is essential for organizing the dataset and facilitating the classification process in the following stages [12]. The first action in preprocessing is splitting the data into three sets: training (for model learning), validation (for model evaluation), and training (for the final assessment). This data split is important to prevent overfitting and ensure that the model generalizes well. The dataset is divided as follows: 80% of the data is used for training, 10% for validation, and the remaining 10% for training. Following the data split, the next step is to resize the images to a standard size of 640x640 pixels, ensuring that all images are of equal dimensions. This step is necessary to speed up the training process and ensure that all images are processed in the same way, minimizing the risk of errors caused by variations in image size [13]. After resizing, data augmentation is performed to increase the size of the training dataset. Data augmentation techniques such as flipping, rotating, and changing the brightness or contrast of the images help generate more diverse training samples. Techniques such as rotating, flipping, and adjusting the brightness or contrast of images generate more training data, thereby enhancing the model's ability to generalize and improving its accuracy.

2.3. YOLOv5 Model Training

YOLOv5 is an improved version of the previous YOLO series, specifically an upgrade from YOLOv4. One of the most impressive features of YOLOv5 is its enhanced execution speed, which exceeds YOLOv4 by 10 FPS. This faster performance allows YOLOv5 to be used in real-time applications where quick decision-making is crucial. Alongside the performance boost, YOLOv5 has a file size that is 90% smaller than YOLOv4, which enables its deployment on devices with lower computational capacity, such as embedded systems or mobile devices. Another significant improvement is its accuracy, as YOLOv5 shows better performance in object detection, particularly for smaller objects, when compared to its predecessor YOLOv4. These enhancements make YOLOv5 an optimal choice for various applications, from industrial to mobile systems, requiring a balance of speed, accuracy, and computational efficiency [14].

YOLOv5 is the first version of the YOLO algorithm written in PyTorch, which represents a significant shift from previous YOLO versions that utilized Darknet by Redmon. Previous versions of YOLO relied on Darknet by Redmon, which is quite different from the PyTorch framework used in YOLOv5. PyTorch's architecture is smaller and more efficient, enabling faster and lighter execution of the YOLO algorithm. PyTorch's more compact and efficient architecture allows YOLOv5 to be implemented faster and with less computational overhead. This results in faster execution times, with YOLOv5 displaying a minimal GPU delay during computation. The combination of these factors makes YOLOv5 highly efficient, delivering fast and accurate object detection in images, making it an ideal choice for real-time applications [14].

The YOLO (You Only Look Once) algorithm works by dividing an image into several sections, known as grids, which are typically divided into nine or more smaller grids. Typically, the image is split into nine or more grids, and each grid undergoes a convolution process. The convolution process helps to extract important features from the image, allowing the model to predict the type and location of objects in each region. The convolution step, which is crucial for detecting these objects, is demonstrated in Figure 6 [15].

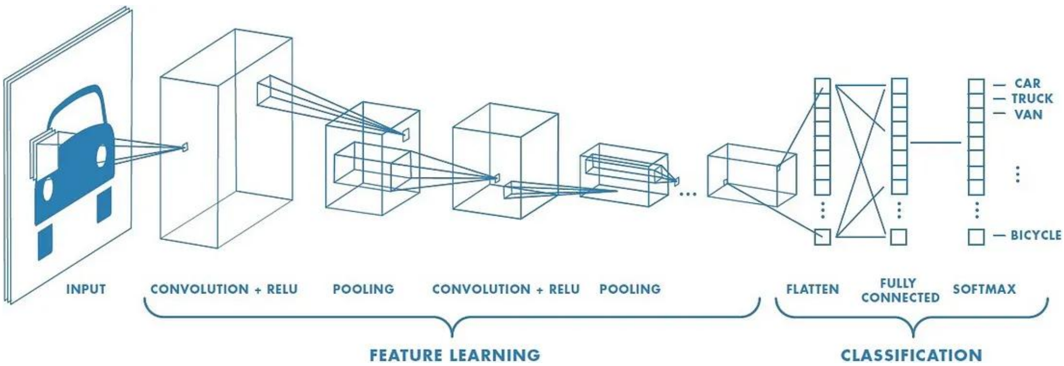

Figure 6. Image Convolution Process

Once the grids are generated, each grid undergoes matrix calculations. Every grid contains its own matrix, which is then filled with data about the objects detected. The process continues with filling the matrix in each grid, where the number of objects present within each grid is checked [16][17]. The matrix is then populated with prediction values reflecting the objects with the highest probability of being present in each grid. These predictions are based on distinct features that help identify the objects being detected. After bounding boxes are established in each grid, the next step is to eliminate low-probability predictions and merge them into a single bounding box. The final step uses Intersection Over Union (IoU) and Non-Maximum Suppression (NMS) technologies to refine and finalize the predictions, eliminating duplicate bounding boxes and ensuring accurate object detection. This process allows YOLO to effectively detect objects in images [16].

2.4. YOLO Model Evaluation

Evaluation parameters are used to measure the accuracy of a model and its level of confidence when classifying new data [18]. This evaluation step is designed to gauge how well the model can identify patterns in data it has not encountered during training. This step is an important part of the model evaluation process, as it assesses the model's quality in classifying vehicle types using data that differs from the training dataset. During this stage, a set of test data is used to evaluate the model's performance. Key evaluation metrics, such as precision, recall, and mean average precision (mAP), will be calculated and used to assess the accuracy of the model. These metrics provide important insight into the model's performance and its ability to make accurate predictions on unseen data.

2.5. YOLO Model Performance Analysis

This step involves the calculation phase to determine the results and accuracy values of the YOLO algorithm in classifying different types of vehicles, such as convertible, double cabin, sport, and van. Key performance metrics are computed, such as accuracy, precision, recall, and F1-score. These values are derived from a confusion matrix, which helps quantify the model's performance by comparing predicted classifications to actual classifications. By evaluating these metrics, we can assess how well the model performs in differentiating between vehicle categories, thus providing a clear measure of the algorithm's classification accuracy.

A confusion matrix is a summary of the results of predictions in a classification problem. It counts both the correct and incorrect predictions, organizing them by class. This allows for a deeper understanding of the types of errors made by the model. In the confusion matrix, there are several important parameters [18]:

1. TP (True Positive), which shows the number of positive data points correctly classified by the system.
2. TN (True Negative), which shows the number of negative data points correctly classified by the system.
3. FN (False Negative), which shows the number of negative data points incorrectly classified as positive by the system.
4. FP (False Positive), which shows the number of positive data points incorrectly classified as negative by the system.

As a metric, mAP (mean Average Precision) is an essential indicator for evaluating object detection models, as it not only assesses the success of the model in locating objects but also ensures that the predictions made are accurate and relevant. A high mAP score indicates that the model performs consistently well in detecting objects across various categories, which is why mAP is frequently used in computer vision research and the development of object detection systems. In addition to mAP, other evaluation metrics such as accuracy, precision, recall, and F1-Score (F-Measure) are calculated to provide a comprehensive assessment of the model's performance. Accuracy, for instance, can be calculated by comparing the number of correctly classified instances with the total number of instances, which provides a clear indication of how well the model classifies data overall:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision is a key evaluation metric that measures the accuracy of the positive classifications made by the model. In other words, precision measures how accurate the positive predictions are. Precision helps to understand how reliable the positive predictions are. The formula for calculating precision is:

$$Precision = \frac{TP}{TP + FN} \times 100\% \quad (2)$$

Recall is a metric that shows the percentage of positive category data that has been correctly classified by the system. It measures the model's ability to detect all relevant positive cases in the dataset. The equation to calculate recall is as follows:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

The F1-score is calculated using the harmonic mean of precision and recall values to provide a combined measure of both metrics. This metric is valuable in situations where there is a trade-off between precision and recall, and it seeks to balance the two. The formula to calculate the F1-score is as follows:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

3. Results and Discussion

The training was conducted by inputting images outside the training data to evaluate the performance of the trained model.

Figure 7. Test Results

Figure 7 shows the results of training the vehicle classification system, displaying the vehicle types and their respective accuracy percentages. In this experiment, the convertible car had an accuracy rate of 87.0% (a), the double cabin car had an accuracy rate of 84.0% (b), the sports car had an accuracy rate of 91.0% (c), and the van had an accuracy rate of 87.0% (d). These outcomes meet expectations, and the program is functioning as anticipated.

The performance evaluation of the model, which has gone through the training procedure, will be conducted based on the confusion matrix generated during the training process and measured using accuracy, precision, recall, and F1-score parameters to assess how well the developed model performs. Accuracy calculates the percentage of total test data correctly classified by the system. Precision calculates the percentage of correct positive predictions compared to all positive predictions made by the system. Recall evaluates the percentage of data correctly classified by the system. Precision calculates the percentage of correct positive predictions compared to all positive predictions made by the system. Recall evaluates the percentage of data correctly classified as members of a specific class compared to the total test data from that class. The F1-score is the harmonic mean of precision and recall. If the YOLOv5 model shows high accuracy, along with good precision and recall, and a high F1-score, it can be concluded that the model performs well. The performance of the YOLOv5 model is displayed in the confusion matrix in Figure 8.

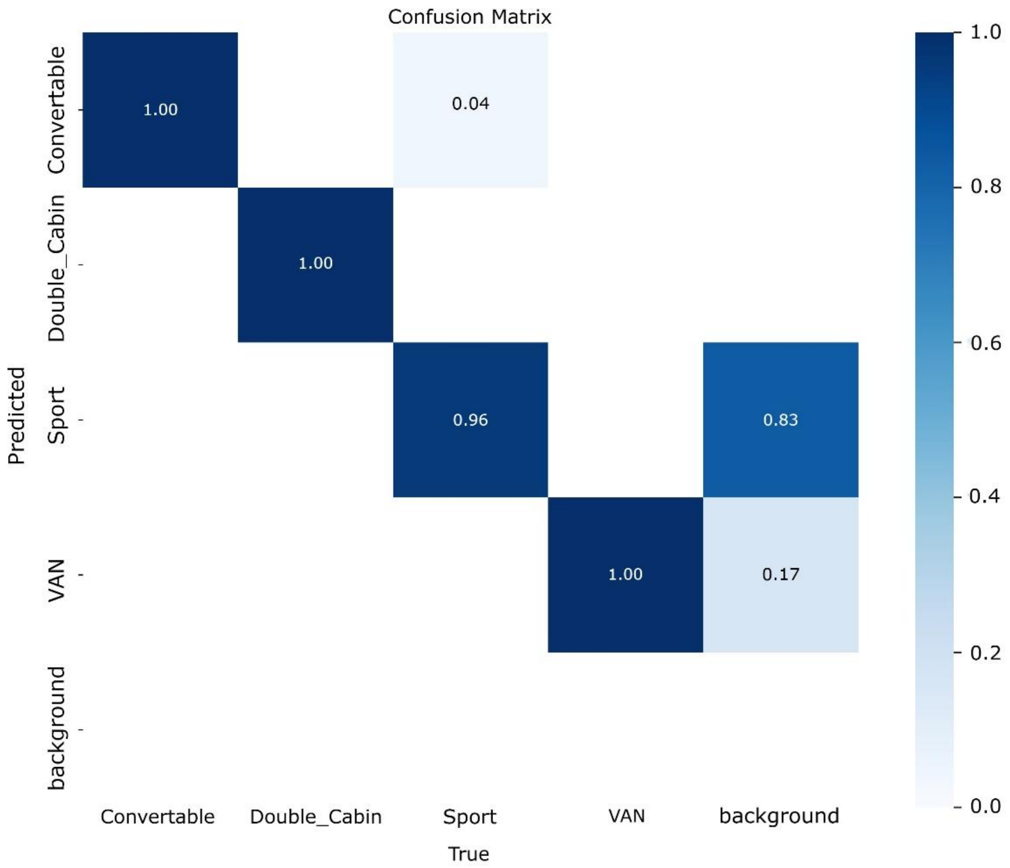

Figure 8. YOLO Model Confusion Matrix

From the confusion matrix model, the accuracy value is obtained using the existing formula.

$$Accuracy = \frac{(3{,}96) + (1{,}00)}{(3{,}96) + (0{,}04) + (1{,}00)} = \frac{4{,}96}{5} = 99{,}2$$

Using the formula above, an accuracy of 99.2% was obtained. From the values in the confusion matrix, the analysis can be based on the accuracy, precision, recall, and F1-score, as shown in Table 1 and Table 2.

Table 1. YOLOv5 All Class Model Performance

Model Performance (%)	All Class
Accuracy	99,2
Precision	95,1
Recall	98,9
F1-score	96,9

Table 2. YOLOv5 Model Performance for Each Category

Category	Performa Model (%)		
	Precision	Recall	F1-Score
Convertible	96,1	99,5	97,7
Double cabin	99,9	100,0	99.9
Sport	89,0	96.0	92,3
Van	95.3	100.0	97,5

From the two performance tables above, it can be seen that the developed model has an average accuracy of 99.2%, an average precision of 95.1%, an average recall of 98.9%, and an average F1-score of 96.9%. Each category has similar precision, recall, and F1-score values. For the Convertible category, the precision is 96.1%, recall is 99.5%, and F1-score is 97.7%. For the Double Cabin category, the precision is 99.9%, recall is 100.0%, and F1-score is 99.9%. For the Sport category, the precision is 89.0%, recall is 96.0%, and F1-score is 92.3%. For the Van category, the precision is 95.3%, recall is 100.0%, and F1-score is 97.5%. From the model performance above, it can be concluded that the results are quite satisfactory and the model is ready to be implemented in a vehicle type classification system.

4. Conclusion

Based on the research titled "Implementation of the YOLOv5 Algorithm for Vehicle Type Classification and Detection Using Deep Learning," the developed model demonstrates impressive results with an accuracy of 96.0%, precision of 96.6%, recall of 97.0%, and an F1-score of 97.0%. From these results, it can strengthen the findings of Dwiyanto et al. who explained that YOLOv5 is effective in detecting and classifying vehicle types from CCTV in Tulungagung district with high accuracy [1]. These metrics indicate that the model performs excellently in detecting and classifying various vehicle types, providing reliable and consistent results across different categories. Specifically, for the Convertible category, the precision is 94.9%, recall is 96.0%, and F1-score is 95.0%. For the Double Cabin category, the precision is 100.0%, recall is 96.1%, and F1-score is 99.0%. In the Sport category, the precision is 96.3%, recall is 96.0%, and F1-score is 96.0%. Finally, for the Van category, the precision is 95.4%, recall is 100.0%, and F1-score is 98.0%. This result also reinforces the findings of Naftali et al. who compared object detection algorithms and stated that YOLOv5 is quite superior in identifying highway vehicle objects efficiently [3]. These findings suggest that the model is highly effective in classifying vehicle types with a strong overall performance across all categories. The high performance of the model makes it potential to be implemented in vehicle classification systems, offering a reliable solution for real-time car counting and other related applications.

To further increase the scope and applicability of the research could be to explore the use of newer algorithms such as YOLOv10 or YOLOv11, which show significant improvements in accuracy, computational efficiency and object detection capabilities. Extending the application of this model to real-time video-based vehicle classification systems also offers significant potential, especially when combined with object tracking algorithms to facilitate dynamic traffic monitoring. In addition, incorporating datasets that reflect local vehicle conditions, including angkot, buses and trucks prevalent in Indonesia, will improve the contextual accuracy and relevance of the model. In line with ongoing technological advancements, the integration of more adaptive and intelligent Artificial

Intelligence (AI) approaches is equally important to enable the system to respond more effectively to the complexities of real-world environments.

References

[1] R. Dwiyanto, D. W. Widodo, and P. Kasih, “Implementasi Metode You Only Look Once (YOLOv5) Untuk Klasifikasi Kendaraan Pada CCTV Kabupaten Tulungagung,” *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, vol. 6, no. 3, pp. 102–104, Nov. 2022, doi: https://doi.org/10.29407/inotek.v6i3.2669.

[2] R. M. Alamgir, A. A. Shuvro, M. Al Mushabbir, M. A. Raiyan, N. J. Rani, M. M. Rahman, et al., “Performance analysis of YOLO-based architectures for vehicle detection from traffic images in Bangladesh,” *in Proceedings of the 2022 25th International Conference on Computer and Information Technology (ICCIT)*, pp. 982–987, Dec. 2022, doi: https://doi.org/10.1109/ICCIT57492.2022.10055758.

[3] M. G. Naftali, J. S. Sulistyawan, and K. Julian, “Comparison of object detection algorithms for street-level objects,” *arXiv preprint* arXiv:2208.11315, Aug. 2022, doi: https://doi.org/10.48550/arXiv.2208.11315.

[4] H. Henny, M. A. Baiquni, B. Mulyanti, M. F. Nasution, and A. H. S. Budi, “Aplikasi Penghitung Kecepatan Mobil dengan Akurasi Tinggi Menggunakan YOLO untuk Meminimasi Kecelakaan,” *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, vol. 11, no. 2, pp. 140–149, 2023, doi: https://doi.org/10.34010/telekontran.v11i2.10900.

[5] A. Nugraha, A. Walidani, D. Arochman, M. N. Fahrezi, S. A. H. Agat, and P. Rosyani, “Systematic Literatur Review Mendeteksi Kendaraan Menggunakan Metode YOLO (You Only Look Once),” *JRIIN: Jurnal Riset Informatika dan Inovasi*, vol. 1, no. 3, pp. 559–562, 2023, doi: https://doi.org/10.24002/ijis.v1i2.1916.

[6] N. Thakur, P. Nagrath, R. Jain, D. Saini, N. Sharma, and J. Hemanth, “Object detection in deep surveillance,” 2021, doi: https://doi.org/10.21203/rs.3.rs-901583/v1.

[7] K. Khairunnas, E. M. Yuniarno, and A. Zaini, “Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot,” *Jurnal Teknik ITS*, vol. 10, no. 1, pp. A50–A55, 2021, doi: http://dx.doi.org/10.12962/j23373539.v10i1.61622.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 779–788, 2016, doi: https://doi.org/10.1109/CVPR.2016.91.

[9] M. Y. Efendi and M. H. F. Abidin, “Implementasi Klasifikasi Jenis Kendaraan di Indonesia Menggunakan YOLO,” *in Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika*, pp. 1–7, Apr. 2024, doi: https://doi.org/10.31284/p.snestik.2024.5788.

[10] D. I. Mulyana and M. Zikri, “Optimasi Mendeteksi Klasifikasi Citra Digital Logo Mobil Indonesia dengan Metode Single Shot Multibox Detector,” *Jurnal Sistem Informasi dan Telematika (Telekomunikasi, Multimedia dan Informatika)*, vol. 13, no. 2, pp. 88–94, Dec. 2022, doi: http://dx.doi.org/10.36448/jsit.v13i2.2660.

[11] Da_Vinci_Code, "kaggle," [Online]. Available: https://www.kaggle.com/datasets/ademboukhris/cars-body-type-cropped. [Accessed 15 September 2024].

[12]L. Lusiana, A. Wibowo, and T. K. Dewi, "Implementasi Algoritma Deep Learning You Only Look Once (YOLOv5) Untuk Deteksi Buah Segar Dan Busuk," *Paspalum: Jurnal Ilmiah Pertanian*, vol. 11, no. 1, pp. 123–130, 2023, doi: https://doi.org/10.35138/paspalum.v11i1.489.

[13]S. Saponara and A. Elhanashi, "Impact of image resizing on deep learning detectors for training time and model performance," *in International Conference on Applications in Electronics Pervading Industry, Environment and Society, Cham: Springer International Publishing*, pp. 10–17, Sept. 2021, doi: http://dx.doi.org/10.1007/978-3-030-95498-7_2.

[14]L. Rahma, H. Syaputra, A. H. Mirza, and S. D. Purnamasari, "Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once)," *Jurnal Nasional Ilmu Komputer*, vol. 2, no. 3, pp. 213–232, 2021, doi: https://doi.org/10.47747/jurnalnik.v2i3.534.

[15]O. E. Karlina and D. Indarti, "Pengenalan objek makanan cepat saji pada video dan real time webcam menggunakan metode You Only Look Once (YOLO)," *Jurnal Ilmiah Informatika Komputer*, vol. 24, no. 3, pp. 199–208, 2020, doi: http://dx.doi.org/10.35760/ik.2019.v24i3.2362.

[16] J. Redmon, "YOLOv3: An incremental improvement," *arXiv preprint* arXiv:1804.02767, 2018, doi: https://doi.org/10.48550/arXiv.1804.02767.

[17]R. A. Asmara, M. R. S. Anugrah, D. W. Wibowo, K. Arai, M. A. Burhanuddin, A. N. Handayani, and F. A. Damayanti, "YOLO-based object detection performance evaluation for automatic target aimbot in first-person shooter games," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 4, pp. 2456–2470, 2024, doi: https://doi.org/10.11591/eei.v13i4.6895.

[18]J. Vaicenavicius, D. Widmann, C. Andersson, F. Lindsten, J. Roll, and T. Schön, "Evaluating model calibration in classification," *in Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, vol. 89, pp. 3459–3467, Apr. 2019, doi: https://doi.org/10.48550/arXiv.1902.06977.