E-ISSN: 2962-6110
P-ISSN: 2964-7843
JSRET
jsret.knpub.com
Journal of Scientific Research, Education, and Technology

Traffic Sign Recognition System Using YOLOv8 Algorithm

Yoga Dwi Rizki Fauzi, Rodhiyah Mardhiyyah
Yogyakarta University of Technology, Yogyakarta, Indonesia

ABSTRACT

The advancement of artificial intelligence has strengthened intelligent transportation systems, particularly autonomous vehicles that depend on high-precision real-time traffic sign detection. This research develops an enhanced web-based traffic sign recognition system using the YOLOv8 algorithm, with a focus on model optimization and practical deployment. A dataset of 5,224 annotated images across 20 traffic sign classes was preprocessed using resizing and augmentation techniques to improve generalization. Model training on Google Colab using the optimal configuration (50 epochs, batch size 32, learning rate 0.001) achieved strong detection performance with a Precision of 0.9406, Recall of 0.9395, and mAP50 of 0.9748. The trained model was further integrated into an interactive Flask-based web application supporting image, video, and real-time camera input. The findings highlight YOLOv8's capability to provide fast and accurate detection suitable for intelligent transportation environments, demonstrating a practical contribution for future deployment in autonomous driving and road-safety monitoring systems.

Keywords: *Object Detection, YOLOv8, Traffic Sign Recognition, Intelligent Transportation System*

Corresponding author
Name: Yoga Dwi Rizki Fauzi
Email: yoga.dwirf32@gmail.com

INTRODUCTION

Traffic violations remain one of the primary causes of the high number of road accidents. Traffic accidents occur throughout the year, with most incidents resulting from drivers' non-compliance with traffic signs and regulations (Mita Nalsalisa Br Barus et al., 2024). These violations include actions such as exceeding speed limits, running red lights, and disregarding warning signs. In addition to driver behavior, limited ability to recognize or respond to information conveyed by traffic signs also contributes significantly to the occurrence of incidents on the road (Fadilla Putri et al., 2024). With the increasing number of vehicles worldwide and the growing complexity of modern transportation networks, the need for systems capable of assisting in the automatic recognition and interpretation of traffic signs has become increasingly important.

Along with the advancement of artificial intelligence technology, various innovations have emerged in the field of transportation, including autonomous vehicles. One of the key technologies employed in autonomous vehicles is object detection. Object

Detection is a major domain within computer vision that focuses on the automatic identification and localization of specific objects within an image or video (Tessa Ningrum et al., 2024). This capability makes object detection an essential component in various modern applications, such as traffic surveillance systems, security systems, facial recognition, and autonomous vehicles (Kresnanto & Bahy, 2023).

YOLO, an abbreviation for "You Only Look Once," is an algorithm that enables real-time object detection. YOLO is designed to predict bounding boxes by dividing an image into small grids and computing class probabilities directly in a single processing stage, making it highly fast and efficient (Ikbal & Saputra, 2024). YOLO is known for its high processing speed without compromising accuracy, making it ideal for implementation in autonomous vehicles that require rapid response times (Taufiqurrahman et al., 2024). YOLO is highly suitable for implementation in real-time systems such as autonomous vehicles.

Autonomous vehicle technology integrates various sensors, such as cameras, radar, lidar, and GPS, to observe environmental conditions in real time (Ayala & Mohd, 2021). The information collected by these sensors is then analyzed by artificial intelligence (AI) algorithms to enable the vehicle to make decisions autonomously. The data obtained from various sensors, particularly cameras, is processed by the software to determine system responses such as accelerating, braking, turning, or stopping (Liyanthy, 2024).

Traffic signs are essential components of the transportation system that function as instruments for conveying information, warnings, and prohibitions to road users in order to support smooth and safe driving. These signs may take the form of symbols, numbers, letters, or a combination of these elements, and are adapted to the traffic regulations applicable in each respective country (Akbar, 2022). Therefore, the ability of autonomous vehicles to automatically recognize traffic signs is crucial, as it enables them to comply with traffic regulations and adjust driving behavior according to the surrounding environment.

However, despite the advances in the development of autonomous vehicle systems, their implementation in real-world environments still faces various challenges (Arya Satya Pratama, 2024). Several challenges encountered in the automatic recognition of traffic signs include variations in shape and color across different countries or regions, visual disturbances such as shadows, low illumination, and extreme weather conditions that may affect detection accuracy, as well as computational efficiency in processing images (Wang et al., 2024).

Several previous studies have been conducted on the development of traffic sign detection systems using various versions of the YOLO algorithm. Dedy Kusuma & Mauizah (2023), in their research, developed a traffic light detection system for autonomous vehicles using the YOLO method, which is capable of accurately identifying the position and color of traffic lights, allowing the vehicle to respond appropriately to traffic signals. The resulting model demonstrated a relatively high level of accuracy, achieving an mAP@0.50 score of 82.71%, and remained stable under various weather conditions, including rain and low-light environments. Furthermore, Oladri Renuka (2024) proposed the implementation of the YOLOv8 algorithm for multiclass traffic sign recognition. The model was trained using a dataset from Kaggle that included diverse environmental conditions and achieved an accuracy of 80.64% and a recall of 65.67% on the test data.

Furthermore, Mareeswari (2025) emphasized that traffic sign detection is a critical component in supporting road safety and the development of autonomous vehicles. The study proposed a real-time traffic sign detection and recognition method using the YOLOv8 algorithm, utilizing an integrated webcam and a dataset sourced from Roboflow. The experimental results showed that YOLOv8 achieved an accuracy of 94%, surpassing the performance of YOLOv7 (90.1%) and YOLOv5 (81.3%). In addition, the system was equipped with an audio warning feature to enhance driver awareness. These findings demonstrate that YOLOv8 exhibits superior performance in traffic sign detection with an average precision of 0.945 and holds significant potential for implementation in efficient and reliable intelligent transportation technologies.

Furthermore, Sumit Shevtekar & Shrinidhi kulkarni (2024) analyzed the effectiveness of the YOLOv8 architecture in detecting and recognizing traffic signs within autonomous vehicle systems and Advanced Driver Assistance Systems (ADAS). Through comprehensive preprocessing, training, and evaluation procedures using standard performance metrics such as mAP, precision, and recall, the YOLOv8 model achieved an accuracy of 95.2%, a recall of 98%, and an F1-score of 0.92. The study also highlighted several challenges, including lighting variations, weather conditions, and occlusion, while recommending the development of lightweight YOLOv8-based models for implementation on embedded systems. In addition, Mawaring Wening (2025) developed a YOLOv8-based traffic sign detection system implemented on a Raspberry Pi device using a dataset of 11,157 traffic sign images. The model achieved a precision of 96%, a recall of 83%, and an accuracy of 80% under daylight conditions.

In contrast to previous studies, this research focuses on the implementation of a web-based traffic sign detection system using the YOLOv8 algorithm. The model was not only evaluated in terms of accuracy but also integrated into an interactive Flask-based web system capable of processing both images and videos in real time while displaying visual detection results. Thus, this study complements earlier works by providing a direct application of a traffic sign detection model in the form of a web-based system that is ready for practical use in efficiently detecting traffic signs.

However, although the existing studies have shown promising performance results, most of them still remain at the experimental stage and do not provide direct implementation in a practical environment. Many previous systems are limited to offline testing or single-input detection, without integrating model deployment into platforms accessible to end users. In addition, research exploring YOLOv8 in real-time web-based environments is still limited, especially those that support multiple input modes such as images, videos, and camera streams simultaneously.

Therefore, this study not only focuses on optimizing the YOLOv8 model for traffic sign detection but also emphasizes system deployment by integrating the trained model into a Flask-based web interface. This novelty ensures that the proposed system is not only accurate but also practical and ready to be applied in real-world use cases, bridging the gap between research development and real implementation in intelligent transportation environments.

METHOD

The research methodology in this study is systematically designed following five main stages, as illustrated in the figure.

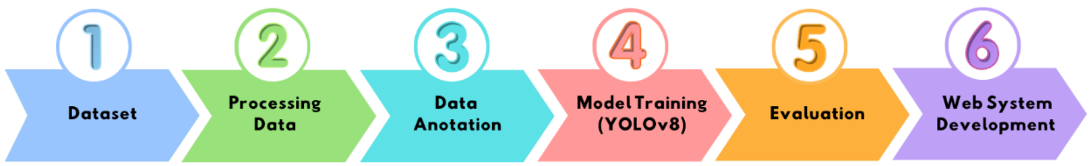

Figure 1 Research Flow

These stages encompass the entire process, starting from dataset collection to the final evaluation of the model, with a primary focus on developing a web-based traffic sign detection system using the YOLOv8 algorithm. This structured approach is implemented to ensure accurate and efficient detection results that can be deployed in real-time.

1. Dataset

The data used in this study consists of a collection of digital images of traffic signs, which serve as training data for the YOLOv8-based object detection model. All images were collected directly using a camera through field image acquisition at various roadway locations.

The dataset was then managed using the Roboflow platform, which facilitates storage, organization, and preprocessing prior to the training stage. The dataset comprises 20 classes of traffic signs with a total of 5,224 images, consisting of 4,558 images (87%) for training, 438 images (8%) for validation, and 228 images (4%) for testing.

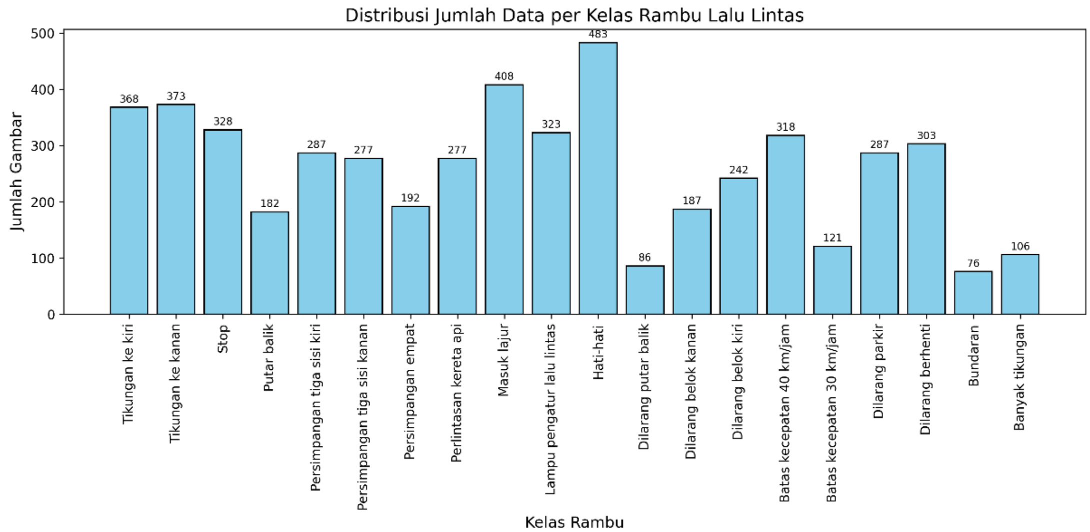

Figure 2 Data Distibution

To ensure reliability, every annotated image underwent manual verification to minimize labeling inconsistencies and maintain high data quality. Data validity was ensured by adhering to official Indonesian traffic sign standards and capturing real-world traffic conditions through direct field acquisition. However, the dataset was sourced from limited roadway locations, which may restrict the generalization capability of the model when applied to different regions with varying traffic sign characteristics. Future development is recommended to include broader environmental diversity and larger sample coverage to improve robustness and system adaptability in various operational conditions.

2. Processing Data

This stage focuses on improving image quality to meet the requirements of the YOLOv8-based detection model. The preprocessing procedure is performed to standardize the size and orientation of images, ensuring consistency prior to the training phase. Each image automatically undergoes an Auto-Orient process to adjust its orientation to the correct direction, allowing the system to recognize objects without positional interference. Subsequently, all images are resized to 640×640 pixels using the stretch-resize method, in accordance with the input specifications of the YOLOv8 model. This ensures uniformity of input dimensions, which is essential for maintaining performance stability and computational efficiency during training.

In addition, data augmentation is applied to enrich image variations within the dataset without manually increasing the dataset size. The augmentation techniques employed include rotation between –15° and +15° to simulate variations in camera viewing angles, as well as blur up to 2.5 pixels to replicate blurry conditions caused by vehicle motion or vibrations. Each training image generates up to three new augmented variations. These steps aim to improve the model's generalization ability so that it can accurately detect traffic signs even under suboptimal visual conditions, such as low lighting or unfocused images.

3. Data Annotation

The annotation process is carried out after all images have undergone the preprocessing stage, with the aim of ensuring that the data used possesses the appropriate structure and labeling required for model training. Each image is assigned a bounding box to indicate the area of the traffic sign and is labeled according to its corresponding class. This process is performed manually using the Roboflow platform, which supports YOLO output format (.txt) containing the object class label, center coordinates (x, y), and the width and height of the bounding box.

4. Model Training (YOLOv8)

The model training stage represents a critical component of this research, in which the YOLOv8 algorithm is employed to detect traffic sign objects. The dataset that has undergone annotation and preprocessing is divided into three subsets, consisting of training data (87%), validation data (8%), and testing data (4%). The training process is conducted on the Google Colab platform equipped with an NVIDIA Tesla T4 GPU (16 GB VRAM) to accelerate computation and support efficient deep learning model training.

Several key parameters were adjusted iteratively to obtain the optimal configuration for model performance. The parameters tested included epochs of 10, 20, 30, 40, and 50; batch sizes of 16, 32, and 64; and learning rates (lr0) of 0.001 and 0.01. Each combination was evaluated through a series of experiments to observe its effect on the model’s training outcomes.

Model evaluation was performed using three primary metrics—Precision, Recall, and mean Average Precision (mAP50)—which assess the model’s detection accuracy, completeness, and overall performance in recognizing traffic sign objects. Based on the experimental results, the best configuration was achieved with 50 epochs, a batch size of 32, and a learning rate of 0.001, producing a Precision of 0.9406, Recall of 0.9395, and mAP50 of 0.9748. These results indicate that the model is capable of detecting objects with high accuracy and efficiency.

5. Evaluation

The evaluation stage aims to assess the model’s performance in accurately detecting and classifying traffic signs. The evaluation was conducted using three primary metrics: Precision, Recall, and mean Average Precision (mAP50). These metrics provide a comprehensive overview of the model’s ability to deliver precise detections, maintain consistency, and achieve a high level of accuracy.

In addition to metric-based evaluation, manual testing was also performed on a set of test images to verify the consistency of detection results with real-world conditions. This step is essential for assessing the model's reliability when applied in practical environments and ensuring that the system can operate efficiently and accurately in real-time web-based implementations.

6. Web System Development

In addition to evaluating the model's performance, this traffic sign detection system was also developed as a web-based application using the Flask framework. Flask was selected due to its lightweight and flexible nature, as well as its compatibility with Python-based deep learning models such as YOLOv8. The system is designed to allow users to perform traffic sign detection directly through a web interface without requiring additional installation.

The user interface (UI) was developed using a combination of HTML, CSS, and JavaScript, ensuring a responsive and accessible layout across various devices. The application consists of several main pages, each serving a specific function, including:

1. Home Page, which presents a brief overview of the system's functionality and provides navigation to the main features.
2. Detection Page, which enables users to upload images or videos to be processed using the YOLOv8 model.
3. Real-Time Detection Page, which supports real-time detection through the user's device camera and displays annotated results, including bounding boxes and class labels.

Training Page, which allows users to retrain the model by configuring parameters such as epochs, batch size, and learning rate according to their needs.

FINDING AND DISCUSSION

RESEARCH RESULT

The model training process was conducted to examine the influence of parameter configurations on the performance of the YOLOv8-based traffic sign detection system. The model was trained using the dataset that had undergone preprocessing and annotation stages, with various parameter settings tested to obtain the optimal results.

The primary parameters evaluated included the number of epochs, batch size, and learning rate (lr0). Each parameter combination was executed independently to observe its impact on Precision, Recall, and mAP50. The training results from all parameter configurations are presented in Table 2 below.

Table 1 Hyperparameter optimization results

NO	Epochs	Batch Size	LR (lr0)	Precision	Recall	mAP50
1	19	16	0.001	0.8881	0.8565	0.9044

2	20	32	0.01	0.8939	0.8961	0.9269
3	30	64	0.001	0.9361	0.9293	0.9435
4	40	16	0.001	0.8975	0.9574	0.9529
5	50	32	0.001	0.9406	0.9395	0.9748

Based on the results presented in the table above, increasing the number of epochs and adjusting the batch size had a significant impact on improving model performance. The Precision value of 0.9406, Recall of 0.9395, and mAP50 of 0.9748 indicate that the model demonstrates excellent capability in recognizing objects with high accuracy and consistent detection. This configuration was selected as the final model implemented in the Flask-based web system.

To ensure the model's reliability in real-world conditions, manual testing was conducted using several test images representing variations in lighting conditions and camera viewpoints. The system's testing results are presented in Table 3, showing that the system was able to recognize all traffic sign categories with accuracy levels above 0.80.

Table 2 Manual Testing Results of the Traffic Sign Detection System

No	Factual Object	Detection Result	Remarks
1		Batas Kecepatan 40km/jam Acc 0,89	Valid
2		Dilarang berhenti Acc 0,85	Valid

3		Persimpangan Tiga Sisi Kiri dengan Prioritas Acc 0,93	Valid
4		Dilarang Belok Kanan Acc 0,90	Valid
5		Stop Acc 0,88	Valid

In addition to evaluating the model's performance, this study also produced a web-based traffic sign detection system integrated with the YOLOv8 model. The application was developed using the Flask framework, featuring an interactive and user-friendly interface. The primary objective of developing this application was to facilitate the detection of images, videos, and real-time camera streams through a responsive and efficient web display. Each page within the system has a specific function that supports the overall workflow of detection, training, and visualization of model results.

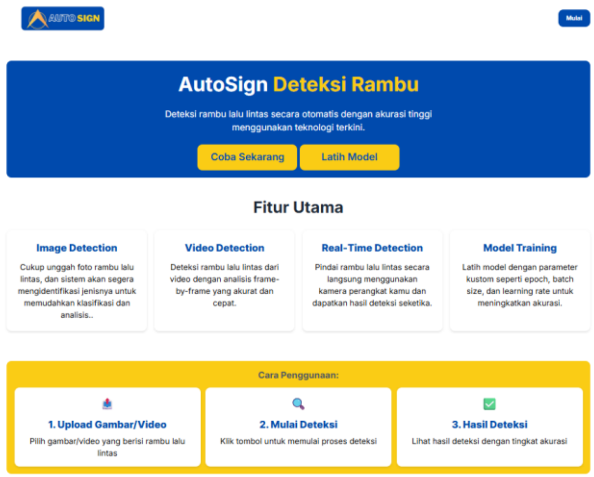

Figure 3 Home Page

Figure 3 illustrates the initial page of the application that appears when users first access the system. This page is designed to be informative and user-friendly, featuring a navigation layout that directs users to the main functionalities, including traffic sign detection and model training. The interface also provides a brief explanation of the core features and usage instructions, enabling users to gain an overall understanding before proceeding to the subsequent detection processes.

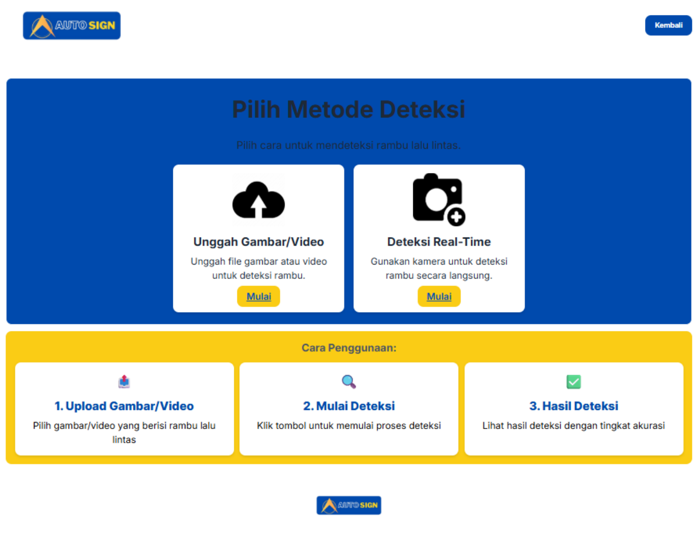

Figure 4 detection page

Figure 4 presents the detection page, which serves as the initial interface for users to select the type of input to be used in the detection process. On this page, users are provided with two main options: uploading an image or video, or utilizing the device's camera directly.

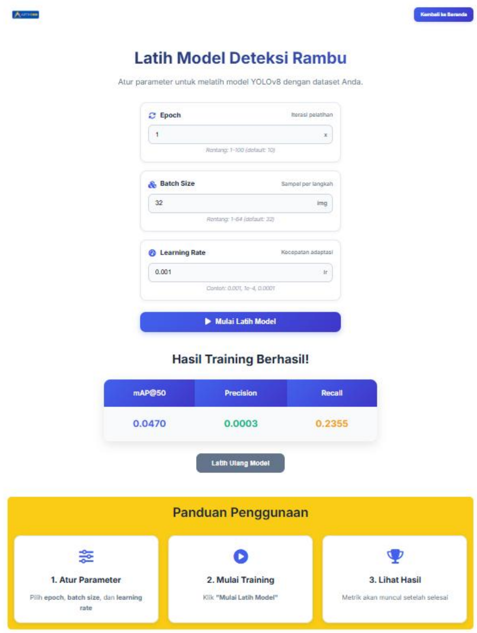

Figure 5 training page

Figure 5 illustrates the Model Training page, which functions to configure the training parameters for the traffic sign detection model. This page is designed to enable users to easily input parameter values such as epochs, batch size, and learning rate before initiating the training process. In addition, a "Mulai Latih Model" button is provided to execute the training according to the specified parameters.

The lower section of the page is equipped with step-by-step guidance, covering parameter adjustment, initiation of the training process, and viewing the resulting model metrics. The interface is designed to be simple yet informative, aiming to enhance user interaction and ensure that users clearly understand each stage of the model training process.

Figure 6 Image/Video Detection Page

Pada gambar 6 menunjukkan tampilan ketika pengguna memilih opsi deteksi gambar atau video. Setelah memilih model yang telah dilatih dan mengunggah foto atau video rambu, sistem akan memproses input tersebut menggunakan model YOLOv8 yang telah dipilih untuk mendeteksi rambu lalu lintas yang terdapat di dalamnya. Hasil dari proses ini akan divisualisasikan dalam bentuk citra yang telah dianotasi dengan bounding box dan label klasifikasi rambu, serta ditampilkan kembali pada halaman yang sama. Hal ini memudahkan pengguna dalam menilai keakuratan deteksi secara langsung.

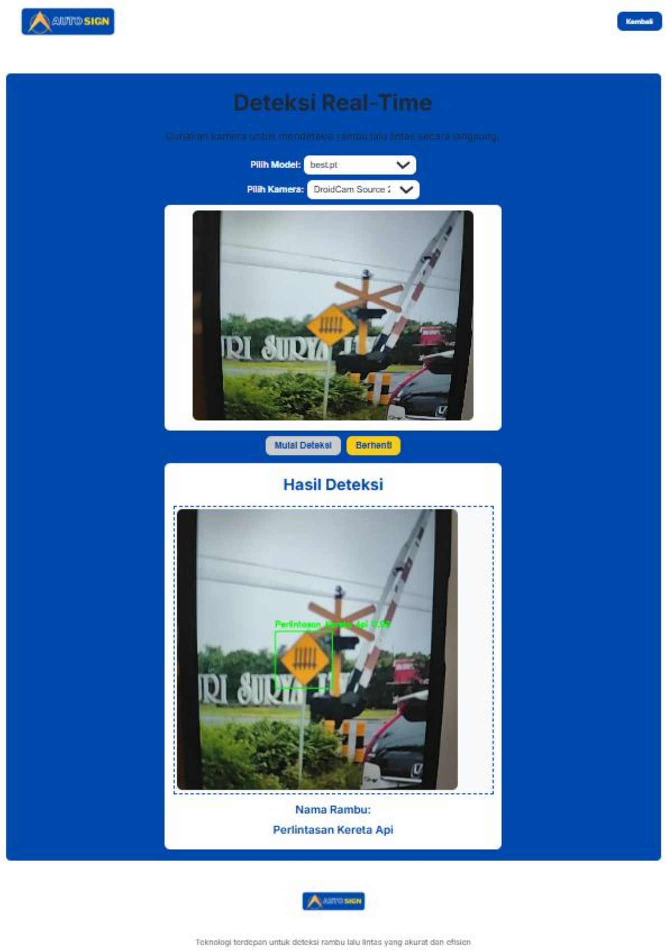

Figure 7 Real-Time Detection Page

Figure 6 shows the interface displayed when the user selects the image or video detection option. After choosing the trained model and uploading a traffic sign image or video, the system processes the input using the selected YOLOv8 model to detect the traffic signs contained within it. The results of this process are visualized in the form of annotated images featuring bounding boxes and classification labels, which are then presented on the same page. This allows users to conveniently assess the detection accuracy in real time.

DISCUSSION

The results of the study indicate that the traffic sign detection system based on the YOLOv8 algorithm and integrated into a Flask web application is capable of delivering optimal detection performance. The model, trained on 5,224 annotated images representing 20 classes of traffic signs, achieved high accuracy with a Precision of 0.9406, Recall of 0.9395, and an mAP50 of 0.9748 using the best configuration (50 epochs, batch size 32, and learning rate 0.001). These values demonstrate that the model exhibits stable detection capabilities and can accurately identify various types of traffic signs under different lighting conditions and viewing angles.

From a technical perspective, the findings confirm that the applied data preprocessing and augmentation stages significantly contributed to improving model performance. Processes such as rotation, blur, and resizing enhanced the model's ability to adapt to real-world visual variations, enabling effective detection even in challenging conditions such as low lighting or blurred images. Furthermore, the YOLOv8 architecture proved efficient in terms of processing time due to its lightweight design and real-time inference capability, making it suitable for deployment in web-based systems as well as field devices such as traffic surveillance cameras.

The integration of the model into a Flask-based application also provides added value in terms of usability. The system can process images, videos, and live camera feeds while displaying detection results visually with clearly annotated labels and bounding boxes. The interactive interface allows users to interpret detection outcomes easily without requiring extensive technical knowledge. The real-time detection feature is particularly important, as it enables immediate traffic sign identification and rapid output generation, making the system adaptable for implementation in autonomous vehicles or intelligent traffic monitoring systems.

Compared to prior works, the model performance obtained in this study (Precision 0.9406, Recall 0.9395, mAP50 0.9748) is superior to YOLOv8-based detection implemented by Oladri Renuka (2024) with 80.64% accuracy and also demonstrates improved robustness compared to the Raspberry-Pi deployment by (Mawaring Wening (2025) with 96% precision but lower recall. These improvements show a more balanced detection outcome while maintaining real-time responsiveness. Additionally, the successful deployment in a web environment indicates better system accessibility than studies that remain limited to offline execution. Thus, this research strengthens the implementation aspect of traffic sign detection and contributes new insight into the applicability of YOLOv8 within Intelligent Transportation Systems.

CONCLUSION

This study successfully developed a web-based traffic sign detection system using the YOLOv8 algorithm as the primary model for real-time object detection. Based on the training and testing results, the model demonstrated excellent performance with a Precision of 0.9406, Recall of 0.9395, and an mAP50 of 0.9748, indicating a high level of accuracy and consistent reliability in traffic-sign recognition.

The developed system is not limited to model implementation alone but is also integrated into an interactive Flask-based web application. This application allows users to perform detection on images, videos, and real-time camera streams. The detection results are displayed in the form of informative and easy-to-interpret visual annotations, supporting practical deployment in autonomous vehicle systems or intelligent traffic-monitoring applications.

Overall, the findings confirm that the YOLOv8 algorithm provides an effective solution for traffic-sign recognition with high computational efficiency. Nevertheless, the study still has limitations, particularly regarding model performance under extreme conditions such as low lighting, adverse weather, and unconventional viewing angles. Therefore, future research is recommended to expand the dataset with more diverse environmental variations and to incorporate deep-learning-based image enhancement techniques to improve model robustness and generalization in real-world scenarios.

REFERENCES

Akbar, M. (2022). *Pengenalan Rambu Lalu-lintas menggunakan Convolutional Neural Network (Studi Kasus: Rambu Lalu-lintas Indonesia)*. *6*(2). https://doi.org/10.30743/infotekjar.v6i2.4564

Arya Satya Pratama, D. (2024). *Pengembangan Kontrol Adaptif untuk Kendaraan Otonom dengan Studi Kasus pada Mobil Elektrik Berbasis Deep Reinforcement Learning*. 18–2024.

Ayala, R., & Mohd, T. K. (2021). Sensors in Autonomous Vehicles: A Survey. *Journal of Autonomous Vehicles and Systems*, *1*(3). https://doi.org/10.1115/1.4052991

Dedy Kusuma, & Mauizah. (2023). Deteksi Lampu Lalu Lintas Menggunakan YOLOuntuk Autonomous Car. *Journal of Computer Engineering, Network, and Intelligent Multimedia (JCENIM)*.

Fadilla Putri, R., Hidayatullah, Y. P., Aryansyah, A. D., & Feni, R. (2024). SOSIALISASI PENGENALAN RAMBU LALU LINTAS SERTA PENERAPAN HUKUM SEJAK DINI PADA SISWA SDN 26 KOTA BENGKULU. In *Jurnal Kajian Ilmiah Interdisiplinier* (Vol. 8, Issue 9).

IKBAL, MUH., & Saputra, R. A. (2024). PENGENALAN RAMBU LALU LINTAS MENGGUNAKAN METODE YOLOV8. *JIKA (Jurnal Informatika)*, *8*(2), 204. https://doi.org/10.31000/jika.v8i2.10609

Kresnanto, N. C., & Bahy, M. B. (2023). Method for Establishing a Spatial Database of Traffic Signs with Machine Learning. *MEDIA KOMUNIKASI TEKNIK SIPIL*, *29*(1), 61–69. https://doi.org/10.14710/mkts.v29i1.49928

Liyanthy, M. (2024). Perancangan Data Pipeline Computer Vision untuk Mendukung Pelatihan Agen Otonom dalam Mengenali Rambu Lalu Lintas. In *Journal of Computer and Information Systems Ampera* (Vol. 5, Issue 1). https://journal-computing.org/index.php/journal-cisa/index

Mareeswari, V., Vijayan, R., Nisthar, S., & Krishnan, R. B. (2025). Traffic Sign Detection and Recognition Using Yolo Models. *International Journal of Information Technology and Computer Science*, *17*(3), 13–25. https://doi.org/10.5815/ijitcs.2025.03.02

Mawaring Wening, P., Hendry, J., & Wicaksono Santoso, A. (2025). Traffic Signs Detection System Based on You Only Look Once (Yolov8) using Raspberry Pi. *Jurnal Pekommas*, *10*(1). https://doi.org/10.56873/jpkm.v9i1.5832

Mita Nalsalisa Br Barus, Clarissa Oktoferin Sinaga, Vadillah Raihto Hutasuhut, & Syuratty Astuti Rahayu Manalu. (2024). Analisis Dampak Pelanggaran Lalu Lintas terhadap Keselamatan Masyarakat dan Upaya Mempromosikan Nilai-Nilai Kemanusiaan dan kesadaran berkendara. *Jurnal Insan Pendidikan Dan Sosial Humaniora*, *2*(3), 150–160. https://doi.org/10.59581/jipsoshum-widyakarya.v2i3.3438

Oladri Renuka. (2024). A YOLOv8-based approach for multi-class traffic sign detection. *International Journal of Science and Research Archive*, *11*(2), 824–829. https://doi.org/10.30574/ijsra.2024.11.2.0499

Sumit Shevtekar, & Shrinidhi kulkarni. (n.d.). *Traffic-sign Recognition and Detection using Yolo-v8*. https://doi.org/10.13140/RG.2.2.19007.11683

Taufiqurrahman, T., Hadi, A. P., & Siregar, R. E. (2024). Evaluasi Performa Yolov8 Dalam Deteksi Objek Di Depan Kendaraan Dengan Variasi Kondisi Lingkungan. *Jurnal Minfo Polgan*, *13*(2), 1755–1773. https://doi.org/10.33395/jmp.v13i2.14228

Tessa Ningrum, A., Wijay, R., Rizal Abdul Aziz, M., Yudha Mauluda, M., & Rosyani, P. (2024). Face Deteksi Objek pada Gambar dan Video dengan YOLOv8 (Counting Objects). In *Jurnal Artificial Inteligent dan Sistem Penunjang Keputusan* (Vol. 2, Issue 2). https://jurnalmahasiswa.com/index.php/aidanspk

Wang, T. S., Kim, G. T., Shin, J., & Jang, S. W. (2024). Hierarchical Image Quality Improvement Based on Illumination, Resolution, and Noise Factors for Improving Object Detection. *Electronics (Switzerland)*, *13*(22). https://doi.org/10.3390/electronics13224438