Journal of Computer Science

Original Research Paper

AI Drive Assist: Enhancing Road Safety through Advanced Road Sign Detection and Driver Alerts

Lazeen Manasia, Mufaddal Bharmal, Harshita Singhvi, Gaurav Mehta, Aruna Gawade, Nilesh Rathod and Angelin Florence

Department of Artificial Intelligence and Machine Learning, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

Article history
Received: 10-02-2024
Revised: 25-04-2024
Accepted: 09-05-2024

Corresponding Author:
Nilesh Rathod
Department of Artificial Intelligence and Machine Learning, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
Email: nilesh.rathod@djsce.ac.in

Abstract: Road accidents claim approximately 1.35 million lives annually, as highlighted by the World Health Organization, making them a leading cause of death worldwide. With this sobering statistic in mind, addressing road safety concerns through advanced AI technologies becomes increasingly imperative. This research paper highlights practical implications for both road safety enhancement and the advancement of artificial intelligence technology within the automotive industry. The proposed model, "AI-drive assist," investigates the effectiveness of an AI-driven driving assistant system in improving road safety. By providing real-time auditory alerts to drivers, the driving assistant system facilitates a better understanding of road conditions and encourages the development of safer driving habits. The methodology entails the utilization of the You Only Look Once (YOLO)v8 model within a ResNet-50 CNN framework, allowing for the efficient extraction of relevant information from input photographs. Through rigorous evaluation, the system achieves an impressive precision-recall rate of 94% in identifying various road signs, indicating its potential to enhance driver awareness and promote compliance with traffic regulations. Additionally, data augmentation techniques are employed to diversify the training dataset, further improving accuracy and robustness. The findings of this study underscore the significant impact of AI technologies on promoting safer driving practices. Overall, this study contributes to the ongoing discourse on road safety improvement and demonstrates the tangible benefits of integrating AI technologies into driving assistance systems.

Keywords: YOLO, Computer Science, CNN, Artificial Intelligence

Introduction

Missed road signs due to human errors pose a significant challenge to road safety despite notable advancements in technology. According to road accident statistics, speeding accounted for a substantial portion of incidents in 2019, contributing to 71% of accidents and resulting in numerous fatalities and injuries. Lane indiscipline and balance violations such as drunk driving and using mobile phones also played significant roles, collectively contributing to 11% of accidents and 14% of deaths. Additionally, distracted driving remains a persistent issue, especially among younger drivers, posing risks at any time of the day. Addressing these factors through stricter enforcement measures is imperative to improve road safety (Hugar *et al*., 2021).

Efforts to address the challenges posed by overspeeding, lane indiscipline, and distracted driving must be complemented by advancements in road safety technology. The landscape of road safety technology has demonstrated notable advancements, yet persistent limitations in traditional automobile systems highlight the urgent need for innovation (Dhawan *et al*., 2023) Current technologies often struggle to promptly convey vital road sign information to drivers, posing significant safety risks. Moreover, the integration of Artificial Intelligence (AI) remains insufficient, hindering immediate comprehension of road signals and potentially leading to driver errors. These limitations primarily stem from inadequate computational capabilities and an inability to adapt swiftly to evolving driving scenarios. The present technology is constrained by two primary limitations:

1. A lack of high-performance computer capabilities
2. A restricted capacity to adjust to swiftly evolving driving situations

The main objective of AI-drive assist is to mitigate the risks associated with missed road signs due to human errors by addressing the limitations posed by a lack of high-performance computer capabilities and a restricted capacity to adjust to swiftly evolving driving situations. In response to the research problem, the AI-drive assist system emerges as a pioneering solution poised to revolutionize road safety. By seamlessly integrating state-of-the-art AI technology, including the renowned You Only Look Once (YOLO)v8 concept, this system promises to address critical shortcomings in existing infrastructure (Flores-Calero *et al*., 2024). The YOLOv8 model, celebrated for its rapid object recognition capabilities, offers precise detection of road signs, particularly crucial during high-speed driving situations. Moreover, to enhance the robustness of object detection, data augmentation techniques have been employed to balance class distributions, ensuring equal representation across various sign types.

Beyond mere detection, the AI-drive assist system incorporates an innovative feature where detected road signs are converted into speech alerts delivered directly to the driver (Manawadu and Wijenayake, 2024). This proactive approach not only informs the driver of pertinent road information but also minimizes distractions, thereby enhancing overall safety. Central to this system is the seamless integration of advanced AI technology with real-time feedback mechanisms, ensuring timely and accurate communication of critical information to the driver.

The benefits of the AI-drive assist technology are multifaceted (Zhu and Yan, 2022). Not only does it provide prompt information to drivers, but it also possesses the agility to adapt to rapidly changing road conditions. This enhancement of transportation system intelligence optimizes efficiency and ensures heightened safety. With signage now incorporating advanced decision-making algorithms (Wan *et al*., 2021), proactive messages regarding potential hazards and evolving road conditions can be seamlessly integrated, further improving driving safety and convenience.

The AI-drive assist system has a wide-ranging applicability in the whole vehicle industry and it will play a crucial role in enhancing road safety and efficiency in the future. The system's flawless incorporation of cutting-edge AI technology facilitates a transformative shift in road safety, setting a precedent for the benefit of society at large. It establishes the conditions for a future where transportation and technology collaborate seamlessly, ushering in a new era of road safety through the deliberate integration of cutting-edge advancements in AI. This game-changing system is evidence of the covert potential of Artificial Intelligence (AI) to create covertly safer and more effective roads, paving the way for a future in which transportation and technology coexist peacefully for the covert benefit of all.

Literature Survey

The field of traffic sign detection in difficult contexts is explored in the 2022 research paper "M-YOLO: Traffic sign detection algorithm applicable to complex scenarios" by Liu, Y.; Shi, G.; Li, Y.; Zhao, Z., which was published in symmetry. Using the You Only Look Once (YOLO)v3 algorithm, the study concentrates on identifying 11 commonly found traffic signs throughout Europe. The paper utilizes a dataset acquired from front-view camera footage in Osijek. The collection comprises photos depicting many weather situations, encompassing cloudy, bright, rainy, and nocturnal landscapes. The collection comprises 5567 photos with 6751 annotated traffic signs, which were captured from 28 video sequences. Remarkably, the proposed M-YOLO method demonstrates exceptional performance on this vast dataset. This study effectively recognizes and alerts drivers to common European traffic signs, hence enhancing road safety and driver awareness (Liu *et al*., 2022).

The research paper "traffic sign detection algorithm based on improved YOLOv4-Tiny," published in 2022 by Yao *et al.* (2022) introduces a refined method for enhancing traffic sign detection using the YOLOv4 tiny algorithm. The article, accessible on Science Direct, employs Receptive Field Blocks (RFB) and an Adaptive Feature Pyramid Network (AFPN) to augment feature fusion and extraction, therefore surmounting the limitations of YOLOv4-Tiny. The examination of the CCTSDB and GTSDB datasets demonstrates the effectiveness of these enhancements in traffic sign detection, as seen by their superior precision, recall, map, and competitive speed (Yao *et al*., 2022).

Zhang *et al.* (2020) presented a real-time traffic sign detection technique based on YOLOv3. The study, published in IEEE access, concentrates on innovative techniques for identifying small traffic signs and attains commendable outcomes in terms of precision, recall, and map metrics. This study contributes to the development of real-time small-sign detection algorithms and demonstrates the potential of YOLOv3 in this specific context (Zhang *et al*., 2020).

In their 2022 research paper titled "Traffic Sign Detection Algorithm Based on Improved YOLOv4," Wu and Cao introduce an upgraded traffic sign recognition algorithm based on YOLOv4. The paper was published in the Journal of Physics: Conference series. The project leverages YOLOv4's efficient real-time object detection capabilities to enhance the ability of self-driving automobiles to recognize traffic signs. The effectiveness of their approach was proven by the exceptional outcomes of the two models that underwent training, one using the GTSDB dataset and the other using a custom dataset. The models attained a Mean Average Precision (MAP) of 94% on the German Traffic Sign Detection Benchmark (GTSDB) and 92% on their own custom dataset (Wu and Cao, 2022).

DOI: 10.3844/jcssp.2024.885.897

Zhang X.'s work, titled "traffic sign detection based on YOLOv3", focuses on a more efficient version of YOLOv4 for identifying traffic signs in unmanned driving vehicles. To optimize model efficiency and accelerate detection, this approach replaces YOLOv4's feature extraction network with MobileNetV2, thereby reducing the number of model parameters. To enhance the process of extracting features and managing gradients, attention, and residual structures are incorporated. As a result, a more compact model is achieved, surpassing YOLOv4 in terms of overall accuracy, while simultaneously improving detection speed by 58.8% and reducing it by 2.5% (Zhang, 2023).

In their publication in axioms, Bai *et al*. (2023) introduce two novel YOLOv5-based models for traffic sign identification. The article utilizes YOLOv4, a rapid and real-time object recognition technology, to aid self-driving vehicles in identifying traffic signs. Two models are trained on both the GTSDB and a bespoke dataset, resulting in an impressive performance with a 94% mean average precision (map) on GTSDB and a 92% map on their dataset. This demonstrates the efficacy of their technique in enhancing the efficiency of traffic sign detection for autonomous vehicles (Bai *et al*., 2023).

The publication "traffic sign detection in an unconstrained environment using improved YOLOv4" by Saxena *et al*. (2024) introduces a lightweight traffic sign detection system that is based on YOLOv4. To enhance efficiency and decrease the number of parameters, the method employs depth-wise separable convolution and MobileNetv3. Additional developments include the incorporation of SPP modules within the feature pyramid and enhancements made to the MobileNetv3 network. According to the German Traffic Sign Detection Benchmark (GTSDB), the system exhibits superior performance in detecting traffic signs compared to YOLOv4. The model attains a 1.7% enhancement in Mean Average accuracy (MAP), reduces the number of parameters by 197 million, and improves processing time by 25% (Saxena *et al*., 2024).

The 2022 study conducted by researchers from New York, which was published in the journal Heliyon, centers on the process of identifying and classifying traffic signs. This is achieved through the utilization of advanced computer vision models, specifically YOLOv4 and Faster R-CNN. To address issues pertaining to small signs, the study employs CSPDarknet53 to develop an adapted model based on YOLOv4 for accurate and resilient identification of traffic signs. By applying data preprocessing techniques, picture augmentation methods, and considering low-light conditions at night, the model achieves an impressive accuracy of 94.80 and 80.71% on the TT-100 K and MTSD datasets, respectively. The model's flexibility is demonstrated through cross-data testing on GTSDB and ITSD datasets, where it achieves superior performance compared to other models, with accuracy rates of 91.74 and 63.64%, respectively (Youssouf, 2022).

Comparison Analysis of YOLOv8 with State-of-Theart Models

In their publication in axioms, Bai *et al*. (2023) introduce two state-of-the-art models examined in this research that have significantly advanced traffic sign detection, each bringing unique contributions to the field. M-YOLO, leveraging YOLOv3, showcases exceptional performance in identifying common European traffic signs across varied weather conditions, bolstering road safety and driver awareness. Yao *et al*. (2022) YOLOv4-tiny improvement, integrating receptive field blocks and an adaptive feature pyramid network, demonstrates superior precision and recall in traffic sign detection, particularly evident in CCTSDB and GTSDB datasets. Similarly, Zhang *et al*. (2024) YOLOv3-based real-time detection technique proves effective in identifying small traffic signs, showcasing potential for real-time applications. Wu and Cao's enhanced YOLOv4 algorithm excels in real-time object detection, showcasing high precision on both standard and custom datasets, further enhancing self-driving car capabilities.

On the other hand, Zhang's optimization of YOLOv4 with MobileNetV2 exhibits improved efficiency and accuracy in unmanned driving vehicles, presenting a more compact yet effective solution. Bai *et al*. (2023) YOLOv5-based models demonstrate impressive performance in aiding self-driving vehicles, showcasing the adaptability of YOLOv4 for traffic sign identification. Saxena *et al*. (2024) lightweight YOLOv4-based system, employing depth-wise separable convolution and MobileNetv3, offers enhanced efficiency and parameter reduction, showcasing superior performance compared to standard YOLOv4 in detecting traffic signs.

In comparison, the utilization of YOLOv8 presents a compelling advancement in traffic sign detection. YOLOv8 integrates the strengths of previous models while offering a unified architecture for enhanced efficiency and accuracy. By combining the efficiency of YOLOv4 with the adaptability of faster R-CNN, YOLOv8 simplifies the research pipeline and offers scalability and generalization capabilities. Moreover, YOLOv8's advanced techniques, including CSPDarknet53 and MobileNetV2, ensure high accuracy even in challenging conditions, making it a practical choice for real-time applications. Overall, YOLOv8 represents a significant advancement in traffic sign detection technology, offering researchers a powerful and versatile tool for addressing real-world challenges in traffic management and autonomous driving systems (Gašparović *et al*., 2023).

Materials and Methods

The following Fig. 1 illustrates the block diagram outlining the comprehensive workflow of the application's model.

Data Collection and Preprocessing

During the early development of the system, we utilized the LISA dataset sourced from Roboflow as the primary repository of traffic sign photos. The Long-term Infrastructure and Short-term Activities (LISA) dataset offers a comprehensive collection of images tailored for tasks involving traffic sign detection and identification. Covering a diverse range of real-world scenarios, this dataset provided a valuable resource for the development and evaluation of computer vision algorithms. With annotations for various traffic signs, including those pertinent to autonomous vehicles and traffic management, the LISA dataset proved invaluable for both system development and evaluation.

Upon meticulous examination, we identified a significant disparity in the distribution of classes within the dataset, underscoring the need for a purposeful data preprocessing strategy. The primary objective was to enhance the dataset to optimize the efficacy of model training. To accomplish this, we leveraged the Albumentations library, a powerful tool for image augmentation. This framework facilitated the implementation of augmentation procedures aimed at enriching the diversity of the dataset, thereby fostering improved model generalization.

The augmentation techniques employed included random cropping to ensure consistent dimensions of 250×250 pixels, as well as random adjustments to brightness and contrast with a probability of 0.2. Additionally, horizontal flips were integrated with a probability of 0.5. By deliberately introducing variability through these augmentation strategies, the methodology is aimed to artificially enrich the richness of the dataset, thereby enhancing the model's ability to generalize across different scenarios (Mumuni and Mumuni, 2022).

Furthermore, to facilitate effective training and evaluation of the system, the dataset was split into distinct subsets. The largest portion, comprising 86% of the dataset, was allocated for training the image segmentation model. A smaller subset, accounting for 9%, served as the validation set to monitor model performance during training and prevent overfitting. Finally, a 6% portion was reserved as the test set for the final evaluation of the model's performance after training completion.

In addition to these preprocessing steps, further enhancements were applied to ensure the consistency and reliability of the dataset. Auto-orientation was applied to standardize the orientation of images, while all images were resized to a uniform resolution of 416×416 pixels to streamline the training process. Moreover, augmentations were performed on each training example, introducing variations to enhance the robustness of the model. Notably, a small amount of noise, up to 1% of pixels, was injected into bounding boxes to enhance the model's resilience to minor inaccuracies in object localization.

By integrating these preprocessing techniques and dataset splits into the methodology, we aimed to lay a robust foundation for the subsequent training and evaluation of the system.

Model Selection and Training

A comprehensive evaluation was carried out on multiple object detection architectures, such as faster R-CNN, SSD, RetinaNet, EfficientDet, and Mask R-CNN, throughout the crucial stage of Model Selection and Training. Every option was carefully examined for its unique advantages and disadvantages. Although Faster R-CNN was a popular model because of its multi-stage approach, it had disadvantages because it was slower at inference than the selected YOLOv8. YOLOv8 outperformed SSD in real-time object detection due to its better speed-accuracy balance, which is especially important for real-time traffic sign detection. In the intended domain, RetinaNet's capacity to manage imbalanced classes was not superior to YOLOv8's performance. Even while algorithms like Mask R-CNN and EfficientDet were more effective, YOLO-especially the YOLOv8s variant was the best option. YOLOv8 showed an unparalleled balance between speed and accuracy, allowing it to smoothly adapt to the real-time requirements of the AI-drive assist system's traffic sign detection (Zhang and Zhao, 2022).

The Table 1 shows that the comparative analysis of object detection algorithms, Python programming language was employed alongside prominent deep learning frameworks such as TensorFlow and PyTorch. These frameworks facilitated the implementation of various state-of-the-art algorithms, including R-CNN, SSD, RetinaNet, EfficientNet, Mask R-CNN, and YOLOv8 (Lou *et al*., 2023). The implementation process included loading pre-trained models available within each framework and conducting inference on the test dataset. This dataset comprised diverse real-world images and OpenCV was utilized for image processing tasks and model evaluation. To assess the effectiveness of the algorithms, performance metrics such as speed measured in Frames Per Second, (FPS), mean Average Precision (mAP), and inference time (measured in milliseconds) were computed. These metrics provided insights into the efficiency and accuracy of each algorithm in detecting objects within the images (Ahmad *et al*., 2020).

The YOLOv8 model, a focus of the analysis, was trained on the Google Colab platform using the Tesla T4 GPU for enhanced computational power. The training process spanned 25 epochs to ensure the model's optimization for reliable results.

Table 1: Comparative analysis of algorithms

Algorithms	Speed (FPS)	Metrics for evaluation: Mean Average Precision (mAP) %	Inference time (ms)
R-CNN	1-2	25-30	500-600
SSD	15-20	30-35	50-60
RetineNet	5-7	35-40	150-200
EfficientNet	20-25	40-45	40-50
Mask R-CNN	1-2	30-35	500-600
YOLOv8	30-40	50-55	25-30

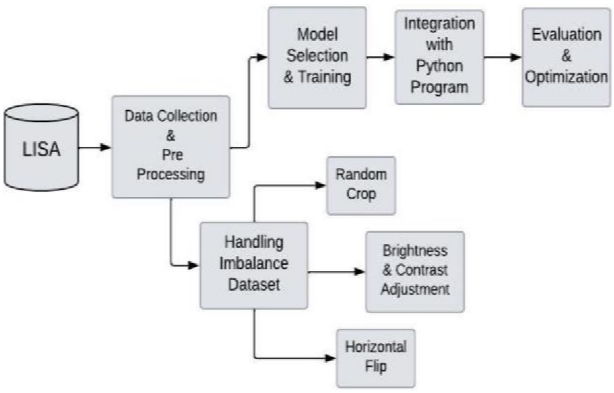

Fig. 1: Block diagram of a methodology of AI drive assist

This version is designed to respond quickly to the changing road environment by meeting the real-time requirements of traffic sign detection. The model uses gradient descent and backpropagation to fine-tune its parameters throughout 25 epochs of training. The model training utilized 25 epochs with images resized to 640×640 pixels. The optimizer employed was the Adam optimizer, while the loss function used was the detection loss function. Data augmentation techniques included random cropping, as well as brightness and contrast adjustments, with probabilities of 0.2 and 0.5, respectively. Gradient descent iteratively optimizes parameters to minimize the detection loss function, while backpropagation computes and modifies parameter gradients to enable enhanced prediction capabilities. The care taken in the selection and training phases demonstrates the dedication to developing a highly accurate and performant model for the AI-drive assist system's real-time traffic sign identification.

Integration with Python Program

Owing to the flexible OpenCV library, the trained YOLOv8 model converges smoothly with a Python-based architecture. With the help of this integration, camera feeds are processed in real-time and the YOLOv8 model performs in-depth analysis on each collected frame to detect road signs. By using the pyttsx3 library, the system's responsiveness is increased even further by producing audio alarms that are precisely timed to coincide with the model's detections (Wang *et al*., 2023). This audio response, which is closely connected to the traffic signs that are being seen, provides a quick and efficient means of informing the driver of important directions and information. The Python program's integration of these technologies creates a dynamic and responsive AI-drive assist system that improves user awareness and makes a substantial contribution to the overall safety and intelligence of driving (De Pra and Fontana, 2020).

System Operation

The front camera, which is placed inside the car in a strategic way to take a constant stream of pictures of the surrounding road scene, is activated first by the AI-drive assist system. The basis for real-time monitoring and analysis of the dynamic road environment is laid by this continuous image capture.

YOLOv8 Model Processing

Using the state-of-the-art YOLOv8 model, which is well known for its effectiveness in object detection, the collected images go through a complex processing stage. One of YOLOv8 key differentiators is its ability to analyze the full image in a single pass, which is essential for quick response in a real-time traffic scenario driving (Zhang *et al*., 2023).

Object Detection in YOLOv8

YOLOv8, expanding upon its original processing, creates a grid of cells out of the input image and gives each one the job of estimating bounding boxes and class probabilities for items that might be inside its spatial bounds. By incorporating anchor boxes, the model improves bounding box predictions by using predetermined forms that change dynamically as the model is trained.

Confidence Scores and Class Probabilities

YOLOv8 creates several bounding boxes, each with a confidence score and a class probability, inside each grid cell. Class probabilities measure the chance that an object will belong to a specific predefined class, whereas confidence ratings indicate the model's certainty that an object exists within a given bounding box.

Non-Maximum Suppression

A crucial post-processing technique called Non-Maximum Suppression (NMS) is used to improve the forecasts. In this step, redundant or low-confidence bounding boxes are progressively filtered out, leaving just the most accurate and confident predictions. The result doesn't get cluttered with overlapping or less certain forecasts thanks in large part to NMS.

Alert Trigger Mechanism

After NMS, predictions pass via a thresholding process that determines if the road signs it detects are important enough to pay attention to. If the answer is yes, a voice alarm is set off, which is intended to quickly notify the driver of the type of traffic sign that has been spotted as well as any relevant instructions.

Driver Notification

The system rapidly notifies the driver following a positive determination from the thresholding process. The driver is guaranteed to receive up-to-date information on the identified road signs either the vehicle's audio system or a dedicated alert system. Overall road safety and the driver's situational awareness are greatly enhanced by this open communication.

Continuous Operation

All steps of the process run smoothly in an ongoing cycle. As the car travels down the road, the front camera continuously takes pictures and the YOLOv8 model continuously reads, interprets, and categorizes traffic signs. This constantly active system architecture guarantees a continuous, real-time awareness of road signs, encouraging the user to drive defensively and safely.

Handling Imbalanced Dataset

To address the intrinsic class imbalance present in the LISA dataset, the methodology employed a methodical preprocessing strategy that mostly relied on data augmentation approaches. The Albumentations library was a key component of this plan, acting as a flexible instrument to add much-needed variability, especially for underrepresented classes. This intentional decision was made with the dual goals of strengthening the durability of the dataset as a whole and increasing the exposure of underrepresented classes. The choice to utilize data augmentation alone, without including other methods, emphasizes the dedication to a focused and efficient approach that puts simplicity and efficacy first. Although there is room for future iterations to investigate more sophisticated techniques to address class imbalance, the existing methodology is a testament to the strong effectiveness of data augmentation (Shorten and Khoshgoftaar, 2019). We purposefully highlight this method in order to provide the groundwork for demonstrating its adaptability and efficacy as the main strategy for attaining class parity in the complex environment of road sign recognition. In addition to improving model generalization, the intentional introduction of variability through augmentation is in line with the overall objective of creating a dependable and flexible traffic sign-detecting system.

Evaluation and Optimization

To ensure the efficacy and efficiency of the model, a multidimensional approach is used in the rigorous evaluation and optimization phase of the AI-drive assist system (Rathod and Wankhade, 2020). Evaluation metrics function as quantitative measures to evaluate the performance of the model. Examples of these metrics are precision, recall, and F1 score. Recall evaluates the model's ability to catch all pertinent cases, whereas precision shows how accurate positive detections are. The F1 score offers a thorough insight into the model's overall efficacy by striking a balance between precision and recall.

It is essential to optimize the AI-drive assist system continuously. Through careful modifications to the model architecture, this iterative method makes use of insights gleaned from evaluation metrics. Furthermore, sophisticated methods are investigated to extend the limits of detection powers. Hyperparameters are adjusted to improve the system's overall accuracy and efficiency. With unmatched accuracy and dependability, the AI-drive assist system can adapt and withstand changing traffic conditions thanks to its dynamic optimization technique.

Implementation

The following architecture diagram given in Fig 2. illustrates the overall implementation of the system.

The incorporation of the AI-drive assist system into a car camera requires a sophisticated strategy that includes both software and hardware setups. The vehicle environment needs to be equipped with an advanced camera system that can record footage in real-time. Simultaneously, the necessary software stack, which includes the OpenCV library, the Python runtime environment, and model dependencies, needs to be installed on the car's embedded computer system.

On the vehicle computer unit, the pre-trained YOLOv8s model for real-time traffic sign detection is installed. A custom Python application is carefully written to interact with the live video stream, making use of the OpenCV package to process video. The YOLOv8s model is smoothly integrated by this application to perform object detection on the incoming video frames.

Predictions are then evaluated and if relevant traffic signs are identified, audio alarms are initiated. Through the audio system of the car, the auditory alert system provides the driver with timely and relevant feedback (Sahithi *et al*., 2023).

The system's operating paradigm is a never-ending loop that guarantees road sign processing always occurs while the car is moving. Thorough testing protocols, calibration of the system, and possible user interface integrations all help to realize a stable AI-drive assist system that blends in well with the vehicle camera system, improving road safety and driver awareness.

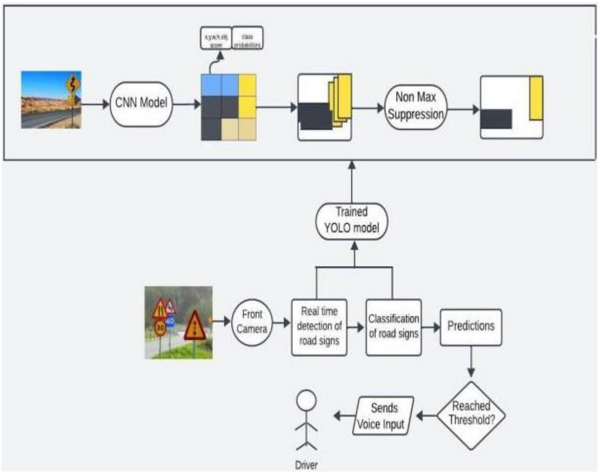

Fig. 2: Architecture diagram

Furthermore, the AI-drive assist system acknowledges the dynamic nature of road situations and places a high priority on adaptability and scalability. The YOLOv8 model is updated and improved upon continuously and these updates and improvements are easily included into the system, guaranteeing its adaptability to changing traffic conditions and new road sign regulations. Because of the software architecture's ability to support remote updates, the most recent developments in computer vision and machine learning may be incorporated without the need for human participation.

Moreover, the AI-drive assist system incorporates sophisticated decision-making algorithms, which go beyond the recognition of traffic signs. The system can deliver anticipatory alerts for potential risks, fluctuating speed limits, and changing road conditions by analyzing the contextual information collected from the camera feed. This comprehensive strategy promotes a symbiotic interaction between cutting-edge technology and human intuition on the road, improving both road safety and the driving experience by making it more comfortable and informed.

Results and Discussion

The ensuing presentation outlines the results, depicting the comprehensive workflow of the application's model.

In binary classification, the F1 confidence curve is a useful visual aid that illustrates the complex relationship between the F1 score and confidence thresholds. With the use of this graphical depiction, practitioners can investigate the effects of changing confidence thresholds on the balance between recall and precision. The curve illustrates the F1-score over a range of confidence levels and sheds light on the dynamics of the model's performance. It helps identify the ideal threshold that minimizes both false positives (precision) and false negatives (recall), enabling practitioners to make well-informed decisions regarding the deployment and tuning of the model for particular use cases. The confidence threshold, or the lowest degree of confidence needed for a detection to be regarded as a positive prediction, is shown by the x-axis of the F1 confidence curve graph. Figure 3 gives that the F1 score, which is the harmonic mean of recall and precision, is represented on the y-axis. When the confidence level is set to 0.709, the F1-score for all classes is 0.94. This results in a high F1-score, indicating that the model's predictions are very accurate at a particular confidence level. At a given confidence threshold, a higher F1-score denotes a better balance between precision and recall; the curve can be used to determine the confidence value that optimizes the model's overall performance.

Additionally, the F1 confidence curve provides valuable insights into the trade-off between precision and recall, allowing practitioners to make informed decisions about the model's performance in real-world scenarios. By analyzing this curve, stakeholders can fine-tune the model's threshold to meet specific requirements and optimize its performance for different applications.

A precision-confidence curve is a graphical depiction that helps assess how different confidence levels affect prediction precision by showing the relationship between precision and varying confidence thresholds in a classification model. With the confidence level set at 1.00, the precision of the model for all classes is 0.98. The precision for each class is shown by the y-axis, while the confidence threshold is represented by the x-axis. When the confidence criterion is set to 1.00 in this instance, the precision for every class is 0.98. This results in a high precision score, indicating that the model's predictions are quite accurate at a particular confidence level. A model that scores higher on precision is more likely to produce fewer false positive predictions. As a result, the precision confidence curve graph is useful for choosing a confidence threshold that maximizes the overall precision of the model.

Figure 4 shows a precision-recall curve, which is a common metric used to evaluate the performance of object detection models. The curve illustrates the tradeoff between precision (the proportion of true positives among all predicted positives) and recall (the proportion of true positives among all actual positives) for a given object detection model. In this case, the curve represents the performance of the model on all classes (all classes). The mAP@0.5 value, which is also indicated on the curve, represents the mean Average Precision (mAP) calculated at an Intersection over Union (IoU) threshold of 0.5. This metric provides a single value to summarize the overall performance of the model across all classes and IoU thresholds. As can be seen from the figure, the model achieves a high mAP@0.5 value of 0.968, indicating good overall performance. The curve also shows that the model

DOI: 10.3844/jcssp.2024.885.897

has a high precision at all recall levels, which means that it is able to correctly identify most of the objects it detects. However, the recall is not as high as the precision, especially at lower recall levels. This suggests that the model may miss some of the smaller or more difficult-to-detect objects which is given in Fig. 5.

Overall, the precision-recall curve shows in Fig.6 demonstrates that the object detection model has good performance on the given dataset. The high mAP@0.5 value and the high precision at all recall levels indicate that the model is able to accurately detect most of the objects. However, the lower recall at lower IoU thresholds suggests that there is still room for improvement in detecting smaller or more challenging objects.

A confidence level of 0.5, the mean Average Precision (mAP) for all classes is 0.968. The best confidence threshold at which the model performs at its best can be found by combining the precision and recall metrics into a single metric called mean average precision. The model performs well generally across all classes, as indicated by the mAP of 0.968 at a confidence level of 0.5. This measure is frequently employed to assess how well object detection models perform.

The training process of the system was monitored by tracking three loss functions: Box loss (Plot 1), DFL loss (Plot 2), and classification loss (cl) (Plot 3), visualized in Figs. 7-8.

Box Loss (Box_Loss)

The difference between the ground-truth and anticipated bounding boxes is measured by this loss function. It is employed to train the model so that it can correctly forecast where items will be found in the pictures. The number of training epochs, or iterations, is shown on the x-axis. The value of the box loss function at each iteration or epoch is shown on the y-axis. The box loss curve exhibits a steady decrease from an initial value of around 0.18 to a minimum of approximately 0.12 by epoch 20. This decline indicates that the model effectively learned to localize objects accurately throughout the training process. Primarily between epochs 20 and 25, slight fluctuations are observed in Table 2.

Classification Loss (CLS_Loss)

The difference between the ground truth and predicted class labels is measured by this loss function. It is employed to educate the model so that it can correctly identify the things in the pictures. The number of training epochs, or iterations, is shown on the x-axis. The value of the classification loss function at each iteration or epoch is shown on the y-axis. The classification loss curve demonstrates a consistent downward trend, starting from around 0.32 and steadily decreasing to reach a minimum value of approximately 0.25 by epoch 20. This continuous improvement signifies that the model's ability to distinguish between different object classes consistently improved throughout the training process. There is a slight increase towards the end, reaching a value of around 0.27 by epoch 2, it shows in Table 3.

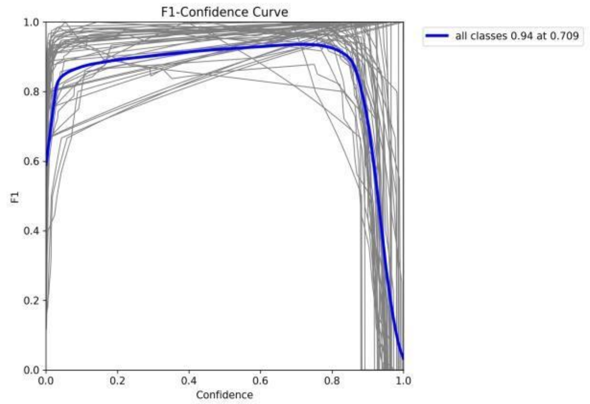

Fig. 3: F1-confidence curve

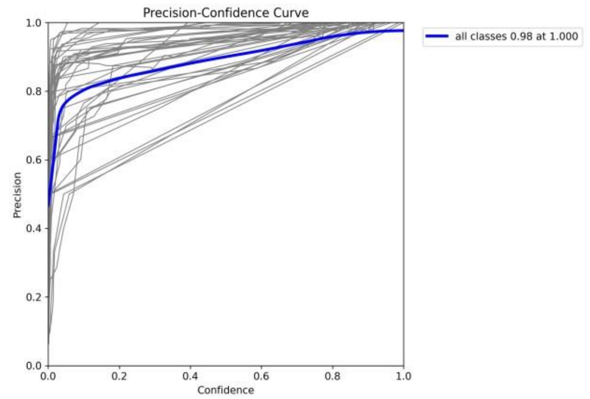

Fig. 4: Precision-confidence curve

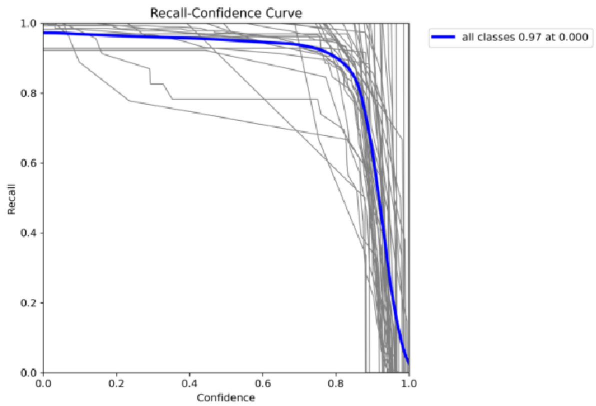

Fig. 5: Recall-confidence curve

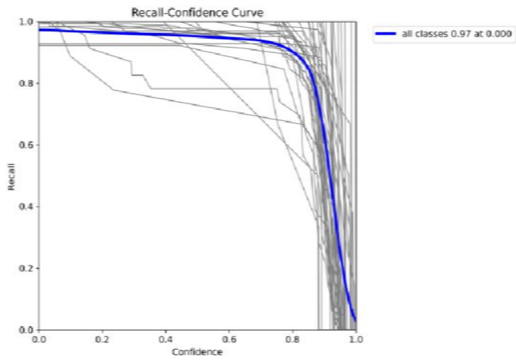

Fig. 6: Precision-recall curve

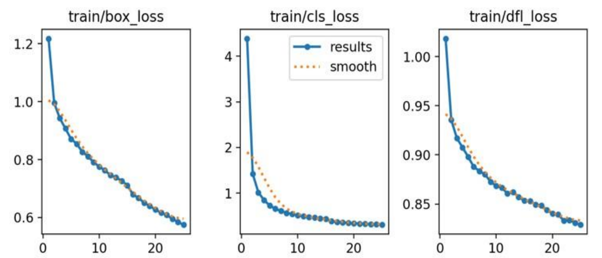

Fig. 7: Train results

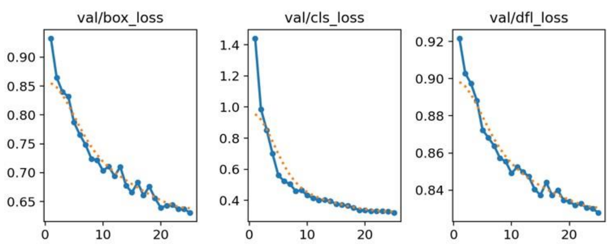

Fig. 8: Validation results

Table 2: Training results

	Loss function		
Epoch	Box loss		Box loss
1	1.22	1	1.22
5	0.87	5	0.87
10	0.78	10	0.78
15	0.71	15	0.71
20	0.63	20	0.63
25	0.57	25	0.57

Table 3: Validation results

	Loss function		
Epoch	Box loss	Classification loss	Distribution focal loss
1	0.93	1.44	0.92
5	0.79	0.56	0.87
10	0.72	0.46	0.85
15	0.68	0.40	0.84
20	0.65	0.34	0.83
25	0.63	0.32	0.83

Distribution Focal Loss (DFL_Loss)

Table 3 explores the Distribution Focal Loss (DFL_loss) is a newly proposed loss function tailored to tackle class imbalance issues encountered in object detection. It builds upon the foundation laid by Focal Loss, incorporating distributional insights to better handle varying densities of examples within each class. Strategically assigning higher weights to challenging instances, enabling the model to focus on learning discriminative features crucial for precise object detection. This augmentation empowers the model to better differentiate between easy and hard examples, thereby enhancing its overall performance in detecting objects accurately. The DFL loss curve follows a similar pattern to the box loss, starting from around 0.35 and gradually decreasing to nearly 0.3 by epoch 20. This trend suggests that the model successfully addressed class imbalance and diverse object shapes during training.

To enhance the model's performance during training, these loss functions are computed and modified at each epoch. Keeping an eye on these loss functions can assist in pinpointing problem areas and enhance the functionality of the model. A model may not be functioning effectively in a particular area if one of its loss functions is consistently higher than the others. In this case, the training configuration or hyperparameters may need to be changed to improve the model's performance.

In Table 2, the box loss values, classification loss values, and distribution focal loss values are demonstrated as observed during the training of the system for the given epoch values.

The testing process of the system was monitored by tracking three loss functions: Box loss (Plot 1), DFL loss (Plot 2) and classification loss (cl) (Plot 3), visualized.

Validation Box Loss (Val/Box_Loss)

The first graph, labeled 'val/box_loss', shows a rapid decrease in loss from the initial epoch to around the fifth epoch, followed by a more gradual decline. The loss stabilizes after approximately 20 epochs, indicating that the model's ability to predict bounding boxes has plateaued. This loss function calculates the percentage of inaccuracy in estimating bounding box coordinates during validation. As a result, the model is encouraged to match the projected bounding boxes with the ground truth boxes. The number of validation epochs, or iterations, is shown on the x-axis.

Validation Classification Loss (Val/Cls_Loss)

The second graph, labeled 'val/cls_loss', depicts a similar trend with a sharp decline in classification loss within the initial epochs, followed by a steady convergence to a lower loss value. This suggests that the model's classification accuracy is improving and stabilizing as training progresses. During the validation phase, this loss function measures the mistake in guessing the object class for each bounding box. It guarantees that the category of the object is correctly identified by the model. The number of validation epochs, or iterations, is shown on the x-axis. The value of the classification loss function at each validation iteration or epoch is shown on the y-axis.

Validation Distribution Focal Loss (Val DFL_Loss)

The third graph, labeled 'val/dfl_loss', also shows a decrease in loss, but the trend is less steep compared to the other two. The 'dfl_loss' metric appears to converge slowly, suggesting that whatever aspect of the model's performance it measures is more challenging to optimize. The number of validation epochs, or iterations, is shown on the x-axis. The value of the distribution focal loss function at each validation iteration or epoch is shown on the y-axis.

These loss functions are computed and updated throughout validation to assess the model's performance on hypothetical data. Keeping an eye on these loss functions can assist in pinpointing problem areas and enhance the functionality of the model.

In Table 3, the box loss values, classification loss values, and distribution focal loss values are demonstrated as observed during the validation of the system for the given epoch values.

Precision (B)

In order to evaluate the model's capacity to prevent false positives, precision measures the percentage of true positives among all positive predictions for a given class (B). In order to compute it, divide the number of True Positive detections (TP) by the number of False Positive detections (FP), or TP/(TP + FP). The graph indicates an initial increase in precision, followed by some fluctuations and eventual stabilization, suggesting that the model is maintaining a high precision rate after a certain number of epochs.

Recall (B)

For a given class (B), recall quantifies the percentage of real positive detections among all of the bounding boxes. It is computed as TP/(TP + FN), where TP is the number of true positive detections and FN is the number of false negative detections. It is also referred to as sensitivity or true positive rate. The recall value increases sharply at the beginning and then plateaus, indicating that the model is consistently identifying a high proportion of the actual positive cases as the training progresses.

mAP50 (B)

Mean average precision for a given class (B) at an Intersection over Union (IoU) criterion of 0.50. It is an indicator of how accurate the model is when simply taking into account "easy" detections. The graph shows a rapid increase to a high mAP50 score, which then levels off, demonstrating that the model achieves a strong performance in detecting objects with a moderate IoU threshold.

mAP50-95 (B)

The mean Average Precision (mAP) for a specific class is calculated by evaluating the precision of the model's detections across various Intersections over Union (IoU) thresholds, typically ranging from 0.50-0.95. This comprehensive analysis provides a detailed understanding of how well the model performs at different levels of detection precision.

The trend observed in the mAP curve for Class B is similar to that of mAP50, which mainly focuses on a single IoU threshold of 0.50. Initially, there's a rapid increase in the mAP as the IoU threshold increases, followed by a stabilization indicating a consistent performance across a range of IoU thresholds. However, it's noteworthy that the final mAP values obtained for Class B are typically lower than mAP50. This difference is expected due to the increased difficulty of achieving high precision across a broader range of IoU thresholds.

These metrics, including mAP and class-specific mAPs, serve as fundamental tools for assessing the efficacy of object detection models. They offer valuable insights into the model's ability to accurately identify objects of interest under various conditions and are essential for making informed decisions about model optimization and deployment.

In Table 4, the precision (B), Recall (B), mAP50 (B), and mAP50-95 (B) values are demonstrated as observed for the given epoch values.

Figure 9, the system accurately detects the school sign with an accuracy of 0.60.

Figure 10, the system accurately detects the stop sign with an accuracy of 0.88.

Figures 11-12, the system accurately detects the signal ahead sign with an accuracy of 0.59.

Table 4: Results

	Metric for evaluation epoch			
	Precision(B)	Recall(B)	mAP50(B)	mAP50-95(B)
1	0.64	0.26	0.32	0.24
	0.83	0.64	0.77	0.62
5	0.83	0.92	0.94	0.78
	0.85	0.94	0.95	0.80
10	0.94	0.94	0.97	0.82
	0.94	0.94	0.97	0.83

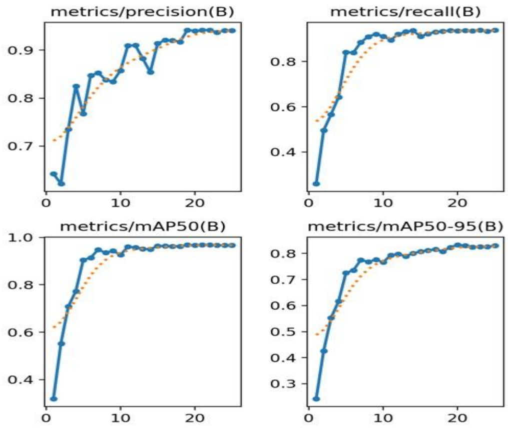

Fig. 9: Metrics for evaluation

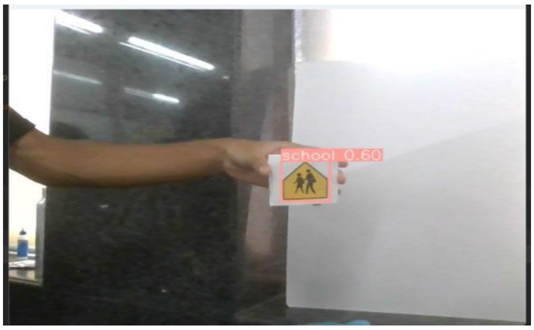

Fig. 10: Detection of school road sign

Fig. 11: Detection of stop road sign

Fig. 12: Detection of signal ahead road sign

Conclusion

This study describes the successful implementation of a real-time traffic sign identification and interpretation system employing text-to-speech synthesis and the You Only Look Once (YOLO) object detection paradigm. The goal of the article was to develop an intelligent system that could identify different traffic signs from a live video stream and speak instructions according to the signs it identified. With the help of the YOLO model, the system was able to detect objects in video frames in real-time, which made it possible to recognize traffic signs without sacrificing system performance. By mapping the model's output to human-readable traffic sign labels, the identified signs were made easier to interpret. In order to guarantee that the user received appropriate and pertinent instructions, conditional statements were utilized to dynamically construct spoken instructions based on the indicators that were observed. Text-to-speech synthesis was integrated into the system to provide audible communication of recognized signs, making it accessible and user-friendly.

However, some of the limitations include the inability to detect partially occluded signs, limited detection under varying lighting conditions, and misclassification of signs with similar structures. The probable cause for misclassification is believed to be the smaller model size and insufficient training data under various real-world conditions. The future scope would be to better deal with cluttered scenes and partially occluded objects and use larger YOLOv8 models such as YOLOv8 and YOLOv8l instead of YOLOv8 (which is limited to 11.2 million parameters), on better hardware to capture more detail of the road signs. The use of datasets such as Tsinghua-Tencent 100 k is recommended since it is much larger and contain images with large variations in illuminance and weather conditions.

Acknowledgment

We acknowledge Dr. Hari Vasudevan, Principal, D J Sanghavi College of Engineering for their valuable support which motivates us to research in this field.

Funding Information

We have not received any funds from any organization.

Author’s Contributions

Lazeen Manasia: Led the exploration and selection of the primary road sign dataset, emphasizing real-world scenarios. Implemented and supervised the YOLOv8 model training, ensuring proficient detection of road signs.

Harshita Singhvi: Played a pivotal role in dataset research, ensuring the selection of a diverse and representative dataset for road sign images. Led the testing phase, validating the YOLOv8 model's accuracy and reliability in real-world driving scenarios.

Gaurav Mehta: Contributed insights into dataset requirements, influencing the decision-making process for effective model training. Fine-tuned training parameters and optimized the YOLOv8 model for real-time traffic sign detection.

Mufaddal Bharmal: Contributed to the technical aspects of the dataset training process, ensuring the YOLOv8 model's proficiency in road sign detection. Collaborated in algorithm refinement, enhancing the model's adaptability to varying driving conditions.

Nilesh Rathod: Supervised and directed the data augmentation process, ensuring that the generated data were of high quality and aligned with the research objectives. They provided guidance on the selection of augmentation techniques and parameters to enhance the dataset used for analysis.

Aruna Gawade: Performed a thorough comparative analysis of the collected data, examining different variables, trends, and patterns; utilizing statistical methods and analytical tools to compare datasets, identify similarities and differences, and draw meaningful insights from the data.

Angelin Florence: Pioneered the development of innovative algorithms and methodologies for processing and analyzing the data. Implementing cutting-edge techniques to extract valuable information from complex datasets, contributes significantly to the advancement of data analysis in the research field.

Ethics

We have followed the research ethics like scientific integrity, human rights and dignity, and collaboration between science and society. These principles make sure that participation in studies is voluntary, informed, and safe for research subjects.

References

Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., Nazir, S., & Haq, A. ul. (2020). Object Detection through Modified YOLO Neural Network. *Scientific Programming*, *2020*, 8403262. https://doi.org/10.1155/2020/8403262

Bai, W., Zhao, J., Dai, C., Zhang, H., Zhao, L., Ji, Z., & Ganchev, I. (2023). Two Novel Models for Traffic Sign Detection Based on YOLOv5s. *Axioms*, *12*(2), 160. https://doi.org/10.3390/axioms12020160

De Pra, Y, & Fontana, F. (2020). Time Sound in Python. *Applied Sciences*, *10*(12), 4214. https://doi.org/10.3390/app10124214

Dhawan, K., Perumal R., S., & Nadesh R. K. (2023). Identification of traffic signs for advanced driving assistance systems in smart cities using deep learning. *Multimedia Tools and Applications*, *82*, 26465–26480. https://doi.org/10.1007/s11042-023-14823-1

Flores-Calero, M., Astudillo, C. A., Guevara, D., Maza, J., Lita, B. S., Defaz, B., Ante, J. S., Zabala-Blanco, D., & Armingol Moreno, J. M. (2024). Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review. *Mathematics*, *12*(2), 297. https://doi.org/10.3390/math12020297

Gašparović, B., Mauša, G., Rukavina, J., & Lerga, J. (2023). Evaluating YOLOV5, YOLOV6, YOLOV7, and YOLOV8 in Underwater Environment: Is There Real Improvement? *2023 8th International Conference on Smart and Sustainable Technologies (SpliTech)*, 1–4. https://doi.org/10.23919/splitech58164.2023.10193505

Hugar, Dr. J. G., Naseer, M. M., Waris, A., & Khan, M. A. (2021). Road Traffic Accident Research in India: A Scientometric Study from 1977-2020. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3893062

Liu, Y., Shi, G., Li, Y., & Zhao, Z. (2022). M-YOLO: Traffic Sign Detection Algorithm Applicable to Complex Scenarios. *Symmetry*, *14*, 952. https://doi.org/10.3390/sym14050952

Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., & Chen, H. (2023). DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor. *Electronics*, *12*(10), 2323. https://doi.org/10.3390/electronics12102323

Manawadu, M., & Wijenayake, U. (2024). Voice-Assisted Real-Time Traffic Sign Recognition System Using Convolutional Neural Network. *ArXiv*. https://doi.org/10.48550/arXiv.2404.07807

Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258. https://doi.org/10.1016/j.array.2022.100258

Rathod, N., & Wankhade, S. B. (2020). Improving Extreme Learning Machine Algorithm Through Optimization Technique. In H. Vasudevan, A. Michalas, N. Shekokar, & M. Narvekar (Eds.), *Advanced Computing Technologies and Applications* (pp. 157–163). Springer Singapore. https://doi.org/10.1007/978-981-15-3242-9_16

Sahithi, A., Teja, B. S., Shastry, C. V., Venugopal, C., & Rajyalakshmi, CH. (2023). Enhancing Object Detection and Tracking from Surveillance Video Camera Using YOLOv8. *2023 International Conference on Recent Advances in Information Technology for Sustainable Development (ICRAIS)*, 228–233. https://doi.org/10.1109/icrais59684.2023.10367122

Saxena, S., Dey, S., Shah, M., & Gupta, S. (2024). Traffic sign detection in unconstrained environment using improved YOLOv4. *Expert Systems with Applications*, *238*, 121836. https://doi.org/10.1016/j.eswa.2023.121836

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*, 60. https://doi.org/10.1186/s40537-019-0197-0

Wan, H., Gao, L., Su, M., You, Q., Qu, H., & Sun, Q. (2021). A Novel Neural Network Model for Traffic Sign Detection and Recognition under Extreme Conditions. *Journal of Sensors*, *2021*, 9984787. https://doi.org/10.1155/2021/9984787

Wang, J., Chen, Y., Dong, Z., & Gao, M. (2023). Improved YOLOv5 network for real-time multi-scale traffic sign detection. *Neural Computing and Applications*, *35*(10), 7853–7865. https://doi.org/10.1007/s00521-022-08077-5

Wu, X., & Cao, H. (2022). Traffic Sign Detection Algorithm Based On Improved YOLOv4. *Journal of Physics: Conference Series*, *2258*(1), 012009. https://doi.org/10.1088/1742-6596/2258/1/012009

Yao, Y., Han, L., Du, C., Xu, X., & Jiang, X. (2022). Traffic sign detection algorithm based on improved YOLOv4-Tiny. *Signal Processing: Image Communication*, *107*, 116783. https://doi.org/10.1016/j.image.2022.116783

Youssouf, N. (2022). Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4. *Heliyon*, *8*(12), e11792. https://doi.org/10.1016/j.heliyon.2022.e11792

Zhang, H., & Zhao, J. (2022). Traffic Sign Detection and Recognition Based on Deep Learning. *Engineering Letters*, *30*(2).

Zhang, H., Qin, L., Li, J., Guo, Y., Zhou, Y., Zhang, J., & Xu, Z. (2020). Real-Time Detection Method for Small Traffic Signs Based on YOLOV3. *IEEE Access*, *8*, 64145–64156. https://doi.org/10.1109/access.2020.2984554

Zhang, L. J., Fang, J. J., Liu, Y. X., Feng Le, H., Rao, Z. Q., & Zhao, J. X. (2024). CR-YOLOv8: Multiscale Object Detection in Traffic Sign Images. *IEEE Access*, *12*, pp. 219–228. https://doi.org/10.1109/access.2023.3347352

Zhang, X. (2023). Traffic Sign Detection Based on YOLOv3. *2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA)*, pp. 1044–1048. https://doi.org/10.1109/icpeca56706.2023.10075795

Zhu, Y., & Yan, W. Q. (2022). Traffic sign recognition based on deep learning. *Multimedia Tools and Applications*, *81*, pp. 17779–17791. https://doi.org/10.1007/s11042-022-12163-0