www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor: 5.8[illegible]

AI Drive Assist: Enhancing Road Safety with Intelligent Sign Detection and Alerts

Mr.K.Kishan Kumar
Assistant Professor, Department of ECE,
Malla Reddy College of Engineering for Women.,
Maisammaguda., Medchal., TS, India (@gmail.com

Abstract: According to the World Health Organization, road accidents are a major cause of death globally, taking the lives of around 1.35 million people each year. Given this alarming number, it is more important than ever to use cutting-edge AI solutions to address road safety issues. The practical consequences for improving road safety and advancing artificial intelligence technologies in the automotive sector are highlighted in this study paper. The "AI-drive assist," the suggested model, examines how well an AI-powered driving aid system might increase traffic safety. The driving aid technology helps drivers better understand road conditions and promotes the adoption of safer driving practices by giving them real-time auditory alerts. In order to effectively extract pertinent information from input images, the methodology uses the You Only Look Once (YOLO)v8 model within a ResNet-50 CNN framework. After a thorough examination, the system's remarkable 94% precision-recall rate in recognizing different road signs suggests that it has the ability to improve driver awareness and encourage adherence to traffic laws. To further increase accuracy and resilience, data augmentation techniques are used to diversify the training dataset. The study's conclusions highlight the important role AI technologies play in encouraging safer driving habits. All things considered, this study shows the real advantages of incorporating AI technologies into driver assistance systems and adds to the continuing conversation about enhancing road safety.

Keywords: artificial intelligence, CNN, computer science, and YOLO

Introduction

Despite substantial technological developments, human error leading to missed road signs remains a key problem to road safety. Speeding was the primary cause of 71% of accidents in 2019 and was responsible for many fatalities and injuries, according to data on traffic accidents. Together, lane indiscipline and balance violations—such as driving while intoxicated and using a cell phone—accounted for 11% of collisions and 14% of fatalities. Furthermore, distracted driving is still a problem, particularly for younger drivers, and it can be dangerous at any time of day. To increase road safety, these factors must be addressed with stronger enforcement methods (Hugar et al., 2021).

Attempts to tackle the problems caused by inattentive driving, lane indiscipline, and excessive speeding must be supplemented by technological developments in road safety. Although there have been significant developments in road safety technology, enduring shortcomings in conventional automotive systems underscore the pressing need for innovation. (Dhawan and others, 2023) There are serious safety issues since current technology frequently fail to quickly provide drivers with critical road sign information. Furthermore, there is still a lack of adequate Artificial Intelligence (AI) integration, which makes it difficult to immediately understand traffic signals and may result in driver errors. These drawbacks are mostly caused by insufficient processing power and a slow response to changing driving conditions. There are two main restrictions on current technology:

1. A deficiency in powerful computer capabilities

2. A limited ability to adapt to rapidly changing driving conditions

By overcoming the constraints imposed by a lack of high-performance computer capabilities and a limited ability to adapt to rapidly changing driving scenarios, AI-drive assist aims to reduce the risks associated with missing traffic signals caused by human error. The AI-drive assist system is a ground-breaking solution that has the potential to completely transform road safety in answer to the research challenge. This system claims to solve significant flaws in current infrastructure by smoothly incorporating cutting-edge AI technologies, such as the well-known You Only Look Once (YOLO)v8 idea (Flores-Calero et al., 2024). Known for its quick object recognition, the YOLOv8 model provides accurate road sign detection, which is especially important when driving at high speeds. Additionally, data augmentation approaches have been used to balance class distributions and provide equal representation across different sign types in order to improve the robustness of object detection.

The AI-drive assist system goes beyond simple identification with a novel function that transforms recognized traffic signs into voice alerts that are sent straight to the driver (Manawadu and Wijenayake, 2024). This proactive strategy improves overall safety by reducing distractions and providing the driver with relevant road information. The smooth integration of cutting-edge AI technology with real-time feedback mechanisms is essential to this system since it guarantees that the driver receives crucial information in a timely and correct manner.

The AI-drive assist system has several advantages (Zhu and Yan, 2022). It has the flexibility to adjust to quickly shifting road conditions in addition to giving drivers timely information. By improving the intelligence of the transportation system, efficiency is maximized and increased safety is guaranteed. In order to improve driving safety and convenience, proactive warnings about possible risks and changing road conditions can be seamlessly integrated into signage that now uses sophisticated decision-making algorithms (Wan et al., 2021).

The AI-drive assist system will be essential to improving road safety and efficiency in the future and has broad applications across the whole automotive sector. The system's seamless integration of state-of-the-art AI technology enables a revolutionary change in traffic safety, establishing a standard for the good of society as a whole. It creates the framework for a future in which technology and transportation work together harmoniously, bringing about a new era of road safety by purposefully incorporating state-of-the-art AI developments. This revolutionary approach demonstrates the hidden potential of artificial intelligence (AI) to build secretly safer and more efficient roadways, opening the door to a future where technology and transportation coexist harmoniously for the secret benefit of all.

Review of the Literature

The 2022 research paper "M-YOLO: Traffic sign detection algorithm applicable to complex scenarios" by Liu, Y.; Shi, G.; Li, Y.; Zhao, Z., published in Symmetry, delves into the topic of traffic sign detection in challenging situations. The study focuses on recognizing 11 typical traffic signs across Europe using the You Only Look Once (YOLO)v3 algorithm. The dataset used in the article was obtained from Osijek front-view camera footage. The photographs in the collection show a variety of weather conditions, including overcast, sunny, rainy, and nighttime scenes. The collection consists of 28 film sequences that produced 5567 images with 6751 captioned traffic signs. Surprisingly, the suggested M-YOLO approach performs really well on this enormous dataset. Road safety and driver awareness are improved by this study's efficient recognition and warning of common European traffic signs (Liu et al., 2022).

Yao et al.'s 2022 research work, "traffic sign detection algorithm based on improved YOLOv4-Tiny," presents an improved technique for YOLOv4 tiny algorithm-based traffic sign recognition. In order to overcome YOLOv4-Tiny's limitations, the article, which is available on Science Direct, uses Receptive Field Blocks (RFB) and an Adaptive Feature Pyramid Network (AFPN) to enhance feature fusion and extraction. The improved precision, recall, map, and competitive speed of these improvements in traffic sign detection are evidenced by the analysis of the CCTSDB and GTSDB datasets (Yao et al., 2022).

A real-time traffic sign identification method based on YOLOv3 was presented by Zhang et al. (2020). The work, which focuses on novel methods for recognizing tiny traffic signs and was published in IEEE Access, achieves impressive results in terms of precision, recall, and map metrics. This study shows the potential of YOLOv3 in this particular scenario and aids in the development of real-time small-sign identification systems (Zhang et al., 2020).

In their research article "Traffic Sign Detection Algorithm Based on Improved YOLOv4," published in 2022, Wu and Cao present an improved YOLOv4-based traffic sign recognition system. The Journal of Physics: Conference series published the paper. The project improves self-driving cars' recognition of traffic signs by utilizing YOLOv4's effective real-time object detection capabilities. The remarkable results of the two models that were trained—one using the GTSDB dataset and the other using a bespoke dataset—proved the efficacy of their methodology. The models achieved a Mean Average Precision (MAP) of 92% on their own custom dataset and 94% on the German Traffic Sign Detection Benchmark (GTSDB) (Wu and Cao, 2022).

The study "traffic sign detection based on YOLOv3" by

Zhang X. focuses on a more effective variant of YOLOv4 for recognizing traffic signs in autonomous driving cars. This method reduces the amount of model parameters by substituting MobileNetV2 for YOLOv4's feature extraction network in order to maximize model efficiency and speed detection. Attention and residual structures are added to improve the gradient management and feature extraction process. Therefore, a more compact model is obtained, which improves detection speed by 58.8% and decreases it by 2.5%, while also outperforming YOLOv4 in terms of overall accuracy (Zhang, 2023).

Bai et al. (2023) provide two new YOLOv5-based models for traffic sign detection in their paper published in Axioms. The article helps self-driving cars recognize traffic signs by using YOLOv4, a quick and real-time object recognition algorithm. With a mean average precision (map) of 94% on the GTSDB and 92% on their dataset, two models that were trained on both the GTSDB and a custom dataset demonstrate remarkable performance. This illustrates how well their method works to improve autonomous cars' ability to detect traffic signs (Bai et al., 2023).

A lightweight traffic sign detection system based on YOLOv4 is presented in the paper "traffic sign detection in an unconstrained environment using improved YOLOv4" by Saxena et al. (2024). The technique uses MobileNetv3 and depth-wise separable convolution to increase efficiency and reduce the number of parameters. The addition of SPP modules to the feature pyramid and improvements to the MobileNetv3 network are further developments. In comparison to YOLOv4, the system performs better at recognizing traffic signs, according to the German Traffic Sign Detection Benchmark (GTSDB). The model decreases the number of parameters by 197 million, improves processing time by 25%, and achieves a 1.7% increase in Mean Average accuracy (MAP) (Saxena et al., 2024).

The process of recognizing and categorizing traffic signs is the focus of a 2022 study by New York researchers that was published in the journal Heliyon. This is accomplished by using sophisticated computer vision models, particularly Faster R-CNN and YOLOv4. The study uses CSPDarknet53 to create a modified model based on YOLOv4 for reliable and accurate traffic sign detection in order to overcome problems related to small signs. Through the use of picture augmentation techniques, data preprocessing techniques, and consideration of low-light circumstances at night, the model achieves excellent accuracy of 80.71% on the MTSD dataset and 94.80 on the TT-100 K dataset. Cross-data testing on the GTSDB and ITSD datasets demonstrates the model's adaptability, since it accomplishes

higher performance with accuracy rates of 63.64% and 91.74 percent, respectively, in comparison to other models (Youssouf, 2022).

Comparative Evaluation of YOLOv8 with Cutting-Edge Models

Two cutting-edge models that have greatly improved traffic sign detection and each make distinct contributions to the field are presented by Bai et al. (2023) in their publication in Axioms. Road safety and driver awareness are enhanced by M-YOLO's outstanding performance in recognizing common European traffic signs in a variety of weather situations by utilizing YOLOv3. In the CCTSDB and GTSDB datasets, Yao et al. (2022) show that YOLOv4-tiny improvement, which combines receptive field blocks with an adaptive feature pyramid network, has higher precision and recall in traffic sign detection. Similar to this, Zhang et al.'s (2024) YOLOv3-based real-time detection method shows promise for real-time applications by successfully recognizing tiny traffic signals. Wu and Cao's improved YOLOv4 method substantially improves the capability of self-driving cars by demonstrating great precision in real-time object recognition on both standard and custom datasets.

However, Zhang's YOLOv4 optimization with MobileNetV2 shows increased accuracy and efficiency in unmanned driving cars, offering a more portable yet powerful solution. According to Bai et al. (2023), YOLOv5-based models show remarkable performance in supporting autonomous cars, demonstrating the versatility of YOLOv4 for traffic sign recognition. The lightweight YOLOv4-based system developed by Saxena et al. (2024) using MobileNetv3 and depth-wise separable convolution provides improved efficiency and parameter reduction, demonstrating better performance than standard YOLOv4 in traffic sign detection.

By contrast, the use of YOLOv8 offers a compelling improvement in the identification of traffic signs. YOLOv8 offers a unified architecture for improved accuracy and efficiency while incorporating the best features of earlier editions. YOLOv8 streamlines the research pipeline and provides scalability and generalization capabilities by fusing the flexibility of faster R-CNN with the efficiency of YOLOv4. Furthermore, YOLOv8 is a viable option for real-time applications because to its sophisticated approaches, such as CSPDarknet53 and MobileNetV2, which guarantee great accuracy even under difficult circumstances. All things considered, YOLOv8 is a major development in traffic sign identification technology, providing researchers with a strong and adaptable instrument to tackle practical issues in autonomous driving systems and traffic management (Gašparović et al., 2023).

Materials and Methods

The block diagram describing the application's model's entire workflow is shown in Fig. 1 below.

Gathering and Preparing Data

In the initial stages of system development, we used Roboflow's LISA dataset as the main source of traffic sign images. For tasks involving the detection and identification of traffic signs, the Long-term Infrastructure and Short-term Activities (LISA) dataset provides an extensive collection of photos. This dataset, which covered a wide variety of real-world situations, was an invaluable tool for computer vision algorithm development and assessment. The LISA dataset, which included annotations for a variety of traffic signs, including those related to autonomous cars and traffic management, was extremely helpful for system development and assessment.

After careful analysis, we discovered a notable discrepancy in the dataset's class distribution, highlighting the necessity of a deliberate data pretreatment approach. Improving the dataset was the main goal in order to maximize the effectiveness of model training. We used the Albumentations library, a potent image augmentation tool, to do this. This approach made it easier to apply augmentation techniques meant to increase the dataset's diversity and promote better model generalization.

Random cropping to guarantee uniform 250x250 pixel dimensions and random brightness and contrast modifications with a chance of 0.2 were among the augmentation techniques used. Furthermore, a probability of 0.5 was used to integrate horizontal flips. The methodology aims to artificially expand the dataset's richness by purposefully introducing variability through these augmentation strategies, which will improve the model's capacity to generalize across many contexts (Mumuni and Mumuni, 2022).

Additionally, the dataset was divided into discrete subsets to enable efficient system training and assessment. 86% of the dataset, or the greatest fraction, was used to train the image segmentation model. The validation set was a smaller subset (9%), which was used to track model performance during training and avoid overfitting. Lastly, 6% of the sample was set aside as the test set for the last assessment of the model's performance following training.

To guarantee the consistency and dependability of the dataset, additional improvements were implemented in addition to these preprocessing methods. To expedite the training process, all photos were reduced to a consistent resolution of 416, 416 pixels, and auto-orientation was used to standardize image orientation. Additionally, augmentations were carried out on every training sample, adding variables to improve the model's resilience. Notably, to increase the model's resistance to slight errors in object localization, a tiny quantity of noise—up to 1% of pixels—was introduced into bounding boxes.

Our goal in incorporating these preprocessing methods and dataset splits into the methodology was to provide a solid basis for the system's subsequent training and assessment.

Choosing and Training Models

Several object detection architectures, including faster R-CNN, SSD, RetinaNet, EfficientDet, and Mask R-CNN, were thoroughly evaluated throughout the critical Model Selection and Training phase. Each option's distinct benefits and drawbacks were thoroughly considered. Faster R- CNN's multi-stage method made it a popular model, but it had drawbacks, including slower inference than the chosen YOLOv8. Because of its superior speed-accuracy balance—which is crucial for real-time traffic sign detection—YOLOv8 fared better than SSD in real-time object detection. RetinaNet's ability to handle unbalanced classes was not better than YOLOv8's in the target domain. However, YOLO—especially the YOLOv8s variant—was the best choice, even if algorithms like Mask R-CNN and EfficientDet were more effective. In order to seamlessly adjust to the real-time demands of the AI-drive assist system's traffic sign detection, YOLOv8 demonstrated an unmatched speed-accuracy balance (Zhang and Zhao, 2022).

Table 1 demonstrates how the Python programming language was used in conjunction with well-known deep learning frameworks like TensorFlow and PyTorch to compare object detection techniques. According to Lou et al. (2023), these frameworks made it easier to construct a number of cutting-edge algorithms, such as R-CNN, SSD, RetinaNet, EfficientNet, Mask R-CNN, and YOLOv8. As part of the implementation process, each framework's pre-trained models were loaded, and inference was performed on the test dataset. OpenCV was used for image processing tasks and model validation on this dataset, which included a variety of real-world photos. Performance parameters such speed (measured in frames per second, or FPS), mean average precision (mAP), and inference time (measured in milliseconds) were calculated to evaluate the algorithms' efficacy. These measures shed light on each algorithm's effectiveness and precision in identifying items in the pictures (Ahmad et al., 2020).

The analysis's main model, YOLOv8, was trained on the Google Colab platform with the help of the Tesla T4 GPU for increased processing capacity. To guarantee that the model was optimized for dependable outcomes, the training procedure was conducted over 25 epochs.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Table 1: Comparative analysis of algorithms

		Metrics for evaluation	
Algorithms (ms)	Speed (FPS)	Mean Average Precision (mAP) %	Inference time
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

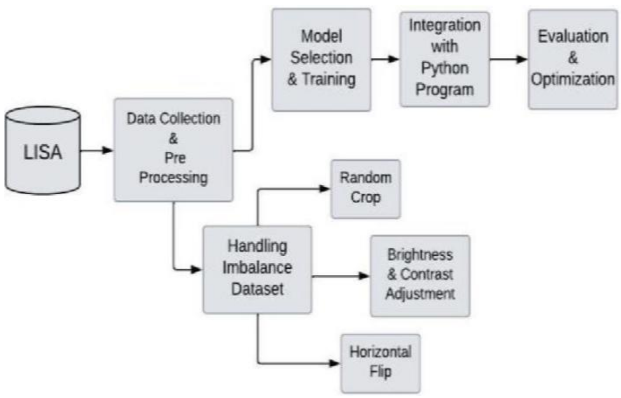

Fig. 1: Block diagram of a methodology of AI drive assist

By satisfying the real-time needs of traffic sign detection, this version is made to react swiftly to the ever-changing road environment. Over the course of 25 training epochs, the model adjusts its parameters via gradient descent and backpropagation. Images scaled to 640 × 640 pixels were used in 25 epochs for the model's training. The detection loss function was utilized as the loss function, while the Adam optimizer was utilized as the optimizer. Random cropping and brightness and contrast tweaks with probabilities of 0.2 and 0.5, respectively, were among the data augmentation strategies. Backpropagation calculates and adjusts parameter gradients to allow for improved prediction capabilities, whereas gradient descent iteratively optimizes parameters to minimize the detection loss function. The effort put into creating a highly precise and effective model for the AI-drive assist system's real-time traffic sign detection is evident in the selection and training stages.

Combining the Python Program

The trained YOLOv8 model converges nicely using a Python-based architecture because of the adaptable OpenCV library. This integration enables real-time processing of camera feeds, and the YOLOv8 model analyzes each gathered frame in-depth to identify traffic signs. The system's responsiveness is further enhanced by employing the pyttsx3 package, which generates audio alerts that are precisely timed to coincide in conjunction with the model's findings (Wang et al., 2023). Important directions and information can be quickly and effectively communicated to the driver through this auditory response, which is tightly linked to the visible traffic signs. By combining these technologies, the Python program produces a dynamic and responsive AI-drive assist system that raises user awareness and significantly enhances driving safety and intelligence (De Pra and Fontana, 2020).

System Function

The AI-drive assist system initially activates the front camera, which is strategically positioned inside the vehicle to continuously capture images of the road environment. This continuous image collection lays the groundwork for real-time monitoring and analysis of the dynamic road environment.

Model Processing for YOLOv8

The gathered photographs undergo a sophisticated processing step using the cutting-edge YOLOv8 model, which is renowned for its efficacy in object recognition. The capacity of YOLOv8 to evaluate the entire image in a single pass is one of its primary differentiators. This capability is crucial for prompt reaction in real-time traffic situations when driving (Zhang et al., 2023).

YOLOv8 Object Detection

Building on its initial processing, YOLOv8 uses the input image to generate a grid of cells and assigns each one the task of estimating bounding boxes and class probability for objects that may be inside its spatial limits. By employing predefined forms that vary dynamically during training, anchor boxes help the machine improve bounding box predictions.

Class Probabilities and Confidence Scores

Inside each grid cell, YOLOv8 generates many bounding boxes, each of which has a class probability and a confidence score. While confidence ratings show how certain the model is that an object exists within a certain bounding box, class probabilities quantify the likelihood that an object will belong to a particular predefined class.

Insufficient Suppression

To enhance the forecasts, a critical post-processing method known as Non-Maximum Suppression (NMS) is applied. Only the most precise and certain predictions remain after redundant or low-confidence bounding boxes are gradually filtered out in this step. Thanks in large part to NMS, the result is not overloaded with overlapping or less confident forecasts.

Mechanism for Triggering Alerts

Following NMS, predictions go through a thresholding procedure that assesses whether the traffic signs it recognizes are significant enough to warrant attention. A voice alarm is activated if the response is affirmative. Its purpose is to promptly inform the motorist of the kind of traffic sign they have noticed and any pertinent

Index in Cosmos
Oct 2021, Volume 11, ISSUE 4
UGC Approved Journal

instructions.

Notification of Drivers

When the thresholding process yields a favorable result, the system promptly alerts the driver. Either the car's audio system or a special alert system will ensure that the driver has the most recent information on the recognized road signs. This open communication significantly improves the driver's situational awareness and overall road safety.

Constant Function

In a continuous cycle, every stage of the operation operates without a hitch. The YOLOv8 model continually scans, decodes, and classifies traffic signs while the front camera continuously takes photographs of the road. The user is encouraged to drive defensively and safely by this continuously active system architecture, which ensures a continuous, real-time awareness of road signs.

Dealing with an Unbalanced Dataset

The methodology used a systematic preprocessing strategy that primarily relied on data augmentation techniques in order to solve the inherent class imbalance in the LISA dataset. An essential part of this strategy was the Albumentations library, which served as a versatile tool to provide much-needed variation, particularly for underrepresented classes. This deliberate choice was taken with the objective of enhancing the overall durability of the dataset and expanding the visibility of underrepresented classes. The decision to just use data augmentation and exclude alternative techniques highlights the commitment to a targeted, effective strategy that prioritizes simplicity and effectiveness. The current approach demonstrates the high efficacy of data augmentation, even though there is potential for further iterations to explore more complex strategies to alleviate class imbalance (Shorten and Khoshgoftaar, 2019). We intentionally draw attention to this approach to provide the foundation for proving its versatility and effectiveness as the primary tactic for achieving class equity in the challenging field of road sign identification. The deliberate incorporation of variability through augmentation not only enhances model generalization but also aligns with the overarching goal of developing a reliable and adaptable traffic sign-detecting system.

Assessment and Enhancement

The rigorous evaluation and optimization phase of the AI-drive assist system employs a multifaceted strategy to guarantee the model's effectiveness and efficiency (Rathod and Wankhade, 2020). Evaluation metrics serve as numerical measurements to assess the model's performance. F1 score, recall, and precision are a few examples of these measurements. While precision indicates the accuracy of positive detections, recall assesses the model's capacity to identify all relevant cases. By balancing recall and precision, the F1 score provides a comprehensive understanding of the model's overall effectiveness.

Continuous optimization of the AI-driven help system is crucial. This iterative approach leverages insights from evaluation metrics by carefully adjusting the model design. Additionally, advanced techniques are being researched to increase the detection powers. To increase the overall accuracy and efficiency of the system, hyperparameters are changed. The AI-drive assist system's dynamic optimization technique allows it to adapt and endure changing traffic situations with unparalleled accuracy and reliability.

Execution

The system's entire implementation is depicted in the architecture diagram that follows, which is found in Figure 2.

A complex approach involving both hardware and software configurations is needed to integrate the AI-drive assist system into a car camera. An advanced camera system with real-time video recording capabilities must be installed in the vehicle environment. The car's embedded computer system must also have the required software stack loaded, which consists of the OpenCV library, the Python runtime environment, and model dependencies.

The pre-trained YOLOv8s model for real-time traffic sign detection is installed on the vehicle computing unit. The OpenCV package is used to process the video in a specially designed Python application that interacts with the live video stream. This application seamlessly integrates the YOLOv8s model to recognize objects in incoming video frames.

After evaluating the predictions, audio alerts are activated if pertinent traffic signs are found. The auditory alert system gives the driver timely and pertinent feedback through the vehicle's audio system (Sahithi et al., 2023).

Road sign processing is always done while the vehicle is moving thanks to the system's never-ending loop operating paradigm. A solid AI-drive assist system that works well with the car camera system and enhances road safety and driver awareness is achieved through rigorous testing procedures, system calibration, and potential user interface integrations.

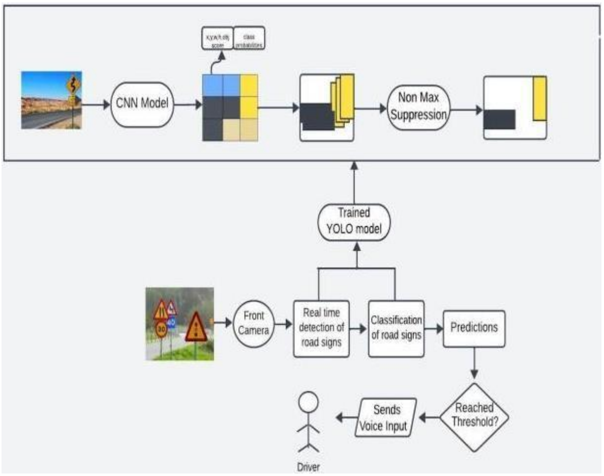

Fig. 2: Architecture diagram

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 presentation that follows presents the findings and shows the application's model's whole workflow.

The F1 confidence curve is a helpful visual tool for binary classification that shows the intricate connection between the F1 score and confidence levels. Practitioners can see how shifting confidence levels affect the ratio of recall to precision by using this graphical representation. The curve provides insight into the dynamics of the model's performance by displaying the F1-score across a range of confidence levels. It assists in determining the optimal cutoff point that reduces false positives (precision) and false

drawbacks (recall), empowering practitioners to make knowledgeable choices about the model's deployment and customization for specific use cases. The x-axis of the F1 confidence curve graph displays the confidence threshold, or the minimum level of confidence required for a detection to be considered a positive prediction. The y-axis in Figure 3 displays the F1 score, which is the harmonic mean of recall and precision. The F1-score for every class is 0.94 when the confidence level is set to 0.709. As a result, the model's predictions are very accurate at a specific confidence level, as indicated by the high F1-score. The curve can be used to identify the confidence value that maximizes the model's overall performance; at a given confidence threshold, a higher F1-score indicates a better balance between precision and recall.

Furthermore, the F1 confidence curve offers insightful information about the trade-off between recall and precision, enabling practitioners to make well-informed decisions regarding the model's performance in practical situations. Stakeholders can adjust the model's threshold to satisfy certain needs and maximize its performance for various applications by examining this curve.

By illustrating the link between precision and fluctuating confidence thresholds in a classification model, a precision-confidence curve provides a visual representation that aids in evaluating how varied confidence levels impact prediction precision. The model's precision for every class is 0.98 when the confidence level is set at 1.00. The x-axis represents the confidence threshold, and the y-axis displays the precision for each class. In this case, the precision for each class is 0.98 when the confidence criterion is set to 1.00. As a result, the model's predictions are highly accurate at a specific confidence level, as indicated by the high precision score. A model is more likely to generate fewer false positive predictions if it has a higher accuracy score. Therefore, selecting a confidence threshold that optimizes the model's overall precision can be aided by the precision confidence curve graph. A precision-recall curve, a popular metric for assessing object identification models' performance, is displayed in Figure 4. For a certain object identification model, the curve shows the trade-off between accuracy (the percentage of true positives among all predicted positives) and recall (the percentage of true positives among all real positives). In this instance, the curve shows how well the model performs across all classes. The mean

Average Precision (mAP) determined at an Intersection over Union (IoU) threshold of 0.5 is represented by the mAP@0.5 number, which is also displayed on the curve. This measure offers a single figure that encapsulates the model's overall performance across all classes and IoU criteria. The figure shows that the model performs well overall, achieving a high mAP@0.5 value of 0.968. Additionally, the curve indicates that the model is capable of accurately identifying the majority of the things it detects, as seen by its high precision across all recall levels. However, particularly at lower recall levels, the precision is higher than the recall. This implies that some of the smaller or more challenging-to-detect objects shown in Fig. 5 may be missed by the model.

Overall, the object detection model performs well on the provided dataset, as seen by the precision-recall curve in Fig. 6. The model can correctly detect the majority of items, as evidenced by the high mAP@0.5 score and the high precision at all recall levels. The reduced recall at lower IoU thresholds, however, indicates that detection of smaller or more difficult items may still be improved.

The mean Average Precision (mAP) for all classes is 0.968 with a confidence level of 0.5. By integrating the precision and recall measures into a single statistic known as mean average precision, it is possible to determine the optimal confidence threshold at which the model operates at its peak performance. The mAP of 0.968 with a confidence level of 0.5 shows that the model does well overall across all classes. This metric is commonly used to evaluate the performance of object detection models.

Three loss functions—Box loss (Plot 1), DFL loss (Plot 2), and classification loss (cl) (Plot 3), which are shown in Figs. 7-8—were tracked during the system's training process.

Box Loss (Box_Loss)

This loss function quantifies the difference between the ground-truth and expected bounding boxes. It is used to train the model to accurately predict the locations of objects in the images. The x-axis displays the quantity of training epochs, or iterations. The y-axis displays the box loss function's value at each iteration or epoch. By epoch 20, the box loss curve shows a consistent decline from its starting value of about 0.18 to a minimum of about 0.12. This decrease suggests that the model successfully acquired the ability to precisely pinpoint objects during the training phase. Table 2 shows minor changes, mostly between epochs 20 and 25.

Loss of Classification (CLS_Loss)

This loss function quantifies the discrepancy between the predicted class labels and the ground truth. It is used to train the model to accurately recognize the objects in the images. The x-axis displays the quantity of training epochs, or iterations. The y-axis displays the classification loss function's value at each iteration or epoch. The categorization loss curve exhibits a steady downward trend, beginning at approximately 0.32 and progressively declining to a minimum.

value by epoch 20 of roughly 0.25. This ongoing development shows that during training, the model's capacity to discriminate between various object classes steadily increased. Table 3 indicates a little increase toward the end, with a value of approximately 0.27 by epoch 2.

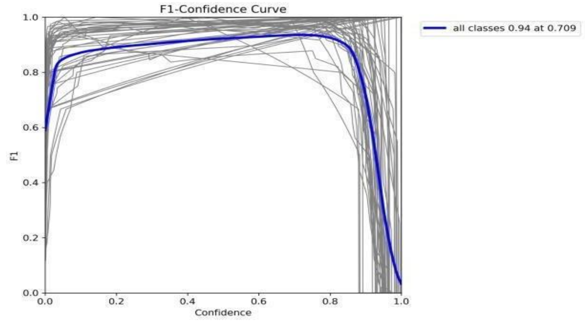

Fig. 3: F1-confidence curve

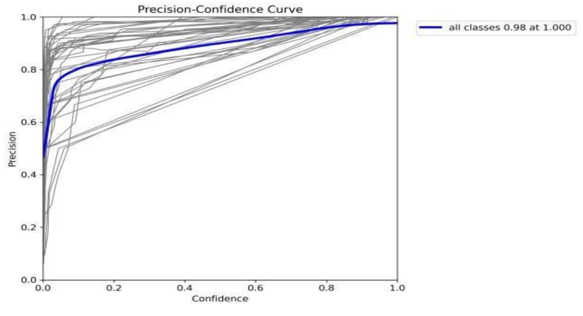

Fig. 4: Precision-confidence curve

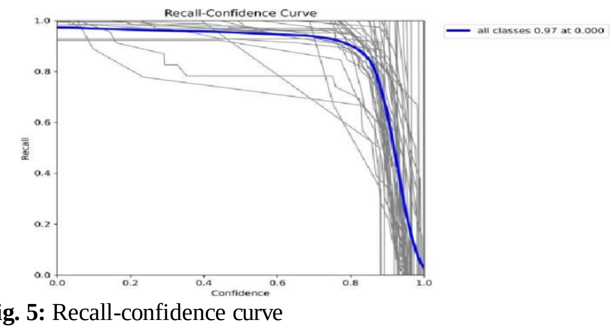

Fig. 5: Recall-confidence curve

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

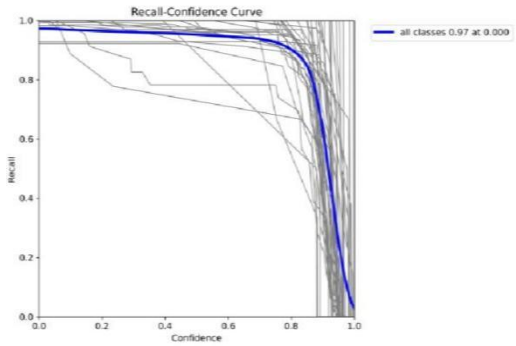

Fig. 6: Precision-recall curve

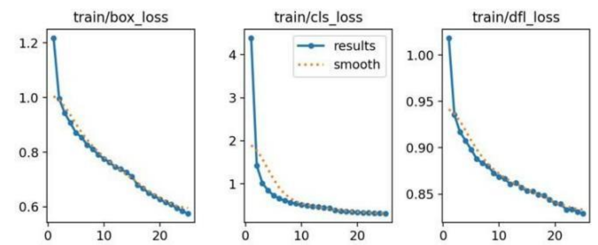

Fig. 7: Train results

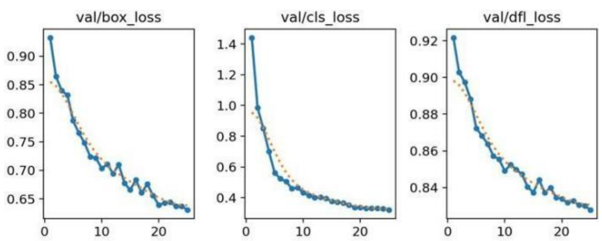

Fig. 8: Validation results

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

epoch	Metric for evaluation			
	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

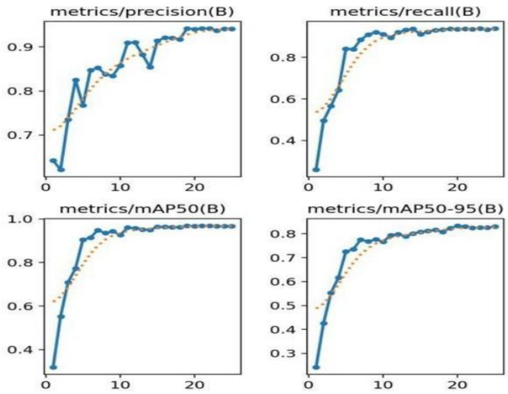

Fig. 9: Metrics for evaluation

Fig. 10: Detection of school road sign

Fig. 11: Detection of stop road sign

Fig. 12: Detection of signal ahead road sign

Conclusion

In this study, the You Only Look Once (YOLO) object detection paradigm and text-to-speech synthesis are used to successfully create a real-time traffic sign identification and interpretation system. The article's objective was to create an intelligent system that could recognize various traffic signs from a live video stream and provide suggestions based on those indications. The system was able to identify traffic signs without compromising system performance by using the YOLO model to detect objects in video frames in real-time. Interpretation of the recognized signs was facilitated by mapping the model's output to human-readable traffic sign labels. Conditional statements were used to dynamically build spoken instructions depending on the observed indicators, ensuring that the user received relevant and appropriate information. The system was made accessible and user-friendly by integrating text-to-speech synthesis to offer aural communication of identified signs.

Nevertheless, some of the drawbacks include misclassifying signs with similar structures, limited recognition in different lighting conditions, and the inability to recognize signs that are partially obscured. The smaller model size and lack of training data under different real-world scenarios are thought to be the most likely causes of misclassification. In order to capture more detail of the road signs, the future scope would be to employ larger YOLOv8 models, like YOLOv8 and YOLOv8l, instead of YOLOv8 (which has a limit of 11.2 million parameters) and better handle congested situations and partially obscured items. It is advised to use datasets like Tsinghua-Tencent 100 k since they are considerably larger and include photos with significant differences in weather and illumination.

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.