Comparison of Different Models for Traffic Signs Under Weather Conditions Using Image Detection and Classification

Amal Alshahrani[1], Leen Alshrif[2], Fatima Bajawi[3], Razan Alqarni[4], Reem Alharthi[5], Haneen Alkurbi[6]
Department of Computer Science and Artificial Intelligence-College of Computing,
Umm Al-Qura University, Makkah, Saudi Arabia

***Abstract*—This study focuses on enhancing the accuracy of traffic sign detection systems for self-driving. With the increasing proliferation of autonomous vehicles, reliable detection and interpretation of traffic signs is crucial for road safety and efficiency. The primary goal of this research was to improve the performance of traffic sign detection, particularly in identifying unfamiliar signs and dealing with adverse weather conditions. We obtained a dataset of 3,480 images from Roboflow and utilized deep learning techniques, including Convolutional Neural Networks (CNNs) and algorithms such as YOLO and the Vision Engineering (VGG) toolkit. Unlike previous studies that focused on a single version of YOLO, this study conducted a comparative analysis of different deep-learning models, including YOLOv5, YOLOv8, and VGG-16. The study results show promising outcomes, with YOLOv5 achieving an accuracy of up to 94.2%, YOLOv8 reaching 95.3% accuracy, and VGG-16 outperforming the other techniques with an impressive 98.68% accuracy. These findings highlight the significant potential for future advancements in traffic sign detection systems, contributing to the ongoing efforts to enhance the safety and efficiency of autonomous driving technologies.**

Keywords—Traffic signs; detection; classification; YOLO; VGG16

I. Introduction

Traffic sign detection plays a crucial role in the development of autonomous driving systems. The ability of these systems to accurately identify and understand road traffic signs is essential for ensuring road safety and efficiency. In recent years, there has been an increasing reliance on autonomous vehicles [1], which makes accurate detection and interpretation of traffic signs even more important. This research aimed to enhance the accuracy of traffic sign detection systems, with a particular focus on detecting unusual traffic signs that may not be widely recognized. Deep learning techniques and algorithms were important in the field of self-driving cars, as they were increasingly being used to detect traffic signs [2]. This led to increased efficiency and safety in self-driving cars. In particular, deep learning algorithms such as YOLOv8, YOLOv5, and VGG-16 were useful in achieving accurate traffic sign detection. These algorithms enabled traffic signs to be recognized and interpreted, thus enhancing the capabilities of autonomous driving systems. By training YOLO models on labeled datasets, algorithms could learn to identify the different shapes, colors, and symbols associated with road signs. The use of these algorithms to detect road signs ensured that vehicles were able to proactively respond to traffic signals, thus improving road safety and efficiency. By leveraging these algorithms, self-driving cars could effectively recognize different types of road signs, including speed limits, stop signs, yield signs, and more. This information was then used to make informed decisions and adapt the vehicle's behavior accordingly.

II. Literature Review

Qian et al. [3] found that the recognition of traffic signs has gained significant importance in applications such as self-driving cars, traffic mapping, and traffic surveillance in recent years. The dataset used is the German Traffic Sign Recognition Benchmark (GTSRB). It is a benchmark dataset specifically designed for traffic sign recognition. The dataset consists of images of traffic signs captured under various conditions, such as different lighting and weather conditions. Each image is labeled with the corresponding traffic sign category. The algorithm used in the paper is a Convolutional Neural Network (CNN). The proposed CNN architecture consists of multiple convolutional layers, activation layers, max pooling layers, fully connected layers, and a softmax layer for classification. The CNN Committee achieved high accuracy, which is 99.46%. The advantages of the proposed approach are that it achieves outstanding performance on the GTSRB dataset, indicating its effectiveness in traffic sign recognition tasks, and The deep learning model (CNN) used in the system has powerful representational learning capabilities, allowing it to extract discriminative features from traffic sign images.

Arcos- García et al. [4] improved traffic sign classification using deep learning in diverse real-world scenarios. They compared different optimization algorithms, including Stochastic Gradient Descent (SGD), SGD with Nesterov momentum (SGD-Nesterov), RMSprop, and Adam, and analyzed the impact of integrating Spatial Transformer Networks (STNs) into CNN. The authors utilized publicly available traffic sign datasets from Germany and Belgium, specifically the German Traffic Sign Recognition Benchmark (GTSRB). Their proposed CNN achieved an impressive recognition rate accuracy of 99.71% in the GTSRB, surpassing previous methods and demonstrating improved memory efficiency.

Morillo et al. [5] provided a comprehensive analysis of state-of-the-art object detection systems, along with several feature extraction tools, for traffic sign detection. The study utilizes the GTSDB dataset, which contains 900 images of traffic lights with various orientations and lighting conditions. The authors fine-tuned object detection models, namely Faster R-CNN, R-FCN, SSD, and YOLO V2, all of which employ the CNN algorithm. The results indicate that Faster R-CNN Inception Resnet V2 achieves the highest accuracy (95.77%), followed by R-FCN Resnet 101 with an accuracy of 95.15%. Additionally, the YOLO V2 and SSD Mobilenet models are highlighted for their competitive performance and lightweight design. Overall, the researchers provide valuable insights for practitioners and researchers working in the field of traffic signal detection.

Rajendran et al. [6] addressed the challenges that traffic sign detection systems using Yolo methods are facing, such as poor accuracy and small object detection issues, unlike the CNN-based methods that provide high accuracy and real-time performance. The authors proposed an approach for traffic sign recognition using YOLOv3 for detection and a CNN-based classifier for classification. The methodology is evaluated using the German Traffic Sign Detection Benchmark (GTSDB) dataset, which contains 600 training images and 300 test images. They also utilized the German Traffic Sign Recognition Benchmark (GTSRB) dataset, which consists of more than 50000 traffic sign images divided into 39209 training images and 12630 test images. The YOLOv3 detector and the CNN-based classifier are implemented using Keras with a TensorFlow backend. The detector performance results were compared with another detector called the faster R-CNN-based method. According to the results, the proposed YOLOv3 outperformed the other detector in terms of accuracy, with an mAP of 92.2% on the GTSDB test set and a frame rate of 10 fps. The CNN-based classifier was evaluated using the GTSRB test set, and it achieved a high accuracy of 99.6%. The future work involves exploring simulation detection and classification using single-stage detectors without the need for an additional traffic sign classification network.

Tabernik et al. [7] addressed the problem of automating traffic signal detection and recognition. They propose a deep learning-based approach using the Mask R-CNN algorithm and present a new dataset called DFG, consisting of 200 classes of traffic lights. The dataset contains a total of 13,000 traffic light instances and 7,000 high-resolution images. The results of their study demonstrate the effectiveness of their approach, as they achieved error rates of less than 3%. This makes it suitable for practical applications in traffic signal inventory management. The researchers present a comprehensive deep learning analysis for dealing with traffic signals with different appearances. Additionally, it provides a challenging dataset that serves as a benchmark. However, one limitation of the paper is that the dataset is limited to the categories chosen by the researchers, and its generalizability to a wider range of traffic signals remains uncertain.

Sichkar et al. [8] presented a holistic model for real-time traffic sign detection and classification, which was important for car vision systems and future autonomous vehicles. The model utilized YOLO version 3 for traffic sign localization and CNN for classification. The detection model was trained on the German Traffic Sign Detection Benchmark (GTSDB) dataset, consisting of 630 training RGB images and 111 validation images. Meanwhile, the classification model was trained on the German Traffic Sign Recognition Benchmark (GTSRB) dataset, which included 66,000 RGB images. The YOLO-based detection model achieved a 97.22% mAP accuracy on four traffic sign categories, while the CNN-based classification model achieved an accuracy of 0.868% on the test dataset.

Zhu et al. [9] explored the application of deep learning techniques, specifically the latest version of YOLOv5, for accurate and efficient traffic sign detection and recognition. The dataset used in the paper is referred to as "our dataset" and was specifically created for Traffic Sign Recognition (TSR) experiments. It contains 2,182 images with eight classes of traffic signs. The algorithm used for TSR in the paper is YOLOv5, which stands for "You Only Look Once" Version 5. It compares the performance of YOLOv5 with another algorithm called SSD (Single Shot MultiBox Detector). YOLOv5 achieved a mean Average Precision (mAP) of 97.70% at a threshold of 0.5 for all classes in terms of TSR. On the other hand, SSD obtained a mAP of 90.14% under the same conditions. It is also mentioned that YOLOv5 outperformed SSD in terms of recognition speed. The advantages of using YOLOv5 for TSR are its improved accuracy compared to previous models like YOLOv3, faster detection speed, and the ability to simultaneously predict bounding box coordinates, target confidence, and class probabilities. As for the disadvantages, the paper does not provide a comprehensive analysis of the limitations or potential drawbacks of YOLOv5 compared to SSD or other TSR algorithms.

Song et al. [10] proposed a deep learning-based algorithm that aims to improve the performance of intelligent vehicles in accurately detecting and recognizing traffic signs. The study utilized the CCTSDB 2021 dataset, which includes 16,356 images with 13,876 prohibitive signs, 4598 warning signs, and 8363 mandatory signs. They improved the algorithm; TSR-YOLO is built upon YOLO (You Only Look Once) and achieved a high detection accuracy of 96.62%. Furthermore, this paper specifically focuses on Chinese traffic signs, making it difficult to assess the generalizability of the algorithm for all types of traffic signs.

Qu et al. [11] proposed an algorithm for traffic sign detection in complex weather conditions based on an improved version of the YOLOv5s model. The study utilized the CCTSDB 2021 dataset, which includes 5268 new traffic scene images. The algorithm employed is PSG-Yolov540, an enhanced version of YOLOv5s, which incorporates improvements such as coordinate attention (CA), an additional prediction head, and the utilization of Alpha-IoU to enhance the original positioning loss CIoU. The algorithm achieves a precision increase of 12.5% and an improved recall rate of 23.9% compared to the original YOLOv5s model, resulting in a precision of 88.1% and a recall rate of 79.8%. However, the paper lacks a thorough discussion of the algorithm's limitations and does not explore potential challenges or failure cases that may arise in real-world scenarios.

Liu et al. [12] introduced an enhanced methodology called ETSR-YOLO, a modified version of the YOLOv5 object detection algorithm. The study introduced two improved C3 modules that aim to suppress background noise interference and enhance the feature extraction capabilities of the network. This paper introduced several enhancements to YOLOv5, including the upgrade of the path aggregation network to capture more contextual information, which improves the detection of traffic signs of varied sizes. Second, we incorporated a coordinated attention method into the backbone network to adaptively improve key features while suppressing noise. Third, the ConvNeXt block increases the network's receptive field and minimizes information loss during feature fusion. Finally, during post-processing, they utilized the WIoU function to improve the predictability and robustness of the model. They utilized the TT100K (Tsinghua-Tencent 100K) dataset, which contains 6634 training images and 1659 test images, and also the CCTSDB2021 (CSUST Chinese Traffic Sign Detection Benchmark 2021) dataset, which contains 14258 training images and 3571 test images. According to the experimental results, ETSR-YOLO increases MAP at 0.5 by 6.6% on the TT100K dataset and 1.9% on the CSUST Chinese Traffic Sign Detection Benchmark 2021 (CCTSDB2021) dataset. Future research aims to enhance the model's performance in complicated road situations and improve computing efficiency for more accurate traffic sign recognition on embedded platforms in vehicles.

One limitation in many studies that train models on traffic signs is that they focus on traffic signs in clear weather and not traffic signs with difficult weather conditions such as rain and fog. This gap in training data can lead to reduced performance and accuracy when the models encounter these difficult weather signs in real-world scenarios. Secondly, many studies do not provide a comprehensive analysis of the limitations or potential drawbacks of YOLOv5 compared to other algorithms. Thirdly, most of the previous studies didn't make a comparison between the different models and their results.

III. Data Collection and Methodology

A. Dataset

It was necessary to have a dataset of images to train deep-learning models. In the context of traffic sign detection and classification, the dataset needed to include various types of traffic signs, including clear and unclear signs, covering most of the possible factors that affect the visibility of traffic signs. After conducting a comprehensive search, an existing dataset was found to meet these specific requirements. Additionally, the available dataset of traffic signs varied in size, encompassing different weather conditions. Also, these types of traffic signs varied in shape, size, and popularity in terms of usage. Images were collected from the Roboflow dataset named "Road Sign Detector Image Dataset Computer Vision Project".

Finding a sufficient number of traffic signs was difficult, as they weren't abundantly available in most dataset sources, and it was challenging to find images in challenging weather conditions due to their limited availability. Extensive searching was conducted on multiple sources to assist in finding a wide range of traffic signs in challenging weather conditions. As a result, 3480 images (3,006 for training, 186 for testing, and 288 for validation) of traffic signs encompassing different and numerous classes were collected from Roboflow, with the aim of ensuring diversity and clarity to assist autonomous vehicles under challenging weather conditions. We did not find these data from any other free or open-source datasets, and had to use the data available on Roboflow. We searched other sources like Kaggle and GitHub, but could not find a ready-to-use dataset that met our requirements, so we could not train our models on different datasets.

B. Methodology

1) Yolo algorithms: The YOLO (You Only Look Once) algorithm is a highly popular and efficient object detection algorithm known for its innovation and speed [15]. YOLO works uniquely by analyzing the entire image in a single pass. Instead of using a proposal-based detection approach, YOLO divides the image into a grid of cells and predicts the bounding boxes and confidence scores for each cell in a single pass. This holistic approach gives YOLO the ability to leverage the overall context of the image to improve the accuracy of its predictions. Additionally, YOLO is characterized by its high response speed, making it suitable for applications that require fast object detection, such as autonomous driving and surveillance.

a) YOLOv8: YOLOv8 is an advanced object detection algorithm in computer vision. It has revolutionized the field by achieving superior detection accuracy and real-time performance using a single end-to-end neural network. YOLOv8 is widely utilized in various applications, such as autonomous driving, surveillance systems, and robotics, where rapid and accurate object detection is crucial. Its impressive performance and versatility have made it a popular choice among researchers and practitioners in the computer vision community.

b) YOLOv5: YOLOv5 is an enhanced version of the YOLO (You Only Look Once) architecture, renowned for its improved efficiency, accuracy, and speed in object detection tasks. It features a streamlined design and incorporates advanced techniques like a novel backbone network and multi-scale prediction strategy. YOLOv5 has gained significant popularity in domains such as autonomous driving, surveillance systems, and robotics, thanks to its balanced trade-off between detection accuracy and computational efficiency. It offers faster inference times while maintaining competitive performance, making it a preferred choice for real-time object detection applications.

2) VGG-16: VGG-16, or Visual Geometry Group 16, is a renowned deep convolutional neural network architecture known for its simplicity and effectiveness in image classification tasks. With 16 layers, including 13 convolutional layers and 3 fully connected layers, VGG-16 captures complex features from input images. Despite newer models surpassing its performance, VGG-16 remains a popular choice for transfer learning due to its strong feature extraction capabilities and publicly available pre-trained weights.

3) Training methodology: The primary objective of this study was to compare the performance of YOLO with previous studies in detecting traffic signs. Additionally, we employed the VGG-16 model to perform the same task, but with the classification of traffic signs. This comparison allowed us to assess and evaluate the effectiveness of both YOLO and VGG-16 in the context of traffic sign detection and classification. The models were trained using a dataset consisting of 3480 images and a set of hyperparameters that included epochs varying from 20 to 45 and batch sizes of 16 for Yolov5s, 16 for Yolov8n, and 16 for VGG-16. Below are the Table I, and II that show the hyperparameter settings.

TABLE I. THE HYPERPARAMETERS SET FOR YOLOV5 AND YOLOV8

Hyperparameters	YOLOv5	YOLOv8
Input image size	640	640
Epochs	45	32
Batch size	16	16
Optimizer	AdamW	AdamW
Initial learning rate	0.01	0.01
Final learning rate	0.01	0.01
Momentum	0.937	0.937
Weight decay	0.0005	0.0005

TABLE II. THE HYPERPARAMETERS SET FOR VGG-16

Hyperparameters	VGG-16
Target size	224
Epochs	20
Batch size	16
learning rate	0.01

4) Training environment: To meet our training requirements for both YOLO and VGG-16, we utilized Google Colab. This platform provided us with the necessary infrastructure to execute the Python code and leverage advanced computational power, including GPUs. By leveraging the capabilities of Google Colab, we were able to efficiently train the models and take advantage of the accelerated processing provided by GPUs. This expedited the training process and enabled us to achieve optimal performance for both YOLO and VGG-16.

IV. RESULTS AND DISCUSSION

In this section, we present the results obtained from training three different models: YOLOv5, YOLOv8, and VGG16. We discuss the performance of each model and provide an analysis of their strengths and areas for improvement.

A. YOLO Object Detection and Classification

YOLO versions 5, and 8 were used for object detection and classification of traffic signs under weather conditions.

YOLOv5, as shown in Table III, achieved mAP50s of 79.3%, 89.6%, 91.3%, 94.1%, and 94.2% over epochs 5, 10, 20, 40, and 45, respectively. The results show the high performance of the model. Furthermore, Fig. 1 displays the results for YOLOv5 at epoch 45. Additionally, Fig. 2 presents performance metrics for YOLOv5, including the precision of 92.37%, the recall rate of 90.85%, the mean average precision at an IoU threshold of 0.5 (mAP50) of 94.23%, and the mean average precision at IoU thresholds ranging from 0.5 to 0.95 (mAP50-95) of 70.28%. Fig. 3 shows the recall confidence curve for all classes 0.97 at 0.000, the precision confidence curve for all classes 1.00 at 0.964, the precision-recall curve for all classes 0.947 mAP 0.5, and the F1-confidence curve for all classes 0.92 at 0.689. Fig. 4 shows the training batch. Fig. 5 shows a sample of the validating batch prediction. Fig. 6 shows a sample of the validating batch label. The total estimated VRAM usage during training for YOLOv5 on a Tesla T4 GPU is approximately 8-12 GB. This includes memory for model parameters, activation maps, gradients, and batch data. The total estimated VRAM usage during inference is around 4-8 GB, primarily due to model parameters and activation maps, with lower requirements as no gradients are stored. YOLOv5 computation time is 61.68 minutes.

TABLE III. YOLOV5 MAP50 OVER 45 EPOCHS

Model	Epoch	mAP50
YOLOv5	5	79.3%
	10	89.6%
	20	91.3%
	40	94.1%
	45	94.2%

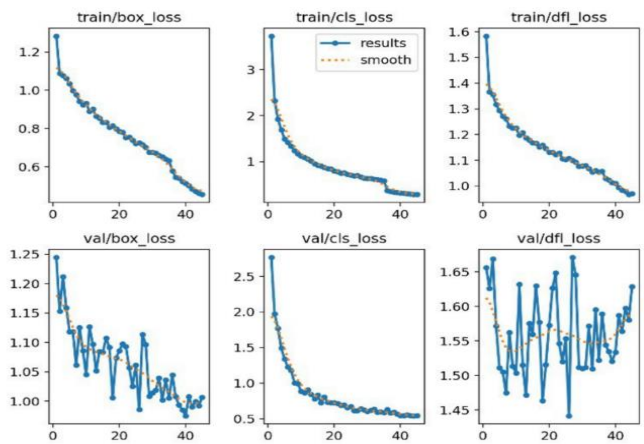

Fig. 1. Results obtained by YOLOv5 at epoch 45.

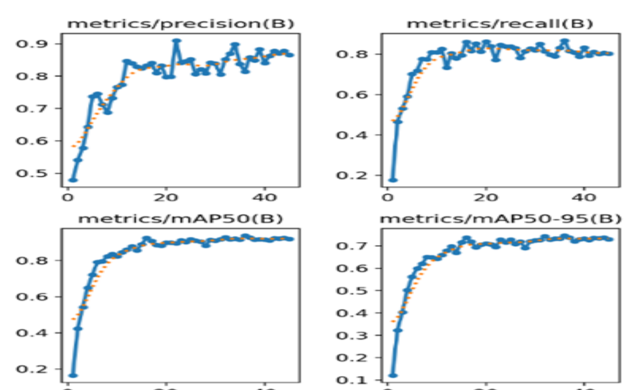

Fig. 2. Performance metrics for YOLOv5.

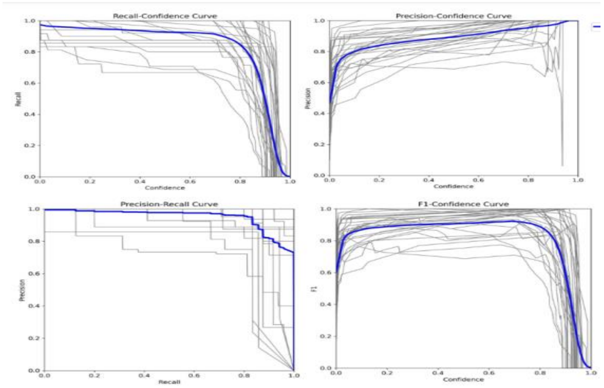

Fig. 3. Confidence curve results for YOLOv5.

Fig. 4. Sample of train_batch for YOLOv5.

Fig. 5. Sample of val_batch_pred for YOLOv5.

Fig. 6. Sample of val_batch_label for YOLOv5.

YOLOv8, as shown in Table IV, achieved mAP50s of 75.6%, 89.3%, 94%, 94.2%, and 95.3% over epochs 5, 10, 20, 25, and 32, respectively. The results show the high performance of the model. Additionally, Fig. 7 displays the results for YOLOv8 at epoch 32. Next, Fig. 8 presents performance metrics for YOLOv8, including the precision of 92.58%, the recall rate of 92.73%, the mean average precision at an IoU threshold of 0.5 (mAP50) of 95.31%, and the mean average precision at IoU thresholds ranging from 0.5 to 0.95 (mAP50-95) of 71.23%. Fig. 9 shows the recall confidence curve for all classes 0.97 at 0.000, the precision confidence curve for all classes 1.00 at 0.979, the precision-recall curve for all classes 0.958 mAP 0.5, and the F1-confidence curve for all classes 0.93 at 0.535. Fig. 10 shows the training batch. Fig. 11 shows a sample of the validating batch prediction. Fig. 12 shows a sample of the validating batch label. The total estimated VRAM usage during training for YOLOv8m on a Tesla T4 GPU is approximately 10-12 GB. This includes the memory for model parameters, activation maps, gradients, and batch data. The total estimated VRAM usage during inference is around 5-8 GB, primarily due to model parameters and activation maps, with lower requirements as no gradients are stored. YOLOv8 computation time is 59.64 minutes.

TABLE IV. YOLOV8 MAP50 OVER 32 EPOCHS

Model	Epoch	mAP50
YOLOv8	5	75.6%
	10	89.3%
	20	94%
	25	94.2%
	32	95.3%

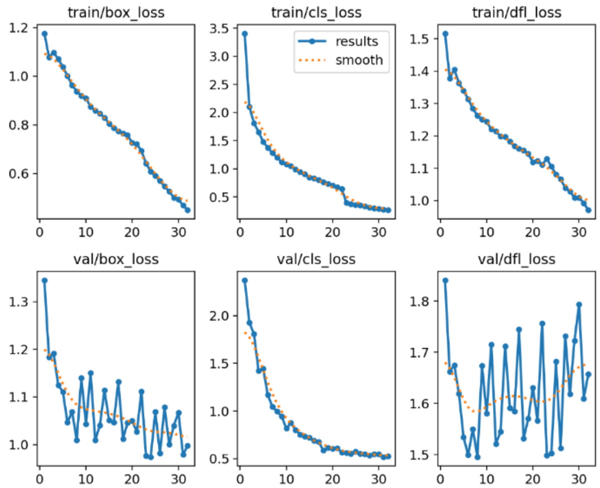

Fig. 7. Results obtained by YOLOv8 at epoch 32.

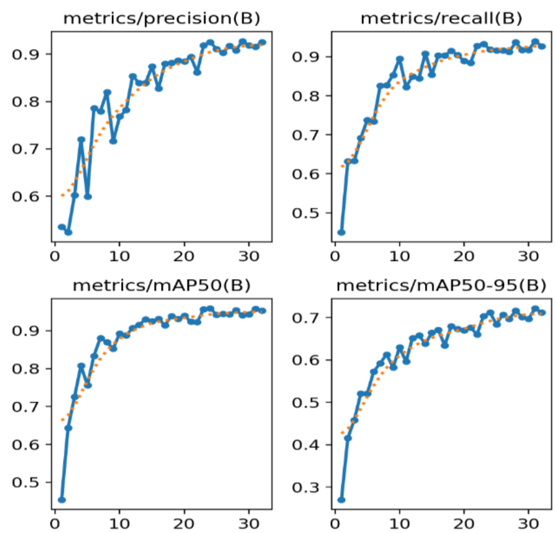

Fig. 8. Performance metrics for YOLOv8.

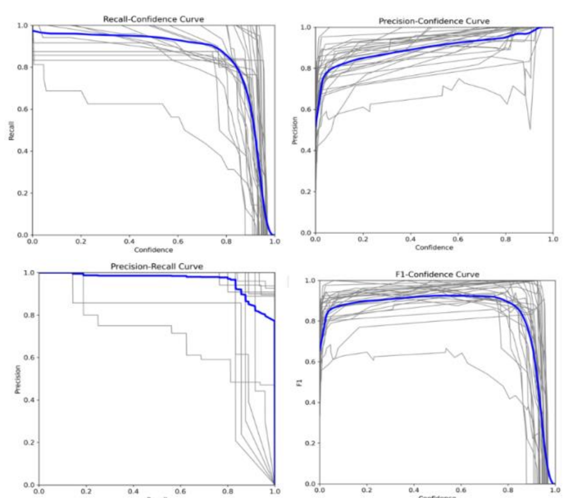

Fig. 9. Confidence curve results for YOLOv5.

Fig. 10. Sample of train_batch for YOLOv8.

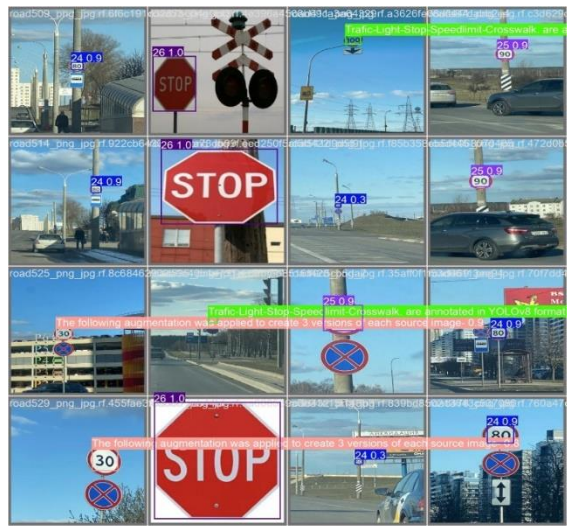

Fig. 11. Sample of val_batch_label for YOLOv8.

B. VGG16

Lastly, the VGG-16 model [13], [14] that we trained exhibited excellent performance, as demonstrated in Table V. It had achieved accuracies of 68%, 96.4%, 99.5%, and 100% for epochs 5, 10, 15, and 20, respectively. These results showcased the model's remarkable ability to classify images with a very high degree of accuracy. Fig. 13 shows the accuracy of the VGG16 model for the training epochs. Fig. 14 shows the loss of the VGG16 model for the training epochs. The accuracy curve steadily increased, reaching 98.68% by the 20th epoch, while the loss curve correspondingly decreased, indicating the model's effective learning and optimization during the training process. The VGG16 model has strong performance metrics, as shown in Fig. 15. It achieves a recall of 98.59%, precision of 100%, and F1 score of 99.29%. As you see in Figure 16, the confusion matrix shows the VGG16 model made 178 false positive predictions, where it incorrectly classified a sample as belonging to a certain class when the true class was different. However, it only made 8 false negative predictions, where it failed to correctly identify a sample's true class.

- False positives: the cases where something is incorrectly identified as positive or present when in reality it is negative or absent.
- False negatives: the cases where the diagnosis fails to identify something as positive or present when in reality it is positive or present.

TABLE V. VGG16 ACCURACY OVER EPOCHS

Model	Epoch	Accuracy
VGG16	5	61.73%
	10	96.65%
	15	97.79%
	20	98.68%

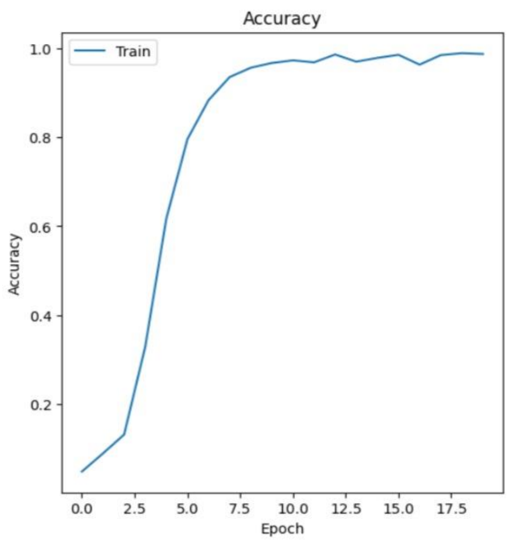

Fig. 12. Accuracy over 20 epochs of VGG16.

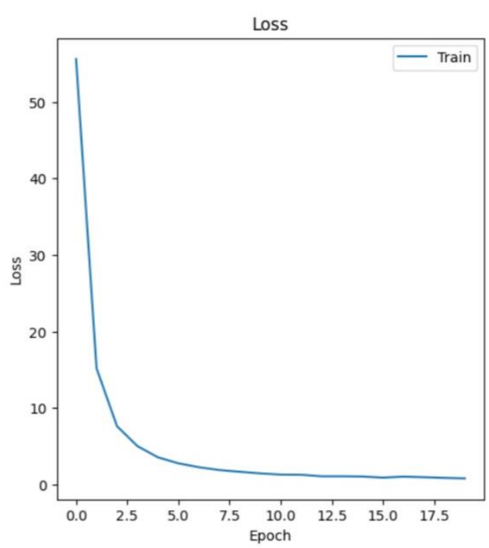

Fig. 13. Loss over 20 epochs of VGG16.

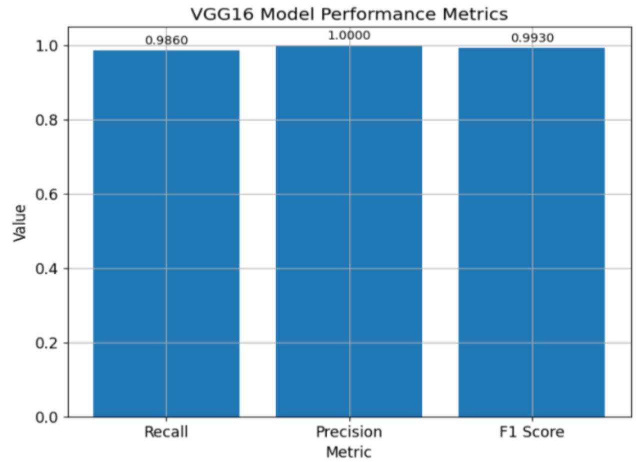

Fig. 14. Performance metrics for the VGG16 model.

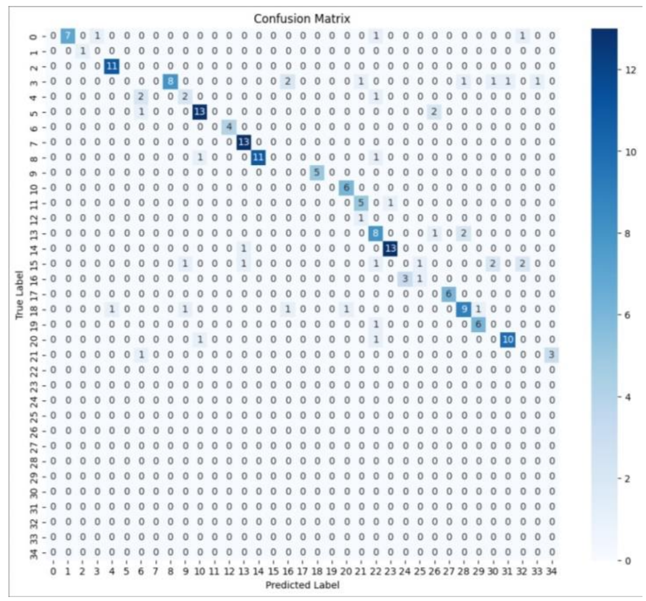

Fig. 15. Confusion matrix for the VGG16 model.

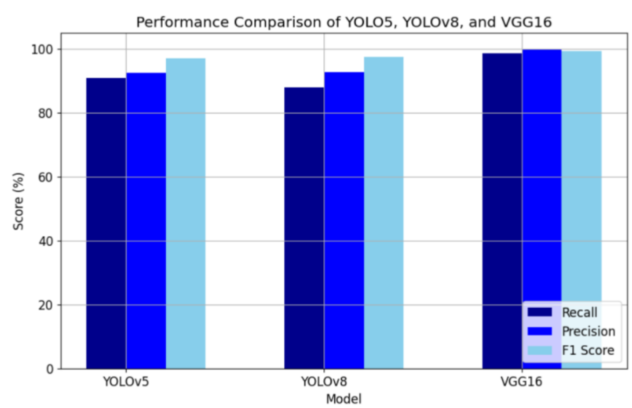

Fig. 16. Performance comparison of YOLO5, YOLOv8, and VGG16.

V. Overall Comparison and Insights

After reviewing the results of the three models, we found that the YOLOv5 model confirmed its fast inference speed, making it suitable for real-time applications such as autonomous driving. Although the accuracy is good, it is slightly lower than the other two models we evaluated. Also, there is a risk of overfitting, as this model may sometimes struggle to generalize to new, unseen data, and the model's performance may degrade in adverse weather conditions or with occluded or partially visible traffic signs. As for YOLOv8, it is one of the latest versions of the YOLO algorithm, and this model is generally characterized by its lightweight and efficiency. However, one of its drawbacks is that maintaining a balance between accuracy and inference speed may be a challenge, as the increasing model complexity can impact real-time performance. And the VGG16 model gave us remarkable results, as the model's ability to achieve an accuracy of 98.68% indicates its high capability to handle the challenges posed by adverse weather conditions and detect unfamiliar traffic signs. However, despite this, VGG16 is a relatively larger and more complex model compared to the YOLOv5 and YOLOv8 models, and the training and fine-tuning of the VGG16 model may be more time-consuming and resource-intensive compared to the YOLO models.

TABLE VI. YOLO and VGG-16 Results

Model	Epoch	Performance measure	for each class (Training)	The performance measure for all
YOLOv5	5	mAP50	79.3%	94.2%
	10		89.6%	
	20		91.3%	
	40		94.1%	
	45		94.2%	
YOLOv8	5	mAP50	75.6%	95.3%
	10		89.3%	
	20		94%	
	25		94.2%	
	32		95.3%	
VGG16	5	Accuracy	61.73%	98.68%
	10		96.65%	
	15		97.79%	
	20		98.68%	

In terms of results, in our evaluation of the YOLOv5, YOLOv8, and VGG16 models, we gained valuable insights. Regarding the model trained using YOLOv8, it achieved largely satisfactory results. It demonstrated a precision of 92.58%, a recall of 88%, a mAP50 of 95.31%, a mAP50-95 of 71.23%, and an F1 score of 97.5%. These metrics indicate its effectiveness in accurately detecting traffic signs under weather conditions. In the case of the model trained using YOLOv5, it achieved a precision of 92.37%, a recall rate of 90.85%, a mAP50 of 94.23%, a mAP50-95 of 70.28%, and an F1 score of 97%. These results indicate its ability to detect and classify objects with a reasonable level of precision and consistency across varying IoU thresholds. On the other hand, the model trained using VGG-16 exhibited a highly satisfactory result, achieving an accuracy of 98.68%, a recall of 98.59%, a precision of 100%, and an F1 score of 99.29%. This showcases its capability to classify images with a high level of accuracy. Table VI presents all the model's results for a clear comparison between them. . In terms of the overall evaluation, the performance comparison as shown in Figure 17, including their recall, precision, and F1 scores, indicates that the VGG-16 model outperforms the YOLOv5 and YOLOv8 models in terms of precision, recall, and F1 score. However, the YOLO models offer faster inference speed, which can be crucial for real-time applications like autonomous driving. The choice of the most suitable model ultimately depends on the specific requirements and trade-offs between accuracy, inference speed, and computational resources for the given application.

Fig. 17. The performance of different models using error rate.

VI. Analyze the Performance of Different Models Using Error Rate

The error rate is a measure that determines the accuracy of a machine learning model in making predictions. It is calculated by finding the percentage of incorrect predictions out of the total predictions. This metric is important for evaluating model performance and identifying which one performs better. In this analysis, we calculated the error rates for three different models: VGG16, YOLO5, and YOLOV8. These error rates were calculated with respect to three key performance metrics: Accuracy, Recall, and F1 Score.

1) Error Rate from Accuracy

$$Accuracy = 100 - model\_accuracy$$

2) Error Rate from Recall

$$Recall = 100 - model\_recall$$

3) Error Rate from the F1 Score

$$F1\ Score = 100 - model\_f1\_score$$

- Performance metrics for VGG16:

  Accuracy: 98.68

  Recall: 98.59

  F1score: 99.29

  - VGG16 Error Rate from Accuracy: 1.32%
  - VGG16 Error Rate from Recall: 1.41%
  - VGG16 Error Rate from F1 Score: 0.71%

- Performance metrics for YOLOv5:

  Accuracy: 94.2

  Recall: 94.2

  F1score: 97.0

  - YOLOv5 Error Rate from Accuracy: 5.80%
  - YOLOv5 Error Rate from Recall: 5.80%
  - YOLOv5 Error Rate from F1 Score: 3.00%

- Performance metrics for YOLOv8:

  Accuracy: 95.3

  Recall: 95.3

  F1score: 97.5

  - YOLOv8 Error Rate from Accuracy: 4.70%
  - YOLOv8 Error Rate from Recall: 4.70%
  - YOLOv8 Error Rate from F1 Score: 2.50%

By comparing these results, we can observe that VGG16 outperforms YOLOv5 and YOLOv8 in terms of all the mentioned performance metrics. It also exhibits the lowest average error rate. Therefore, VGG16 is the best performing model among the three studied.

The main claims of our paper revolve around the comprehensive evaluation of multiple deep learning models for traffic sign detection and classification, with a particular focus on improving accuracy, especially in difficult weather conditions and with unfamiliar signs. Our findings highlight the great potential for further progress in this area, which is critical to enhancing the safety and efficiency of autonomous driving technologies.

VII. Conclusion

The field of traffic sign detection plays a crucial role in advancing autonomous driving systems and ensuring road safety. Many studies on traffic sign detection focus on detecting signs in normal weather conditions rather than challenging weather. This research aims to enhance the accuracy of traffic sign detection systems, particularly in challenging weather conditions such as rain and fog. Deep learning techniques and algorithms, including various versions of YOLO such as YOLOv5, YOLOv8, and VGG16, were employed to achieve precise recognition and interpretation of traffic signs.

The YOLOv5 model achieved a mAP50 of 94.2% after 45 iterations, while the YOLOv8 model demonstrated satisfactory results, with a mAP50 of 95.3% after 32 iterations and 95.2% after 45 iterations. The VGG16 model, which focuses on object classification, displayed high accuracy in training, reaching 98.68% after 15 iterations. Overall, the utilization of deep learning models, such as YOLOv5, YOLOv8, and VGG16, has shown significant potential in improving the accuracy and efficiency of traffic sign detection systems under challenging weather conditions. These models can be trained on labeled datasets to learn and recognize various shapes, colors, and symbols associated with road signs.

The research presented promising results in traffic sign detection under challenging weather conditions, contributing to the advancement of autonomous driving systems and promoting safer and more efficient roadways. Further optimization and refinement of the models can lead to even better performance.

The novelty of this study lies in its holistic evaluation of multiple YOLO versions and the VGG-16 model, which provides a more nuanced understanding of the performance and applicability of these deep learning techniques for traffic sign detection under diverse environmental conditions. This comparative approach represents a significant contribution to the field, as it goes beyond the limitations of previous studies that focused on a single YOLO version, and offers valuable insights for the development of advanced autonomous driving systems.

References

[1] The rise of Autonomous Vehicles: Pros & Cons of self-driving cars: Study (2024) SteinLaw.com. Available at:https://www.steinlaw.com/resources/studies/the-rise-of-autonomous-vehicles/ (Accessed: 14 May 2024).

[2] Katalesanket (2023) Machine learning in self-driving cars,Medium.Availableat:https://medium.com/@katalesanket90/machine-learning-in-self-driving-cars-8b5d1c685d3b (Accessed: 14 May 2024).

[3] Qian, R. Yue, Y. Coenen, F. & Zhang, B. (2016). Traffic Sign Recognition with Convolutional Neural Network Based on Max Pooling Positions. 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). 578-582, DOI: 10.1109/FSKD.2016.7603237

[4] Arcos-García, Á., Álvarez-García, J.A., and Soria-Morillo, L.M. (2018) 'Deep Neural Network for Traffic Sign Recognition Systems: An analysis of spatial transformers and stochastic optimization methods', Neural Networks, 99, pp. 158–165. doi: 10.1016/j.neunet.2018.01.005.

[5] Álvaro Arcos-García, Juan A. Álvarez-García, Luis M. Soria-Morillo. (2018). Evaluation of deep neural networks for traffic sign detection systems, Neurocomputing, 316, 332-344, DOI: https://doi.org/10.1016/j.neucom.2018.08.009

[6] Rajendran, S. P., Shine, L., R., Pradeep, & Vijayaraghavan, S. (2019)" Real-Time Traffic Sign Recognition using YOLOv3 based Detector"10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-7, doi: 10.1109/ICCCNT45670.2019.8944890.

[7] D. Tabernik and D. Skočaj. (2020). Deep Learning for Large-Scale Traffic-Sign Detection and Recognition, IEEE Transactions on Intelligent Transportation Systems, 21, 1427- 1440, DOI:10.1109/TITS.2019.2913588.

[8] Sichkar, V.N. and Kolyubin, S.A. (2020) 'Real time detection and classification of traffic signs based on Yolo Version 3 algorithm', Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 20(3), pp. 418–424. Doi: 10.17586/2226-1494-2020-20-3- 418-424.

[9] Zhu, Y. &Yan, W.Q. (2022) Traffic sign recognition based on Deep Learning - multimedia tools and applications, SpringerLink. 81:17779–17791. DOI: https://doi.org/10.1007/s11042-022-12163-0

[10] Song, W., & Suandi, S. A. (2023). TSR-YOLO: A Chinese Traffic Sign Recognition Algorithm for Intelligent Vehicles in Complex Scenes. Sensors, 23(2), 749. DOI: https://doi.org/10.3390/s23020749

[11] Qu, S., Yang, X., Zhou, H., & Xie, Y. (2023). Improved YOLOv5-based for small traffic sign detection under complex weather. Scientific Reports, 13, 16219. DOI: https://doi.org/10.1038/s41598-023-42753-3

[12] Liu H, Zhou K, Zhang Y, Zhang Y (2023) ETSR-YOLO: An improved multi-scale traffic sign detection algorithm based on YOLOv5. PLoS ONE 18(12): e0295807. https://doi.org/10.1371/journal. pone.0295807

[13] Colab Research. (2023). Google Colab. [online] Available at: vgg16 (1).ipynb - Colab (google.com)

[14] Roboflow. (2023). Road Sign Detector Image Dataset Dataset and Pre-Trained Model by college. [online] Available at: https://universe.roboflow.com/college-g19gz/road-sign-detector-image-dataset/dataset/1/images?split=train .

[15] Redmon, J. et al. (2016) 'You only look once: Unified, real-time object detection', 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Preprint]. doi:10.1109/cvpr.2016.91.