Volume 4 Issue IX, September 2025, No. 73, pp. 953-961
Submitted 2/8/2025; Final peer review 8/9/2025
Online Publication 12/9/2025
Available Online at http://www.ijortacs.com

A REAL-TIME TRAFFIC SIGN DETECTION SYSTEM USING A TRANSFER LEARNING APPROACH

[1*]Edet Offiong E., [1]Oleka Chioa V., [2]Bosede Olawale J.
[1,2]Department of Computer Engineering, Enugu State University of Science and Technology
Email: [1*]eyooffiong76@yahoo.com, [1]chioma.oleka@esut.edu.ng, [2]walebose@gmail.com,

Abstract

This study presents the design and implementation of a real-time traffic sign detection system using a transfer learning approach based on the YOLOv8s model. The system was constructed in a manner that it identified two important road signs which included checkpoint ahead and cattle crossing in diverse environmental conditions including daytime, night and weather conditions. The data set of 640 traffic sign images labelled by the Kaggle repository had been pre-processed with resizing, annotation, normalising, and converting to YOLO format before a 70-20-10 train-test-validation split. YOLOv8s is its architecture based on a CSPDarkNet53 backbone, Efficient Layer Aggregation Network (ELAN), Feature Pyramid Network (FPN), Path Aggregation Network (PAN), and decoupled detection head that was additionally trained using Bayesian hyper-parameter optimization to improve accuracy and decrease overfitting. In the experimental results, the system realised incremental performance gains over the training epochs, with the steepest performance improvements realised between epochs 10 and 30. A model achieved a precision of 0.95, a recall of 0.92 and a mean Average Precision (mAP5095) of 0.83 at the final epoch (epoch 50). Performance analysis in terms of classes showed that both category of traffic signs were identified with a high precision with the checkpoint ahead sign slightly doing better than the cattle crossing sign. These findings represent the efficiency, precision, and the computing power of the YOLOv8s model, which proves that this model is highly applicable in the framework of real-time driver assistance and smart traffic surveillance applications. This paper concludes that the presented system is a credible solution that can be used to detect traffic signs and implement it in a real-life road situation.

Keywords: Traffic Sign; YOLOv8; Agile, Nigeria; Road Accident; Transportation Management.

1. INTRODUCTION

Traffic sign presents crucial data to drivers about managing speed limits in order to automate driving system and avoid accident. This is because, in the case of autonomous cars and human drivers, proper recognition and interpretation of traffic signs will ensure safety and compliance with the traffic law (Piralkar, 2021). Although useful, the classic computer vision techniques often do not reflect the richness and variety of the real-world driving conditions (Gan et al., 2020). Lighting, weather, and sign designs, among other factors in various regions may significantly influence the effectiveness of the traditional algorithms (Lahare et al., 2019).

CNNs in particular are potent deep learning tools that have become a possible solution to these issues (Shivayogi et al., 2023). CNNs can automatically recognise and classify traffic signs with a high level of accuracy by extracting hierarchical characteristics in large datasets of traffic signs (Sinh and Malik, 2022). The ability to generalise in different settings makes CNNs highly useful in real-life scenarios where signs on the roads can be distorted or partially covered, or have different lighting conditions due to the potential to generalise (Konkyana et al., 2023; Triki et al., 2023). Moreover, as it allows identifying signs in various conditions and enabling its accurate recognition, the implementation of deep learning in autonomous vehicles and Advanced Driver Assistance Systems (ADAS) will offer more reliability and safety to the given systems (Zhu and Yan, 2022).

Road accidents remain the number 1 cause of deaths in the world, and human factor, unfavourable weather conditions, and mechanical issues are the significant factors that contribute to it (Zhang and Zhao, 2022). Traditional methods of accident prevention have mostly focused on the reactive response, such as the implementation of traffic laws and the investigation of accidents (Sinh et al., 2022).Nonetheless, as the fields of deep learning advance, the opportunities in the field of predictive and preventive actions that could significantly reduce the possibility of accidents increase (Gan et al., 2020).

Zhang and Zhao (2022) note that deep learning algorithms like Residual Neural Network (ResNet) are especially suitable when analysing temporal data and detecting patterns that can indicate the presence of high risk of accidents. Adewopo and Elsayed (2023) disclosed that deep learning models possess high accuracy capability in forecasting accident-prone places or times because it is trained on massive data, including past traffic data, and driving activity. The combination of accident alert system and traffic sign identification within one deep learning framework is a great step towards creating safer road conditions (Chen et al., 2020). By combining a combination of ResNet sign detector with alert system of the simple type, it is possible to compute road sign interpretation and forecasting of the prospective accidents along with the alert system to carry out in real time accident control mechanisms. By applying ResNet and fixing the deficiencies of the traditional methods, the thesis offers credible solutions to accurate sign detection and timely accident prevention in Nigerian Roads.

2. SYSTEM DESIGN

The study proposes the adoption of transfer learning approach for the development of traffic sign detection scheme. The traffic sign detection scheme applied in this study uses You Only Look Once version 8 (YOLOv8) transfer learning approach for real-time detection. YOLOv8 is one of the latest versions (Jocher et al., 2023) of the YOLO (You Only Look Once) models. The YOLO models are popular for their accuracy and compact size. It is a state-of-the-art model that could be trained on any powerful or low-end hardware. Alternatively, they can also be trained and deployed on the cloud. The block diagram of the proposed system which uses YOLOv8 model for processing and detection of traffic signs is presented in Figure 1.

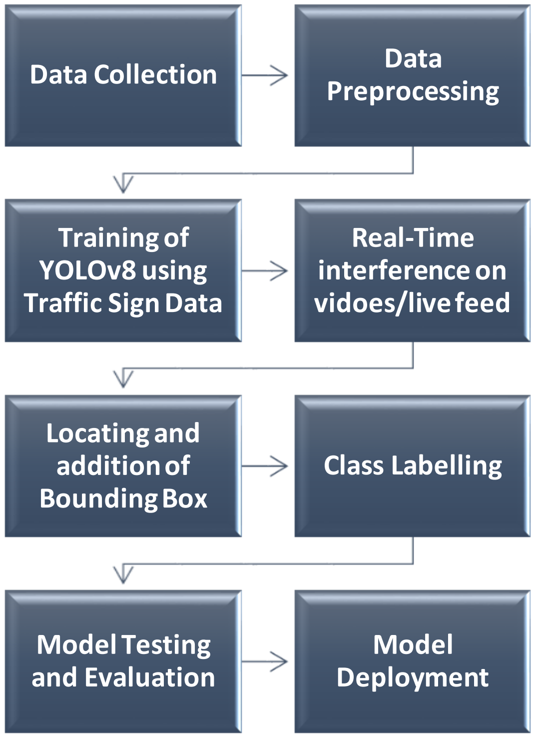

Figure 1: Block Diagram of the Proposed System

In Figure 1, the proposed system work flow is presented in block diagram. The block diagram illustrates how the system will collect data, pre-process the data, Train a transfer learning model (YOLOv8), detect and map out the location of traffic sign in a real-time inference video frame. The performance of the model is finally tested and evaluated before being deployed for use.

2.1 Data Collection

Dataset applied for training the proposed model is the Traffic Sign Dataset focused on check point ahead and cattle crossing signs collected from Kaggle dataset repository. The dataset is comprised of traffic signs used in day and night sights considering various weather conditions. The Traffic-Signs-Dataset is a comprehensive collection of images used for computer vision tasks, particularly in traffic sign detection and recognition. Each image is labelled, enabling supervised learning algorithms to be trained effectively. The dataset is made up of 640 images of labelled data for training the model. Samples of the data used is presented in Figure 2

(a) Cattle crossing sign

(b) Checkpoint ahead sign

Figure 2: Sample Datasets

2.2 Data Preparation

The data preparation step is an essential phase for training an effective YOLOv8 model for the detection of road traffic signs. The data preparation phase adopted in this study involves the cleaning of the data through resizing and removal of noisy data, then annotation and labelling of the data is applied using Roboflow toolbox. Then, the data pixel values are normalized and converted to tensor format using Pytorch transforms. The next phase involves the conversion of the data to YOLOv8 directory structure. Finally, data splitting phase is applied where the data is divided into training set (70%) for training the YOLOv8 model and fine-tuning hyperparameters, test set (20%) and validation set (20%) for evaluation of unseen data and the performance of the model.

2.3 Architecture of the Proposed YOLOv8 Model

Having made the dataset ready, the second step is the training of the YOLOv8 model. This includes environment setup, loading of the configured dataset in the YOLO format, model configuration, and training. The model that will be used to study is the YOLOv8s (small) because it is fast and accurate. The training was conducted in 16 batches each of 50 epoch and 640 image inputs. In training, YOLOv8 gave measurements of loss, mean average precision and training time per epoch. The training process engages the loss of the model in terms of the box and class and objectness and the mean average precision is the most important accuracy measures anticipated to fall within the range of mAP50-95.

In training, optimising the performance of the YOLOv8 model to detect traffic sign involves a very important step of hyperparameter tuning. The model is proposed to be improved with the help of Bayesian optimization technique. The method involves probabilistic model predictive control of optimising the best hyperparameters, and an exploration-exploitation trade-off to trade-off between exploration of new values and refinement of promising values. The hyperparameters that were optimised when training the YOLOv8 include initial learning rate, L2 regularisation to prevent overfitting, batch sizes, input images, Intersection Over Union (IOU) threshold to match objects, anchor box threshold and HSV augmentation parameters. The model architecture suggested is in Figure 3.

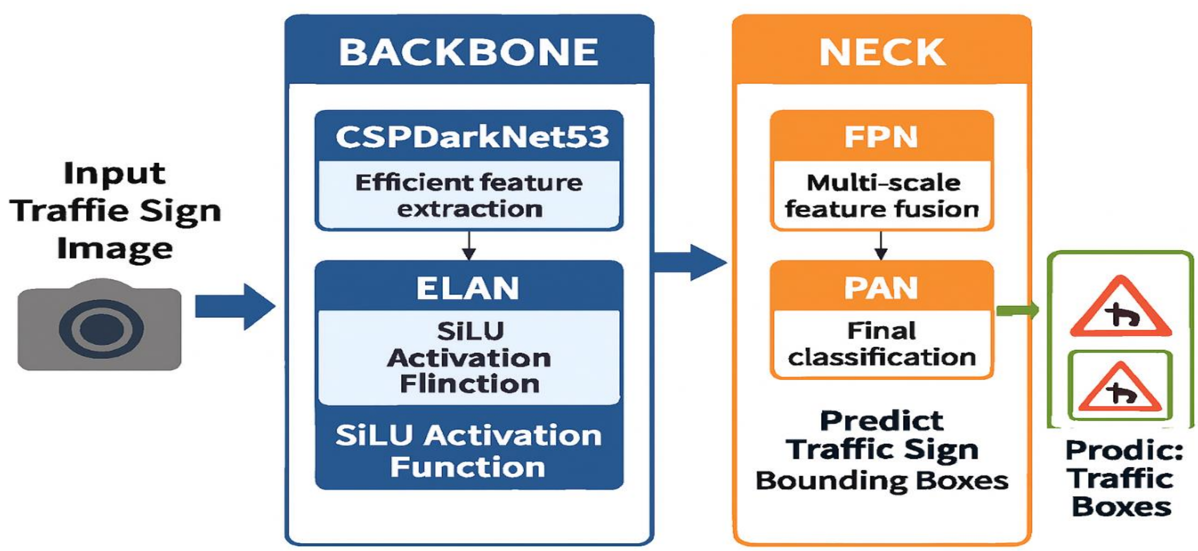

Figure 3: Architecture of the proposed YOLOv8 model

Figure 3 has the Cross Stage Partial Networks (CSPDarkNet53), Efficient Layer Aggregation Networks (ELAN) and Swish (SiLU) activation function as the backbone of the model. The CSPDarkNet53 will be a feature extractor to process the input image, making the model more efficient and less costly to compute, and the ELAN will be utilised to improve gradient flow to increase feature learning and the SiLU will be used to ensure smoother gradients. The Feature Pyramid Network (FPN) of the neck of the model is to improve object detection across varying scales and PAN to improve information flow across levels, between lower and higher and small object detection. The decoupling head of the model is employed to isolate classification and regression instruments and forecast bounding boxes, confidence scores and class probabilities.

2.4 System Implementation

It employs MATLAB development environment to implement the traffic sign detection system that is centred on the identification of road signs Police cheque point ahead and cattle crossing. A driver assistance system application, the traffic sign detection and recognition, assists and informs the driver of road signs to perform necessary actions. In this traffic sign detection and recognition system in the detection of two significant classes of road signs, three detection steps followed are - detection, Non-Maximal Suppression (NMS) and recognition. The system first invokes an object detector network that is the You Only Look Once (YOLOv8) network to detect the traffic signs on an input image. Subsequently, the overlapping detections are eliminated with the help of the NMS algorithm. Lastly, the recognition network categorises the traffic signs spotted as represented in Figure 4.

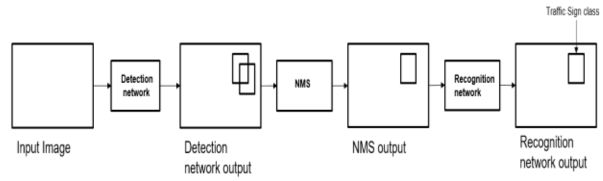

Figure 4: The System Implementation Steps

The intelligent is specialised in producing CUDA MEX to support a machine learning application with particular third-party specifications. Requirements To successfully build CUDA MEX, you require a CUDA-powered NVIDIA 8000-series or later GPU and an appropriate driver. These are fundamental to the assurance that the model can capitalise on the computational power of the GPU. Also, you will require the NVIDIA toolkit and the NVIDIA cuDNN library to use non-MEX builds like the static or dynamic libraries or executables. One should also establish the required environment variables of the compilers and libraries.

You can use coder.checkGpuInstall to cheque that your GPU environment is properly configured. This feature assists in making sure that all necessary compilers and libraries are found so that example should run smoothly. These steps and requirements will enable you to effectively operate the model and utilise the power of the GPU that is essential when it comes to performance-intensive operations in machine learning and deep learning applications.

3 SYSTEM RESULTS

Mean Average Precision (mAP) is a metric used to evaluate the effectiveness of object detection algorithms in identifying and locating objects within images. It considers both precision and recall across different categories. By calculating the Average Precision (AP) for each category and taking the average, mAP provides an overall assessment of the algorithm's performance.

3.1 Training Performance across Epochs

The performance of the system at selected epochs is summarized in Table 1 and the mAP performance over epochs is equally reported in Figure 5.

Table 1: Training and Validation Metrics of YOLOv8 across Epochs

Epoch	Box Loss	Class Loss	Objectness Loss	Precision	Recall	mAP50	mAP50–95
10	0.076	0.020	0.015	0.87	0.85	0.89	0.71
20	0.052	0.014	0.010	0.91	0.89	0.92	0.77
30	0.040	0.010	0.008	0.93	0.90	0.94	0.80
40	0.032	0.009	0.007	0.94	0.91	0.95	0.82
50	0.028	0.008	0.006	0.95	0.92	0.96	0.83

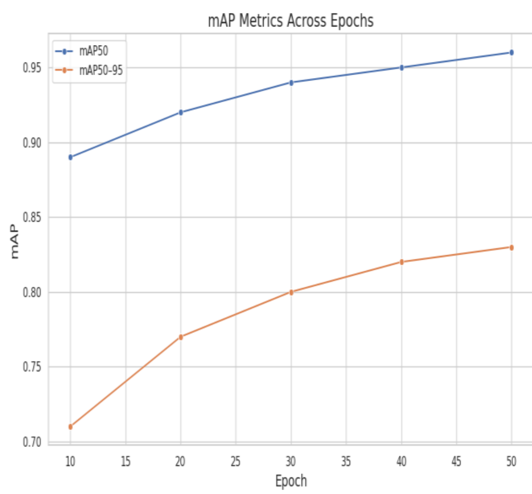

Figure 5: mAP Performance of the Model

When considering only the mAP, the results show a steady improvement in the system's detection accuracy as training progressed, with the most significant gains occurring between epochs 10 and 30, where mAP50–95 rose sharply from 0.71 to 0.80. After epoch 30, the performance gains became more gradual, indicating that the model was approaching convergence and had already learned most of the key visual patterns of the traffic signs. By epoch 50, the mAP50–95 stabilized at 0.83 with mAP50 reaching 0.96, confirming that the model had achieved high accuracy and strong generalization, making it reliable for real-time traffic sign detection.

3.2 Final Class-wise Performance

At the final training epoch (epoch 50), class-level performance for the two traffic sign categories is shown in Figure 6.

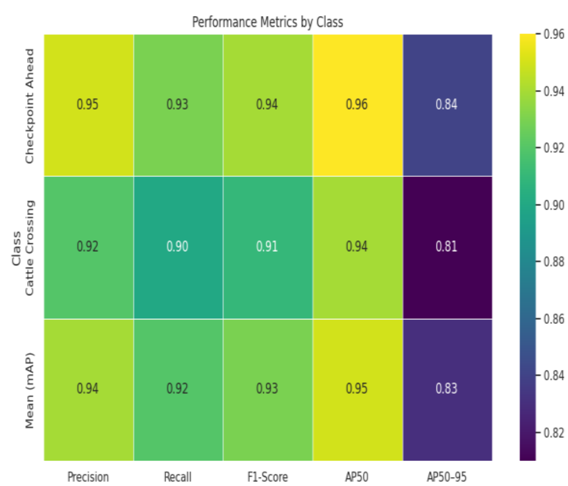

Figure 6: Class-wise Detection Results at Epoch 50

The outcomes of the suggested YOLOv8s model indicate that the two classes of traffic signs were identified with a high accuracy rate, with the checkpoint ahead sign being slightly better in performance than the cattle crossing sign. This variation can be explained by the apparent and more standard visual qualities of the checkpoint sign, in contrast to the cattle crossing sign that tends to be more variable in its appearance, because of the background textures and light. In spite of this minor variation, the precision and recall values of both classes were over 0.90 and this indicates the consistent nature of the system in terms of performing traffic sign recognition tasks.

The mAP increased gradually throughout the training epochs, which showed that the model continually was able to detect and classify the traffic signs. The most significant changes happened in the epochs 10 and 30 and, at that, the mAP levelled off, which proves that the model has come to a successful conclusion. The last mAP5095 of 0.83 and mAP50 of 0.96 indicate that the system can be used to provide high detection accuracy under different environmental conditions such as daytime, night time, rainy days, and other unfavourable weather conditions. This highlights the strength of the model and its applicability in the real world.

The other significant observation is that the model had a good balance of training and validation performance implying that overfitting was successfully reduced. This is attributable to the application of Bayesian optimization during hyperparameter optimization, which made sure that the model had both the accuracy and generalisation. Moreover, small size of the YOLOv8s, along with the use of GPUs, makes the system computationally efficient and able to be deployed in real-time. These findings validate the fact that the proposed method is not just correct but also realistic to the field of intelligent driver aids systems and automated traffic scanners.

4 CONCLUSION

In this paper, the design and implementation of a traffic sign detector system through a transfer learning model on the YOLOv8s model was provided. The system centred on identifying two important road signs including checkpoint ahead and cattle crossing, in different environmental conditions like day and night, and the weather condition. The methodology of the research included the data collection based on a labelled Kaggle dataset, pre-processing by annotation, resizing, normalisation and split into training, validation and test sets, and training a model with optimization of hyperparameters to use Bayesian techniques. The YOLOv8s model, featuring CSPDarkNet53 backbone, ELAN layers, FPN-PAN neck, and decoupled detection head architecture allowed feature extraction, multi-scale detection, and strong classification. The system was coded into MATLAB with the use of GPUs to make sure that it can detect items in real-time.

These findings showed that the YOLOv8s model consistently increased in performance with training epochs, and the accuracy improvements were the greatest between the 10th and 30th epoch. The model scored a precision of 0.95, recall of 0.92, and mean mAP50-95 of 0.83 at the final epoch (epoch 50) demonstrating that the model has high generalisation and that the recall is strong and reliable in detecting all classes of traffic signs. The class-wise analysis showed that the accuracy of the checkpoint ahead sign was slightly higher than the cattle crossing sign, yet both classes retained the accuracy and recall of over 0.90, which proves the strength of the system. The steady decrease in losses (box, class and objectness) over epochs and the levelling of mAP values was a sign of successful model convergence without important overfitting.

To sum up, the designed YOLOv8s-based traffic sign detector system is an effective and efficient solution to driver assistance and intelligent monitoring of traffic in real-time. It is not only highly accurate, computationally efficient, and able to generalise across a range of circumstances, but these traits also make it appropriate to use in a real-world setting. Future extensions of this work could involve expanding the system to detect a broader range of traffic signs, integrating it with autonomous driving platforms, and testing it on larger, more diverse datasets to further enhance reliability.

REFERENCES

Adewopo, V., & Elsayed, N. (2023). Smart city transportation: Deep learning ensemble approach for traffic accident detection. *arXiv Preprint*. https://arxiv.org/abs/2310.10038

Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv Preprint*. https://arxiv.org/abs/2004.10934

Chen, R., Hei, L., & Lai, Y. (2020). Image recognition and safety risk assessment of traffic sign based on deep convolution neural network. *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.3032581

Gan, J., Li, L., Zhang, D., Yi, Z., & Xiang, Q. (2020). An alternative method for traffic accident severity prediction: Using deep forests algorithm. *Journal of Advanced Transportation*, *2020*, Article 1257627. https://doi.org/10.1155/2020/1257627

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

Konkyana, B. L., Korada, D., Sakalabhakthula, B., Pusarala, P., & Kaja, R. T. (2023). Deep learning based real-time traffic sign board recognition and voice alert system. *International Journal of Research Publication and Reviews*, *4*(4), 3321–3327.

Lahare, S., Mishra, A., Nair, A., & Wankhede, V. (2019). Road traffic sign recognition and vehicle accident avoidance system. *Open Access International Journal of Science and Engineering*, *4*(8). ISSN: 2456-3293.

Piralkar, A. (2021). *A deep learning framework to traffic sign recognition in all weather conditions* (Master's thesis, National College of Ireland, School of Computing).

RangeKing, &Jocher, G. (2023). Brief summary of YOLOv8 model structure. *GitHub*. https://github.com/ultralytics/ultralytics/issues/189

Shivayogi, A. B., Dharmendra, C. N. M., Ramakrishna, A. M., & Subramanya, K. N. (2023). Real-time traffic sign recognition using deep learning. *Pertanika Journal of Science & Technology, 31*(1), 137–148.

Singh, K. U., Mishra, V., Shyam, R., & Kumar, A. (2022). Real-time traffic sign detection and mapping for driver assistance using CNN and open-source libraries. *International Journal of Engineering Applied Sciences and Technology, 7*(12), 103–107.

Singh, K., & Malik, N. (2022). CNN based approach for traffic sign recognition system. *Advanced Journal of Graduate Research, 11*(1), 23–33. https://doi.org/10.21467/ajgr.11.1.23-33

Sohan, M., SaiRam, T., &RamiReddy, V. (2024). A review on YOLOv8 and its advancements. In *Proceedings of the School of Computer Science and Engineering, VIT-AP University, Amaravati, India*. https://doi.org/10.1007/978-981-99-7962-2_39

Triki, N., Karray, M., &Ksantini, M. (2023). A real-time traffic sign recognition method using a new attention-based deep convolutional neural network for smart vehicles. *Applied Sciences, 13*(4793). https://doi.org/10.3390/app13084793

Zhang, H., & Zhao, J. (2022). Traffic sign detection and recognition based on deep learning. *Engineering Letters, 30*(2), EL_30_2_31.

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