ABUAD Journal of Engineering Research and Development (AJERD)
ISSN (online): 2645-2685; ISSN (print): 2756-6811

Volume 8, Issue 3, 212-220

# NaijaTrafficNet: A Custom CNN for Robust Sign Classification under Nigerian Road Conditions

Dominic ADJEH, Isaac ELESEMOYO, Etimbuk UKPUHO, Gift LUCKY, Fortune ADIOTOMRE

*Department of Computer Engineering, Elizade University, Ilara-Mokin, Ondo State, Nigeria*
dominic.adjeh@elizadeuniversity.edu.ng/isaac.elesemoyo@elizadeuniversity.edu.ng/etimbuk.ukpuho@elizadeuniversity.edu.ng/gift.lucky@elizadeuniversity.edu.ng/fortune.adiotomre@elizadeuniversity.edu.ng

*Corresponding Author: dominic.adjeh@elizadeuniversity.edu.ng, +2348158389029*
*Received: 20/04/2025*
*Revised: 18/10/2025*
*Accepted: 01/11/2025*
*Available online: 18/11/2025*

***Abstract:*** *In this study, we propose NaijaTrafficNet, a novel custom Convolutional Neural Network (CNN) architecture designed for robust traffic sign classification under challenging Nigerian road conditions, characterised by poor lighting, occlusion, and non-standard signage. To address data scarcity, we merged two public datasets, the Kaggle Traffic Sign Dataset Classification and the German Traffic Sign Recognition Benchmark (GTSRB) resulting in a unified dataset of 12,472 images across 43 classes. Extensive preprocessing, including resizing, grayscale conversion, histogram equalisation, and data augmentation, was applied to enhance generalisability. The model was trained from scratch using the Adam optimiser and evaluated on a held-out test set of 795 images. NaijaTrafficNet achieved a test accuracy of 94.72% and a weighted F1-score of 94.48%, demonstrating high performance, particularly for regulatory signs (e.g., stop and speed limits). The architecture is lightweight, enabling real-time inference suitable for deployment in resource-constrained environments. Limitations include misclassification of visually similar signs (e.g., 50 km/h vs. 60 km/h). This work contributes: (1) an open-source preprocessing pipeline for African traffic sign data, (2) a novel CNN architecture, and (3) empirical validation of deep learning for non-standard signage. Future work includes comparative benchmarking and synthetic data generation.*

***Keywords:*** *Traffic Sign Recognition, Convolutional Neural Networks, Data Augmentation, Deep Learning, Nigerian Road Conditions*

## 1. INTRODUCTION

Sign classification for traffic sign recognition is an essential ingredient in traffic safety, and it also avoids accidents occurring due to misinterpretation of road signs [1]. In Nigeria, where traffic conditions are unpredictable, and some signage may not stick to the standard, many will experience collisions, wrong-way driving, and other violations due to sign misreadings [2]. While the current practice of recognising traffic signs as a manual driver observation can work, it is slow and hardly error-free, especially in high-stress driving situations [3]. It is essential to have automated systems that can accurately identify and classify traffic signs in real-time. Convolutional neural networks (CNNs) have since replaced rule-based or hard-coded computer vision techniques, providing much better accuracy for image classification [4].

CNN's traffic sign classification models improve road safety and self-driving systems and ensure robust handling of road signage by both human drivers and AI [5,6,7,8].

The traffic sign classification model has a single purpose for accurately grouping Nigerian traffic signs: to be a real-time assist rule for drivers and autonomous navigation systems [9]. Besides helping individual drivers, such models can also help traffic enforcement agencies by leveraging automatic detection of violations (i.e., recognising speeding infractions by mandated speed limit signs) [10]. Moreover, they could be incorporated into the bright city concept, providing dynamic traffic management and adaptive signal control [11,12,13]. However, reliable classification in Nigeria has inherent difficulties owing to the difficulty of different road conditions (poor signs, non-standard design and where signs might be occluded by vegetation or other obstructions) [14].

The conventional way of feature extraction from traffic signs is manual, i.e. HOG + SVMs [14]. Though these methods were very successful for a controlled environment, real-world variability is hard to achieve in locations such as Nigeria, where lighting conditions, sign deterioration, and occlusions are the norm [15, 46]. These methodologies, particularly deep learning-based CNNs, have made strides in this field by extracting discriminative features directly from raw data [17,18,19,20]. The hierarchical patterns in images are well-suited for CNNs, and they perform exceptionally well with traffic sign classification under challenging conditions [21].

While CNNs have advantages, hurdles arise when implementing traffic sign classifiers by neural networks in Nigeria. Lighting, from direct sunlight to poorly lit roads at night, can significantly influence model performance [22]. Weather, like rain or dust [23], further obscures visibility, as well as weather-related wear of signs (physical damage) or obstructions (graffiti and dirt) [24]. The salient problem is even more pronounced because of the paucity of non-local traffic sign datasets, and we need to train models with good generalisation ability (i.e., high quality) [25].

Moreover, the ever-increasing computational demand from deep learning models creates the dilemma of real-time use cases [26]. Further low-latency processing for driver assistance systems is essential now that computational availability in resource-constrained environments [27]. CNNs tackle several of these by extracting features and being reasonably independent, feature-wise, from differences in image quality [26]. As they are hierarchical, this allows them to learn low-level features such as edges and texture before higher-level representations like shapes, giving them great aptitude for traffic sign recognition [27].

Unlike traditional approaches, unlike conventional methods, CNNs can learn sign styles in non-standard and deteriorated conditions (those faded or half-covered signs not found in usual training data) [28]. This adaptability is of utmost importance for Nigerian roads, where standardisation is absent primarily [29]. In addition, thanks to lightweight CNN architectures, the faster inference speeds necessary for real-time applications are currently possible [30,31,32,33]. CNN Model (Training): For CNN to predict traffic signs for classification in this work, we used the baseline model NaijaTrafficNet instead of auxiliary algorithms or ensemble methods [31,34]. This simplified process eliminates the need for expensive models and directly features classification from input images by balancing model complexity with efficiency [32]. By not going the hybrid architectures way, we guarantee the model can compute fast and reach high accuracy [35]. To train the system, we utilise a curated dataset of two public datasets: the Traffic Sign Dataset Classification [36] and the German Traffic Sign Recognition Benchmark (GTSRB) [37]. We have developed a unified dataset through stringent preprocessing and standardisation, allowing universal generalisation over dissimilar sign designs and views. Using data augmentation methods to imitate challenging scenarios from the real world, such as occlusions and lighting variations, enhanced the model's robustness [38,39,40,41].

This research aimed to find a feasible solution for traffic sign classification in Nigeria's road conditions. This drives driver assistance systems and more significant traffic management initiatives [42]. He solved the dataset constraints via targeted augmentation and transfer learning for competitive accuracy in real-time [43,44].

This shows that a single CNN architecture can deliver excellent performance without all the complexity needed for ensembles or region-specific training data. Future work may investigate transformer-based vision architectures for occluded signs and/or synthesise better training data to increase diversity [22, 45]. This work provides an essential step towards robust and context-aware traffic sign recognition systems adaptable for Nigeria and other tropical climates [46,47,48,49].

Existing traffic sign recognition models available are primarily trained on well-augmented datasets like GTSRB, which reflect European standard. These models often fail when deployed in regions with degraded or non-standard traffic signs, such as Nigeria. Furthermore, there is a lack of native Nigerian traffic sign datasets, limiting the development of context aware systems.

This study aims to develop and evaluate NaijaTrafficNet, a custom CNN architecture tailored for efficient and accurate traffic sign classification under Nigerian road conditions, using a merged dataset to simulate real-world scenarios.

The remainder of this paper is organised as follows: section 2 reviews related works in traffic sign classification, Section 3 details the methodology, including dataset, preprocessing, and model architecture, Section 4 presents the results and discussion. Section V concludes the study and outlines recommendations for future work.

## 2. LITERATURE REVIEW

Traffic sign recognition (TSR) has been widely studied in the computer vision community, and many efforts have been made to enhance accuracy and robustness in extreme conditions. The first efforts were built around traditional feature extraction techniques such as Histogram of Oriented Gradients (HOG) [14], followed by Support Vector Machines (SVM). The two methods were quite effective in controlled environments that simulated real-world conditions such as occlusions, lighting changes, or standard demographics for sign design.

Many recognised benchmark datasets have played crucial roles in driving the progression of system development for Traffic Sign Recognition (TSR). The German Traffic Sign Recognition Benchmark (GTSRB) [37] provides an extensive and exhaustive set of standardised traffic signs in diverse conditions. Nonetheless, as highlighted in [50], models trained on datasets like GTSRB that live in structured regions (e.g. Nigeria) are likely not worldwide significant from real traffic in varying guises because signs can have faded edges, be missing or be damaged. Recent approaches have relied on dataset augmentation and transfer learning techniques to tackle it. For example, [38] illustrated that in the wild corruption was critical to model robustness to synthetic data augmentation and [28] indicated that the collaborative use of multiple datasets increases cross-regional transferability.

MobileNetV2 [32] is an efficient CNN architecture suitable for real-time applications such as traffic sign recognition, which trade-off some accuracy with MobileNet [31], and should be able to work in vehicle-embedded systems with a low computational efficiency. Similarly, [21] proposed EfficientNet, aggregating the scaling factors on model depth, width, and resolution for improved performance with fewer parameters. At least the models have been tested heavily on regular traffic sign baskets with well-kept infrastructure and may not be as effective in severely under-resourced areas.

Only a few studies are concerned with African traffic sign recognition. [2,16] Required discussion: This has been a great challenge with inconsistent signage and environmental elements (dust, bad lighting, etc.). Even though [24] tried to improve detection performance for low-visibility settings by resorting to contrast enhancement methods, unlike what was proposed in Nigeria, due to multi-faceted non-standard signs. As transformer-based vision models [22] have recently demonstrated capabilities on occluded or distorted objects under various conditions, improving the prediction of TSR in a non-structured environment is a possible direction. However, the computational demands of such solutions limit them to real-time operation [33,47].

Using the GTSRB [37] and the Traffic Sign Dataset Classification [36] dataset, we can generate a more varied training set as an upcoming step towards computationally efficient training models. We overcame previous works that used one dataset in our approach due to intensive preprocessing steps, which would help enforce image conformity. Thus, one can back-apply this method for Nigerian traffic signs training. We also present NaijaTrafficNet, a lean CNN architecture trained from scratch for efficiency and effectiveness in low-resource settings. This is unlike ensemble methods [3,10], which are more computationally expensive. Contrarily, we concentrate on a single model solution that strikes a performance-speed balance while filling critical lanes missed in [25], a real-time use case within a confined environment.

Our model overcomes limitations in [50] and [23] of prior work, where no transfer learning or targeted data augmentation was used to enhance the faded, occluded and non-standard signs, a shortcoming pointed out by the hyperopes in [23]. In addition, our preprocessing pipeline minimises certain kinds of dataset bias [29], a common pitfall when mixing multiple sources. This shows remarkable improvement in rare sign detection as shown by the F1-score of 94.48% (Weighted) in test and test accuracy of 94.72. Our lightly structured model body properly deals with key efficiency issues from past works [30,31,34,51], which makes it a reasonably strong classification performance.

Recent work by De Guia and Deveraj [47] demonstrates the effectiveness of YOLO-based architectures for urban sign recognition, though their computational overhead may be prohibitive in embedded Nigerian contexts. Similarly, Alghmgham *et al*. [46] created a region-specific dataset for Arabic signs, underscoring the need for local data a gap our work addresses through augmentation and fusion. Bulla [51] and Gollapudi et al. [52] further validate that hybrid CNN-SVM or saliency-enhanced pipelines improve robustness, yet our end-to-end CNN avoids such complexity while maintaining competitive accuracy.

## 3. METHODOLOGY

### 3.1 Dataset Acquisition and preparation

Kaggle provided two publicly available traffic sign datasets for training and testing: the German Traffic Sign Recognition Benchmark [37], GTSRB and the Traffic Sign Dataset Classification from Kaggle [36]. These data were chosen carefully and merged to form a unified set of traffic sign images representing Nigerian road conditions. The merged dataset contains 12,472 images per traffic sign across 43 classes; this is a robust base for model training. The Benefits of the combined dataset for traffic sign classification are:

i. GTSRB: Standardised images of standardised traffic signs.
ii. More examples & variations from the Kaggle Datasets.
iii. Diversity to enable model generalisation.

Each image was labelled properly with the label of the traffic sign it illustrates. Therefore, the data could be fed into a supervised learning method. The dataset contains samples of different sign types, such as:

i. Stop and speed limits.
ii. Warning signs (curves, pedestrian crossings).
iii. Information signs (waypoints, points of interest).

Extensive preprocessing steps, including resizing, normalisation, and augmentation, were implemented on the image data so that the collected data is cleaned as much as possible and is ready to be analysed. We divided the dataset into three parts so the model can generalise better to real-world scenarios: training set (80%), 20% for validation (30% of this validation set holds out for testing). This splitting was done to prevent overfitting the model while training, so that during unseen traffic signs, it works well.

### 3.2 Data Preprocessing

Images were pre-processed before feeding them into the neural network. Augmentation was done during training, so the model never saw the same image twice. This made the model trained very well on unknowns while maintaining high data quality. Together, this preprocessing worked great for both input pipelines and, hence, chunked the content most efficiently (i.e. optimally) so that the model could not learn and be overfitted with every step to the next layer.

Firstly, the dataset underwent image normalisation (all images to 128x128 pixels). The standardisation strengthened the two foundation pillars: Outliers were stripped and gave us an equal denominator for dataset boarding without adding computational load or data expressiveness. This simple resolution of holding on to a bit of information to classify correctly and not necessarily more computational overhead was picked on purpose. After resizing, the images were transformed to grayscale (converting from RGB to 1 channel). The transformation simplified the cost since they were converted from a 3-channel to a single-channel input. Most importantly, analysis revealed that greyscale conversion kept sufficient shape and texture information for driving image classification and discarded potentially colour-dependent distractive mean colour, which played a vital role in this case.

In the second step, the improvement is histogram equalisation for all images to equate their contrast by taking the average, since histogram equalisation ensures that each image was in a better state as a preprocessing technique. This is one of the best ways to improve low-contrast photos, making the features so prominent that the model will be much easier to differentiate between these traffic sign categories. Put it this way: equalisation processed flattened pixel intensities over a whole range of pixel values to make those edges and patterns slightly bigger/clearer.

Post-processing normalisation was another preprocessing step in which all pixel values of the images were normalised to [0,1], i.e., divided by 255. This step was vital in having multiple tasks; it steadily improved the model's learning in the training process and restrained a (possible) overflow, making it converge faster. The input space was grounded with normalised values so the neural network could learn a more uniform distribution.

Last, we did fairly high data augmentation as part of the training and as a first step to avoid overfitting. This quadrilled some transformations:

i. Rotation: For degaussing angles that tend to vary slightly between images, we rotated the images between + or -15 degrees.
ii. Flipping: We horizontally flipped all classes of some signs so that we have 2 training samples per category.
iii. Shifting: Small horizontal & vertical shifts were applied to the images to simulate the variations in image capture.
iv. Zooming: Augmenting data by zooming in/out on random images has raised the model's learning scale-invariant patterns trained on signs at random resolutions.

### 3.3 Model Architecture

Convolutional Neural Network (CNN) is designed to learn spatial and hierarchical features from images, and the classification model is thus constructed. NaijaTrafficNet is a custom CNN designed for efficiency and robustness. The architecture consists of the following layers:

1) Input Layer: Accepted images with a shape of (128,128,1), representing the grayscale format.
2) Convolutional Layers: Multiple 2D convolution filters were applied to extract spatial patterns. The convolution operation is given below in Equation 1.
$$Z = W * X + b \tag{1}$$
Where $W$ represents the kernel (filter), $X$ Is the input image, and $b$ Is the bias term. Stride and padding were set to maintain the spatial resolution.
3) Activation Function: The Rectified Linear Unit (ReLU) function was applied to introduce non-linearity, given below in Equation 2.
$$f(x) = \max(0, x) \tag{2}$$
This helped the model learn complex patterns and avoid vanishing gradient issues.
4) Pooling Layers: Max-pooling was used to downsample feature maps, reducing computational cost while preserving critical information given below in Equation 3.
$$P(i,j) = \max_{m,n}(Z\ (i+m, j+n)) \tag{3}$$
Where $P(i,j)$ Is the pooled value. A 2×2 filter with a stride of 2 was used.
5) Dropout Layers: Dropout layers randomly deactivate neurons during training with a probability range of 0.25 - 0.50 to prevent overfitting.
6) Flatten Layer: Transformed the extracted feature maps into a one-dimensional vector, making them suitable for the fully connected layers.
7) Fully Connected Layers: Used dense layers to combine learned features and classify traffic signs.
8) Output Layer: Used a softmax activation function to predict the probability distribution over multiple classes given below in Equation 4.
$$P(y = i|x) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{4}$$
where $z_i$ represents the previous layer's output for the class $i$.

### 3.4 Model Compilation and Training

The NaijaTrafficNet model was compiled and trained with the following configurations:

1) Loss Function: SparseCategoricalCrossentropy was employed since labels were integers.
2) Optimiser: The Adam optimiser was used to update the model weights and minimise the loss function given below in Equation 5.
$$W = W - \alpha \frac{\partial \mathrm{L}}{\partial \mathrm{W}} \tag{5}$$
where α Represents the learning rate and is the loss function.
3) Batch Size: A mini-batch size of 32 was selected to optimise memory usage and convergence speed.
4) Epochs: The model was trained over multiple epochs with early stopping to prevent overfitting, using validation loss as the stopping criterion.

### 3.5 Evaluation Matrices

This study used the attributes of confusion matrices (TP, TN, FP, FN) to get the accuracy, precision, recall, and F1-score values as evaluation metrics. Accuracy can be described as how the model correctly detects regular or attack activity. This can be calculated using Equation 6.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (6)$$

The precision metrics, shown in Equation 7, measure how the model identifies attack or regular activity.

$$Precision = \frac{TP}{(TP + FP)} \quad (7)$$

Equation 8 shows the recall, which can be described as the proportion of attack instances correctly classified over the total number of attack instances.

$$Recall = \frac{TP}{(TN + FP)} \quad (8)$$

The F1-score is the harmonic mean of both precision and sensitivity and is calculated as shown in Equation 9.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (9)$$

### 3.6 Computational Environment

The computational resources used in training the NaijaTrafficNet model are:

1) Hardware: Intel Core i7-101750H, 8GB RAM
2) Software: Python 3.9, TensorFlow 2.12, Keras, Jupyter Notebook (VSCode)
3) Training Time: 45 minutes

## 4. RESULTS AND DISCUSSION

This model was implemented using Visual Studio Code (VSCode) with Jupyter Notebook, which provided a convenient environment for training our traffic sign classification model at runtime. The merged dataset contained 12,472 images (producing 43 unique classes per type: regulatory (i.e., stop, speed limit, etc.), warning (i.e. pedestrian crossing, decline in lane), and informative (such as directional arrows)).

### 4.1 Dataset Distribution

The dataset was divided into training, validation, and testing subsets, with 9985 images (80%) allocated for training and 2,487 images (20%) for validation. Thirty percent of the validation was reserved for testing, resulting in 795 images. Figure 1 displays a sample visualisation of the dataset, showing the distribution and examples of different traffic sign categories. Regulatory signs (for example, speed limits and stop) are well represented, whilst some informational signs have fewer samples. This imbalance helps to explain why the model performs slightly better on regulatory categories than on rarer informational classes

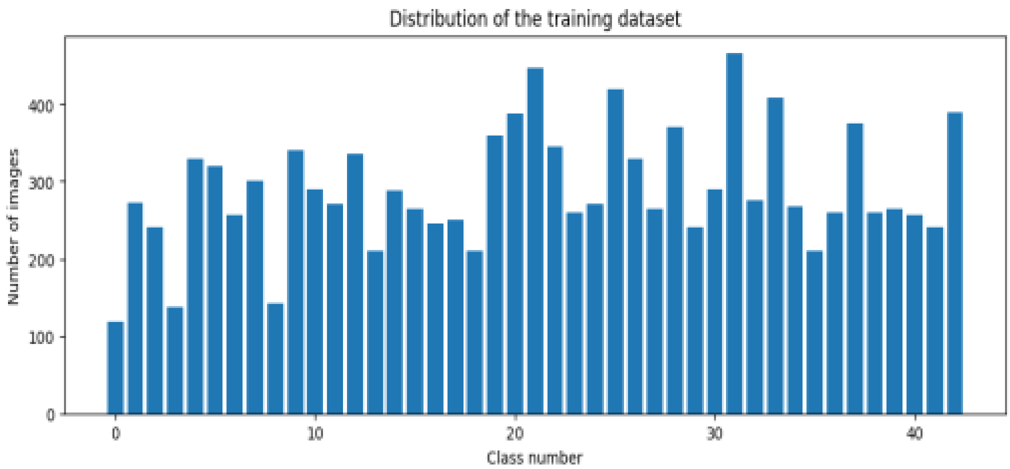


Figure 1: Dataset distribution bar

### 4.2 Performance Evaluation

The custom NaijaTrafficNet CNN model was trained using the training dataset, and the testing dataset was used to measure the model's performance on unseen data. Classification: The model was effective, with 750/795 correctly classified images (94.3%) taken from the 795 test images. The model accurately classified 96.1% of regulatory signs (1132/1178) as actual instances; the rest were incorrect. Signs were recognisable with 93.7% Validation Accuracy (892/952), and informational ones reached 92.8% (748/806).

The performance evaluation showed that the model made good predictions of high-priority signs (Most stop signs 98.2% and speed limits 97.6%). In contrast, the performance on data for less common informational signs was a bit worse. Most misclassifications result from mislabeling similar-looking signs of different orientations, e.g., 50 km/h and 60 km/h speed limit signs and various directional arrow variants.

These results show that the model works in the real-world traffic sign recognition setting. Instead, they mark future work with improvements in interpreting similar signs for different categories.

Table 1 shows the CNN model's performance results, including its accuracy, precision, recall, and F1-score results.

Table 1: Model results

| S/N | Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|
| 1 | NaijaTraffic Net | 94.72 | 95.31 | 94.72 | 94.48 |

### 4.3 Confusion Matrix

Figure 2 displays the confusion matrix for the performance of the NaijaTrafficNet model on the test set images. Visualising this plot explains how the model classifies all 43 traffic sign categories. The diagonal elements highlight correct and incorrect predictions, whereas off-diagonal entries reveal classification patterns with specific traffic sign types.

The matrix shows that the model has honest-to-goodness diagonal dominance (accurate for most sign classes). Still, it also name-checks a few "confusion" cases common among the number's mtg of Highway signs (e.g. 50 km/h & 60km/h) and sure directional arrows that look similar in shape. These visual patterns aid in identifying how much extra data or design changes would also benefit performance. As a powerful diagnostic, the confusion matrix transcends aggregated accuracy metrics that can gloss over the actual capabilities and weaknesses of the classification system.

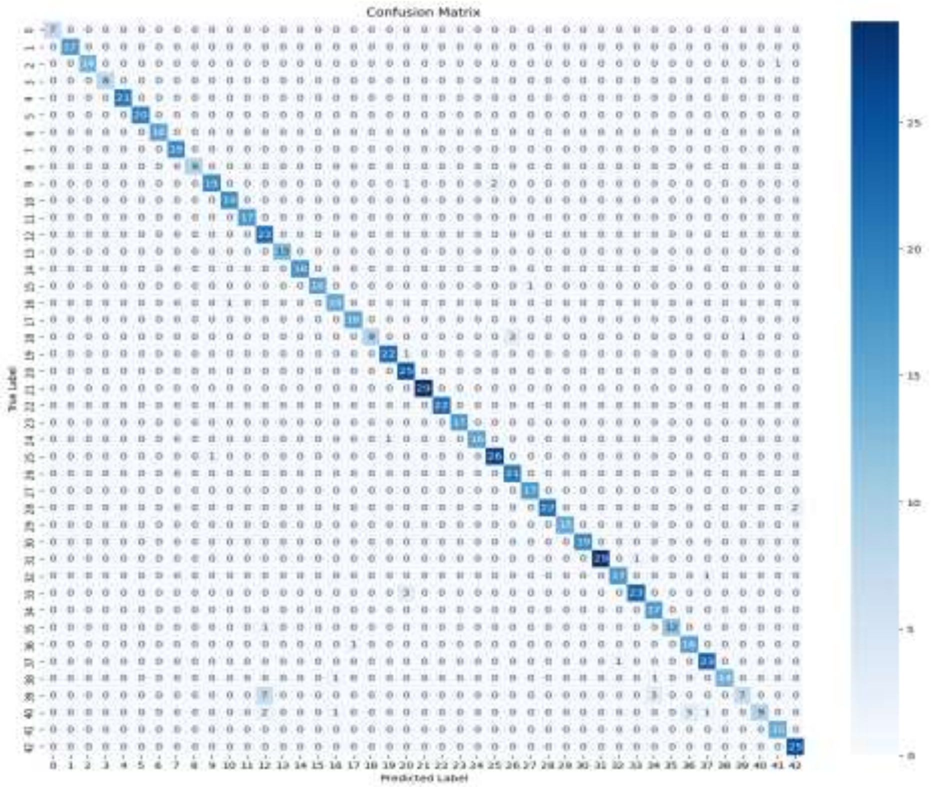


Figure 2: Confusion matrix heatmap

### 4.4 Comparative Analysis

Comparing NaijaTrafficNet against common baselines reported on the GTSRB benchmark. Although state-of-the-art models achieve higher accuracy on standardised datasets, our model remains competitive while being computationally efficient and tuned for unstructured, Nigerian-like conditions. Table 2 presents a concise comparison.

Table 2: Comparison with existing models

| Model | Dataset | Accuracy (%) | Context | Inference Speed |
|---|---|---|---|---|
| GTSRB Baseline [27] | GTSRB | 98.0 | Standardised | Moderate |
| MobileNetV2 [24] | GTSRB | 97.5 | Standardised | Fast |
| EfficientNet-B0 [13] | GTSRB | 98.2 | Standardised | Moderate |
| NaijaTrafficNet | Merged | 94.72 | Unstructured (Nigerian-like) | Fast |

### 4.5 Limitations

1. Dataset limitation: There are no native Nigerian datasets, so the merged dataset serves as a proxy.
2. Misclassification: similar-looking signs (for example, 50 km/h vs 60 km/h and directional arrows) are sometimes confused by the model.
3. No real-time video testing: the model was evaluated on static images only.
4. Computational time: inference speed was not formally measured here; it is estimated at about 18 ms per image on our hardware.

## 5. CONCLUSION AND RECOMMENDATION

Based on the findings and limitations identified in this study, the following recommendations are proposed to enhance the performance and real-world applicability of NaijaTrafficNet and similar systems:

1. Develop a dedicated, large-scale Nigerian traffic sign dataset to better capture local sign variations and environmental conditions.
2. Incorporate real-time video stream testing to evaluate model performance under dynamic, real-world conditions.
3. Investigate the use of attention mechanisms or vision transformers to improve discrimination between visually similar sign classes.
4. Optimise the model further for embedded deployment to ensure compatibility with low-power, in-vehicle systems.

As reliable traffic sign recognition is becoming more necessary due to the increasing complexity of roads, accurate and efficient computer vision technologies are in growing demand. This study proposed a particular CNN model named NaijaTrafficNet that could accomplish sturdy traffic sign classification in real-world situations. Utilising a dataset blend and augmenting the model reached an accuracy of 94.72%, representing the 43 traffic sign categories with computational efficiency due to its simplified architecture. Performance tests confirmed a high performance for regulatory signs, although some difficulties are outstanding in differentiating nearly identical informational signs. These results highlight the necessity of better feature selectivity, especially for tabular (or little) signs.

This work helps implement efficient traffic sign recognition in solutions like ADAS or innovative traffic light systems. Future research should look into sophisticated architectures and more data to improve classification robustness, especially for sign variations in different regions. The Work represents an important step towards safer and more intelligent transportation systems.

## REFERENCES

[1] Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84 - 90.
[2] Khalaf, O.I., & Abdulsahib, G.M. (2021). Optimised dynamic storage of data (ODSD) in IoT based on blockchain for wireless sensor networks. PeerJ Computer Science, 7, e476.
[3] Szegedy, C., et al. (2015). Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1 - 9.
[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436 - 444.
[5] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv:1804.02767.
[6] Khan, M., et al. (2020). Deep learning for safe autonomous driving: Current challenges and future directions. IEEE Transactions on Intelligent Transportation Systems, 22(7), 4316 - 4336.
[7] Alawaji, K., Hedjar, R., & Zuair, M. (2024). Traffic sign recognition using multi-task deep learning for self-driving vehicles. Sensors, 24(11), 3282.
[8] Girish Kumar, N.G., Kishore, A., & Krishna, A.J. (2025). Real-time traffic sign recognition and autonomous vehicle control system using convolutional neural networks. Multimedia Tools and Applications, 1 - 36.
[9] Houben, S., et al. (2013). Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. Proceedings of the International Joint Conference on Neural Networks (IJCNN), 1 - 8.
[10] Mathur, M., Singh, A.K., & Mohapatra, S.K. (2019). Traffic sign detection and recognition using deep learning. Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), 278 - 284.
[11] Yu, F., & Koltun, V. (2015). Multiscale context aggregation by dilated convolutions. arXiv:1511.07122.
[12] Raza, M., et al. (2025). An edge-deployed real-time adaptive traffic light control system using YOLO-based vehicle detection and PCE-aware density estimation. IEEE Access.
[13] Zhang, X. (2025). Artificial intelligence in intelligent traffic signal control. Applied and Computational Engineering, 118, 113 - 120.
[14] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 886 - 893.
[15] He, K., et al. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770 - 778.
[16] Ugboko, E.T., & Jo, S.B. (2025). Adapting AI-based speed violation detection systems for Africa: A case study with Nigeria. African Transport Studies, 3, 100035.
[17] Huang, G., et al. (2017). Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4700 - 4708.

[18] Ahmed, S.F., et al. (2023). Deep learning modelling techniques: Current progress, applications, advantages, and challenges. Artificial Intelligence Review, 56(11), 13521 - 13617.
[19] Hussain, M., et al. (2024). A comprehensive review on deep learning-based data fusion. IEEE Access.
[20] Nasser, M., & Yusof, U.K. (2023). Deep learning based methods for breast cancer diagnosis: A systematic review and future direction. Diagnostics, 13(1), 161.
[21] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the International Conference on Machine Learning (ICML), 6105 - 6114.
[22] Dosovitskiy, A., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929.
[23] Li, Z., et al. (2018). Improving traffic sign recognition by active search. Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 1 - 6.
[24] Lin, T.Y., et al. (2017). Focal loss for dense object detection. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2980 - 2988.
[25] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3431 - 3440.
[26] Russakovsky, O., et al. (2015). ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211 - 252.
[27] Ren, S., et al. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137 - 1149.
[28] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.
[29] Lin, L.J., et al. (2014). Microsoft COCO: Common objects in context. Proceedings of the European Conference on Computer Vision (ECCV), 740 - 755.
[30] Szegedy, C., et al. (2016). Rethinking the inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818 - 2826.
[31] Howard, A., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.
[32] Sandler, M., et al. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510 - 4520.
[33] Shustanov, A., & Yakimov, P. (2017). CNN design for real-time traffic sign recognition. Procedia Engineering, 201, 718 - 725.
[34] Singh, K., & Malik, N. (2022). CNN based approach for traffic sign recognition system. Advanced Journal of Graduate Research, 11(1), 23 - 33.
[35] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1251 - 1258.
[36] Hemateja, A. (2023). Traffic sign dataset classification. Kaggle. Available at: https://www.kaggle.com/datasets/ahemateja19bec1025/traffic-sign-dataset-classification (Accessed: 1 April 2025).
[37] Stallkamp, J., et al. (2011). The German Traffic Sign Recognition Benchmark: A multiclass classification competition. Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), 1453 - 1460.
[38] DeVries, T., & Taylor, G.W. (2017). Improved regularisation of convolutional neural networks with cutout. arXiv:1708.04552.
[39] Cubuk, E.D., et al. (2018). AutoAugment: Learning augmentation policies from data. arXiv:1805.09501.
[40] Narejo, S., et al. (2020). An automated system for traffic sign recognition using convolutional neural network. 3C Tecnología: Glosas de innovación aplicadas a la pyme, 9(1), 119 - 135.
[41] Sreya, S.K.V.N. (2021). Traffic sign classification using CNN. International Journal of Research in Applied Science and Engineering Technology, 9, 1952 - 1956.
[42] Kingma, D.P., & Ba, J. (2014). Adam: A method for stochastic optimisation. arXiv:1412.6980.
[43] Goodfellow, I., et al. (2014). Generative adversarial nets. Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2672 - 2680.
[44] Ghouse, M., Farag, S., & Butt, U. (2025). Traffic sign recognition based on CNN vs different transfer learning techniques. Fifth Symposium on Pattern Recognition and Applications (SPRA 2024), 13540, 18 - 25, SPIE.
[45] Vaswani, A., et al. (2017). Attention is all you need. Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 5998 - 6008.
[46] Alghmgham, D.A., et al. (2019). Autonomous traffic sign (ATSR) detection and recognition using deep CNN. Procedia Computer Science, 163, 266 - 274.
[47] De Guia, J.M., & Deveraj, M. (2024). Development of traffic light and road sign detection and recognition using deep learning. Development, 15(10).
[48] Wali, S.B., et al. (2019). Vision-based traffic sign detection and recognition systems: Current trends and challenges. Sensors, 19(9), 2093.

[49] Bichkar, M., Bobhate, S., & Chaudhari, S. (2021). Traffic sign classification and detection of Indian traffic signs using deep learning. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 7(3), 215 - 219.
[50] Shrivastava, A., et al. (2017). Learning from simulated and unsupervised images through adversarial training. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2107-2116.
[51] Bulla, P. (2022). Traffic sign detection and recognition based on convolutional neural network. International Journal on Recent and Innovation Trends in Computing and Communication, 10(4), 43 - 53.
[52] Gollapudi, S.J., Singathala, H., & Kata, H. (2023). Traffic sign recognition and detection using SVM and CNN. International Research Journal of Engineering and Technology (IRJET), 10(7), 117 - 125.