VFAST Transactions on Software Engineering
https://vfast.org/journals/index.php/VTSE@ 2025, ISSN(e): 2309-3978, ISSN(p): 2411-6246
Volume 13, Number 4, October-December 2025 pp:149-158

Enhancing Traffic Sign Recognition in Rough Weather to Reduce Traffic Accidents

Rizwan Ayazuddin [1*], **Haitham Alqahtan** [2], **Noor Ul Amin** [1], **Manzoor Hussain** [3], **Muhammad Yaqoob Koondhar** [4]

[1]School of Computer Science, Taylor's University, Subang Jaya, Malaysia; [2]College of Engineering, University of Technology Bahrain, Kingdom of Bahrain; [3]Indus University, Karachi, Pakistan; [4]Information Technology Centre, Sindh Agriculture University Tandojam

Keywords: *Arabic Traffic Sign Classification, EfficientNet, Attention Mechanisms, Deep Learning, Computer Vision.*

Journal Info:
Submitted: November 28, 2025
Accepted: December 26, 2025
Published: December 31, 2025

Abstract

Background: Automated traffic sign recognition is a technology that helps in detecting and understanding the traffic signs on the road. These signs may include stop signs, speed limits and some other warning signs. These signs are important in maintaining road safety and helping drivers obeying the traffic rules. Machine learning and computer vision together have advanced in this field and many methods have been proposed for automated traffic sign detection and recognition. Despite these advancements, there are still many challenging situations including occlusion, varying lighting conditions, difficult environmental conditions and sign variations which still need attention. Methods: We use Arabic traffic sign dataset to train EfficientNetB3 architecture with attention mechanism to classify the traffic sign under the unique visual and linguistic complexities as well as diverse environmental conditions. We also improved the dataset by augmenting and adding extra images to cover actual scenarios like fog, heavy rain, low light etc. making it stronger for testing and future research. Results: Our trained model achieved the accuracy of 99.61% in testing better than the compared to the baseline CNN-Resnet model. Conclusion: In this research we addressed the existing limitations and sets a benchmark for the effective and efficient classification of Arabic traffic signs, particularly in challenging weather conditions.

***Correspondence author email address:** rizayazuddin@gmail.com
DOI: 10.21015/vtse.v13i4.2306

1 Introduction

Traffic sign recognition (TSR) is one of the critical component of intelligent transportation systems (ITS) [1]. This technology helps in safer driving through automated detection and interpretation of road signs [2, 3]. In driver assistance systems, autonomous vehicles, and road safety monitoring, where even minor recognition errors can lead to serious accidents [4], it is very important to on time recognize, detect and classify the road traffic signs. In the fast extensive research have been done to recognize these signs in European and American road environments. However, specific research in conditions having extreme weather challenges like Arabic road signs still remains limited. Compared to traffic signs present in western countries the signs in Middle Eastern and Eastern countries introduce unique challenges due to linguistic variations, visual diversity, and distinct sign structures. Environmental factors are also intense and varied here such as changes in light, dust from the desert, vehicle occlusion [16], and sign deterioration because of harsh weather on the roads. These factors highlight the need for robust domain specific solutions that go beyond conventional deep learning architectures [25, 26]. Figure 1 illustrates the multi-faceted challenges of reliable traffic sign detection in real-world environments.

Recently the deep learning algorithms have been advanced like the CNN have improved the accuracy of many classification tasks including traffic sign classification. However, most existing models are trained on datasets like the German Traffic Sign Recognition Benchmark (GTSRB) [5] or the Belgian Traffic Sign Dataset [6], which differ substantially from Saudi traffic sign designs. When we apply the models trained on these datasets they may struggle with Arabic text, culturally specific icons, and regional variations.

In this research we try to fulfill the identified gap by training EfficientNetB3-based deep learning model integrated with attention mechanisms [13] on Arabic traffic sign dataset.

Below we summarize our main contributions:

1. Our proposed model enhanced the visual and linguistic complexities for road signs to avoid road accidents.
2. Our implemented EfficientNetB3 with attention mechanisms, is specifically optimized to handle the unique visual and linguistic complexities of traffic signs under rough weather.
3. We further augmented the dataset to capture real-world challenges [24] such as illumination variations, occlusions, and environmental distortions, providing a robust benchmark for future research of the domain.
4. We conducted extensive experiments with hyperparameter tuning and achieved 99.61% classification accuracy which is higher than the CNN-Resnet architecture setting a new benchmark for Arabic TSR in Saudi urban contexts.

The remaining paper contains the following sections.

- Section 2 is literature review where we discuss some past studies related to traffic sign recognition overall and in specific to challenging environmental conditions particular Saudi Arab.
- Section 3 is Methodology where we describe the data source and preprocessing strategies. Also we discuss the architecture of EfficientNetB3 model.
- Section 4 is the Result & Discussion. In this section we outline the experimental setup and other implementation details. Then we discuss the obtained results and its comparison to baseline models.
- Section 5 is conclusion.

2 Literature Review

TSR has undergone significant evolution over the past three decades, moving from traditional computer vision techniques to advanced deep learning and transformer-based approaches. We present this transformation and progression in figure 2. This figure is a comprehensive timeline of key TSR approaches. We can see a transition from handcrafted features to modern attention-based and transformer-driven methods.

The handcrafted features relived mainly on methods like color segmentation, shape descriptors, Histogram of Oriented Gradients (HOG) [7, 9], and Scale-Invariant Feature Transform (SIFT) [10, 11]. Later they were integrated and coupled with classical classifiers of ML like SVM and KNN. While these methods achieved moderate success, they lacked robustness under changing and variable conditions such as illumination changes, occlusions, and environmental distortions. We see in diagram that later the research mainly focused on using CNN for traffic sign

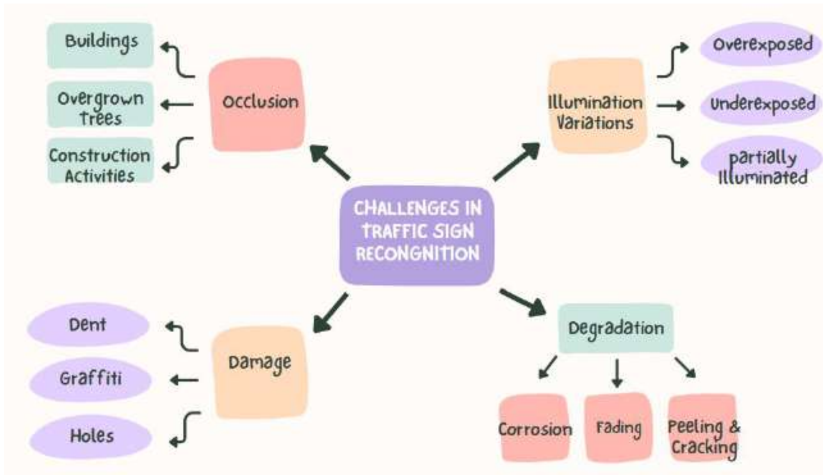

Figure 1. Challenges Faced by Traffic Signs

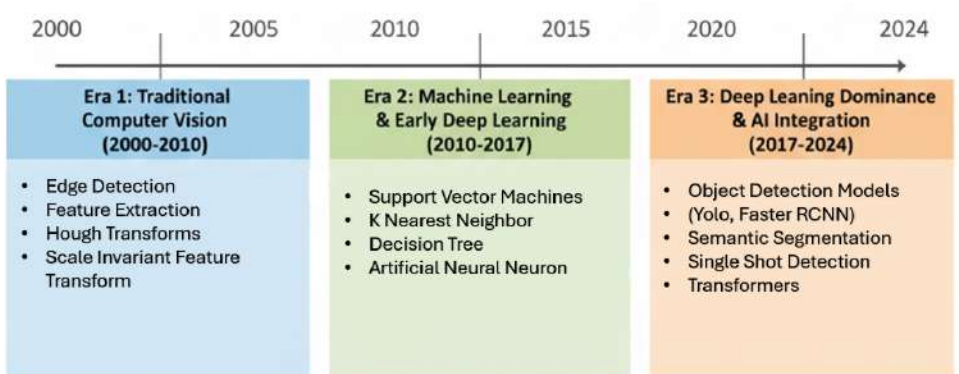

Figure 2. Timeline of TSR Techniques

recognition. Early CNN-based studies demonstrated the capability of deep architectures to automatically extract hierarchical features, significantly outperforming handcrafted feature-based methods. Subsequent advancements, such as hierarchical CNNs [12] and optimization strategies like the Adam optimizer [14], further improved classification accuracy. CNNs were also adopted for real-time TSR, enabling deployment in embedded and automotive systems. Although often with trade-offs in accuracy compared to deeper models [15]. CNNs were also ensemble in different ways and have been explored by many researchers to get multiple learned representations for improved robustness [17]. Apart from CNN detection-based architectures such as Faster R-CNN have been also adapted for TSR. All these methods offer strong performance in complex environments with multiple signs [18].

When we come to Arabic traffic sign classification, we find fewer studies and this area remains underexplored. Some researchers used CNN-based recognition on small-scale Arabic datasets, but these approaches often lack generalization and fail to account for environmental complexities found in Saudi urban areas.

Recent works have introduced EfficientNet architectures and attention mechanisms to enhance feature extraction and achieve higher accuracy in object classification tasks. However, their application to Arabic TSR is still in its infancy. This gap motivates our research, which integrates EfficientNetB3 with attention to deliver robust classification results under actual and dynamic conditions.

In table 1 we provide a comparative summary of existing traffic sign classification approaches.

3 Methodology

In this study we took a comprehensive approach by dividing the main task into several key steps, including data collection and preparation, model architecture, model training, hyperparameter tuning and model evaluation. In the following subsections we write the details of each part of the methodology, from the preparation of the dataset to the model architecture and specific techniques used to enhance performance.

3.1 Data Source

We used dataset ArTS [27] which contain images of traffic signs collected from diverse locations in the Eastern Province of Saudi Arabia, including Khobar, Dammam, and Dhahran. The mentioned dataset contained 2,718 total images.

3.2 Data Collection and Preparation

The dataset includes the images thoughtfully divided into training(2,200 images) and testing sets(518). We also added new images to the original dataset that simulate different real-world challenges. The added scenarios include overexposure, underexposure, dents, occlusion (blending), corrosion, fading, peeling, and cracking. We added occlusion scenarios by introducing overlap images of construction activities, overgrown trees, and buildings.

3.3 Data Split

The dataset is distributed into an 80% training set and a 20% testing set. This was chosen to ensure a balance between sufficient data for training and a robust set for evaluating model performance.

3.4 Challenging Situations Handling

Addressing real-world challenges was central to our methodology. Robust augmentation strategies were applied to the training set, incorporating various filters and effects. For instance, overexposure and underexposure challenges were simulated through adjustments in image exposure parameters. Dents were introduced through localized deformations, while occlusion was mimicked using blending techniques. Corrosion, fading, peeling, and cracking were replicated through texture alterations. We simulated the fog condition by adding semi-transparent layers of white noise. We also used blur filters with reduced contrast to which created visibility reduction as seen or observed in the foggy conditions. For rain augmentation we added the semi transparent lines (visual noise streaks). We reduced the image contrast and brightness to simulate low visibility scenarios, making it harder for the algorithm to identify signs.

3.5 Base Deep Learning Model

For training the deep learning model we used was EfficientNet. This is known for being fast and accurate in image classification tasks. We chose the EfficientNetB3 version because it has shown a good balance between performance and efficiency. We fine-tuned the model using our dataset.

3.6 Model Customization

In figure 4 we present the flow of information through the layers of the proposed model, from the input through the EfficientNetB3 backbone, attention layer, flattening layer, fully connected layers with ReLU activation, and finally, the output layer with softmax activation.

3.6.1 Base Model: EfficientNetB3

EfficientNetB3 is a pre-trained convolution neural network (CNN) that is used in the classification of images. It forms the basis feature extractor of our model. This layer learns complex patterns of hierarchical features of input images.

3.6.2 Attention Layer

Attention layer is integrated to make the model more focused on particular regions. It looks at the context based on the output and input tensor of the base model. The attention mechanism enables the model to dynamically weight various inputs enhancing feature extraction.

Table 1. Summary of Existing Traffic Sign Classification Approaches

Author/Year	Dataset Used	Method/Architecture	Reported Accuracy	Limitations
[19]	GTSRB	CNN	98.5%	Limited to German signs
[20]	GTSRB	Hybrid CNN	99.6%	Not real-time, lacks Arabic applicability
[21]	Belgian Dataset	CNN + Augmentation	97%	Limited dataset diversity
[22]	Custom Arabic dataset	Faster RCNN	92%	Struggles with occlusion & lighting
[23]	GTSRB + Real-world images	ResNet-50	99%	High computation cost
Proposed	Saudi Arabic TSR Dataset	EfficientNetB3 + Attention	99.61%	Computational demand (manageable with GPUs)

(a) Sample Image showing Overexposure

(b) Sample Image showing Underexposure

Figure 3. Sample Images Showing Illumination Variations

3.6.3 Flattening Layer

The flattening layer converts the output to a one dimensional output ready to receive the following fully connected layers. Flattening simplifies the data of dense layers and reduces the multi-dimensional result to a flat one.

3.6.4 Fully Connected Layers with Regularization

Two dense (fully connected) layers are added, each with ReLU activation and L2 regularization.The first layer consists of 256 hidden units, and the second one consists of 128. These layers introduce non-linearity, capturing intricate patterns. Regularization acts to avoid the overfitting through penalizing large weight values.

3.6.5 Output Layer

The last dense layer is the output layer with the use of softmax activation. The classes in the dataset (24 in this case) are the number of nodes. The use of softmax activation gives probabilities of classes, which makes it possible to use multi-class classification. Every node is the probability of the membership to a particular class.

In table 2 below lists the most important hyperparameter that was employed in our model structure. These parameters entail parameters of the base EfficientNetB3 model, the attention layer, density layers, output layer, as well as compilation parameters. Experimentation and optimization of these hyperparameters were done based on performance evaluations.

4 Results and Discussion

The model was trained in 15 epochs with a dataset size of 69 batches in each epoch. The model was able to significantly reduce the loss and increase the accuracy during training. The training loss and correctness rate of the first epoch were 8.9592 and 76.32 percent, respectively. The loss also steadily declined in the later epochs to a low of 0.6338 in the last epoch, with accuracy of 98.59 percent. This means that the model successfully acquired complex patterns in the data attaining a high degree of accuracy.

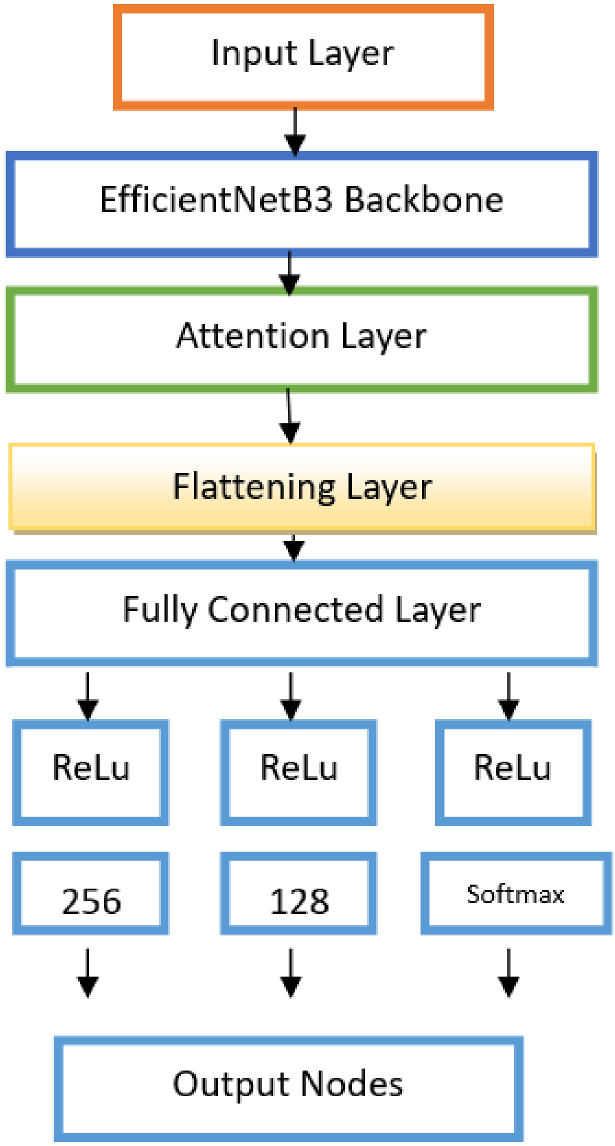

Figure 4. Block Diagram of Proposed Model Architecture

The training process was used to get validation results to evaluate the generalization capabilities of the model. This model had low validation loss of 6.0198 during the first epoch with peak at 0.4860 during the last epoch. The validation accuracy was elevated to a whopping 99.61 percent as compared to 87.26 percent, which proves the strength of the model and its generalization to the unknown data. Figure 5 shows the training accuracy and loss curves of the model.

After training, the model was tested on a different test sample to determine how it performs on samples that were not seen before. Loss on the test was at 0.4715 and was accurate at 99.61. These findings are consistent with the great training accuracy and validation accuracy, which indicates the success of the model in providing accurate predictions on novel and different samples.

The confusion matrix provides a graphic represen-

Table 2. Model Hyperparameters For Each Layer

Layer	Hyperparameter	Value/Description
EfficientNetB3 Base Model	weights	'imagenet' (pre-trained on ImageNet)
EfficientNetB3 Base Model	include_top	False (Exclude final fully connected layer)
EfficientNetB3 Base Model	input_shape	(224, 224, 3) (Input image dimensions)
Attention Layer	input_tensor	Input tensor for attention layer
Sequential Model	Sequential model	Combining EfficientNetB3 and Attention
Dense Layer (1st)	units	256
Dense Layer (1st)	activation	'relu'
Dense Layer (1st)	kernel_regularizer	l2(0.01)
Dense Layer (2nd)	units	128
Dense Layer (2nd)	activation	'relu'
Dense Layer (2nd)	kernel_regularizer	l2(0.01)
Output Layer	units	24 (actual number of classes)
Output Layer	activation	'softmax'
Compilation	optimizer	'adam'
Compilation	loss	'categorical_crossentropy'
Compilation	metrics	['accuracy']

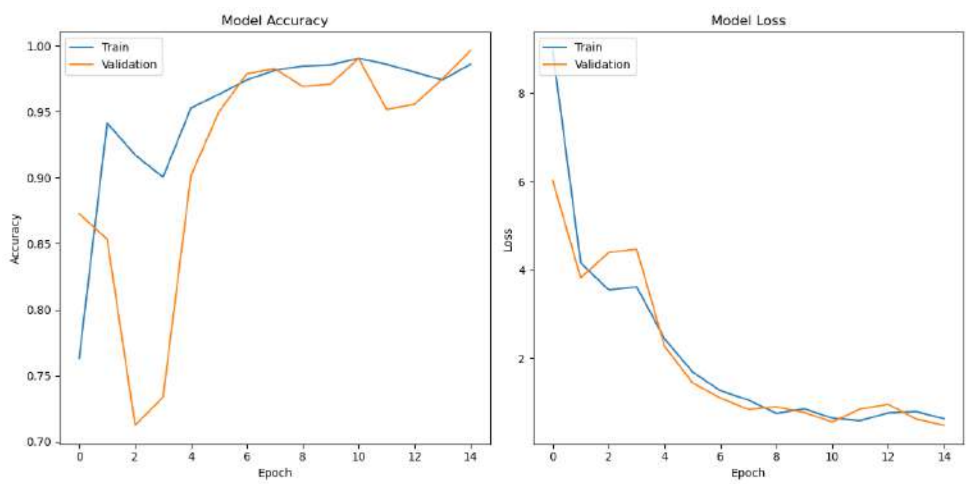

Figure 5. Training and Validation Accuracy and Loss Curves

tation of the model classification performance, which demonstrates the number of correct and incorrect predictions by the model in different classes. The matrix will help to define the possible areas of improvement and learn which classes are more challenging to the model. Figure 6 represents the confusion matrix..

Reading the confusion matrix and the heatmap that it produces in relation to our traffic sign classification, The diagonal values indicate the number of correctly classified examples of each class. These values are largely high in our case which was a good outcome. Nonetheless, there are those classes whose values on the diagonal are low, but those off the diagonal are high. These are the courses in which our model is doing worse. Considering the classes, the following ones appear to yield lower correct classification: classes 6, 11, 12, 13, and 22.

The high accuracy of the model in training, valida-

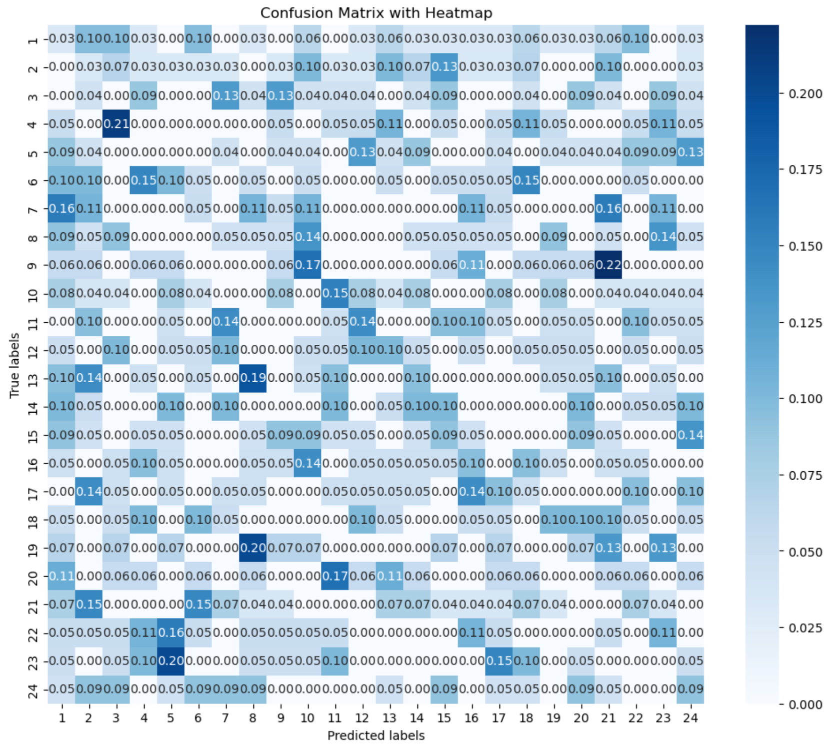

Figure 6. Confusion Matrix

tion, and testing, as well as the detailed metrics, is a testament of the success of the model in learning intricate patterns in the Arabic traffic sign dataset. The focus on regularization methods has probably also played a role in the generalizability of its model that overfitting reduces and improves its overall performance. All these findings support the effectiveness of the given architecture, which incorporates the effectiveness of EfficientNetB3 with extra attention and dense layers to achieve high accuracy levels in detecting and classifying Arabic traffic signs.

When we tested our model with the baseline architecture [27], we observed that the training accuracy consistently remained high, reaching approximately 95.83%, while the validation accuracy remained constant at around 95.83% across all epochs. The loss values also remained stable. This pattern suggests that the model did not learn from the training data, as it may not have effectively captured the underlying patterns or features necessary for generalization. Hence, the model seems to predict the majority class (or one class) for all instances, resulting in high accuracy on both the training and validation sets. Thus, the simple CNN may be overfitting the model. The comparison of our model with the baseline model is presented in Table 3.

5 Conclusions

The proposed traffic sign classification model we have used EfficientNetB3 architecture with attention mechanisms. During training, the model consistently improved, achieving a final accuracy of 98.59%. Validation and testing phases further validated its robustness, yielding impressive accuracies of 99.61%. The confusion matrix highlighted strong overall performance, yet identified specific classes, such as 6, 11, 12, 13, and 22, where the model struggled. Despite the good performance, limitation of class imbalance need attention as it potentially impact on model performance was evident.

In future research we aim to train our model on multilingual datasets having extensive and diverse collection of traffic sign images. This time the augmen-

Table 3. Comparison of EfficientNet with CNN

Model	Training Accuracy	Validation Accuracy	Testing Accuracy	Training Time (s)	Testing Time (s)
Simple CNN	91.23	91.02	92.00	2918	15
ResNet35	95.83	95.83	95.83	3597	21
EfficientNet	98.59	99.61	99.61	5553	12

tation ratio must be kept high to have more resilient and robust model training. We plan to implement GAN as our baseline model in future with de-hazing and de-raining modules as separate layers. This will help us in augmenting the dataset with synthetic images simulating many edge cases scenarios like heavy snow accumulation or extreme sun glare that are rare in current training sets.

Author Contributions

Rizwan Ayazuddin: Conceptualization, Methodology, Software. **Haitham Alqahtan**: Data curation, Writing- Original draft preparation. **Noor Ul Amin**: Visualization, Investigation. **Manzoor Hussain**: Writing- Reviewing and Editing.**Muhammad Yaqoob Koondhar**: Software, Validation.

Compliance with Ethical Standards

It is declared that all authors do not have any conflict of interest. Furthermore, informed consent was obtained from all individual participants included in the study.

References

[1] A. Escalera, J. Armingol, and M. Mata, “Traffic sign recognition and analysis for intelligent vehicles,” *Image and Vision Computing*, vol. 21, pp. 247–258, 2003.

[2] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, “Traffic sign recognition—How far are we from the solution?” in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 2013, pp. 1–8.

[3] M. Fu and Y. Huang, “A survey of traffic sign recognition,” in *Proc. Int. Conf. Wavelet Analysis and Pattern Recognition*, 2010, pp. 119–124.

[4] H. Fleyeh and M. Dougherty, “Road and traffic sign detection and recognition,” in *Proc. 16th Mini-EURO Conf. and 10th Meeting of EWGT*, 2005, pp. 644–653.

[5] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “The German traffic sign recognition benchmark: A multi-class classification competition,” in *Proc. Int. Joint Conf. Neural Networks*, 2011, pp. 1453–1460.

[6] W. Li, D. Li, and S. Zeng, “Traffic sign recognition with a small convolutional neural network,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 688, p. 044034, 2019.

[7] I. Creusen, R. Wijnhoven, E. Herbschleb, and P. With, “Color exploitation in HOG-based traffic sign detection,” in *Proc. IEEE Int. Conf. Image Processing*, 2010, pp. 2669–2672.

[8] N. Jhanjhi, “Investigating the influence of loss functions on the performance and interpretability of machine learning models,” in *Proc. Int. Conf. Mathematical Modeling and Computational Science*, 2025, pp. 480–491.

[9] A. Sugiharto and A. Harjoko, “Traffic sign detection based on HOG and PHOG using binary SVM and k-NN,” in *Proc. 3rd Int. Conf. Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 2016, pp. 317–321.

[10] F. Ren, J. Huang, R. Jiang, and R. Klette, “General traffic sign recognition by feature matching,” in *Proc. 24th Int. Conf. Image and Vision Computing New Zealand*, 2009, pp. 409–414.

[11] M. Kus, M. Gokmen, and S. Etaner-Uyar, “Traffic sign recognition using scale invariant feature transform and color classification,” in *Proc. 23rd Int. Symp. Computer and Information Sciences*, 2008, pp. 1–6.

[12] X. Mao, S. Hijazi, R. Casas, P. Kaul, R. Kumar, and C. Rowen, “Hierarchical CNN for traffic sign recognition,” in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, 2016, pp. 130–135.

[13] N. Jhanjhi, “Comparative analysis of frequent pattern mining algorithms on healthcare data,” in *Proc. IEEE 9th Int. Conf. Engineering Technologies and Applied Sciences (ICETAS)*, 2024, pp. 1–10.

[14] S. Mehta, C. Paunwala, and B. Vaidya, "CNN based traffic sign classification using Adam optimizer," in *Proc. Int. Conf. Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 1293–1298.

[15] A. Shustanov and P. Yakimov, "CNN design for real-time traffic sign recognition," *Procedia Engineering*, vol. 201, pp. 718–725, 2017.

[16] G. K. N, A. Kishore, and A. J. Krishna, "Real-time traffic sign recognition and autonomous vehicle control system using convolutional neural networks," *Multimedia Tools and Applications*, pp. 1–36, 2025.

[17] A. Vennelakanti *et al.*, "Traffic sign detection and recognition using a CNN ensemble," in *Proc. IEEE Int. Conf. Consumer Electronics (ICCE)*, 2019, pp. 1–4.

[18] X. Li, Z. Xie, X. Deng, Y. Wu, and Y. Pi, "Traffic sign detection based on improved Faster R-CNN for autonomous driving," *J. Supercomputing*, vol. 78, pp. 7982–8002, 2022.

[19] A. Sharma and P. Ranjan, "Traffic sign board prediction using ensemble model of YOLOv8 and Detectron2," in *Proc. Int. Conf. Advances in Computation, Communication and Information Technology (ICAICCIT)*, 2023, pp. 378–384.

[20] G. Yildiz *et al.*, "Hybrid image improving and CNN stacking ensemble method for traffic sign recognition," *IEEE Access*, vol. 11, pp. 69536–69552, 2023.

[21] R. Fonolla *et al.*, "Multi-modal classification of polyp malignancy using CNN features," in *Proc. IEEE Int. Symp. Biomedical Imaging (ISBI)*, 2019, pp. 74–78.

[22] R. Alawwad, O. Bchir, and M. Ismail, "Arabic sign language recognition using Faster R-CNN," *Int. J. Advanced Computer Science and Applications*, vol. 12, 2021.

[23] M. Rahman *et al.*, "Evaluating the reliability of CNN models on classifying traffic and road signs using LIME," *arXiv preprint arXiv:2309.05747*, 2023.

[24] M. Humayun *et al.*, "Traffic management: Multi-scale vehicle detection in varying weather conditions using YOLOv4," *Electronics*, vol. 11, p. 2748, 2022.

[25] F. Ashfaq *et al.*, "Using dual attention BiLSTM to predict vehicle lane changing maneuvers," *Systems*, vol. 11, p. 196, 2023.

[26] A. Faisal *et al.*, "A comprehensive review of machine learning models," *Authorea Preprints*, 2025.

[27] D. Alghmgham *et al.*, "Autonomous traffic sign detection and recognition using deep CNN," *Procedia Computer Science*, vol. 163, pp. 266–274, 2019.