

Article

Optimizing AI-Based Traffic Sign Recognition in Electric Vehicles with GELU-Activated CNNs

Ahmet Serhat Yildiz * , Hongying Meng  and Mohammad Rafiq Swash 

Department of Electronic and Electrical Engineering, Brunel University of London, London UB8 3PH, UK; hongying.meng@brunel.ac.uk (H.M.); rafiq.swash@brunel.ac.uk (M.R.S.)

* Correspondence: ahmetserhat.yildiz@brunel.ac.uk

Abstract

Traffic sign recognition is critical for intelligent transportation systems and autonomous driving. Conventional convolutional neural networks (CNNs) typically utilize the ReLU activation function for its computational efficiency; however, alternative activation functions can improve computing effectiveness capacity in recognition tasks. In this study, we propose a CNNs model enhanced with the Gaussian Error Linear Unit (GELU) activation function. We evaluate its performance on benchmark datasets and compare it against both ReLU and Leaky ReLU baseline. Experimental results show that the proposed GELU-activated CNNs achieves a recognition accuracy of 99.75% and provides small but consistent improvements over ReLU and Leaky ReLU models, particularly under challenging conditions such as occlusion and low lighting. These findings highlight GELU's potential to enhance the robustness and reliability of traffic sign recognition in Electric Vehicles for autonomous driving applications.

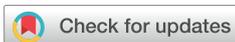
Keywords: traffic sign recognition; electric vehicles (EVs); convolutional neural networks (CNNs); intelligent transportation systems (ITS); rectified linear unit (ReLU); leaky rectified linear unit (Leaky ReLU); gaussian error linear unit (GELU)

1. Introduction

The rapid advancement of intelligent transportation systems (ITS) and the global shift toward electric vehicles (EVs) have transformed modern road environments. EVs are increasingly integrated with advanced perception and decision-making modules to support autonomous driving, driver-assistance functions, and energy-efficient navigation. Among these modules, traffic sign recognition plays a crucial role in ensuring driving safety, maintaining regulatory compliance, and enabling context-aware driving decisions. Accurate identification of traffic signs allows EVs to adjust speed, optimise energy usage, and enhance passenger safety under various environmental and lighting conditions.

Despite significant progress in computer vision and deep learning, traffic sign recognition remains a challenging problem due to factors such as occlusions, illumination changes, motion blur, and variations in sign design across regions. Traditional convolutional neural networks (CNNs) have achieved strong performance; however, their efficiency and generalisation capability can be limited by activation function saturation and vanishing gradient issues that are particularly problematic in real-time EVs applications where low-latency and energy-efficient inference are critical.

To address these challenges, this study explores the integration of the Gaussian Error Linear Unit (GELU) activation function into CNNs-based architectures for traffic sign



Academic Editor: Grzegorz Sierpiński

Received: 27 January 2026

Revised: 5 March 2026

Accepted: 5 March 2026

Published: 12 March 2026

Copyright: © 2026 by the authors.

Published by MDPI on behalf of the

World Electric Vehicle Association.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

recognition. The GELU function, known for its smooth probabilistic behaviour, provides improved nonlinearity compared to conventional activation functions such as ReLU and Leaky ReLU. This property enhances flow of gradients during training and contributes to better model convergence and performance stability.

This research focuses on developing an optimised GELU-activated CNNs framework for electric vehicle environments. The model's performance is evaluated using multiple benchmark datasets to ensure robustness and generalisation across different real-world conditions.

1.1. Research Questions and Motivation

Traffic sign recognition is an important part of the perception systems in electric vehicles (EVs). However, recognition performance can be affected by design choices in CNNs architectures, including the activation function, which is often selected by default without detailed evaluation. To address this gap, this study investigates how different activation functions affect the performance of a CNNs-based traffic sign recognition model.

The research questions are defined as follows:

1. How does the choice of activation function (ReLU, LeakyReLU, and GELU) affect traffic sign recognition performance in a fixed CNNs architecture?
2. Does GELU provide more stable and reliable recognition results compared to ReLU and LeakyReLU when the dataset has difficult driving conditions?

1.2. List of Main Contributions

The main contributions of this paper are summarised as follows:

- We suggest a CNNs-based architecture for classifying traffic signs that is made for EVs perception systems.
- We implement and evaluate the proposed CNNs using three activation functions (ReLU, LeakyReLU, and GELU) to study their effect on recognition performance.
- We provide a comparative analysis of the proposed activation-based variants and the existing traffic sign recognition methods published in the literature.

We discuss the implications of integrating this model into real-time electric vehicle systems to support advanced driving systems.

2. Related Works

The current literature on traffic sign recognition focuses on two primary domains: CNNs-based traffic sign recognition utilising conventional activation functions (ReLU) and the application of one-stage and two-stage object detection models to traffic signs. This research establishes an outline for traffic sign recognition and evaluates the advantages and disadvantages of CNNs model techniques based on ReLU, Leaky ReLU, and GELU for electric vehicles (EVs).

CNNs with the usual Rectified Linear Unit (ReLU) activation were used a lot in early work on traffic sign recognition. Arcos-García et al. presented a deep neural network incorporating Spatial Transformer Networks (STN) and ReLU-based activation functions to improve the consistency of traffic sign recognition, achieving 97.70% accuracy in the GTSRB dataset. They demonstrated that this design, together with effective optimisation methods, enhance recognition performance [1]. He et al. proposed a lightweight CNNs-based traffic sign recognition model designed for portable and embedded applications. The model focuses on speed and size. The GTSRB dataset showed that their network was 99.30% accurate and shows that their compact architecture works well for real-time traffic sign recognition tasks under resource-constrained conditions [2].

In 2021, a couple of studies looked at how to improve traffic sign recognition better by using CNNs with ReLU architectures that are both light and accurate. These approaches contributed to more efficient network designs, such as DeepThin, improved VGG-based and multi-layer CNNs models, which significantly reduced the number of parameters while still performing well. Even though these approaches had small architectures, they were very accurate in the GTSRB dataset, with results of 99.29%, 99.21%, and 99.45%. This shows that lightweight models can still provide high recognition accuracy [3–5].

In 2022, Real-time CNNs with ReLU models trained on hybrid datasets attained exceptional performance, achieving 99.85% accuracy on GTSRB. However, evaluations of Vision Transformers showed their performance remained lower than that of CNNs. Further studies explored transfer learning in adapted LeNet-based models, demonstrating satisfactory accuracy despite minimal data in real-time applications [6–10].

Research in 2023 continued to advance traffic sign recognition by fusing predictions from models such as ResNet50, DenseNet121, and VGG16 with ReLU achieved strong performance, reaching 98.84% on the GTSRB datasets. Other studies also examined end-to-end detection and classification frameworks using GTSRB datasets, demonstrating that customised CNNs with ReLU architectures can improve real-time accuracy [11–13].

In 2024, research on traffic sign identification continued to improve CNNs architectures with ReLU activation to accommodate differences in environmental factors. Several optimised models trained on GTSRB demonstrated strong performance, with reported accuracies reaching up to 97.76%. Better preprocessing and simplified CNNs designs made recognition more reliable for smart transportation systems. CNNs with ReLU generally show strong performance, but they still face difficulties under challenging conditions [14–19].

In 2025, Deep CNNs with ReLU models trained in GTSRB worked to improve the stages of preprocessing, feature extraction, and recognition to ensure that performance was consistent in different road and weather conditions. Several models show deep learning pipelines that could deal with different weather and lighting conditions, obtaining accuracies of up to 97% on GTSRB [20,21].

Overall, the reviewed studies show that ReLU-based CNNs work well on standard datasets but still struggle in difficult conditions or when training data are limited, which is common in electric and autonomous vehicles. ReLU blocks all negative values, and although Leaky ReLU allows small negative output, both activations have limitations in handling the full range of inputs. In contrast, GELU activation works smoothly with both negative and positive values from $-\infty$ to $+\infty$, allowing the network to learn more flexible and expressive features. For this reason, this work focuses on a CNNs model that uses GELU and compares its performance against CNNs using ReLU and Leaky ReLU to determine which activation function offers better accuracy and stability for traffic sign recognition in electric vehicles.

3. Methodology

The proposed study aims to improve traffic sign recognition performance for advanced driver-assistance systems in electric vehicles (EVs) by integrating the Gaussian Error Linear Unit (GELU) activation function into a Convolutional Neural Network (CNNs) architecture. Traditional CNNs-based recognition systems commonly used in the Rectified Linear Unit (ReLU) due to its simple and fast; however, ReLU discards all negative input values, which may cause the loss of delicate and useful features in complex and noisy driving environments. An alternative version, Leaky ReLU, addresses this issue by allowing a small and non-zero gradient for negative inputs, thereby reducing neurone inactivity. Despite this improvement, Leaky ReLU still uses a piecewise-linear transformation. In contrast, GELU provides smoother and more probabilistic activation behaviour, which makes it

possible to extract higher-level features. This is especially useful for EVs perception systems that need to be very reliable. This study investigates the potential of GELU to provide enhanced traffic sign recognition with higher accuracy in comparison to both ReLU and Leaky ReLU. Figure 1 shows the comparison of the ReLU [22], Leaky ReLU [23], and GELU [24] activation functions, highlighting their different nonlinear characteristics. These activation functions impact how neural networks learn complex patterns, which directly affects the performance of traffic sign recognition models.

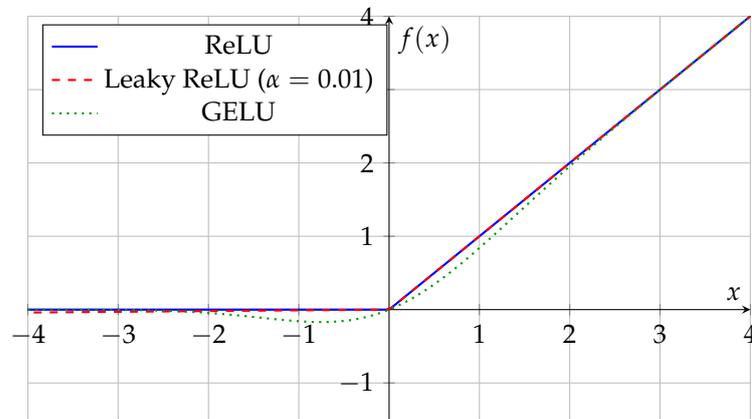


Figure 1. Comparison of the ReLU, Leaky ReLU, and GELU activation functions illustrating their different nonlinear characteristics.

The CNNs pipeline used for activation function comparison consists of four identical convolutional blocks (Figure 2). Each block includes a Conv2d layer followed by batch normalisation and max pooling. A non-linear activation function is applied in each block. The same network architecture, layer depth, and pooling strategy are used in all experiments, while the activation function is changed between ReLU, Leaky ReLU, and GELU. All other components remain unchanged. After the final convolutional block, the extracted feature maps are flattened and passed through Linear layers with Dropout for final recognition.

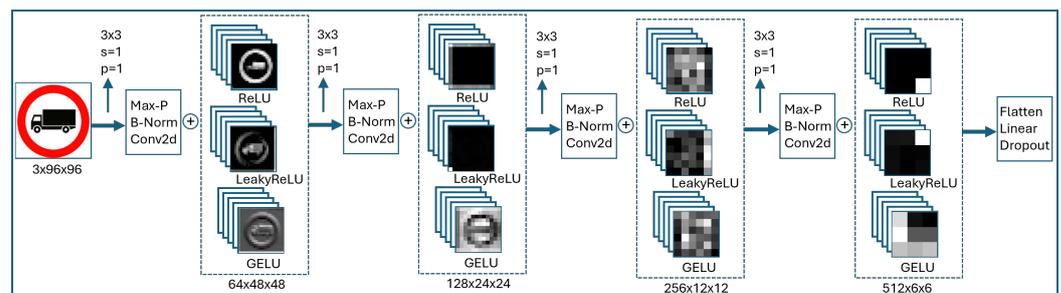


Figure 2. CNNs pipeline used for activation function comparison. The network consists of four identical convolutional blocks, each composed of Conv2d, Batch Normalization (B-Norm), and Max Pooling (Max-P). The same architecture, layer depth, and pooling strategy are maintained across all experiments, while the non-linear activation function is changed between ReLU, Leaky ReLU, and GELU. The extracted features are flattened and passed through Linear layers with Dropout for final recognition.

3.1. ReLU Activation

Activation functions significantly affect the learning capacity of deep neural networks. The Rectified Linear Unit (ReLU) is one of the most widely used activation functions and is defined as:

$$\text{ReLU}(x) = \max(0, x), \tag{1}$$

ReLU helps with the vanishing gradient problem since it makes the gradient a value of zero (for negative inputs) or one (for positive inputs). However, it discards all negative input values, which could cause the dying ReLU problem and this problem makes harder for the network to extract subtle features in difficulty visual situations.

3.2. Leaky ReLU Activation

Leaky ReLU addresses the zero-gradient problem with ReLU by adding a small, non-zero slope to the negative area. This helps neurones avoid becoming inactive (known as the dying ReLU problem) when their inputs are below zero, enabling the network to continue updating weights even for negative activations.

$$\text{Leaky ReLU}(x) = \begin{cases} x, & \text{if } x > 0, \\ \alpha \cdot x, & \text{if } x \leq 0, \end{cases} \quad (2)$$

where α is a small value, usually 0.01. Leaky ReLU provides negative values to contribute a little to feature learning, which helps prevent neurones from becoming inactive. But it is still a piecewise-linear function with almost sharp transition at zero.

3.3. GELU Activation

The Gaussian Error Linear Unit (GELU) is a smoother and more flexible alternative to the ReLU activation function. GELU is defined as:

$$\text{GELU}(x) = x \cdot \Phi(x), \quad (3)$$

where $\Phi(x)$ is the standard normal distribution's cumulative distribution function. The frequently used definition is as follows:

$$\text{GELU}(x) = \frac{1}{2}x \left(1 + \tanh(\sqrt{2/\pi}(x + 0.044715x^3)) \right). \quad (4)$$

Unlike ReLU and Leaky ReLU, GELU smoothly scales inputs according to the probability that they are to be negative and positive. This improves gradient flow and makes it possible to extract more features. This is especially helpful for activities that need very precise image recognition, such as traffic sign recognition. A summary of common activation functions is presented in Table 1.

Table 1. Summary of common activation functions.

Function	Formula	Output Range	Common Use
ReLU	$\max(0, x)$	$[0, \infty)$	Standard CNNs layers
Leaky ReLU	$\max(0.01x, x)$	$(-\infty, \infty)$	Avoids dying ReLU problem
GELU	$x \cdot \Phi(x)$	$(-\infty, \infty)$	Modern architectures

3.4. Dataset

The German Traffic Sign Recognition Benchmark (GTSRB) dataset, which has more than 50,000 images sorted into 43 traffic sign classes, was used to train and test the model [25]. The size of each image was adjusted to 96×96 pixels, and the values were normalized to the range $[0, 1]$.

This study used the German Traffic Sign Recognition Benchmark for recognition. The official GTSRB benchmark provides two training archives (GTSRB_Final_Training_Images.zip and GTSRB_Training_fixed.zip) and one test archive (GTSRB_Final_Test_Images.zip). In this work, the training archives were used as provided by the benchmark. Since the

dataset does not include a predefined validation set, each training archive was divided into training (80%) and validation (20%) subsets to enable model validation during training.

The GTSRB_Final_Training_Images comprises 39,209 images. The dataset is divided into training (80%) and validation (20%) subsets, comprising 31,367 images for training and 7842 images for validation. The split was performed randomly while preserving the class distribution across the 43 traffic sign categories.

The GTSRB_Training_fixed includes 26,640 images. The dataset is divided into training (80%) and validation (20%) subsets, comprising 21,312 images for training and 5328 images for validation. The same random splitting strategy was applied to ensure consistent class distribution between training and validation sets.

The official GTSRB_Final_Test_Images dataset, containing 12,630 test images, was used exclusively for testing. No test images were used during training or validation to prevent data leakage. The recognition model's performance is assessed using accuracy, precision, recall, and F1-score. An overview of the German Traffic Sign Recognition Benchmark (GTSRB) dataset splits, including training, validation, and test sets, is shown in Table 2.

Table 2. Overview of the German Traffic Sign Recognition Benchmark (GTSRB) dataset splits used in this study.

Dataset File	Total Images	Training (80%)	Validation (20%)
GTSRB_Final_Training_Images	39,209	31,367	7842
GTSRB_Training_fixed	26,640	21,312	5328
Test Dataset			
GTSRB_Final_Test_Images	12,630	–	–

3.5. Proposed Model Architecture

The proposed network consists of four convolutional blocks that are used to extract features from input images. Each block includes a 3×3 convolutional layer with stride $s = 1$ and padding $p = 1$, followed by batch normalisation, the GELU activation function, a dropout2d layer with a dropout rate of 0.1, and a 2×2 max pooling layer with stride 2. This structure reduces the spatial size of the feature maps while increasing the depth of the extracted features. In the third stage, the number of kernels is increased to 256 to learn more complex features. Batch normalisation, the GELU activation function, dropout2d, and max pooling are applied to further reduce spatial resolution. In the fourth convolutional step, the number of filters is increased to 512, enabling the extraction of more detailed features. As in previous steps, batch normalisation, GELU activation, dropout2d, and max pooling are applied. After the fourth convolutional block, an adaptive average pooling layer with output size 6×6 is applied to standardise spatial dimensions before flattening. After the final convolutional step, the feature maps are flattened to a one-dimensional vector. Then, this vector is passed through a fully connected layer with 512 units, followed by the GELU activation function. Dropout is applied during training to reduce over fitting. After that, the model applies a second fully connected layer that reduces the feature size from 512 to 128, followed by GELU activation and dropout to improve generalisation. Finally, a third fully connected layer maps the 128 features to 43 output classes corresponding to the traffic sign categories. Also, the softmax function is used to produce class probability scores. The relatively large first fully connected layer ($18432 \rightarrow 512$) is designed to preserve high-dimensional feature representations after extraction of convolutional features. This is important for determining the difference between classes of visually similar traffic signs. The subsequent dimensionality reduction balances the representational capacity and regularization.

3.6. Batch Normalization

Batch normalisation is applied after each convolutional layer to improve training stability and accelerate resolution. This usually allows for the use of larger learning rates and normalises intermediate feature distributions. Additionally, batch normalisation provides a slight regularisation effect that improves generalisation performance [26].

$$\text{BN}(x_i) = \gamma \frac{x_i - \frac{1}{m} \sum_{j=1}^m x_j}{\sqrt{\frac{1}{m} \sum_{j=1}^m \left(x_j - \frac{1}{m} \sum_{k=1}^m x_k\right)^2 + \epsilon}} + \beta \quad (5)$$

The mini-batch size is the number of training samples processed together in a single forward and backward passes during training. m represents the size of the mini-batch. x_i denotes the i -th input activation within the mini-batch. γ is a trainable scale parameter that adjusts the variance of normalised activations. β is a trainable shift (bias) parameter that adjusts the mean of the normalised output. ϵ is a minor constant added for numerical stability. The indices i , j , and k denote the summation indices used in calculating the mini-batch mean and variance.

3.7. Adaptive Average Pooling

An adaptive average pooling layer is employed to reduce the spatial size of the feature maps to a fixed dimension before the fully connected layer. Calculate the average of values inside specific regions of the feature map so that the output has a predefined size. This ensures that the classifier receives a consistent input size regardless of the original feature map dimensions [27].

The proposed network is well suited for real-time traffic sign recognition, as it provides a good balance between model depth and prediction accuracy. This computational efficiency makes our proposed network suitable for real-time traffic sign recognition in electric vehicles with limited computational resources. The architecture of the convolutional neural network proposed with GELU is summarised in Table 3, which presents the types of layers, input and output dimensions, kernel sizes, and the number of parameters used in the model.

Table 3. Summary of proposed model layers with input/output shapes, kernel sizes, and parameters (Input: $3 \times 96 \times 96$).

Layer	Type	Input Shape	Output Shape	Kernel	Parameters
0	Input	(3, 96, 96)	–	–	–
1	Conv2d	(3, 96, 96)	(64, 96, 96)	3×3	1728
2	BatchNorm	(64, 96, 96)	(64, 96, 96)	–	128
3	GELU	(64, 96, 96)	(64, 96, 96)	–	0
4	Dropout2d	(64, 96, 96)	(64, 96, 96)	$p = 0.1$	0
5	Max Pooling	(64, 96, 96)	(64, 48, 48)	2×2	0
6	Conv2d	(64, 48, 48)	(128, 48, 48)	3×3	73,728
7	BatchNorm	(128, 48, 48)	(128, 48, 48)	–	256
8	GELU	(128, 48, 48)	(128, 48, 48)	–	0
9	Dropout2d	(128, 48, 48)	(128, 48, 48)	$p = 0.1$	0
10	Max Pooling	(128, 48, 48)	(128, 24, 24)	2×2	0
11	Conv2d	(128, 24, 24)	(256, 24, 24)	3×3	294,912
12	BatchNorm	(256, 24, 24)	(256, 24, 24)	–	512
13	GELU	(256, 24, 24)	(256, 24, 24)	–	0
14	Dropout2d	(256, 24, 24)	(256, 24, 24)	$p = 0.1$	0
15	Max Pooling	(256, 24, 24)	(256, 12, 12)	2×2	0
16	Conv2d	(256, 12, 12)	(512, 12, 12)	3×3	1,179,648
17	BatchNorm	(512, 12, 12)	(512, 12, 12)	–	1024
18	GELU	(512, 12, 12)	(512, 12, 12)	–	0

Table 3. Cont.

Layer	Type	Input Shape	Output Shape	Kernel	Parameters
19	Dropout2d	(512, 12, 12)	(512, 12, 12)	$p = 0.1$	0
20	Max Pooling	(512, 12, 12)	(512, 6, 6)	2×2	0
21	AdaptiveAvgPool2d	(512, 6, 6)	(512, 6, 6)	6×6	0
22	Flatten	(512, 6, 6)	(18432)	–	0
23	Linear	(18432)	(512)	–	9,437,184
24	GELU	(512)	(512)	–	0
25	Dropout	(512)	(512)	$p = 0.3$	0
26	Linear	(512)	(128)	–	65,664
27	GELU	(128)	(128)	–	0
28	Dropout	(128)	(128)	$p = 0.3$	0
29	Linear	(128)	(43)	–	5547

3.8. Training Configuration

The model was implemented using PyTorch (version 2.7.1) with CUDA 12.8 support. Training was performed using the Adam optimiser with a learning rate of 1×10^{-4} , a batch size of 64, and for 100 epochs. Categorical cross-entropy was used as a loss function, while accuracy, precision, recall, and F1-score were employed as evaluation metrics. All experiments were conducted on a laptop equipped with an NVIDIA GPU.

3.9. Baseline Comparison

Two baseline CNNs models were implemented for comparison.

1. Baseline 1: CNNs + ReLU. This model uses the same CNNs architecture with the ReLU activation function and serves as a standard reference.
2. Baseline 2: CNNs + Leaky ReLU. In this model, ReLU is replaced with Leaky ReLU using a negative slope of $\alpha = 0.01$ to allow limited negative activations.
3. Proposed Method: CNNs + GELU. In the proposed model, the ReLU activation function is replaced with the GELU activation function. This modification enables smoother non-linear transformations and allows negative inputs to contribute to the learning process.

All models were trained under identical conditions to ensure a fair comparison. The results, presented in Section 5, show that CNNs with GELU activation achieves higher accuracy and better generalisation, demonstrating its suitability for traffic sign recognition tasks.

4. Evaluation

Figure 3 shows how different activation functions affect the first convolutional layer. ReLU creates feature maps that are very sparse, with many values becoming zero and some channels showing little activity. Leaky ReLU keeps small negative values, which helps keep more information; therefore, the feature maps have stronger contrast. GELU makes feature maps that are smooth and continuous, keeping both positive and negative values. These differences are due to the mathematical features of each activation function and show how the first layer processes the traffic sign image differently.

In Figure 4, the second layer captures the higher-level structures of the traffic sign, and the effects of different activation functions can be clearly seen in the feature maps. ReLU produces the sparsest activations, with several channels appearing almost completely black because negative values are hard zeroed out, which means that weaker features are lost. Leaky ReLU does not completely delete negative values, which allows weak features to remain visible. This leads to stronger contrast and clearer edges in feature maps compared to ReLU. GELU allows both positive and negative activation values and scales them smoothly instead of applying a hard threshold. This keeps both large and small

activation values in the feature maps, preserving the circular border and the interior details of the traffic sign.

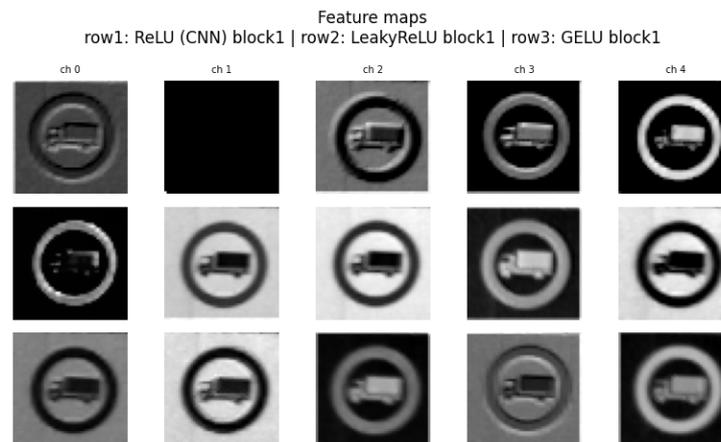


Figure 3. Visualization of Layer 1 feature maps for three activation functions: ReLU (top row), Leaky ReLU (middle row), and GELU (bottom row). Columns represent different channels (ch0–ch4). The feature maps are generated using a sample traffic sign image from the GTSRB dataset as input. Five representative channels are selected for visualization, and the feature maps are normalized to the range [0,1] for display.

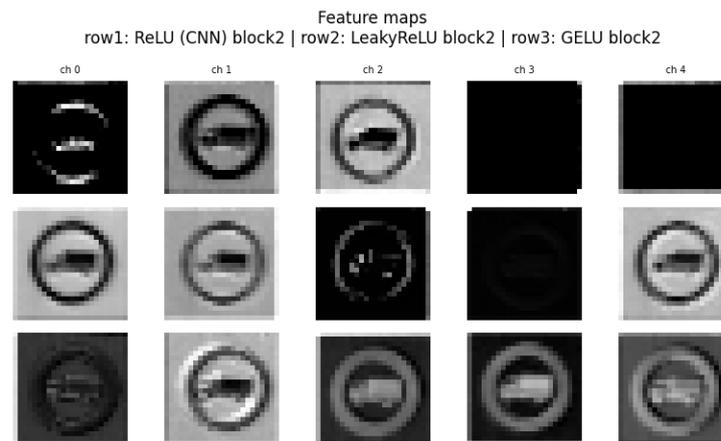


Figure 4. Visualization of Layer 2 feature maps for three activation functions: ReLU (top row), Leaky ReLU (middle row), and GELU (bottom row). Columns represent different channels (ch0–ch4). The feature maps are produced using the same input image from the GTSRB dataset. Selected channels are displayed to demonstrate the evolution of spatial information over deeper convolutional layers. The feature maps are normalized to the range [0,1] for visualization.

Figure 5 shows that the spatial dimensions ($H \times W$) of the feature maps are reduced to 12×12 , which means that each feature map represents a much larger portion of the input image. Therefore, the network no longer focuses on fine details, such as edges or textures. However, it instead captures more general and high-level information, such as the overall shape and structure of the sign. The activation function plays an even larger role at this deeper level of the network and determines which features are preserved and which are suppressed. Consequently, different activation functions lead to noticeable differences in how much information is retained in the feature maps. ReLU sets all negative input values to zero. Hence, many neurones become inactive and produce no output, which causes several feature map channels to be almost completely deactivated. Leaky ReLU allows for a small number of negative values to pass through instead of making them completely to zero. Because of this, fewer neurones are reduced to zero compared to ReLU, so some

structural information is still preserved. However, many weak activations are still reduced, which results in feature maps with stronger contrast and some dark areas. This means that only part of the structure is kept. GELU does not completely prevent neurones with small values. Instead, it enables many small signals to pass through in a smooth way. This helps the network keep more data, even when the feature maps are small. This helps keep the sign's shape and the structure of the sign in a better condition.

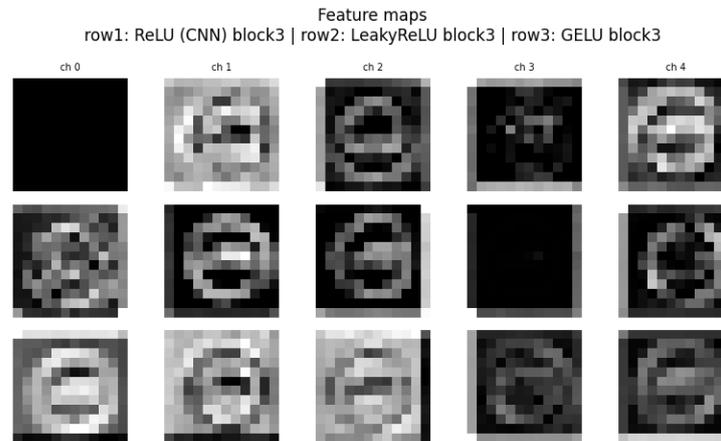


Figure 5. Visualization of Layer 3 feature maps for three activation functions: ReLU (top row), Leaky ReLU (middle row), and GELU (bottom row). Columns represent different channels (ch0–ch4). The visualizations show representative channels extracted from the convolutional layer using the same input image. Feature maps are normalized before visualization to highlight differences in activation patterns between the activation functions.

Figure 6 shows that the feature maps have been reduced to a spatial resolution of 6×6 . This compresses the representations and makes the differences between activation functions more visible. At this point, ReLU makes the feature maps the least dense. Many channels are weakly activated or completely inactive, resulting in large dark regions. This is caused by the hard threshold of ReLU, which removes all negative responses. At very small spatial resolutions, this behaviour leads to substantial information loss. Leaky ReLU keeps more information than ReLU because it allows small negative values to remain. The feature maps are less empty and show clearer differences between regions. Because of this, the network can still show small but significant trends in the feature maps instead of making them completely dark. GELU produces the most balanced feature maps even when the spatial size is very small. The activations change smoothly instead of being sharply cut off, which helps to keep small but important differences. Because of this, GELU shows that they are better at mathematical abstraction than ReLU and Leaky ReLU.

Figure 7 shows how activations from the fc1 layer are distributed for different activation functions. ReLU produces sparse activations, with many values equal to zero. This means that a large number of neurones are inactive, which limits the amount of information passed through the layer. Leaky ReLU reduces this sparsity by allowing small negative values. Hence, fewer activations are exactly zero and more information is able to flow through the layer compared to ReLU. GELU produces a smoother and more continuous distribution of activations, with both positive and negative values. This indicates that information is preserved more effectively and that neurone activity is more balanced than with ReLU and Leaky ReLU.

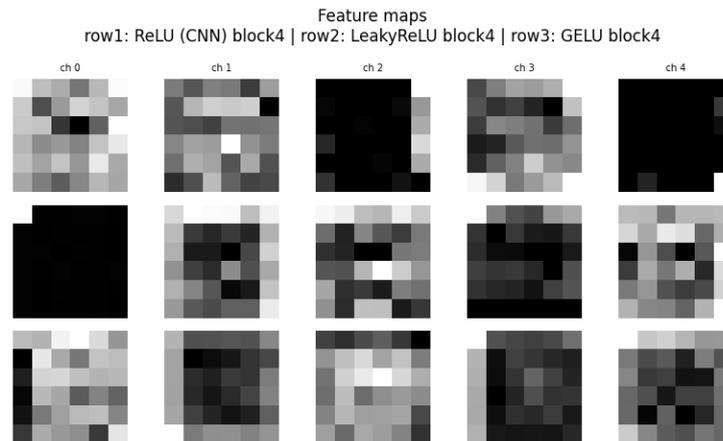


Figure 6. Visualization of Layer 4 feature maps for three activation functions: ReLU (top row), Leaky ReLU (middle row), and GELU (bottom row). Columns represent different channels (ch0–ch4). The feature maps show high-level spatial representations learned by the network. Channels are selected for visualization, and all feature maps are normalized to ensure consistent comparison between activation functions.

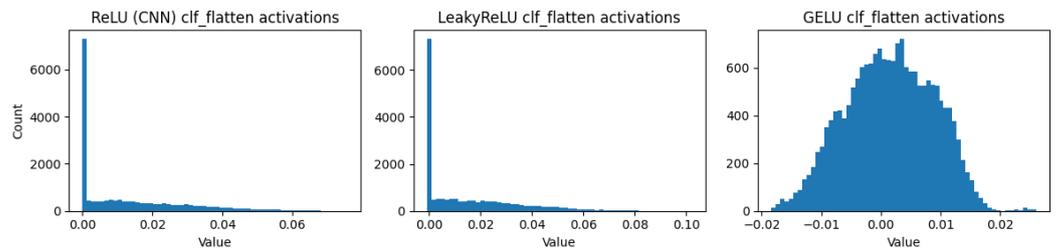


Figure 7. Histogram visualization of activations from the flatten layer for three activation functions: ReLU (left), Leaky ReLU (middle), and GELU (right). The activations are obtained using the same input sample from the GTSRB dataset, illustrating how different activation functions affect the activation distribution before the fully connected classifier layer.

5. Results

The experimental results are summarised in Table 4, which presents a comparison between previous traffic sign recognition approaches and the proposed CNNs using different activation functions. Earlier studies (2011–2018) report test accuracies ranging from 96.14% to 99.71%, demonstrating steady progress in traffic sign recognition performance. More recent methods (2024–2025) also achieve competitive results, with test accuracies up to 97.75%.

For the proposed method, three activation functions (ReLU, Leaky ReLU, and GELU) were evaluated using the same CNNs architecture. Among them, Leaky ReLU achieved a test accuracy of 99.60%, while ReLU slightly improved performance with 99.61%. The GELU activation function outperformed both the ReLU and Leaky ReLU variants, achieving 99.98% training accuracy, 99.99% validation accuracy, and a test accuracy of 99.75%, along with higher precision, recall, and F1-score values. Compared with the ReLU and Leaky ReLU baselines, this corresponds to absolute improvements of approximately 0.14% and 0.15% in test accuracy, respectively. These results indicate that the choice of activation function has a significant impact on the performance of the model, even when the network architecture remains unchanged.

The confusion matrix in Figure 8 illustrates the classification performance of the proposed model in all 43 traffic sign classes. Most predictions are concentrated along the central line, indicating that the majority samples are correctly classified. Only a small number of

incorrectly recognised signs appear, representing minor misclassification between visually similar traffic sign categories.

Table 4. Comparison of reported performance metrics for traffic sign recognition methods across different years, including accuracy-based results from prior studies and activation-function experiments conducted in this work.

Year	Training Accuracy	Validation Accuracy	Precision	Recall	F1	Test Accuracy
2011 [28]	-	-	-	-	-	98.31
2011 [29]	-	-	-	-	-	96.14
2012 [30]	-	-	-	-	-	99.46
2012 [25]	-	-	-	-	-	98.84
2017 [31]	-	-	-	-	-	98.87
2018 [1]	-	-	99.71	99.71	99.71	99.71
2024 [18]	-	-	-	-	-	97.75
2024 [32]	99.90	97.98	-	-	-	-
2025 [21]	98.70	99.70	96	97	96	97
2025 [20]	97.5	-	-	-	-	-
ReLU	99.96 ¹	99.99 ²	99.62	99.61	99.61	99.61
LeakyReLU	99.97 ¹	99.99 ²	99.61	99.60	99.60	99.60
GELU	99.98 ¹	99.99 ²	99.76	99.75	99.75	99.75

¹ Training accuracy: Mean of the training accuracies obtained separately on GTSRB_Final_Training_Images and GTSRB_Training_fixed. ² Validation accuracy: Mean of the validation accuracies obtained separately on GTSRB_Final_Training_Images and GTSRB_Training_fixed.

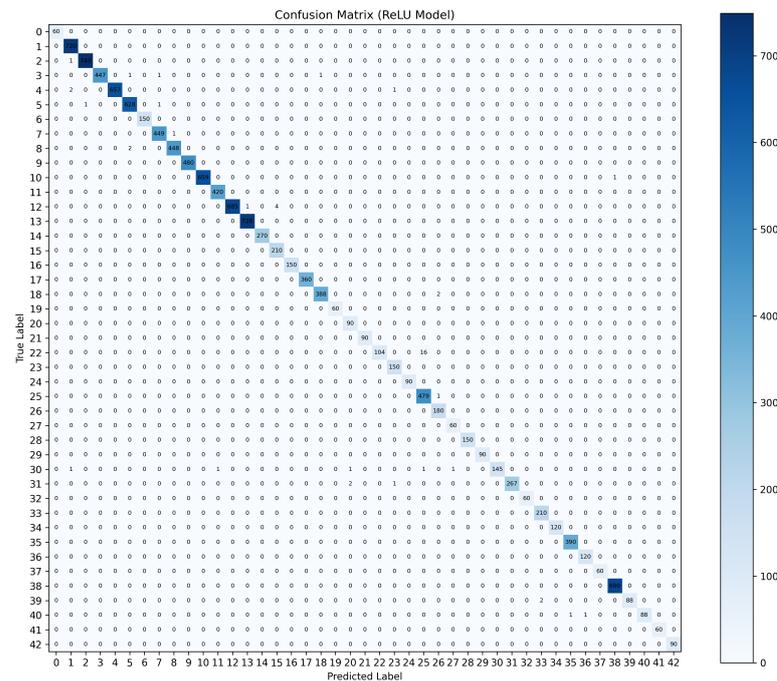


Figure 8. Confusion matrix of the ReLU-based model on the GTSRB test dataset. The rows represent the true class labels and the columns represent the predicted class labels. Color intensity in the bar chart indicates the number of samples.

Figure 9 presents the confusion matrix obtained with the LeakyReLU activation function. The matrix shows that most traffic sign samples are assigned to their corresponding classes. Only a few values appear outside the correct class positions, indicating that misclassification occurs in a limited number of cases.

Figure 10 presents the confusion matrix obtained using the GELU activation function. The results show that most of the traffic sign images are correctly classified in the

43 categories. Compared with the previous activation functions, GELU provides improved classification performance, with fewer misclassified samples. The results obtained from the confusion matrices show that all three activation functions provide strong classification performance. However, the model using the GELU activation function achieves slightly better accuracy and fewer misclassification compared to the ReLU and LeakyReLU models.

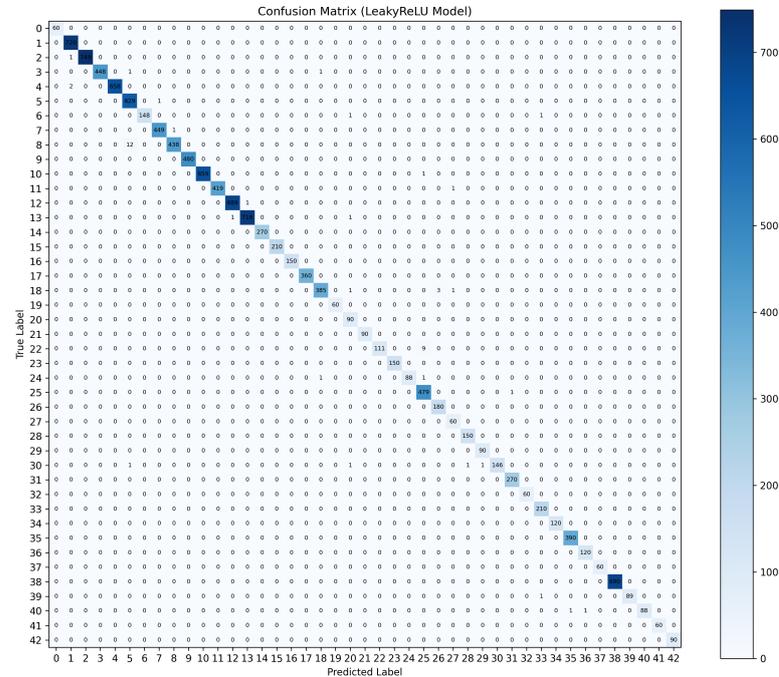


Figure 9. Confusion matrix of the LeakyReLU-based model on the GTSRB test dataset. The rows represent the true class labels and the columns represent the predicted class labels. Color intensity in the bar chart indicates the number of samples.

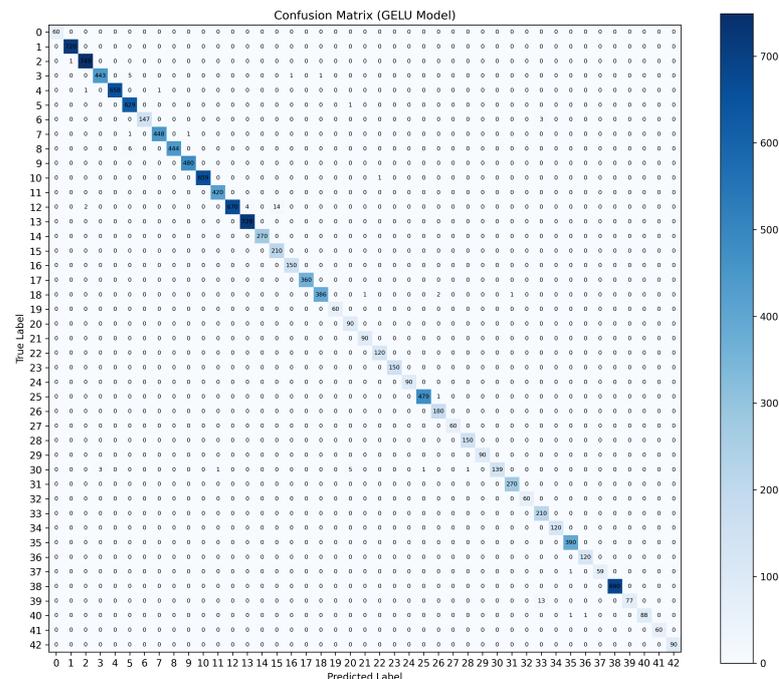


Figure 10. Confusion matrix of the GELU-based model on the GTSRB test dataset. The rows represent the true class labels and the columns represent the predicted class labels. Colour intensity in the bar chart indicates the number of samples.

6. Conclusions

This study investigated the effect of different activation functions on a fixed CNNs architecture for traffic sign recognition. By maintaining the same network structure and training configuration, performance differences could be directly attributed to the activation function. The experimental results demonstrate that GELU consistently outperforms ReLU and Leaky ReLU in terms of accuracy, precision, recall, and F1-score. While ReLU-based models provide acceptable performance, the smoother non-linear behaviour of GELU enables better feature representation and generalisation. As observed from the result table and the confusion matrix analysis, the GELU-based model produces fewer misclassification compared to the other activation functions. In general, the findings confirm that the selection of the activation function plays a crucial role in CNNs-based traffic sign recognition systems, and GELU is a strong candidate for improving classification performance in such tasks.

Author Contributions: Conceptualization, H.M., M.R.S. and A.S.Y.; methodology, A.S.Y.; software, A.S.Y.; validation, A.S.Y.; formal analysis, H.M. and A.S.Y.; investigation, A.S.Y.; resources, A.S.Y.; data curation, A.S.Y.; writing—original draft preparation, A.S.Y.; writing—review and editing, H.M., M.R.S. and A.S.Y.; visualization, A.S.Y.; supervision, H.M.; project administration, A.S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this study are from the German Traffic Sign Recognition Benchmark (GTSRB), a publicly available multi-class image classification benchmark. The dataset is available at https://benchmark.ini.rub.de/gtsrb_dataset.html (accessed on 10 October 2025).

Acknowledgments: Ahmet Serhat Yildiz's Ph.D. is sponsored by the Ministry of National Education of Türkiye.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EVs	Electric Vehicles
CNNs	Convolutional Neural Networks
ITS	Intelligent Transportation Systems
GTSRB	German Traffic Sign Recognition Benchmark
STN	Spatial Transformer Networks
ReLU	Rectified Linear Unit
Leaky ReLU	Leaky Rectified Linear Unit
GELU	Gaussian Error Linear Unit
BN	Batch Normalization
MaxPool	Max Pooling

References

1. Arcos-García, Á.; Alvarez-Garcia, J.A.; Soria-Morillo, L.M. Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods. *Neural Netw.* **2018**, *99*, 158–165. [[CrossRef](#)] [[PubMed](#)]
2. He, Z.; Xiao, Z.; Yan, Z. Traffic sign recognition based on convolutional neural network model. In *Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020*; IEEE: Piscataway, NJ, USA, 2020; pp. 155–158.
3. Haque, W.A.; Arefin, S.; Shihavuddin, A.; Hasan, M.A. DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements. *Expert Syst. Appl.* **2021**, *168*, 114481. [[CrossRef](#)]
4. Bi, Z.; Yu, L.; Gao, H.; Zhou, P.; Yao, H. Improved VGG model-based efficient traffic sign recognition for safe driving in 5G scenarios. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 3069–3080. [[CrossRef](#)]

5. Fan, S. Road sign detection and recognition system based on multi-layers convolutional neural network model trained with German Traffic Sign Recognition Benchmark. In *Proceedings of the 2021 4th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, Online, 12–14 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 395–399.
6. Bhatt, N.; Laldas, P.; Lobo, V.B. A real-time traffic sign detection and recognition system on hybrid dataset using CNN. In *Proceedings of the 2022 7th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 28–30 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1354–1358.
7. Mishra, J.; Goyal, S. An effective automatic traffic sign classification and recognition deep convolutional networks. *Multimed. Tools Appl.* **2022**, *81*, 18915–18934. [[CrossRef](#)]
8. Zheng, Y.; Jiang, W. Evaluation of vision transformers for traffic sign classification. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 3041117. [[CrossRef](#)]
9. Nadeem, Z.; Khan, Z.; Mir, U.; Mir, U.I.; Khan, S.; Nadeem, H.; Sultan, J. Pakistani traffic-sign recognition using transfer learning. *Multimed. Tools Appl.* **2022**, *81*, 8429–8449. [[CrossRef](#)]
10. Abougarair, A.; Elmaryul, M.; Aburakhis, M. Real time traffic sign detection and recognition for autonomous vehicle. *Int. Robot. Autom. J.* **2022**, *8*, 82–87. [[CrossRef](#)]
11. Lim, X.R.; Lee, C.P.; Lim, K.M.; Ong, T.S. Enhanced traffic sign recognition with ensemble learning. *J. Sens. Actuator Netw.* **2023**, *12*, 33. [[CrossRef](#)]
12. Kunte, O.; Reddy, S. Optimizing Traffic Sign Recognition: Custom CNN vs. Pretrained Models. In *Proceedings of the International Conference on Computer Communication and Informatics (ICCCI)*, Okayama, Japan, 26–28 June 2023; IEEE: Piscataway, NJ, USA, 2023.
13. Sharma, V.; Kumar, V.; Aditya, H. Traffic Sign Detection and Classification. In *Proceedings of the 2023 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI)*, Gwalior, India, 10–11 December 2023; IEEE: Piscatway, NJ, USA, 2023; pp. 1–6.
14. Islam, M.S.; Pias, M.M.; Tasnim, N.; Hashan, R.; Uddin, J.; Al Mahmud, T.H. Advancing Traffic Sign Detection and Recognition using Optimized Convolutional Neural Network. In *Proceedings of the 2024 IEEE International Conference on Computing, Applications and Systems (COMPAS)*, Chattogram, Bangladesh, 25–26 September 2024; IEEE: Piscatway, NJ, USA, 2024; pp. 1–6.
15. Zhan, B.; Temirgazyeva, S. Development of Deep Convolutional Neural Network for Road Sign Detection. In *Proceedings of the 2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST)*, Astana, Kazakhstan, 15–17 May 2024; IEEE: Piscatway, NJ, USA, 2024; pp. 308–313.
16. Singh, G.; Kumar, A.; Bajaj, P.; Verma, M. Enhancing Road Safety: Convolutional Neural Networks in Traffic Sign Recognition. In *Proceedings of the 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAIC)*, Salem, India, 5–7 June 2024; IEEE: Piscatway, NJ, USA, 2024; pp. 65–72.
17. Rubio, A.; Demoor, G.; Chalmé, S.; Sutton-Charani, N.; Magnier, B. Sensitivity Analysis of Traffic Sign Recognition to Image Alteration and Training Data Size. *Information* **2024**, *15*, 621. [[CrossRef](#)]
18. Li, J.; Chen, Y.; Lin, H.; Chen, F.; Yang, B. Traffic Sign Recognition Method Based on Convolutional Neural Network. In *CICTP 2024*; ASCE Library: Reston, VA, USA, 2024; pp. 490–499.
19. PP, H.K.; Ravindran, S.; Vijean, V. Traffic Sign Classification for Road Safety using CNN. In *Proceedings of the 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC)*, Bhubaneswar, India, 9–10 February 2024; IEEE: Piscatway, NJ, USA, 2024; pp. 462–466.
20. Janu, N.; Shrotriya, N.; Goel, A. Intelligent Road Sign Recognition and Autonomous Driving. In *Proceedings of the 2025 International Conference on Computing Technologies (ICOCT)*, Bengaluru, India, 13–14 June 2025; IEEE: Piscatway, NJ, USA, 2025; pp. 1–8.
21. Gul, H.; Gul, M. Traffic Sign Recognition Using a Customized Convolutional Neural Network. *Mach. Algorithms* **2025**, *4*, 57–67. [[CrossRef](#)]
22. Agarap, A.F. Deep learning using rectified linear units (relu). *arXiv* **2018**, arXiv:1803.08375.
23. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, Atlanta, GA, USA, 17–19 June 2013; Volume 30, p. 3.
24. Hendrycks, D. Gaussian Error Linear Units (Gelus). *arXiv* **2016**, arXiv:1606.08415.
25. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.* **2012**, *32*, 323–332. [[CrossRef](#)] [[PubMed](#)]
26. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, Lille, France, 7–9 July 2015; pp. 448–456.
27. Rao, M.; Tang, P.; Zhang, Z. A developed siamese CNN with 3D adaptive spatial-spectral pyramid pooling for hyperspectral image classification. *Remote Sens.* **2020**, *12*, 1964. [[CrossRef](#)]
28. Sermanet, P.; LeCun, Y. Traffic sign recognition with multi-scale convolutional networks. In *Proceedings of the 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, 31 July–5 August 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 2809–2813.
29. Zaklouta, F.; Stanculescu, B.; Hamdoun, O. Traffic sign classification using kd trees and random forests. In *Proceedings of the 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, 31 July–5 August 2011; IEEE: Piscatway, NJ, USA, 2011; pp. 2151–2155.

30. Dan, C.; Ueli, M.; Jonathan, M.; Jürgen, S.H. Multi-column deep neural network for traffic sign classification. *Neural Netw.* **2012**, *32*, 333–338.
31. Gecer, B.; Azzopardi, G.; Petkov, N. Color-blob-based COSFIRE filters for object recognition. *Image Vis. Comput.* **2017**, *57*, 165–174. [[CrossRef](#)]
32. Ferencz, C.; Zöldy, M. Neural network-based multi-class traffic-sign classification with the German traffic sign recognition benchmark. *Acta Polytech. Hung.* **2024**, *21*, 203–220. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.