electronics

Article

A Lightweight Convolutional Neural Network (CNN) Architecture for Traffic Sign Recognition in Urban Road Networks

Muneeb A. Khan [1], Heemin Park [1] and Jinseok Chae [2,*]

[1] Department of Software, Sangmyung University, Cheonan 31066, Republic of Korea; muneebkhan046@gmail.com (M.A.K.); heemin@smu.ac.kr (H.P.)
[2] Department of Computer Science and Engineering, Incheon National University, Incheon 22012, Republic of Korea
* Correspondence: jschae@inu.ac.kr

Abstract: Recognizing and classifying traffic signs is a challenging task that can significantly improve road safety. Deep neural networks have achieved impressive results in various applications, including object identification and automatic recognition of traffic signs. These deep neural network-based traffic sign recognition systems may have limitations in practical applications due to their computational requirements and resource consumption. To address this issue, this paper presents a lightweight neural network for traffic sign recognition that achieves high accuracy and precision with fewer trainable parameters. The proposed model is trained on the German Traffic Sign Recognition Benchmark (GTSRB) and Belgium Traffic Sign (BelgiumTS) datasets. Experimental results demonstrate that the proposed model has achieved 98.41% and 92.06% accuracy on GTSRB and BelgiumTS datasets, respectively, outperforming several state-of-the-art models such as GoogleNet, AlexNet, VGG16, VGG19, MobileNetv2, and ResNetv2. Furthermore, the proposed model outperformed these methods by margins ranging from 0.1 to 4.20 percentage point on the GTSRB dataset and by margins ranging from 9.33 to 33.18 percentage point on the BelgiumTS dataset.

Keywords: traffic sign recognition; intelligent transportation system; convolutional neural network; German traffic sign recognition benchmark dataset; Belgium traffic sign benchmark dataset

Citation: Khan, M.A.; Park, H.; Chae, J. A Lightweight Convolutional Neural Network (CNN) Architecture for Traffic Sign Recognition in Urban Road Networks. *Electronics* **2023**, *12*, 1802. https://doi.org/10.3390/electronics12081802

Academic Editor: José Santa

Received: 8 March 2023
Revised: 6 April 2023
Accepted: 8 April 2023
Published: 11 April 2023

1. Introduction

Traffic signs play a vital role in regulating and maintaining traffic flow, ensuring road safety, and helping drivers. According to the World Health Organization (WHO) report on the Global Status Report on Road Safety, more than 1.35 million people died from road accidents in 2018 [1]. As a result, traffic sign recognition and classification have gained a lot of attention from the research and scientific community due to their various applications, such as autonomous driving, driver assistance, and behavior monitoring. Advances in Artificial Intelligence and Machine Learning (AI&ML) have led to remarkable progress in object detection and classification with high precision and accuracy [2,3]. In recent years, multiple frameworks have been proposed by leveraging AI&ML base computer vision techniques to design a robust and reliable traffic sign recognition system (TSRs) [4–7].

In intelligent transportation systems (ITS), the classification of traffic signs is an essential task for designing a reliable autonomous self-driving and driver assistance system (DAS) due to the complexity of the external environment. The DAS can help people in real-time by directing and alerting drivers, reducing road accidents, and improving road safety. However, the classification of real-time traffic signs is much more complex than any other computer vision task [8]. Some major challenges of traffic sign classification systems are: (i) traffic signs may be similar within or between classes, (ii) frequency variations between classes may need to be balanced, (iii) the size of the traffic signs could change

significantly, (iv) partial occlusions can make signs incomplete when traffic is not visible, (v) variable lighting can cause color changes due to weather conditions, and (vi) images collected from moving vehicles can be blurred.

Countries often have different traffic sign designs, which makes it an extensive and tiresome task to obtain different traffic sign data and label them respectively for each country. Typically, most of the research is carried out on publicly available datasets such as the German Traffic Sign Recognition Benchmark (GTSRB) [9], the Belgium Traffic Sign (BelgiumTS) dataset [10], the Italian Traffic Signs dataset (DITS) [11], and the Chinese Traffic Sign Database (CTSD) [12]. Therefore, designing a framework that can generalize correct detection and reliable real-time TSRs remains a challenging task.

The primary goal of TSR is to extract relevant visual information from input images, perform efficient segmentation, and achieve accurate classification of traffic signs. Most of the proposed TSR frameworks employ extreme learning machine (ELM) [13] or support vector machine (SVM) [14,15]-based methods with hand-crafted features for traffic sign classification (TSC). However, when the image is blurred or occluded, it suffers a significant information loss, or its classification rate is drastically reduced. In [16], a hybrid kernel-based ELM model was proposed to be integrated with deep perceptual features to reduce computational overhead while maintaining a good traffic signal classification rate. The authors of [17] proposed a hybrid multi-layer perceptron (MLP) framework integrated with a template matching algorithm to improve the feature extraction process and overall model precision. In [18], a field programmable gate array (FPGA)=based hardware integrated with the template matching algorithm is proposed for real-time TSR.

Several convolutional neural network (CNN)-based frameworks have been proposed for TSR applications. In [19], a single-branch CNN architecture was proposed for TSR with a higher accuracy rate compared to traditional methods. MicronNet [20], a compact deep CNN architecture, achieves an accuracy of 92.3% using spectral macroarchitecture augmentation and parameter precision optimization techniques for real-time TSR. In [21], the authors utilized ELM as a classifier in combination with feature extraction from CNNs, thus leveraging the strengths of both deep learning and conventional machine learning techniques. The authors leveraged the benefits of both deep learning and conventional machine learning by utilizing ELM as a classifier in conjunction with feature extraction from CNNs. In [22], a hybrid CNN integrated with the MLP base framework was proposed to improve the overall accuracy of the TSR, while a multi-CNN ensemble classifier-based approach was proposed for efficient TSR [23,24]. Although deep CNN-based frameworks performed well in TSR applications due to their strong representational ability, this emerges from the millions of trainable parameters, i.e., ResNet has 40.3 million parameters, while VGG16 and VGG 19 have 134 and 138 million parameters, respectively.

However, these deep CNN-based frameworks are not optimal for practical applications due to excessive computational requirements, storage, and power needs. The design of a lightweight and robust TSR framework with fewer trainable parameters and lower computational costs is essential to achieve accuracy and speed requirements for practical TSRs implementations.

The computational requirements and resource consumption of an AI&ML base TSRs are crucial in an automotive system, where traffic sign recognition must be performed in real-time to ensure safety. A delay caused by excessive computational requirements or resource consumption could have catastrophic consequences. The size and complexity of the model directly impact the system's memory requirements and processing time, affecting the system's performance. Therefore, an efficient TSR system must reduce the number of parameters and operations while maintaining accuracy.

Our research objective is to develop a CNN framework that is both lightweight and robust, computationally efficient, and resource-effective, to address challenges associated with real-world TSR implementations. Our primary focus is to minimize the number of trainable parameters and reduce computational overhead (resource usage), both of

which are critical for the effective deployment of TSRs on embedded devices with limited computing power and memory. The main contributions are summarized as follows.

1. An efficient CNN-based framework has been proposed for accurate traffic sign recognition, which addresses the computational constraints of existing systems. By optimizing the number of convolutional layers and filter size, the proposed framework reduces the number of trainable parameters and computational overhead. A systematic approach such as a grid search was employed to optimize the number of convolutional layers and filter sizes, resulting in the best performance with the fewest number of trainable parameters. The framework was designed to balance the complexity and computational efficiency of the model while maintaining high accuracy.
2. The proposed framework addresses to the computational limitations of current TSR systems, making it more accessible and affordable for deployment in automotive systems with limited computational resources. Its minimal computational requirements and inference time enable it to operate efficiently and improve road safety, thereby reducing the incidence of accidents.
3. To tackle the challenges posed by variations in traffic sign appearance, such as changes in lighting conditions, scale, perspective, viewing angle, blur, and shadow, as well as intra-category appearance variations and low inter-category variations, a variety of convolutional approaches have been explored. Furthermore, extensive research has been conducted on optimization strategies and design principles for Convolutional Neural Networks (CNNs) to design efficient TSRs.
4. The proposed framework has been evaluated using two benchmark real-world datasets, GTSRB and BelgiumTS, which has provided confidence in its ability to operate effectively in practical settings. These evaluations provide a comprehensive understanding of the proposed system capabilities, enabling confidence in its potential for real-world deployment.
5. A comprehensive comparative analysis with several state-of-the-art models is presented, such as GoogleNet, AlexNet, VGG16, VGG19, MobileNetv2, and ResNetv2. The evaluation parameters for comparison include accuracy, precision, recall, and f1-score. The analysis was conducted in detail and provides valuable insights into the performance of the proposed framework.
6. Overall, the contribution of this research paper is significant as it helps to improve road safety by developing a practical and efficient TSR system that can operate in low-resource settings.

The remainder of this paper is structured as follows. In Section 2, we present the relevant literature work in this field. A detailed discussion on the dataset, pre-processing techniques, and the design and implementation of the proposed methods is discussed indetail in Section 3. The evaluation of our framework is presented in Section 4, followed by a brief discussion on the results in Section 5. Finally, the conclusion is presented in Section 6.

2. Related Work

DAS and automated TSRs have been hot research topics in the last decade. As a result, multiple frameworks have been proposed for efficient TSRs in complex and resource-constrained environments [25]. These proposed frameworks can be divided into traditional and AI&ML-based methods, respectively. This section discusses the traditional and latest deep learning-based frameworks proposed in this area.

The main objective of TSRs is to extract useful visual information about the traffic sign from input images, efficient segmentation, and accurate classification of traffic signs [26]. Traditional ML base frameworks require salient features such as the traffic sign's color and shape to improve the TSRs accuracy and precision. These frameworks utilize template-matching techniques (using similar and dissimilar visual features among images) for TSR. In addition to that, they often opt for specific features, such as the histogram of oriented gradients (HOG), the scale-invariant feature transform (SIFT), or Haar-like features, for the

training and classification of traffic signs [27]. However, blurred images, partial occlusion, deformed or warped images, and similarities between multiple traffic sign classes are still among the biggest challenges to these frameworks [28,29].

The authors in [30] developed a model to detect traffic signs in adverse weather conditions and dirty camera lenses but the accuracy was limited to 80%. In [31], an amalgam of the semantic segmentation model (Seg-Net) integrated with U-Net was proposed for TSR. The proposed SegU-Net model outperformed conventional methods such as the Deep Neural Network and Spatial Transformer Networks, and achieved a precision score of 95.29% on the German Traffic Sign Detection Benchmark (GTSDB) dataset. In [20], a compact and highly optimized neural network called MicroNet was presented for embedded real-time TSR. MicroNet achieved a top-1 accuracy of 98.0% on the GTSRB benchmark dataset. The authors of [32] introduced an enhanced version of Cascade R-CNN called the Attribute-Refinement Cascaded CNN for detecting traffic signs. This two-stage algorithm is computationally intensive but highly effective in detecting traffic signs. In [33], the authors discussed the problems affecting the detection and recognition of Chinese characters on traffic signs. The main limitation of this method is that occlusion has a negative impact on the accuracy of recognizing Chinese characters, resulting in some characters being either not recognized or recognized incorrectly.

Neural network-based frameworks use multilayer neural networks for feature extraction and training, which is useful for image processing. In [34], a multi-task convolutional neural network (MCNN)-based framework was proposed for TSR using region-of-interest, while a multi-feature fusion enhancement base framework was presented to detect and classify the traffic sign. In the pre-processing phase, a guided filter was implemented on the GTSRB dataset in [35] to reduce ghosting artifacts, occlusion, and blurry effects. The processed images were forwarded to the CNN base model to train. In [36], a hierarchical-CNN (HCNN) architecture for GTSRB has been proposed, which clusters the GTSRB signs into new subsets (families) using a CNN-oriented approach. The new clustering improves the correct classification rate, as it is based on CNNs. The goal is to create a more efficient and effective architecture for TSR.

A cascade-saccade network is proposed using a class hierarchy structure for real-time detection and classification of the traffic sign. The proposed model achieves an accuracy of 89.7% on the Chinese Traffic Sign Database [37]. A CNN-based two-stage model is presented to classify traffic signs in real-time [38]. In the first step, the model detects the location and shape of the traffic signs, while the second step is used for the TSR. The proposed framework achieves an accuracy of 97.22% and 86.8% in GTSDB and GTSRB, respectively. These multi-stage base TSR algorithms show promising results, but require high-performance hardware and computation cost. In [39], a deep learning model was presented using a CNN-based Refined Mask R-CNN (RM R-CNN) for TSR. The model was refined in architecture and data augmentation, achieving a precision of 97.08%, outperforming conventional models such as Fast R-CNN and Mask R-CNN.

Meanwhile, the single-stage TSR framework, such as You Only Look Once (YOLO) and the Single Shot Multibox Detector (SSD), eliminates the region-of-interest detection step and directly performs regression and classification of traffic signs [40,41]. A comparative performance analysis between YOLOv5 and SSD was performed for TSR [42]. YOLOv5 outperforms SSD with accuracy and mean precision of 97.70% and 90%, respectively. The authors of [43] developed a methodology to evaluate the performance of GPUs and TPUs for TSDR using the CNN base model such as ResNet50 and MobileNet combined with SSD and Feature Pyramid Network (FPN). The results illustrate that the TPU processing time is 16 times faster and outperforms GPU in terms of precision, recall, and f1-score. The aim of [44] was to improve the detection accuracy and speed for TSRs using YOLO, a CNN-based approach, that focuses on different regions of interest in high-definition images, but there is a risk of selecting regions without any traffic signs. A novel approach to object detection has been proposed in [45], which addresses the limitations of traditional SSD methods that lack interaction between multi-level feature maps. The proposed method,

enhanced SSD, utilizes a parallel structure that enables interaction between multi-level feature maps and maintains high detection accuracy. Unlike traditional SSD methods that only use feature maps for object prediction, the enhanced SSD proposed in this study achieves better performance by incorporating a novel interaction mechanism between low-level and high-level feature maps.

3. Methodology

In this section, we will provide a detailed and comprehensive discussion of the proposed methodology and its implementation. First, we will focus on the dataset and the preprocessing techniques that were used to clean and transform it. Subsequently, we will delve into the methodology to provide a thorough understanding of the framework.

The methodology of this study can be divided into three main components, as illustrated in Figure 1. First, traffic sign images were collected and used as the dataset. In the first block, the images were pre-processed using various methods to improve the TSR. Next, the pre-processed data were forwarded into the training phase, where the proposed model was trained on the training set with optimized hyperparameters. Finally, the performance of the model was evaluated in the testing set to determine its overall performance. Figure 1 provides a visual representation of the workflow and the different stages involved in the process, facilitating a better understanding of the methodology used in this study.

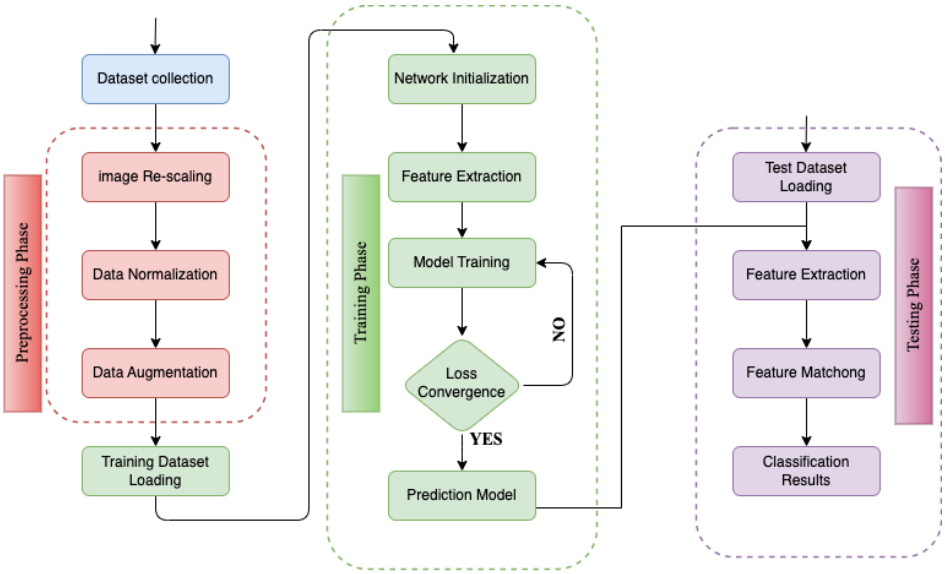

Figure 1. Workflow of the proposed system.

3.1. Dataset

This study utilizes two well-known publicly available datasets, GTSRB [9] and BelgiumTS [10], which are multiclass benchmark datasets containing a diverse range of challenging and complex traffic signs. The datasets comprise traffic signs with various complexities, such as tilt, uneven lighting, distortion, occlusion, and similar background colors, simulating real-world scenarios. The model's effectiveness was thoroughly evaluated by presenting a wide range of complex and hard-to-distinguish traffic signs from these datasets.

3.1.1. GTSRB [9]

The German Traffic Sign Recognition Benchmark (GTSRB) dataset comprises 51,922 images saved in the Portable Network Graphics (PNG) format. The dimensions of the images in this dataset range from 15 × 15 pixels to 250 × 250 pixels. The GTSRB dataset was divided into three groups to train, validate, and test the model: 31,433 images for training, 7859 images for validation, and 12,630 images for testing. It consists of 43 distinct

categories (as shown in Figure 2), such as prohibitory signs, danger signs, and mandatory signs. Each category includes 100 and 1000 images, providing a comprehensive and diverse range of images to train and test the model's ability to recognize various traffic signs.

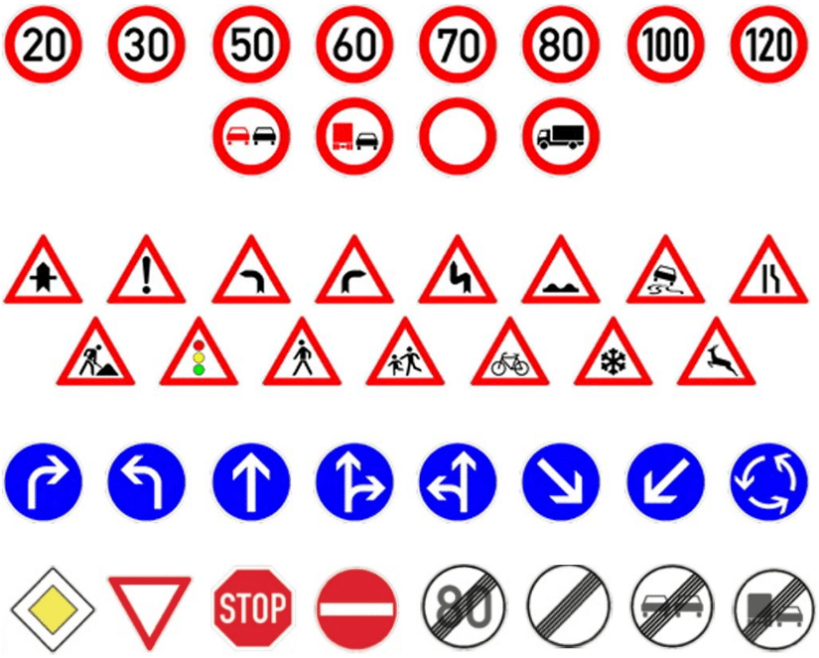

Figure 2. German traffic sign recognition benchmark (GTSRB) dataset.

3.1.2. BelgiumTS [10]

The Belgium Traffic Sign (BelgiumTS) dataset comprises 7095 images, saved in Portable Pixmap (ppm) format. This dataset includes 4575 images used for training and 2520 images for testing. The dimensions of the images in BelgiumTS vary from 11 × 10 pixels to 562 × 438 pixels. In contrast to the GTSRB dataset, the BelgiumTS consists of 62 different traffic sign classes (as shown in Figure 3). However, this dataset contains fewer training samples than the GTSRB dataset, which can make classification more challenging.

Figure 3. Belgium traffic sign (BelgiumTS) dataset.

3.2. Data Pre-Preocessing

The data pre-processing stage comprises three steps: image rescaling, normalization, and data augmentation.

3.2.1. Image Re-Scaling

The datasets used in this study (GTSRB and BelgiumTS) include images with aspect ratios ranging from 15 × 15 to 250 × 250 pixels and 11 × 10 to 562 × 438 pixels, respectively. In order to be compatible with neural networks, it is necessary to have fixed image sizes. Furthermore, it is important to mention that reducing the image size to a lower pixel ratio,

such as 80 × 80 or 50 × 50, would reduce the model's complexity. However, this may have negatively affected the model's ability to represent visual information accurately and potentially reduced classification performance.

In our experiments, we tested our model with different image sizes and found that 100 × 100 pixels provides the optimal trade-off between computational complexity and classification accuracy. Thus, we resized our images to 100 × 100.

3.2.2. Data Normalization

Data normalization is critical to ensure a consistent distribution of input parameters (pixel values), allowing faster convergence during network training. To achieve this, the mean value of each pixel is subtracted, and the result is divided by its standard deviation, resulting in a Gaussian distribution centered at zero.

3.2.3. Data Augmentation

Data augmentation is an important step to address the imbalance in the dataset, where some labels have a large number of images compared to others. This method creates additional data from the existing samples by applying transformations such as flips, zooming, rotation, and brightness adjustments. In this paper, the Keras library, ImageDataGenerator function, is opted to perform data augmentation. This function allows us to adjust parameters such as the rescale factor, height and width shift range, rotation range, horizontal and vertical flips, and others, to generate synthetic data that can improve model training.

3.3. Proposed Methodology

The rise of deep learning has prompted the computer vision research community to strive for better accuracy in tasks such as image classification. As a result, we have seen the emergence of deeper architectures, such as [46], which has 23.8 million parameters, and [47] has 143.6 million parameters. However, simply increasing depth does not always lead to improved accuracy [48,49]. In fact, using too many kernels can actually cause performance degradation without mitigating the high variance-high bias problem. The recursive process of learning new features based on previously learned ones is not always optimal. In response to the challenge of balancing depth and accuracy, a lightweight neural network architecture is proposed for traffic sign recognition that achieves higher accuracy and precision while utilizing fewer trainable parameters.

The proposed framework utilizes a single CNN that incorporates a variety of layers, including convolutional, max-pooling, and batch normalization layers (as shown in Figure 4). These layers serve the purpose of extracting features by converting the raw pixel information from the input image into a tensor. This process enables the model to identify patterns and significant features within the image. The tensor was then classified into a specific traffic sign category using the global average pooling layer. In addition, the variable parameters of all these layers were optimized together by minimizing the misclassification error across the training set.

Convolutional layers are fundamental components of neural networks that convolve each of their $n-1$ input maps with a two-dimensional filter of size $\mathbb{F}_x^n \times \mathbb{F}_y^n$, where x and y represent the input dimensions. Each convolutional layer comprises neurons with learnable biases and weights that allow the layer to learn and adapt over time. During the feedforward process, $\mathbb{F}_x$ and $\mathbb{F}_y$ convolved across the height and width of the input maps, generating a two-dimensional activation map of the filter through a dot product. The output map of the n layer was generated by summing the convolutional responses of the $n-1$ layers.

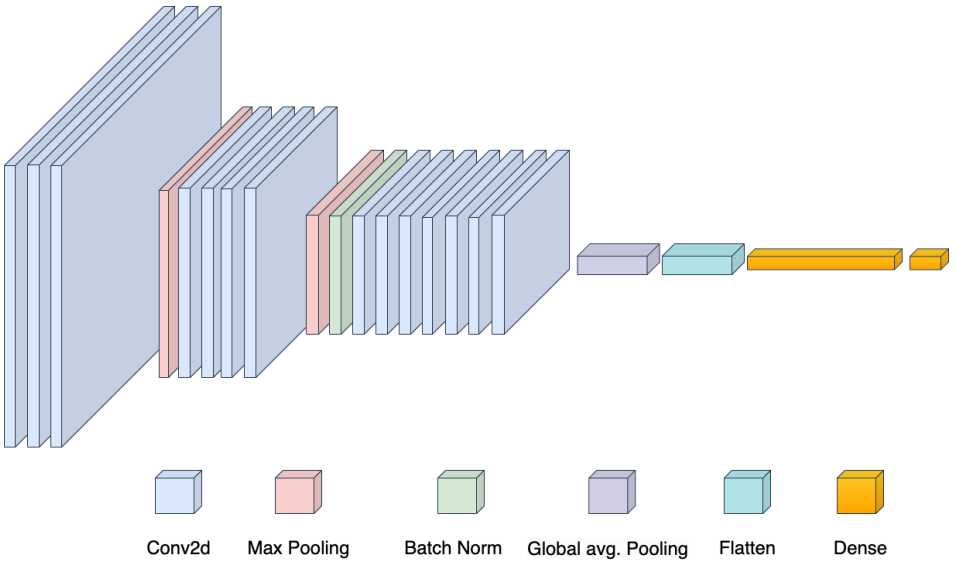

Figure 4. Proposed methodology.

The weights ω_j^i represent a filter of size $\mathbb{F}_x^n \times \mathbb{F}_y^n$, which connects the input map ($\imath$) and the output maps ($\jmath$), while $\beta_\jmath$ represents the bias of the output map. The primary purpose of convolutional layers is to extract and detect specific features from the input maps and to create higher-level abstractions of these features by combining filter activations using Equation (1).

$$\mathbb{R}(z) = \sum_{\imath=1}^{n-1} \alpha_\imath^{n-1} * \omega_{\imath\jmath}^n + \beta_\jmath^n \tag{1}$$

In this paper, we used the Rectified Linear Unit (ReLU) activation function due to its non-linearity, computational efficiency, and ability to handle non-negative inputs such as pixel values. Equations (2) and (3) represent the mathematical representation of the ReLU activation function with conditions. By zeroing out negative values, the ReLU activation function helps prevent vanishing gradients and improves the model's performance.

$$\mathbb{R}(z) = \begin{Bmatrix} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{Bmatrix} \tag{2}$$

$$\mathbb{R}(z) = \max(0, z) \tag{3}$$

Max-pooling layers play an important role in CNN by reducing the spatial size of feature maps, thereby reducing the number of parameters and computational expenses. In addition to that, they can help prevent overfitting by selecting superior invariant features, which improve the model's ability to generalize. The maximum activation over non-overlapping regions of a filter with size $\mathbb{F}_x^n \times \mathbb{F}_y^n$ determines the output of the max-pooling layer. During this process, the input map is downsampled by a factor of $\mathbb{F}_x^n$ and $\mathbb{F}_y^n$ while preserving the depth dimension.

For TSR, autonomous self-driving and DAS, where the ability to accurately recognize and classify traffic signs is crucial for road safety, batch normalization can be highly advantageous. It normalizes each layer's activations across a mini-batch of training samples, thereby reducing internal covariate shifts and allowing the proposed model to learn more effectively. Furthermore, by improving training stability and efficiency, the proposed model can lead to faster convergence and better generalization performance.

In our proposed framework, we incorporated global average pooling before the flatten and dense layers to reduce overfitting and computational costs while improving generalization performance. This technique computes the average value of each feature map, creating a single feature vector for the entire image for classification. It serves as a

regularizer, improving interpretability and preventing overfitting while streamlining the architecture and reducing trainable parameters. The use of global average pooling has demonstrated improving the overall model performance and reduced computational costs in various computer vision applications.

The proposed framework was optimized through an iterative process that aimed to balance computational complexity and model performance while minimizing the misclassification error across the training set. To select the optimal values for hyperparameters, such as kernel size and output size, we performed a grid search across a range of values, trained, and evaluated the model for each combination of hyperparameters, and selected the set that produced the highest accuracy on the validation set. We employed a grid search technique with cross-validation to evaluate the performance of each combination of hyperparameters and select the optimal configuration.

However, the selection of the number and size of convolutional layers is a crucial decision that can significantly impact the performance of a CNN architecture. In our proposed framework, we used a single CNN that incorporates multiple layers, including convolutional, max-pooling, and batch normalization layers. The number and size of these layers were determined through empirical evaluation and experimentation.

In summary, the proposed methodology involves several steps, including data preprocessing, model architecture design, and training and evaluation. By using the ReLU activation function, batch normalization, and appropriate hyperparameters, improved accuracy in traffic sign recognition has been achieved. Table 1 provides an overview of the different stages of the proposed methodology, including the activation functions, kernel sizes, strides, and output shapes of each layer.

Table 1. An overview of proposed framework architecture for traffic sign classification.

Layer Name	Output Size	Kernel Size	Strides	Activation	Number of Layers
Input	$100 \times 100 \times 3$	–	–	–	1
Conv2d	$98 \times 98 \times 16$	3×3	1	ReLU	1
Conv2d	$98 \times 98 \times 32$	3×3	1	ReLU	2
MaxPooling2D	$49 \times 49 \times 32$	2×2	2	–	1
Conv2d	$49 \times 49 \times 32$	3×3	1	ReLU	1
Conv2d	$49 \times 49 \times 64$	5×5	1	ReLU	2
Conv2d	$49 \times 49 \times 64$	3×3	1	ReLU	1
MaxPooling2D	$24 \times 24 \times 64$	2×2	2	–	1
Batch Normalization	$24 \times 24 \times 64$	–	–	–	1
Conv2d	$24 \times 24 \times 64$	3×3	1	ReLU	1
Conv2d	$24 \times 24 \times 128$	5×5	1	ReLU	4
Conv2d	$24 \times 24 \times 256$	3×3	1	ReLU	2
Global Average Pooling 2D	$24 \times 24 \times 128$	–	–	–	1
Flatten	256	–	–	–	1
Dense1	1024	–	–	ReLU	1
Dense2	Number of Classes	–	–	Softmax	1

3.4. Loss Function and Optimization Algorithm

The proposed model employs the Adam optimizer and the categorical cross entropy loss function during training. The Adam optimizer is a widely used optimization algorithm that utilizes adaptive learning rates to converge more efficiently to the global minimum than traditional stochastic gradient descent. On the other hand, categorical cross entropy was chosen as the loss function to compare the predicted and actual distributions of the classification problem. This function assigns a probability of 1 to the true class and a probability of 0 to the other classes. Our proposed model includes a softmax classifier in the

last layer that utilizes the categorical cross entropy loss function $\mathcal{L}$, as shown in Equation (4). In this equation, k represents the different classes, $\wp$ is the predicted probability distribution, and $\hat{\wp}$ is the true distribution represented as a one-hot vector.

$$\mathcal{L}(\wp, \hat{\wp}) = \sum_{k} [\wp_k log(\hat{\wp}_k) + (1 - \wp_k) log(1 - \wp_k)] \quad (4)$$

As shown in Equation (5), the softmax function σ was then applied to compute y, which takes a K-dimensional vector of arbitrary real-valued scores $\vec{z}$ and maps it to a K-dimensional vector $\sigma(\vec{z}_k)$ of values between 0 and 1, which add up to 1. In our experiments, the combination of Adam optimizer and categorical cross-entropy loss function has demonstrated promising results in enhancing the accuracy of our model.

$$\sigma(\vec{z_k}) = \frac{e^{z_k}}{\sum_{l=1}^{N}(e^{z_j})} \quad \therefore \; k = 1, 2, \ldots, N \quad (5)$$

4. Experimentation and Result

4.1. Experimental Setup

We conducted a preliminary analysis on a subset of the GTSRB dataset by testing different image resolutions. The aim of this analysis was to determine how the size of the image impacted the performance of the classification model. Subsequently, we tested our model with different image sizes, including 50 × 50, 80 × 80, and 100 × 100, to find the best balance between computational complexity (time and resource usage) and classification performance. However, we did not consider resolutions of 150 × 150 and 200 × 200 due to their excessive time and resource requirements, which exceed 3000 seconds.

After considering these factors, we decided to use an image resolution of 100 × 100, as it provided a good balance between classification accuracy and model complexity. Our findings showed that lower resolutions had a significant negative impact on the model's accuracy in classifying traffic signs, while higher resolutions led to increased model complexity and longer training times (as shown in Table 2).

Table 2. Comparative analysis of detection performance on GTSRB.

Image Resolution Size	Time (in Seconds)	Resources Usage (in MByte)	Accuracy (%)	F1-Score (%)
50 × 50	578.0	9.2	96.97	96.97
80 × 80	1001.1	17.1	97.56	97.51
100 × 100	1264.2	20.7	98.41	98.42

The design objectives of our research in terms of image size were to achieve high accuracy while minimizing training times and resource usage. After considering multiple factors, including the trade-off between accuracy and computational efficiency, we determined that using 100 × 100 pixel images was the optimal choice for our proposed TSR system.

Additionally, our proposed model is designed for real-world deployment in safety-critical situations, where accuracy and reliability are paramount. We believe that using higher resolution images leads to more accurate detection and classification of traffic signs, which is crucial in ensuring the safety of drivers and pedestrians. Henceforth, our proposed model was then trained and evaluated on the GTSRB and BelgiumTS datasets to assess its efficacy and reliability for TSR. During the training process, we employed the Adam optimizer with a learning rate of 0.00001 and a batch size of 64. Additionally, we incorporated various data augmentation techniques such as random rotations, zooming, and horizontal flips to expand the size of the training set and improve the model's generalization ability.

Our experiments were conducted on a Jupyter-notebook that was equipped with a GeForce RTX 2080 Ti GPU. We implemented the primary code using Python 3.8.

4.2. Key Performance Indicator

Accuracy, precision, recall, and f1-score are commonly used key performance indicators (KPIs) to evaluate the effectiveness of classification models. Accuracy measures the overall correctness of the predictions made by the model and is calculated as the ratio of the number of correctly classified samples to the total number of samples, as shown in Equation (6). Precision, on the other hand, measures the proportion of true positives among all the positive predictions made by the model and is calculated as shown in Equation (7). Recall measures the proportion of true positives among all the actual positive samples in the dataset, and is calculated as shown in Equation (8). The f1-score is the harmonic mean of precision and recall, which provides a balance between the two metrics and is calculated as shown in Equation (9). In our experimentation, we will use these KPIs to measure the effectiveness of our proposed model. These metrics will enable us to evaluate the model's ability to classify traffic signs accurately.

$$\mathbb{A} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{6}$$

$$\mathbb{P} = \frac{T_P}{T_P + F_P} \tag{7}$$

$$R = \frac{T_P}{T_P + F_N} \tag{8}$$

$$\mathbb{F}\text{1-score} = 2 \times \frac{\mathbb{P} \times R}{\mathbb{P} + R} \tag{9}$$

4.3. Experimental Results

4.3.1. Performance Results on GTSRB

To assess the efficacy and reliability of our proposed CNN-based model for TSR, we conducted extensive experiments on the GTSRB dataset. To establish a benchmark, we compared our results with several state-of-the-art techniques, including GoogleNet, AlexNet, VGG19, VGG16, MobileNetv2, and ResNetv2. Our experimental results demonstrated that our proposed model achieved exceptional performance, underscoring its potential for real-time TSRs and DAS deployment. A comprehensive comparison of our model with other state-of-the-art models on the GTSRB dataset is presented in Table 3. Our proposed model achieved better accuracy compared to GoogleNet, AlexNet, VGG19, VGG16, MobileNetv2, and ResNetv2 by 0.1%, 1.37%, 4.20%, 0.82%, 1.14%, and 1.4%, respectively. It is pertinent to mention that the computational efficiency and small model size of the proposed model make it more practical and suitable for real-time TSRs.

Table 3. Comparative analysis of detection performance on GTSRB.

Method	Accuracy (%)	Precison (%)	Recall (%)	F1-Score (%)	Parameters (in Millions)	Training Time (in Seconds)	Response Time (in millieconds)	Resources Usage (in MByte)
GoogleNet	98.31	98.37	98.31	98.27	3.70	2983.7	94.80	22.3
AlexNet	97.08	97.17	97.08	97.07	15.78	1638.5	**73.05**	28.0
VGG19	94.44	94.82	94.44	94.43	22.40	2454.8	90.68	23.7
VGG16	97.60	97.67	97.60	97.59	22.40	2442.9	96.21	24.1
MobileNetV2	97.30	97.45	97.30	97.30	12.77	1319.7	100.17	23.0
ResNet50V2	96.97	97.09	96.97	96.92	40.37	1974.9	105.93	26.5
Ours	**98.41**	**98.51**	**98.41**	**98.42**	**2.61**	**1264.2**	74.34	**20.7**

When comparing the performance of the proposed model and several state-of-the-art models, we found that the proposed model achieves the best overall performance in terms of both runtime and memory consumption. The proposed model achieves a runtime of

1264.2 s, which is approximately 57.6%, 23.3%, 48.3%, 48.1%, 3.8%, and 36.0% faster than GoogleNet, AlexNet, VGG19, VGG16, MobileNetV2, and ResNet50V2, respectively. Furthermore, the proposed model achieves a memory consumption of 20.7 MByte, which is approximately 7.2%, 25.0%, 12.7%, 14.0%, 10.0%, and 21.7% lower than GoogleNet, AlexNet, VGG19, VGG16, MobileNetV2, and ResNet50V2, respectively. These results demonstrate that the proposed model achieves a better balance between runtime and memory consumption, making it an attractive solution for real-world deployment on embedded devices with limited computational power and memory. Subsequently, the proposed model also stands out for its exceptional response time, as shown in Table 3. Its response time of 74.34 milliseconds is significantly faster than all other state-of-the-art models tested, with the exception of AlexNet. The proposed model has a slightly higher response time than AlexNet (1.75% faster). Still, its overall performance in terms of response time, training time, and memory consumption makes it a suitable candidate for real-time deployment in TSRs and DAS.

The proposed framework highlights and addresses the limitations of existing TSRs in real-world deployment, particularly in safety-critical scenarios where response time is crucial. The proposed framework is designed to be lightweight and efficient, utilizing advanced feature engineering and CNN to improve overall accuracy and robustness while reducing the number of parameters required for training and inference. Figure 5 illustrates the performance of the proposed model on the GTSRB dataset. The model effectively detects and classifies a diverse range of traffic signs with different shapes, sizes, and colors, indicating its robustness and adaptability in real-world scenarios. The model's success in detecting and classifying a variety of traffic signs highlights its potential to enhance safety measures in traffic management and autonomous driving systems. Furthermore, the superior performance of this proposed model in traffic sign detection and classification demonstrates the potential for its application in safety-critical areas, such as autonomous vehicles, where accurate and fast traffic sign detection and classification are essential.

Figure 5. Detection and classification performance of proposed model on GTSRB.

To gain a deeper understanding of the performance of our proposed model, we performed a comprehensive comparison with some traffic sign detection algorithms such as Faster R-CNN [50], Mask R-CNN [51], Cascaded R-CNN [52], Multiscale Cascaded R-CNN [53], and Deep QNN [54]. The results of this analysis are presented in Table 4, which shows that our model is highly effective in accurately recognizing traffic signs of varying shapes, sizes, and lighting conditions. Our CNN-based model outperformed Faster R-CNN [50], Mask R-CNN [51], Cascaded R-CNN [52], Multiscale Cascaded R-CNN [53], and Deep QNN [54] with a higher recall rate, precision, and f1-score on the GTSRB dataset. Our proposed model achieved a precision rate of 98.51%, which is 1.7% and 0.21% higher than Cascaded R-CNN [52] and Multiscale Cascaded R-CNN [53], respectively.

The proposed model achieved an f1- score of 98.41%, which is 8.17%, 7.2%, and 4.18% higher than Faster R-CNN [50], Mask R-CNN [51], and Deep QNN [54], respectively. These results demonstrate the robustness and reliability of our proposed method in real-world scenarios, where detection and recognition can be challenging due to various environmental factors.

Table 4. A performance comparison of traffic sign detection methods on GTSRB.

Method	Precison (%)	Recall (%)	Missing Rate (%)	F1-Score (%)	Parameters (in Millions)
Li J [4]	84.50	97.81	2.19	90.67	2.92
Kamal U [31]	95.29	89.01	10.99	92.04	5
Faster R-CNN [50]	96.10	86.30	13.70	90.94	16
Mask R-CNN [51]	97.10	86.90	13.10	91.72	18
Cascaded R-CNN [52]	96.80	88.60	11.40	92.52	27
Multiscale Cascaded R-CNN [53]	98.30	85.00	15.00	91.71	55
Deep QNN [54]	94.40	94.94	5.06	94.46	AlexNet: 61.1 VGG16: 138.3 ResNet-50: 25.6 QP: 6 qubits QCNN: 10 qubits
Xu X [55]	93.96	95.27	4.73	94.61	N/A
Li C [56]	90.70	81.7	18.3	86	4.5
Ours	**98.51**	**98.41**	**1.59**	**98.41**	**2.61**

4.3.2. Performance Results on BelgiumTS

To evaluate the proposed model's effectiveness and robustness for TSR, we conducted a comparative analysis with several well-known models, including GoogleNet, AlexNet, VGG19, and VGG16, on the BelgiumTS dataset. However, due to the limited size of the BelgiumTS dataset, which only consists of 4575 training images and 2520 testing images, we employed various augmentation techniques, such as random rotations, zooming, and horizontal flips, to generate synthetic images and increase the dataset size. This allows the model to learn from more diverse images, reduce generalization errors, and improve its performance.

The detection performance of the proposed and other methods is shown in Table 5. The results show that the proposed model outperforms the other models on BelgiumTS with the minimum number of parameters. The proposed model achieved higher accuracy, precision, recall, and f1-score than GoogleNet, AlexNet, VGG19, and VGG16. It achieved an accuracy rate of 92.06%, which is 9.33, 30.19, 9.49, and 33.18 percentage point higher than GoogleNet, AlexNet, VGG19, and VGG16, respectively. Additionally, the proposed model achieved a precision of 94.60%, surpassing GoogleNet and VGG19, which achieved a precision of 89.82% and 88.47%, respectively. Moreover, in terms of recall, the proposed model outperformed GoogleNet and VGG19 with a score of 92.06% compared to 84.20% and 84.08%, respectively.

Figure 6 illustrates our proposed model's detection and classification outcomes on the BelgiumTS dataset. The figure shows several traffic signs along with their actual and predicted labels, detected and classified accurately by the model. In addition, the model shows its ability to effectively detect and classify traffic signs of various shapes, sizes, and colors, demonstrating its robustness and adaptability in real-world scenarios.

Furthermore, the comparison of our proposed model with existing models and algorithms reveals its advantages in terms of computational efficiency, model size, and accuracy, which makes it particularly suitable for real-time traffic sign recognition applications. The practical value of our method is further supported by its potential to improve road safety by enabling timely and accurate detection and recognition of traffic signs, thus reducing the risk of accidents and traffic violations. These results highlight the efficacy and

practical value of our proposed method and suggest its potential for further development and application in the field of TSR.

Table 5. Comparative analysis of detection performance on BelgiumTS.

Method	Accuracy (%)	Precison (%)	Recall (%)	F1-Score (%)	Parameters (in Millions)
GoogleNet	84.20	89.82	84.20	85.46	3.71
AlexNet	70.71	80.77	70.71	72.02	15.79
VGG19	84.08	88.47	84.08	84.74	22.41
VGG16	69.12	79.09	69.12	69.82	22.41
MobileNetV2	36.98	71.89	36.98	35.75	12.77
ResNet50V2	12.46	40.15	12.46	12.86	40.37
Ours	**92.06**	**94.60**	**92.06**	**92.54**	**2.63**

Figure 6. Detection and classification performance of proposed model on BelgiumTS.

5. Discussion

When designing AI&ML models for practical applications such as TSRs and DAS, it is crucial to consider computational efficiency and model size. In addition to that, it is also important to consider the relationship between the number of parameters/operations and the time to respond when designing such models. Response time is a crucial metric that needs to be considered as it is directly related to the complexity of the model. Generally, a model with fewer parameters and operations is expected to have a faster response time than a more complex model. This is because a simpler model requires fewer computations, enabling faster predictions.

In safety-critical applications like automotive systems, it is essential to design such frameworks which are accurate and efficient frameworks with low computational overhead and minimal memory requirements. This can be achieved by reducing the number of trainable parameters and operations while maintaining high accuracy, ensuring that the models can be deployed on embedded devices with limited computing power and memory while still delivering reliable and fast performance. The proposed model has a small model size and high computational efficiency, making it practical and suitable for real-time deployment in TSRs and DAS.

In real-world scenarios, it is crucial to consider response time alongside model complexity. As the number of parameters and operations in AI&ML models increases, the model becomes more complex and requires more computational resources to train. Therefore, it is

necessary to balance complexity and performance to ensure high accuracy and low response times. The proposed research aims to develop a lightweight and efficient TSR system that uses deep learning techniques such as CNNs and advanced feature engineering to improve accuracy while reducing the number of parameters required for training and inference. This system will operate effectively in real-world scenarios with low computational resources.

Our experiments demonstrate that our proposed CNN-based model is highly effective and reliable for TSR and DAS applications, outperforming several state-of-the-art techniques such as GoogleNet, AlexNet, VGG19, VGG16, MobileNetv2, and ResNetv2 in terms of accuracy, precision, recall, and f1-score on the GTSRB dataset (as shown in Table 3). Our model achieved exceptional detection and classification results for traffic signs of varying shapes, sizes, and colors, demonstrating its robustness and adaptability in real-world situations.

While our experiments demonstrate that our proposed CNN-based model performs highly effectively and reliably for TSR and DAS applications, it is worth noting that our model and GoogleNet exhibit similar performance. However, our model is specifically designed for traffic sign recognition, taking into consideration the unique characteristics of traffic signs, such as their shapes, colors, and symbols. This tailored approach may result in better performance compared to GoogleNet for this particular task. Additionally, our proposed model is lightweight and computationally efficient, making it well-suited for real-time applications where resource usage is a concern.

Furthermore, we thoroughly evaluated the proposed model with several traffic sign detection algorithms such as Faster R-CNN, Mask R-CNN, Cascaded R-CNN, Multiscale Cascaded R-CNN, and Deep QNN on the GTSRB dataset. As a result, our proposed model outperformed them in terms of precision, recall, and f1-score. In addition to that, our model achieved exceptional results on the BelgiumTS dataset (as shown in Table 5), outperforming several well-known models such as GoogleNet, AlexNet, VGG19, VGG16, MobileNetv2, and ResNetv2 with higher accuracy, precision, recall, and f1-score.

In conclusion, our proposed CNN-based model is highly effective and reliable for TSR and DAS applications, outperforming several state-of-the-art techniques. The compact size and computational efficiency of the proposed model make it practical and suitable for real-time deployment, particularly in safety-critical applications where a fast response time can be critical to avoiding accidents. Our study provides a foundation for future research in this area, including the development of more robust and reliable models for TSR and DAS applications. However, there are some limitations to our study. Firstly, we only evaluated our proposed model on the GTSRB and BelgiumTS datasets. Future studies should evaluate the performance of the proposed model on a wider range of datasets. Secondly, we used augmentation techniques to increase the dataset size, which may affect the model's generalizability to real-world scenarios. Future studies should investigate the impact of data augmentation on the performance of the proposed model.

6. Conclusions

Traffic sign recognition is an important aspect of autonomous vehicles, as it helps the vehicle navigate and make decisions based on road conditions. Conventional methods for traffic sign detection and classification are computationally intensive and often require high processing power, making them unsuitable for deployment on resource-constrained platforms. To address this issue, we propose a lightweight neural network architecture that balances accuracy and computational efficiency. The proposed model was trained and evaluated on the GTSRB and BelgiumTS benchmark traffic sign datasets. On GTSRB, it achieved an accuracy rate of 98.41%, outperforming various state-of-the-art methods such as GoogleNet, AlexNet, VGG19, VGG16, MobileNetv2, and ResNetv2 by margins of 0.1%, 1.37%, 4.20%, 0.82%, 1.14%, and 1.4%, respectively. On the other hand, for the BelgiumTS dataset, the proposed model achieved an accuracy of 92.06%, surpassing the accuracy of existing state-of-the-art methods such as GoogleNet, AlexNet, VGG19, and VGG16 by margins of 9.33%, 30.19%, 9.49%, and 33.18%, respectively. The results show that our

proposed method outperforms existing state-of-the-art approaches in terms of recognition accuracy, precision, and computational efficiency. In the future, we aim to enhance the evaluation of our models by adopting professional metrics that will provide a comprehensive assessment of their performance from multiple perspectives. This will enable us to gain deeper insights into the strengths and limitations of our model and make more informed decisions regarding their optimization and improvement.

Author Contributions: Conceptualisation, M.A.K., H.P. and J.C.; methodology, M.A.K., H.P. and J.C.; software, M.A.K.; validation, M.A.K.; formal analysis, M.A.K. and H.P.; investigation, M.A.K. and H.P.; resources, H.P. and J.C.; data curation, M.A.K. and H.P.; writing—original draft preparation, M.A.K.; writing—review and editing, M.A.K., H.P. and J.C.; visualisation, M.A.K. and H.P.; supervision, H.P. and J.C.; project administration, H.P.; funding acquisition, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Incheon National University Research Grant in 2018.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

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

References

1. Organization, W.H. *Global Status Report on Road Safety 2018*; World Health Organization: Geneva, Switzerland, 2018.
2. Tao, X.; Gong, Y.; Shi, W.; Cheng, D. Object detection with class aware region proposal network and focused attention objective. *Pattern Recognit. Lett.* **2020**, *130*, 353–361. [CrossRef]
3. Dewi, C.; Chen, R.C. Random forest and support vector machine on features selection for regression analysis. *Int. J. Innov. Comput. Inf. Control* **2019**, *15*, 2027–2037.
4. Li, J.; Wang, Z. Real-time traffic sign recognition based on efficient CNNs in the wild. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 975–984. [CrossRef]
5. Shahud, M.; Bajracharya, J.; Praneetpolgrang, P.; Petcharee, S. Thai traffic sign detection and recognition using convolutional neural networks. In Proceedings of the 2018 22nd International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 21–24 November 2018; pp. 1–5.
6. Sanyal, B.; Mohapatra, R.K.; Dash, R. Traffic sign recognition: A survey. In Proceedings of the 2020 International Conference on Artificial Intelligence and Signal Processing (AISP), Amaravati, India, 10–12 January 2020; pp. 1–6.
7. Tai, S.K.; Dewi, C.; Chen, R.C.; Liu, Y.T.; Jiang, X.; Yu, H. Deep learning for traffic sign recognition based on spatial pyramid pooling with scale analysis. *Appl. Sci.* **2020**, *10*, 6997. [CrossRef]
8. Wang, X.; Guo, J.; Yi, J.; Song, Y.; Xu, J.; Yan, W.; Fu, X. Real-time and efficient multi-scale traffic sign detection method for driverless cars. *Sensors* **2022**, *22*, 6930. [CrossRef] [PubMed]
9. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. The German traffic sign recognition benchmark: A multi-class classification competition. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 1453–1460.
10. Mathias, M.; Timofte, R.; Benenson, R.; Van Gool, L. Traffic sign recognition—How far are we from the solution? In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013; pp. 1–8.
11. Youssef, A.; Albani, D.; Nardi, D.; Bloisi, D.D. Fast traffic sign recognition using color segmentation and deep convolutional networks. In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems, Lecce, Italy, 24–27 October 2016; Springer: Cham, Switzerland, 2016; pp. 205–216.
12. Lai, Y.; Wang, N.; Yang, Y.; Lin, L. Traffic Signs Recognition and Classification based on Deep Feature Learning. In Proceedings of the ICPRAM, Funchal, Portugal, 16–18 January 2018; pp. 622–629.
13. Wang, G.; Ren, G.; Wu, Z.; Zhao, Y.; Jiang, L. A robust, coarse-to-fine traffic sign detection method. In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013; pp. 1–5.
14. Chen, Y.; Xu, W.; Zuo, J.; Yang, K. The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier. *Clust. Comput.* **2019**, *22*, 7665–7675. [CrossRef]
15. Chen, Y.; Xiong, J.; Xu, W.; Zuo, J. A novel online incremental and decremental learning algorithm based on variable support vector machine. *Clust. Comput.* **2019**, *22*, 7435–7445. [CrossRef]
16. Zeng, Y.; Xu, X.; Shen, D.; Fang, Y.; Xiao, Z. Traffic sign recognition using kernel extreme learning machines with deep perceptual features. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 1647–1653. [CrossRef]

17. Hechri, A.; Hmida, R.; Mtibaa, A. Robust road lanes and traffic signs recognition for driver assistance system. *Int. J. Comput. Sci. Eng.* **2015**, *10*, 202–209. [CrossRef]
18. Hmida, R.; Ben Abdelali, A.; Mtibaa, A. Hardware implementation and validation of a traffic road sign detection and identification system. *J. Real-Time Image Process.* **2018**, *15*, 13–30. [CrossRef]
19. 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]
20. Wong, A.; Shafiee, M.J.; Jules, M.S. MicronNet: A highly compact deep convolutional neural network architecture for real-time embedded traffic sign classification. *IEEE Access* **2018**, *6*, 59803–59810. [CrossRef]
21. Zeng, Y.; Xu, X.; Fang, Y.; Zhao, K. Traffic sign recognition using deep convolutional networks and extreme learning machine. In Proceedings of the International Conference on Intelligent Science and Big Data Engineering, Suzhou, China, 14–16 June 2015; Springer: Cham, Switzerland, 2015; pp. 272–280.
22. Cireşan, D.; Meier, U.; Masci, J.; Schmidhuber, J. A committee of neural networks for traffic sign classification. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 1918–1921.
23. CireAan, D.; Meier, U.; Masci, J.; Schmidhuber, J. Multi-column deep neural network for traffic sign classification. *Neural Netw.* **2012**, *32*, 333–338. [CrossRef]
24. Jin, J.; Fu, K.; Zhang, C. Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 1991–2000. [CrossRef]
25. Sun, W.; Du, H.; Nie, S.; He, X. Traffic sign recognition method integrating multi-layer features and kernel extreme learning machine classifier. *Comput. Mater. Contin.* **2019**, *60*, 147–161. [CrossRef]
26. Wang, C. Research and application of traffic sign detection and recognition based on deep learning. In Proceedings of the 2018 International Conference on Robots & Intelligent System (ICRIS), Changsha, China, 26–27 May 2018; pp. 150–152.
27. Ellahyani, A.; El Ansari, M.; Lahmyed, R.; Trémeau, A. Traffic sign recognition method for intelligent vehicles. *JOSA A* **2018**, *35*, 1907–1914. [CrossRef] [PubMed]
28. Lim, K.; Hong, Y.; Choi, Y.; Byun, H. Real-time traffic sign recognition based on a general purpose GPU and deep-learning. *PLoS ONE* **2017**, *12*, e0173317. [CrossRef] [PubMed]
29. Qin, Z.; Yan, W.Q. Traffic-sign recognition using deep learning. In *Proceedings of the International Symposium on Geometry and Vision*; Springer: Cham, Switzerland, 2021; pp. 13–25.
30. Temel, D.; Chen, M.H.; AlRegib, G. Traffic sign detection under challenging conditions: A deeper look into performance variations and spectral characteristics. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3663–3673. [CrossRef]
31. Kamal, U.; Tonmoy, T.I.; Das, S.; Hasan, M.K. Automatic traffic sign detection and recognition using SegU-Net and a modified Tversky loss function with L1-constraint. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1467–1479. [CrossRef]
32. Xie, K.; Ge, S.; Ye, Q.; Luo, Z. Traffic sign recognition based on attribute-refinement cascaded convolutional neural networks. In Proceedings of the Advances in Multimedia Information Processing-PCM 2016: 17th Pacific-Rim Conference on Multimedia, Xi'an, China, 15–16 September 2016; Proceedings, Part I; Springer: Cham, Switzerland, 2016; pp. 201–210.
33. Guo, J.; You, R.; Huang, L. Mixed vertical-and-horizontal-text traffic sign detection and recognition for street-level scene. *IEEE Access* **2020**, *8*, 69413–69425. [CrossRef]
34. Luo, H.; Yang, Y.; Tong, B.; Wu, F.; Fan, B. Traffic sign recognition using a multi-task convolutional neural network. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 1100–1111. [CrossRef]
35. Xing, J.; Yan, W.Q. Traffic sign recognition using guided image filtering. In Proceedings of the International Symposium on Geometry and Vision, Auckland, New Zealand, 28–29 January 2021; Springer: Cham, Switzerland, 2021; pp. 85–99.
36. Mao, X.; Hijazi, S.; Casas, R.; Kaul, P.; Kumar, R.; Rowen, C. Hierarchical CNN for traffic sign recognition. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016; pp. 130–135.
37. Liu, Z.; Qi, M.; Shen, C.; Fang, Y.; Zhao, X. Cascade saccade machine learning network with hierarchical classes for traffic sign detection. *Sustain. Cities Soc.* **2021**, *67*, 102700. [CrossRef]
38. Sichkar, V.N. Real time detection and classification of traffic signs based on YOLO version 3 algorithm. *J. Sci. Tech. Inf. Technol. Mech. Opt.* **2020**, *127*, 418–424. [CrossRef]
39. Megalingam, R.K.; Thanigundala, K.; Musani, S.R.; Nidamanuru, H.; Gadde, L. Indian traffic sign detection and recognition using deep learning. *Int. J. Transp. Sci. Technol.* **2022**. [CrossRef]
40. Bangquan, X.; Xiong, W.X. Real-time embedded traffic sign recognition using efficient convolutional neural network. *IEEE Access* **2019**, *7*, 53330–53346. [CrossRef]
41. Valeja, Y.; Pathare, S.; Patel, D.; Pawar, M. Traffic Sign Detection using Clara and Yolo in Python. In Proceedings of the 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 19–20 March 2021; Volume 1, pp. 367–371.
42. Zaklouta, F.; Stanciulescu, B. Real-time traffic-sign recognition using tree classifiers. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1507–1514. [CrossRef]
43. Lopez-Montiel, M.; Orozco-Rosas, U.; Sanchez-Adame, M.; Picos, K.; Ross, O.H.M. Evaluation method of deep learning-based embedded systems for traffic sign detection. *IEEE Access* **2021**, *9*, 101217–101238. [CrossRef]
44. Avramović, A.; Sluga, D.; Tabernik, D.; Skočaj, D.; Stojnić, V.; Ilc, N. Neural-network-based traffic sign detection and recognition in high-definition images using region focusing and parallelization. *IEEE Access* **2020**, *8*, 189855–189868. [CrossRef]

45. Zhou, S.; Qiu, J. Enhanced SSD with interactive multi-scale attention features for object detection. *Multimed. Tools Appl.* **2021**, *80*, 11539–11556. [CrossRef]
46. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
47. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
48. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway networks. *arXiv* **2015**, arXiv:1505.00387.
49. He, K.; Sun, J. Convolutional neural networks at constrained time cost. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5353–5360.
50. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, *1–9*. [CrossRef] [PubMed]
51. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
52. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
53. Zhang, J.; Xie, Z.; Sun, J.; Zou, X.; Wang, J. A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection. *IEEE Access* **2020**, *8*, 29742–29754. [CrossRef]
54. Kuros, S.; Kryjak, T. Traffic Sign Classification Using Deep and Quantum Neural Networks. In *Proceedings of the Computer Vision and Graphics: Proceedings of the International Conference on Computer Vision and Graphics ICCVG 2022*; Springer: Cham, Switzerland, 2023; pp. 43–55.
55. Xu, X.; Jin, J.; Zhang, S.; Zhang, L.; Pu, S.; Chen, Z. Smart data driven traffic sign detection method based on adaptive color threshold and shape symmetry. *Future Gener. Comput. Syst.* **2019**, *94*, 381–391. [CrossRef]
56. Li, C.; Chen, Z.; Wu, Q.J.; Liu, C. Deep saliency detection via channel-wise hierarchical feature responses. *Neurocomputing* **2018**, *322*, 80–92. [CrossRef]