IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 May 2024, accepted 16 June 2024, date of publication 20 June 2024, date of current version 25 September 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3417023

RESEARCH ARTICLE

Sign-YOLO: Traffic Sign Detection Using Attention-Based YOLOv7

RUTURAJ MAHADSHETTI[1], JINSUL KIM[2], AND TAI-WON UM[1]
[1]Graduate School of Data Science, Chonnam National University, Gwangju 61186, Republic of Korea
[2]School of Electronics and Computer Engineering, Chonnam National University, Gwangju 61186, Republic of Korea

Corresponding authors: Tai-Won Um (stwum@jnu.ac.kr) and Jinsul Kim (jsworld@jnu.ac.kr)

This work was partly supported by the Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Ministry of Science and ICT(MSIT), Korea [IITP-2024-00156287, 50%] and the IITP grant funded by MSIT, Korea [No.RS-2021-II212068, Artificial Intelligence Innovation Hub, 50%].

ABSTRACT Traffic sign detection (TSD) is crucial for real-world applications like driverless vehicles, intelligent driver-assistance systems, and traffic management. Recent advancements in TSD have demonstrated promising outcomes. Nonetheless, challenges persist in terms of speed, accuracy, memory consumption, the capability of the backbone to generate features, and computational cost, especially in handling diverse traffic sign characteristics. To overcome these challenges, we propose Sign-YOLO (You Only Look Once), a novel attention-based one-stage method that integrates YOLOv7 with the squeeze-and-excitation (SE) model and special attention mechanism. Sign-YOLO enhances the feature representation capacity of the model in the presence of variations in traffic sign sizes. The SE block adjusts channel-specific feature responses by actively considering the relationships between channels. By selectively focusing on relevant features, the attention mechanism helps the model better capture and understand the distinctive characteristics of traffic signs, thereby improving detection accuracy. Sign-YOLO effectively reduces the computational cost and memory consumption; further, it effectively enriches the robustness of extracted features. The proposed approach enables the model to allocate more attention to relevant regions of the input, thereby reducing the impact of size discrepancies and contributing to the overall robustness of the system. The experimental findings highlight the success of Sign-YOLO in TSD tasks. Our proposed method exhibits cutting-edge performance on the German Traffic Signs Detection Benchmark (GTSDB) dataset, simultaneously achieving a 98% reduction in model size and memory consumption. Sign-YOLO attains a 99.10% mean average precision (mAP) on the GTSDB dataset. In comparison to both two- and one-stage detectors, our approach exhibits an improvement of 3.33%. The proposed approach is the swiftest and most lightweight framework in terms of memory usage, making it the ideal option for implementation in real-time applications.

INDEX TERMS Traffic sign detection, deep learning, driver-assistance system, automated driving system.

I. INTRODUCTION

In recent years, computer vision has shown great success in real-world applications such as self-piloted vehicles, medical diagnostics, augmented reality/virtual reality, safety measures, and surveillance. Traffic sign detection (TSD) is a crucial element in advanced driver-assistance systems (ADAS). It plays a crucial role in practical applications including driverless vehicles, traffic monitoring, improving driver safety, offering support, managing road networks, and analyzing traffic scenarios. The primary goal of TSD is to enable automated systems such as ADAS and autonomous vehicles. An effective TSD system aids vehicles in perceiving their surroundings. In the context of ADAS, the TSD system serves to notify drivers about traffic regulations. In ADAS, apart from perceiving the surroundings, the TSD system

The associate editor coordinating the review of this manuscript and approving it for publication was Rajesh Kumar.

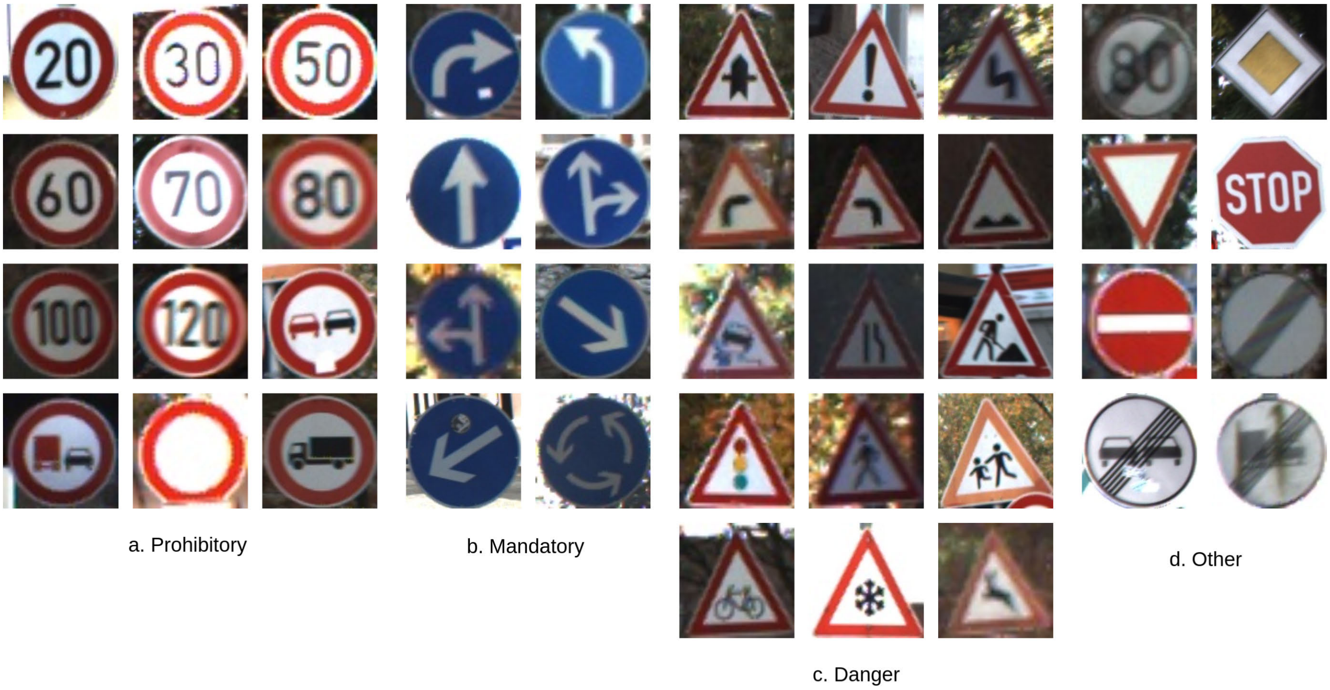

FIGURE 1. German Traffic Signs Detection Benchmark's (GTSDB) collection of traffic signs.

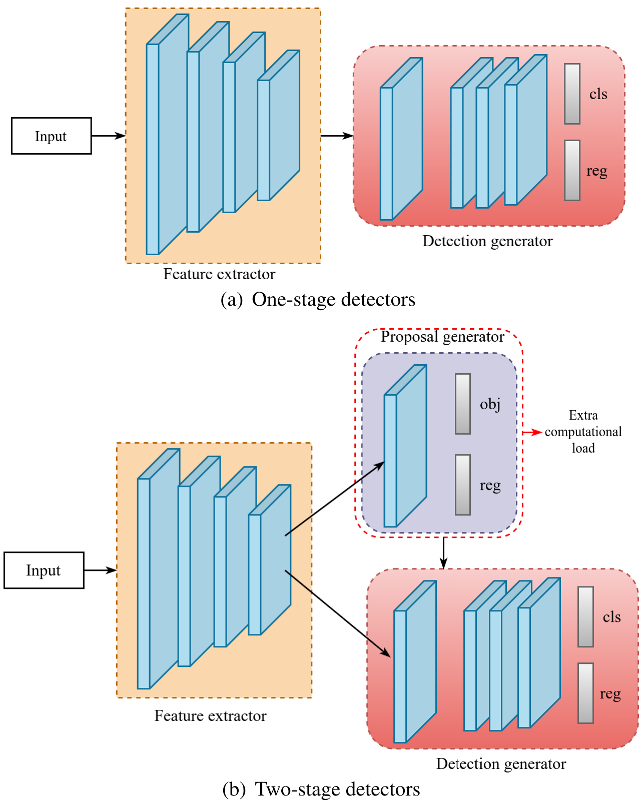

FIGURE 2. Meta-architectures for deep-learning-based detection models.

also supplies the vehicle navigation system with information about the locations of traffic signs. This location data can serve as unique reference points for generating a high-definition (HD) road map. Recent advancements in TSD have yielded favorable results. Nonetheless, challenges related to speed, accuracy, memory consumption, the capability of the backbone to extract features, and the computational cost of the detection process persist. This is particularly evident in addressing the diverse characteristics of various traffic signs.

The identification and detection of traffic signs involve two distinct tasks: TSD and traffic sign recognition (TSR). Traditional TSD methods heavily depended on manually extracting features from the original dataset. Manual algorithms were employed for feature extraction based on shade information, perimeter detection, and geometric patterns. Color-centric approaches generally segmented regions with traffic signs in detailed color spaces such as hue-saturation-intensity (HSI) [1] and hue-chroma-luminance (HCL) [2]. However, these solutions required the manual curation of large numbers of images and were thus expensive and time-consuming. Shape- and color-oriented techniques [3] were also widely utilized; however, they had drawbacks such as sensitivity to changes in illumination, occlusions, scale variations, rotations, and translations. Although machine learning (ML) can mitigate some of these challenges, it requires extensive annotated datasets. In recent years, deep learning (DL) has emerged as a powerful approach for TSR and has demonstrated excellent performance.

Traffic signs are designed to attract human attention quickly. Methods that depend on handcrafted features [1], [4], [5], [6], [7] utilize the visual attributes of these signs during the feature extraction process. However, these approaches lack robustness in distinguishing genuine signs from counterfeit ones globally, as numerous objects resemble traffic signs. The use of low-level handcrafted features proves challenging in accurately representing the unique characteristics that differentiate traffic sign

Significant progress has been made in object detection owing to advancements in DL algorithms. Convolutional neural network (CNN) applications including You Only Look Once (YOLO) [8], SSD [9], Cascade R-CNN [10], Faster R-CNN [11], and Fast R-CNN [12] have been widely adopted as detectors and have demonstrated excellent performance. Figure 1 illustrates the four classes derived from the German Traffic Signs Detection Benchmark (GTSDB) dataset: prohibitory, mandatory, danger, and other [14]. These classes are further divided into 43 subclasses. For identifying the optimal detector for a particular application, conventional accuracy metrics such as mean average precision (mAP) as well as other variables including memory consumption and processing times must be evaluated. For instance, autonomous vehicles require both precise detection and real-time performance, and self-driving cars benefit from streamlined model architectures with minimal memory usage.

As shown in Figure 2, advanced detectors are categorized into two groups: one-stage detectors [8], [9] that can predict the entire image in a single pass and two-stage detectors [10], [11], [12] that employ a multistage detection process using a series of detectors, where each stage refines the results of the previous one. Recent two-stage traffic sign detectors [13], [14], [15] have shown promising results. However, the use of a region proposal generator (RPN) to generate candidate object bounding box proposals in an image that likely contains objects imposes an additional computational burden, thus limiting the real-world application performance and accuracy. Furthermore, these methods require a larger memory size compared to one-stage approaches. Earlier frameworks have attained state-of-the-art performance on several benchmarks. However, recent TSD frameworks are not capable of generating robust features, thus limiting the model performance. To overcome these challenges, we propose a Sign-YOLO approach that integrates YOLOv7 [16] with a Squeeze-and-Excitation (SE) block and a special attention mechanism. YOLOv7 is a single-stage detector that can generate predictions for the entire image in a single pass, making it highly suitable for real-time applications. The SE block enhances the quality of representations generated by a network by explicitly capturing the relationships among the channels in its convolutional features. It enables the network to recalibrate features, thereby leveraging global information to highlight important features selectively while downplaying less relevant ones. The special attention mechanism allows the model to selectively enhance or suppress features of different scales and positions, thereby mitigating the problem of over-segmentation. This mechanism contributes to more accurate sign representations. Additionally, it helps the model better filter background areas, reduce false alarms, and integrate detected sign fragments, ultimately leading to improved precision and recall in sign detection tasks. Finally, it highlights important spatial locations in the feature maps. Sign-YOLO enhances the feature extractor, generating robust features that significantly improve accuracy and effectively reduce the model size.

The primary contributions of this article are as follows:

- We have developed a robust one-stage detector, Sign-YOLO, based on YOLOv7. Sign-YOLO effectively enriches extracted features in real-time capability with good performance for TSD tasks.
- We propose an attention-based approach for the TSD framework that extracts high-frequency features. This approach achieves superior or competitive performance while overcoming false positives.
- Sign-YOLO effectively reduces the memory consumption and reduces the model size by 98% compared to those of a two-stage traffic sign detector.
- Sign-YOLO considers all categories, including "other," from the GTSDB dataset, whereas other methods neglected it.
- Extensive experiments were performed to evaluate the effectiveness of Sign-YOLO on the GTSDB dataset. The results revealed that it achieved excellent performance, with a mAP of 99.10%.

The rest of this paper is organized as follows. In Section II, we review related works. In Section III, we describe the proposed method. In Section IV, we introduce the datasets and evaluation metrics and describe the experiments and their analyses. Lastly, in Section V, we present the conclusions of this study.

II. RELATED WORK

A. TSD USING CNNs

The CNN algorithm, a popular DL technique, currently finds widespread applications across diverse domains including computer vision, natural language processing, and visual-semantic alignments [20], [21], [22], [23]. Depending on the necessity of the region proposal, it can be categorized into two types: one-stage detection and two-stage detection. One-stage detection is commonly employed in scenarios like traffic detection owing to its swift performance. For example, Shao et al. [24], [25] presented an enhanced iteration of Faster R-CNN designed for the identification of traffic signs. They optimized the Gabor wavelet through the implementation of a regional suggestion algorithm, thereby enhancing the network's recognition speed. Zhang et al. [26] enhanced a single-stage traffic sign detector by modifying the convolutional layers in a YOLOv2-based network. They employed the China Traffic Sign Dataset during training to improve its alignment with Chinese traffic road scenes. Another study created a new perceptual generative adversarial network for the recognition of small-sized traffic

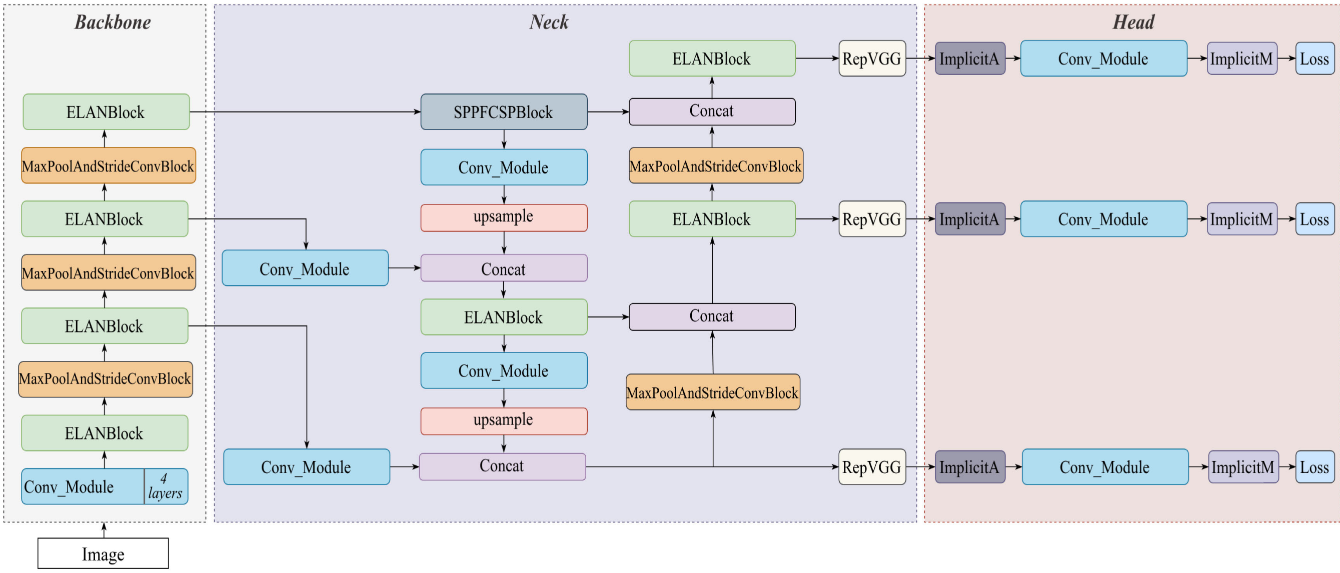

FIGURE 3. **Fundamental structure of YOLOv7's architecture.**

signs [27]. This approach improved the detection accuracy by creating high-resolution representations specifically designed for compact traffic signs.

In addressing the challenge of diverse scales in TSD, SADANet [28] integrates a domain adaptive network with a multiscale prediction network to enhance the capability of extracting features across various scales. Many previous networks employed one-stage detection and relied solely on one-scale depth features, making it difficult to achieve optimal performance in complex traffic scenarios. The visual characteristics of traffic signs at various scales exhibit notable distinctions, and the overall proportion of traffic signs in a given traffic scene image is minimal. As a result, scale variety emerges as the main challenge in TSD and TSR. Creating a scale-invariant representation is essential for accurate target recognition and localization [29]. Wang et al. [17] introduced the attention fusion feature pyramid network (AF-FPN), an enhanced feature pyramid model, to augment the representational capabilities of the feature pyramid within the YOLOv5 network, especially for real-time multiscale TSD. The AF-FPN model reduced information loss and improved detection accuracy for multiscale traffic signs. Tang et al. [15] presented the integrated feature pyramid network with feature aggregation (IFA-FPN) method to enhance both the precision and efficiency of TSD. They also explored the constraints associated with conventional FPN methodologies and compared the performance of their proposed approach and other CNN-based methods for TSD. Hong et al. [19] introduced the CCSPNet feature extraction module based on CNN and Transformer as well as the joint training method CCSPNet Joint. Another study presented the CCTSDB-AUG dataset that includes images with foggy, rainy, and blurry perspectives [19]. YOLOv7-TS [48] uses sub-pixel convolution to preserve rich information and enhance the effectiveness of feature fusion. It is more complex and struggles to handle real-world conditions. Unfortunately, these approaches have high computational cost and low resilience, and they require a large memory capacity. Sign-YOLO overcomes these challenges by achieving better performance and reducing the model size.

B. DATA AUGMENTATION

Data augmentation plays a crucial role in optimizing networks and has proved effective in various vision tasks [2], [11], [18]. It enhances the CNN performance, serves as a safeguard against overfitting [30], and offers a straightforward implementation [31]. Data augmentation methods are of two main types: color transformation (including contrast, blur, noise, and color casting) and geometric transformation (including zoom, translation, random cropping, and rotation) [32]. These techniques artificially expand the size of the training dataset by employing data warping or oversampling. Tabernik and Skočaj [42] proposed a Mask R-CNN framework to enhance recall rates for small traffic signs and presented a novel augmentation technique suitable for traffic sign categories. Additionally, a new dataset with 200 traffic sign categories and 13,000 instances is introduced as a benchmark for complex traffic signs. Temel et al. [43] emphasized the significance of utilizing diverse data augmentation methods and combining simulated and real-world data to enhance recognition performance. Jöckel et al. [44] address the challenge of insufficient training data by developing an approach to create realistic image augmentations with various quality deficits. They also highlighted the need to incorporate contextual information and consider the influence of the quality deficit combination.

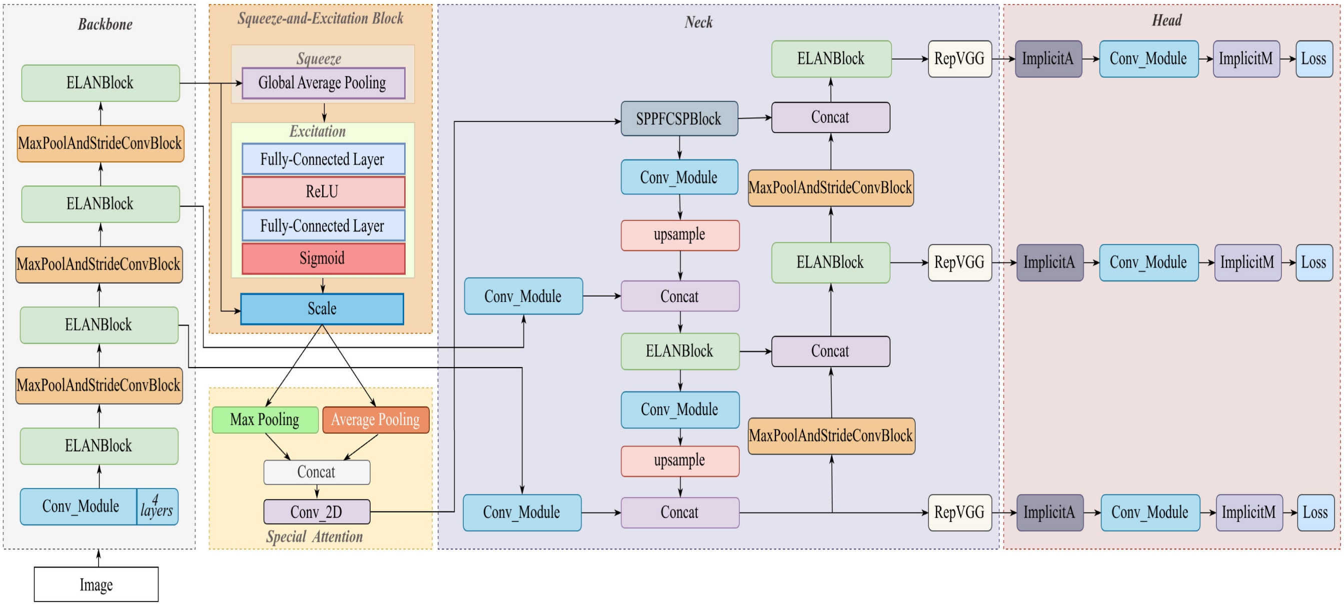

FIGURE 4. Detailed architecture of Sign-YOLO framework.

Other studies [45], [46] proposed a framework using different color enhancement techniques.

C. BASELINE

YOLO is a widely adopted and efficient object detection algorithm in the field of computer vision. Its effectiveness lies in its ability to perform real-time object detection with high accuracy. Compared with the latest YOLO model, YOLOv7 [16] emerges as the optimal choice for inference at high resolution, albeit with a slightly lower speed. YOLOv7 comprises six models: YOLOv7, YOLOv7-X, YOLOv7-W6, YOLOv7-E6, YOLOv7-D6, and YOLOv7-E6E. YOLOv7 is lightweight and utilitarian, making it suitable for real-world applications that require low memory space and high accuracy. Figure 3 shows the fundamental structure of the baseline.

III. PROPOSED METHOD

The detailed architecture of the proposed framework is shown in Figure 4. We integrated the SE block and a special attention approach with the backbone. The backbone takes an image as the input and extracts features. Subsequently, the neck is employed for feature aggregation. Finally, the head is used for prediction.

As illustrated in Figure 3, the basic architecture of YOLOv7 is divided into three parts: Backbone, Neck, and Head. The backbone is responsible for extracting hierarchical features from the input image. It usually consists of a CNN that processes the input image in a series of convolutional and pooling layers. The neck may include additional convolutional layers, skip connections, or other structures that enhance feature representations before they are fed into the head. The neck helps in improving the model's ability to detect objects at various scales and resolutions. The head makes predictions based on the features extracted by the backbone and processed by the neck. It typically consists of a set of convolutional and fully connected layers. The output of the head is a set of bounding boxes, class probabilities, and confidence scores for each object class.

A. SQUEEZE-AND-EXCITATION BLOCK

The SE block aims to enhance the representational capabilities of CNNs. The pivotal feature of the SE block is a mechanism called "channel-wise attention"; it enables the framework to adaptively recalibrate the feature maps at each layer. The core concept behind the SE block is to explicitly model the interdependencies between the different channels in the feature maps. This is done to selectively emphasize the informative features while suppressing the less relevant ones. This is achieved through a two-step process:

1) SQUEEZE PHASE

During the squeeze phase, spatial information from feature maps is consolidated through global average pooling. In essence, it compresses spatial dimensions into a singular descriptive vector for every channel.

2) EXCITATION PHASE

To leverage the data consolidated during the squeeze phase, we proceed with another step aimed at comprehensively grasping the dependencies between channels. This step requires the function to satisfy two conditions: flexibility, especially in learning nonlinear interactions among channels, and the ability to learn non-mutually-exclusive relationships.

We prefer multiple channels to be highlighted instead of enforcing a one-hot activation. The sigmoid function is employed with the gated approach to fulfill these requirements.

$$F_{exc}(x) = \sigma(FC(ReLU(FC(x)))) \tag{1}$$

Here, σ denotes the sigmoid function, and FC stands for the fully connected layer. A method involving two FC layers is employed to enhance its capability to generalize and diminish framework complexity by incorporating a gating mechanism around the nonlinear part. This involves adding a layer to reduce the dimensionality by a reduction ratio (r), followed by a ReLU activation function, and then a layer to restore dimensionality to the original channel dimension of the output transformation. After the excitation process, element-wise multiplication is performed across channels between the features extracted by the backbone and the outcome of the excitation phase.

B. SPECIAL ATTENTION

In this study, an attention-based approach to TSD, based on [39], is introduced to enhance the feature extractor's capability, further improve the model performance, and reduce the memory consumption and model size. Following the SE block, we employ the special attention mechanism illustrated in Fig. 4. The primary objective of the attention-based approach is to derive resilient features from the output of the backbone and SE block, thereby enhancing detection accuracy.

Regarding the spatial attention module, we employ max-pooling and average-pooling along the channel axis and combine them to create the feature descriptors $F^{MP} \in R^{1\times H\times W}$ and $F^{AP} \in R^{1\times H\times W}$, respectively. Subsequently, we apply a convolution layer and a sigmoid layer to produce the spatial attention map $F \in R^{H\times W}$, where H and W represent the height and width, respectively.

The spatial attention is computed as follows:

$$F(x) = \sigma(conv2d([F^{MP}(x) \odot F^{AP}(x)])) \tag{2}$$

Here, MP and AP denote max-pooling and average-pooling, respectively; σ is the sigmoid function; $\odot$ is the concatenate operation; and conv2d is a convolution layer. Using the attention mechanism enables the model to combine multilevel features to form more accurate representations of text. After obtaining features, these extracted characteristics are utilized as inputs for prediction in both the Neck and Head stages.

The pseudocode outlines the basic steps involved in the proposed framework, including initializing the network; loading pre-trained weights; preprocessing the input image; performing convolution and fully connected operations; applying the SE block; applying the attention approach; extracting feature maps; and predicting the bounding box coordinates, objectness scores, and class probabilities.

Algorithm 1 Pseudocode of Proposed Framework

```
Input: input_image
Output: bounding_box, objectness_score, class
Initialize neural network parameters
Load pre-trained backbone network weights
Preprocess input_image
while training not converged do
    Read batch of training data (images and labels)
    Forward pass through the backbone network
    Apply squeeze-and-excitation block
        - Apply Global Average pooling
        - Apply fully-connected (FC) layer
        - Apply ReLU
        - Apply fully-connected (FC) layer
        - Apply Sigmoid
        - Apply scale and combine
    Apply attention mechanism
        - Apply max-pooling
        - Apply average-pooling
        - Feature concatenation
        - Apply convolution layer
    Generate feature maps at multiple scales
    Apply detection head to each scale for bounding
      box predictions
    Compute loss for bounding box predictions
    Update weights using backpropagation and
      optimization algorithm
end while
while testing do
    Read input image
    Forward pass through the backbone network
    Apply squeeze and excitation block
        /* Same as training phase    */
    Apply attention mechanism
        /* Same as training phase    */
    Generate feature maps at multiple scales
    Apply detection head to each scale for bounding
      box predictions
    Apply non-maximum suppression to filter
      redundant detections
end while
```

IV. EXPERIMENT

In this section, we demonstrate the effectiveness of the Sign-YOLO approach. First, we provide an overview of the datasets used for training and analysis. Then, we present the implementation details of the proposed model along with the evaluation information. Finally, we compare the proposed framework against state-of-the-art (SOTA) approaches.

A. DATASET

Numerous publicly accessible datasets containing traffic sign information have been compiled from various countries, including Germany [14], Belgium [34], the United

TABLE 1. Comparison of performance metrics of proposed method with those of other TSD approaches.

Model	Class	Percision	Recall
Shao et al. [25]	Prohibitory	98.76	-
	Mandatory	97.96	-
	Danger	98.41	-
	Other	-	-
Zhang et al. [26]	Prohibitory	96.81	-
	Mandatory	94.02	-
	Danger	96.12	-
	Other	-	-
Yang et al. [40]	Prohibitory	97.46	-
	Mandatory	93.45	-
	Danger	91.12	-
	Other	-	-
Faster R-CNN Inception Resnet v2 [13]	Prohibitory	91.38	98.75
	Mandatory	81.21	70.00
	Danger	85.07	79.45
	Other	-	-
YOLOv7 [16]	Prohibitory	97.00	98.30
	Mandatory	98.00	80.00
	Danger	97.90	93.30
	Other	95.10	92.00
Sign-YOLO without SE (Ours)	Prohibitory	**100**	**100**
	Mandatory	**99.40**	**100**
	Danger	**99.60**	**95.10**
	Other	**99.70**	**96.00**
Sign-YOLO with SE (Ours)	Prohibitory	**100**	**100**
	Mandatory	**99.70**	**100**
	Danger	**99.50**	**95.80**
	Other	**99.70**	**97.10**

States [35], Italy [36], China [37], Croatia [38], and Sweden [5]. In this study, we performed experiments using the GTSDB dataset [14], which has been widely used for comparing TSD methods.

The GTSDB dataset encompasses natural traffic scenes captured across diverse road types (e.g., highway, rural, urban) in both daytime and twilight conditions, illustrating various weather scenarios. This dataset comprises 900 full images containing 1206 traffic signs; it was divided into a training set of 600 images (with 846 traffic signs) and a testing set of 300 images (with 360 traffic signs). Each image may contain zero, one, or multiple traffic signs, and these signs often exhibit variations in orientation, lighting, or occlusion. The signs are categorized into four main types: mandatory, prohibitory, dangerous, and others, with 43 subcategories as shown in Figure 1. This dataset was used for the subsequent training and evaluation of deep neural networks for TSD.

B. EVALUATION MATRIX

To evaluate the accuracy of our proposed method, we employ standard metrics like precision (P) and recall (R). In this context, we define a closed bounding box containing a sign as a true positive (TP) and a rectangular box without any sign inside as a false positive (FP). If a sign exists without a rectangular box, it is categorized as a true negative (TN).

The precision (P) is calculated by determining the ratio of correctly identified signs by our proposed method to the total sum of both correctly and incorrectly detected signs. This metric gauges the accuracy of identified sign regions. Meanwhile, the recall (R) is the ratio of correctly detected signs; it serves as a measure of the method's capability to identify all sign instances in the scene. Additionally, we utilize the intersection over union (IoU) ratio as a threshold for classifying predicted outcomes as either TP or FP, with a specified value of 0.2 in this study.

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$R = \frac{TP}{TP + FN} \tag{4}$$

$$IoU = \frac{Obj \cap DB}{Obj \cup DB} \tag{5}$$

Here, Obj is the correct object area; DB, the predicted candidate area; "Obj ∩ DB," the region where the two areas overlap; and IoU, a numerical value (ranging from 0 to 1) that signifies the degree of overlap between these regions.

C. IMPLEMENTATION DETAILS

We performed experiments with Ubuntu 20.04 LTS, CUDA 11.4, and CuDNN 8.2.1 on a PC equipped with an AMD® Ryzen Threadripper Pro 3955WX (16 cores) and two NVIDIA RTX A5000 GPUs, each with 24 GB of

TABLE 2. Evaluating the characteristics of the Sign-YOLO approach in contrast to a two-stage detector model. "TD" represents a two-stage detector.

Model		TD	mAP	FPS	Memory (MB)	GigaFLOPS	Parameters (10^6)
Arcos-García et al. [13]	Faster R-CNN Inception Resnet V2	✓	95.77	2.26	18250.45	1837.54	59.41
	R-FCN Resnet 101	✓	95.15	11.7	3509.75	269.9	64.59
	Faster R-CNN Resnet 101	✓	95.08	8.11	6134.71	625.78	62.38
	Faster R-CNN Resnet 50	✓	91.52	9.61	5256.45	533.58	43.34
	Faster R-CNN Inception V2	✓	90.62	17.08	2175.21	120.62	12.89
	YOLO V2	✓	78.83	46.55	1318.11	62.78	50.59
	SSD Inception V2	✓	66.10	42.12	284.51	7.59	13.47
	SSD Mobilenet	✓	61.64	66.03	94.70	2.30	5.57
Zang et al. [49]	-	✗	93.36	62	-	95.04	-
YOLO-v7 [16]	-	✗	97.00	35	321.67	220	70.31
Sign-YOLO without SE (Ours)	-	✗	**98.90**	**24**	**332.40**	**230**	**72.84**
Sign-YOLO with SE (Ours)	-	✗	**99.10**	**26**	**349.96**	**260**	**88.94**

memory. The proposed Sign-YOLO network was trained for 500 epochs with a batch size of 32. We used the Adam optimizer approach with a learning rate of 0.01 to optimize our network. We evaluated the performance of our proposed Sign-YOLO model for traffic signs across four different classes: prohibitory, mandatory, danger, and other. The ground truth annotation file for the GTSDB dataset is available in the PASCAL VOC format. We used the Roboflow annotation tool to convert the ground truth into the YOLOv7 format. Further, we used subclasses of the GTSDB dataset. The resolution of the input images used for training and testing the proposed model was 640 and 640, respectively. Sign-YOLO uses the weights of YOLOv7 to train the model.

D. ABLATION STUDY

In this section, we demonstrate the validity and performance of Sign-YOLO. Sign-YOLO was evaluated in terms of various metrics including accuracy, precision, recall, number of parameters, floating point operations (FLOPs), memory usage, processing time, and memory consumption.

1) EFFECTIVENESS OF SIGN-YOLO

Table 1 presents the precision and recall scores achieved for each superclass of traffic signs by the proposed method and previous methods. Whereas previous methods focused solely on the Prohibitory, Mandatory, and Danger superclasses from the GTSDB dataset and neglected the Other class, even though it plays an important role in the self-driving car system, our proposed method considers all classes for training and proficiency validation. We assess the effectiveness of the suggested framework in two variations: one incorporating both the SE block and attention mechanism and the other utilizing only the attention approach without the SE block. As shown in Table 1, Sign-YOLO exhibits notable performance compared to that of one- and two-stage detectors. We used YOLOv7 as a baseline; it achieved a precision of 97%, 98%, 97.90%, and 95.10% and recall of 98.30%, 80%, 93.30%, and 92.00%, respectively. The proposed method, employing only the attention mechanism, attained precision scores of 100%, 99.40%, 99.60%, and 99.70% with corresponding recall rates of 100%, 100%, 95.10%, and 96.00%. Specifically, when integrating both

TABLE 3. Comparison of average precision and recall of the proposed method with other TSD approaches.

Model	Precision	Recall
SADANet [28]	82.25	81.90
Shao et al. [25]	98.53	-
YOLOv7 [16]	97.00	96.80
Sign-YOLO without SE(Ours)	**99.70**	**97.90**
Sign-YOLO with SE (Ours)	**99.70**	**98.10**

TABLE 4. Experimental comparison of one- and two-stage detectors and proposed Sign-YOLO.

Model	mAP	mAP@.5:.95
Tang et al. [15]	80.30	-
Kaleybar et al. [41]	93.50	-
Shao et al. [24]	82.26	-
Shao et al. [25]	69.56	-
YOLOv3 [8]	53.70	-
Cascade-RCNN [10]	70.40	-
Arcos-García et al. [13]	95.77	-
Faster RCNN [11]	63.40	-
Zang et al. [49]	93.36	-
YOLOv7 [16]	97.00	83.00
Sign-YOLO without SE(Ours)	**98.90**	**85.40**
Sign-YOLO With SE(Ours)	**99.10**	**86.30**

the SE block and a special attention approach, Sign-YOLO achieved precision rates of 100%, 99.70%, 99.50%, and 99.70% with corresponding recall rates of 100%, 100%, 95.80%, and 97.10%, respectively, on the superclasses of the GTSDB dataset.

Table 2 shows a comparison of Sign-YOLO with two-stage detectors in terms of various model properties including mAP, FPS, memory consumption, GigaFLOPS, and number of parameters (in millions). The proposed Sign-YOLO framework utilizes only the attention approach and attains a 3.13% improvement while reducing the memory consumption. Upon combining the SE block and attention mechanism with the baseline, the Sign-YOLO system achieves a 3.33% enhancement on the German traffic sign dataset compared to the system that utilizes only the attention approach. Additionally, this integration reduces memory usage significantly by 98%, making it advantageous for practical implementations.

Table 3 presents the average precision and recall achieved using the proposed method. When compared to previous

TABLE 5. Validation of the proposed method on sub-classes of the GTSDB.

Sign-YOLO (With attention approach)					Sign-YOLO (With SE block and attention approach)				
Class	Precision	Recall	mAP@.5	mAP@.5:.95	Class	Precision	Recall	mAP@.5	mAP@.5:.95
speed limit 30	94.90	100	99.50	90.30	speed limit 30	95.80	100	99.60	92.00
speed limit 50	90.30	100	99.60	94.80	speed limit 50	92.80	100	99.70	96.90
speed limit 60	96.90	100	99.50	79.60	speed limit 60	97.10	100	99.60	82.00
speed limit 70	98.20	100	99.60	92.00	speed limit 70	98.90	100	99.70	93.60
speed limit 80	95.80	100	99.60	82.60	speed limit 80	96.40	100	99.60	84.10
speed limit 100	85.40	100	99.50	89.60	speed limit 100	87.00	100	99.70	91.50
no overtaking	86.40	100	99.50	84.60	no overtaking	88.30	100	99.60	87.00
no overtaking (trucks)	95.40	100	99.50	89.70	no overtaking (trucks)	96.30	100	99.60	91.70
priority at next intersection	96.10	100	99.50	88.30	priority at next intersection	97.00	100	99.70	90.00
priority road	96.80	100	99.60	92.30	priority road	97.20	100	99.60	93.10
give way	87.00	100	99.50	87.80	give way	89.50	100	99.60	88.00
stop	96.10	100	99.60	92.10	stop	96.50	100	99.60	93.40
no traffic both ways	88.50	100	99.50	94.50	no traffic both ways	90.00	100	99.60	95.00
no entry	100	75.20	99.50	84.60	no entry	100	78.00	99.50	86.10
danger	100	78.30	97.80	85.60	danger	100	80.00	98.50	87.50
uneven road	100	64.90	99.50	69.70	uneven road	100	67.70	99.60	72.10
slippery road	82.90	100	99.50	99.50	slippery road	83.90	100	99.50	99.60
road narrows	74.40	100	99.50	89.60	road narrows	74.40	100	99.60	91.10
construction	100	100	99.50	99.50	construction	100	100	99.60	99.60
traffic signal	94.20	100	99.60	83.90	traffic signal	95.10	100	99.60	85.40
pedestrian crossing	97.10	87.50	98.20	80.40	pedestrian crossing	98.00	89.60	98.50	83.00
school crossing	92.40	100	99.50	99.50	school crossing	93.20	100	99.50	99.50
cycles crossing	99.20	100	99.50	99.50	cycles crossing	99.30	100	99.60	99.60
restriction ends	91.80	100	99.50	82.10	restriction ends	92.90	100	99.50	84.20
go right	89.40	100	99.50	94.50	go right	91.20	100	99.50	95.00
go straight	87.10	100	99.50	89.60	go straight	88.40	100	99.50	90.10
go right or straight	100	100	99.50	99.50	go right or straight	100	100	99.60	99.60
keep right	86.10	100	99.60	91.10	keep right	88.20	100	99.60	92.30
roundabout	95.70	100	99.50	84.60	roundabout	96.10	100	99.50	86.10
restriction ends (overtaking (trucks))	94.10	100	99.60	89.60	restriction ends (overtaking (trucks))	95.00	100	99.60	91.00

methods, baseline, and Sign-YOLO, Sign-YOLO exhibits superior average precision and recall. The Sign-YOLO approach demonstrates enhancements of 1.17% and 2.70% in overall average precision and 16% and 1.10% in overall recall accuracy, respectively, when excluding the SE block. When incorporating the SE block, the proposed method demonstrates improvements of 1.3% and 0.2% in average recall compared to both the baseline and the framework with the special attention mechanism. In comparison to SADANet [28], the proposed framework demonstrates a gain of 17.45% in average precision and 16% in overall average recall.

Furthermore, it outperforms the earlier framework, as illustrated in Table 4, and attains a SOTA mAP value. Notably, Sign-YOLO demonstrates outstanding accuracy compared to one-stage detectors [8], [24], and it outperforms two-stage detectors [10], [11], [13], [15], [25], [41] and the baseline [16]. In comparison to the baseline without the SE block, the proposed approach achieves a 1.9% and 2.4% improvement in mAP. Upon combining the SE block and a special attention mechanism, Sign-YOLO demonstrates 2.1% and 3.2% improvements in mAP over the baseline. Table 4 provides an overview of the mAP across various IoU thresholds ranging from 0.5 to 0.9.

2) EFFECTIVENESS OF SIGN-YOLO ON SUBCLASSES

To evaluate the efficiency of Sign-YOLO, we used the subsets within the GTSDB. As noted earlier, this dataset comprises 43 subclasses that are classified into four primary superclasses: prohibitory, mandatory, danger, and other. Table 5 presents Sign-YOLO's validation performance with the detected classes. We demonstrate the performance of the framework with the SE block against the baseline incorporating a special attention mechanism. Our proposed framework shows promising results across subclasses by leveraging both the SE block and the special attention mechanism.

E. RESULT AND ANALYSIS

We evaluate the Sign-YOLO framework against previous leading methods by using the standard dataset, as presented in Table 4. Generally, one-stage detectors are more user-friendly than two-stage ones. The Sign-YOLO framework achieves superior accuracy on the GTSDB dataset by dynamically enhancing the model's capacity to comprehend and interpret intricate visual cues in natural scenes. This, in turn, improves the accuracy and resilience of TSD. Our framework can be seamlessly integrated into real-world applications without requiring specific adjustments. Experimental results show

FIGURE 5. Instances of accurate detections in a road environment with small, medium, and large traffic signs belonging to various categories, as generated by the Sign-YOLO framework. All detections shown in these examples are accurate.

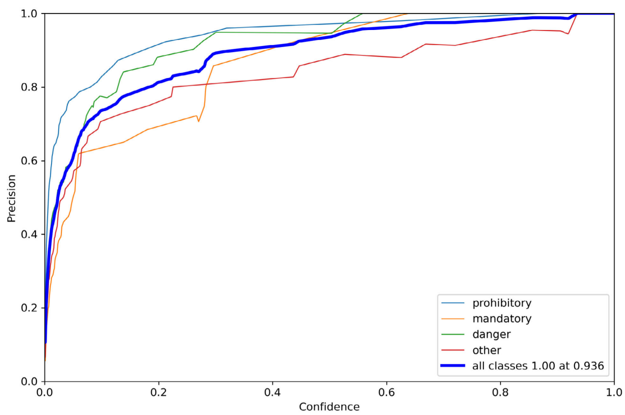

FIGURE 6. Precision.

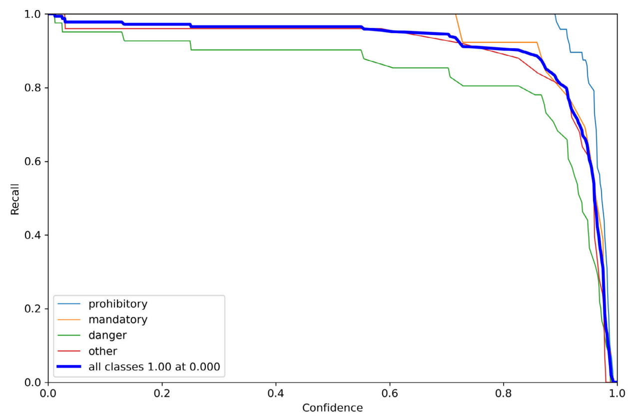

FIGURE 7. Recall.

that our approach outperforms existing methods, demonstrating remarkable performance and a notable 98% reduction in memory consumption.

The results in Tables 1–5 clearly show that the proposed Sign-YOLO method outperforms current SOTA techniques. Figure 5 shows accurately identified signs within ordinary scene photographs. The visualized detections have a score exceeding the threshold of 0.5. These figures depict three typical scenarios, including a road scene featuring traffic signs of small, medium, and large size. The precision and recall curves calculated using Equations (3) and (4) are respectively shown in Figures 6 and 7 for the GSTDB dataset, and they indicate that the proposed method improves the accuracy.

V. CONCLUSION

In this study, we introduce a real-time TSD network built upon a refined version of YOLOv7. This network demonstrates superior detection performance compared to contemporary one- and two-stage detectors. The proposed Sign-YOLO architecture enhances both the capability to extract information from feature maps and its effectiveness in representing multiscale objects during detection. We emphasize the significance of strong extracted features.

Precise and resilient visual features are crucial for detecting traffic signs. The Sign-YOLO approach significantly boosts the performance of the feature extractor, producing durable attributes. In contrast to previous models, Sign-YOLO demonstrates superior feature capacity. Experimental results confirmed that it achieved cutting-edge performance with rapid inference speed and accuracy. Specifically, Sign-YOLO significantly diminishes the model size by 98% compared to the case of two-stage detectors, making it highly effective and practical for real-time applications. The proposed architecture exhibits promising SOTA accuracy, demonstrating a 3.33% improvement on the GTSDB dataset. In future work, we aim to investigate a detection model that offers enhanced performance for high-speed moving targets.

REFERENCES

[1] Z. Huang, Y. Yu, J. Gu, and H. Liu, "An efficient method for traffic sign recognition based on extreme learning machine," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 920–933, Apr. 2017.

[2] J. F. Khan, S. M. A. Bhuiyan, and R. R. Adhami, "Image segmentation and shape analysis for road-sign detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 83–96, Mar. 2011, doi: 10.1109/TITS.2010.2073466.

[3] C. Liu, F. Chang, and Z. Chen, "Rapid multiclass traffic sign detection in high-resolution images," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2394–2403, Dec. 2014, doi: 10.1109/TITS.2014.2314711.

[4] F. Larsson, M. Felsberg, and P. E. Forssen, "Correlating Fourier descriptors of local patches for road sign recognition," *IET Comput. Vis.*, vol. 5, no. 4, Jul. 2011, Art. no. 244254.

[5] F. Larsson and M. Felsberg, "Using Fourier descriptors and spatial models for traffic sign recognition," in *Proc. 17th Scandin. Conf. Image Anal.*, 2011, pp. 238–249.

[6] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A robust, coarse-tofine traffic sign detection method," in *Proc. Int. Joint Conf. Neural Netw.*, Aug. 2013, p. 15.

[7] T. Chen and S. Lu, "Accurate and efficient traffic sign detection using discriminative AdaBoost and support vector regression," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, Jun. 2016, Art. no. 40064015.

[8] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, p. 2137.

[10] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, Jun. 2017, Art. no. 11371149.

[12] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jul. 2015, pp. 1440–1448.

[13] A. Arcos-Garcia, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems," *Neurocomputing*, vol. 316, Nov. 2018, Art. no. 332344.

[14] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German traffic sign detection benchmark," in *Proc. Int. Joint Conf. Neural Netw.*, Aug. 2013, p. 18.

[15] Q. Tang, G. Cao, and K. -H. Jo, "Integrated feature pyramid network with feature aggregation for traffic sign detection," *IEEE Access*, vol. 9, pp. 117784–117794, 2021, doi: 10.1109/ACCESS.2021.3106350.

[16] C. Y. Wang, A. Bochkovskiy, and H. Y. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Oct. 2023, pp. 7464–7475.

[17] J. Wang, Y. Chen, Z. Dong, and M. Gao, "Improved YOLOv5 network for real-time multi-scale traffic sign detection," *Neural Comput. Appl.*, vol. 35, no. 10, pp. 7853–7865, 2023.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[19] H. Hong, Y. Zhou, X. Shu, and X. Hu, "CCSPNet-Joint: Efficient joint training method for traffic sign detection under extreme conditions," 2023, *arXiv:2309.06902*.

[20] X. Ning, K. Gong, W. Li, L. Zhang, X. Bai, and S. Tian, "Feature refinement and filter network for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3391–3402, Sep. 2021, doi: 10.1109/TCSVT.2020.3043026.

[21] X. Ning, P. Duan, W. Li, and S. Zhang, "Real-time 3D face alignment using an encoder–decoder network with an efficient deconvolution layer," *IEEE Signal Process. Lett.*, vol. 27, pp. 1944–1948, 2020, doi: 10.1109/LSP.2020.3032277.

[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[23] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, and X. Tang, "DeepID-Net: Deformable deep convolutional neural networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2403–2412.

[24] F. Shao, X. Wang, F. Meng, T. Rui, D. Wang, and J. Tang, "Real-time traffic sign detection and recognition method based on simplified Gabor wavelets and CNNs," *Sensors*, vol. 18, no. 10, p. 3192, Sep. 2018, doi: 10.3390/s18103192.

[25] F. Shao, X. Wang, F. Meng, J. Zhu, D. Wang, and J. Dai, "Improved faster R-CNN traffic sign detection based on a second region of interest and highly possible regions proposal network," *Sensors*, vol. 19, no. 10, p. 2288, May 2019.

[26] J. Zhang, M. Huang, X. Jin, and X. Li, "A real-time Chinese traffic sign detection algorithm based on modified YOLOv2," *Algorithms*, vol. 10, no. 4, p. 127, Nov. 2017, doi: 10.3390/a10040127.

[27] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1951–1959.

[28] Z. Liu, C. Shen, M. Qi, and X. Fan, "SADANet: Integrating scale-aware and domain adaptive for traffic sign detection," *IEEE Access*, vol. 8, pp. 77920–77933, 2020, doi: 10.1109/ACCESS.2020.2989758.

[29] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection–SNIP," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3578–3587.

[30] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[31] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 1542–1547.

[32] H. Zhang and Q. M. J. Wu, "Pattern recognition by affine legendre moment invariants," in *Proc. 18th IEEE Int. Conf. Image Process.*, Brussels, Belgium, Sep. 2011, pp. 797–800, doi: 10.1109/ICIP.2011.6116676.

[33] J.-J. Lv, C. Cheng, G.-D. Tian, X.-D. Zhou, and X. Zhou, "Landmark perturbation-based data augmentation for unconstrained face recognition," *Signal Process., Image Commun.*, vol. 47, Sep. 2016, Art. no. 465475.

[34] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 633–647, Apr. 2014, doi: 10.1007/s00138-011-0391-3.

[35] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1484–1497, Dec. 2012.

[36] A. Youssef, D. Albani, D. Nardi, and D. D. Bloisi, "Fast traffic sign recognition using color segmentation and deep convolutional networks," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.*, 2016, pp. 205–216.

[37] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2110–2118.

[38] F. Jurišic, I. Filkovic, and Z. Kalafatic, "Multiple-dataset traffic sign classification with OneCNN," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 614–618.

[39] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.

[40] Y. Yang and F. Wu, "Real-time traffic sign detection via color probability model and integral channel features," in *Proc. Pattern Recognit., 6th Chin. Conf.*, 2014, pp. 545–554.
[41] J. Mirzapour Kaleybar, H. Khaloo, and A. Naghipour, "Efficient vision transformer for accurate traffic sign detection," 2023, *arXiv:2311.01429*.
[42] D. Tabernik and D. Skocaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1427–1440, Apr. 2020.
[43] D. Temel, G. Kwon, M. Prabhushankar, and G. AlRegib, "CURE-TSR: Challenging unreal and real environments for traffic sign recognition," 2017, *arXiv:1712.02463*.
[44] L. Jöckel, M. Kläs, and S. Martínez-Fernández, "Safe traffic sign recognition through data augmentation for autonomous vehicles software," in *Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2019, pp. 540–541.
[45] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1498–1506, Dec. 2012.
[46] S. Kim and S. Kwon, "Improvement of traffic sign recognition by accurate ROI refinement," in *Proc. 15th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2015, pp. 926–928.
[47] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[48] S. Zhao, Y. Yuan, X. Wu, Y. Wang, and F. Zhang, "YOLOv7-TS: A traffic sign detection model based on sub-pixel convolution and feature fusion," *Sensors*, vol. 24, no. 3, p. 989, Feb. 2024.
[49] J. Zhang, Y. Lv, J. Tao, F. Huang, and J. Zhang, "A robust real-time anchor-free traffic sign detector with one-level feature," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 2, pp. 1437–1451, Apr. 2024, doi: 10.1109/tetci.2024.3349464.

RUTURAJ MAHADSHETTI received the B.E. degree in computer science and engineering from Shivaji University, India, in 2020, and the M.S. degree in artificial intelligence from Chonnam National University, Republic of Korea, in 2023. He is currently a Researcher with the Graduate School of Data Science, Chonnam National University. His research interests include computer vision, image processing, the AIoT, and deep learning.

JINSUL KIM received the B.S. degree in computer science from the University of Utah, Salt Lake City, Utah, USA, in 1998, and the M.S. and Ph.D. degrees in digital media engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2004 and 2008, respectively. Previously, he was a Researcher with the Broadcasting/Telecommunications Convergence Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, from 2004 to 2009, and was a Professor with Korea Nazarene University, Cheonan-si, South Korea, from 2009 to 2011. He is currently a Professor with Chonnam National University, Gwangju, South Korea. He is a member of Korean national delegation for ITU-T SG13 international standardization. He is also the Co-Research Director of the AI Innovation Hub Research and Development Project hosted by Korea University and the Director of the G5-AICT Research Center.

TAI-WON UM received the B.S. degree in electronic and electrical engineering from Hongik University, Seoul, South Korea, in 1999, and the M.S. and Ph.D. degrees from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2000 and 2006, respectively. From 2006 to 2017, he was a Principal Researcher with the Electronics and Telecommunications Research Institute (ETRI), a leading government institute on information and communications technologies in South Korea. He is currently an Associate Professor with Chonnam National University, Gwangju, South Korea. He has been actively participating in standardization meetings, including ITU-T SG 20 (Internet of Things, smart cities, and communities). His research interests include the IoT, trust management, digital twin, and metaverse.