IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 July 2024, accepted 10 August 2024, date of publication 21 August 2024, date of current version 2 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3446990

RESEARCH ARTICLE

Traffic Sign Detection Under Adverse Environmental Conditions Based on CNN

QIANG GAO[1], HUIPING HU[2], AND WEI LIU[1]
[1]School of Information Engineering, Guangzhou Railway Polytechnic, Guangzhou, Guangdong 511300, China
[2]Department of Primary Education, Shangrao Preschool Education College, Shangrao, Jiangxi 334001, China

Corresponding author: Qiang Gao (gaoq11@qq.com)

The research was supported in part by Guangzhou Quality Engineering Project under Grant 3017221030, and in part by Guangzhou Railway Polytechnic Newly Introduced Talents Scientific Research Start-Up Project under Grant GTXYR2401 and Grant GTXYR2427.

ABSTRACT A strong and reliable Traffic Sign Detection and Recognition (TSDR) system is essential for the effective deployment of autonomous driving technology. In this field, numerous scholars have conducted extensive research, but current studies only consider TSDR under ideal conditions, neglecting scenarios such as rain, snow, fog, which can cause image blurring. This paper investigates the challenges of TSDR performance degradation caused by five adverse environmental conditions: rain, snow, fog, lens dirt, and lens blur. To overcome the adverse effects of these conditions on TSDR, this paper proposes a Convolutional Neural Network (CNN)-based TSDR method, consisting of three modules: adverse environment classification module, image enhancement module, and traffic sign detection module. The adverse environment classifier, based on the VGG19 architecture, identifies whether the image includes the aforementioned five adverse weather conditions. The image enhancement module named Enhance-Net enhances each of the five adverse environments separately and specifically enhances the traffic sign regions within the image, rather than the entire image area. To increase the speed of the proposed method, the traffic sign detection module utilizes the YOLOv4 framework. The proposed method's effectiveness is assessed using the CURE-TSD dataset, which includes traffic videos recorded in various adverse environmental conditions. Experimental results demonstrate that under five different levels of adverse environments, the pro-posed method achieves 95.03% accuracy and runs at a rate of 12.79 fps (frames per second). In contrast to the current benchmark, although there is a 2.81% reduction in accuracy resulting from training the proposed method on a subset of the dataset, the speed has increased by 12.03 fps, demonstrating the efficacy of the proposed approach.

INDEX TERMS CNN, adverse environments, image enhancement, TSDR.

I. INTRODUCTION

Autonomous driving has emerged as a transformative technology poised to revolutionize the transportation industry by enhancing safety, efficiency, and convenience. Central to the functioning of autonomous vehicles is the ability to accurately perceive the surrounding environment using a diverse array of onboard sensors. These sensors, which include radar, ultrasonic sensors, GPS (Global Positioning System), magnetometers, and cameras, collectively provide comprehensive data that enable the vehicle to navigate complex roadways autonomously [1]. The seamless integration of data from these sensors allows the autonomous driving system to dynamically plan paths, respond to real-time road conditions, and ensure automatic, safe, and reliable vehicle operation. This multi-sensor approach is critical for the robust and accurate perception necessary for autonomous driving.

The architecture of an autonomous driving system comprises three fundamental components: the positioning and navigation system, the environmental perception system, and the planning and control system. The positioning and navigation system, leveraging GPS and magnetometers, determines the vehicle's precise location and charts its course. Meanwhile, the environmental perception system utilizes radar, cameras, and ultrasonic sensors to detect and interpret obstacles, traffic signs, and other vehicles on the road. The

The associate editor coordinating the review of this manuscript and approving it for publication was Krishna Kant Singh.

planning and control system synthesizes this information to make real-time decisions regarding speed, direction, and braking, thereby safely guiding the vehicle to its destination. Among these components, the visual perception system, which includes traffic sign recognition, plays an indispensable role in ensuring the safety and efficiency of autonomous driving. Accurate recognition of traffic signs is crucial for adhering to road regulations and enhancing driver awareness, especially within the framework of Advanced Driver Assistance Systems (ADAS) [2].

Leading technology companies such as Google, Uber, Tesla, Volkswagen, Hyundai, NVIDIA, and Baidu are at the forefront of developing and testing autonomous driving systems. These companies are not only advancing the technology but also actively implementing pilot programs and commercial applications worldwide. For instance, in countries like China, the United States, and Germany, autonomous vehicles are already being deployed for taxi services and long-distance cargo transportation. These real-world applications highlight the practical benefits and potential of autonomous driving technology. However, the deployment of these systems in varied and often unpredictable real-world conditions poses significant challenges. One of the most critical challenges is ensuring that autonomous vehicles can operate reliably under diverse environmental conditions, such as rain, snow, fog, and dirt, which can adversely affect the quality of traffic sign recognition and overall sensor performance.

To address these challenges, extensive research focused on traffic sign recognition in complex environments is imperative. Such research is vital for improving the robustness and reliability of environmental perception systems in autonomous vehicles. Enhancing traffic sign recognition capabilities under adverse weather and lighting conditions will significantly boost the safety and operational effectiveness of autonomous driving technology. Moreover, advancements in this area will facilitate broader applicability of autonomous vehicles across various road environments, paving the way for wider adoption and integration of this transformative technology. The ongoing research and innovation in this field are crucial for achieving the goal of fully autonomous, safe, and reliable vehicles, capable of navigating the complexities of real-world environments.

II. LITERATURE REVIEW

In this section, we briefly review related topics: 1) traditional methods for traffic sign recognition; 2) deep learning-based traffic sign recognition.

A. TRADITIONAL METHODS FOR TRAFFIC SIGN RECOGNITION

Vitabile et al. [3] perform traffic sign detection in the Hue Saturation Value (HSV) color space, introducing dynamic thresholds for color segmentation. The method employing dynamic thresholds for segmentation can mitigate the impact of lighting variations on traffic sign detection, but it suffers from high computational complexity. Chakraborty and Yeb [4] initially utilize the YCbCr color model to mitigate the lighting sensitivity of image segmentation, then apply statistical thresholds for color segmentation, followed by labeling and filtering for shape extraction, and finally employ dis-tance vectors to the boundary to validate the extracted regions of interest. This method is advantageous for its simplicity and high accuracy but is relatively slow. Lu [5] extracts the outer contour of traffic signs and employs polygon approximation based on the Douglas Peucker (DP) algorithm and shape features for sign detection. While this method exhibits good detection performance, it is susceptible to false positives due to environmental influences. Alam and Jaffery [6] utilize the nearest neighbor matching method for traffic sign recognition, where features extracted are compared with those in a traffic sign feature database. SURF (Speeded-Up Robust Features) features are employed in the nearest neighbor matching method for traffic sign identification, offering scale, translation, and rotation invariance along with fast processing speed. While this method achieves high recognition accuracy, its results are prone to distortion and occlusion effects. Boi [7] propose a ''HOG+SVM'' method, which utilizes a preprocessing module to extract Histograms of Oriented Gradients (HOG) features from images, followed by analysis of the extracted features using the Support Vector Machines (SVM) algorithm. This method exhibits good stability but suffers from slow image processing due to high computational complexity. Xu et al. [8] propose a multi-class AdaBoost-based Extreme Learning Machine (ELM) ensemble algorithm, which offers better learning speed and generalization performance compared to the SVM algorithm. This algorithm achieves high recognition accuracy with relatively lower computational complexity.

In summary, traditional traffic sign recognition methods offer the benefits of straightforward principles and high accuracy. However, they suffer from drawbacks such as high computational complexity, relatively slow recognition speed, and poor performance in complex scenarios.

B. DEEP LEARNING-BASED TRAFFIC SIGN RECOGNITION

The use of deep learning for traffic sign recognition has become mainstream. Deep learning methods are classified into two-stage object detection methods and one-stage object detection methods.

R-CNN (Region-based Convolutional Neural Networks), Fast R-CNN, Faster R-CNN, are typical classic two-stage object detection algorithms. An improved traffic sign recognition method based on Faster R-CNN is proposed by Huang and Feng [9], replacing the VGG16 network with a residual network. Faster R-CNN utilizes a region proposal network for initial bounding box regression, followed by ROI pooling to generate proposal boxes, and finally outputs classification results and predicted boxes through fully connected layers. While this method is applicable in most cases, its recognition speed does not meet real-time requirements. Wang et al. [10] developed a recognition method utilizing Cascade R-CNN, in which the Cascade R-CNN algorithm

links various R-CNN detectors based on Intersection over Union (IoU). The improved algorithm incorporates the Feature Pyramid Networks (FPN) structure as a feature extraction network, replacing the original loss function. This method improves recognition accuracy, but the recognition speed is only 2.74 frames per second (fps), and it can only classify traffic signs into four categories. Yu [11] proposed an algorithm to improve the R-FCN (Region-based Fully Convolutional Network) model by pruning the model to retain only the first 25 convolutional layers. Due to the irregular shape of traffic signs, this algorithm utilizes deformable convolutions for feature extraction. The algorithm employs the K-means method to find suitable anchor boxes and utilizes an online hard example mining strategy to increase the learning intensity of hard examples. The improved algorithm achieves both accuracy and speed enhancements but can only classify signs into three major categories.

One-stage object detection algorithms can simultaneously perform bounding box regression and object classification. Representative algorithms include SSD (Single Shot MultiBox), YOLOv3, YOLOv4, YOLOv5, etc. Wang [12] proposed an im-proved FCOS (Fully Convolutional One-Stage Object Detection) model, which incorporates the FPN (Feature Pyramid Networks) structure into the network. This structure ensures that features of different scales are available for targets of corresponding sizes. The algorithm introduces a central sampling method to address the problem of a high proportion of negative samples generated during target recognition. The im-proved algorithm includes a self-attention module, enhancing the ability to identify regions of interest. However, the recognition speed of the algorithm is relatively slow. Peng et al. [13] proposed an improved RetinaNet algorithm, which incorporates a traditional feature pyramid module into the original algorithm, enabling the network to better distinguish foreground and background information. The improved algorithm utilizes focal loss as the classification loss function, allowing for thorough learning of positive samples. The algorithm adopts the lightweight MobileNetV2 network as the backbone network, dramatically decreasing the computational burden of the model. However, the recognition speed of the algorithm does not meet real-time requirements.

The authors in [14] proposed a network architecture that allows for simultaneous computation and localization of the region where the sign is located. In [15], the authors further divided localization and character recognition into two separate neural networks: the hybrid SegU-Net architecture, created based on SegNet [16] and U-Net [17], which employs the VGG16 architecture [18] as a locator for recognizing traffic signs. This algorithm only considers traffic sign recognition under ideal conditions and does not account for adverse environmental effects on sign recognition, thus lacking robustness. The study [19] improved upon the SegU-Net framework by using VGG16 to identify adverse environmental conditions in images, such as rain, snow, fog, or lens blur. If adverse conditions are detected, the images are enhanced to mitigate the negative impact on traffic sign recognition, resulting in higher road recognition accuracy. However, experimental results indicate that its recognition speed is slow, and re-al-time performance is poor.

To improve the recognition speed of traffic signs in complex environments, this study improved upon the method proposed in [19] by replacing the SegU-Net with YOLOv4 [20], effectively enhancing the speed of traffic sign recognition. Moreover, the size of the training samples was significantly reduced by removing regions from the images that do not contain any traffic sign characters.

III. PROPOSED METHOD

A. OVERVIEW OF THE METHOD

The method presented in this paper consists of three separate modules. The first module is the adverse environment classifier, which detects the types of adverse environments present in traffic images. The second module is the image enhancement module, which performs the necessary enhancements for different types of adverse environments. Lastly, the third module is responsible for detecting traffic signs within the image. The workflow of our proposed method is shown in Fig 1.

The process of traffic sign detection under adverse environments is as follows: Firstly, the image is inputted into the adverse environment classifier, which detects whether there are adverse environments present in the image. If the classifier detects the presence of any of the five adverse environments: rain, snow, fog, dirty lens, or lens blur, the image will be forwarded to the image enhancement module for adverse environment elimination. Images without adverse environments will not undergo enhancement. Then, the enhanced image or the image without adverse environments is inputted into the traffic sign detection module for sign detection. Finally, the module outputs the localization and name of the detected sign.

B. ADVERSE ENVIRONMENT CLASSIFIER

The adverse environment classifier utilizes a pretraining approach on the ImageNet dataset with VGG19, followed by fine-tuning on the CURE-TSD [21] dataset. Transfer learning leverages the characteristics of convolutional neural networks, where shallow layers recognize angles, edges, simple shapes, etc., while deeper layers recognize complex features. All information about image features is stored in the 16 convolutional layers, with generalization information stored in three fully connected output layers. Therefore, during transfer learning, only the last three layers are trained, while the remaining layers are not trained.

The feature extraction and the classification are two parts consisted for the adverse environment classifier. The feature extractor reduces the information from the image into a compact, low-dimensional feature space. These features are then employed in the classification phase to conduct the necessary classification.

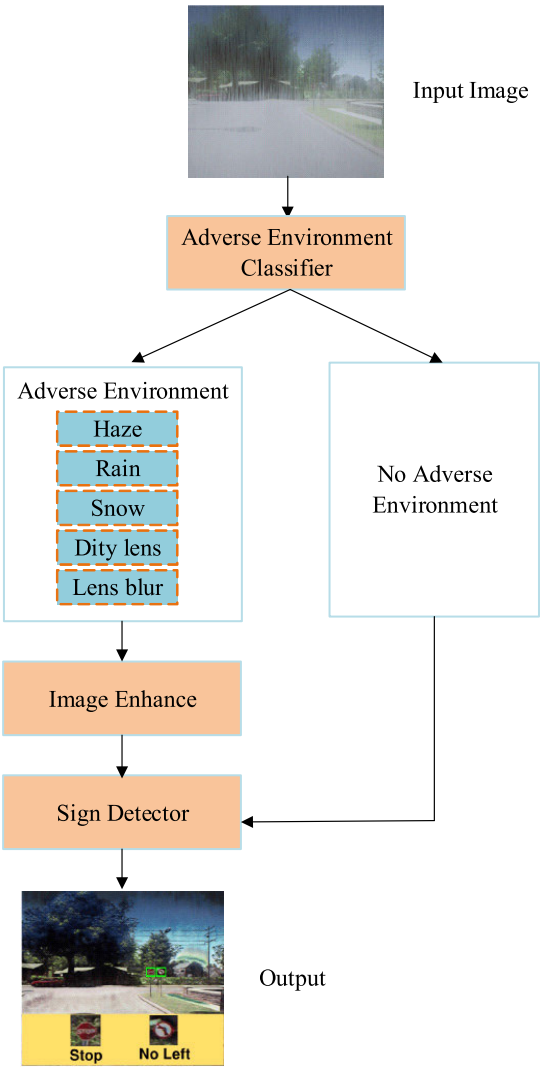

FIGURE 1. The workflow diagram of the proposed method in this paper.

Feature Extraction: The feature extractor includes 16 convolutional blocks, each consisting of a 3 × 3 convolutional kernel, batch normalization layer, and ReLU activation layer. Max-pooling operations are used for down-sampling, and global average pooling is applied in the final stage to further condense the features.

Classification: The classification component includes a fully connected layer that utilizes the extracted features for classification. The output comprises 6 classes: five different types of adverse environments and one class indicating no adverse environment, employing softmax activation.

Using frames with a resolution of 1236 × 1628 in the adverse environment detection stage im-poses a significant computational burden due to the size of the images. Additionally, the effects of adverse environments manifest as global features in the images. S Given that the adverse environment classifier primarily aims to discern various types of adverse environments, downsampling operations are expected to augment these overarching features. Thus, we resize all chosen adverse environment images to 512 × 512 pixels and employ them to train the adverse environment classifier with 6 classes: 5 classes for specified adverse environments and 1 class for no adverse environment. During training, we employ categorical cross-entropy as the loss function and optimize our network using Adam [22]. The initial learning rate is initialized at 10^{-4}, and if the validation score fails to improve over 3 epochs, a learning rate scheduler is implemented to halve the rate. Finally, following these specifications, we conduct training for our network over 30 epochs.

C. IMAGE ENHANCEMENT

The paper utilizes the Enhance-Net from [19] for implementing the image enhancement function. Enhance-Net employs five enhancement blocks to address five different types of adverse environments. Each block contains the same base CNN network architecture, which is independently trained for each of the five types of adverse environments. This allows for easy adaptation to additional types of adverse environments by adding more enhancement blocks. The structure of Enhance-Net is shown in Fig 2.

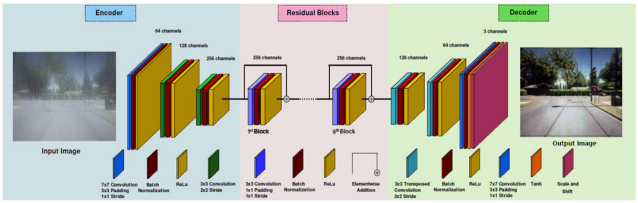

FIGURE 2. Enhance-Net architecture.

Encoder, residual blocks, and decoder are three parts consisted of Enhance-Net.

Encoder: The encoder begins with a convolutional block featuring a 7 × 7 convolutional kernel, followed by instance normalization and ReLU activation. Subsequently, there are two more convolutional blocks, each employing a 3 × 3 convolutional kernel, instance normalization, and ReLU activation. The convolutional blocks consist of 64, 128, and 256 kernels, with strides of 1×1, 2×2, and 2×2, respectively. The encoder transforms the image into latent feature maps, which are subsequently refined by the residual blocks.

Residual Blocks: Each residual block is composed of two convolutional layers using a 3 × 3 convolutional filter, followed by an instance normalization layer and a ReLU activation layer, incorporating a shortcut skip connection. Enhance-Net utilizes 9 residual blocks, facilitating faster convergence and minimization of the loss function.

Decoder: During each decoding step, the feature maps produced by the residual blocks undergo up-sampling through two transpose convolutional blocks. Each block consists of a 3 × 3 transpose convolution with a stride of 2 × 2, followed by an instance normalization layer and a ReLU activation. The final decoding stage includes a convolutional block with a 7 × 7 kernel size and a hyperbolic tangent (Tanh) activation function.

The overall enhancement of images containing adverse environments may not necessarily result in the optimal enhancement of traffic sign regions. Therefore, we modified the loss computation to prioritize the traffic sign regions over the entire image. According to research [23], employing Mean Absolute Error (MAE) as the loss function enhances image reconstruction performance.

Thus, the lower-level intermediate layers of a CNN pretrained on the ImageNet dataset can act as feature extractors for edges, contours, and low-level image details. We utilize a pretrained VGG19 network from ImageNet to extract features from intermediate layers for both the reconstructed image and the target traffic sign region. Subsequently, we aim to minimize the MAE between these reconstructed traffic sign features and the target traffic sign features. This minimization ensures that the reconstructed image accurately restores essential details [24]. Let R and T denote the recon-structed and target traffic sign regions, respectively, with H, W, and C representing their height, width, and channels. We define Len-hancement(sign) as follows:

$$L_{enhancement(sign)} = \frac{1}{H \times W \times C} \sum_{k=0}^{C} \sum_{j=0}^{W} \sum_{i=0}^{H} \left| R_{ijk} - T_{ijk} \right| \tag{1}$$

Another approach is the so-called perceptual loss function, where $\emptyset(R)$ and $\emptyset(T)$ represent the image features extracted from the seventh layer of the VGG19 network, pretrained on the ImageNet dataset, for the reconstructed image and the original image, respectively. The main idea behind using this function (2) is to preserve as much information as possible about the small portion of the sign, and the internal layers of the pretrained VGG19 network already contain the necessary feature information such as corners, edges, etc.

This paper adopts following equation as the loss function during training. adverse environment classifier utilizes a pretraining approach on the ImageNet dataset with VGG19, followed by fine-tuning on the CURE-TSD [21] dataset.

$$L_{content(sign)} = \frac{1}{H \times W \times C} \sum_{k=0}^{C} \sum_{j=0}^{W} \sum_{i=0}^{H} \left| \emptyset(R)_{ijk} - \emptyset(T)_{ijk} \right| \tag{2}$$

When training this module, first, we extract all frames containing specific adverse environ-mental types of traffic signs from the training video sequences. There are 5 difficulty levels, each with 29,400 frames, resulting in a total of $29{,}400 \times 5 = 147{,}000$ frames, with each frame size being 1236×1628 pixels. Next, we crop random blocks containing traffic signs with a size of 1024×1024 from these frames. Given hardware constraints, we train the model with a batch size of 1.

To compensate for the small batch size, we employ gradient accumulation. Gradients from 5 consecutive batches are accumulated before updating the weights, ensuring that these batches represent 5 distinct adverse environmental conditions. We use these images to train the CNN enhancement blocks with an initial learning rate of 10^{-4}. We use Adam [22] as our optimizer. The training spans 30 epochs. If the validation score fails to improve over 3 consecutive epochs, a learning rate scheduler reduces the learning rate by a factor of 0.5.

D. TRAFFIC SIGN LOCALIZATION AND RECOGNITION

The enhanced images or images without adverse environmental conditions are inputted into YOLOv4 for traffic sign detection. The principle of training the traffic sign localization and recognition module on the YOLOv4 architecture is similar to that of training the normalization module [20]. The images are divided into two equal segments of 1024×1024 pixels in the two upper corners of the image. The training is conducted for 4000 epochs, a large number necessitated by the architecture of the network itself, as each image undergoes multiple processing stages. The architecture of YOLOv4 is shown in Fig 3.

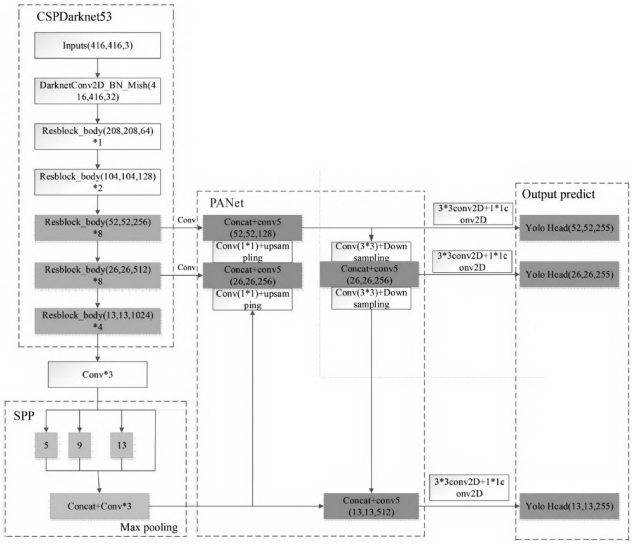

FIGURE 3. YOLO v4 architecture.

YOLO v4 is an improved version of the YOLO object detection model, offering better performance and accuracy compared to its predecessors. The architecture of YOLO v4 consists of several key components that work together to perform object detection efficiently.

1) BACKBONE

The backbone of YOLO v4 is responsible for extracting essential features from the input images. In YOLO v4, the CSPDarknet53 (Cross Stage Partial Darknet53) is used as the backbone. CSPDarknet53 is an improvement over the Darknet53 backbone used in YOLO v3. It incorporates the CSPNet (Cross Stage Partial Network) strategy, which helps to improve learning capability and reduce computation by partitioning the feature map of the base layer into two parts and merging them through a cross-stage hierarchy. This

backbone is designed to provide a good balance between accuracy and speed.

2) NECK

The neck of the YOLO v4 architecture is designed to generate feature pyramids, which are crucial for detecting objects at different scales. YOLO v4 utilizes several techniques in its neck component:

1) SPP (Spatial Pyramid Pooling): This helps in increasing the receptive field and separating out the most significant contextual features.

2) PAN (Path Aggregation Network): This network improves information flow and feature pyramid by fusing low-level features with high-level features, thereby enhancing the detection of small objects and improving the robustness of the detector.

3) HEAD

The head of the YOLO v4 architecture is where the actual object detection takes place. It predicts bounding boxes, class probabilities, and abjectness scores. The head component consists of multiple convolutional layers that output predictions for different scales. The predictions are made at three different scales, which helps in detecting objects of varying sizes.

4) ADDITIONAL TECHNIQUES AND OPTIMIZATIONS

YOLO v4 incorporates several additional techniques and optimizations to improve its performance. Some of these include:

1) Bag of Freebies: These are methods that improve detection accuracy without increasing the inference time. Techniques include CutMix and Mosaic data augmentation, Drop Block regularization, Class label smoothing, Cross mini-Batch Normalization (CmBN), Self-Adversarial Training (SAT), and more.

2) Bag of Specials: These are methods that slightly increase the inference cost but provide significant accuracy improvements. Techniques include Mish activation, CSP connections, MiWRC, CIoU loss, DropBlock, and PANet.

3) YOLO Loss Function: YOLO v4 uses a modified loss function that takes into account the bounding box regression loss, objectness loss, and class probability loss, improving the overall performance of the model.

5) ACTIVATION FUNCTIONS

YOLO v4 uses the Mish activation function in its architecture. Mish is a smooth, non-monotonic activation function that improves the flow of information and gradients through the network, leading to better accuracy.

IV. EXPERIMENTAL RESULTS

A. EXPERIMENTAL PLATFORM AND DATASET ANALYSIS

The hardware platform used in this experiment is the Dell PowerEdge T640, con-figured with an Intel Xeon Gold 6226R CPU, featuring 16 cores and a clock speed of 3.22GHz. It is equipped with an NVIDIA GeForce RTX 3080 GPU with 8GB of VRAM and 16GB of RAM. The operating system is Ubuntu 20.04 LTS. The software platform includes PyCharm 2022.2, Python 3.8, PyTorch 1.7, CUDA Toolkit 11.0, and cuDNN 8.0.

This paper employs the CURE-TSD dataset for experimental purposes, which comprises 49 video sequences, each containing 300 frames. Additionally, for each video sequence, there are 70 segments covering 14 weather conditions, with each condition having 5 severity levels. The study focuses on five common adverse environmental conditions: rain, snow, fog, dirty lens, and lens blur. Each adverse condition is further categorized into five levels, ranging from nearly all signs being clearly visible (minimal impact of adverse conditions on recognition quality) to signs in the far distance being almost entirely invisible. Therefore, to conduct recognition, 26 video sequences are required to capture each segment, totaling 1274 segments. Additionally, the camera's angle relative to the road needs to be considered, determining which part of the image actually contains specific traffic signs and how many pixels will never be used for recognition. Analysis of the dataset reveals that most traffic signs are situated on the right side along the OY axis, and the pixel area of the characters follows a nor-mal distribution, mainly ranging from 0 to 646 pixels. Thus, at this camera angle, approximately 52% of the entire image area can be cropped without sacrificing recognition accuracy.

Furthermore, each frame may contain one to five characters, with distances be-tween characters exceeding 1024 pixels. This indicates that the method used by the authors in the referenced paper [19], which involves randomly sampling image segments of 1024 × 1024 pixels, is only applicable when the characters in the image do not exceed one or two and are guaranteed to be within the same segment. Hence, this is likely to negatively impact the accuracy of character recognition. On the other hand, the authors in another referenced paper [13] opted for four segments of 1024 × 1024 pixels each, training the SegU-Net network by cropping images at the four corners (top and bottom). The minimum width of signs used to train the traffic sign locator is 8, while the maximum is 246. Thus, the minimum size of segments for training the second and third modules could be either 512 or 1024 (256 is not used as it might result in losing information about sign edges). The segment size must be a power of 2; otherwise, information about image context may be lost during the max-pooling operation in the convolutional network's internal layers. Reducing the segment size to 512 pixels would shrink the network's size but not decrease training time. Therefore, for this study, the training segment size is chosen as 1024 × 1024 pixels.

B. THE EFFECT OF IMAGE ENHANCEMENT

The image enhancement effect was tested using rain as an adverse condition, de-fining levels 2 (light rain), 3 (moderate rain), and 4 (heavy rain). Fig 4. to Fig 6. respectively

illustrate the image enhancement effects under these three rainy conditions. It can be observed from Figure 4 to Figure 6 that the Enhance-Net adopted in this paper exhibits good image enhancement effects under light rain, moderate rain, and heavy rain conditions. This is attributed to the excellent feature learning ability of Enhance-Net, which demonstrates good adaptability to image enhancement under different environmental conditions.

FIGURE 4. Image enhancement effect under light rain conditions.

FIGURE 5. Image enhancement effect under moderate rain conditions.

FIGURE 6. Image enhancement effect in heavy rain conditions.

Moreover, the enhancement module should primarily focus on the traffic sign region to improve traffic sign detection performance. To validate this, we proposed using the Structural Similarity Index (SSIM) between the enhanced traffic sign region and the traffic sign region in a normal environment, comparing both overall and localized enhancement methods. Table 1 shows the SSIM values for different severity levels of adverse conditions, representing the average values across all types of adverse conditions. From the table, it is evident that our proposed method attains a higher SSIM score of 0.82 for the entire traffic sign area, where as the overall enhancement method yields an SSIM score of 0.79. It is also evident that in both cases, as the severity in-creases, the SSIM values decrease. However, our proposed method exhibits less performance degradation compared to the conventional approach.

TABLE 1. Comparison of Structural Similarity Measure (SSIM) values for overall enhancement versus traffic sign area enhancement.

Leve	Overall Enhancement	Sign Area Enhancement
1	0.86	0.88
2	0.83	0.85
3	0.80	0.83
4	0.77	0.80
5	0.71	0.75
Average	0.79	0.82

TABLE 2. The accuracy of the proposed method under different adverse environmental conditions.

Adverse Environment	Level	Proposed Method (%)	Literature [17] (%)	Difference (%)
Normal Environment	N/A	95.91	99.02	-3.11
Rain	1	97.28	98.15	-0.87
	2	97.41	97.72	-0.31
	3	97.41	97.38	0.03
	4	96.17	96.88	-0.71
	5	96.05	96.03	0.02
	Average	96.86	97.23	-0.37
Snow	1	96.67	98.95	-2.28
	2	96.42	98.24	-1.82
	3	96.54	98.57	-2.03
	4	96.42	96.04	0.38
	5	93.29	94.94	-1.65
	Average	95.87	97.35	-1.48
Haze	1	95.68	98.31	-2.63
	2	94.08	98.76	-4.68
	3	93.71	98.85	-5.14
	4	93.02	98.38	-5.36
	5	92.83	97.74	-4.91
	Average	93.86	98.41	-4.55
Dirty lens	1	95.99	98.55	-2.56
	2	94.81	98.56	-3.75
	3	94.19	98.33	-4.14
	4	92.58	97.94	-5.36
	5	92.11	97.33	-5.22
	Average	93.94	98.14	-4.20
Lens blur	1	96,30	98,62	-2.32
	2	95,19	98,51	-3.32
	3	94.45	98.22	-3.77
	4	93.99	97.57	-3.58
	5	93.12	97.31	-4.19
	Average	94.61	98.05	-3.44
Average	N/A	95.03	97.84	-2.81

C. TRAFFIC SIGN DETECTION PERFORMANCE

Due to limitations in the experimental platform, the training set used by the algorithm in this paper is limited to the first 20 video sequences. To prevent overfitting, the training, testing, and validation sets consist of 16, 2, and 2 segments, respectively. As shown in Table 2 and Table 3, based on accuracy as the evaluation metric, comparisons were made for different levels of adverse environments and the same levels of adverse environments with different types,

TABLE 3. The accuracy of the proposed method on the CURE-TSD dataset testing set across various adverse environmental conditions at the same leve.

Level	Adverse Environ -ment	Proposed Method (%)	Literature [17] (%)	Differen ce (%)
1	Rain	97.28	98.15	-0.87
	Snow	96.67	98.95	-2.28
	Haze	95.68	98.31	-2.63
	Dirty lens	95.99	98.55	-2.56
	Lens blur	96.30	98.62	-2.32
	Average	96.38	98.51	-2.13
2	Rain	97.41	97.72	-0.31
	Snow	96.42	98.24	-1.82
	Haze	94.08	98.76	-4.68
	Dirty lens	94.81	98.56	-3.75
	Lens blur	95.19	98.51	-3.32
	Average	95.58	98.36	-2.78
3	Rain	97.41	97.38	0.03
	Snow	96.54	98.57	-2.03
	Haze	93.71	98.85	-5.14
	Dirty lens	94.19	98.33	-4.14
	Lens blur	94.45	98.22	-3.77
	Average	95.26	98.27	-3.01
4	Rain	96.17	96.88	-0.71
	Snow	96.42	96.04	0.38
	Haze	93.02	98.38	-5.36
	Dirty lens	92.58	97.94	-5.36
	Lens blur	93.99	97.57	-3.58
	Average	94.44	97.36	-2.92
5	Rain	96.05	96.03	0.02
	Snow	93.29	94.94	-1.65
	Haze	92.83	97.74	-4.91
	Dirty lens	92.11	97.33	-5.22
	Lens blur	93.12	97.31	-4.19
	Average	93.48	96.67	-3.19

TABLE 4. Performance of the proposed method in terms of runtime on the CURE-TSD dataset test set across various adverse environmental conditions.

Adverse Environment	Proposed Method (fps)	Literature [17] (fps)
Normal Environment	14.04	4.91
Rain	13.91	3.12
Snow	12.86	2.56
Haze	12.04	2.01
Dirty lens	12.15	2.14
Lens blur	11.74	1.79
Average	12.79	2.76

in addition to normal environments. The average accuracy tested on the validation set for the algorithm in this paper is 95.03%, while the average accuracy reported in [19] is 97.84%, resulting in a difference of 3.11%. This significant difference arises because [19] utilized the entire training set consisting of all 49 video segments for training. Furthermore, the accuracy reported in [19] for fog, dirty lens, and lens blur conditions is much higher than that of the algorithm in this paper, indicating that there is room for improvement in the recognition accuracy of small target traffic signs when using YOLOv4. Finally, both the proposed method in this paper and the method in [19] achieve the highest recognition accuracy under level 1 adverse conditions, indicating that adverse environments have a significant impact on the accuracy of detection algorithms on the validation set of the CURE-TSD dataset.

For further assessment of the proposed algorithm's efficacy, the evaluation running speed in normal and five different adverse environmental conditions was used as a metric, comparing the proposed algorithm with the method in reference [19]. The results are shown in Table 4. From Table 4, it can be observed that under normal conditions, both the proposed method and the method in reference [19] have the fastest running speeds, reaching 14.04 frames per second (fps) and 4.91 fps, respectively. However, in rainy, snowy, foggy, dirty lens, and blurry lens conditions, the running speed decreases. Finally, the overall running speed of the proposed algorithm averages 12.79 fps, while the average running speed of the method in reference [19] is 2.76 fps, indicating an average increase of 10.03 fps. This demonstrates that replacing SegU-Net with YOLOv4 significantly improves the detection speed of traffic signs.

V. CONCLUSION

In this paper, we proposed a modular CNN-based approach for traffic sign detection under various adverse environmental conditions. This method addresses the performance degradation of existing traffic sign detection algorithms under various ad-verse environmental conditions, effectively reducing the negative impact of adverse conditions on traffic sign detection. In our approach, the input image is passed through an adverse environmental classifier based on VGG19, and then optionally fed into an enhancement network, which restores useful features for successfully detecting the traffic sign region. Unlike existing methods based on overall image enhancement, our enhancement network is trained using an innovative loss function and training pipe-line that integrates Mean Absolute Error (MAE) specifically targeting the traffic sign region in both pixel and feature domains constrained by a sign detection loss, effectively ensuring enhanced detection of the traffic sign region. Through experiments, we have also shown that prioritizing enhancement of the traffic sign region leads to improved detection performance. Finally, the enhanced images are fed into the traffic sign detection module, where we adopt YOLOv4 as the detection framework. Experimental results show that compared to the study [19], the accuracy of YOLOv4 de-creases by an average of 2.81% under each type of adverse environmental condition. This is due to the incomplete training set used in our approach. However, the detection speed increases from 2.76 frames per second in [17] to 12.79 frames per second, indicating that the

algorithm's detection speed can be effectively improved with minimal loss in detection accuracy.

Our method adopts a modular design where each module can be independently designed and operated, which expands the algorithm design space. Moving forward, we aim to explore and optimize the architecture of each module to effectively handle all adverse environmental conditions found in the CURE-TSD dataset.

In the next phase of our research, we will focus on refining the modular architecture of our CNN-based approach to further enhance traffic sign detection under adverse environmental conditions. Specifically, we plan to optimize the adverse environmental classifier and enhancement network to better adapt to varying levels of environmental challenges, leveraging advanced techniques such as transfer learning and domain adaptation. Additionally, we intend to extend our training set to cover a broader range of environmental conditions and improve the robustness of our detection framework. By conducting extensive experiments with the updated CURE-TSD dataset, we aim to achieve a more balanced trade-off between detection accuracy and speed, ultimately ensuring our method's applicability in real-world scenarios.

REFERENCES

[1] H. H. Meng, H. B. Jiang, and T. B. Tang, "Global development status and trends of autonomous driving (Part 1)," *East China Sci. Technol.*, vol. 23, no. 9, pp. 66–68, 2014.

[2] 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.

[3] S. Vitabile, G. Pollaccia, G. Pilato, and E. Sorbello, "Road signs recognition using a dynamic pixel aggregation technique in the HSV color space," in *Proc. 11th Int. Conf. Image Anal. Process.*, 2001, pp. 572–577.

[4] S. Chakraborty and K. Deb, "Bangladeshi road sign detection based on YCbCr color model and DtBs vector," in *Proc. Int. Conf. Comput. Inf. Eng. (ICCIE)*, Nov. 2015, pp. 158–161.

[5] Y. J. Lu, "Research on automatic detection and recognition algorithms for traffic signs," M.S. thesis, School Inf. Sci. Eng., Wuhan Univ. Sci. Technol., Wuhan, China, 2015.

[6] A. Alam and Z. A. Jaffery, "Indian traffic sign detection and recognition," *Int. J. Intell. Transp. Syst. Res.*, vol. 18, no. 1, pp. 98–112, Jan. 2020.

[7] F. Boi and L. Gagliardini, "A support vector machines network for traffic sign recognition," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 2210–2216.

[8] Y. Xu, Q. Wang, Z. Wei, and S. Ma, "Traffic sign recognition based on weighted ELM and AdaBoost," *Electron. Lett.*, vol. 52, no. 24, pp. 1988–1990, Nov. 2016.

[9] L. Q. Huang and S. T. Feng, "Improving faster R-CNN for traffic sign recognition," *Laser J.*, vol. 41, no. 4, pp. 57–60, 2020.

[10] H. Wang, K. Wang, Y. F. Cai, Z. Liu, and L. Chen, "Traffic sign recognition based on improved cascaded convolutional neural network," *Autom. Eng.*, vol. 43, no. 9, pp. 1256–1262, 2020.

[11] Q. T. Yu, "Research on traffic sign detection algorithm based on convolutional neural networks," M.S. thesis, School Inf. Eng., Nanchang Univ., Nanchang, China, 2021.

[12] F. Wang, "Research on road traffic sign detection method based on deep learning," M.S. thesis, School Transp., Beijing Jiaotong Univ., Beijing, China, 2021.

[13] J. Y. Peng, Z. Y. Liu, C. X. Feng, F. Fang, Z. W. Luo, Y. Y. Liu, and J. Qin, "Scale aware bidirectional feature pyramid network for traffic sign detection," *J. Comput. Aided Design Graph.*, vol. 34, no. 1, pp. 133–141, 2022.

[14] H. S. Lee and K. Kim, "Simultaneous traffic sign detection and boundary estimation using convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1652–1663, May 2018.

[15] U. Kamal, T. I. Tonmoy, S. Das, and Md. K. Hasan, "Automatic traffic sign detection and recognition using SegU-net and a modified Tversky loss function with L1-constraint," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1467–1479, Apr. 2020.

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 21st Med. Image Comput. Comput. Assist. Intervent. Int. Conf.*, 2015, pp. 234–241.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[19] S. Ahmed, U. Kamal, and Md. K. Hasan, "DFR-TSD: A deep learning based framework for robust traffic sign detection under challenging weather conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5150–5162, Jun. 2022.

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

[21] D. Temel, T. Alshawi, M. H. Chen, and G. AlRegib. (Sep. 2021). *CURE-TSD: Challenging Unreal and Real Environments for Traffic Sign Detection*. IEEE Dataport. [Online]. Available: https://ieee-dataport.org/open-access/cure-tsd-challenging-unreal-and-real-environment-traffic-sign-detection

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[23] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for neural networks for image processing," 2015, *arXiv:1511.08861*.

[24] H. Zhang and V. M. Patel, "Densely connected pyramid dehazing network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3194–3203.

QIANG GAO received the master's degree from South China Normal University, in 2010. He is currently an Associate Professor with the School of Information Engineering, Guangzhou Railway Polytechnic, Guangzhou, China. He has published several articles, such as computer system applications and computer technology and development. His research interests include pattern recognition, image recognition, deep learning, and image enhancement.

HUIPING HU received the Ph.D. degree in educational management from the Graduate School, St. Paul University Philippines. She is currently an Associate Professor with the Department of Primary Education, Shangrao Preschool Education College, Shangrao, China. She has published several journal articles, such as *Applied Mathematics and Nonlinear Science* and other Chinese journals. Her research interests include English education, vocational English teaching, and language deep learning.

WEI LIU received the master's degree from South China Normal University, in 2008. She is currently an Associate Professor with the School of Information Engineering, Guangzhou Railway Polytechnic, Guangzhou, China. Her research include artificial intelligence and big data technology.