

DFR-TSD: A Deep Learning Based Framework for Robust Traffic Sign Detection Under Challenging Weather Conditions

Dr. D. B. K. Kamesh¹, S Meghana², S. Neha³, S. Nithya⁴, Sk. Sharmila⁵

Professor, Dept of Computer Science and Engineering Hyderabad, Telangana, India

^{2,3,4,5} UG Scholar, Dept of Computer Science and Engineering, Malla Reddy Engineering College for Women, Secunderabad

meghana16112@gmail.com, nehabhithu@gmail.com, nithyasanganabhatla@gmail.com, shaiksharmila84473@gmail.com

Abstract— Robust traffic sign detection and recognition (TSDR) is of paramount importance for the successful realization of autonomous vehicle technology. The importance of this task has led to vast amount of research efforts and many promising methods have been proposed in the existing literature. However, most of these methods have been evaluated on clean and challenge-free datasets and overlooked the performance deterioration associated with different challenging conditions (CCs) that obscure the traffic-sign images captured in the wild. In this paper, we look at the TSDR problem under CCs and focus on the performance degradation associated with them. To this end, we propose a Convolutional Neural Network (CNN) based prior enhancement focused TSDR framework. Our modular approach consists of a CNN-based challenge classifier, Enhance-Net—an encoder-decoder CNN architecture for image enhancement, and two separate CNN architectures for sign- detection and classification. We propose a novel training pipeline for Enhance-Net that focuses on the enhancement of the traffic sign regions (instead of the whole image) in the challenging images subject to their accurate detection. We used CURE-TSD dataset consisting of traffic videos captured under different CCs to evaluate the efficacy of our approach. We experimentally show that our method obtains an overall precision and recall of 91.1% and 70.71% that is 7.58% and 35.90% improvement in precision and recall, respectively, compared to the current benchmark. Furthermore, we compare our approach with different CNN- based TSDR methods and show that our approach outperforms them by a large margin.

Index Terms— Traffic sign detection, traffic sign recognition, convolutional neural network, challenging condition, enhancement, modular approach.

I. INTRODUCTION

TRAFFIC sign detection and recognition play a crucial part in driver assistance systems and autonomous vehicle technology. One of the major prerequisites of safe and widespread implementation of this technology is a TSDR algorithm that is not only accurate but also robust and reliable in a variety of real-world scenarios. However, in addition to the large variation among the traffic signs to detect, the traffic images that are captured in the wild are not ideal and often obscured by different adverse weather conditions and motion artifacts that substantially increase the difficulty level of this

Being a challenging research problem, several studies have been carried out on TSDR. A detailed overview of these studies can be found in [1] and [2]. The whole task of TSDR can

be subdivided into two independent tasks: Traffic Sign Detection (TSD) and Traffic Sign Recognition (TSR). Traditional research methodologies of TSD mostly rely on manual feature extraction of various attributes such as geometrical shapes, edge detection, and color information. Color-based approach mostly comprises of threshold-based segmentation of traffic sign region in a particular color space such as Hue-Saturation-Intensity (HSI) [3], Hue-Chroma-Luminance (HCL) [4] and others [5]. However, one major drawback of these color- based approaches is that these are highly susceptible to the change in illumination that can frequently occur in real- world scenarios [6]. To overcome this challenge, shape-based approaches have been extensively used in the existing literature that comprise of Canny-Edge detection [7], Histogram- Oriented Gradients (HOG) [5], Haar-Wavelet features [8] and Fast Fourier Transform (FFT) [9]. However, size and scale variation, disorientation and occlusions of traffic sign regions due to the motion artifacts during real-time video feed hinder the practical application of these approaches. On the other hand, the methodologies for TSR utilize color and/or shape- based features to train classifiers such as Random Forest [10] and Support Vector Machine (SVM) [5]. However, there is no comprehensive method to identify the best features set and the best classifier for TSR.

Recent advancement of deep learning (DL) in different computer vision tasks such as image recognition [11], image segmentation [12] and object detection problems [13] has led to a large scale adoption of these algorithms for TSDR [14]–[16]. Lee *et al.* have used a custom designed CNN to simultaneously detect and estimate the boundary of the traffic sign regions [17]. Compared to the objects that frequently appear in the existing object detection datasets, traffic sign regions are very small and thus have a very small region of interest (ROI) to background ratio. To address this challenge, Yuan *et al.* have proposed a CNN-based multi-resolution feature fusion architecture [18]. They have also proposed a vertical spatial sequence attention (VSSA) module to gain more context information for better detection performance. A major advantage of DL-based methods is that they are

completely data-driven that does not require any manual feature engineering. In order to facilitate the development and evaluation of these algorithms, a number of datasets have been introduced such as GTSDb [19], LISA [1], BelgiumTS [20] and TT100K [20]. However, neither these methods nor these datasets have considered sufficient types and levels of CCs that can frequently arise during the capture of the traffic image [6]. In order to overcome this shortcoming of the existing datasets, Temel *et al.* have introduced a video dataset that has considered both the Challenging Unreal (stimulated) and Real Environments for Traffic Sign Detection (CURE-TSD) [21]. This dataset comprises of different CCs that can commonly arise in practical scenarios with more than 1.7 million frames that makes it by far the largest dataset available for traffic sign detection research. And in [6], Temel *et al.* have analyzed the effect of various CCs on the performance of benchmark algorithms. They have presented the two benchmarks provided by the top two winners of the IEEE Video and Image Processing (VIP) Cup 2017 and showed that the CCs can reduce the F_2 score by 65%. Such significant performance degradation

undoubtedly calls for special attention to these challenges.

In [22], Kamal *et al.* have proposed a deep CNN-based modular approach that achieves very promising performance on this dataset under challenge free conditions. The authors have also highlighted their performance degradation associated with the CCs. A straightforward solution to alleviate this problem is to use an image enhancer that retains proper color and shape information corrupted by various CCs. Many researches have been conducted on image enhancement task for the purpose of better visibility. However, in most of these studies, only one challenge case is considered [23]–[25]. In very few studies, two challenge cases are considered [26], [27]. Therefore, the development of a robust image enhancer that can address multiple challenge cases with varying severity is one of the most important considerations for accurate sign detection in real-world scenarios.

To alleviate this problem, we propose Enhance-Net, a deep CNN-based image enhancer, that performs a prior enhancement of these traffic images. In [6], Temel *et al.* has stated that all of the twelve different types of CCs present in the dataset need not be addressed at a time and also highlight the fact that benchmark algorithms performance is more vulnerable in challenging weather conditions such as Rain, Snow, and Haze. In addition, Lens blur and Dirty Lens are two frequently occurring CCs in the real world scenarios. Therefore, in this work, we highlight the focus of our approach on five different CCs: Rain, Snow, Haze, Dirty lens, and Lens blur associated with five different levels of severity. Due to the different nature of these challenges, training a single network for all these challenges may result in suboptimal performance. Therefore, we use five different Enhance-Nets trained with a single type of challenge at a time to ensure the best possible enhancement for each of the CCs. We incorporate the modular TSDR approach proposed in [22] for sign detection and classification from the enhanced traffic images. Therefore, our approach consists of four separate modules— challenge classifier, enhancement blocks, sign detector and sign classifier. The challenge classifier classifies the challenge present

in the image and forwards it to the appropriate enhancement block that enhances the corresponding challenging image. The enhanced image is then passed through the sign detector that localizes the traffic signs and the sign classifier assigns them to the appropriate sign class. The challenge classifier that classifies these challenges can also detect the challenge-free class and therefore our approach is robust in challenge-free condition as well.

In addition, we propose a novel training pipeline for Enhance-Net that emphasizes the enhancement of traffic sign regions in contrast to the existing training methods that perform enhancement of the whole image. Although these existing methods may lead to better visibility of the overall image, we experimentally show that it offers very limited improvement in traffic sign detection performance. Because an overall enhancement of the image does not necessarily ensure equally enhanced traffic sign regions. To address this issue, our proposed method incorporates the calculation of the loss function only on the traffic sign regions. As the end goal of the approach is to ensure better TSDR, the purpose of the enhancement blocks is not only to ensure the better visibility of the sign regions but also their better detection by the sign detector. In addition, Ledig *et al.* have demonstrated that pixel-wise MSE-loss optimization alone cannot ensure the proper reconstruction of high-frequency contents and, therefore, proposed an MSE-loss in the feature domain of a pretrained deep CNN, termed as content loss [28]. In TSDR, such high-frequency contents can be crucial for the subsequent sign detection and classification module. Therefore, to ensure the best possible performance of the Enhance-Nets, we propose a novel loss function that combines the pixel-domain and feature-domain reconstruction loss as well as the detection loss of this enhanced sign regions. With proper optimization of this loss function and our training pipeline that ensures both the enhancement and successful detection of the sign regions, we are able to achieve a substantial improvement of the current benchmark on the challenging part of the dataset.

In this work, we extend the benchmark result of Kamal *et al.* on the challenge-free data of the CURE-TSD dataset by incorporating the results of our approach on five different CCs with five different levels of severity. To demonstrate the efficacy of our proposed modular solution, we compare its performance with four deep CNN-based end-to-end approaches— Faster R-CNN Inception Resnet V2 [29], R-FCN Resnet 101 [29], RetinaNet-DenseNet [30] and MSA-YOLOv3 [31] that are trained with both challenge free and challenging data. The experimental results reveal that our proposed solution not only improves the existing benchmark result on the challenging data but also outperforms the aforementioned methods by a large margin.

In the following sections of this paper, we provide a detailed description of each part of our proposed method as well as their underlying motivation (Section II), a brief description of the CURE-TSD dataset along with the training procedure of the networks (Section III), a thorough discussion of the experimental results obtained after evaluating our method (Section IV) and finally, some concluding remarks of this work (Section V).

II. METHODOLOGY

A. Motivation

Traffic sign detection is often viewed as an image segmentation and recognition problem. The authors of SegU-Net [22] has reported a benchmark result on CURE-TSD dataset. While their performance on the challenge-free dataset is appealing, they show that incorporation of the challenging dataset significantly deteriorates the performance. This is due to the fact that the adversaries present in those images obscure the color and shape information of small traffic sign regions in particular. Therefore, an overall approach that is robust to the adverse effect of these CCs is of utmost importance for practical application [16].

While many promising CNN-based approaches have been proposed in the literature for traffic sign detection and classification, none of them considers such CCs. As a result, the performance of these methods are severely vulnerable to the quality of the captured image. One straightforward approach can be re-training the networks on the challenging datasets. However, training of these large scale models is not only a tremendous task but also it does not guarantee superior performance in these conditions, which is experimentally demonstrated in the later section. Therefore, we propose a solution to this problem that is completely modular in structure that allows others to incorporate their sign detection model already trained on the challenge-free condition and achieve significantly better performance even in the CCs without requiring any modification to their architectures or trained weights. Although it is possible to work on a general-purpose architecture to handle all kinds of challenges, this would result in suboptimal performance. This is due to the fact that the signal generation model for different kinds of challenges are significantly different in nature. For example, blur is a convolutive noise while rain is additive noise. Therefore, to ensure optimum performance, different networks can be constructed for different challenges. However, this approach requires careful consideration of an extensively large design space that is not feasible. To alleviate this problem, we utilize a challenge classifier module that enables us to utilize the same challenge enhancer architecture for different challenges. We train the same enhancement module for different challenges independently and it is the challenge classifier that guides the input image to the appropriate enhancement module.

While most of the image enhancement works involve the enhancement of the whole image, this is not necessarily required in our case. For successful traffic sign detection, it is only the sign regions that require special attention during the enhancement operation. To validate this proposition, we perform extensive experiments (discussed in the later section) which demonstrate the fact that an overall enhanced image indeed does not introduce any significant performance gain. Therefore, we propose a novel training scheme and a carefully designed loss function that makes the enhancement module focus on the traffic sign regions during training. We also emphasize on the fact that enhancement of these sign regions should be done in such a way that works in favor of the

subsequent sign detection module. For instance, an enhanced image with better visibility can lose some spectral or inherent information during the reconstruction process that can be cru-

cial for its successful detection by the sign-detector. Therefore, to ensure the optimum performance of the detection module on these enhanced images, our proposed method also incorporates the sign detection loss during the training of the enhancement blocks so that an overall minimization of the loss ensures better quality of the sign regions as well as their successful detection.

Overall, our proposed method consists of four separate modules. The first module, Challenge Classifier, detects the type of challenge present in the traffic image. The second module, Enhancement Block, performs the required enhancement for different challenges. Finally, the third and the fourth module consist of a Sign Detection and a Classification block, respectively. For these two modules, we use the architectures proposed in [22] that achieves superior performance on challenge-free part of the CURE-TSD dataset compared to the other considered methods. The outline of our proposed method is as follows (see Fig. 1):

- 1) First, the challenge classifier detects the type of challenge present in the traffic image.
- 2) If the detected challenge type falls within 5 categories (Rain, Snow, Haze, Dirty lens and Lens blur), the image is passed to the corresponding enhancement block.
- 3) The enhanced image is finally passed to SegU-Net and sign classifier for detection and classification of traffic signs.
- 4) If the detected challenge type is 'No challenge', the image bypasses the enhancement blocks and is passed directly to SegU-Net and the sign classifier.

B. Challenge Classifier

The challenge classifier is implemented by utilizing transfer learning on VGG16 [32] CNN architecture pretrained on the ImageNet dataset. The network consists of two stages: feature extraction stage and classification stage. The feature extractor compresses the information present in the image to a lower-dimensional feature space. The subsequent stage uses these features to perform the desired classification.

1) *Feature Extractor*: The feature extractor consists of thirteen convolution blocks, each with 3×3 convolution kernel, a batch normalization layer and a ReLU activation layer. For downsampling purpose, the max-pooling operation is used. Finally, a global average-pooling operation is performed in the final stage to compress the features further.

2) *Classification*: Classification stage has a fully connected layer - that has an output size of 6 (five different types of challenge and one challenge-free condition) with softmax activation. This stage utilizes the extracted features for classification.

C. Enhancement Blocks

We use five enhancement blocks for addressing the five different challenges. Each of these blocks consists of the same CNN-based network architecture that is trained on five

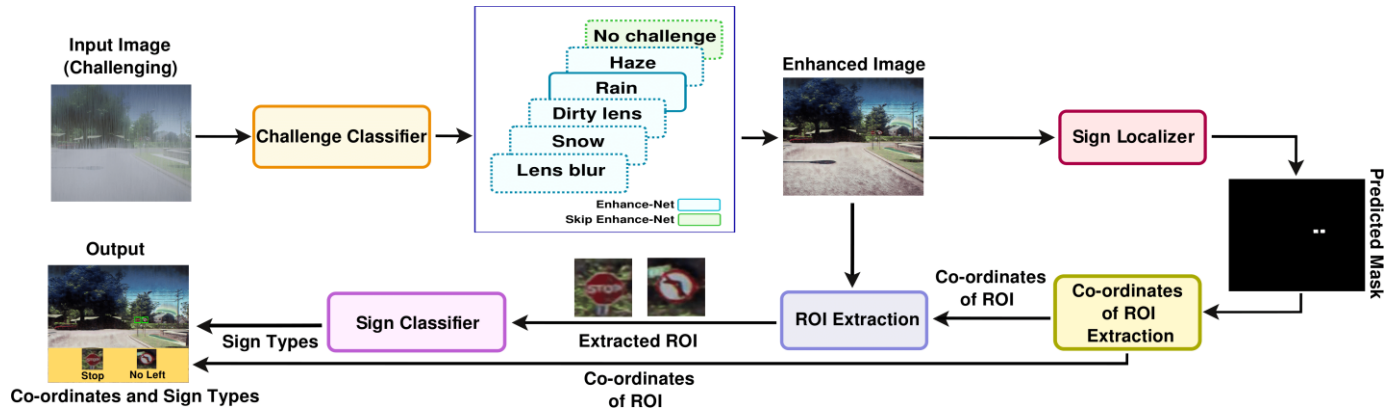


Fig. 1. Outline of the proposed pipeline for robust traffic sign detection and classification.

different types of challenge separately and independently. This provides ease of adding more enhancement blocks to address other challenges as well.

1) *CNN Enhancement Block*: The CNN enhancement block is made of a fully convolutional architecture [33], which is inspired by the architecture of the generator part of DeblurGAN [34], a CNN-based image deblurring method. The network is designed using an encoder (downsampling stage), nine residual blocks (with identity skip connection) followed by a decoder (upsampling stage).

a) *Encoder*: The encoder stage has one convolution block with 7×7 convolution kernels, an instance normalization and a ReLU activation layer followed by two convolutional blocks, each with 3×3 convolution kernels, an instance normalization, and a ReLU activation layer. The number of kernels for convolutional blocks are 64, 128, and 256 and with a stride of 1×1 , 2×2 , and 2×2 , respectively. The encoder encodes the image to the latent feature maps that are further enhanced by the residual blocks.

b) *Residual Block*: Each residual block is composed of two convolution layers each with 3×3 convolution filters, an instance normalization layer and a ReLU activation layer with shortcut skip connection. In this proposed architecture, 9 residual blocks are used. Residual blocks were first used in ResNet [35], where the authors empirically demonstrate that in deeper networks, residual block results in faster convergence and better loss function minimization.

c) *Decoder*: At each step of decoding, the feature maps generated by the residual blocks are up-sampled by using two transposed-convolution blocks, each with 3×3 transposed-convolution with strides of 2×2 , an instance normalization layer, and a ReLU activation layer. Finally, the decoding stage involves a convolution block that has 7×7 sized kernels with Hyperbolic Tangent (Tanh) activation. The whole architecture is presented in Fig. 2.

D. Sign Localizer and Classifier

In this section, we provide a brief architectural overview of the remaining blocks of our modular pipeline, i.e., sign localizer and sign classifier. As mentioned in section II-A, our proposed method is an extension of the benchmark

method [22] that is designed for the challenge-free traffic-sign images of the CURE-TSD dataset. For sign localizer, we use SegU-Net, a fully convolutional, encoder-decoder based pixel-wise segmentation architecture. This particular architecture is inspired from both UNet and SegNet, which combines both of their merits— skip-connection and pooling-unpooling with indices, in a unified architecture [22]. It consists of several convolutional blocks followed by max-pool and up-sample layer in encoder and decoder part, respectively. Each convolutional block consists of 3×3 convolution, batchnorm, and ReLU activation layer. On the other hand, for sign classifier, we use the VGG-16 like architecture proposed in [22] which consists of two convolutional blocks as feature extraction module. Each of these blocks has two 3×3 convolution layers, followed by ReLU activation and max-pooling layer. In order to prevent the model from overfitting, a dropout layer is also introduced after each block with dropout probability of 0.25. The features are finally propagated through two consecutive fully connected layers for sign classification. The architectural details of these networks can be found in [22].

III. TRAINING

In this section, we discuss the details of our training procedure. We also discuss about the dataset briefly. All of our experiments are performed in a hardware environment that includes a Intel Core-i7 7700K, 4.50 GHz CPU and Nvidia Tesla V100 (16 GB Memory) GPU. All of the necessary codes are written in Python and we used Pytorch DL library to implement the neural networks.

A. Dataset

CURE-TSD dataset [21] is by far the largest video dataset that emulates the practical scenarios with case of different challenging and challenge-free conditions for traffic sign detection.

This dataset comprises of videos of twelve different CCs as well as videos of challenge-free condition. The details of the dataset are discussed elaborately in [21], [22]. We split the dataset into train, validation and test partitions according to the rules of IEEE Video and Image Processing (VIP) cup 2017. Table I contains a brief outline of this partitioning

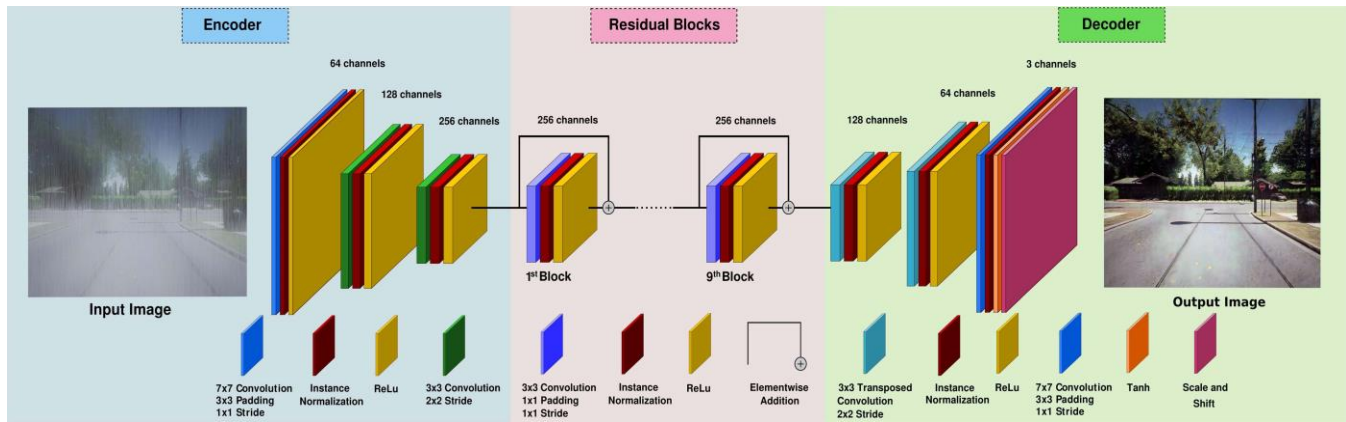


Fig. 2. Architecture of Enhance-Net which consists of three blocks: Encoder block, Residual blocks and Decoder block. The downsampling operation in encoder block is performed by 2×2 strided convolution operation. Element-wise addition operation has been adopted in residual blocks. Transposed convolution operation in the decoder blocks regenerates the enhanced version of the image from the compressed feature space. Finally, the Shift and Scale layer converts the output into pixel domain.

TABLE I
DATA DISTRIBUTION OF CURE-TSD

Challenge	Number of Videos Per Level		
	Train	Validation	Test
Haze	29	5	15
Rain, Snow, Dirty lens, Lens blur	58	10	30

scheme. We perform all of our experiments on the training and validation dataset and evaluate the models performance on the holdout test dataset. In this work, we address and enhance the images of five different CCs (Rain, Snow, Haze, Dirty lens, and Lens blur) that can arise frequently in real- world applications. Some of the challenges such as, Codec Error, Shadow and Exposure are conditions that might arise due to hardware malfunctions and technical faults. These are not expected to occur due to outside environment and therefore can be controlled. Thus we direct our focus on CCs that can arise frequently due to the variation of outside environment. The nature of these challenges associated with their varying level of difficulties are presented in Fig. 3. It is worthwhile to mention that, for Haze, CURE-TSD dataset does not have any simulated environment data, as a result, the volume of data for Haze is halved compared to other CCs.

B. Challenge Classifier

The frames in the CURE-TSD dataset are of 1236×1628 resolution. Using this large size of image in the challenge detection stage is a computationally expensive procedure. Moreover, the impact of weather is present in the image as a global feature. As the main focus of the challenge classifier is to identify the type of challenge, downsampling operation is more likely to strengthen this global feature. Therefore, we resized all the images of selected challenges to 512×512 pixels and using them we train our challenge classifier for

6 classes– 5 for the selected challenges and 1 for the challenge-free type. We used Categorical Cross-Entropy as the loss function and used Adam [36] to optimize our network during training. The initial learning rate was set to 10^{-3} and a learning rate schedule was used to decrease it by a factor of 0.5 if the validation score did not improve for 3 epochs. Finally, using these specifications, we trained our network for 20 epochs.

C. CNN Enhancement Block

For training each CNN enhancement block, first, we extract all the frames from the training video sequences that contain traffic signs of a particular challenge type. Each difficulty level has 29400 number of frames which gives a total of $29400 \times 5 = 147000$ frames, each with a frame size of 1236×1628 pixels. Next, we crop random patches of size 1024×1024 from the frames such that it contains the traffic signs. Due to hardware constraints, we use a batch size of 1 for training. Because of such a small batch size, we perform gradient accumulation. We accumulate the gradients obtained for 5 consecutive batches and update the weights ensuring that the 5 consecutive batches are from 5 different levels of a challenge.

We trained our CNN enhancement block using these images for 35 epochs using an initial learning rate of 10^{-3} . We used Adam [36] as our optimizer function. A learning rate schedule was used to decrease the learning rate by a factor of 0.5 if the validation score did not improve after 3 consecutive epochs.

1) *Loss Function:* As an image enhancement problem in the context of traffic sign detection, the goal of our CNN enhancement block is to enhance the challenging images so that SegU-Net [22], already trained on challenge-free dataset, can achieve elevated performance even in the CCs. However, as we mentioned earlier, this enhancement of the image should be performed in such a way that ensures SegU-Net's successful localization of the traffic signs from these enhanced images. We achieve this by incorporating the sign detection module in training the enhancement blocks. Due to our modular approach, it is important to keep the performance of the

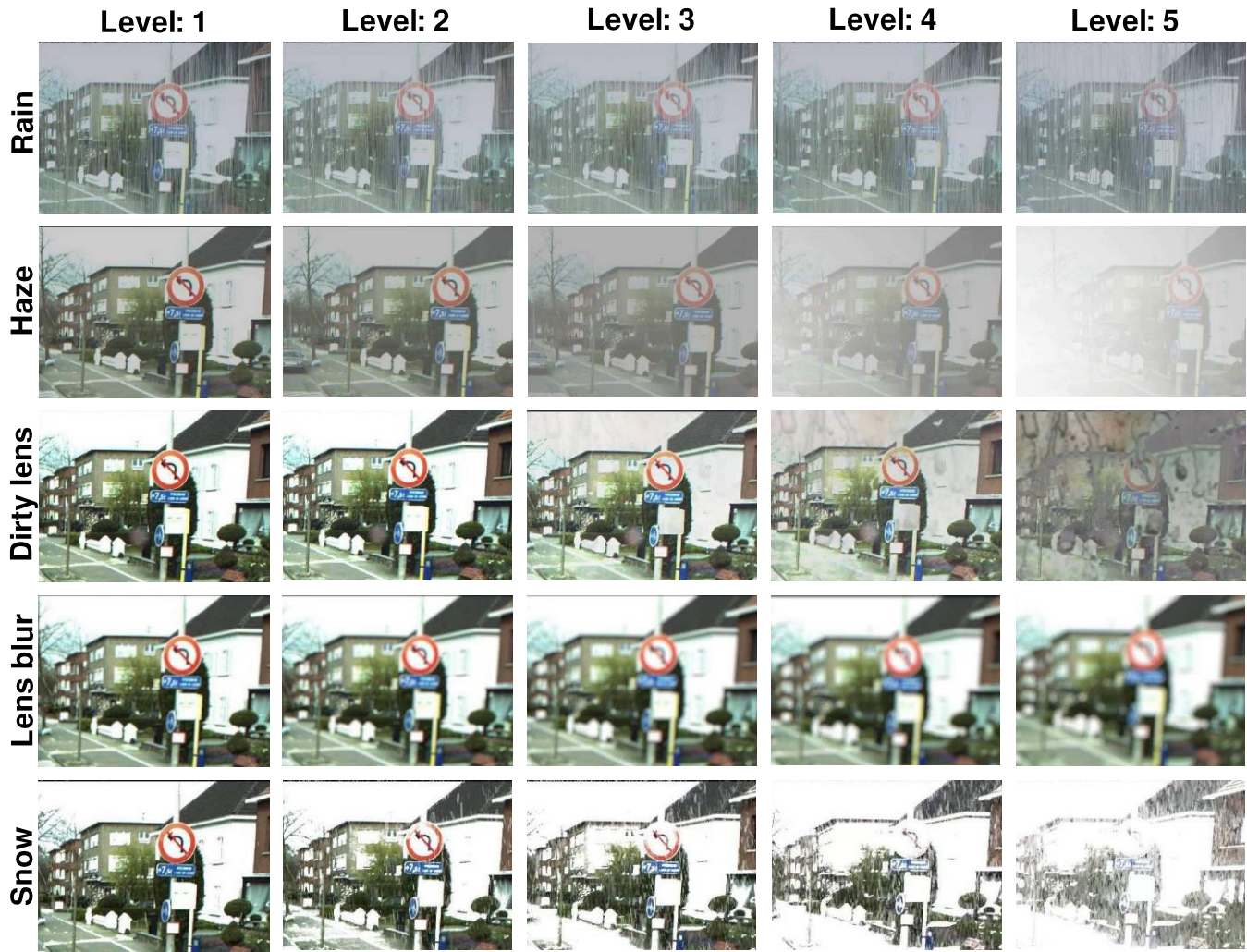


Fig. 3. Selected challenges corresponding to a sample frame with five increasing levels of severity.

detection block in the challenge-free dataset unaffected. Therefore, in all our experiments, we use SegU-Net pre-trained on the challenge-free dataset and we keep this weight unaltered. Overall, we train the image enhancer primarily focusing on the traffic sign regions but at the same time use the detection block to constrain the model to improve traffic sign detection performance. We define the proposed loss function as

$$L_{total} = L_{enhance(sign)} + \lambda_1 \cdot L_{content(sign)} + \lambda_2 \cdot L_{localizer}, \quad (1)$$

where $L_{enhance(sign)}$ is the pixel-level loss across sign regions, $L_{content(sign)}$ is the loss across sign regions in the feature space and $L_{localizer}$ is the loss associated with the detection block, and λ_1, λ_2 are hyperparameters acting as coupling factors that control the combination of the losses altogether.

As overall enhancement of the challenging image may not lead to optimally enhanced traffic sign regions, we introduce a modification to the loss calculation that makes the model emphasize on the traffic sign regions only rather than the whole image. Therefore, unlike existing methods, only the sign regions contribute to the reconstruction loss in our

approach. Our reconstruction loss is calculated with Mean Absolute Error (MAE) between the reconstructed and target traffic sign regions. Our choice of MAE instead of Mean Square Error (MSE) is inspired by [37] where the authors have demonstrated that using MSE loss function yields blurry reconstructed images. This loss function is not suitable for our purpose, since the blurry effect can obscure the fine details present in the low-resolution traffic signs present in the images. On the other hand, Zhao *et al.* [38] have demonstrated that using MAE as the loss function leads to better image reconstruction compared to MSE. Therefore, MAE is a better choice as a reconstruction loss for our approach. Another insight into the problem is that some challenges obscure the high-frequency components that accounts for the sharpness and fine-details of the images and thereby, adversely affect the performance of the sign detection algorithms. This suggests that alongside proper color and shape information, the high-frequency details need to be recovered as well. To meet this demand, we incorporate another loss that ensures better recovery of such details. It is widely accepted that edges, contours and other details present in an image are captured by the shallow layers of a CNN [39]. Therefore, a lower

level intermediate layer of a CNN pretrained on ImageNet dataset can act as the feature extractor that corresponds to the edges, contours and low-level details of the image. We adopt a VGG19 network pretrained on the ImageNet dataset and we extract features of the reconstructed image and target image traffic sign regions from its intermediate layers. Then we minimize the MAE between these reconstructed traffic sign features and target traffic sign features. This minimization ensures that the reconstructed image has recovered the necessary fine details [40]. The overall calculation of loss and training scheme of the CNN enhancement block is illustrated in Figure 4. If R and T are the reconstructed and target traffic sign regions, and H , W , and C represent height, width, channels of the traffic sign regions, respectively, we define $L_{enhance(sign)}$ and $L_{enhance(content)}$ as

$$L_{enhance(sign)} = \frac{1}{H \times W \times C} \sum_{k=0}^C \sum_{j=0}^W \sum_{i=0}^H |R_{ijk} - T_{ijk}|, \quad (2)$$

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

where $\varphi(R)$ and $\varphi(T)$ represent feature extracted by the VGG19 network from the traffic sign region of the reconstructed and target image, respectively.

We incorporate the sign detection module, i.e., SegU-Net, to train the enhancement block by passing the enhanced image to the detector for generating a binary segmentation mask. It is then used along with the ground truth mask to calculate the modified Tversky loss with L1 constrain that is used in [22] to train the detector on challenge-free dataset. This loss is defined as

$$L_{localizer} = 1 - \frac{|P \cap G|}{|P \cap G| + \alpha \cdot |P - G| + \beta \cdot |G - P| + \gamma \sum_{i,j} |P_{ij} - G_{ij}|}, \quad (4)$$

where P and G are the set of predicted mask and ground truth binary labels, respectively. To avoid numerical instability a small safety factor z is added. To weight false positives and false negatives, α and β are used as parameters where $0 \leq \alpha, \beta \leq 1$ with $\alpha + \beta = 1$. Here γ is another hyperparameter that acts as a coupling factor. We use the value of α, β and γ as mentioned in [22].

Furthermore, to get the optimum values of λ_1 and λ_2 , extensive experiments are carried out and based on the results achieved on validation data, a different set of coupling factors values are obtained for different challenges that are given in Table II.

IV. EXPERIMENTAL RESULTS

In this section, we demonstrate the efficacy of our proposed method on the CURE-TSD dataset and also present results of ablation study to signify the importance of each module of our architecture. Furthermore, comparative results are also

TABLE II
OPTIMUM COUPLING FACTOR VALUES FOR EACH CHALLENGE

Challenge	Optimum Coupling Factors	
	λ_1	λ_2
Rain	5	5
Snow	1	10
Dirty lens	10	10
Lens blur	0.5	1
Haze	0.5	5

end-to-end trained approaches to show the superiority of our modular approach.

A. Performance Metrics

We evaluate the performance of our proposed method based

on the following metrics:

- True Positive (TP) = estimated sign region has at least 50% overlap with ground truth sign region with sign type

(3) presented with different high performing deep CNN-based being correctly classified.

- False Positive (FP) = estimated sign region has no overlapping with ground truth sign region or sign type is incorrectly identified.
- True Negative (TN) = no identification of traffic sign in non sign regions.
- False Negative (FN) = no identification of traffic sign in sign regions.

Using these metrics, two scores are defined that summarize the performance of the model:

- Precision (Pr) = $\frac{TP}{TP + FP}$,
- Recall (Rc) = $\frac{TP}{TP + FN}$.

B. Performance of the Challenge Classifier

Performance of the challenge classifier plays a vital role in our proposed method. Therefore, we first compare the performance of different CNN architectures for our challenge classifier module and show them in Table III. It is important to note the total number of frames for ‘No challenge’ is 9000 whereas the number of total traffic signs in No challenge can be greater than 9000, since there can be multiple traffic signs present in a single frame. From Table III, we can see that VGG16 as challenge classifier achieves the highest accuracy—99.98%, with reasonable number of parameters and execution time. Therefore, we select it as our challenge classifier module. One possible phenomenon associated with enhancement of CCs is that, although the performance in CCs improve, performance in challenge-free condition may deteriorate in the case when challenge classifier misclassifies the challenge-free image as a challenging image. In this scenario, a challenge-free image will pass through an enhancement block that can degrade the quality of the image performance deterioration.

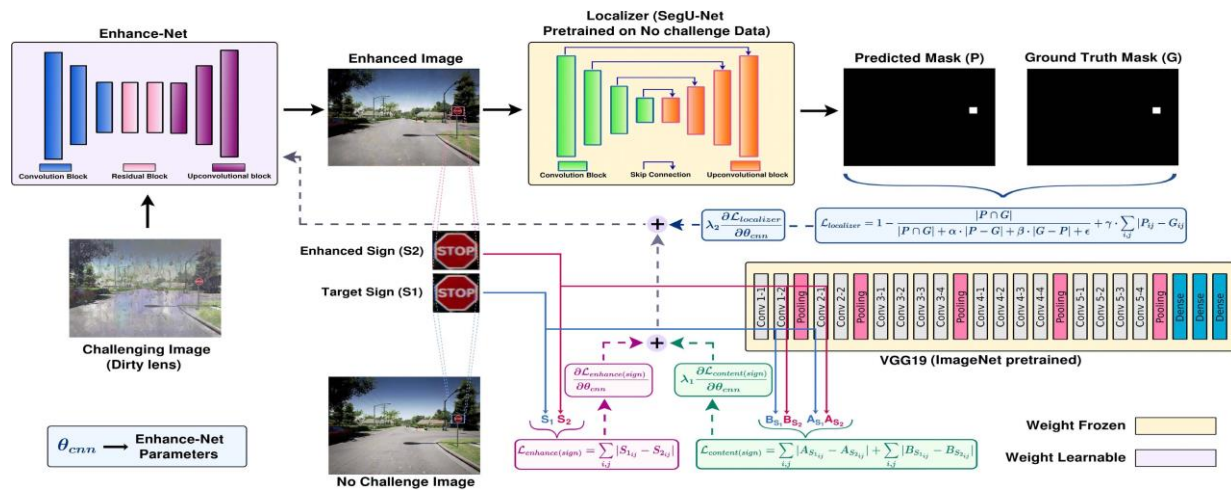


Fig. 4. Training pipeline of the proposed method.

TABLE III
PERFORMANCE COMPARISON AMONG DIFFERENT CNN ARCHITECTURES
AS CHALLENGE CLASSIFIER ON CURE-TSD

Model	Accuracy (%)	No. of Parameters (Million)	Execution Time (ms)
VGG16	99.98	14.73	3.469
Densenet121	99.95	6.96	2.872
Resnet18	99.89	11.18	1.328

100% classification accuracy on the CURE-TSD test dataset in detecting challenge-free condition. This rules out the possibility of performance deterioration in challenge-free condition. Misclassification is only encountered in some level 1 challenges, such as, Snow and Dirty lens, where the challenge classifier misclassifies the conditions as challenge-free condition. From the third and fifth row of the first column of Fig. 3, it is evident that the level 1 Snow and Dirty Lens challenge are indeed visually very similar to challenge-free condition which makes them comparatively difficult to separate from the challenge-free class.

C. Experiments With Different Approaches

Next, we select the SegU-Net pretrained on the challenge-free part of the dataset as our sign detection block. The authors in [22] have used a fixed threshold of 0.5 to generate the binary mask from the detectors raw prediction. However, we varied this threshold from 0.1 to 0.9 with an interval of 0.1 and observed the change in the precision and recall score. Unlike existing CNN-based object detection approaches, where the threshold is used as a minimum object-level confidence score, in our case, this threshold works on pixel-level, since we are performing segmentation. Therefore, there was very small change in precision and recall score by varying this threshold— $\pm 5\%$ from that of the 0.5 baseline threshold. Therefore, in all our subsequent experiments, we use this threshold to generate the binary mask. In order to validate our motivation mentioned

		No challenge	Rain	Snow	Haze	Dirty lens	Lens blur
Actual Class	No challenge	9000 (100 %)	0 (0 %)	0 (0 %)	0 (0 %)	0 (0 %)	0 (0 %)
	Rain	0 (0 %)	45000 (100 %)	0 (0 %)	0 (0 %)	0 (0 %)	0 (0 %)
	Snow	4 (0.01 %)	0 (0 %)	44996 (99.99 %)	0 (0 %)	0 (0 %)	0 (0 %)
	Haze	0 (0 %)	0 (0 %)	0 (0 %)	22500 (100 %)	0 (0 %)	0 (0 %)
	Dirty lens	27 (0.06 %)	0 (0 %)	0 (0 %)	0 (0 %)	44973 (99.94 %)	0 (0 %)
	Lens blur	0 (0 %)	0 (0 %)	0 (0 %)	0 (0 %)	0 (0 %)	45000 (100 %)
		No challenge	Rain	Snow	Haze	Dirty lens	Lens blur

Fig. 5. Performance of the challenge classifier.

in Section II-A and demonstrate the efficacy of the Enhance-Net, we carried out several experiments that are outlined as below:

- 1) SegU-Net retrained with challenge-free as well as challenging images by using the $L_{localizer}$ loss. No prior enhancement was done on the input image.
- 2) Inputs of the SegU-Net are the traffic images enhanced by Enhance-Net. The enhancement block was trained using $L_{enhance(overall)}$ loss calculated across the whole image.
- 3) Inputs of the SegU-Net are the traffic images enhanced by Enhance-Net. The enhancement block was trained using $L_{enhance(sign)}$ loss calculated over the sign region only.
- 4) Inputs of the SegU-Net are the traffic images enhanced by Enhance-Net. The enhancement block was trained using whole image region-based $L_{enhance(overall)}$ loss coupled with the $L_{localizer}$ loss.

TABLE IV

PERFORMANCE ON THE TEST SET OF CURE-TSD DATASET WITH PROGRESSIVE ASSOCIATION TO THE PROPOSED FRAMEWORK

Model	Data	Loss	Precision	Recall
SegU-Net	challenge-free	$L_{localizer}$	83.55	34.81
SegU-Net	challenge-free + challenging	$L_{localizer}$	80.84	39.29
SegU-Net + separately trained Enhance-Net	challenging	$L_{enhance(overall)}$	93.19	60.69
SegU-Net + separately trained Enhance-Net	challenging	$L_{enhance(sign)}$	92.98	64.70
SegU-Net + jointly trained Enhance-Net	challenging	$L_{enhance(overall)}$ + $L_{localizer}$	93.01	66.49
SegU-Net + jointly trained Enhance-Net	challenging	$L_{enhance(sign)}$ + $L_{localizer}$	91.76	68.71
SegU-Net + jointly trained Enhance-Net (with unity coupling factors)	challenging	$L_{enhance(sign)}$ + $L_{content(sign)}$ + $L_{localizer}$	91.68	69.89
SegU-Net + jointly trained Enhance-Net (with optimum coupling factors)	challenging	$L_{enhance(sign)}$ + $\lambda_1 \cdot L_{content(sign)}$ + $\lambda_2 \cdot L_{localizer}$	91.13	70.71

TABLE V

COMPARISON OF STRUCTURAL SIMILARITY MEASURE (SSIM) VALUES BETWEEN OVERALL AND ONLY SIGN REGION ENHANCEMENT APPROACH ON CURE-TSD

Challenge Levels	SSIM _{OE} (Overall Enhanced)	SSIM _{SRE} (Sign Region Enhanced)
1	0.8594	0.8750
2	0.8325	0.8537
3	0.7994	0.8258
4	0.7656	0.7972
5	0.7097	0.7465
Average	0.7933	0.8196

- 5) Inputs of the SegU-Net are the traffic images enhanced by Enhance-Net. The enhancement block was trained using sign region-based $L_{enhance(sign)}$ loss coupled with the $L_{localizer}$ loss.
- 6) Inputs of the SegU-Net are the traffic images enhanced by Enhance-Net. The enhancement block was trained using sign region-based $L_{enhance(sign)}$ loss coupled with the $L_{localizer}$ loss and $L_{content(sign)}$ loss with all coupling factors value 1.
- 7) Inputs of the SegU-Net are the traffic images enhanced by Enhance-Net. The enhancement block was trained using sign region-based $L_{enhance(sign)}$ loss coupled with the $L_{localizer}$ loss and $L_{content(sign)}$ loss with the optimum coupling factors.

The experimental results are summarized in Table IV. From the first result in this table, we observe that training the sign detection module with abundant training data is not effective in alleviating performance degradation caused by the challenging conditions. This justifies our motivation for using the enhancement module. From the other results, we can infer that enhancement of traffic sign regions should be the main focus of the enhancement module for better traffic sign detection performance. To further confirm this, we present the Structural Similarity Measure (SSIM) between the enhanced and challenge-free traffic sign regions for both overall and only sign region enhancement approach. Table V shows the challenges severity level-wise SSIM values, averaged over all the challenge types. From this table, we see that our proposed method achieves higher SSIM score— 0.8196 across the traffic sign regions compared to the 0.7933 SSIM score from overall enhancement approach. It is also evident that there is a degradation of SSIM values associated with the increasing level of difficulty for both cases. However, our proposed approach suffers from less performance deterioration than the conventional approach. We also show some outputs of our proposed method and the outputs of the overall enhanced method side by side in Fig. 6. From this figure, we can see that although the overall image achieves better visibility in the case of total enhancement approach, the sign region still suffers from harsh visibility. Whereas, in our approach, the sign region achieves almost similar visibility compared to the challenge-free condition. Finally, the incorporation of $L_{localizer}$ brings forth further performance gain in the overall approach because



Fig. 6. Instances of prediction by Enhance-Net on a selected challenging image from CURE-TSD dataset. The first column represents a challenge-free image from CURE-TSD dataset with its zoomed-in sign region. A challenging (Rain) input image and its zoomed-in version of the sign region are in the second column. The third column shows the output of Enhance-Net trained with overall enhancement-based approach and its corresponding zoomed-in version of the sign region. The last column shows the output of Enhance-Net for the same input, but this time trained with our proposed sign-region focused enhancement approach and its corresponding zoomed-in version of the sign region.

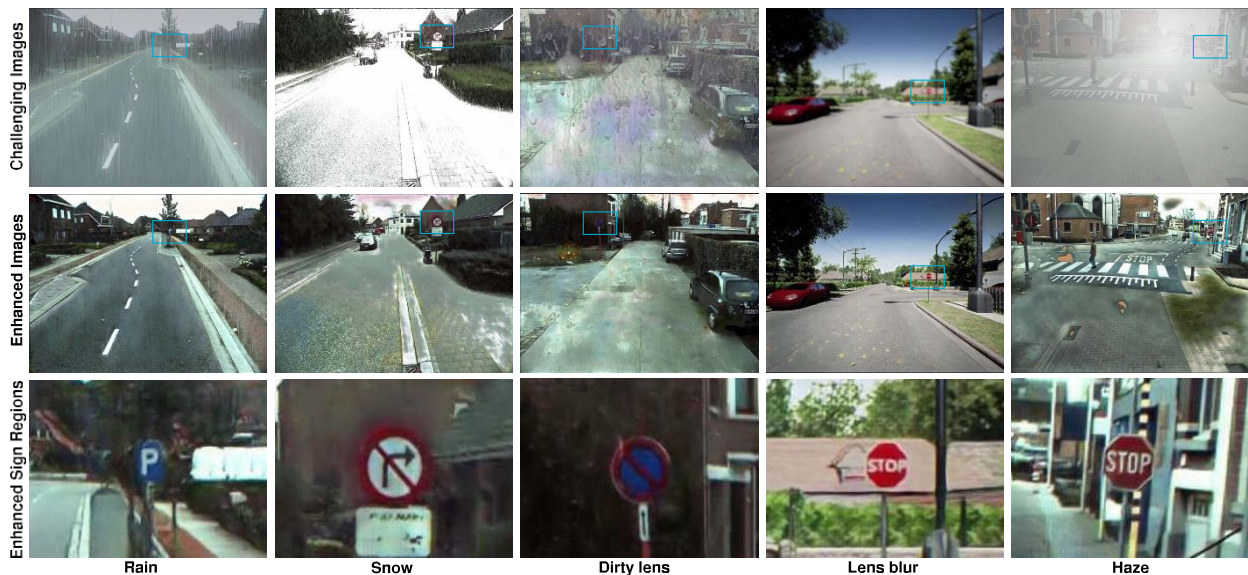


Fig. 7. Typical outputs of our proposed Enhance-Net on selected images of different challenge types and levels from CURE-TSD dataset. The first, second, and third row shows, the challenging images, enhanced images and zoomed-in sign regions of enhanced images, respectively. Instances of challenging images (Rain, Snow, Dirty lens, Lens blur and Haze) are displayed in column-wise order.

it constrains the training of Enhance-Net to learn to enhance the sign regions subject to their accurate detection by the subsequent sign detection module. It is noteworthy to mention that, overall enhancement-based approaches achieve higher precision values which can be observed from the third and fifth row of Table IV. It is due to the fact that, overall enhancement based approach ensures better image quality for the background region, which can assist the sign detector module in the false positive reduction. However, if we consider both precision and recall, our approach achieves better

overall performance. Fig. 7 contains five example outputs of Enhance-Net along with their challenging counterparts on the test dataset by considering different challenge types and levels. It is important to mention that, depending on the challenge type and severity, the quality of the enhanced test image varies to some extent as apparent from the figure. However, as our algorithmic emphasis is on the enhancement of the sign region, our approach is able to recover the necessary color and shape information as can be seen from Figure 7.

TABLE VI

RESULTS OF THE PROPOSED METHOD ON THE TEST SET OF CURE-TSD DATASET FOR DIFFERENT CHALLENGING CONDITIONS

Challenge	Level	Sign Classifier Accuracy (%)	Performance Metrics	
			Precision (%)	Recall (%)
No challenge	N/A	99.02	80.21	94.60
Rain	1	98.15	77.43	91.50
	2	97.72	74.41	90.59
	3	97.38	72.26	90.24
	4	96.88	69.99	89.54
	5	96.03	63.76	87.84
Snow	1	98.95	81.30	93.81
	2	98.24	77.89	92.21
	3	98.57	72.66	93.35
	4	96.04	62.01	91.22
	5	94.94	44.74	89.80
Dirty lens	1	98.55	81.62	93.97
	2	98.56	81.74	94.15
	3	98.33	80.61	93.87
	4	97.94	79.50	93.38
	5	97.33	75.70	92.58
Lens blur	1	98.62	80.71	93.64
	2	98.51	79.85	93.03
	3	98.22	78.58	93.04
	4	97.57	76.97	92.66
	5	97.31	73.82	92.64
Haze	1	98.31	58.96	86.81
	2	98.76	58.70	87.02
	3	98.85	57.82	86.94
	4	98.38	56.12	87.07
	5	97.74	50.66	87.30

D. Performance on CURE-TSD

We evaluate the performance of our proposed method on the selected sets of the challenging part of CURE-TSD. We report the performance of our approach on the five selected challenge types along with the challenge-free type from the test set of CURE-TSD dataset in Table VI. Precision and recall for ‘No challenge’ as shown in Table VI are the same as in [22] because of three reasons. Firstly, we use the same sign detection and classification module of [22] pretrained on the challenge-free part of the dataset. Secondly, our proposed challenge classifier has a 100% accurate ‘No challenge’ detection performance (Fig. 5). Thirdly, we keep the pretrained model of [22] frozen while training Enhance-Net. Therefore, the number of TPs, FPs, FNs and TNs for ‘No challenge’ in our approach are the same as in [22]. Results on different challenge types reveal that there is still some performance degradation associated with the higher level of CCs, specially for level 4 and 5. These higher levels of challenge create an extremely harsh visibility condition as can be observed from

Fig. 3. For instance, from the fifth column of the fifth row of Fig. 3, i.e., ‘Snow’ challenge of level 5, it is evident that the sign region has almost completely lost its shape and color information that adversely affects the performance of the sign detector. As a result, this particular case has the least recall of 44.74% among all other cases. Also, for Haze, the overall performance metrics are relatively lower compared to the other CCs. This observation may be attributed to insufficient training data for this case. As stated in section III-A, for this specific condition, the amount of training data was half as compared to other CCs. Furthermore, we investigate the performance of sign classifier for each challenge levels as presented in Table VI. The sign classifier achieves the best accuracy of 99.02% on ‘No challenge’ case. For all other challenge types, it is apparent that the performance of the classifier deteriorates with increasing challenge level. This performance degradation can be attributed to the sign regions shape and color information loss associated with the higher level of challenges.

E. Performance Comparison With Other Methods

In this section, we present a comparative performance analysis of our proposed method with several other deep CNN-based TSDR methods on CURE-TSD dataset. In addition to the Faster R-CNN (Inception Resnet V2 backbone) and R-FCN (Resnet 101 backbone) [29] which have been used by Kamal *et al.* for comparison, we use two recently proposed deep CNN-based TSDR methods namely–Pyramidal Convolutional Network (Retinanet-Densenet) [30] and MSA-YoloV3 [31]. We train these four networks on CURE-TSD dataset and evaluate their performance on the same test set for fair comparison.

Table VII contains the average performance indexes achieved by the aforementioned four methods as well as our proposed method. With an average 91.13% precision and 70.71% recall, our method outperforms all other methods by a large margin. This suggests the superiority of our modular approach over other methods that involve end-to-end training of the networks. Table VII also shows comparison of our approach with other methods in terms of execution time along with parameter count. Although MSA-YOLOv3 achieves the highest 15.7 FPS, it suffers from the lowest recall score of 33.41%. On the other hand, our proposed method can process approximately 3 FPS with a parameter count of approximately 33 million. Compared to other methods, our approach stands in an intermediate position in terms of execution time and parameter count.

F. Performance Degradation With Increasing Challenge Level

In this section, we investigate the extent of performance degradation with respect to increasing challenge intensities for all the considered TSDR methods. Fig. 8 illustrates a comparative analysis of the performance degradation associated with the increasing challenge levels for different approaches. It is evident that precision score suffers from less deterioration compared to recall score since increasing challenge level makes the sign regions as well as the corresponding

TABLE VII

PERFORMANCE, MODEL PARAMETER COUNT AND INFERENCE SPEED COMPARISON AMONG DIFFERENT METHODS ON CURE-TSD DATASET

Method	Precision (%)	Recall (%)	No. of Parameters (Million)	Frame Per Second (FPS)
R-FCN [29]	53.31	44.48	65.10	0.24
Faster-RCNN [29]	58.17	46.03	59.49	0.22
RetinaNet-DenseNet [30]	64.41	45.34	17.07	5.19
MSA-YOLOv3 [31]	70.31	33.41	62.61	15.7
Proposed Method	91.13	70.71	33.29	2.79

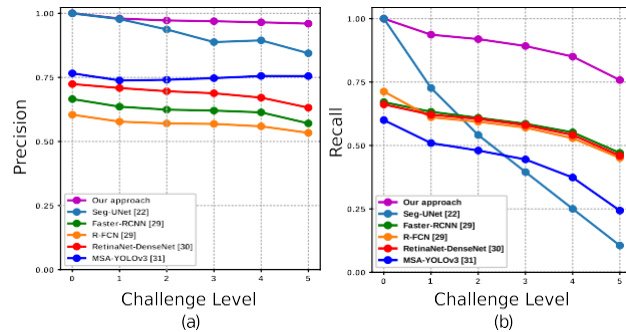


Fig. 8. Performance degradation of different methods on CURE-TSD dataset with the increasing difficulty of the challenges.

background regions more obscured which reduces the overall number of detected signs. Among all the approaches, SegU-Net which is trained on the challenge-free dataset only, suffers from the highest performance degradation due to the lack of challenging train dataset. If we compare the performance of the other deeper CNN models that are trained on both challenging and challenge-free dataset, we can see that they suffer from less performance degradation with increasing challenge level. This suggests that abundant training data can help to reduce performance deterioration with increasing challenge level. However, without any special attention for the challenge parts, their methods fail to achieve the best performance. This justifies our initial hypothesis that end-to-end training with abundant data is not sufficient to achieve the desired result for this task. On the other hand, our modular approach integrated with separate enhancement blocks for each challenge type achieves the least performance (both precision and recall) degradation while maintaining the highest performance for the challenge-free type as well. This experimentally proves the superiority of our enhancement-based modular approach over the existing high-performing methods for robust traffic sign detection under severe CCs.

V. CONCLUSION

In this paper, we have presented a deep CNN-based modular and a robust framework for TSDR under various CCs. We have highlighted the performance degradation of the existing TSDR algorithms due to the presence of different CCs and proposed a deep CNN-based approach that effectively alleviates the problem. A VGG16 architecture-based challenge classifier, that successfully detects and classifies the challenge, directs the image to the appropriate Enhance-Net which recovers

the features that are useful for the successful detection of the traffic sign regions. Unlike the existing whole image enhancement-based methods, the Enhance-Nets are trained by our proposed novel loss function and training pipeline that incorporate traffic sign region focused MAE in both pixel and feature domain with the sign detection loss as a constraint. This effectively ensures the enhancement of the sign regions subject to their accurate detection. We have also experimentally showed that traffic sign regions are more important for enhancement, in order to obtain higher detection performance. Finally, we evaluate the efficacy of the modular structure of our approach by comparing its performance with different end-to-end trained deep CNN-based object detection networks where our approach outperforms all of them. Due to our modular approach, each module of our framework can be designed independently. This opens up a vast research scope in this field. In our future work, we wish to explore different architectural and compression techniques to design a nearly optimum network for each module so that all of the CCs present in the CURE-TSD dataset are best addressed both in terms of performance and inference speed.

ACKNOWLEDGMENT

The authors gratefully acknowledge Intelligence Machines Limited (IML) for giving the required computational resource to carry out this research. Also, the authors are grateful to NVIDIA corporation for granting accelerated computational resources.

REFERENCES

- [1] 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.
- [2] A. Gudigar, S. Chokkadi, and U. Raghavendra, "A review on automatic detection and recognition of traffic sign," *Multimedia Tools Appl.*, vol. 75, no. 1, pp. 333–364, Jan. 2016.
- [3] S. Xu, "Robust traffic sign shape recognition using geometric matching," *IET Intell. Transp. Syst.*, vol. 3, no. 1, pp. 10–18, Mar. 2009.
- [4] 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.
- [5] I. M. Creusen, R. G. J. Wijnhoven, E. Herbschleb, and P. H. N. de With, "Color exploitation in hog-based traffic sign detection," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 2669–2672.
- [6] D. Temel, M.-H. Chen, and G. AlRegib, "Traffic sign detection under challenging conditions: A deeper look into performance variations and spectral characteristics," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3663–3673, Sep. 2020.

- [7] J. Canny, "A computational approach to edge detection," in *Readings in Computer Vision*. Amsterdam, The Netherlands: Elsevier, 1987, pp. 184–203.
- [8] X. Baró, S. Escalera, J. Vitrià, O. Pujol, and P. Radeva, "Traffic sign recognition using evolutionary AdaBoost detection and forest-ECOC classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 113–126, Mar. 2009.
- [9] P. G. Jiménez, S. M. Bascón, H. G. Moreno, S. L. Arroyo, and F. L. Ferreras, "Traffic sign shape classification and localization based on the normalized FFT of the signature of blobs and 2D homographies," *Signal Process.*, vol. 88, no. 12, pp. 2943–2955, Dec. 2008.
- [10] F. Zaklouta and B. Stanculescu, "Real-time traffic sign recognition in three stages," *Robot. Auto. Syst.*, vol. 62, no. 1, pp. 16–24, Jan. 2014.
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [12] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*. [Online]. Available: <http://arxiv.org/abs/1412.7062>
- [13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [14] H. Luo, Y. Yang, B. Tong, F. Wu, and B. Fan, "Traffic sign recognition using a multi-task convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1100–1111, Apr. 2018.
- [15] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2022–2031, Jul. 2016.
- [16] J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient CNNs in the wild," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 975–984, Mar. 2019.
- [17] 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.
- [18] Y. Yuan, Z. Xiong, and Q. Wang, "VSSA-NET: Vertical spatial sequence attention network for traffic sign detection," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3423–3434, Jul. 2019.
- [19] 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. (IJCNN)*, Aug. 2013, pp. 1–8.
- [20] 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.
- [21] D. Temel, T. Alshawi, M.-H. Chen, and G. AlRegib, "CURE-TSD: Challenging unreal and real environments for traffic sign detection," *IEEE Dataport*, Oct. 2019, doi: [10.21227/en9z-mq69](https://doi.org/10.21227/en9z-mq69).
- [22] U. Kamal, T. I. Tonmoy, S. Das, and M. 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.
- [23] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Single image rain streak decomposition using layer priors," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3874–3885, Aug. 2017.
- [24] Y. Luo, J. Zhu, J. Ling, and E. Wu, "Fast removal of rain streaks from a single image via a shape prior," *IEEE Access*, vol. 6, pp. 60069–60078, 2018.
- [25] J. Li, G. Li, and H. Fan, "Image dehazing using residual-based deep CNN," *IEEE Access*, vol. 6, pp. 26831–26842, 2018.
- [26] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2658–2670, Sep. 2015.
- [27] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2838–2847.
- [28] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.
- [29] Á. Arcos-García, J. A. Álvarez-García, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems," *Neurocomputing*, vol. 316, pp. 332–344, Nov. 2018.
- [30] Z. Liang, J. Shao, D. Zhang, and L. Gao, "Traffic sign detection and recognition based on pyramidal convolutional networks," *Neural Comput. Appl.*, vol. 32, no. 11, pp. 6533–6543, Jun. 2020.
- [31] H. Zhang et al., "Real-time detection method for small traffic signs based on Yolov3," *IEEE Access*, vol. 8, pp. 64145–64156, 2020.
- [32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [33] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.
- [34] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8183–8192.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [37] M. Mathieu, C. Couprie, and Y. Lecun, "Deep multi-scale video prediction beyond mean square error," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, May 2016.
- [38] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for neural networks for image processing," 2015, *arXiv:1511.08861*. [Online]. Available: <http://arxiv.org/abs/1511.08861>
- [39] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8689, Nov. 2013, pp. 818–833.
- [40] 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.