Traffic-Sign Detection and Classification Under Challenging Conditions: A Deep Neural Network Based Approach

Uday Kamal*, Sowmitra Das†, Abid Abrar‡, and, Md. Kamrul Hasan§
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology, Dhaka-1205, Bangladesh
E-mail: *udday2014@gmail.com, †sowmitra.josephite@gmail.com, ‡abidabrar.1bd@gmail.com, §khasan@eee.buet.ac.bd

Abstract—**Traffic sign detection and classification is of paramount importance for the future of autonomous vehicle technology. Many promising methods have been proposed in the literature to perform this task. However, almost all of them work well only with clear and noise-free images, and, do not provide satisfactory results in the presence of challenging image-capture conditions (such as noise, variation of illumination, motion artifacts, etc.) In this paper, we propose a Convolutional Neural Network (CNN)-based approach for detecting and classifying traffic signs which is robust against such challenging environmental conditions. In the proposed model, our approach is to consecutively (i) detect the type of challenge present in the image, (ii) selectively preprocess the image to enhance the sign-features (iii) localize traffic sign proposals from the preprocessed image using a network specifically trained for the detected challenge-type, and, finally, (iv) classify the sign proppsals extracted from the image. These tasks are carried out by implementing CNNs of varying architecture and depth. The challenge-detector and classifier are implemented by using a VGG-16-type architecture, whereas the localizer is similar to the U-Net architecture. The proposed model has been evaluated on the Traffic-Sign Recognition Dataset provided for the IEEE VIP Cup 2017. Experiments demonstrate that, it is robust enough to attain satisfactory detection and classification accuracy even in the presence of the given challenges, with an overall precision and recall of 62% and 42% respectivley.**

I. Introduction

Traffic-sign detection and classification is an interesting topic in computer-vision and it is especially important in the context of autonomous vehicle technology. Robust and real-time traffic-sign detection algorithms have to be employed if self-driving cars are to become commonplace in the roads of the future.

Quite a number of methods have been proposed to this end in the literature. However, the current state of the art has only been trained tested on images which were captured in natural lighting conditions and in the absence of noise [8]. A comprehensive study of detecting traffic-signs in the presence of challenging conditions that a car may face on the road is yet to be undertaken. Besdies, the challenges faced during real-time image-capture on the roads may be multifarious. Even the reported models, which are able to handle illumination-variation efficiently, have not been tested against other types of challenges, such as, blur, decolorization, etc. as they are not likely to produce satisfactory results in these cases [10]. Thus, designing a unified model that can handle all of these problems is quite challenging.

In order to address these problems, we present a novel method of detecting traffic-signs completely based on deep Convolutional Neural Networks (CNNs). The motivation for our model is twofold. First of all, due to the variety of the types and level of challenges present in the dataset, any algorithmic or statistical approach to denoising, segmentation and classification proves to be quite computationally intensive. As a result, this approach would be unsuitable for real-time applications. Thus, one of the reasons to consider a neural-network based detection and classification scheme is its computational efficiency during implementation. Secondly, CNNs have achieved phenomenal accuracy and efficiency in image-classification in recent years. An analysis of the dataset reveals that, there are only a finite number of challenge-types and sign-types to be considered, and, the type of noise in each image is distinct and unique. Therefore, a neural-network based classification of the noise is likely to yield good results.

The following sections of this paper explain the outline of the model, detailed architecture of the networks employed in each stage, implementation and training procedure of the networks, and, the results obtained after testing the model against the testing-data provided with the dataset.

II. Materials and Methods

A. Analysis of Dataset

A video dataset was provided for the IEEE VIP Cup 2017, in order to test and train the traffic-sign detection model. A brief summary of the dataset is given Table I.

There are a total of 98 unique video sequences in the dataset, half of which are real and the other half synthesized. 14 types of traffic signs are present in the dataset, and, 12 types of challenges have been simulated in the video sequences. The challenge types are categorized and stated below (with their index numbers):

- No Challenge (00)
- Color Type: Decolorization (01)
- Blur Type: Lens blur (02), Gaussian blur (07)
- Illumination Type: Darkening (04), Shadow (10), Haze (12), Exposure (06)

TABLE I
SUMMARY OF IEEE VIP CUP VIDEO DATASET

Unique Video Sequences	49 (Real) + 49 (Synthesized)
Frames per Video	300
Frame Size	1236×1628
Sign Types	14
Challenge Types	12 + 1 (No Challenge)
No. of Levels per Challenge	5
Total Number of Videos	5,733
Total Number of Images	1,719,900
Train-Test Split	70%-30%

- Weather Type: Rain (09), Snow (11)
- Other: Codec Error (03), Dirty Lens (05), Noise (08)

5 levels of severeness have been simulated for each challenge type. Thus, with 300 frames per video, the total number of images for each specific challenge type and level is $98 \times 300 = 29,400$.

B. Model Proposal

The outline of our proposed model is as follows (see Fig. 1):

1) First, we detect the type of challenge present in the image by using a CNN, similar to the VGG-16 architecture [3].
2) Then, we preprocess the image to enhance traffic-sign features depending on the type of challenge detected. We adopt a variant of Histogram-Equalization for preprocessing images affected with challenges of Illumination Type and Decolorization. We also use a CNN of ResNet-type architecture to denoise images affected with Rain.
3) Next, we localize traffic-sign proposals from the preprocessed image. There are 3 localizing networks in this stage - one for Lens Blur and Gaussian Blur, one for Noise and a Common Localizer for all of the other types of challenges. The task of localizing sign-proposals is performed by using a deep CNN of U-Net Architecture.
4) Finally, we extract the sign-proposals predicted by the localizer from the preprocessed image, and, pass them to yet another CNN for classification. The architecture of this CNN is similar to that used in detecting the challenge-type in the first step, but, with more depth, so that it can attain higher classification accuracy.

C. Architecture

1) Challenge Detection: The description of the Challenge-Detection CNN is given in Table II. It consists of four blocks, each with 2 convolution layers, a ReLU Activation Layer and a Max-Pooling layer. The output after these 4 blocks is then flattened, and, a fully connected layer is used to map this flattened feature vector to 12 classes. These 12 classes represent the 12 types of challenge that are to be detected.

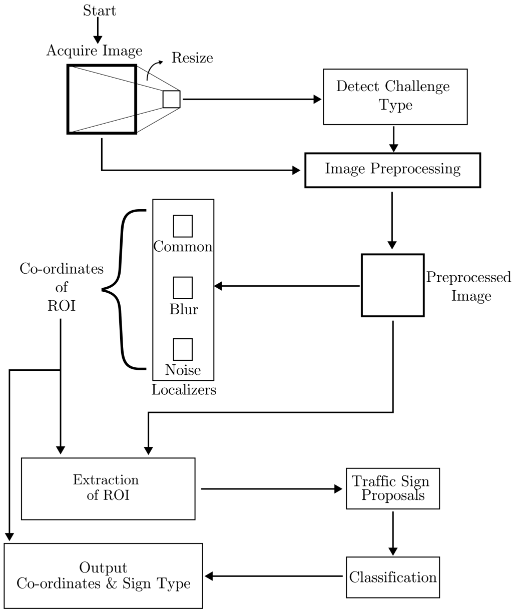

Fig. 1. Outline of the model proposed for traffic-sign detection and classification.

Since the type of challenge present in the image is a global and dominant feature of the image, we do not need a lot of parameters to classify this feature. Therefore, a CNN with low depth, such as this, can be used to detect the type of challenge.

TABLE II
CNN PARAMETERS FOR CHALLENGE DETECTION

Layer No.	Layer Type	No. of Parameters	Kernel Size
	Input	$3 \times 309 \times 407$	
1.1	Convolution	$5 \times 307 \times 405$	3×3
1.2	Convolution	$2 \times 305 \times 403$	3×3
1.3	Max Pooling	$2 \times 76 \times 100$	4×4
2.1	Convolution	$2 \times 74 \times 98$	3×3
2.2	Convolution	$2 \times 72 \times 96$	3×3
2.3	Max Pooling	$2 \times 36 \times 48$	2×2
3.1	Convolution	$2 \times 34 \times 46$	3×3
3.2	Convolution	$2 \times 32 \times 44$	3×3
3.3	Max Pooling	$2 \times 16 \times 22$	2×2
4.1	Convolution	$2 \times 14 \times 20$	3×3
4.2	Convolution	$2 \times 12 \times 18$	3×3
4.3	Max Pooling	$2 \times 6 \times 9$	2×2
5	Flatten	$1 \times 1 \times 108$	
6	Fully Connected (ReLU)	$1 \times 1 \times 500$	
7	Fully Connected (Softmax)	$1 \times 1 \times 12$	

2) Selective Preprocessing: Before passing the image on to the corresponding localizer for detecting traffic-signs, it is first preprocessed. The preprocessing method that we choose to use is a variant of Histogram-Equalization, called, Contrast Limited Adaptive Histogram Equalization (CLAHE). This version of histogram equalization works on a local neighborhood of

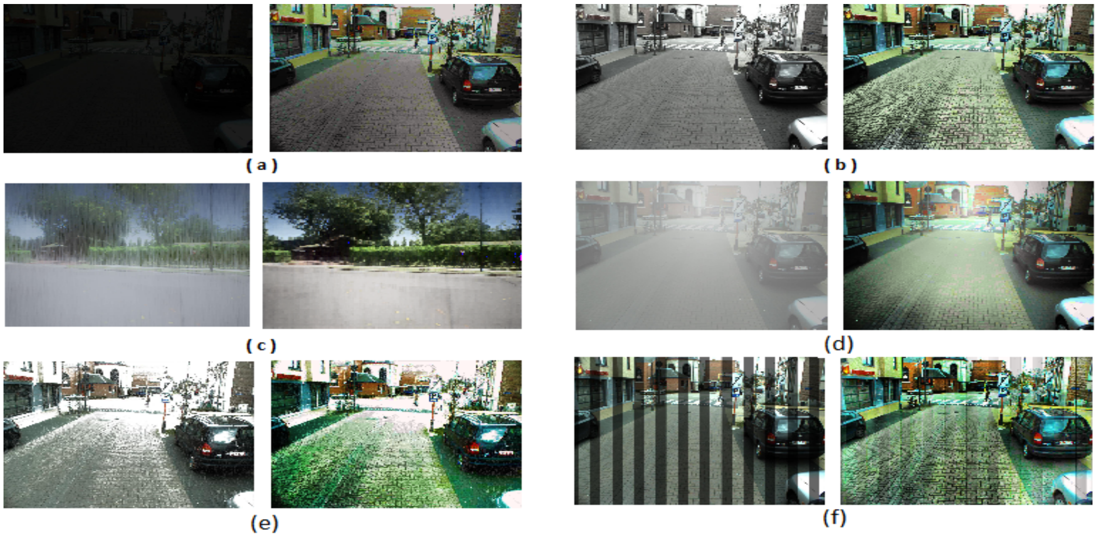

Fig. 2. Results of pre-processing images affected with different challenge-types. (a) Darkening, (b) Decolorization, (c) Rain, (d) Haze, (e) Snow, (f) Shadow.

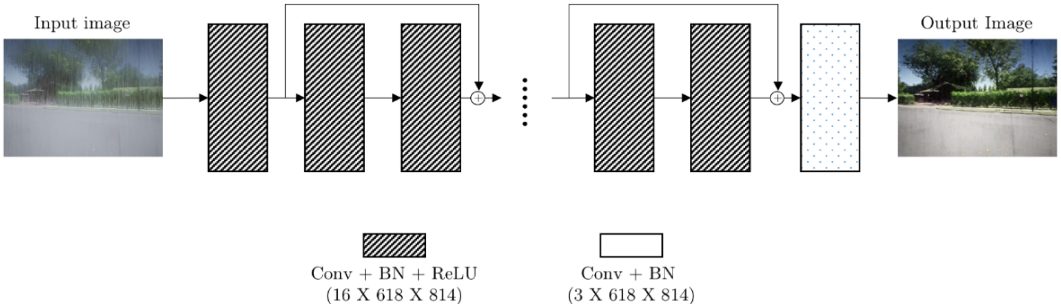

Fig. 3. CNN architecture for removing rain.

the image and limits contrast to prevent enhancement of noise. To do this, we first convert the image from RGB to HSV (Hue, Saturation, Value) space, and then, equalize the histogram of the proper HSV channel. The equalizing operations that are performed for the different challenge types are given in Table III. The table states the number of applications of CLAHE to the Saturation or Value channel, and, the Contrast Clip-Limit used. The result of our preprocessing method, performed on level 4 noisy images, is shown in Fig. 2.

TABLE III
NO. OF APPLICATIONS OF CLAHE (AND CONTRAST LIMITS) TO CORRESPONDING HSV CHANNELS FOR DIFFERENT CHALLENGE TYPES

Challenge Type	Saturation Channel	Value Channel
Decolorization	2 (10)	1 (10)
Darkening	1 (10)	1 (50)
Exposure	1 (5)	1 (5)
Shadow	1 (5)	1 (10)
Snow	1 (10)	1 (5)
Haze	1 (5)	1 (5)

For images affected with Rain, we use a CNN of ResNet-type architecture to remove the rain [7]. The architecture of this network is shown in Fig. 3. It consists of consecutive Convolution blocks, each with a Convolution layer, a Batch Normalization layer and a ReLU layer. Skip connections are used which help to directly propagate lossless information throughout the entire network. This is useful for estimating the de-rained image. Finally the Residual image generated by the model is added to the original image to output the de-rained image. Here, all pooling operations are removed to preserve spatial information.

3) *Traffic-Sign Localization:* This stage of the model is implemented by using a Fully Convolutional Neural Network inspired by the U-Net Architecture [4]. The detailed network architecture is illustrated in Fig. 4. The network consists of 2 stages: a downsampling stage and an upsampling stage, which are connected by using a Residual Learning strategy.

a) Down-sampling: This stage consists of several convolution blocks, each with 2 convolution layers, a Batch Normalization layer and a ReLU Activation layer. After each block, a Max-Pooling operation is performed for down-sampling. The number of feature channels is doubled at each down-sampling step.

b) Up-sampling: During upsampling of the feature map, at each step, it is concatenated with a corresponding cropped

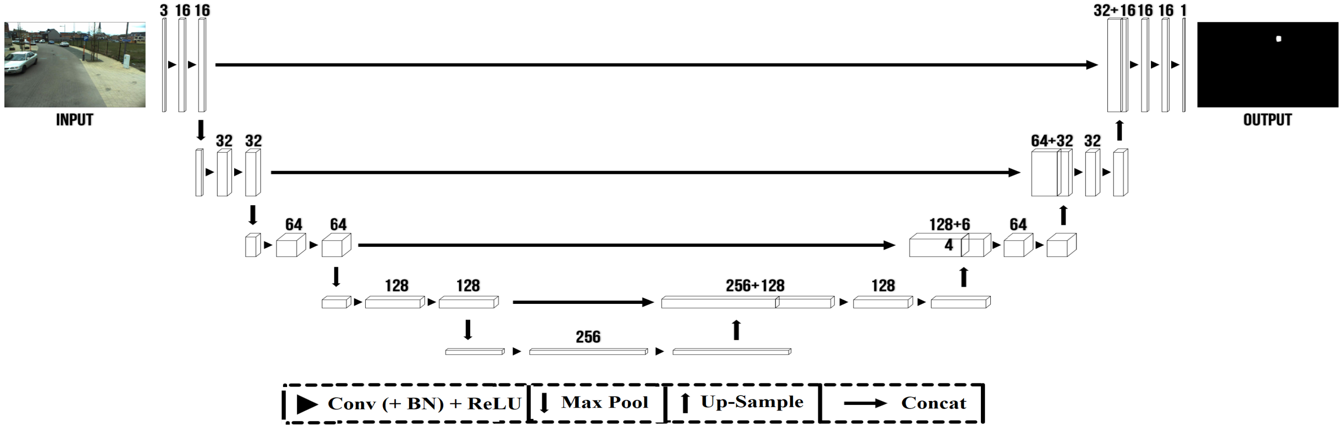

Fig. 4. CNN Architecture for Trafic-Sign Localization.

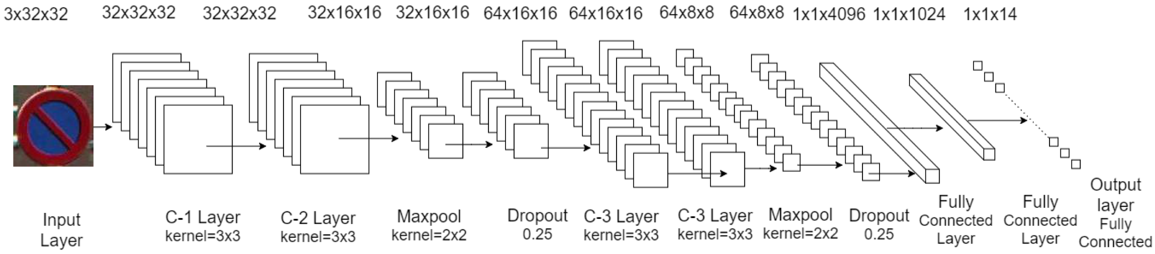

Fig. 5. CNN Architecture for Trafic-Sign Classification.

feature map from the downsampling stage. Here, cropping is necessary for the loss of border pixels in every convolution. Finally, two more convolution layers are added, which map each component feature vector from the previous layer to 2 classes (traffic sign and background).

The motivation for using the U-Net architecture is its robustness against low-resolution features of the image during training. In the up-sampling layers employed within it, there are a large number of feature channels which allow the network to propagate context information to higher resolution layers. This is quite an attractive feature for localizing objects in noisy conditions where the target object may not be dominantly visible in the image.

The U-Net architecture produces the location of the object by performing a pixel-wise image-segmentation. It predicts a probability for each pixel based on its likelihood of being present in a region corresponding to one of 2 classes (Traffic-Sign and Background). So, the traffic signs are detected as islands of high-probability regions in the image. The bounding box of these regions are the predicted traffic signs in this image.

4) Classification: The architecture of the CNN used for classifying the traffic sign proposals is shown in Fig. 5. It has 2 Convolution Blocks, each with two consecutive Convolution layers, a ReLU activation layer, and a Max-Pooling layer. A dropout layer is added after each block. Finally, there are 2 fully connected layers - the first one has a hidden size of 1024 with ReLU activation, and, the latter has a hidden size of 14 with Softmax activation. The output of this layer are the class-probabilities for each of the 14 sign-classes.

III. Experiments and Results

All of the experiments regarding training and implementation of the model were performed in an Intel Core-i7,3.60 GHz CPU with Nvidia GeForce GTX 1050 Ti (4 GB Memory) GPU hardware environment. The necessary codes were written in Python, and, the neural network models were implemented by using Tensorflow Backend in the Keras API.

A. Training

1) Challenge Detection Network: For training the challenge detector, our aim was to reduce the amount of background features as much as possible, so that the global feature (i.e, the challenge type) could be easily detected. To do this, we resized the image to 309×407 pixels, which is 1/16-th of the original size of the image. We do this in order to wash out all the local variations in the image.

In order to decrease the training time even further without sacrificing detection and classification accuracy, we train this network only with images of challenge level-3. Because, most of the traffic signs in the images below this threshold can be localized by using the no-challenge localizer. So, further increasing the training data would only redundantly increase the training time. But in case of Dirty lens, level 05 frames are quite different than it's previous level, so only for this case,

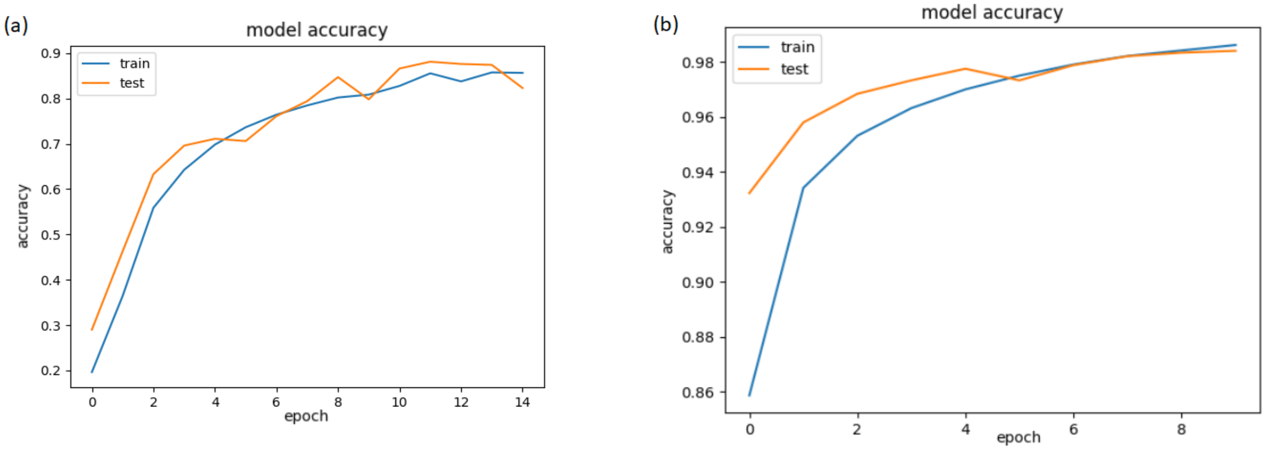

Fig. 6. Training History of (a) Challenge-Detector, and, (b) Classifier.

we used both level 03 and 05 frames for detecting challenge types properly.

We selected random 2000 frames for each challenge types which made in total 24000 samples. Among them, we used 23500 samples for training and remaining 500 samples for validation. The whole training was performed in total 20 epochs where random 5000 samples were used per epoch. We used checkpoint after every epoch so that weights can be saved only when the validation accuracy increases. We also used learning rate scheduler to reduce the learning rate to half after every 2 epochs. We set the initial learning rate to 1e-3 and used Adam as the optimizer. The training history of the challenge detection network is shown in Fig. 6(a).

TABLE IV
TRAINING RESULTS OF DIFFERENT LOCALIZERS

Localizer Model	Training IOU	Validation IOU
Common	0.83	0.70
Blur	0.76	0.65
Noise	0.75	0.66

2) Localizing Network: The main goal of the localizer model is to predict an image mask where the traffic sign regions are to be valued as 1 and all other regions are to be valued as 0.First, the Common Localizer is trained. The frames in the training datatset which contain traffic signs are first extracted by using the annotation files. Among them, the frames corresponding to the no-challenge and level-1 challenge are selected to train this network. At first, the model is trained with no-challenge frames and weights are saved. Then loading the pre-trained weights, we further trained the model using the level-1 challenge images. These images are used in order to bring variation in the training data, and, thus, to make the network robust to any false detection in the previous stage. For the first stage, total number of samples is 26904, number of epoch is 4 with initial learning rate 1e-3 and for the second stage, total number of samples is 3,22,848 among which 50,000 random samples are selected and the number of epoch is 10 with decreased initial learning rate of 1e-4.

The images are re-sized to 618 × 814 pixels (1/4-th of the original image)Re-sizing is done in order to reduce the number of parameters in the network.

After the Common Localizer is trained, the pre-trained weights of this network are then used to train the localizers for blur and noise. This results in faster convergence of the loss function, and, greatly reduces the time required to train the latter networks.

We used a custom loss function to train the localizers. The loss function for localization, L, is given by:

$$L = -\frac{2 \times (A_a \cap A_p) + \epsilon}{A_a + A_p + \epsilon}$$

where, A_a is the area of the bounding box regions in the actual image, A_p is the area of the predicted bounding box regions, and, $(A_a \cap A_p)$ is the area of intersection of these 2 regions. ϵ is a small safety factor added to handle division by 0 if there are no bounding boxes in the original image. We define the negative of L as the IOU (Intersection Over Union)-Score for the data.

We used Adam as optimizer, learning rate scheduler and model weight save checkpoint as before for training all these three models. The performance of all of these models on training and validation data is given in Table IV.

3) Classification Network: From the given video sequences, we first extracted the traffic sign regions from the bounding box co-ordinates provided in the annotation files. Then, we resized all the extracted regions to 32×32 pixels. An inspection of the dataset reveals that, there isn't much variation between the level-1 challenge and the no-challenge images. On the other hand, the images of challenge level-5 are so severely distorted that they may affect classification accuracy drastically. So, in order to minimize training time without compromising overall accuracy, we only used the traffic signs extracted from the images with no-challenge ,level-1 and level-3 challenge for training the classification network.

TABLE V
PRECISION-RECALL VALUES FOR DIFFERENT CHALLENGE TYPES AND LEVELS

	Challenge Type												
	00	01	02	03	04	05	06	07	08	09	10	11	12
Precision	0.733	0.643	0.559	0.44	0.644	0.612	0.512	0.582	0.640	0.482	0.635	0.631	0.520
Recall	0.610	0.372	0.485	0.342	0.461	0.435	0.325	0.378	0.353	0.288	0.572	0.420	0.234

Besides, while preparing the training dataset, it was found that, the data was highly biased towards the sign-type 14 (Parking Sign). It contained about twice the number of data than any other class. To reduce this bias, we randomly chose 3/4-th of the bounding boxes extracted for this sign-type, and, shuffled them with the remaining data to prepare the training-dataset. The network is trained with the entire training-dataset for 10 epochs. The training history of the Classifier is given in Fig. 6(b).

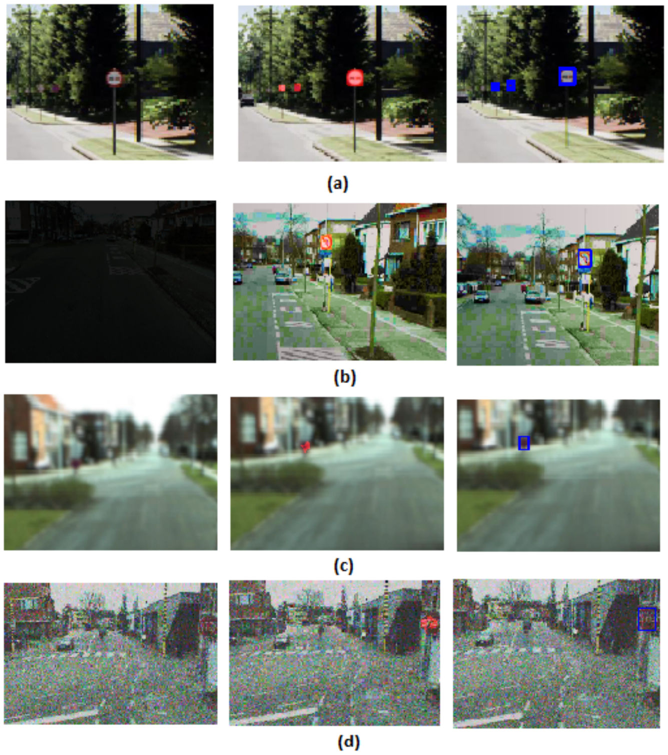

Fig. 7. Results of Our Trained Localizers on Different Challenge Type. Left: original image, Middle: predicted coarse sign-region, Right: bounding box. (a) No-Challenge, (b) Darkening (level-04), (c) Blur (level-04), (d) Noise (level-04).

B. Implementation

Our model was evaluated on the testing data provided with the IEEE VIP-Cup 2017 video dataset.

In the preprocessing stage of the model, Contrast Limited Adaptive Histogram Equalization was implemented by using the function createCLAHE() in the OpenCV library of Python.

After preprocessing, the input image is resized to 618×814 pixels and passed through the corresponding localizing network. The result of our trained localizers on different challenge type is shown in Fig. 7.

TABLE VI
PRECISION-RECALL VALUES FOR THE ENTIRE DATASET

	Real	Synthesized	Overall
Precision	0.32	0.65	0.62
Recall	0.19	0.43	0.41

After the extraction of ROIs, these are then fed into the sign classifier.

The precision and recall values for each challenge type averaged over all of the corresponding challenge levels are given in Table V. Finally, the net precision and recall values obtained after testing the pipeline against the entire dataset (real, synthesized and both) are given in Table VI.

IV. CONCLUSION

In this paper, we proposed a CNN-based model for detecting and classifying traffic signs from images which were captured in challenging conditions. We used CLAHE for preprocessing images, and, also used a CNN-based denoiser for images affected with Rain. The challenge-detector and classifier were built using VGG-16 type architecture, whereas, the localizers used a deep U-Net architecture. The model was tested and trained on the IEEE VIP Cup 2017 video dataset. The proposed model shows an overall precision and recall of 32% and 19% for real data, and, 65% and 43% for synthesized data. The reason for this bias is that there were much more frames with bounding boxes in the synthesized videos than there were in real videos. Besides, there were also some traffic-signs in the real video sequences which were not classified in the dataset. This resulted in erroneous training of the localizing models.

Our training procedure ensures a low training time for the model. The model is also suitable for real-time application due to its low computational cost at each stage. As such, it can be implemented in a standard PC configuration by using a low-power GPU.

REFERENCES

[1] X. Changzhen, W. Cong, M. Weixin, and S. Yanmei, “A Traffic Sign Detection Algorithm Based on Deep Convolutional Neural Network,” In *IEEE International Conference on Signal and Image Processing (ICSIP)*, Aug 2016, pp. 676-679.

[2] R. Qian, B. Zhang, Y. Yue, Z. Wang, and F. Coenen, “Robust Chinese Traffic Sign Detection and Recognition with Deep Convolutional Neural Network,” In *11th International Conference on Natural Computation (ICNC)*, Aug 2015, pp. 791 - 796.

[3] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Visual Recognition*. *CoRR*, abs/1409.1556v6, 10 April, 2015.

[4] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*. *CoRR*, abs/1505.04597v1, 18 May, 2015.

[5] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, “Traffic Sign Detection and Recognition using Fully Convolutional Network Guided Proposals”, *Neurocomputing*, 214(C): p.758-766, November 2016.

[6] M. H. A. Wahab, R. Latip, N. Zakaria, and R. A. Salam “Image Contrast Enhancement for Outdoor Machine Vision Applications,” In *2013 International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, Dec. 2013, pp. 377 - 383.

[7] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, J. Paisley, “Removing rain from single images via a deep detail network”. [Online] Available: www.columbia.edu/~jwp2128/Papers/FuHuangetal2017.pdf

[8] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, “Traffic sign recognition - how far are we from the solution?” In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug 2013, pp. 1-8.

[9] K. Lim, Y. Hong, Y. Choi, and H. Byun, “Real-time traffic sign recognition based on a general purpose GPU and deep-learning,” PLoS ONE 12(3): e0173317, March 6, 2017.

[10] I. Filkovic,*“Traffic Sign Localization and Classification Methods: An Overview”*. [Online] Available: http://www.fer.unizg.hr/_download/repository/KDI,_Ivan_Filkovic.pdf

[11] R. Qian, Y. Yue, F. Coeneny and B. Zhang, “Traffic Sign Recognition with Convolutional Neural Network Based on Max Pooling Positions,” In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, August 2016, pp. 578 - 582.

[12] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, “Traffic-Sign Detection and Classification in the Wild,” In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 2110 - 2118.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”. *CoRR*, abs/1406.4729v4, 23 April, 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. *CoRR*, abs/1506.01497v3, 6 Jan, 2016.

[15] D. Tom‘e, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi, and S. Tubaro, “Deep convolutional neural networks for pedestrian detection”. *CoRR*, abs/1510.03608v5, 7 Mar, 2016.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition”. *CoRR*, abs/1512.03385v1, 10 Dec, 2015.

[17] S. Dodge and L. Karam, “Understanding How Image Quality Affects Deep Neural Networks,” *CoRR*, abs/1604.04004v2, 21 April, 2016.

[18] L. Xu, J. S. Ren, C. Liu, and J. Jia, “Deep Convolutional Neural Network for Image Deconvolution”, In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, Dec 2014.

[19] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani1, “Self-Taught Object Localization with Deep Networks”. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2016, pp. 1 - 9.

[20] 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”, *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug. 2013, pp. 1 - 8.

[21] K. Audhkhasi, O. Osoba, and B. Kosko, “Noise-enhanced convolutional neural networks”, *Neural Networks* 2016, Vol.78, pp. 15 - 23.