NOISE REMOVAL IN TRAFFIC SIGN DETECTION SYSTEMS

Mohan Kumar G, M Shriram, and Dr. Rajeswari Sridhar

National Institute of Technology Tiruchirappalli, Tiruchirappalli, India

ABSTRACT

The application of Traffic sign detection and recognition is growing in traffic assistant driving systems and automatic driving systems. It helps drivers and automatic driving systems to detect and recognize the traffic signs effectively. However, it is found that it may be difficult for these systems to work in challenging environments like rain, haze, hue, etc. To help the detection systems to have better performance in challenging conditions like rain and haze, we propose the use of a deep learning technique based on a Convolutional Neural Network to process visual data. The processed data could be used in the detection. We are using the NoiseNet model [11], a noise reduction network for our architecture. The model is trained to enhance images in patches instead of as a whole. The training is done using the Challenging Unreal and Real Environment - Traffic Sign Detection Dataset(CURE-TSD) which contains videos of different roads in various challenging situations. The enhanced images obtained are compared using the object detection algorithms YOLO and Faster RCNN. The Mean Absolute Error(MAE) of original and enhanced images are calculated and compared for two classes of images - rain and haze for both the algorithms. The proposed approach achieved an average Peak Signal to Noise Ration(PSNR) of 25.30 and an Structural Similarity(SSIM) of 0.88. The average MAE values of YOLO and Faster RCNN model reduced by 0.11 and 0.30 respectively on using enhanced images.

KEYWORDS

Deraining, Dehazing, Noise Reduction, Object Detection.

1. INTRODUCTION

The research on automatic Traffic Sign Detection and Recognition (TSDR) systems constitutes a key component in the development of Advanced Driver-Assistance Systems (ADAS). TSDR systems have a wide range of applications including autonomous driving, driver safety, traffic surveillance, and road network maintenance. They have three main functions: detection, tracking, and classification. Traffic Sign Detection(TSD) involves localization of the traffic sign and Traffic Sign Recognition(TSR) involves classification of signs based on their type. TSD is used along with tracking to locate objects in real-time.

TSDR systems employ vehicle-mounted cameras that scan the roadside for traffic signs while driving on the road. The captured images are then processed through an image processing software and the located traffic sign is displayed on the vehicle dashboard. The software uses state-of-art object detection algorithms such as YOLO[5], Faster RCNN[6], RFCN[3] for the detection and recognition of the traffic signs.

While the TSRS works well under normal conditions the Real-time image processing faces some recognition challenges which include weather, lighting, occluded signs, image distortions, etc. Hence there is a need for preprocessing of the image captured in the mounted cameras to reduce the noise present in the images. For this purpose, a Convolutional Neural Network-based Noise

DOI:10.5121/cseij.2022.12608

Reduction model can be applied to the captured images before applying the algorithms for better detection.

In one of the works [7], researchers applied Deep Neural networks for denoising images. The work performs image sequences reconstruction rendered using Monte Carlo methods. It uses recurrent connections in a deep auto-encoder. The use of RNN helps in the retention of information between consecutive image frames, and also helps in attaining the information about the change of illumination over time.

Another work [8] proposes a Conditional Variational Image Deraining (CVID) network for the application of deraining, removing rain from images. It makes use of Conditional Variational Auto-Encoder (CVAE) for better predictions of grainy images. It provides a strong capability to model the latent distribution of image priors, from which the clean de-rain images can be generated.

This paper employs a noise reduction model which uses noisenet[54] architecture. This model is trained using the Challenging Unreal and Real Environment Traffic Sign Detection(CURE-TSD) dataset which contains videos of Traffic Signs on Roads under various challenging conditions like haze, rain, blur, etc. This paper analyzes the results of noise reduction. It also uses this model in the existing state-of-the-art object detection algorithms YOLO and Faster RCNN and compares the results of the detection.

The rest of the paper is organized as follows: Section 2 discusses some existing work highlighting the advantages and disadvantages, Section 3 discusses the proposed model, section 4 discusses the results of the model and section 5 discusses the conclusion obtained from results.

2. LITERATURE REVIEW

This paper [1] focuses on the 10 most common Thai traffic signs using a dataset derived from the combination of their dataset via Google Street View and the Belgium Traffic Sign Detection(BTSD) as well as the German Traffic Sign Recognition Benchmark(GTSRB) dataset. They employ the use of the YOLO version3 as their state-of-the-art recognition system that uses global instead of local features in its detection. Since they use only 10 traffic signs it greatly reduces the real-time use of this model. It is also a region-specific model which could only be applied in Thai. The dimensions of the image used are low. Hence it reduces the accuracy.

In the German Traffic Sign Detection Benchmark(GTSDB) detection system [2], the algorithms must only detect traffic signs in one of 4 major categories. Also, no negative samples should be disrupting the classification. In real-world tasks, the main difficulty when detecting and classifying traffic signs in an ordinary image is their very small size, often less than 1 percent of the image. Furthermore, the algorithm must filter out many potential negative cases while retaining true traffic signs. The paper proposed a new, more realistic benchmark, and has also used it to evaluate a combined CNN approach to traffic sign detection and classification. Compared with previous traffic sign benchmarks, images in this benchmark are more variable, and signs in these images are much smaller. It contains more images that are of high resolution. Their approach overall had a better recall and accuracy compared to the Fast-RCNN model. The main drawback here is that it is slow and not suitable to run on mobile devices in real-time. Since all images are panoramic, the model's accuracy is less when the traffic signs are larger.

Existing datasets are limited in terms of challenging environmental conditions and they lack metadata of challenge conditions and levels, and are also limited in terms of size and annotated continuous frames. It is not possible to determine the relationship between individual environmental conditions and algorithmic performance because multiple conditions change simultaneously. To overcome these shortcomings, in this work [3] they introduced the CURE-TSD dataset to benchmark algorithms under challenging conditions. All top-performing algorithms are based on deep architectures whose performance degrades significantly under challenging conditions. Therefore, conducted study articulates the urgent need for developing more robust algorithms that can adapt to the variations in environmental conditions. In this model, only datasets are introduced. There is no working model developed or implemented.

Here [4] researchers have addressed the problem of detecting and recognizing a large number of traffic-sign categories for the main purpose of automating the traffic-sign inventory management. Due to a large number of categories with a small inter-class but a high intra-class variability, researchers proposed detection and recognition utilizing an approach based on the Mask R-CNN detector. Overall, we showed that the deep learningbased approach can achieve extremely good performance for many traffic-sign categories, including several complex ones with large intra-class variability. The analysis revealed that ideal performance is still not achieved, mostly due to several missed detections that are being lost by the classification network. In this paper, we intend to handle the drawbacks of the existing approaches to denoising and is discussed in the next section.

3. DESIGN AND IMPLEMENTATION

3.1. Flow Diagram

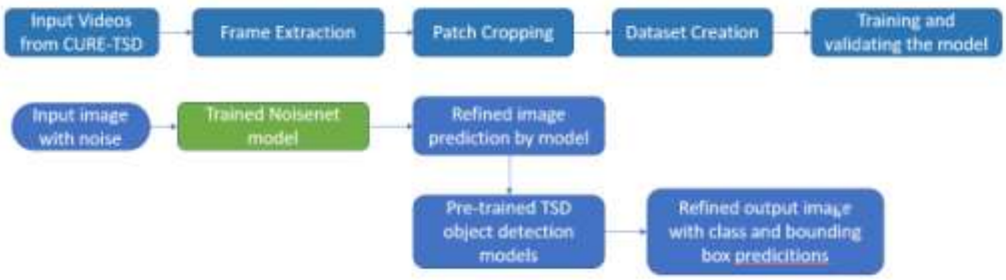

Fig.1: The figure shows the flow diagram of the proposed approach

Based on the drawbacks identified from existing literature, Figure 1 proposes the flow diagram of the innovated work. First, we extract frames from the videos of CURE-TSD dataset. From these frames randomly selected patches are taken as input for training the model and the enhanced patches are stitched together to form a new image. These images are used in the pre-trained object detection algorithms for comparison.

3.2. CURE Traffic Sign Detection Dataset

The CURE-TSD dataset contains video sequences of traffic signs under two classes, namely, Real data and Unreal data. The Real data contain real-world sequences without any challenging conditions while the Unreal data contains synthesized sequences of challenging conditions with a total of 5733 video sequences and 1.72 million frames.

3.3. Frame Extraction

Each video sequence in the CURE-TSD dataset has a length of 30 seconds running at 10 frames per seconds frame rate with a resolution of 1628 x 1236 px. Hence there are 300 frames extracted from each video sequence.

3.4. Patch Cropping

Since the image contains set of neighbouring pixels which have similar properties, it is more feasible to crop the image into patches than using the whole image. Patching allows us to reduce the number of parameters as well as the number of features to identify. It also helps model to detect noise with high granularity.

3.5. Dataset Creation

From each frame of the video sequence, 8 randomly selected patches of size 128 x 128 pixels are generated per frame in a video sequence. With 8 patches for each frame taken from 49 real video sequences, there are a total of 1,176,000 patches for the dataset.

3.6. Pre Trained Traffic Sign Detection Models

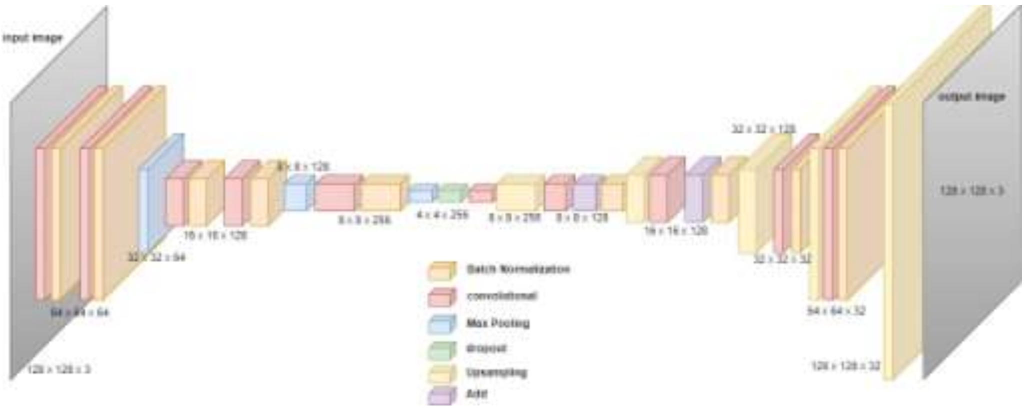

Fig.2: This is the autoencoder model used for the image enhancement

YOLO YOLOv2 is a widely used real-time object detection system and is efficient in terms of speed. It has been trained on image inputs of different sizes varying from 320x320 pixels to 608x608 pixels. It gives state-of-the-art accuracy on the PASCAL VOC and COCO datasets. YOLOv2 also uses the proposed Darknet-19 as the backbone for feature extractor, which achieves high accuracy while also requiring a lesser number of operations compared to the traditional VGG-16.

Faster RCNN Faster R-CNN is a state-of-the-art object detection system that uses a Region Proposal Network (RPN) to predict proposals from features. RPN is a convent that is used to classify whether an object is present or not, which is implemented with an nxn convolutional layer followed by sibling 1x1 convolutional layer, each of regression and classification.

3.7. Training the model

The Noise Reduction network architecture makes use of convolutional autoencoders for the task of removing unfavorable effects, such as rain, haze, snow, shadow, etc, on-road images.

The autoencoder architecture is modified to accept input in small patches of size 128x128

Table 1: Training data for Deraining

	Loss		PSNR		SSIM	
Epoch	loss	val loss	psnr	val psnr	ssim	val ssim
0	0.05169	0.06813	24.11414	22.40896	0.80655	0.81368
1	0.04384	0.04877	25.40016	24.42855	0.83841	0.83783
2	0.04154	0.04967	25.847	24.36257	0.85022	0.84501
3	0.04017	0.04647	26.11793	24.78131	0.85662	0.84782
4	0.03927	0.04037	26.30347	25.99954	0.86061	0.86039
5	0.03845	0.04536	26.46859	25.15241	0.86366	0.85929
6	0.03793	0.03822	26.5776	26.45248	0.86596	0.86409
7	0.0373	0.04181	26.70992	25.5658	0.86809	0.86233
8	0.03683	0.04285	26.80784	25.32444	0.86955	0.86017
9	0.03638	0.072	26.89968	21.67234	0.87093	0.83212

Table 2: Training data for Dehazing

	Loss		PSNR		SSIM	
Epoch	loss	val loss	psnr	val psnr	ssim	val ssim
0	0.03658	0.03172	26.75954	28.07038	0.84617	0.87362
1	0.02984	0.02528	28.46653	29.67486	0.88491	0.89692
2	0.02799	0.03485	29.07474	27.21154	0.89973	0.90026
3	0.02684	0.02883	29.45848	28.69486	0.90712	0.90914
4	0.02602	0.02837	29.73504	29.12993	0.91176	0.906
5	0.02544	0.02558	29.9304	29.81732	0.91455	0.91398
6	0.02514	0.02867	30.03151	28.95769	0.91626	0.91106
7	0.02467	0.0314	30.17931	28.0332	0.91744	0.90977
8	0.02421	0.02673	30.34277	29.45561	0.91964	0.91724
9	0.02365	0.02821	30.53555	28.92132	0.92164	0.91857

pixels. This helps in reducing the parameters required and time taken for training. The model consists of various convolutional layers, normalization, and pooling for Encoding and Decoding. The detailed architecture is shown in Figure 2. The architecture consists of a total of 1,638,883 parameters, out of which 1,636,963 parameters are trainable, and the rest are non-trainable parameters.

The dataset consisting of cropped patches contains both the original and the noisy data. The patches consisting of noise are given as input to the model. The autoencoder architecture has an encoder which encodes the feature data of the given patch into the latent space which is then decoded by the decoder to be converted back. These features are compared against the original data and evaluated for PSNR, SSIM and loss in each epoch. This helps the model learn how to efficiently encode the noisy input features and decode it into enhanced image features.

The framework used for training is Tensorflow/Keras. The model is trained using the created dataset for 10 epochs. For better optimization, Adam Optimizer is used. The patches are passed through the model for enhancement and the enhanced patches are stitched together to form the new enhanced image.

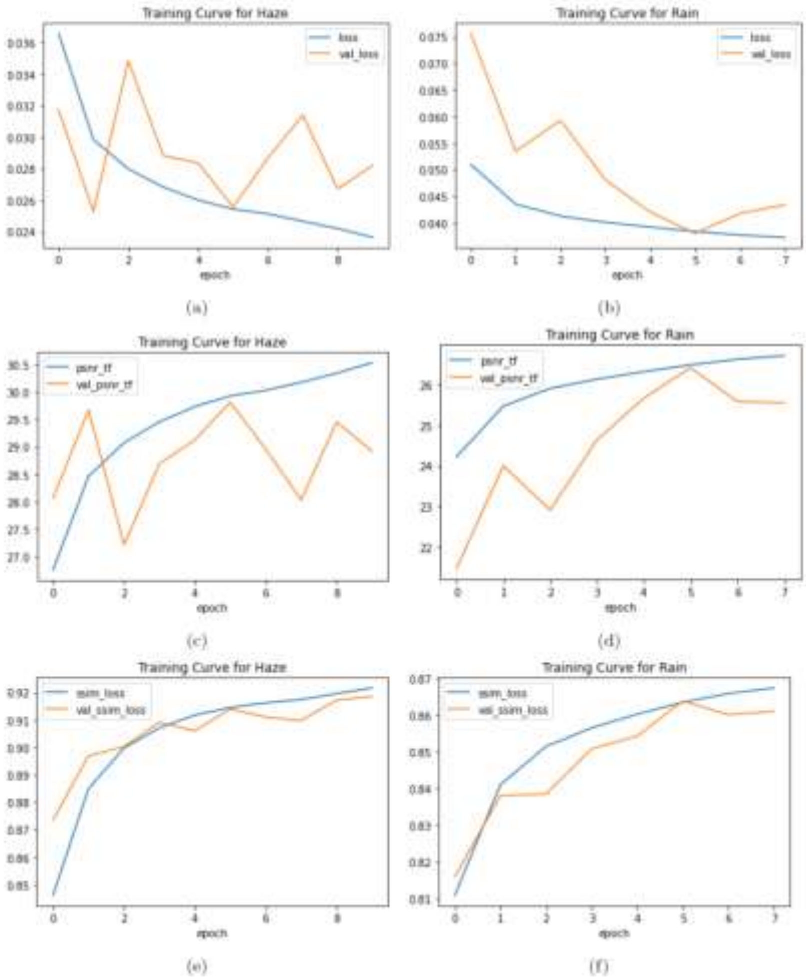

Fig.3: Training Curves

4. Results and Discussion

The model was trained with Adam Optimizer and L1 loss which is the Mean Absolute Error. The metrics used for the evaluation of the model are SSIM(Structural Similarity) and PSNR (Peak Signal-to-Noise Ratio).

SSIM is a method for measuring the similarity between two images. The SSIM index is used to compare the quality of one image concerning another image that is considered to be of perfect quality. The PSNR is used as a quality measurement between the original and the reconstructed image. A higher PSNR value indicates better quality of reconstruction.

Tables 1 and 2 shows the results of training the model for deraining and dehazing respectively. Figures 3a to 3f shows training and validation curves indicating SSIM loss, PSNR, and an overall loss for the dehazing and deraining models across 10 epochs. Figures 4 and 5 shows the comparison of original and enhanced image in YOLO for dehazing and deraining respectively. The figures 6 and 7 shows the original and enhanced image comparison in Faster RCNN for dehazing and deraining respectively.

Fig.4: Haze Image comparisons for YOLO

Fig.5: Rain Image comparisons for YOLO

The new images obtained via image enhancement are faded in as inputs to object detection algorithms Faster RCNN and YOLO. The results show that the detection in enhanced images is better and more accurate than in the unenhanced images.

The proposed work achieved a lesser MAE for the images and the trained model has an average PSNR of 25.30 and an average SSIM of 0.88. For YOLO, the MAE of the original rainy and enhanced images are 0.74 and 0.64 respectively while that of original hazy and enhanced images are 0.38 and 0.27 respectively. For Faster RCNN, the MAE of the original rainy and enhanced images are 0.59 and 0.23 respectively while that of original

Fig.6: Haze Image comparisons for Faster RCNN

Fig.7: Rain Image comparisons for Faster RCNN

hazy and enhanced images are 0.30 and 0.07 respectively which is a big improvement than the existing approaches as mentioned in 2. The noisenet architecture is well suited for this dataset as it captured the blurred image due to weather conditions like rain and haze better than other architectures

5. Conclusion and Future Work

The results show that the detection and recognition accuracy of the object detection models has improved when they refine the image based on the environment before directly sending it into the TSDR systems. The dehazing and deraining of images can improve traffic sign detection in real-time situations.

From the results obtained, we can infer that the Faster RCNN model has produced significant enhancements, especially in the de-hazed images, compared to the YOLOv2 model. An observation to be made is that even though the no of epochs for enhancing images of both rainy and hazy types were taken to be 10, the de-hazed images seemed to be more sharper as compared to de-rained images. Hence, in order to further improve the results, we could induce data augmentation in our pre-processing steps, and also increase the no of training epochs as long as we do not overfit our image enhancement model. To compensate for the sharpness lost during image enhancement, the TSD models could also be further trained on lower resolution images in order to make better and more accurate detections.

As future work, we could also explore more of the challenging conditions, as some of them which are given in the CURE-TSD dataset, like snow, shadow, dirty lens, codec error, and darkening.

REFERENCES

[1] Alvaro´ Arcos-Garc´ıa, Juan A. Alvarez-Garc´ıa,´ Luis M. Soria-Morillo, Evaluation of deep neural networks for traffic sign detection systems, Neurocomputing, Volume 316, 2018, Pages 332-344, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2018.08.009. (https://www.sciencedirect.com/science/article/pii/S092523121830924X)

[2] M. Shahud, J. Bajracharya, P. Praneetpolgrang and S. Petcharee, ”Thai Traffic Sign Detection and Recognition Using Convolutional Neural Networks,” 2018 22nd International Computer Science and Engineering Conference (ICSEC), 2018, pp. 1-5, doi: 10.1109/ICSEC.2018.8712662.

[3] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li and S. Hu, ”Traffic-Sign Detection and Classification in the Wild,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2110-2118, doi: 10.1109/CVPR.2016.232.

[4] Dogancan Temel, Tariq Alshawi, Min-Hung Chen and Ghassan AlRegib, ”Challenging Environments for Traffic Sign Detection: Reliability Assessment under Inclement Conditions”, 2019, url: https://doi.org/10.48550/arXiv.1902.06857

[5] J. Redmon and A. Farhadi, ”YOLO9000: Better, Faster, Stronger,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.

[6] Shaoqing Ren, Kaiming He and Ross Girshick and Jian Sun, ”Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, 2016, url: https://doi.org/10.48550/arXiv.1506.01497

[7] Jifeng Dai and Yi Li and Kaiming He and Jian Sun, ”R-FCN: Object Detection via Region-based Fully Convolutional Networks”, 2016. url: https://doi.org/10.48550/arXiv.1506.01497

[8] Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. ACM Trans. Graph. 36, 4, Article 98 (August 2017), 12 pages. https://doi.org/10.1145/3072959.3073601

[9] D. Temel, M. -H. Chen and G. AlRegib, ”Traffic Sign Detection Under Challenging Conditions: A Deeper Look into Performance Variations and Spectral Characteristics,” in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 9, pp. 3663-3673, Sept. 2020, doi: 10.1109/TITS.2019.2931429.

[10] Dogancan Temel, Tariq Alshawi, Min-Hung Chen and Ghassan AlRegib, ”Challenging Environments for Traffic Sign Detection: Reliability Assessment under Inclement Conditions”, 2019, url: https://doi.org/10.48550/arXiv.1902.06857

[11] Suhail Najeeb, ”Removal of Artifacts from Vehicle Mounted Images using Convolutional Autoencoders”, feb 2021, url : https://github.com/suhailnajeeb/cure-tsd-revisit

AUTHORS

Mohan Kumar G is currently pursuing his B.Tech Computer Science and Engineering from the National Institute of Technology, Tiruchirappalli. His research interests include Machine Learning, Natural Language Processing, and Computer Vision.

M Shriram is currently pursuing his B.Tech Computer Science and Engineering from the National Institute of Technology, Tiruchirappalli. His research interests include Machine Learning, Natural Language Processing, and Computer Vision.

Dr. Rajeswari Sridhar received M.S. from City University, WA, USA. She did Bachelor’s degree in Electronics and Communication Engineering. She received Ph.D. in Computer Science from Anna University. Her research interests include Compilers, Algorithms, Machine learning, NLP, Cloud Computing, and Social Media analysis.