Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-08848-z

Performance enhancement techniques for traffic sign recognition using a deep neural network

Jameel Ahmed Khan[1] **· Yunfan Chen**[1] **· Yawar Rehman**[2] **· Hyunchul Shin**[1]

Received: 13 May 2019 / Revised: 4 January 2020 / Accepted: 13 March 2020
Published online: 20 April 2020

Abstract
An advanced traffic sign recognition (ATSR) system using novel pre-processing techniques and optimization techniques has been proposed. During the pre-processing of input road images, color contrasts are enhanced and edges are made clearer, for easier detection of small-sized traffic signs. YOLOv3 has been modified to build our traffic sign detector, since it is an efficient and effective deep neural network. In this YOLOv3 modifications, grid optimization and anchor box optimization were done to optimize the detection performance on small-sized traffic signs. We trained the system on our traffic sign dataset and tested the recognition performance using the Mean Average Precision (MAP) on the Korean Traffic Sign Dataset (KTSD) and German Traffic Sign Detection Benchmark (GTSDB). We used the bisection method for selecting the optimum threshold of confidence score to reduce false predictions. Our ATSR system is capable of recognizing Prohibitory, Mandatory, and Danger class traffic signs from road images. ATSR can detect small-sized traffic signs accurately along with big-sized traffic signs. It shows the best recognition performance of 98.15% on the challenging KTSD (the previously reported best performance was 90.07%) and 100% on the GTSDB. Result comparisons show that ATSR significantly outperforms ITSR, TS detector, YOLOv3, and D-patches, on KTSD.

Keywords Traffic sign recognition · Deep neural network · Pre-processing · Optimization · YOLOv3

1 Introduction

Real-time and accurate recognition of objects from road images becomes an important challenge, as self-driving vehicles and Advance Driver Assistance Systems (ADASs) are

✉ Hyunchul Shin
shin@hanyang.ac.kr

[1] Division of Electrical Engineering, Hanyang University, Ansan 15588, Republic of Korea

[2] Department of Electronic Engineering, NED University, Karachi 75270, Pakistan

getting popular. Traffic sign recognition systems can recognize the road conditions from road images, and it can warn the driver about any dangerous condition. Recognition of traffic signs from road images is challenging because there are large variations in shape, size, and color of traffic signs. In addition, these signs are often occluded by road objects like trees and poles. Furthermore, the size of traffic signs also varies with the distance. In ITSR [7], dark area sensitive tone mapping (DASTM) technique is used to cope with the luminance problems especially when the traffic signs look dark. Still ITSR [7] frequently fails to detect small-sized traffic signs. In this work, we have focused on resolving the small-sized traffic sign detection problem by using novel pre-processing and deep neural network optimization techniques. Our proposed advanced traffic sign recognition (ATSR) is capable of detecting small-sized traffic signs owing to new pre-processing techniques, optimization of the grids and thresholds of the deep neural network based on YOLOv3 [14]. The ASTR is trained by using our new dataset containing three super classes of traffic signs. This system can recognize Prohibitory, Mandatory, and Danger class traffic signs, which are shown in Fig. 1. After optimization, the system becomes sensitive for small traffic signs, but it became so sensitive that it detects many other false objects as traffic signs. These false positives cause errors, and the mean average precision (MAP) of detection was affected. To resolve this issue, the threshold on the confidence score of detection was optimized by using the bisection method. We tested our ATSR system on Korean Traffic Sign Dataset (KTSD) and German Traffic Sign Dataset Benchmark (GTSDB) [5]. On KTSD, our ATSR gives the best performance. Figure 2 shows examples of prohibitory, mandatory, and danger class traffic signs detected by ATSR. Experimental results show that ATSR outperforms D-patches [19], YOLOv3 [14], TS detector [11], and ITSR [7], by more than 8%, in terms of the Mean Average Precision (MAP).

2 Related works

Yawar et al. proposed D-patches [19], a traffic sign detection framework that is capable of handling partial occlusions. They divided traffic signs into discriminative patches and extracted the features from these patches. These features were used for classification of traffic signs.

Fig. 1 Example signs of the three classes in the Korean Traffic Sign Dataset

Fig. 2 Experimental detection results of ATSR, on prohibitory, mandatory, and danger class traffic signs

Sliding windows and Aggregate Channel Features (ACF) [2] are used for classification. Although the D-patches [19] algorithm is slow, but it is effective in occlusion cases due to patch-wise classification. However, D-patches [19] can detect only one class of traffic signs at a time. For detecting three classes of traffic signs, an image must be tested three times for each image, eventually increasing the CPU time by three folds.

In recent years, many object detection algorithms have been proposed that use deep neural networks. AlexNet [9], Fast RCNN [4], and Faster RCNN [16] are the famous neural networks. We have selected YOLOv3 [14] because of its good performance as it can detect many classes with near real time processing speed. Multilayer fusion techniques [1] also use the convolutional neural network for object detection from thermal and visible images. Zhu et al. [21], collected a huge amount of data for traffic sign detection and used it for Chinese traffic sign detection. Their key contribution is the collection and annotation of big data. Two parallel neural networks were trained on 10,000 images. One for detection and the other for detection and classification. However, this doubled the computation time.

The TS detector [11], a deep learning-based traffic sign detection framework, uses optimized YOLOv3 [14] for traffic sign detection. Although the TS detector [11] is a fast detector, it fails in low illumination, small-sized, and low-resolution conditions. The TS detector also shows inferior performance in occlusion conditions. The performance of the TS detector was improved in ITSR [7] by applying DASTM as pre-processing, and it became robust for low illumination conditions. DASTM made the system effective, and MAP performance was significantly increased. Among previous methods, ITSR [7] shows the best performance of 90.07% MAP on KTSD. ITSR [7] is further improved in our proposed ATSR by applying grid optimization and image pre-processing. ATSR achieves 98.15% MAP on KTSD.

YOLOv1 [15] divides an image into a grid. YOLO9000 [13] is the improved version of YOLOv1. YOLOv3 [14] is the latest version and it outperforms its previous versions. YOLOv3 is a 106-layered deep neural network that takes the input image and applies convolution with filters in convolutional layers to resize it into a smaller grid. It has three detection stages called YOLO layers. In each YOLO layer, the image has a different grid size. Detection is performed in a YOLO layer using anchor boxes, and the average loss is calculated. After detection, non-max suppression is applied to remove false predictions. Chung et al. [17] used YOLO for traffic sign detection. They added additional convolutional layers in the YOLO architecture and used this model for detecting small sized traffic signs. They trained their model on 13 classes using the Belgium Traffic Sign Dataset (BTSD) [12]. Ayoub et al. [3] used random forests for traffic sign recognition. Image enhancing and thresholding using HSV space was used with circular, triangular and rectangular shape features including random forests to classify the traffic signs. Jameel et al. [6] used Deep Neural Network on hazy images for recognition of traffic signs. They applied dehazing algorithm as pre-processing and then optimized YOLOv3 for recognition of Prohibitory, Mandatory, and Danger classes of traffic signs. Although dehazing algorithm proposed in this paper increases detection performance on hazy day images but it degrades the quality of clear day images, causing loss in recognition performance. Dong et al. [10] proposed an algorithm called DeepSign. This algorithm has three modules, detection module (PoseNet), classification module (PatchNet), and a temporal filter. This algorithm has 87.3% detection accuracy, but it requires more processing time due to temporal filtering. Zhang et al. [20] proposed a 13 layered unique CNN for fruit classification from images and achieved 94.94% accuracy. They used image rotation, gamma correction and noise injection methods for data augmentation. Wenjuan et al. [8] proposed an algorithm which classifies MRI images of human brain into five different classes. They used data augmentation technology for unbalanced dataset, deep stacked sparse autoencoder for training, and softmax layer for classification. They achieved 98.6% accuracy. Yi et al. [18] proposed traffic sign recognition system which is based on color probability model and color histogram of oriented gradient (HOG). They tested their results on German Traffic Sign Detection benchmark (GTSDB) [5] and self-constructed Chinese Traffic Sign Dataset (CTSD). They trained their system on conventional SVM using HOG features. However, it is known that performance of deep neural networks for recognition of traffic signs is frequently much better than conventional SVMs.

3 Proposed method

In our proposed method, pre-processing has been applied before detection. Grid optimization and anchor box optimization have also been applied in detection module. The block diagram of our method and its comparison with other algorithms is shown in Fig. 3.

3.1 Image sharpening and contrast enhancement

Our goal is to make small sized, low-resolution traffic signs sensitive for detection. We observed that the traffic signs with high contrast and sharp edges are more sensitive for detection. Therefore, we have developed a technique that enhances the contrast of images and sharpens the images. This sharpening algorithm is described below.

Let I be the input image. We applied average filtering on I to make it blur and named the filtered image as I_{filt}. We used 7×7 sized kernel for average filtering. Now the sharpened image I_{sharp} is given by

$$I_{sharp} = I + \left(I - I_{filt}\right) \times f \tag{1}$$

where f is the sharpening parameter, and it range is $f \geq 0$. By increasing f, sharpening increases. We tested our detection results for different values of f from 0 to 2, and experimentally achieved our best results when $f = 2$. By increasing f more than 2, the quality of image degrades due to over sharpening and thus decreases the detection performance is affected. The main advantage of this proposed technique is to enhance the contrast and edges of an input image to make small traffic signs more sensitive for detection.

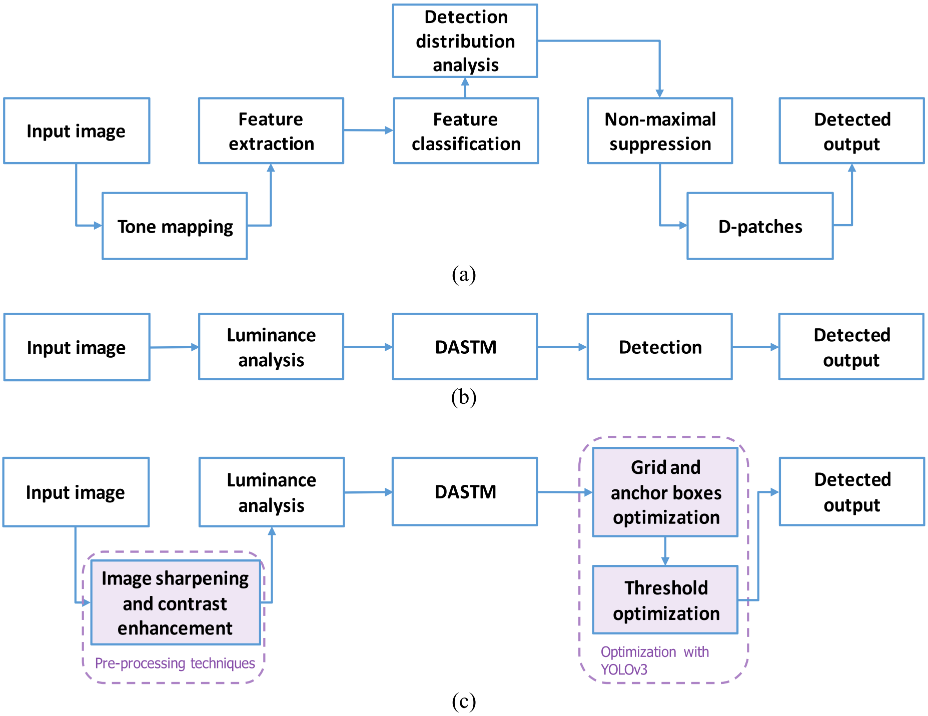

Fig. 3 Recent traffic sign detection flow diagrams vs proposed ATSR flow diagram. **a** D-patches flow diagram [19], **b** ITSR flow diagram [7], **c** Proposed ATSR flow diagram

3.2 Effect of pre-processing on detection

In big sized traffic signs, the features are very prominent, and detector can easily identify them. In Korean traffic sign dataset, most of the traffic signs are in small size, low quality, and low-resolution conditions. Therefore, recognition of these traffic signs is a challenging task. Our novel and self-developed image sharpening technique enhances the edge features of these small-sized traffic signs, causing the detection to become easier. ITSR [7] shows poor detection performance on small-sized traffic signs. In comparison, ATSR enhances the contrast and sharpness on edges, allowing these low-quality traffic signs to become sensitive for detection at high probability. In Fig. 4, we can compare the original image with the sharpened image. The contrast of the image is increased, and the edges are sharpened. In Fig. 5, the detection comparison of our ATSR and ITSR [7] is shown. One can see that some small-sized traffic signs missed by the ITSR can be accurately detected by ASTR. The reason of accurate detection is the contrast enhancement and edges enhancement of small-sized traffic signs by our pre-processing algorithm.

3.3 Grid size optimization

YOLOv3 [14] has a 106-layered architecture. In a convolutional layer, YOLOv3 [14] convolves the input image with filters, and the size of the image is reduced in the next layer. The size of an image in the next layer is calculated by the following formula:

Fig. 4. Original image with low quality small traffic signs (left), the quality of traffic signs increased by our sharpening technique (right)

$$\text{Output image size} = \left\{\frac{\mathrm{N} + 2\mathrm{P} - \mathrm{F}}{\mathrm{S}}\right\} + 1 \qquad (2)$$

where N is the resolution of the input image. For the 480 × 480 image, $N = 480$. The filter size is F, S is the stride, and P is the padding. Along with the convolutional layers, there are up-sampling layers, shortcut layers, and route layers. Detection is performed in YOLO layers using a total number of nine anchor boxes. There are three YOLO layers in YOLOv3 [14]. In these layers, the size of the image (grid size) is different. In YOLOv3 [14], first detection is performed in layer 82, where the grid size is 35 × 35. Second detection is performed in layer 94, where the grid size is 70 × 70. Third detection is performed in layer 106, where the size of the grid is 140 × 140. The reason for the variation in grid size is to cope with the variation of the object size. Big sized objects are detected in the first detection stage, while a denser grid is

(a) (b)

Fig. 5. Detection comparison of our ATSR and ITSR [7]. **a** Detection result of ITSR [7] detector, **b** Detection result of ATSR

needed for detecting small objects. Original YOLOv3 is designed for detecting big sized and medium sized objects. It gives very poor performance for detecting small objects due to the small grid size (N) in third detection stage. We optimized grid size in the third detection stage to get denser grid and to make it suitable for detecting small sized traffic signs. The grid sizes of YOLOv3 [14], TS detector [11], and ATSR are shown in Table 1. The basic architecture of YOLO [15] are shown in Fig. 6.

3.4 Anchor box optimization

YOLOv3 [14] uses anchor boxes in YOLO layers for detection of objects. Let Y be the detection vector, then the size of Y depends upon the number of anchor boxes used in the YOLO layer. The size of Y is given by the following formula

$$\text{Size of Y} \; = (\mathrm{N} \times \mathrm{N}) \times \mathrm{A} \times (5 + \mathrm{C}) \tag{3}$$

where N is the resolution of the grid as stated in Eq. (2). A is the number of anchor boxes used in the detection stage, and C is the number of object classes. If Y is the ground truth vector and $\hat{\mathrm{Y}}$ is the predicted vector, then the loss of the detection in each iteration is calculated by using the following formula

$$\mathrm{L}\left(\widehat{\mathrm{Y}}, \mathrm{Y}\right) = \left(\widehat{\mathrm{Y}}_1 - \mathrm{Y}_1\right)^2 + \left(\widehat{\mathrm{Y}}_2 - \mathrm{Y}_2\right)^2 + \left(\widehat{\mathrm{Y}}_3 - \mathrm{Y}_3\right)^2 + \ldots \left(\widehat{\mathrm{Y}}_\mathrm{B} - \mathrm{Y}_\mathrm{B}\right)^2 \tag{4}$$

where

$$\mathrm{B} = 5 + \mathrm{C} \tag{5}$$

In Eqs. (3) and (5), the constant 5 is used because there are five parameters i.e. *pc*, *bx*, *by*, *bh*, *bw*, in the detection vector Y. The parameter *pc* tells about the existence of the object, and its value is either 1 or 0. If *pc* is 1, it means that an object exists in the grid cell, and *pc* = 0, it means there is no object in the grid cell. The parameters *bx* and *by* tell us about the position of the center of an anchor box, *bh* and *bw* represent the height and width of an anchor box.

In YOLOv3 [14], a total of nine anchor boxes are used in three detection stages, and these boxes are numbered from 0 to 8. Anchor box 0 is the smallest-sized box and anchor box 8 is the largest-sized box. Anchor box numbers 6, 7, 8 are used in the 1st detection stage, 3, 4, 5 in the 2nd detection stage, and 0, 1, 2 in the 3rd detection stage. The 3rd detection stage is only for detecting small objects and thus small sized anchor boxes 0, 1, 2 are used in the 3rd detection stage. We optimized the number of anchor boxes according to our need. We calculated the nine anchor boxes from our training data and numbered them from 0 to 8. In our model, we optimized grid size for detecting small sized traffic signs, so we increased the number of grids in the 3rd detection stage, reducing the grid cell size. In

Table 1 Grid sizes of four detectors

Detection method	N in first detection stage	N in second detection stage	N in third detection stage
YOLOv3 [14]	35	70	140
TS detector [11]	35	70	280
ITSR [7]	35	70	280
ATSR (ours)	35	70	560

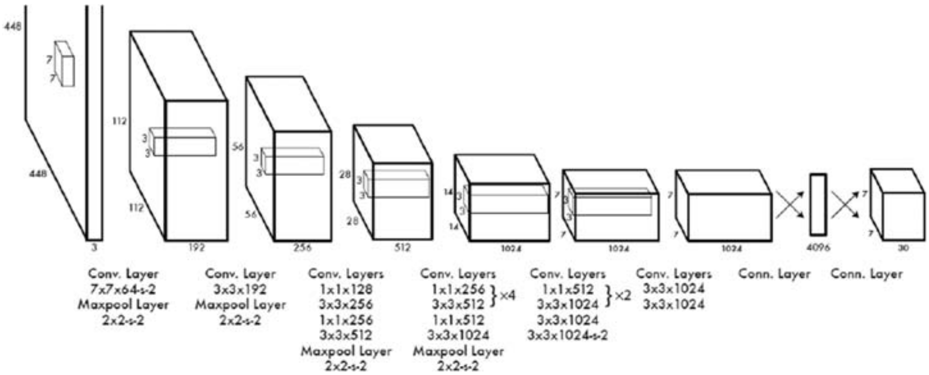

Fig. 6 Basic architecture of YOLO [15]

ATSR, we used anchor box 0,1 in the 3rd detection stage, 2, 3 in the 2nd detection stage, and 4, 5, 6, 7, 8 in the 1st detection stage. This anchor box optimization increased our detection performance.

3.5 Confidence score thresholding by the bisection method

After grid optimization, the system detects many small objects as traffic signs. These false predictions usually have a small confidence score. We applied a threshold on the confidence score for removing false predictions. This thresholding improved the detection performance of ATSR. Selection of the optimum threshold for a high-performance system is also a challenging task. We used bisection method (https://en.wikipedia.org/wiki/Bisection_method) to find the optimum threshold. Initially we set the threshold of confidence score as 25. The value 25 is the mid-point of the interval 20 to 30. The bisection method calculates the detection accuracy automatically at the mid-point above and below the interval. In this case, the accuracy was calculated at 27.5 and at 22.5. After that the maximum accuracy among these two values is selected. This optimization using bisection continues until the optimum threshold with maximum detection accuracy is achieved.

4 Experimental results

4.1 Datasets details

We trained the model on our self-made dataset containing 3200 images and 100 images for validation. This training dataset includes small-sized, medium-sized, and large-sized traffic signs of all three classes. The dataset also contains traffic sign images taken from different view point angles and under different lighting conditions. Moreover, the dataset also contains both high resolution and low-resolution images for better training. We used initial weights trained on the COCO dataset (http://cocodataset.org/#home) and applied fine tuning on our dataset. While training, we continuously monitored the system to avoid over-fitting. Testing was done on the widely used Korean Traffic Sign Dataset (KTSD) and German Traffic Sign Detection Benchmark (GTSDB) [5].

4.2 Detection performance evaluation

Extensive experiments were conducted to demonstrate the effectiveness of our proposed pre-processing techniques and optimization techniques on YOLOv3. Performance was evaluated by the Mean Average Precision (MAP). We used the Cartucho method (https://github.com/Cartucho/mAP) for MAP calculation. Average precision is calculated for each Prohibitory, Mandatory and Danger class traffic signs. By following the same evaluation method as discussed in ITSR [7], the predicted bounding box is matched with the ground truth bounding box using Intersection Over Union (IOU) to decide either the prediction is true or false.

$$\mathrm{IOU} = (\text{Area of intersection of bounding boxes})/(\text{Area of union of bounding boxes}) \quad (6)$$

The prediction is considered as true positive (TP) if the IOU $\geq$ 40%, otherwise it will be considered as false positive (FP). Precision and Recall were calculated by the following equations.

$$\text{Precision} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \quad (7)$$

$$\text{Recall} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \quad (8)$$

4.2.1 Performance verification of YOLOv3 optimization technique

As shown in Table 2, our optimized YOLOv3 significantly outperforms the original YOLOv3 [14], by improving the MAP about 14%. This indicates that our proposed grid size and anchor box optimization method is effective at improving the detection rate of small-sized traffic signs. As shown in Table 2, pre-processing algorithm further improves the MAP about 9.91%. Hence our ATSR out-performs the original YOLOv3 [14], by improving the MAP about 24.21% at the cost of very small processing time. One can see in Table 4 that ATSR uses 0.063 s while YOLOv3 [14] uses 0.050 s to process an image of 800 × 600 resolution.

Figure 7 illustrates a visual comparison of the original YOLOv3 detection results versus the optimized YOLOv3 detection results. One can see that the detection bounding boxes in Fig. 7a are not optimum, and small-sized traffic signs are missed by the detector, while both of these problems have been resolved in Fig. 7b.

4.2.2 Performance verification of the pre-processing technique

By adding our proposed pre-processing techniques in the optimized YOLOv3, the detection MAP is further improved by almost 10%, from 88.24% to 98.15%, as shown in Table 2. This

Table 2 Performance comparison of several detectors at different settings

Detector	MAP on KTSD
Original YOLOv3 [14]	73.94%
Optimized YOLOv3	88.24%
Optimized YOLOv3 + pre-processing (ATSR)	98.15%

(a) (b)

Fig. 7 Detection results comparison of original YOLOv3(left) and optimized YOLOv3(right)

comparison demonstrates that the proposed pre-processing techniques are excellent to improve the image quality. The visual comparison of the optimized YOLOv3 detection results versus the optimized YOLOv3 + pre-processing detection results are shown in Fig. 8. One can easily compare the performance of our ATSR with YOLOv3 for detection of small sized, and low-resolution traffic signs. In Fig. 8a, small-sized traffic signs were missed by the detector YOLOv3 while in Fig. 8b, these signs were successfully detected by ATSR. This shows that performance of ATSR is significantly improved.

4.2.3 Performance comparison of ATSR with other state-of-the-art methods

The Mean Average Precision (MAP) and CPU time performance of several methods are depicted in Tables 3 and 4. The MAP comparison in Table 3 shows that our ATSR outperforms all other detectors on the challenging Korean Traffic Sign Dataset (KTSD) and it also shows 100% recognition rate on GTSDB. MAP is calculated using Cartuchos method (https://github.com/Cartucho/mAP).

4.2.4 CPU time comparison

For our experiment, we used a personal computer with 16 GB RAM, core i7 CPU, and single NVIDIA TITAN X GEFORCE GTX GPU installed on it. Processing speed can further be accelerated by using multiple GPUs in a server. Table 4 shows the comparison of the processing times of traffic sign detectors. Although YOLOv3 [14] is fastest, its performance is not satisfactory. The TS detector [11] is faster than all the other detectors except YOLOv3 [14]. However, the TS detector [11] often fails in low illumination conditions. ITSR [7] uses DASTM to improve the detection of low illuminated signs. ATSR uses image sharpening and DASTM.

(a) (b)

Fig. 8 Detection results comparison of optimized YOLOv3 (left) and optimized YOLOv3 + pre-processing(right)

4.2.5 Testing in different weather conditions

Weather noises like clouds, snow, and rain, usually degrade the quality of an image causing a decrease in detection performance. We tested our system on images taken in different weather conditions. The detection results are shown in Fig. 9. The images shown in Fig. 9 were not the part of KTSD but were collected solely for the purpose of testing. One can see that ATSR is robust against slight weather noises.

4.2.6 Error analysis

ATSR gives 98.15% accuracy on KTSD, still 1.85% improvement is required. Although thresholding is applied on confidence score for removing false predictions, but still there are some false positives with high confidence score. ATSR also misses some low quality, small sized traffic signs, causing 1.85% errors. In Fig. 9d, out of three traffic signs, two distant traffic signs were missed by the detector due to dense rain noise. Hence, ATSR works properly in

Table 3 Mean Average Precision (MAP) based comparison of recent detectors

Detection method	MAP on KTSD	MAP on GTSDB
YOLOv3 [14]	73.94%	96.53%
TS detector [11]	86.61%	97.82%
D-Patches [19]	79.60%	100%
ITSR [7]	90.07%	100%
ATSR (ours)	98.15%	100%

Table 4 CPU time comparisons of recent detectors

Detection method	Image resolution	Processing time for 1frame
YOLOv3 [14]	800x600	0.050s on GPU
TS detector [11]	800x600	0.059s on GPU
D-Patches [19]	800x600	2.21s on CPU
ITSR [7]	800x600	0.061s on GPU
ATSR (ours)	800x600	0.063s on GPU

Fig. 9 Detection results for various weather noise conditions **a** clear day **b** cloudy day **c** snowy day **d** rainy day

slight noise, but it fails in dense noise conditions. For resolving this problem, an efficient rain removal algorithm is needed.

The proposed method is robust against rescaling as our detector has been trained on various sized traffic sign images. Thus, any change in the size of traffic signs is not likely to affect the detection ability. In Fig. 9b we can see that side posed traffic signs have been detected successfully. This shows that the size, pose, and the direction of traffic sign does not affect the detection performance.

5 Conclusion

Our new ATSR uses pre-processing to sharpen an input image and thus makes the small-sized and low-resolution traffic signs sensitive for detection. Grid size and anchor box optimization make the detector suitable for small objects at the cost of some false predictions. Fortunately, these false predictions can effectively be removed by using confidence score thresholding. The system is trained on self-developed traffic sign dataset, containing road images taken under various conditions, and tested on KTSD and GTSDB. The dataset contains three super classes of traffic signs, i.e., prohibitory class, mandatory class, and danger class. The system shows 100% recognition performance on GTSDB. The performance is compared in terms of Mean Average Precision (MAP) and ATSR significantly outperforms the other well-known traffic sign detectors. ATSR improved the MAP by 8.08% from the previous best published performance at the cost of longer computation time. (The computation time is increased from 0.050 s/f to 0.063 s/f.) on the challenging KTSD. We believe that our work will make a valuable contribution to the area of traffic sign detection.

Acknowledgements This material is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program (10080619).

References

1. Chen Y, Xie H, Shin H (2018) Multi-layer fusion techniques using a CNN for multispectral pedestrian detection. IET Comput Vis 12(8):1179–1187
2. Dollár P, Appel R, Belongie S, Perona P (2014) Fast feature pyramids for object detection. IEEE Trans Pattern Anal Mach Intell 36(8):1532–1545
3. Ellahyani A et al (2016) Traffic sign detection and recognition using features combination and random forests. Int J Adv Comput Sci Appl 7(1):683–693
4. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
5. Houben S et al (2013) Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. The 2013 international joint conference on neural networks (IJCNN). IEEE
6. Jameel H (2018) Dehazing road images for deep learning based traffic sign recognition. IJAERD 5(11)
7. Jameel K, Yeo D, Shin H (2018) New dark area sensitive tone mapping for deep learning based traffic sign recognition. Sensors 18(11):3776
8. Jia W et al (2019) Five-category classification of pathological brain images based on deep stacked sparse autoencoder. Multimed Tools Appl 78(4):4045–4064
9. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
10. Li D et al (2018) Deepsign: deep learning based traffic sign recognition. 2018 international joint conference on neural networks (IJCNN). IEEE
11. Manocha P, Kumar A, Khan JA, Shin H (2018) Korean Traffic Sign Detection Using Deep Learning. In: 2018 International SoC design conference (ISOCC), IEEE, pp 247–248

12. Mathias M et al (2013) Traffic sign recognition—How far are we from the solution?. The 2013 international joint conference on Neural networks (IJCNN). IEEE
13. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7263–7271
14. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767
15. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
16. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
17. Wang CY, Cheng-Yue R (2016) Traffic sign detection using you only look once framework. Standford, Tech Rep
18. Yang Y et al (2015) Towards real-time traffic sign detection and classification. IEEE Trans Intell Transp Syst 17(7):2022–2031
19. Yawar R et al (2017) D-patches: effective traffic sign detection with occlusion handling. IET Comput Vis 11(5):368–377
20. Zhang Y-D et al (2019) Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. Multimed Tools Appl 78(3):3613–3632
21. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S (2016) Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2110–2118

Jameel Ahmed Khan received his BS degree in electronics engineering from the University of Information Technology, Engineering and Management Sciences of Pakistan in 2008. Currently, He is working toward the master to PhD degree with the Department of electrical engineering, Hanyang University of Korea. His current research interests include computer vision, image processing, and pattern recognition.

Yunfan Chen received her BS degree in electronics engineering from the Wuhan University of Science and Technology of China in 2015. Currently, she is working toward the master to PhD degree with the Department of electrical engineering, Hanyang University of Korea. Her current research interests include computer vision, image processing, and pattern recognition.

Yawar Rehman is an associate professor at NED University, Pakistan. He received BS degree in electronics engineering from the Mehran University of Engineering and Technology of Pakistan in 2008, and a PhD degree in electrical engineering from the Hanyang University of Korea in 2017. His current research interests include computer vision, image processing, and pattern recognition.

Hyunchul Shin is a professor at Hanyang University, Korea. He received BS degree in electronics engineering from the Seoul National University in 1978, an MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology in 1980, and a PhD degree in electrical engineering and computer sciences from the University of California at Berkeley in 1987. He has more than 180 publications in journals and international conferences. His current research interests include computer vision, image processing, and pattern recognition.