CURE-TSR: Challenging Unreal and Real Environments for Traffic Sign Recognition

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we investigate the robustness of traffic sign recognition algorithms under challenging conditions. Existing datasets are limited in terms of their size and challenging condition coverage, which motivated us to generate the Challenging Unreal and Real Environments for Traffic Sign Recognition (CURE-TSR) dataset. It includes more than two million traffic sign images that are based on real-world and simulator data. We benchmark the performance of existing solutions in real-world scenarios and analyze the performance variation with respect to challenging conditions. We show that challenging conditions can decrease the performance of baseline methods significantly, especially if these challenging conditions result in loss or misplacement of spatial information. We also investigate the utilization of simulator data along with real-world data and show that hybrid training can enhance the average recognition performance in real-world scenarios.

1 Introduction

Autonomous vehicles are transforming existing transportation systems. As we step up the ladder of autonomy, more critical functions are performed by algorithms, which demands more robustness. In case of following traffic rules, robust sign recognition systems are essential unless we have prior information about traffic sign types and locations. It is a common practice to test the robustness of these systems with traffic datasets (1; 2; 3; 4; 5; 6; 7; 8; 9; 10). However, majority of these datasets are limited in terms of challenging environmental conditions. There is usually no metadata corresponding to challenge conditions or levels in these datasets, which are also limited in terms of dataset size. Moreover, the relationship between challenging conditions and algorithmic performance is not analyzed in these studies. Lu *et al.* (11) investigated the traffic sign detection performance with respect to challenging adversarial examples and showed that adversarial perturbations are effective only in specific situations. Das *et al.* (12) showed the vulnerabilities of existing systems and suggested JPEG compression to eliminate adversarial effects. Even though both of these studies analyze algorithmic performance variation with respect to specific challenging situations, adversarial examples are inherently different from realistic challenging scenarios.

In this paper, we investigate the traffic sign recognition performance of commonly used methods under realistic challenging conditions. To eliminate the shortcomings of existing datasets, we introduce the Challenging Unreal and Real Environments for Traffic Sign Recognition (CURE-TSR) dataset. The contributions of this paper are 5 folds. First, we introduce the most comprehensive publicly-available traffic sign recognition dataset with controlled challenging conditions. Second, we provide real-world data as well as simulator data, which can enable investigating transfer learning problem between real and simulated environments. Understanding the relationship between real and simulated environments can lead to realistic dataset design, which may eventually eliminate the need for real-world data collection. Third, we provide a benchmark of commonly used methods in the introduced dataset.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

Forth, we provide a comprehensive analysis of algorithmic performance with respect to challenging environmental conditions. Fifth, we utilize simulator data along with real-world data to enhance the performance of baseline methods in real-world scenarios.

2 Dataset

Timofte *et al.* (3) introduced the Belgium traffic sign classification (BelgiumTSC) dataset whose images were acquired with a van that had 8 roof-mounted cameras. Acquisition vehicle cruised in streets of Belgium and images were captured every meter. A subset of these images were selected and traffic signs were cropped to obtain the BelgiumTSC dataset. Stallkamp *et al.* (6; 7) introduced the German traffic sign recognition benchmark (GTSRB) dataset, which was acquired during daytime in Germany. Each traffic sign instance in the dataset is adjusted to have 30 images. BelgiumTSC and GTSRB datasets are limited in terms of challenging environmental conditions and they do not include metadata related to the type of challenging conditions or their levels. Because of limited control in data acquisition setup, it is not possible to perform controlled experiments with these datasets. The total number of annotated signs including BelgiumTSC and GTSRB datasets is around $60,000$, which may not be sufficient to test the robustness of recognition algorithms comprehensively. To compensate the shortcomings in the literature, we introduce the `CURE-TSR` dataset. Main characteristics of BelgiumTSC, GTSRB, and `CURE-TSR` datasets are summarized in Table 1.

Table 1: Main characteristics of BelgiumTSC, GTSRB, and CURE-TSR datasets.

Dataset	Number of images	Number of annotated images	Number of sign types	Sign size	Origin of the videos	Acquisition device
BelgiumTSC (13)	7,095 - 7,125	All images	62	11x10 to 562x438	Captured in Belgium	Color cameras
GTSRB (14)	133,000 - 144,769	51,840	43	15x15 to 250x250	Captured in Germany	Prosilica GC 1380CH color camera
CURE-TSR	2,206,106	All images	14	3x7 to 206x277	Captured in Belgium and Generated in Unreal Engine 4	Color cameras

(a) Real-world (real) image

(b) Simulator (unreal) image

Figure 1: Real and unreal environments.

Traffic sign images in the `CURE-TSR` dataset were cropped from the `CURE-TSD` dataset (15), which includes around 1.7 million real-world and simulator images. Real-world images were obtained from the BelgiumTS video sequences and simulated images were generated with the Unreal Engine 4 game development tool. In Fig. 1, we show a sample real-world image and a simulator image. In the rest of this paper, we refer to simulator generated images as unreal images and real-world images as real images. As observed in sample images, both real and unreal images are usually from urban environments. There are 14 traffic signs with annotations in both environments, which are shown in Fig. 2. Sign types include speed limit, goods vehicles, no overtaking, no stopping, no parking, stop, bicycle, hump, no left, no right, priority to, no entry, yield, and parking.

Figure 2: Traffic signs in real (1^{st} row) and unreal (2^{nd} row) environments.

Unreal and real sequences were processed with state-of-the-art visual effect software Adobe(c) After Effects to simulate challenging conditions, which include rain, snow, haze, shadow, darkness, brightness, blurriness, dirtiness, colorlessness, sensor and codec errors. In Fig. 3, we show sample stop sign images under challenging conditions in both real and unreal environments.

Figure 3: Stop signs under challenging conditions in real (1^{st} row) and unreal (2^{nd} row) environments.

3 Experiments

3.1 Baseline Methods, Dataset, and Performance Metric

In the German traffic sign recognition benchmark (GTSRB) (6), histogram of oriented gradient (HOG) features were utilized to report the baseline results. In the Belgium traffic sign classification (BelgiumTSC) benchmark, cropped traffic sign images were converted into grayscale and rescaled to 28×28 patches, which were included in the baseline. Moreover, HoG features were also used as a baseline method. They classified traffic sign images with methods including support vector machines (SVMs). Similar to GTSRB and BelgiumTS datasets, we use rescaled grayscale and color images as well as HoG features as baseline. In the final classification stage, we utilize one-vs-all SVMs with radial basis kernels and softmax classifiers. In addition to aforementioned techniques, we also use a shallow convolutional neural network, which consists of two convolutional layers followed by two fully connected layers, and a softmax classifier. We preprocessed images using l_2 normalization, mean subtraction, and division by standard deviation.

Traffic sign images originate from 49 video sequences, which were split into approximately 70% training set and 30% test set. Video sequences were split one sign at a time, starting from the least common sign. Once video sequences were assigned to training or testing sets, splitting continued from the remaining sequences until all the sequences were classified. In the first experiment set, we utilized $7,292$ traffic sign images in the training stage obtained from challenge-free real training sequences. In the testing, we utilized $3,334$ images from each challenge category and level, which adds up to $200,040$ images ($3,334$ images $\times 12$ challenge types $\times 5$ levels). As performance metric, we utilized classification accuracy, which corresponds to the percentage of traffic signs that are correctly classified.

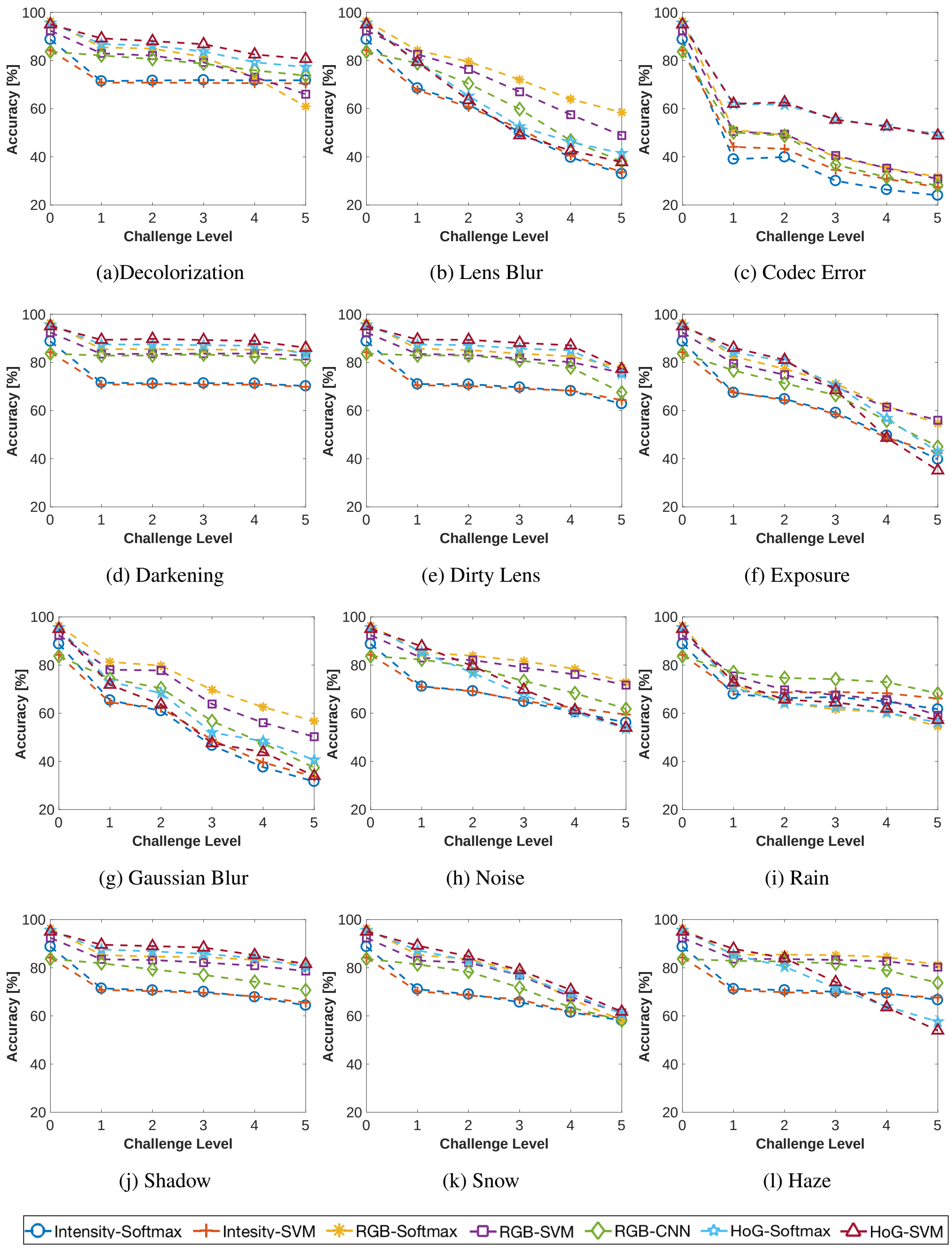

Figure 4: Performance versus challenge levels.

3.2 Experiment 1: Recognition in Real Environments under Challenging Conditions

We analyzed the accuracy of baseline methods with respect to challenge levels for each challenge type and report the results in Fig. 4. Severe decolorization (Fig. 4(a)) leads to at least 10% decrease in accuracy, which is less compared to majority of other challenge categories because remaining information is still sufficient for shape-based recognition. Among all the challenges, codec error is the most effective category that significantly degrades the classification accuracy even with challenge level 1 as shown in Fig. 4(c). We can observe that there is at least 30% decrease for each method after challenge level 1 and at least 46% decrease after challenge level 5. Lens blur (Fig. 4(b)), exposure (Fig. 4(f)), and Guassian blur (Fig. 4(g)) result in significant performance decrease under severe challenging conditions, at least 36% for each baseline method. However, classification accuracy decreases more linearly in these categories compared to codec error because of its steep decrease in level 1. In darkening category (Fig. 4(d)), classification accuracy decreases at least 5% in challenge level 1 for all the methods other than CNN. The normalization operation in convolutional model makes it less sensitive to darkening challenge. When challenge level becomes more severe, performance of baseline methods degrades a few percent at most.

In dirty lens category (Fig. 4(e)), new dirty lens images were overlayed on entire images to increase the challenge level. And, the new dirt patterns do not necessarily occlude traffic signs. Therefore, performance of baseline methods do not always change when challenge level increases. In noise category (Fig. 4(h)), HoG and CNN correspond to a more linear performance decrease compared to intensity and color-based methods, whose performance decreases are steeper for level 1 challenge. In rain category (Fig. 4(i)), particle models are all around the scene, which result in significant occlusion even in level 1 challenge. Therefore, degradation while going from challenge-free to level 1 challenge is steeper than any further relative changes. In shadow category (Fig. 4(j)), vertical shadow lines are all over the images, which lead to relatively steep performance decrease for challenge level 1. We observe slight degradation as challenge level increases because areas under shadow become less visible. In case of snow challenge (Fig. 4(k)), intensity-based methods result in a more significant decrease compared to other methods for level 1 challenge but all methods converge to a similar classification accuracy under severe snow challenge. In haze category (Fig. 4(l)), performance of intensity-based models decrease steeply for level 1 challenge whereas decrease in HoG-based models follow a more linear behavior. Color image-based classifiers and CNN are less sensitive to haze challenge compared to other methods. Haze challenge was generated as a combination of radial gradient operator with partial opacity, a smoothing operator, an exposure operator, a brightness operator, and a contrast operator. Moreover, the location of the operator was adjusted manually per frame to simulate a sense of depth. Because of the complexity of haze model, it is less intuitive to explain the behavior of baseline methods. However, the higher tolerance of CNN model with respect to haze challenge can be explained with its capability to directly learn spatial patterns from visual representations.

3.3 Experiment 2: Recognition in Real Environments under Challenging Conditions with the Help of Challenging Unreal Environments

In Section 3.2, we analyzed the performance of baseline methods with respect to challenging conditions and level. Baseline methods were trained with $7,292$ real-world images and a total of $200,040$ images were used in testing. In this section, we investigate the performance of baseline methods when unreal images are used in the training in addition to real images. Test set is same as experiment 1 but we extended the training set with 20 unreal images for each traffic sign from challenge level 5 sequences. We selected the traffic signs with maximum area to obtain highest resolution samples. Overall, training set includes $3,084$ unreal images (20 images $\times 11$ challenge types $\times 14$ traffic signs) and $7,292$ real-world images. We compared the performance of baseline methods that are trained with and without unreal images and report the performance change in Table 2. Each entry in the table other than the last row and the last column was obtained by calculating the performance change for a baseline method over all the challenge levels for a specific challenge type. Entries in the last row were calculated by averaging the performance change of each baseline method over all challenge types. Finally, entries in the last column were calculated by averaging the performance change over all baseline methods for each challenge type.

Table 2: Classification accuracy change (%) when additional unreal images used in the training.

Challenge Types	Baseline Methods							
	Intensity		Color		HoG		CNN	Average
	Softmax	SVM	Softmax	SVM	Softmax	SVM		
Decolorization	+2.86	+3.32	+1.46	-0.53	+1.43	-0.01	+3.23	+1.68
Lens Blur	+3.98	+2.71	+4.45	+6.60	+3.34	+1.81	-1.78	+3.02
Codec Error	+0.47	-1.21	+1.51	-0.82	-1.55	-1.61	+2.40	-0.12
Darkening	+2.83	+2.98	+2.87	+1.44	+1.68	+0.44	+2.58	+2.12
Dirty lens	+3.14	+2.86	+2.68	+1.63	+2.00	+0.62	+3.11	+2.29
Exposure	+2.54	+1.77	+1.34	+1.97	-0.66	-2.23	+0.54	+0.75
Gaussian Blur	+5.89	+3.98	+4.24	+7.06	+2.03	+1.77	+2.78	+3.97
Noise	+1.62	+1.58	+1.89	+0.58	+1.41	-0.90	+2.25	+1.21
Rain	+2.30	+1.28	+4.73	+2.75	+5.48	+2.34	+0.69	+2.80
Shadow	+2.95	+3.38	+3.27	+1.62	+1.73	+0.64	+3.01	+2.37
Snow	+3.19	+2.81	+2.09	+0.48	+2.63	+0.92	+4.34	+2.35
Haze	+3.28	+3.22	+3.22	+1.41	+2.26	-1.35	+3.51	+2.22
All (average)	+2.92	+2.39	+2.81	+2.02	+1.81	+0.20	+2.22	-

We tested 7 baseline methods over 12 challenge types and report the performance change of each baseline method for each challenge type. Out of 84 result categories (7 baseline methods $\times 12$ challenge types), classification performance increased in 72 of them. On average, classification performance increased for all challenge types other than a slight decrease in codec error. Moreover, average classification performance increased for each baseline method, which is a slight increase for HoG-SVM (0.2%) and more for other methods (at least 1.81%). Additional unreal images in the training set were obtained from all the challenge types except haze category. However, classification accuracy increased for all the baseline methods at least 1.41% other than HoG-SVM in haze category. The performance enhancement in haze can be understood by analyzing the computational model of haze and its perceptual similarity to other challenges. Haze model includes a smoothing operator, an exposure filter, a brightness operator, and a contrast operator. Exposure filter is used in the exposure (overexposure) model and smoothing operator is utilized in blur models. Moreover, perceptually, we can observe similarities between haze and blur challenges in terms of smoothness and similarities between haze and exposure in terms of washed out details. Therefore, perceptually and computationally similar challenges in the training stage can affect the performance of each other in the testing stage.

4 Conclusion

We introduced the `CURE-TSR` dataset, which is the most comprehensive traffic sign recognition dataset in the literature that includes controlled challenging conditions. We provided a benchmark of commonly used methods in the `CURE-TSR` dataset and reported that challenging conditions leads to severe performance degradation for all baseline methods. We have shown that lens blur, exposure, Gaussian blur, and codec error degrade recognition performance more significantly compared to other challenge types because these challenge categories directly result in losing or misplacing shape-related information. In addition to training and testing data-driven methods with real-world data, we also utilized simulator images in the training and reported performance enhancement for most of the baseline methods and challenge categories in real-world scenarios.

References

[1] C. Grigorescu and N. Petkov, "Distance sets for shape filters and shape recognition," *IEEE Trans. Image Proces.*, vol. 12, no. 10, pp. 1274–1286, Oct 2003.

[2] R. Timofte, K. Zimmermann, and L. V. Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," in *WACV*, Dec 2009, pp. 1–8.

[3] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," *Mach. Vis. App.*, vol. 25, no. 3, pp. 633–647, 2014.

[4] R. Belaroussi, P. Foucher, J. P. Tarel, B. Soheilian, P. Charbonnier, and N. Paparoditis, "Road sign detection in images: A case study," in *Proc. ICPR*, Aug 2010, pp. 484–488.

[5] F. Larsson and M. ' Felsberg, "Using fourier descriptors and spatial models for traffic sign recognition," in *Proc. SCIA*, Berlin, Heidelberg, 2011, SCIA'11, pp. 238–249, Springer-Verlag.

[6] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *Proc. IJCNN*, July 2011, pp. 1453–1460.

[7] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man versus computer: Benchmarking machine learning algorithms for traffic sign recognition ," *Neural Networks*, vol. 32, pp. 323 – 332, 2012, Selected Papers from IJCNN 2011.

[8] 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. IJCNN*, Aug 2013, pp. 1–8.

[9] 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.

[10] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. IEEE CVPR*, June 2016, pp. 2110–2118.

[11] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "NO need to worry about adversarial examples in object detection in autonomous vehicles," in *arXiv:1707.03501*, 2017.

[12] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, "Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression," in *arXiv:1705.02900*, 2017.

[13] R. Timofte, K. Zimmermann, and L. V. Gool, "Belgium traffic sign dataset," `http://btsd.ethz.ch/shareddata/`.

[14] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "German traffic sign recognition and detection benchmarks," http://benchmark.ini.rub.de/.

[15] "CURE-TSD: Challenging unreal and real environments for traffic sign detection," `https://ghassanalregib.com/cure-tsd/`.