Rare Traffic Sign Recognition using Synthetic Training Data

Vlad Shakhuro
Lomonosov Moscow State University
Office 701, Leninskiye Gory 1-52, Moscow, Russia
+7 (495) 939-01-90
vlad.shakhuro@graphics.cs.msu.ru

Boris Faizov
Lomonosov Moscow State University
Office 701, Leninskiye Gory 1-52, Moscow, Russia
+7 (495) 939-01-90
boris.faizov@graphics.cs.msu.ru

Anton Konushin
Lomonosov Moscow State University
Office 77, Leninskiye Gory 1-52, Moscow, Russia
+7 (495) 939-01-90
NRU Higher School of Economics
Office S835, 11 Pokrovsky Bulvar, Moscow, Russia
Samsung AI Center
5c Lesnaya str., Moscow, Russia
anton.konushin@graphics.cs.msu.ru

ABSTRACT
Modern computer vision methods usually require lots of labelled data for training. Besides price of labelling, problems with rare object classes and adaptation to new domain or task arise. One of the promising methods to solve these problems is to generate synthetic training data. In this work we focus on task of traffic sign detection. We consider several methods for generating synthetic data for training traffic sign detectors: random placement of signs of different quality (simple synthetic, CGI based and CGI improved using generative adversarial network). We also propose a method to replace real signs with synthetic signs. Experimental evaluation shows that proposed method improves quality of detection of rare traffic signs and that usage of synthetic data is very helpful for improving training of traffic sign classifier.

CCS Concepts
• **Computing methodologies →Object recognition**

Keywords
Traffic sign recognition; synthetic data; generative adversarial networks.

1. INTRODUCTION
Modern computer vision methods based on deep neural networks require lots of training data. To fulfill this requirement, researchers and companies collect thousands or even millions of images and use crowdsourcing services to label these images. Size of final dataset depends on labelling difficulty: ImageNet dataset for multiclass classification contains 14 million images while Cityscapes semantic segmentation dataset contains 5 thousand images. Even after labelling these datasets several problems remain unsolved:

1. Labelled dataset may be insufficient for solving same task in different conditions. For instance, algorithm trained on Cityscapes dataset (which is captured in Germany) will perform poorly on image captured in China.
2. Object classes are usually imbalanced. There may exist rare and frequent classes of objects. Even in large datasets number of examples of rare classes may be too small to train neural network on them.
3. Datasets have to be updated if new class of object appears.

One of the promising ways to solve listed problems is usage of synthetic training data.

Consider traffic sign recognition task. In this task one has to detect and classify all traffic signs in an image shot from car camera. Let's consider modern dataset for traffic sign recognition called RTSD [11] (Russian Traffic Sign Dataset) has 205 classes in testing part and only 106 classes in training part. As Fig. 1 also shows, number of images per class is highly imbalanced. To train traffic sign recognition system properly, we aim to solve these two problems using synthetic training data. In this work we explore several methods for generating synthetic training data. We start with random placement of traffic signs. We consider simple synthetic data, CGI data rendered using modern ray tracing engine and CGI data improved with cycle-consistent generative adversarial network. Baseline methods place traffic signs in random location of existing background frame. We evaluate these methods and propose an advanced method for generating synthetic traffic sign data. Proposed method consists of two stages: inpainting of existing real traffic sign and placement of synthetic one (maybe of different class). As our evaluation shows, such method is more appropriate for training detector since it generates traffic signs in plausible positions. In conclusion, main contribution of our work is analysis and comparison of several synthetic data generation methods for training traffic sign detectors and classifiers.

ICVIP 2019, December 20-23, 2019, Shanghai, China

ACM ISBN 978-1-4503-7682-2/19/12…$15.00
http://dx.doi.org/10.1145/3376067.3376105

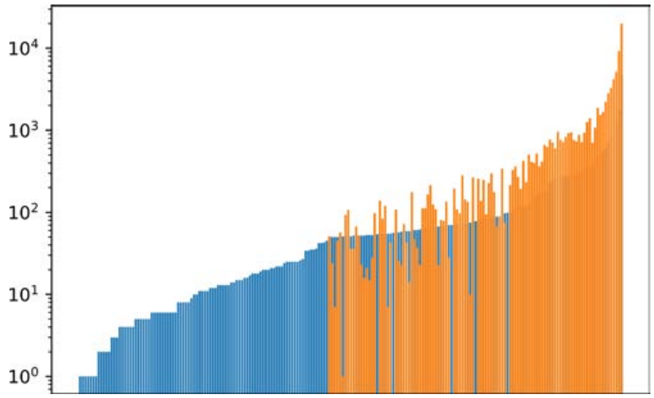

Figure 1. Number of images per class in training (orange columns) and testing (blue columns) parts of RTSD.

2. RELATED WORK

Synthetic images are frequently used for training data-hungry computer vision algorithms in tasks where manual labelling of images is prohibitory expensive. For instance, modern image segmentation dataset Cytiscapes [3] consists of 5000 images. Annotation of each image required 1.5 hours on average per single image. One can divide algorithms for generating synthetic data in to groups: data augmentation and 3D modelling.

Data augmentation methods are very popular for training deep neural networks. In [9] number of training samples is increased by several orders of magnitude using random image crops and horizontal flips. In [2] images of traffic signs are augmented using random rotations, shifts and scalings. Modern data augmentation methods [15,4,13] behave similar to regularization [12] and mix existing real images and their labels with random weights.

3D modelling is used actively in applications which demand highly realistic images. In [10,1] game engine is used to render realistic street scenes for semantic segmentation and detection of cars. Unfortunately, quality of rendering is insufficient for training computer vision algorithms to achieve quality similar to algorithms trained on real data. For this reason, synthetic data is usually mixed with real data for training. Such mix improves algorithm quality in comparison to algorithm trained only on synthetic data. Another downside of such data is that quality of rendered data depends on quality of 3D models and materials. Models and materials of high quality in enough amount for data variability may be prohibitively expensive to create or collect.

However, in several works it is demonstrated that even implausible synthetic training dataset can be sufficient for training good models. For example, in [5] dataset "Flying chairs" is used to train deep neural network to predict optical flow between two frames. Neural network has to learn to compare similarity of image area, thus training sample may be non-realistic.

3. GENERATING SYNTHETIC IMAGES OF TRAFFIC SIGNES

3.1 Random placement of traffic signs

In first three methods we place traffic signs randomly in the frame. We call these methods Synt (simple synthetic using icons), CGI (computer-generated imagery) and CGI-GAN (CGI improved with CycleGAN).

3.1.1 Simple synthetic

To obtain simple synthetic data, we take traffic sign icons and apply transformations with random parameters to them. Transformations are: linear correction in HSV color space, gaussian blur, motion blur, rotation. Transformed icons are placed in the same places as CGI signs. Example of frame with simple synthetic data is shown in Fig. 2.

Figure 2. An example of background frame with placed Synt signs.

3.1.2 CGI (computer-generated imagery)

To obtain second type of synthetic data, we use Hydra Renderer [7] and realistic calibration matrices to insert traffic signs on poles. To obtain traffic sign model, we place its' icon on template models of different shapes (triangle, circle, etc.). Traffic signs are placed on random RTSD frames without any real traffic signs. Example of such frame is shown in Fig. 3.

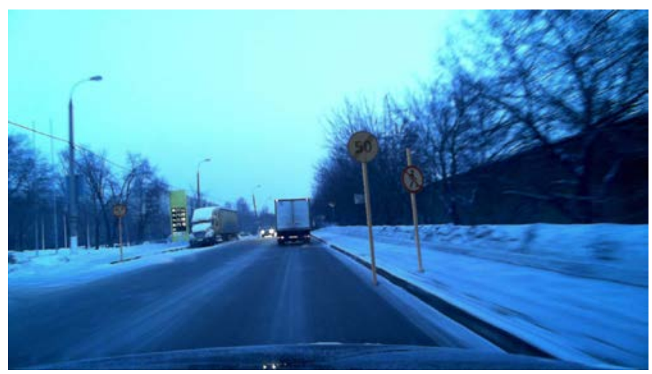

Figure 3. An example of background frame with placed CGI signs.

3.1.3 Improved with CycleGAN CGI synthetic (CGI-GAN)

Improved version of CGI synthetic is obtained using CycleGAN [16] generator trained in the previous year of project. This CycleGAN generator is trained to make image of traffic sign more realistic while preserving class of traffic sign. We take CGI frames, crop rectangles with signs, rescale them to 128×128 resolution, apply generator on them and resize images back to the original scale. Lastly, we insert traffic sign using mask from the original CGI synthetic. Example of frame with such traffic signs is shown in Fig. 4.

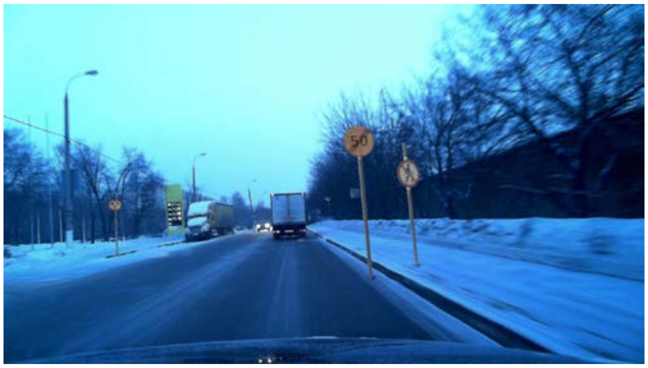

Figure 4. An example of background frame with placed CGI-GAN signs Traffic sign insertion.

3.2 Replacement of original sign with synthetic

In real images places of traffic signs are not random. Therefore, we propose more advanced method for generating training data. We assume that detector analyzes big enough neighborhood of traffic sign and the location of traffic sign has to be very realistic. Random placement isn't such realistic. To obtain real locations, we take existing frames with labelled traffic signs. Labelled real traffic signs are inpainted using encoder-decoder neural network that is trained to inpaint background with Wasserstein GAN approach [6]. Then we place Synt signs in place of the inpainted sign. An example of original frame, frame with inpainted traffic sign and frame with inserted Synt sign is shown in Fig. 5.

Figure 5. An example of traffic sign inpainting and insertion.

4. EVALUATION

We use PVANet [8] as object detector and two traffic sign classifiers based on WideResNet [14].

First classifier is trained WideResNet model with k = 2 and depth = 8. It takes image of size 64×64 pixels and predict one of the 205 sign classes.

Second classifier is specially designed for highly imbalanced traffic sign dataset. It uses WideResNet to extract features for classification. Features are then used in Random Forest to classify whether sign is rare (i.e. is not in the RTSD training part) or frequent. If the sign is frequent, it is classified with Softmax layer on top of features. If the sign is rare, it is passed into k-NN classifier which operates on index which consists of CGI-GAN traffic signs. Our experiments show that such classifier shows better quality compared to the first classifier.

We use AUC (Area Under Curve) to measure detector quality. Detector results are shown in Table 1.

Table 1. Detector results for different training samples.

	AUC		
Training sample	all	freq	rare
Real	0.8908	0.8920	0.8602
Synt	0.1390	0.1385	0.1483
CGI	0.1070	0.1063	0.1323
Inpaint	0.5523	0.5626	0.5526
Real + Synt	0.8848	0.8862	0.8554
Real + CGI	0.8856	0.8872	0.8572
Real + CGI-GAN	0.8853	0.8869	0.8521
Real + Inpaint	0.8861	0.8871	0.8663

We can see that:

1. Detector trained only on frequent traffic sign classes also is able to find rare traffic signs. We can conclude that detector learns to find general notion of traffic sign (for instance, circle with red border) and if rare sign is similar to already seen classes, detector will be able to find it.
2. Random placement of synthetic traffic signs in training sample doesn't lead to sufficient quality of detection. Inpaint signs compared to Synt, CGI show better quality.
3. Adding synthetic data to real data worsens quality of detector. The only exception is usage of Inpaint data that slightly improves rare sign detection, but at cost of slightly lower performance on frequent classes

Detector and classifier results with neural net classifier are shown in Table 2. Detector and classifier results with two-way classifier are shown in Table 3.

Table 2. Detector and neural net classifier results for different training samples.

	AUC		
Training sample	all	freq	rare
Real	0.7544	0.8246	0.0909
Synt	0.0997	0.0999	0.0909
CGI	0.0881	0.0881	0.0853
Inpaint	0.1589	0.1596	0.1379
Real + Synt	0.8022	0.8438	0.3396
Real + CGI	0.8384	0.8515	0.4851

Real + CGI-GAN	0.8445	0.8568	0.4983
Real + Inpaint	0.7641	0.8293	0.3400

Table 3. Detector and two-way classifier results for different training samples.

	AUC		
Training sample	all	freq	rare
Real	0.8606	0.8673	0.5896
Real + CGI-GAN	0.8514	0.8589	0.5857
Real + Inpaint	0.8440	0.8528	0.5942
Real + CGI + Inpaint	0.8330	0.8420	0.5894

We can conclude that:

1. All synthetic data improve classification quality on frequent and rare data. CGI-GAN-improved signs achieve best quality.
2. Two-way classifier with kNN classifier for rare traffic signs on CGI-GAN index substantially improves quality of detection even with detector trained on only Real data. Best quality on rare traffic signs is achieved using Real and Inpaint data.

5. CONCLUSION

In this work we considered generation of synthetic data for training traffic sign detectors and classifiers. Our experiments show that main problem in traffic sign recognition consists in rare traffic sign classification. We proposed and evaluated several methods for generating synthetic data. It should be noted that usage of synthetic data improves accuracy of regular neural network classifier, but two-way classifier specially designed for rare traffic sign classification achieves best quality in pair with traffic sign detector.

6. REFERENCES

[1] Alhaija, H.A., Mustikovela, S.K., Mescheder, L., Geiger, A. and Rother, C. 2018. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision, 126(9). 961-972.

[2] Cireşan, D., Meier, U., and Schmidhuber, J. 2012. Multi-column deep neural networks for image classification. arXiv:1202.2745. Retrieved from https://arxiv.org/abs/1202.2745

[3] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition. 3213-3223.

[4] DeVries, T. and Taylor, G.W. 2017. Improved regularization of convolutional neural networks with cutout. arXiv:1708.04552. Retrieved from https://arxiv.org/abs/1708.04552

[5] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D. and Brox, T. 2015. Flownet: Learning optical flow with convolutional networks. In Proceedings of the *IEEE international conference on computer vision*. 2758-2766.

[6] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A.C. 2017. Improved training of wasserstein gans. In *Advances in neural information processing systems*. 5767-5777.

[7] Hydra Renderer. Retrieved from https://github.com/Ray-Tracing-Systems/HydraAPI

[8] Kim, K. H., Hong, S., Roh, B., Cheon, Y., and Park, M. 2016. Pvanet: Deep but lightweight neural networks for real-time object detection. arXiv:1608.08021. Retrieved from https://arxiv.org/abs/1608.08021

[9] Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097-1105.

[10] Richter, S. R., Vineet, V., Roth, S., and Koltun, V. 2016. Playing for data: Ground truth from computer games. In *European conference on computer vision*. Springer, Cham, 102-118.

[11] Shakhuro, V.I. and Konouchine, A.S. 2016. Russian traffic sign images dataset. Computer Optics, 40(2).294-300.

[12] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1). 1929-1958.

[13] Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J. and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. arXiv:1905.04899. Retrieved from https://arxiv.org/abs/1905.04899

[14] Zagoruyko, S., and Komodakis, N. 2016. Wide residual networks. arXiv:1605.07146. Retrieved from https://arxiv.org/abs/1605.07146

[15] Zhang, H., Cisse, M., Dauphin, Y.N. and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. arXiv:1710.09412. Retrieved from https://arxiv.org/abs/1710.09412

[16] Zhu, J.Y., Park, T., Isola, P. and Efros, A.A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision. 2223-2232.