A SYSTEM FOR LARGE-SCALE AUTOMATIC TRAFFIC SIGN RECOGNITION AND MAPPING

Alexander Chigorin and Anton Konushin

Graphics & Media Lab, Lomonosov Moscow State University, Russia
{aachigorin, ktosh}@graphics.cs.msu.ru

KEY WORDS: Mapping, Surveying, Vision, Recognition, Application, Automation

ABSTRACT:

We present a system for the large-scale automatic traffic signs recognition and mapping and experimentally justify design choices made for different components of the system. Our system works with more than 140 different classes of traffic signs and does not require labor-intensive labelling of a large amount of training data due to the training on synthetically generated images. We evaluated our system on the large dataset of Russian traffic signs and made this dataset publically available to encourage future comparison.

1. INTRODUCTION

Traffic signs are made to be visible and easily distinguishable by humans and that makes them good objects for processing by automatic algorithms. However these algorithms should be able to cope with such problems as intra-class variability, lighting changes, viewpoint changes, occlusions and blur. Modern state-of-the-art traffic sign recognition algorithms rely on machine learning techniques (Stallkamp et al., 2012) and require a large and representative training set to overcome all these problems. To obtain such a training set human operators need to process images from hundreds of thousands kilometres of roads. This is especially hard for some rarely occurring sign classes. Considering the fact that different countries often have different signs this process should be repeated for every new set of signs. To overcome this problem we evaluate the possibility to train on synthetically generated data.

The goal of our system is to take a sequence of geo-located images and produce a map with all encountered traffic signs on it. Contributions of this paper can be summarized as follows:

1. We present practically useful automatic traffic signs recognition and mapping system that requires only one mounted camera with a GPS module.
2. Our system works with more than 140 classes of signs. It has false positive rate that are low enough to use it in practical applications.
3. We systematically explore the possibility to train on synthetically generated data.
4. To measure the accuracy of our system we have constructed a large dataset of Russian traffic signs and made it publically available to encourage future comparison.

2. RELATED WORK

There were several works in the literature that describe large-scale traffic sign recognition systems. In (Baro et al., 2009) a system based on an attention cascade is trained with Adaboost on dissociated dipoles (Balas et al., 2003). It is able to recognize four different types of signs grouped by visual similarity (*danger*, *yield*, *prohibition*, *command*). 50-60% hit ratio (counting on per-image basis) is achieved for monocular video with the false detection encountered every 13-52 frames. Traffic sign recognition performance was measured for a dataset consisting of 2000 training and 600 testing images of 31 classes, grouped into three types - *speed*, *circular* and *triangular*, with recognition rates equal to 91%, 98% and 99% accordingly.

Another multi-view traffic sign recognition system is described in (Timofte et al., 2009). It was trained to detect and recognize 62 classes of signs. In a single-view evaluation it achieved 96.8% detection rate with 2 false alarms per 2MP image. Such a big false positive rate was reduced with the use of information from multiple views that allowed achieving 95.3% physical signs detection rate with one false alarm per 6000 images.

(Mathias et al., 2013) used soft cascade with channel features and presented detection results on two publically available datasets of German and Belgium traffic signs consisting of 43 and 62 classes accordingly. They reach 99% average AUC (area under precision/recall curves) on German dataset and 92.56% average AUC on Belgium dataset. But false positive counting was performed on relatively small number of images: 300 on German and 583 on Belgium dataset.

In (Overett et al., 2011) hardware-oriented implementation of HOG (Dalal et al., 2005) features allowed building a high-throughput system with 99% detection rate and 10^{-10} false negatives rate per detector window. This work is the most similar to ours, but we are presenting results for more than 140 different classes of traffic signs (instead of 3 as in (Overett et al, 2011)) and show that some classes of signs are more difficult to detect than others. For example, blue squared signs are harder than red circles because of color-similarity with the sky and lack of easily distinguishable border. We also systematically explore the possibility of training on synthetically generated data and present results of the evaluation for a full system consisting of detector, recognizer and linker.

Another key aspect of our work is in the usage of synthetically generated training data. It was successfully used in many applications, such as: human pose recognition (Shotton et al., 2011), object 3D structure inferring (Grauman et al., 2003), shape models learning (Stark et al., 2010), pedestrian detection (Marin et al., 2010) (Pishchulin et al., 2011) (Enzweiler et al., 2008), viewpoint-independent object detection (Liebelt et al., 2008), text recognition (Wang et al., 2011) and keypoints recognition (Ozuysal et al., 2007).

There were also some attempts to use pictograms and synthetic data for training. (Møgelmose et al., 2012) compare performance of sign detectors trained on real and synthetic data and come to the conclusion that a detector trained on real data performs better. The difference in hit rate was about 50%. For our system the difference between real and synthetic data is not so big and is equal to 10% on last stages of cascade with 10^{-9} false positive rate per detection window. (Larsson et al., 2011) used Fourier descriptors to describe a sign and a new correlation-based similarity measure to compare it with a prototype pictogram. In (Ihara et al., 2009) SIFT-keypoints from the input sample were matched with keypoints from the sign pictogram. (Paulo et al., 2009) used curvature scale space representation to describe image contours and match them with the prototype pictogram. (Arlicot et al., 2009) proposed a method for circular signs detection, based on color pre-filtering and ellipse detection using RANSAC algorithm. Detected sign verification was performed using Zero Mean Normalized Cross Correlation with pictograms of reference traffic signs.

3. RUSSIAN TRAFFIC SIGNS DATASET (RTSD)

RTSD[1] consists of 9508 images with signs and 71050 background images. It contains 14360 sign bounding boxes, 6387 of which are also labelled with a physical sign id. There are 863 labelled physical signs, thus each physical sign is encountered on average 7.3 times. The dataset is divided into training and test part. Training part contains 4754 images with signs and 44817 background images. The remaining images are in the test part.

4. SYSTEM OVERVIEW

We are working with a mobile mapping system that consists of one camera with a GPS-module. Our camera is capable to produce five 0.9 Mpix images per second. Below we have defined requirements that should be met in an automatic traffic signs mapping system to make it practically useful.

The main parameter that matters in the signs mapping system is physical sign detection rate. This means that a sign could be detected only on one image out of several that actually contain it. So there is no need for detector to have very high detection rate per image. In RTSD each physical sign is seen on average on 7.3 images. And the majority of signs are seen at least 3 times. This leads to 70% per image detection rate requirement that in theory should allow detecting 97.3% of physical signs. The false positive rate should be lesser than 10^{-9} detections per detector window. In our case it means that wrong detection would be encountered every 150 frames. This seems quite frequent at first but this number could be greatly reduced on subsequent stages of the system. For example, during linker stage all false positive detections from neighboring frames would be linked together. It is also possible to take into account recognizer's confidence to filter out wrong detections. Similar requirement on false positive rate was also defined in (Overett et al., 2011).

In this paper we present results for four different classes of Russian traffic signs, grouped by visual similarity (see Figure 1). They include more than 140 different classes, some of which differ only in a few pixels.

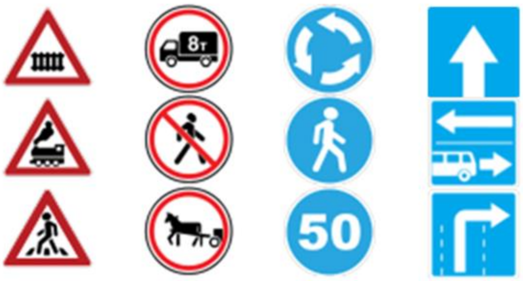

Figure 1: Column-wise examples of traffic sign types (red triangles, red circles, blue circles, blue squares) processed by our system.

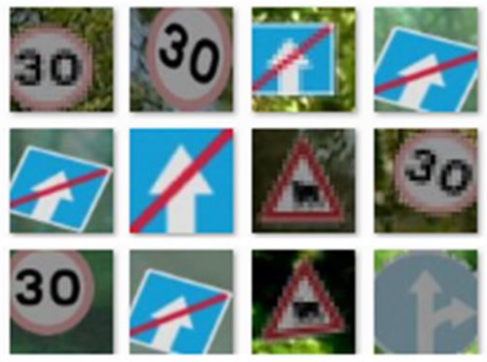

Figure 2: Examples of synthetically generated samples.

Our system consists of four different modules described below:

1. **Detector.** Finds traffic signs on each incoming image.
2. **Recognizer.** Assigns a class label with confidence to each detected traffic sign.
3. **Linker.** Links bounding boxes between neighboring frames producing the new physical sign and assigns a class label to it.
4. **Localizer**. Calculates position of the physical sign in world coordinates using camera internal and external parameters and information from linked bounding boxes.

5. SYNTHETIC DATA GENERATION

A traffic sign is a rigid planar object. That makes generation of synthetic sign images relatively easy. We applied a series of transformations to pictograms[2] of traffic signs from Wikipedia (some examples are depicted in Figure 1) with intent to generate visually appealing synthetic images (see Figure 2).

Here we present the full list of transformations applied to the incoming pictogram:

1. Variation of value (V) and hue (H) components of HSV color space.
2. Rotation, scaling and shifting of a sign in the 3D space.
3. Addition of Gaussian noise, "salt and pepper" noise and "pixelization effect".
4. Sign image blurring.
5. Addition of background from real images with blending on sign edges.

Each transformation is parameterized by its probability of occurrence and, in case of occurrence, probability distributions for each parameter of the transformation.

[1] Dataset could be found at ftp://anonymous@kiviuq.gml-team.ru/AnonymousFTP/RTSD/

[2] Pictograms cold be found at http://yadi.sk/d/juSX-WSe6L6Oi

6. TRAFFIC SIGNS DETECTION

Here we present a brief description of the architecture of the detector module and then justify different design choices. As in (Baro et al., 2009), (Timofte et al., 2009) and (Overett et al., 2011) we use an attention cascade to make a detection problem tractable. First stages of our cascade consist of AdaBoost classifiers trained on dissociated dipoles features (Balas et al., 2003) that were also used in (Baro et al., 2009). We preprocess sign images as described in (Ruta et al., 2011) with filters that intend to amplify certain colors and suppress any others (for example, we amplify blue color in the case of blue squares sign type). On each stage we use for training 10000 synthetic samples of signs and 16000 background patches bootstrapped from real images. Thirteen dipoles-stages are trained until false positive rate become lesser than 10^{-7}.

After that we train a deep convolutional neural network[3] on a much larger synthetically generated dataset of 200000 positive samples and 16000 false positives from real images. Neural network is a good choice for training on synthetically generated data due to its ability to be trained on a datasets of a large size. In our case we increased the size of the training set until recall of the detector has stopped changing.

Our network has 5 layers with weights and 2 max-pooling layers. It takes as input 30x30 images normalized with histogram equalization. First two layer of the network is convolutional with 64 kernels of size 5x5 with a stride of 1 pixel. Each convolutional layer is followed by the max-pooling layer with size 3x3 and stride of 2 pixels. Second pooling layer is followed by two locally-connected layers with size of 3x3 and stride of 1 pixel. The last layer is fully-connected and its output is fed into 2-way softmax which produces a distribution over 2 class labels. Network hyper-parameters were selected via cross-validation. We repeat bootstrapping rounds until false detections rate become lesser than 10^{-9}. We train four detectors for each traffic sign type (as depicted in Figure 1).

Here we present results of several experiments to justify design choices made in the detector module.

6.1 Usage of color filters described in (Ruta et al., 2011) that amplify specific colors.

We trained two detector cascades for the blue squares sign type. First cascade was trained on grayscale images, another cascade on images with amplified blue color as in (Ruta et al., 2011). It is clear from Figure 3 and Figure 4 that color features are doing better job in terms of accuracy and the number of features.

6.2 Training cascade on dissociated dipoles features till false positive rate is lesser than 10^{-7} and training a deep neural network after this point.

Dissociated dipoles are fast and lightweight yet powerful features. We use these features on the first stages of the cascade to accelerate its performance because almost all detectors' work is done on these stages. But our experiments showed that the accuracy of the cascade trained on dipoles features is starting to drop rapidly after the point where false positives rate is equal to 10^{-7} (see Figure 5 and Figure 6). Possible solution is to use

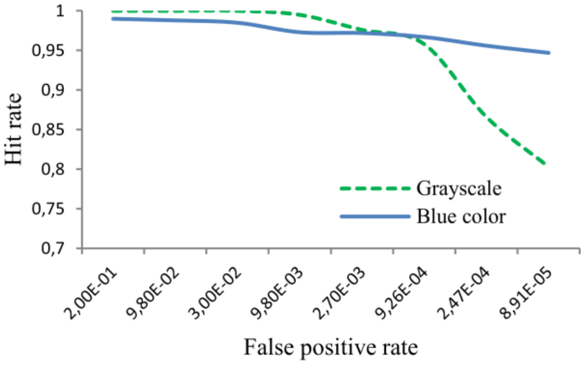

Figure 3: ROC curve for cascades trained with gray and color features.

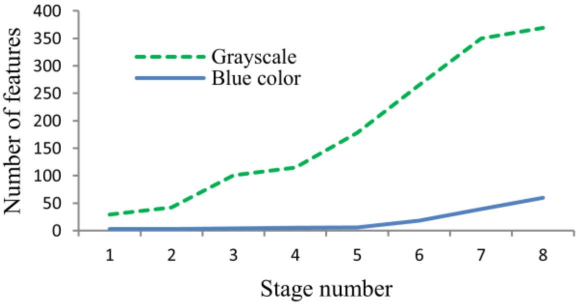

Figure 4: Number of features on the different stages of cascades trained with gray and color features.

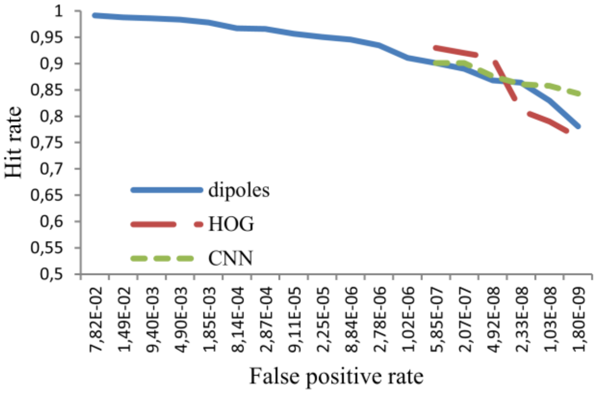

Figure 5: Comparison of different features and classifier used on the last stages of cascade for the blue squares type.

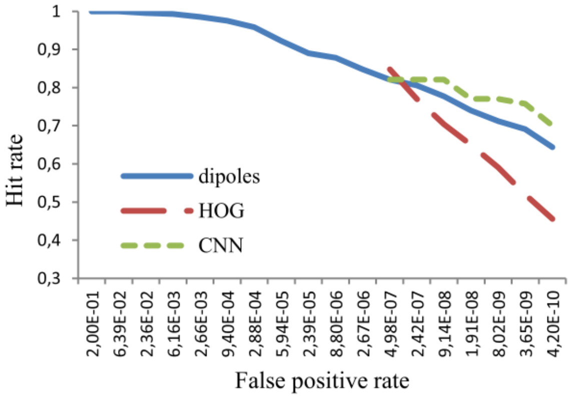

Figure 6**:** Comparison of different features and classifier used on the last stages of cascade for the red triangles type.

[3] We use cuda-convnet library to train the neural network, http://code.google.com/p/cuda-convnet/

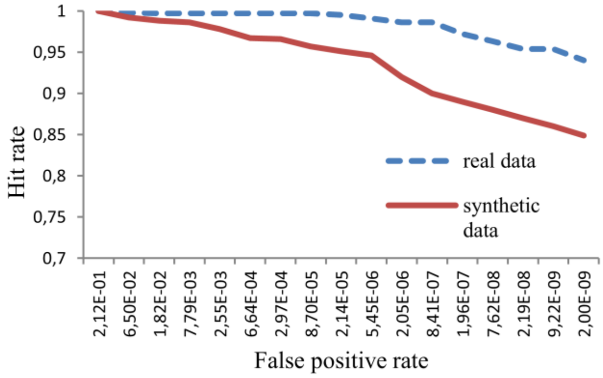

Figure 7: Comparison of detectors trained on real and synthetically generated data.

Sign type	False positive rate	Hit rate per image	Hit rate per physical sign
Blue squares	7E-10	77%	92.18%
Red triangles	7E-10	73.1%	82.35%
Blue circles	6E-10	79.2%	83%
Red circles	2E-9	73.8%	84.7%

Table 1: Final accuracy of the detection cascade.

more powerful features and classifiers after this point. We evaluated two possibilities. First one is to train several stages of the cascade with AdaBoost on HOG features and liner SVM as a weak classifier. Second possibility is to train a deep convolutional neural network with several bootstrapping rounds. Figure 5 and Figure 6 show that HOG features do not increase the accuracy of the cascade whereas a neural network classifier allows reaching the same false positive rate with a hit rate better on average by 7%. Final detector accuracies for four types of signs could be found in Table 1.

Although performance of our detector meets aforementioned requirements we have also tried to train a detector of blue squares signs on real data. It could be seen from Figure 7 that detector trained on real data has superior performance. But if we compare hit rates counted per physical sign difference would be relatively small – 96.88%, compared to 92.18%.

7. TRAFFIC SIGNS RECOGNITION

Our recognition module is based on a deep convolutional neural network which recently proved to be the good choice for this task (Stallkamp et al., 2012). It has 4 layers with weights and 2 max-pooling layers taking as input 30x30 images normalized with histogram equalization. First two layer of the network is convolutional with 112 kernels of size 4x4 with a stride of 1 pixel. Each convolutional layer is followed by the max-pooling layer with size 3x3 and stride of 2 pixels. Second pooling layer is followed by two fully-connected layers. First layer has 100 neurons in it. Second one has as many neurons as there are sign classes for this sign type. Its output is fed into a softmax layer which produces a distribution over class labels. Network hyper-parameters were selected via cross-validation on synthetic data. We train four classifiers for each sign type using synthetically generated data. Results of our classifiers could be found in Table 2.

Lower accuracy on red circles and red triangles is due to the large number of visually similar classes that differ only in a few pixels. For example, “speed limit” signs in red circles or “side-road ahead” in red triangles (Figure **8**).

Sign type	Num. of classes	Num. of training samples	Percent of recognized physical signs (among detected)
Blue squares	31	279 000	96.6%
Red triangles	46	414 000	92.8%
Blue circles	16	144 000	100%
Red circles	47	423 000	93.8%

Table 2: Recognition results for four sign types.

Data type	Num. of classes	Num. of training samples per class	Accuracy
Real	42	> 15	93.7%
Synthetic	42	> 9000	94.1%

Table 3: Comparison of recognizer’s trained on real and synthetically generated data.

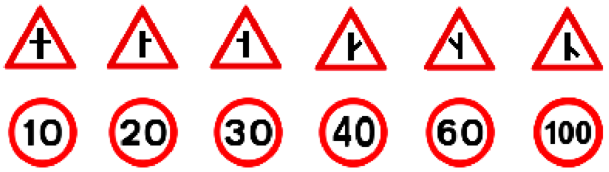

Figure 8: Difficult signs for classification.

We have also compared recognizers trained on real and synthetically generated data. For this experiment we have selected 42 classes that have more than 15 real samples in the dataset. Result from Table 3 show that synthetically trained recognizer performs better. We think this is due to the greater number of training samples and wider set of transformations that could be covered by synthetic data.

8. TRAFFIC SIGNS LINKAGE AND MAPPING

To link detected traffic signs between frames we use a simple algorithm that is working in the image pixels space. It is predicting a position of the sign on the next frame using the equation of a linear uniformly accelerated motion:

$r = r_0 + v_0 t + \frac{at^2}{2}$, where r_0 is the initial position of the sign on the image, v_0 is velocity, a is acceleration and r is the final position of the sign after the time interval.

To use this equation we should know v_0 and a which are easily computed using the finite differences method if we know position of the sign on last 2 and 3 frames accordingly. If we just encountered new traffic sign and do not have enough frames behind then v_0 or a are considered to be equal to zero. We select closes detected sign to prediction if the distance is small enough.

After linking signs from neighboring frames into one physical sign we can obtain its position in world coordinates via triangulation. We can also refine sign’s class label using recognition results from different frames. Frame recognitions are weighted by detector window size, because recognition in a large window tend to be more accurate.

9. SUMMARY AND CONCLUSION

We presented a system for the large-scale traffic signs recognition and mapping and evaluated it on more than 140 classes of Russian traffic signs. For evaluation we created a large dataset captured on the roads of Russia and made it publicly available to encourage future comparison. Our system is trained on synthetically generated data and does not require labor-intensive labelling of the training data.

We show that color features significantly improve the accuracy and speed of the detector. Usage of the deep neural network on the last stage of the detector cascade allowed training on radically larger datasets and improved the hit rate of the detector on average by 7% comparing to cascade trained on dipoles features. For recognition module we presented results of signs classifier trained on synthetic data. We also compared recognizers trained on synthetic and real data and showed that training from synthetic data yields better accuracy.

REFERENCES

Arlicot A., Soheilian B. and Paparoditis N., 2009, Circular road sign extraction from street level images using colour, shape and texture database maps, Proceedings City Models, Roads and Traffic, Paris, France, pp. 205-209.

Balas, B., and Sinha, P., 2003. STICKS: Image-representation via non-local comparisons. Journal of Vision, 3(9).

Baro, X., Escalera, S., Vitria, J., Pujol, O., Radeva, P., 2009. Traffic sign recognition using evolutionary Adaboost detection and Forest-ECOC classification. IEEE Transactions on Intelligent Transportation Systems, 10(1), pp. 113-126.

Dalal, N., Triggs, W., 2005. Histogram of oriented gradients for human detection. Proc. IEEE Conf. Comput. Vis. and Pattern Recog., San Diego, California, pp. 886-893.

Enzweiler M., Gavrila D. M., 2008. A Mixed Generative-Discriminative Framework for Pedestrian Classification, Proceedings Computer Vision and Pattern Recognition.

Grauman K., Shakhnarovich G. and Darrell T., 2003. Inferring 3D structure with a statistical image-based shape model, Proceedings Ninth IEEE International Conference on Computer Vision, pp. 641-647.

Ihara, A., Fujiyoshi, H., Takagi, M., Kumon, H., Tamatsu, Y., 2009. Improved matching accuracy in traffic sign recognition by using different feature subspaces. Proceedings of the 11th IAPR Conference on Machine Viision Applications, Keio, Japan, pp. 130-133.

Larsson, F., Felsberg, M., 2011. Using Fourier Descriptors and Spatial Models for Traffic Sign Recognition. Proc. of Scandinavian Con. on Image Analysis, Sweden, pp. 238-249.

Liebelt J., Schmid C., Schertler K., 2008. Viewpoint-Independent Object Class Detection using 3D Feature Maps, Proceedings Computer Vision and Pattern Recognition.

Marin J., Vazquez D., Geronimo D., Lopez A.M., 2010. Learning Appearance in Virtual Scenarios for Pedestrian Detection, Proceedings Computer Vision and Pattern Recognition, pp. 137-144.

Mathias, M., Radu Timofte, Rodrigo Benenson, and Luc Van Gool, 2013. Traffic Sign Recognition – How far are we from the solution? In International Joint Conference on Neural Networks.

Møgelmose, A., Trivedi, M., Moeslund, T., 2012. Learning to Detect Traffic Signs: Comparative Evaluation of Synthetic and Real-world Datasets, Proc. of the 21st Int. Conf. on Pattern Recognition, Tsukuba, Japan, pp. 3452-3455.

Overett, G. M., Tychsen-Smith, L., Petersson, L., Andersson, L., Pettersson, N., 2011. Creating Robust High-Throughput Traffic Sign Detectors Using Centre-Surround HOG Statistics. Machine Vision and Applications, special issue paper, pp. 1-14.

Ozuysal M., Fua P. and Lepetit V., 2007. Fast Keypoint Recognition in Ten Lines of Code, IEEE Conference on Computer Vision and Pattern Recognition.

Paulo, C., Correia, P. L., 2009. Traffic Sign Recognition Based on Pictogram Contours. Proc. of 9th Int. Workshop on Image Analysis for Multimedia Interactive Services, pp. 67-70.

Pishchulin L., Thorm T., Wojek C., Andriluka M., Thormahlen T., Schiele B., 2011. Learning People Detection Models from Few Training Samples, Proceedings Computer Vision and Pattern Recognition.

Ruta, A., Porikli, F.,Watanabe, S., Li, Y., 2011. In-vehicle camera traffic sign detection and recognition. Mach. Vis. Appl., 22(2), pp. 359–375.

Shotton J., Fitzgibbon A., Cook M., Sharp T., Finocchio M., R. Moore R., Kipman A., Blake A., 2011. Real-Time Human Pose Recognition in Parts from a Single Depth Image, Proceedings IEEE Computer Vision and Pattern Recognition, pp. 1297-1304.

Stark M., Goesele M. and Schiele B., 2010. Back to the Future: Learning Shape Models from 3D CAD Data, Proceedings of the British Machine Vision Conference, pp. 106.1-106.11.

Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C., 2012. Man vs. Computer: Benchmarking Machine Learning Algorithms for Traffic Sign Recognition. *Neural Networks*, 32, pp. 323-332.

Timofte, R., Zimmermann, K., Gool, L, 2009. Multi-view traffic sign detection, recognition, and 3D localization. *Workshop on Applications of Computer Vision*, Snowbird, Utah, pp. 1-8.

Wang K., Babenko B. and Belongie S., 2011. End-to-End Scene Text Recognition, International Conference on Computer Vision