Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 201 (2017) 718–725

Procedia Engineering

www.elsevier.com/locate/procedia

3rd International Conference "Information Technology and Nanotechnology", ITNT-2017, 25-27 April 2017, Samara, Russia

CNN Design for Real-Time Traffic Sign Recognition

Alexander Shustanov[a], Pavel Yakimov[a]*

[a]*Samara National Research University, Moskovskoye shosse 34, Samara, Russia*

Abstract

Nowadays, more and more object recognition tasks are being solved with Convolutional Neural Networks (CNN). Due to its high recognition rate and fast execution, the convolutional neural networks have enhanced most of computer vision tasks, both existing and new ones. In this article, we propose an implementation of traffic signs recognition algorithm using a convolution neural network. The paper also shows several CNN architectures, which are compared to each other. Training of the neural network is implemented using the TensorFlow library and massively parallel architecture for multithreaded programming CUDA. The entire procedure for traffic sign detection and recognition is executed in real time on a mobile GPU. The experimental results confirmed high efficiency of the developed computer vision system.

Keywords: TensorFlow; Convolutional Neural Networks; Traffic Sign Recognition; Image Processing; Computer Vision; Mobile GPU

1. Introduction

Development of the technical level of modern mobile processors enabled many vehicle producers to install computer vision systems into customer cars. These systems help to significantly improve the safety and implement an important step on the way to autonomous driving. Among other tasks solved with computer vision, the traffic sign recognition (TSR) problem is one of the most well-known and widely discussed by lots of researchers. However, the main problems of such systems are low detection accuracy and high demand for hardware computational performance, as well as the inability of some systems classify the traffic signs from different countries.

* Corresponding author. Tel.: +7-927-607-0854;
E-mail address: {alexander.shustanov, pavel.y.yakimov}@gmail.com

10.1016/j.proeng.2017.09.594

Recognition of traffic signs is usually solved in two steps: localization and subsequent classification. There are many different localization methods [1], [2], [3]. In papers [4] and [5], the authors proposed effective implementations of the image preprocessing and traffic signs localization algorithms, which performed in real time. Using a modified Generalized Hough Transform (GHT) algorithm, the solution allowed to determine the exact coordinates of a traffic sign in the acquired image. Thus, in the classification stage, the simple template matching algorithm was used. Combined with precise localization stage, this algorithm showed the final results of 97.3% accuracy of traffic sign recognition. The datasets from GTSRB [6] and GTSDB [7] was used for training and testing the developed algorithms. Fig. 1 shows the images for training the traffic signs recognition algorithm and testing the localization algorithm.

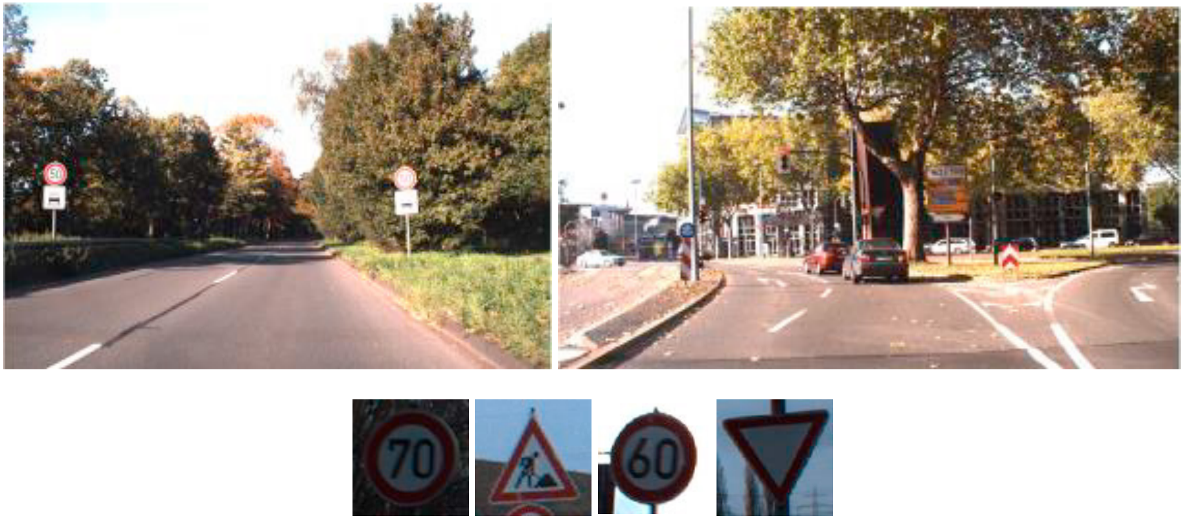

Fig. 1. Images from GTSDB and GTSRB.

While testing the developed technology for detecting and classifying traffic signs in real conditions, i.e. using videos from cameras installed on a windshield, the end-to-end technology showed significant decrease in the efficiency. Studies have shown that such a decrease arose because of too strong variations in the illumination, contrast, and angle of rotation in images of localized traffic signs. Thus, a simple classification algorithm like template matching was not able to achieve high-quality recognition because of a limited set of predefined templates. To improve the system performance, the localization algorithm that has shown good results can be combined with recognition using the convolutional neural networks that have received such a wide application in recent years [8], [9].

In this paper, we describe a revised end-to-end technology for detecting and recognizing traffic signs in real time. The developed system uses the speed received from the vehicle. This allows you to predict not only the presence of the object, but also the scale and its exact coordinates in the neighboring frame. Thus, the accuracy of detection increases, while the computational complexity remains the same. The classification of localized objects is implemented using convolutional neural networks (CNNs). One of the main contributions of this paper is describing the process of designing a convolutional neural network. The use of the GPU allows real-time processing of the frames in the video sequence.

2. Traffic Sign Localization and Tracking

The developed technology for traffic signs recognition consists of three steps: image preprocessing, localization and classification.

During image preprocessing, the HSV color space is used to extract red and blue pixels from an image. Due to errors in the process of images acquiring and the presence of small colored objects, some point-like noise occurs in the images after applying a threshold filter. To address this point-like noise we apply the algorithm described in [4]. Paper [10] shows the effective implementation of the algorithm for noise removal implemented using CUDA. With GPUs, the acceleration reaches 60-80 times as compared with conventional executing on a CPU. The frame size is 1920x1080 pixels. Using the CUDA-enabled mobile GPU NVIDIA Jetson TK1 allows to preprocess one videoframe within 7-10 ms, which satisfies the requirements of video processing in real time.

Paper [5] addresses the algorithms for detecting and tracking traffic signs. The method for localization, which is a modification of the generalized Hough transform, has been developed considering the constraints on the time for processing a single frame. The algorithm shows effective results and functions well with the preprocessed images. Tracking using the value of the vehicle current speed has improved the performance of the system, as the search area in the adjacent frames can be significantly reduced. In addition, the presence of a sign in the sequence of adjacent frames in predicted areas significantly increases the confidence of correct recognition. Classification, which is the final step, ensures that the entire procedure has been executed successfully.

3. Traffic Sign Classification

3.1. Convolutional Neural Networks

Classification with artificial neural networks is a very popular approach to solve pattern recognition problems. A neural network is a mathematical model based on connected via each other neural units – artificial neurons – similarly to biological neural networks. Typically, neurons are organized in layers, and the connections are established between neurons from only adjacent layers. The input low-level feature vector is put into first layer and, moving from layer to layer, is transformed to the high-level features vector. The output layer neurons amount is equal to the number of classifying classes. Thus, the output vector is the vector of probabilities showing the possibility that the input vector belongs to a corresponding class.

An artificial neuron implements the weighted adder, which output is described as follows [11]:

$$a_j^i = \sigma(\textstyle\sum_k a_k^{i-1} w^i j_k), \tag{1}$$

where a_j^i is the j^{th} neuron in the i^{th} layer, w_k^{ij} stands for weight of a synapse, which connects the j^{th} neuron in the i^{th} layer with the k^{th} neuron in the layer i-1. Widely used in regression, the logistic function is applied as an activation function. It is worth noting that the single artificial neuron performs the logistic regression function.

The training process is to minimize the cost function with minimization methods based on the gradient decent also known as backpropagation. In classification problems, the most commonly used cost function is the cross entropy:

$$H(p,q) = -\sum_i Y(i)\log y(i). \tag{2}$$

Training networks with large number of layers, also called deep networks, with sigmoid activation is difficult due to vanishing gradient problem. To overcome this problem, the ELU function is used as an activation function [12]:

$$ELU(x) = \begin{cases} \exp(x) - 1, x \le 0 \\ x, x > 0 \end{cases}. \tag{3}$$

Today, classifying with convolutional neural networks is the state of the art pattern recognition method in computer vision. Unlike traditional neural networks, which works with one-dimensional feature vectors, a convolutional neural network takes a two-dimensional image and consequentially processes it with convolutional layers.

Each convolutional layer consists of a set of trainable filters and computes dot productions between these filters and layer input to obtain an activation map. These filters are also known as kernels and allow detecting the same features in different locations. For example, Fig. 2 shows the result of applying convolution to an image with 4 kernels.

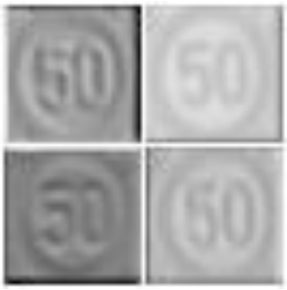

Fig. 2. Input image convolution.

3.2. Proposed Implementation

To solve the traffic sign recognition task, we used the deep learning library TensorFlow [13]. Training and testing were implemented using the dataset from GTSRB [6]. The developed method can classify the 16 most popular traffic signs types.

There are some rules how to build network architecture. Despite of this, network architecture designing process is mostly heuristic. Layers are selected in such way that data dimensionality reduces from layer to layer. But there are no any prescriptions about particular layer macro parameters.

Network depth should correlate with data amount. Huge network and scarce data likely would produce overfitted model. On the other hand, shallow network with large data would not give enough accuracy. So, it is very important to found an equilibrium between network depth and data amount.

Table 1 describes the first developed network architecture. The architecture consists of several convolutional layers, fully connected layers and one softmax layer. All convolutional layers have parameter stride equal to 2. This parameter determines the stride of the convolution sliding window, so layers with parameter stride greater than 1 also combine the pooling operation. The softmax layer normalizes the previous layer output so that its output contains probabilities of belonging to recognizable classes for the original input image.

Table 1. Neural network architecture.

Layer
Convolutional, stride 2, kernel 7x7x4
Convolutional, stride 2, kernel 5x5x8
Convolutional, stride 2, kernel 3x3x16
Convolutional, stride 2, kernel 3x3x32
Convolutional, stride 1, kernel 2x2x16
Convolutional, stride 1, kernel 2x2x8
Convolutional, stride 1, kernel 2x2x4
Fully connected-64
Fully connected-16
Softmax

When training a network with proposed in Table 1 architecture, the classification accuracy reached a value more than 0.9. However, this architecture seems to be excess due to large number of layers. Thus, we decided to reduce the number of convolutional layers, which after several unsuccessful attempts resulted in the architecture presented in Table 2.

Table 2. Second neural network architecture.

Layer
Convolutional, stride 2, kernel 3x3x16
Fully connected-512
Softmax

Fig. 3a shows the accuracy of CNN from Table 2 changing with growing number of training iterations. The plot shows that this network reaches worse results than the previous one. It seems that a single convolutional layer is not enough to obtain the required training accuracy. Table 3 shows the modified architecture of CNN.

Table 3. Final neural network architecture.

Layer
Convolutional, stride 2, kernel 3x3x16
Convolutional, stride 2, kernel 3x3x32
Convolutional, stride 2, kernel 3x3x64
Fully connected-512
Softmax

TensorFlow contains a set of tools to visualize models at different abstraction levels down to low-level mathematical operations. The common name of these tools is TensorBoard. The presented model can be divided into two stacked blocks: the convolutional block and the fully connected block.

To train and evaluate the model, the initial dataset was divided into the train and test datasets with ratio 80/20 correspondently. At the training stage, the network processed the batch of 50 images from the train dataset per one iteration. Every 100 iterations, the intermediate accuracy was computed with batch of 50 images from the test dataset. After successful training, the accuracy was computed using all images from the test dataset. Fig. 3 shows the classification accuracy growing with increasing the number of training iterations for second and final models.

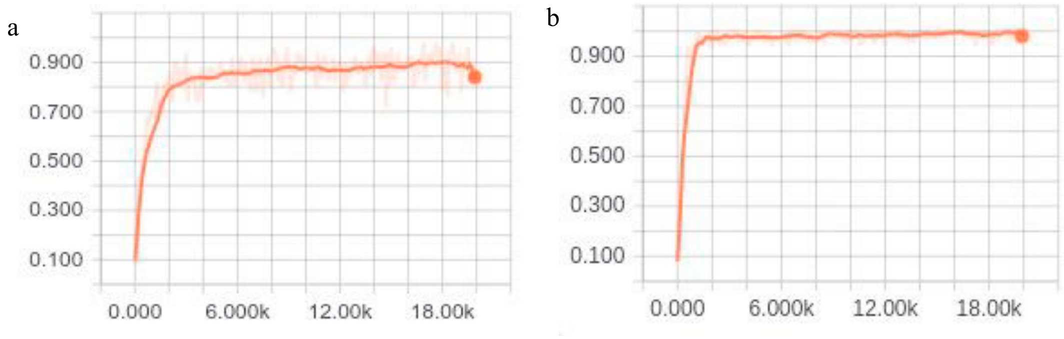

Fig. 3. Classification accuracy changing with training iterations for second (a) and final (b) networks.

4. Experimental Results

As the paper emphasizes on an end-to-end solution to real-time traffic sign localization and recognition, it is necessary to evaluate preprocessing, localization and classification performance. Paper [5] shows an effective implementation of localization with preprocessing algorithms that executes in 20 ms.

To evaluate the classification execution time, we used the GPUs Nvidia GeForce GTX 650 and Nvidia GeForce GT 650M, and CPU Intel Core i7 5500u. Table 4 shows the results.

Table 4. CNN training and classifying execution time.

Hardware	Training	Classifying an image (64x64)
Nvidia GeForce GTX 650	7 min	0.05 ms
Nvidia GeForce GT 650M	12 min	0.14 ms
Intel Core i7	16 min	0.37 ms

To evaluate the localization and recognition algorithms accuracy, we used the German Traffic Sign Detection Benchmark (GTSDB) [7] and the German Traffic Sign Recognition Benchmark (GTSRB) [6]. They contain more than 50,000 images with traffic signs registered in various conditions. To assess the quality of the sign localization, we counted the number of images with correctly recognized traffic signs. When testing the developed algorithms, we used only 9,987 images containing traffic signs of the required shape and with red contours. The experiments showed 99.94% of correctly localized and detected prohibitory and danger traffic signs.

Table 5 shows the resulting accuracy and performance of the detection algorithms from [5], [6], [13] and the method described in this paper.

The accuracy of all methods shown in the table was obtained using the dataset GTSDB. The sliding window method [14] shows the best result with 100% of accuracy. However, the described in this paper modified GHT+CNN reaches the best performance.

One of the most efficient methods for TSR using GTSDB and GTSRB is the method using ConvNet for both localizing and classifying traffic signs [15]. The authors show results reaching precision equal to 99.89% when detecting a sign and 99.55% when classifying it. Also, the method can process 37.72 high-resolution images per second. The method described in this paper shows slightly better results in both precision and performance, but it is difficult to compare FPS as there is no description of the hardware used for experiments in [15].

Fig. 4 shows images of traffic signs that were successfully recognized by the proposed in this paper CNN implementation. The picture shows that the applied method gives good recognition results even with traffic signs images, which are not easy to recognize for a human.

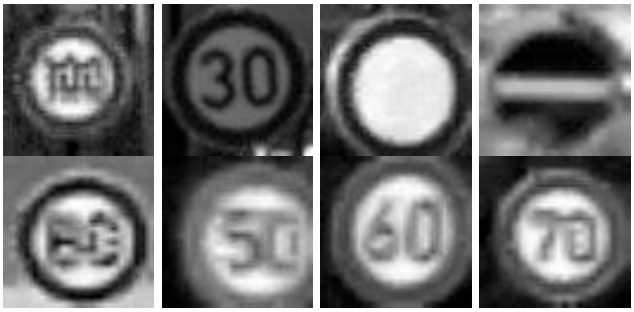

Fig. 4. Successful classification.

However, the accuracy doesn't reach 100 %. Fig. 5 shows the images of traffic signs that were recognized incorrectly.

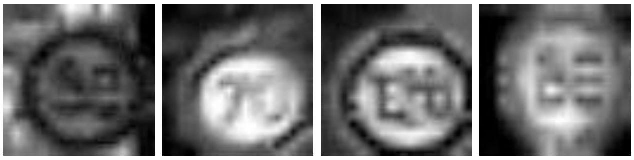

Fig. 5. Unsuccessful classification.

As it is seen in Fig. 5, the quality of input images strongly influences on the recognition rate. It means that such high classification quality will not always be obtainable when using the developed algorithms in real world. However, all the mentioned in Table 5 algorithms will suffer from this input images quality.

Table 5. Accuracy and performance of TSR methods.

Method	Accuracy	FPS
Sliding window + SVN	100 %	1
Modified GHT with preprocessing + CNN (this paper)	99.94 %	50
ConvNet	99.55 %	38
Modified GHT with preprocessing	97.3 %	43
Modified GHT without preprocessing	89.3 %	25
Viola-Jones	90.81 %	15
HOG	70.33 %	20

The developed algorithm was also tested on the video frames obtained in the streets using an Android device Nvidia Shield Tablet built in to a car. Fig. 6 shows the fragments of the original images with marked road signs on them.

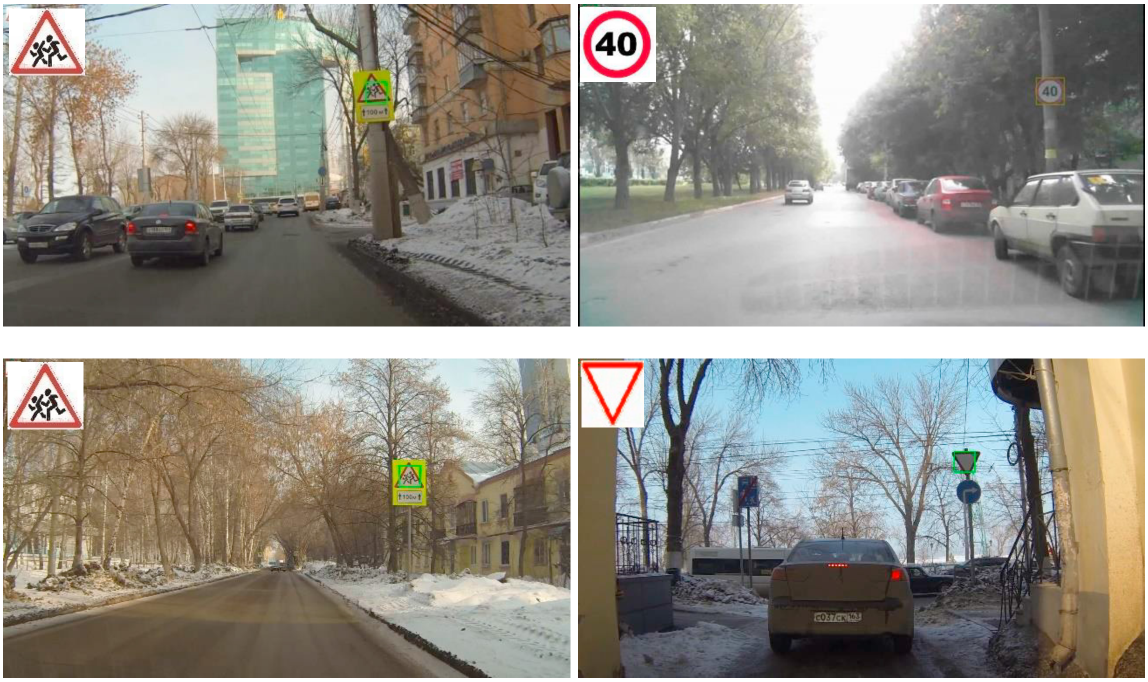

Fig. 6. Localized and recognized traffic signs.

5. Conclusions

This paper considers an implementation of the classification algorithm for the traffic signs recognition task. Combined with preprocessing and localization steps from previous works, the proposed method for traffic signs classification shows very good results: 99.94 % of correctly classified images.

The proposed classification solution is implemented using the TensorFlow framework.

The use of our TSR algorithms allows processing of video streams in real-time with high resolution, and therefore at greater distances and with better quality than similar TSR systems have. FullHD resolution makes it possible to detect and recognize a traffic sign at a distance up to 50 m.

The developed method was implemented on a device with Nvidia Tegra K1 processor. CUDA was used to accelerate the performance of the described methods.

In future research, we plan to train the CNN to consider more traffic sign classes and possible bad weather conditions. Also, we plan to use a CNN not only for classification but for object detection too.

Acknowledgements

This work was supported by the Russian Foundation for Basic Research - Project # 16-37-60106 mol_a_dk.

References

[1] A. Nikonorov, P. Yakimov, M. Petrov, Traffic sign detection on GPU using color shape regular expressions, VISIGRAPP IMTA-4, Paper 8 (2013).

[2] R. Belaroussi, P. Foucher, J.P. Tarel, B. Soheilian, P. Charbonnier, N. Paparoditis, Road Sign Detection in Images, A Case Study, 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 484-488.

[3] A. Ruta, F. Porikli, Y. Li, S. Watanabe, H. Kage, K. Sumi, A New Approach for In-Vehicle Camea Traffic Sign Detection and Recognition, IAPR Conference on Machine Vision Applications (MVA), Session 15: Machine Vision for Transportation, 2009.

[4] V. Fursov, S. Bibkov, P. Yakimov, Localization of objects contours with different scales in images using Hough transform [in Russian], Computer Optics. 37, 4 (2013) 502-508.

[5] P. Yakimov, Tracking traffic signs in video sequences based on a vehicle velocity [in Russian], Computer Optics. 39, 5 (2015) 795-800.

[6] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition, Neural networks. 32 (2012) 323-332.

[7] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, C. Igel, Detection of Traffic Signs in Real-World Images: The {G}erman {T}raffic {S}ign {D}etection {B}enchmark, in: Proc. International Joint Conference on Neural Networks, 2013.

[8] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, S. Hu, Traffic-Sign Detection and Classification in the Wild. Proceedings of CVPR, 2016, pp. 2110-2118.

[9] Y. LeCun, P. Sermanet, Traffic Sign Recognition with Multi-Scale Convolutional Networks, Proceedings of International Joint Conference on Neural Networks (IJCNN'11), 2011.

[10] P. Yakimov, Preprocessing of digital images in systems of location and recognition of road signs [in Russian], Computer Optics. 37, 3 (2013) 401-405.

[11] V. Terehov, D. Efimov, I. Tiukin, Neural network control system [in Russian], Textbook for high schools, High school, 2002.

[12] Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), 2015.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, ´ M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint, 1603.04467, 2016. arxiv.org/abs/1603.04467. Software available from tensorflow.org.

[14] M. Mathias, R. Timofte, R. Benenson, L. Gool, Traffic sign recognition - how far are we from the solution? Proceedings of IEEE International Joint Conference on Neural Networks, 2013, pp. 1-8.

[15] H. Aghdam, E. Heravi, D. Puig, A practical approach for detection and classification of traffic signs using Convolutional Neural Networks, Robotics and Autonomous Systems. 84 (2016) 97-112.