Mokslas – Lietuvos ateitis / Science – Future of Lithuania
ISSN 2029-2341 / eISSN 2029-2252
2018 Volume 10, Article ID: mla.2018.6947, 1–5
https://doi.org/10.3846/mla.2018.6947

Information technologies and multimedia
Informacinės technologijos ir multimedija

TRAFFIC SIGN RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS

Ervin MILOŠ, Aliaksei KOLESAU, Dmitrij ŠEŠOK*

Vilnius Gediminas Technical University, Vilnius, Lithuania

Received 05 December 2018; accepted 10 December 2018

Abstract. Traffic sign recognition is an important method that improves the safety in the roads, and this system is an additional step to autonomous driving. Nowadays, to solve traffic sign recognition problem, convolutional neural networks (CNN) can be adopted for its high performance well proved for computer vision applications. This paper proposes histogram equalization preprocessing (HOG) and CNN with additional operations – batch normalization, dropout and data augmentation. Several CNN architectures are compared to differentiate how each operation affects the accuracy of CNN model. Experimental results describe the effectiveness of using CNN with proposed operations.

Keywords: traffic sign recognition, image pre-processing, classification, convolutional neural network, batch normalization, dropout, experiment.

Introduction

In the 21th of century the car industry became very big. The total amount of vehicles in the planet is more than 1.2 billion (Voelcker, 2014) and this number is growing rapidly. To make the safest life conditions for the citizens, scientists develop a lot of innovative technologies that help to prevent vehicle accidents. Furthermore, the innovation of autonomous vehicles is getting more popular and usable, and it needs the highest quality systems that can control the vehicle safely according to all traffic situations.

There are many vision systems in the vehicle like automatic braking, parking or vehicle location but the traffic sign recognition system is most discussed, because traffic accidents mostly comes from distracted driving, speeding and bad weather conditions (Strongtie Insurance, 2018) which makes it difficult to spot the signs in the road. Traffic sign recognition system is implemented on the vehicle with an aim of recognizing all emerging traffic signs. There are a lot of proposed solutions by researches (Yadav, 2016; Haloi, 2015; Ciresan, Meier, Masci, & Schmidhuber, 2011) etc., but traffic sign recognition task keeps challenging. The main problems with this system is high requirements for the hardware and harsh weather conditions. Figure 1 shows the image with bad illumination and lightning variations which do not give the highest percentage accuracy of recognition.

Figure 1. Images from GTSRB (Stallkamp, Schlipsing, Salmen, & Igel, 2012)

Nowadays Convolutional neural network (Lecun et al., 1989) (CNN) has been widely adopted (Yang et al., 2018; Mao et al., 2016; Yin, Deng, Zhang, & Du, 2017; Boujemaa, Bouhoute, Boubouh, & Berrada, 2017) for traffic sign recognition due to the high accuracy in image classification and recognition. In Shustanov and Yakimov (2017), the authors show an effective implementation of classification using convolutional neural network, which gives very high accuracy. They used the deep learning library TensorFlow and training with testing was implemented using German Traffic Sign Recognition Benchmark (GTSRB). The proposed network architecture gave 99.94% of accuracy which is a significant result.

However, the suggested architecture was only tested on prohibitory and danger traffic signs from GTSRB. Authors suggested to consider more traffic signs. Furthermore, it does not contain normalization operations, which can

*Corresponding author. E-mail: *dmitrij.sesok@vgtu.lt*

lead to over-fitting; that is, CNN will classify correctly training images but poorly testing images.

In this paper, the implementation of convolutional neural network comparing different approaches with additional operations – batch normalization (Loffe & Szegedy, 2015) and dropout (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012). Those operations help to prevent the over-fitting (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) and allows model to learn better on training images and correctly classify testing images. Before the training, the images are preprocessed to get the better quality and vision. The training and testing are processed using full GTSRB dataset, which contains danger, prohibitory, regulatory and designation signs. Additionally, the data augmentation is used for GTSRB dataset, which increases the amount of training images.

1. Traffic sign classification

1.1. Traffic sign classification

The traffic sign recognition contains two major steps: preprocessing and classification. For preprocessing the histogram equalization in V channel is used, the main purpose of which is to adjust image intensities to enhance contrast. Initially, image is transformed from RGB to HSV format, because HSV color space is more suitable for image segmentation. After that, V channel which is related to the brightness is extracted and applied to histogram equalization. Melekhov, Kannala, and Rahtu (2017) describes that histogram equalization for the grayscale image increases accuracy. Figure 2 shows that after applying histogram equalization image becomes lighter and more visible.

Preprocessing step contains additional operation which crops the border. Images in the dataset are not guaranteed to be centered in each image. Every image contains about 10 % border around the actual traffic sign (Chilamkurthy, 2017). After cropping the border as shown in Figure 3, the size of the image will be reduced, that is why all images are resized to the same size 48x48.

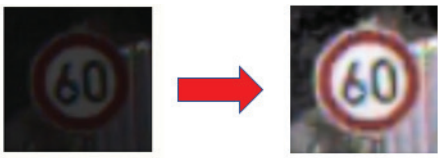

Figure 2. Image after histogram equalization (Melekhov, Kannala, & Rahtu, 2017)

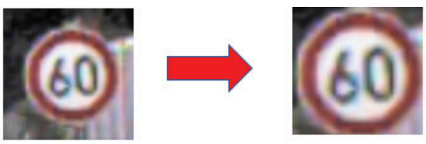

Figure 3. Image after cropping border (Melekhov, Kannala, & Rahtu, 2017)

1.2. Convolutional neural network

Neural networks are getting more popular as approach of classifying traffic signs. Neural networks are a variety of deep learning technologies which generally focus on solving pattern recognition problems (Rouse, 2018).

The convolutional neural network is an extension which is very effective in image recognition and classification. Convolutional neural network allows to pass images and process them with convolution layers. The main purpose of the convolution layer is to extract feature from the traffic sign image. It is done with the help of the two-dimensional filter that can extract features like identity, edge, blur etc. Figure 4 shows how traffic sign image changes after applying different filters during convolution step.

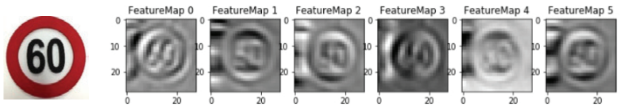

Figure 4. Images after convolution step (Pandiyan, 2017)

After the convolution step the non-linear operation is applied which makes convolutional neural network non-linear, because in this paper the multi-layer network is used. To reduce the number of parameters and calculations the pooling operation is applied for each traffic sign feature map, which reduces dimensionality but retains the most essential information. The Figure 5 shows how traffic sign changes after pooling operation.

Figure 5. Image after pooling step (Pandiyan, 2017)

After pooling step, features of traffic sign are converted into one dimensional vector which is applied to the input layer. Input layer contains feature values which are considered as weights. Those values are transferred through hidden layers to the output layer where outputs are classified into one of traffic signs classes, for example stop sign, speed sign etc.

The convolutional neural network is learning by gradient descent. Gradient descent is an optimization algorithm used to minimize function by iteratively moving in the direction of steepest descent. In the context of learning, backpropagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function.

Adjusted weight in a single connection between two nodes can be calculated with this formula (Arnis, 2017):

$$w = w_i + E \cdot Gradw_c + \alpha \cdot \Delta w_{i-1}, \tag{1}$$

where w_i is current weight; $E-$ learning rate, according to this constant the change of the weight will be small or high; $Gradw_c -$ gradient of the current weight; $\alpha -$ momentum, which used for faster convergence of the loss function. $\Delta w_{i-1} -$ previous change of the weight.

1.3. Convolutional neural network

To implement the traffic sign recognition, deep learning library Keras (Chollet, 2015) is used. Keras is a high-level neural network API that focuses on enabling fast experimentations in neural network. It supports user-friendly functions for designing, modeling and training convolutional neural network.

To make experiments with several convolutional neural network architectures, the collected dataset from German Traffic Sign Recognition Benchmark (GTSRB) is used. It contains more than 39000 images for training and more than 12000 for testing, and for 43 classes (Stallkamp et al., 2012). The input for the network is dataset of training images and the output is the 43 traffic sign classes for the prediction.

There are several proposed architectures for convolutional neural network, which shows very good results in different image recognition tasks (Siddhart, 2017). However, mostly choosing the architecture is heuristic. It is because output depends in the amount of data and hyper parameters. It is important that architecture of the network should correlate with the data. The small amount of data with large network may lead to the over-fitting and in other case the large amount of data with small network will lead to small accuracy of recognition (Shustanov & Yakimov, 2017).

In this paper there are six chosen convolutional neural network architectures to compare how accuracy of classification changes depending on different operations and their combinations. The first architecture is described in the Table 1.

The first architecture contains 6 convolution layers and 2 fully connected layers. The main purpose of this architecture is to check how max pooling affects the neural network output. Max pooling reduces the dimensionality of the feature maps but retain the most essential information. The largest advantage of this operation is that it reduces the number of parameters and computations. Max pooling is applied after each two convolution steps. Activation function is rectified linear unit, which in practice works better. This activation function will be used in all next architectures. Softmax activation function calculates probabilities for each class.

The second architecture extends the first architecture adding additional operation called batch normalization. Batch normalization reduces the amount by what the hidden unit values shift. It makes sure that there is no activation that will go too high or low. Additionally, it reduces the overfitting because it has a slight regularization effects (Doukali, 2017).

The third architecture extends the second architecture adding new operation called dropout. Dropout refers to dropping out some hidden and visible neurons in neural network. Neurons are dropped during the training step and they are chosen randomly. This operation helps to prevent over-fitting, because fully connected layer takes most of the parameters and neurons make a high dependency amongst each other during training step which decreases the individual importance of the neuron (Budhiraja, 2016). Dropout is applied after each batch normalization step.

The main purpose of fourth architecture is to check how accuracy will change if the structure of the neural network will contain smaller amount of convolution layers. Table 2 shows, that architecture contains only 3 convolution layers.

In this architecture all operations from third architectures are used – max pooling, dropout and batch normalization. Those operations are applied after each convolution layer.

The main aim of the fifth architecture is to check how the accuracy of recognition will change if the network contains more convolution layers. The Table 3 shows that architecture contains 8 convolutional layers.

Table 1. First architecture with max pooling

Layer name	Size	Filter size	Operation	Activation function
Convolution layer	32	3x3	Max pooling	ReLU
Convolution layer	32	3x3		ReLU
Convolution layer	64	3x3	Max pooling	ReLU
Convolution layer	64	3x3		ReLU
Convolution layer	128	3x3	Max pooling	ReLU
Convolution layer	128	3x3		ReLU
Fully connected layer	512	–	–	ReLU
Fully connected layer	43	–	–	Softmax

Table 2. The fourth architecture with smaller network

Layer name	Size	Filter size	Operation	Activation function
Convolution layer	32	3x3	Max pooling, Batch Normalization, Dropout	ReLU
Convolution layer	64	3x3	Max pooling, Batch Normalization, Dropout	ReLU
Convolution layer	128	3x3	Max pooling, Batch Normalization, Dropout	ReLU
Fully connected layer	512	–	–	ReLU
Fully connected layer	43	–	–	Softmax

Table 3. The fifth architecture with larger network

Layer name	Size	Filter size	Operation		Activation function
Convolution layer	32	3x3	Batch Normalization	Max pooling	ReLU
Convolution layer	32	3x3	Dropout		ReLU
Convolution layer	64	3x3	Batch Normalization	Max pooling	ReLU
Convolution layer	128	3x3	Dropout		ReLU
Convolution layer	192	3x3	Batch Normalization	Max pooling	ReLU
Convolution layer	256	3x3	Dropout		ReLU
Convolution layer	128	3x3	Batch Normalization	Max pooling	ReLU
Convolution layer	64	3x3	Dropout	Max pooling	ReLU
Fully connected layer	512	–	–		ReLU
Fully connected layer	43	–	–		Softmax

This architecture uses all the same operations from the fourth architecture. The larger network shows the best results that is why, in the final sixth architecture the data augmentation is used to increase the amount of training images. Image augmentation is the process of taking images from training dataset and manipulating them to create many altered versions of the same image. This provides more image to train on and help expose classifier to a wider variety of lighting and coloring situations to make classifier stronger. The Figure 6 shows how image changes during augmentation.

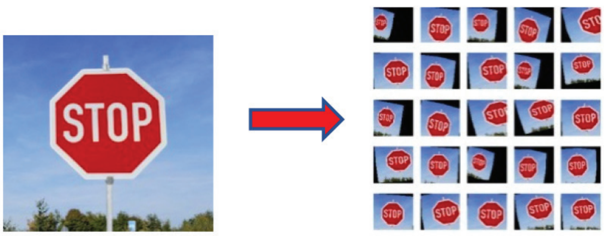

Figure 6. Image augmentation (Yadav, 2016)

To train and evaluate the several major parameters were chosen for all six architectures:

- Batch size – 32;
- Number of epochs – 30;
- Learning rate – 0.01;
- Momentum – 0.9;
- Optimizer – mini batch gradient descent;
- Loss function – cross entropy;
- Metrics – accuracy;

2. Experimental results

6 architectures were tested using graphics card NVIDIA GTX 860m. For the first 5 architectures training time approximately took 45 min per each. For 6-th architecture where data augmentation is applied, training process took 6 hours.

After training the convolutional neural network with the 1-st architecture which used only max pooling operation classified traffic signs with 95% of accuracy. The 2-nd architecture was trained using additional batch normalization operation together with max pooling. The evaluation results showed 95.5% of accuracy which improved the 1-st architecture by 0.5%. The 3-rd architecture was trained to check how accuracy changes after applying dropout operation together with batch normalization and dropout. After testing the network, traffic signs were classified with 97.2% of accuracy which improved the 2-nd architecture by 1.7%. In the 4-th architecture, the CNN size was reduced and using the same operations as in the 3-rd architecture, the results showed accuracy of 96.1%, which is smaller than in the 3-rd architecture. In the 5-th architecture, the CNN size was increased and using same operations as in the 3-rd architecture the traffic signs were classified with a 98.1% of accuracy which is better than in 4-rd architecture by 2% and 3-rd architecture by 1.5%. The final 6-th architecture was tested with data augmentation and it showed the best accuracy of 99.24%.

Table 4 shows that the best accuracy of 99.24% reached the 6-th architecture, which contained all operations – max pooling, batch normalization and dropout. Furthermore, this network was the largest and was trained using data augmentation.

Table 4. Experiments results

Architecture no.	1	2	3	4	5	6
Accuracy	95%	95.5%	97.2%	96.1%	98.1%	99.24%

Table 5 shows the comparisons of traffic sign classification accuracy from (Haloi, 2015; Ciresan et al., 2011; Stallkamp et al., 2012; Yadav, 2016) and the proposed model described in this paper.

Table 5. Classification accuracy comparison

Model	Accuracy
Spatial Transformer + CNN	99.81%
HOG + CNN + BN/Dropout/Augmentation (this paper)	99.24%
CLAHE + CNN	99.15%
Human Accuracy	98.84%
CNN + Dropout/Augmentation	98.8%

The best result shows model described (Haloi, 2015) in paper with 99.81% of accuracy. This model instead of preprocessing uses spatial transformer layer which makes network more robust to deformations such as translation, rotation and scaling of input images.

Conclusions

This paper proposes an implementation for traffic sign classification using together batch normalization, dropout and data augmentation. The model achieved a very high performance of up to 99.24% which is higher than human performance.

This concludes, that batch normalization and dropout are very important operations, because they help to reduce the chance of network over-fitting. Furthermore, the deeper neural network with data augmentation gives better results than shallow network with smaller amount of data.

For the future work, more convolutional neural network architectures by changing hyper-parameters will be considered for testing. Additionally, instead of using preprocessing the spatial transformer layer will be tested together with all convolutional neural network operations described in this paper.

References

Arnis. (2017). *Neural networks for the beginners. Part 2* [in Russian]. Retrieved from http://www.pvsm.ru/algoritmy/242677

Boujemaa, K. S., Bouhoute, A., Boubouh, K., & Berrada, I. (2017). Traffic sign recognition using convolutional neural networks. *International Conference on Wireless Networks and Mobile Communications (WINCOM)* (pp. 1-6). Rabat.

Budhiraja, A. (2016). *Dropout in (Deep) Machine learning*. Retrieved from https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5

Chilamkurthy, S. (2017). *Keras Tutorial – Traffic Sign Recognition.* Retrieved from https://chsasank.github.io/keras-tutorial.html

Chollet, F. (2015). *Keras*. Retrieved from https://github.com/keras-team/

Ciresan, D., Meier, U., Masci, J., & Schmidhuber, J. (2011). A committee of neural networks for traffic sign classification. *Proceedings of the International Joint Conference on Neural Networks. 1918-1921. 10.1109/IJCNN.2011.6033458.* Retrieved from http://www.people.usi.ch/mascij/data/papers/2011_ijcnn_committee.pdf

Doukali, F. (2017). *Batch normalization in Neural Networks*. Retrieved from https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c

Haloi, M. (2015). Traffic Sign Classification Using Deep Inception Based Convolutional Networks. *CoRR, abs/1511.02992*. Retrieved from https://arxiv.org/pdf/1511.02992.pdf

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR abs/1207.0580*.

Yadav, V. (2016). *German sign classification using deep learning neural networks*. Retrieved from https://chatbotslife.com/german-sign-classification-using-deep-learning-neural-networks-98-8-solution-d05656bf51ad

Yang, Y., Liu, S., Ma, W., Wang, Q., Zheng, & Liu. (2018). Efficent Traffic-Sign Recognition with Scale-aware CNN. *BMVC*. Retrieved from BMVC.

Yin, S., Deng, J., Zhang, D., & Du, J. (2017). Traffic Sign Recognition Based on Deep Convolutional Neural Network. *CCCV*, 685-695.

Lecun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computations*, 541-551.

Loffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. ICML.

Mao, X., Hijazi, S. L., Casas, R. A., Kaul, P., Kumar, R., & Rowen, C. (2016). Hierarchical CNN for traffic sign recognition. *Intelligent Vehicles Symposium,* 130-135.

Melekhov, I., Kannala, J., & Rahtu, E. (2017, October 31). *Image Patch Matching Using Convolutional Descriptors with Euclidean Distance.* Retrieved from https://arxiv.org/pdf/1710.11359.pdf

Pandiyan, D. (2017). *Traffic Sign Classifier*. Retrieved from https://github.com/dhnkrn/Traffic-Sign-Classifier

Rouse, M. (2018). *Neural network.* Retrieved from https://searchenterpriseai.techtarget.com/definition/neural-network

Shustanov, A., & Yakimov, P. (2017). CNN Design for Real-Time Traffic Sign Recognition. *3rd International Conference "Information Technology and Nanotechnology".* Samara.

Siddhart, D. (2017, November 17). *CNNs Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more …* Retrieved from https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15(1)*, 1929-1958.

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012, February). *Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition.* Retrieved from http://www.sciencedirect.com/science/article/pii/S0893608012000457

Strongtie Insurance (2018). *What Are The Most Common Reasons for Road Accidents?* Retrieved from https://www.strongtieinsurance.com/common-reasons-road-accidents/

Voelcker, J. (2014). *1.2 Billion Vehicles On World's Roads Now, 2 Billion By 2035: Report*. Retrieved from https://www.greencarreports.com/news/1093560_1-2-billion-vehicles-on-worlds-roads-now-2-billion-by-2035-report

KELIO ŽENKLŲ ATPAŽINIMAS NAUDOJANT NEURONINĮ TINKLĄ

E. Miloš, A. Kolesau, D. Šešok

Santrauka

Kelio ženklų atpažinimas – vienas iš svarbių būdų pagerinti saugumą keliuose. Ši sistema laikoma papildomu autonominio vairavimo žingsniu. Šiandien kelio ženklų atpažinimo problemai spręsti taikomi konvoliuciniai neuroniniai tinklai (KNN) dėl jų našumo, įrodyto vaizdų atpažinimo programose. Šiame straipsnyje siūlomas vaizdų histogramos išlyginimo apdorojimo metodas ir KNN su papildomomis operacijomis – paketo normalizavimas ir neuronų išjungimas / įjungimas. Yra palyginamos kelios KNN architektūros siekiant ištirti, kokią įtaką kiekviena operacija daro KNN modelio tikslumui. Eksperimentiniai rezultatai apibūdina KNN naudojimo efektyvumą su pasiūlytomis operacijomis.

Reikšminiai žodžiai: kelio ženklų atpažinimas, vaizdų apdorojimas, klasifikavimas, konvoliucinis neuroninis tinklas, paketo normalizavimas, neuronų išjungimas / įjungimas, eksperimentai.