Effect of various dimension convolutional layer filters on traffic sign classification accuracy

2 authors:

Valentyn Sichkar
ITMO University
9 PUBLICATIONS   **37** CITATIONS

SEE PROFILE

Sergey A. Kolyubin
ITMO University
94 PUBLICATIONS   **574** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Biomechatronics and Energy-Efficient Robotics View project

Project Computer Vision for an Autonomous Vehicle View project

EFFECT OF VARIOUS DIMENSION CONVOLUTIONAL LAYER FILTERS ON TRAFFIC SIGN CLASSIFICATION ACCURACY

V.N. Sichkar, S.A. Kolyubin

ITMO University, Saint Petersburg, 197101, Russian Federation
Corresponding author: vsichkar@itmo.ru

Abstract
The paper presents the study of an effective classification method for traffic signs on the basis of a convolutional neural network with various dimension filters. Every model of convolutional neural network has the same architecture but different dimension of filters for convolutional layer. The studied dimensions of the convolution layer filters are: 3×3, 5×5, 9×9, 13×13, 15×15, 19×19, 23×23, 25×25 and 31×31. In each experiment, the input image is convolved with the filters of certain dimension and with certain processing depth of image borders, which depends directly on the dimension of the filters and varies from 1 to 15 pixels. Performances of the proposed methods are evaluated with German Traffic Sign Benchmarks (GTSRB). Images from this dataset were reduced to 32×32 pixels in dimension. The whole dataset was divided into three subsets: training, validation and testing. The effect of the dimension of the convolutional layer filters on the extracted feature maps is analyzed in accordance with the classification accuracy and the average processing time. The testing dataset contains 12000 images that do not participate in convolutional neural network training. The experiment results have demonstrated that every model shows high testing accuracy of more than 82%. The models with filter dimensions of 9×9, 15×15 and 19×19 achieve top three with the best results on classification accuracy equal to 86.4 %, 86 % and 86.8 %, respectively. The models with filter dimensions of 5×5, 3×3 and 13×13 achieve top three with the best results on the average processing time equal to 0.001879, 0.002046 and 0.002364 seconds, respectively. The usage of convolutional layer filter with middle dimension has shown not only the high classification accuracy of more than 86 %, but also the fast classification rate, that enables these models to be used in real-time applications.

Keywords
traffic signs classification, convolutional neural network, convolutional layer filters, feature maps extraction, classification accuracy

АНАЛИЗ ВЛИЯНИЯ РАЗЛИЧНОЙ РАЗМЕРНОСТИ ФИЛЬТРОВ СВЕРТОЧНОГО СЛОЯ НА ТОЧНОСТЬ КЛАССИФИКАЦИИ ДОРОЖНЫХ ЗНАКОВ

В.Н. Сичкар, С.А. Колюбин

Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
Адрес для переписки: vsichkar@itmo.ru

Аннотация
Выполнено исследование эффективного метода классификации дорожных знаков на основе сверточной нейронной сети с фильтрами различной размерности. Каждая модель сверточной нейронной сети имеет одинаковую архитектуру, но разную размерность фильтров для сверточного слоя. Исследуемыми размерностями фильтров сверточного слоя являются 3×3, 5×5, 9×9, 13×13, 15×15, 19×19, 23×23, 25×25 и 31×31. В каждом эксперименте входное изображение подвергается операции свертки фильтрами определенной размерности и с определенной глубиной обработки

границ изображения, которая прямо пропорционально зависит от размерности фильтров и варьируется в пределах от 1 до 15 пикселей. Характеристики предложенных методов оцениваются с помощью немецкого набора изображений дорожных знаков (GTSRB). Изображения из данного набора были уменьшены в размерности до 32 × 32 пикселей. Весь набор данных с изображениями был разделен на три части: набор для обучения, набор для валидации и набор для тестирования. Влияние размерности фильтров сверточного слоя на извлеченные карты характеристик анализируется в соответствии с точностью классификации и средним временем обработки. Набор данных для тестирования содержит 12000 изображений, которые не принимают участия в обучении сверточной нейронной сети. Результаты экспериментов показали, что каждая из моделей обладает высокой точностью классификации, которая составляет более 82 %. Модели с размерностью фильтров 9 × 9, 15 × 15 и 19 × 19 вошли в первую тройку с лучшими результатами по точности классификации, которая составила 86,4, 86 и 86,8 % соответственно. Модели с размерностью фильтров 5 × 5, 3 × 3 и 13 × 13 вошли в первую тройку с лучшими результатами по средней скорости обработки, которая составила 0,001879, 0,002046 и 0,002364 секунд соответственно. Использование средней размерности фильтров для сверточного слоя показало не только высокую точность классификации более 86 %, но и высокую скорость классификации, что позволяет использовать такие модели в приложениях для работы в реальном времени.

Ключевые слова
классификация дорожных знаков, сверточная нейронная сеть, фильтры сверточного слоя, извлечение карт характеристик, точность классификации

Introduction

Classification task for traffic signs is important from the traffic safety point of view. A significant part of road traffic violations occurs due to drivers' non-compliance with the speed limit. In this regard, many car manufacturers use traffic sign classification systems to detect a speed limit sign on the image. These systems compare the current speed of the vehicle with the speed allowed on the current section of the road and notify the driver of the excess speed or automatically change the speed. The use of traffic sign classification systems in conjunction with the navigation system makes it possible to obtain data on the speed limit even in cases where the sign has not been determined, namely, the driver will be informed of the possible presence of the sign.

The main problem associated with this task is the classification of traffic signs in real conditions. Night time and bad weather conditions complicate significantly the process of the sign classification on the image.

The methods of image classification based on neural networks are actively used and described in the literature [1–6]. However, every method has its advantages and disadvantages. Therefore, the development of a reliable algorithm is still a problem of open research. When testing sign classification systems in real traffic conditions, some signs may be misinterpreted due to different levels of light, vibration, different angles of shooting traffic signs. To eliminate these shortcomings, convolutional neural networks have proven themselves [7–11]. Such neural networks are more effective in solving image classification problems than fully connected neural networks in terms of computational load, as well as due to considerably less number of configurable parameters. However, the main advantage of convolutional neural networks is that they are invariant with respect to the shape, rotation and color intensity of the input images.

The paper considers the effect of different dimension of convolutional neural network filters on the accuracy and rate of classification. This dimension determines the number of features that will be combined to obtain a new feature at the output feature map. Therefore, the use of small dimension (3 × 3) of convolutional layer filters combines fewer features and can lead to the loss of important information. On the contrary, the use of a large dimension (31 × 31) of convolutional layer filters combines more features, but can lead to redundancy of information on irrelevant or unnecessary characteristics. Training of the convolutional neural network takes place on the GTSRB dataset [12, 13].

Convolutional neural network for traffic sign classification

Classification using convolutional neural networks is a modern method of pattern recognition in computer vision. Convolutional neural network receives an image and processes it in convolutional layers. Every convolutional layer consists of a set of trainable filters that process the input image with a convolution operation. The essence of this operation is that the filter slides over the image and produces an elementwise multiplication of the values of the filter pixels and the current image area. The result is summarized and written in the corresponding position of the output feature map. The peculiarity of the convolution layer filters is that they give the possibility to detect the same specific features in different parts of the image. Mathematically, the convolution operation is described by the following equation:

$$(f \cdot g)[m, n] = \Sigma_{k,l} f[m - k, n - l] \cdot g[k, l],$$

where f is an initial matrix of input image; g is a filter for convolution; m, n are the height and width of the feature map; k, l are the height and width of the filter.

Before feeding to the input of convolutional neural network, every image is preprocessed. Since the images from GTSRB dataset were used for this study, they were first reduced in size to a resolution of 32 × 32 pixels. This dataset was divided into three subsets for training, validation and testing with preserving the proportions of the images for every class. Further, the normalization of images was performed by dividing them by 255 and

subtracting the mean image, which, in turn, was calculated from the training dataset. As a result, the dataset containing 3-channeled RGB images was prepared. Training subset contains 50000 images, validation subset contains 4000 images and testing subset contains 12000 images. The convolutional neural network training takes place with batches of 50 examples at the same time.

The architecture of convolutional neural network is the same for all experiments, but with different dimension of convolutional layer filters. Developed architecture of convolutional neural network under study is shown in Fig. 1.

Three-channeled RGB input image is fed to the convolutional layer, which consists of 32 filters. Since the input image has 3 channels, every filter of convolutional layer also consists of 3 channels. As a result of convolution, 32 feature maps are calculated in accordance with the number of the convolutional layer filters. The ReLU (Rectified Linear Unit) activation function is applied to the received feature maps, which excludes negative values by replacing them with zeros [14, 15]. This is followed by a layer of dimension reduction (also known as pooling layer), followed by a hidden fully connected layer with 500 neurons. The output layer consists of 43 neurons in accordance with number of classes of traffic signs in the GTSRB dataset.

Parameters of the developed convolutional neural network are described in Table 1. As can be seen from Table 1, the loss function in this study is negative log-likelihood function. The cost function in this study is defined as an average of loss functions overall current training batch. The process of convolutional neural network training is to minimize the cost function by the gradient descent method, which is also called back propagation method.
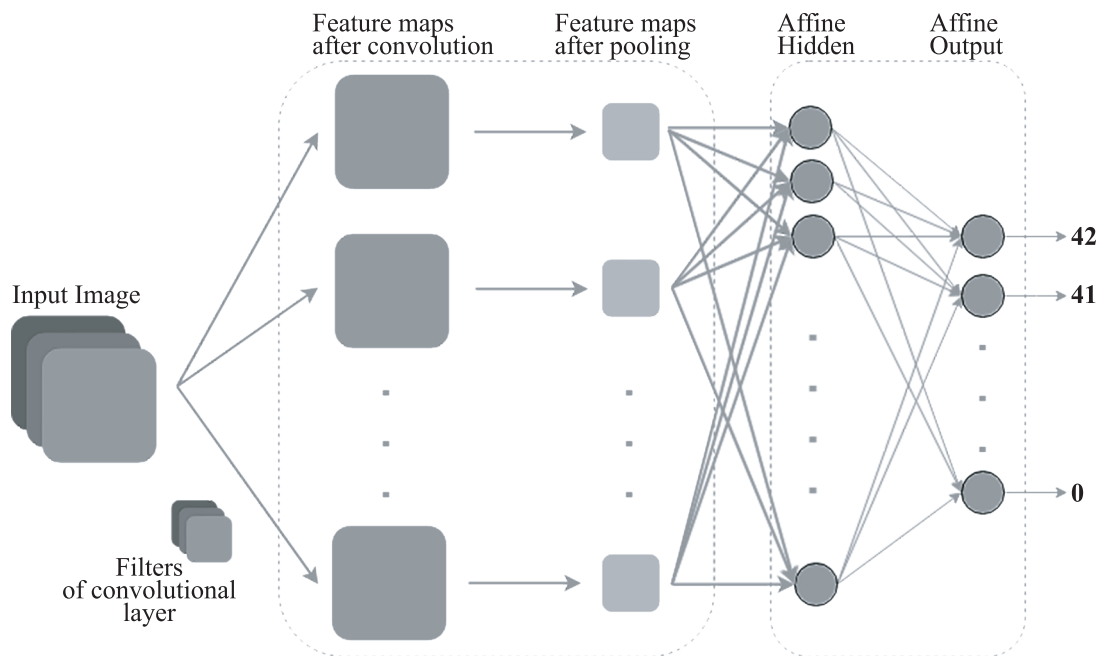

Fig. 1. Architecture of convolutional neural network

Table 1. Parameters of the developed convolutional neural network

Parameter	Description
Weights Initialization	HE Normal
Weights Update Policy	Adam
Activation Function	ReLU
Pooling	2×2 Max
Loss Function	Negative log-likelihood
Cost Function	Average of Loss Functions
Stride for Convolutional Layer	1
Stride for Pooling Layer	2

Negative log-likelihood function is described by the following equation:

$$L(r, y) = - [y \cdot \ln r + (1 - y) \cdot \ln(1 - r)],$$

where r is an obtained probability with convolutional neural network for each of 43 classes; y is a true probability for each of 43 classes.

Cost function is described by the following equations:

$$J(w,b) = \frac{1}{m} \sum_{i=1}^{m} l(r^{(i)}, y^{(i)}),$$

$$j(w,b) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \cdot \ln r^{(i)} + (1 + y^{(i)}) \cdot \ln(1 - r^{(i)})],$$

where w, b are the weights and biases of the output fully connected layer; m is a number of iterations.

There is another important parameter that is directly related to the processing of the input image boundaries. This parameter is a zero frame (also called zero-padding frame) created around the input image before being sent to the convolutional layer. In this study, this parameter is linearly dependent on the dimension of the convolutional layer filters and is calculated by the following equation:

$$pad = \frac{d-1}{2},$$

where d is the dimension of the convolutional layer filters.

Since this study analyzes the dependence of the dimensions of the convolutional layer filters on the accuracy and rate of classification of traffic signs, the zero-padding frame parameter is very important. For example, the convolutional layer filters with 9×9 dimension process an input image with zero padding frame of 4, that is, the input image dimension is increased from 32×32 to 36×36. Consequently, the image border, namely, the extreme pixels of the input image 32, 31, 30, etc., are processed with filters to an additional depth of 4 pixels. With gradual increase of the dimension of the convolutional layer filters, the size of the zero-padding frame and the processing depth for the image borders by the filters will increase. This processing makes it possible not to miss the data located on the border of the image, especially in cases where important information on the image has been cropped.

Experimental results

In this study, training of the convolutional neural network is performed using various dimensions of convolutional layer filters, namely 3×3, 5×5, 9×9, 13×13, 15×15, 19×19, 23×23, 25×25 and 31×31. Consequently, in total, 9 models are trained with the same architecture, but with different filter dimensions. Training is conducted 9 times for every model on the preprocessed GTSRB training dataset with 50000 examples of traffic signs using Python v3 and pure "numpy" library. The training process for every model with its own dimension of the convolutional layer filters consists of 9000 iterations divided into 5 epochs. Also, a training dataset is divided into small batches of 50 examples that are fed into convolutional neural network simultaneously. At the end of every epoch the accuracy is calculated on the training dataset and on the validation dataset. For calculating accuracy, one thousand examples are randomly taken from the training dataset and validation dataset respectively. At the end of the first epoch all current model parameters are written into a file. If in the next epoch validation accuracy is higher than in the previous epoch, the parameters are updated. In this way, after the training process is finished, every model will have the best parameters according to the validation accuracy. Fig. 2 and Fig. 3 show the accuracy data comparison in the training process.

After the 9 models are trained, the accuracy is checked on the testing dataset. This dataset consists of 12000 images that did not participate in the training process. Every model for this operation is loaded with its own found best parameters from the saved file after training. Summary results are shown in Table 2.

The testing process is as follows. The 12000 images are fed to the input of the developed and trained convolutional neural network, and the result is written to a vector. This vector consists of 12000 class numbers of traffic signs classified by convolutional neural network. Further, the obtained classes with convolutional neural network are compared with true classes. The result is converted into the accuracy between 0 and 1. The described process is applied to any and all 9 models with their own filter dimension.

As is clear from Table 2, in accordance with the testing accuracy the best result is obtained with the model where the dimension of the convolutional layer filters is 19×19 pixels and the closest is 9×9.

The testing process is as follows. The 12000 images are fed to the input of the developed and trained convolutional neural network, and the result is written to a vector. This vector consists of 12000 class numbers of traffic signs classified by convolutional neural network. Further, the obtained classes with convolutional neural network are compared with true classes. The result is converted into accuracy between 0 and 1. The described process is applied to all 9 models with their own filter dimension. Convolutional layer filters can be visualized to see the changes from the initial state when they are initialized randomly and the final state when the training process is completed. Fig. 4 and Fig. 5 show comparison of initialized filters and trained filters for the model with dimension of the filters equal to 19×19.

Fig. 4 shows that the initialized filters are a chaotic set of pixels of different colours. After training, the filters have specific characteristics in the form of lines, curves, waves, dots, etc. These specific filters are being looked for in the input image and, in case of finding them, the maximum response is given, which is written in the appropriate place of the feature map.
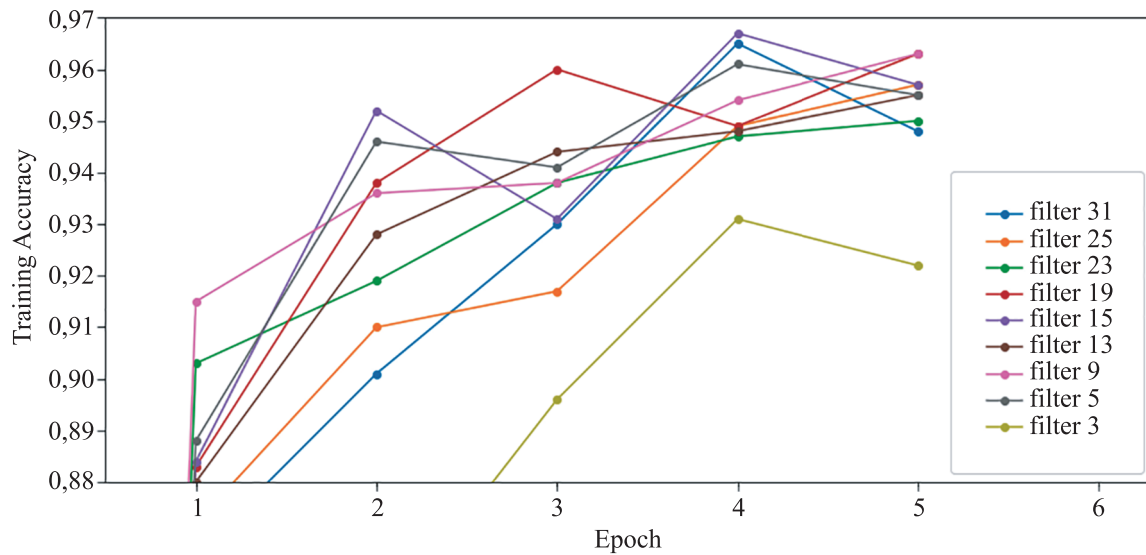
Fig. 2. Training accuracy of models with different dimension of convolutional layer filters
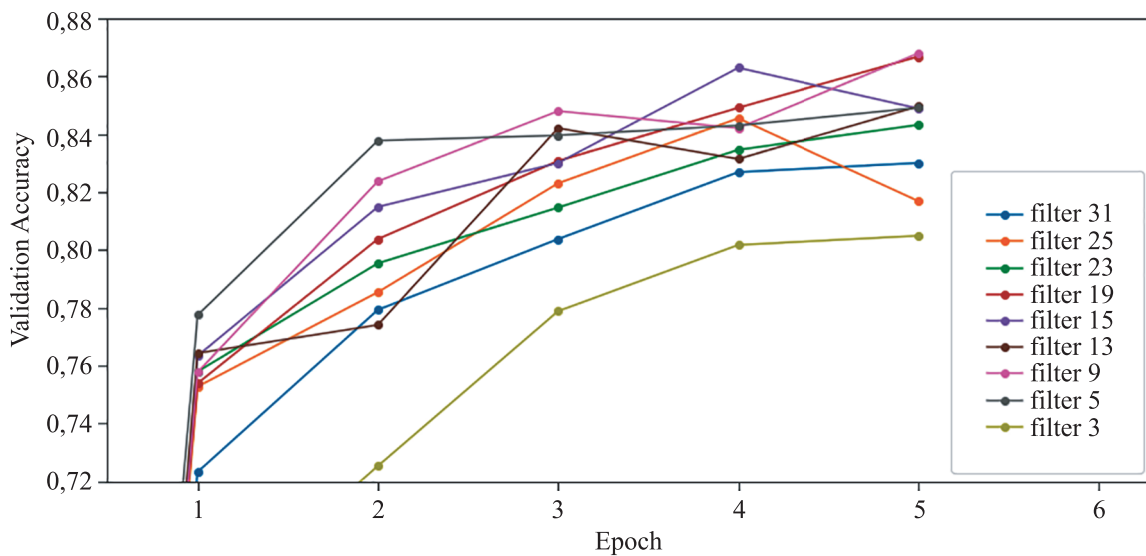

Fig. 3. Validation accuracy of models with different dimension of convolutional layer filters

Table 2. Summarized results for accuracy of every model

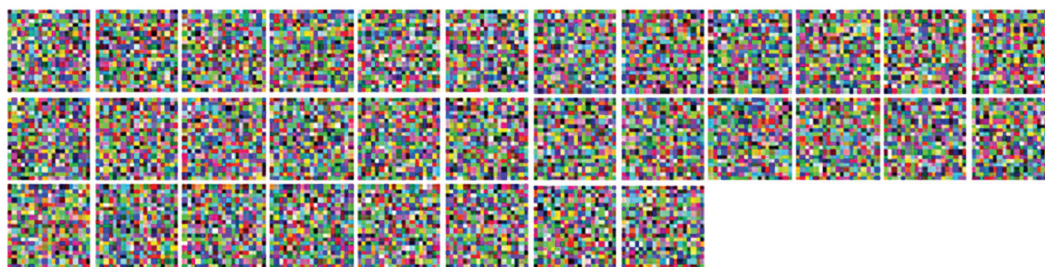| Model	Training Accuracy	Validation Accuracy	Testing Accuracy
31 × 31	0.965	0.83	0.843
25 × 25	0.957	0.846	0.851
23 × 23	0.95	0.843	0.846
19 × 19	0.963	0.867	0.868
15 × 15	0.967	0.863	0.86
13 × 13	0.955	0.85	0.854
9 × 9	0.963	0.868	0.864
5 × 5	0.961	0.849	0.848
3 × 3	0.931	0.805	0.828

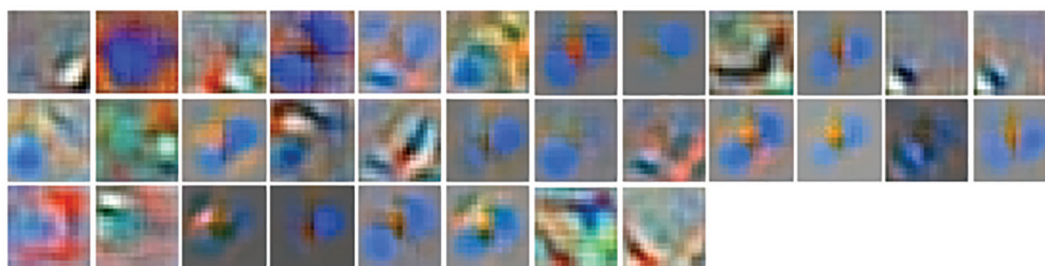*Fig. 4.* Initialized filters for 19×19 model

Fig. 5. Trained filters for 19×19 model

In addition to the accuracy, the image classification rate for each of the 9 models with their own filter dimensions is also estimated. Experimental results are presented in Table 3.

Table 3. The rate of image classification of every model

Model	Classification Time, s
31 × 31	0.005419
25 × 25	0.003651
23 × 23	0.004366
19 × 19	0.002786
15 × 15	0.003181
13 × 13	0.002364
9 × 9	0.004472
5 × 5	0.001879
3 × 3	0.002046

The experiments were carried out on a 64-bit laptop with i3 microprocessor with 4 cores, and 4 GB of RAM. As is clear from Table 3 the highest classification rate was shown by the model with convolutional layer filters of dimension equal to 5 × 5 pixels.

Conclusion

This paper studies the implementation of the classification algorithm for traffic signs based on convolutional neural network. The main contribution is analysis of the effect that convolutional layer filter dimensions have on classification accuracy and rate of traffic signs. The efficiency of the developed algorithm is evaluated on the GTSRB dataset.

Experimental results show that the use of convolutional layer filters with dimension of 9 × 9 and 19 × 19 gives the best accuracy of 0.864 and 0.868 respectively when tested on the testing dataset. The use of convolutional layer filters with 5 × 5 dimension gives the best rate of classification. At the same time, the rate of classification applying convolutional layer filters with 9 × 9 and 19 × 19 dimensions is 0.004472 and 0.002786 seconds, respectively, and enables their usage in real time applications.

For the future studies, we are planning to research the effect of the number of convolutional layers on the classification accuracy. In addition, it is planned to use convolutional neural networks not only for classification, but also for detection of traffic signs.

Литература

1. Balali V., Ashouri Rad A., Golparvar-Fard M. Detection, classification, and mapping of U.S. traffic signs using google street view images for roadway inventory management // Visualization in Engineering. 2015. V. 3. N 1. doi: 10.1186/s40327-015-0027-1
2. Lu Y., Lu J., Zhang S., Hall P. Traffic signal detection and classification in street views using an attention model // Computational Visual Media. 2018. V. 4. N 3. P. 253–266. doi: 10.1007/s41095-018-0116-x
3. Balali V., Golparvar-Fard M. Segmentation and recognition of roadway assets from car-mounted camera video streams using a scalable non-parametric image parsing method // Automation in Construction. 2016. V. 49. P. 27–39. doi: 10.1016/j.autcon.2014.09.007
4. Khalilikhah M., Heaslip K. The effects of damage on sign visibility: an assist in traffic sign replacement // Journal of Traffic and Transportation Engineering. 2016. V. 3. N 6. P. 571–581. doi: 10.1016/j.jtte.2016.03.009
5. Kryvinska N., Poniszewska-Maranda A., Gregus M. An approach towards service system building for road traffic signs detection and recognition // Procedia Computer Science. 2018. V. 141. P. 64–71. doi: 10.1016/j.procs.2018.10.150
6. Khalilikhah M., Heaslip K. Analysis of factors temporarily impacting traffic sign readability // International Journal of Transportation Science and Technology. 2016. V. 5. N 2. P. 60–67. doi: 10.1016/j.ijtst.2016.09.003
7. Shustanov A., Yakimov P. CNN design for real-time traffic sign recognition // Procedia Engineering. 2017. V. 201. P. 718–725. doi: 10.1016/j.proeng.2017.09.594
8. Indolia S., Kumar Goswami A., Mishra S.P., Asopa P. Conceptual understanding of convolutional neural network – a deep learning approach // Procedia Computer Science. 2018. V. 132. P. 679–688. doi: 10.1016/j.procs.2018.05.069
9. Ozturk S., Akdemir B. Effects of histopathological image pre-processing on convolutional neural networks // Procedia Computer Science. 2018. V. 132. P. 396–403. doi: 10.1016/j.procs.2018.05.166
10. Kurniawan J., Syahra S.G.S., Dewa C.K., Afiahayati. Traffic congestion detection: learning from CCTV monitoring images using convolutional neural network // Procedia Computer Science. 2018. V. 144. P. 291–297. doi: 10.1016/j.procs.2018.10.530
11. Aghdam H.H., Heravi E.J., Puig D. A practical approach for detection and classification of traffic signs using Convolutional Neural Networks // Robotics and Autonomous Systems. 2016. V. 84. P. 97–112. doi: 10.1016/j.robot.2016.07.003
12. Stallkamp J., Schlipsing M., Salmen J., Igel C. The German traffic sign recognition benchmark: a multi-class classification competition // Proc. Int. Joint Conference on Neural Networks. San Jose, USA, 2011. P. 1453–1460. doi: 10.1109/IJCNN.2011.6033395
13. Houben S., Stallkamp J., Salmen J., Schlipsing M., Igel C. Detection of traffic signs in real-world images: the german traffic sign detection benchmark // Proc. Int. Joint Conference on Neural Networks. Dallas, USA, 2013. P. 1–8. doi: 10.1109/IJCNN.2013.6706807
14. Eckle K., Schmidt-Hieber J. A comparison of deep networks with ReLU activation function and linear spline-type methods // Neural Networks. 2019. V. 110. P. 232–242. doi: 10.1016/j.neunet.2018.11.005
15. Lin G., Shen W. Research on convolutional neural network based on improved Relu piecewise activation function // Procedia Computer Science. 2018. V. 131. P. 977–984. doi: 10.1016/j.procs.2018.04.239

References

1. Balali V., Ashouri Rad A., Golparvar-Fard M. Detection, classification, and mapping of U.S. traffic signs using google street view images for roadway inventory management. *Visualization in Engineering*, 2015, vol. 3, no. 1. doi: 10.1186/s40327-015-0027-1
2. Lu Y., Lu J., Zhang S., Hall P. Traffic signal detection and classification in street views using an attention model. *Computational Visual Media*, 2018, vol. 4, no. 3, pp. 253–266. doi: 10.1007/s41095-018-0116-x
3. Balali V., Golparvar-Fard M. Segmentation and recognition of roadway assets from car-mounted camera video streams using a scalable non-parametric image parsing method. *Automation in Construction*, 2016, vol. 49, pp. 27–39. doi: 10.1016/j.autcon.2014.09.007
4. Khalilikhah M., Heaslip K. The effects of damage on sign visibility: an assist in traffic sign replacement. *Journal of Traffic and Transportation Engineering*, 2016, vol. 3, no. 6, pp. 571–581. doi: 10.1016/j.jtte.2016.03.009
5. Kryvinska N., Poniszewska-Maranda A., Gregus M. An approach towards service system building for road traffic signs detection and recognition. *Procedia Computer Science*, 2018, vol. 141, pp. 64–71. doi: 10.1016/j.procs.2018.10.150
6. Khalilikhah M., Heaslip K. Analysis of factors temporarily impacting traffic sign readability. *International Journal of Transportation Science and Technology*, 2016, vol. 5, no. 2, pp. 60–67. doi: 10.1016/j.ijtst.2016.09.003
7. Shustanov A., Yakimov P. CNN design for real-time traffic sign recognition. *Procedia Engineering*, 2017, vol. 201, pp. 718–725. doi: 10.1016/j.proeng.2017.09.594
8. Indolia S., Kumar Goswami A., Mishra S.P., Asopa P.. Conceptual understanding of convolutional neural network – a deep learning approach. *Procedia Computer Science*, 2018, vol. 132, pp. 679–688. doi: 10.1016/j.procs.2018.05.069
9. Ozturk S., Akdemir B. Effects of histopathological image pre-processing on convolutional neural networks. *Procedia Computer Science*, 2018, vol. 132, pp. 396–403. doi: 10.1016/j.procs.2018.05.166
10. Kurniawan J., Syahra S.G.S., Dewa C.K., Afiahayati. Traffic congestion detection: learning from CCTV monitoring images using convolutional neural network. *Procedia Computer Science*, 2018, vol. 144, pp. 291–297. doi: 10.1016/j.procs.2018.10.530
11. Aghdam H.H., Heravi E.J., Puig D. A practical approach for detection and classification of traffic signs using Convolutional Neural Networks. *Robotics and Autonomous Systems*, 2016, vol. 84, pp. 97–112. doi: 10.1016/j.robot.2016.07.003
12. Stallkamp J., Schlipsing M., Salmen J., Igel C. The German traffic sign recognition benchmark: a multi-class classification competition. *Proc. Int. Joint Conference on Neural Networks*. San Jose, USA, 2011, pp. 1453–1460. doi: 10.1109/IJCNN.2011.6033395
13. Houben S., Stallkamp J., Salmen J., Schlipsing M., Igel C. Detection of traffic signs in real-world images: the german traffic sign detection benchmark. *Proc. Int. Joint Conference on Neural Networks*. Dallas, USA, 2013, pp. 1–8. doi: 10.1109/IJCNN.2013.6706807
14. Eckle K., Schmidt-Hieber J. A comparison of deep networks with ReLU activation function and linear spline-type methods. *Neural Networks*, 2019, vol. 110, pp. 232–242. doi: 10.1016/j.neunet.2018.11.005
15. Lin G., Shen W. Research on convolutional neural network based on improved Relu piecewise activation function. *Procedia Computer Science*, 2018, vol. 131, pp. 977–984. doi: 10.1016/j.procs.2018.04.239

Authors

Valentyn N. Sichkar — postgraduate, software engineer, ITMO University, Saint Petersburg, 197101, Russian Federation, ORCID ID: 0000-0001-9825-0881, vsichkar@itmo.ru

Sergey A. Kolyubin — PhD, Associate Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 35303066700, ORCID ID: 0000-0002-8057-1959, s.kolyubin@itmo.ru

Авторы

Сичкар Валентин Николаевич — аспирант, программист, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, ORCID ID: 0000-0001-9825-0881, vsichkar@itmo.ru

Колюбин Сергей Алексеевич — кандидат технических наук, доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 35303066700, ORCID ID: 0000-0002-8057-1959, s.kolyubin@itmo.ru