CLASSIFICATION OF TRAFFIC SIGNS BY CONVOLUTIONAL NEURAL NETWORKS

Filip Mivalt, Petr Nejedly
Master Degree Programme (1), FEEC BUT
E-mail: xmival00@stud.feec.vutbr.cz, xnejed07@stud.feec.vutbr.cz

Supervised by: Jiri Chmelik
E-mail: chmelikj@feec.vutbr.cz

Abstract: The paper presented here describes traffic signs classification method based on a convolutional neural network (CNN). The CNN was trained and tested on the public database of German traffic signs with 43 mostly used traffic sign types. Proposed technique achieved overall classification F1 score 89.97 percent on a hidden testing dataset.

Keywords: Machine learning, Convolutional neural networks, Traffic signs recognition

1 INTRODUCTION

Convolutional neural networks were designed by LeCun in 90's [1] for image classification purpose. After it's invention, they were not widely used due to their computational complexity. However, in last few years, CNNs increased its popularity due to an improvement of graphics processing units (GPU), which are used for the computationally expensive training process. Nowadays, CNNs are used as the state of the art for the image classification among different scientific and industrial fields. For example, CNNs are the state of the art for the object detection in self-driving cars. Moreover, in last few years, CNNs started to be also used in a medical applications.

The architecture of the CNNs is biologically inspired from visual cortex neurons that produce a visual perception. This inspiration originally comes from neuroscience research [2], which introduced hypothesis of specialized neurons that are active only in presence of the specified pattern (e.g. horizontal or diagonal edges).

2 TRAINING DATA

For the learning process, we have used traffic signs from the public German database [3], which consist of forty thousand examples in 43 classes. There is shown few training examples of some

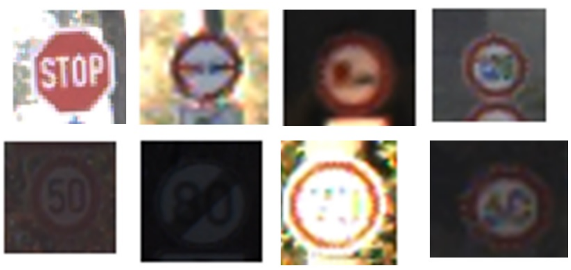

Figure 1: Examples of training images from The German Traffic Sign Recognition Benchmark training dataset [3].

categories in Figure 1. The distribution of training examples through all classes is not equal. Some classes contain more than 2,000 training examples but others contain just 200 examples. Traffic signs were recorded from a moving car in different conditions (sunlight, night, sunrise, raining, snow, etc.). This results in a fact that each traffic sign is on multiple training images but acquired from different distances. The proposed database consists of RGB 24bit images with random size (minimum 15x15 pixels and maximum 250x250 pixels). Images were resized to 50x50 pixels and split into a training set (70%) and hidden testing set (30%).

3 METHODS

The convolutional neural network is a standard feed-forward neural net with weight sharing, that reduce the number of learning parameters. In general, architecture consists of convolutional kernels, non-linearity mapping layers and max-pooling layers. The CNN extracts features from the original image by a convolution operation between the original image and convolutional kernel. Further, the non-linearity mapping layer is used, which helps with the classification of the linearly inseparable data. Next, the max-pooling layer (a region from feature space is downsampled and replaced by its maximal value) is used to introduce a translational invariance. This three-layer structure is repeated several times in order to create a system for the complex features extraction. For example, the first convolutional layer extracts basic features like edges. Further, convolutional layers combine basic features from the previous layer to form more specialized patterns. Lastly, several fully connected layers and soft-max layer are connected to form probability neural network output. Output probability is computed by the soft-max function (1), where z_k is output k from the last fully connected layer

$$p_j(\vec{z}) = \frac{e^{z_j}}{\sum_{k=0}^{K-1} e^{z_k}}, \tag{1}$$

where z_j is j-th activation function value of K-dimensional vector z. An objective of the CNN learning process is to minimize of the mini-batch cross-entropy (2) by stochastic gradient descent with momentum [1]

$$J(\theta) = -\sum_{n=0}^{N-1}\sum_{k=0}^{K-1} t_{nk} ln(p_k(x_n)), \tag{2}$$

where $J(\theta)$ is cross-entropy cost function, θ is learnable parameter vector, t_{nk} is binary indicator for target value and $p_k(x_n)$ is probability of output k for n-th mini-batch sample x_n. K is number of outputs from the CNN and N is mini-batch size

$$\theta_{i+1} = \theta_i - \alpha\nabla J(\theta) - \gamma(\theta_i + \theta_{i-1}), \tag{3}$$

where θ_{i+1} is parameter vector for next iteration $i+1$, α is learning rate and γ is momentum parameter.

The proposed CNN consists of pair convolution layers and followed by pair fully-connected layers. The network was chosen as a compromise between a prediction accuracy, task complexity and rate of the sign evaluation. The first layer extracts basic features. The proposed architecture is using mini-batch normalization for the faster learning process and also as protection against over-fitting. Also, L2 regularization technique and dropout layer are used as over-fitting protection. Figure 2 shows the architecture of the proposed CNN.

4 RESULTS AND DISCUSSION

The proposed CNN architecture achieved a mean F1 score on hidden dataset 89.97 percent and 98.15 on a training dataset. The interval of F1 scores across classes was on a range from 65.17 for class with

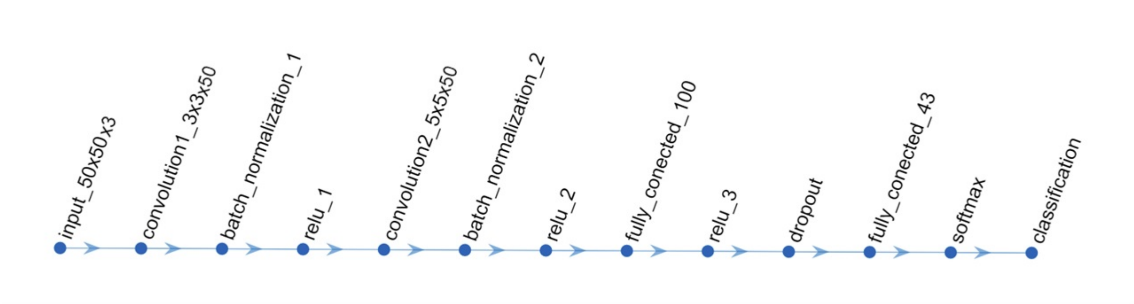

Figure 2: Architecture of proposed convolutional neural network.

the smallest amount of training examples to 99.07 for classes with more than 2000 training images. Increasing the number of training examples in all categories, or at least in categories with a small number of images, could improve the final result. For the computer in the following configuration: ASUS GeForce GTX 1060, 6 GB GDDR5, Intel Core i7-920, 4/8 cores/threads, Core frequency: 2,67 GHz, 24 GB RAM, algorithm achieved 9.2 ms average evaluation time for one classification task. However, we have to point out some drawbacks of this method. For example, classification works only in a situation with single traffic sign in the image. Secondly, the position of a traffic sign is not estimated. On the other hand, this problem might be solved by regionally based methods R-CNN or with YOLO CNN detection technique.

5 CONCLUSION

This paper introduced the traffic sign classification based on convolutional neural networks, which are used as a state of the technique for the image classification in self-driven cars. The proposed CNN was tested on a hidden dataset and achieved an overall F1 score equal to 89.97 percent. In further research, we would like to explore applications of regional based convolutional neural networks (R-CNN) on a real-time localization of traffic signs, vehicles and pedestrians.

REFERENCES

[1] Yann LeCun , Yoshua Bengio, Convolutional networks for images, speech, and time series, The handbook of brain theory and neural networks, MIT Press, Cambridge, MA, 1998

[2] Hubel DH, Wiesel TN. Receptive fields of single neurones in the cat's striate cortex. J Physiol. 1959;148:574–591.

[3] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition, Neural Networks, Available online 20 February 2012, ISSN 0893-6080, 10.1016/j.neunet.2012.02.016.