# Traffic Sign Recognition using Convolutional Neural Networks

## Introduction

Recent years automotive industry is sifted more and more towards autonomous driving technologies. These technologies include, for example, cameras that can be used to detect and classification of different objects on the road. The most critical step for safer autonomous vehicles is to recognize traffic signs. This paper will focus on the traffic sign recognition performance using three different convolutional neural network (CNN) architectures. The three different CNN architectures that are used in this work are LeNet [1], AlexNet [2], and VGGNet [3].

## Problem Formulation

The traffic sign recognition problem can be modelled as machine learning (ML) problem with datapoints representing individual traffic signs. Each traffic sign is characterized by different spectral bands (by different colors) of traffic signs as its features. The type of the traffic sign, such as stop or different speed limit signs, is the label of a datapoint.

Datapoints were gathered from the German Traffic Sign Recognition Benchmark (GTSRB) dataset [4]. The GTSRB dataset consists of 39209 training images, which are divided to 43 different classes. These classes include different speed limit, dangerous, and order traffic signs. The GTSRB dataset consists consist also, 12629 unlabeled testing images for evaluating the performance and quality of the different CNN architectures.

Also, the best performing CNN model was tested the dataset [5] that was created by the writer of this report. The dataset was created by taking images of different traffic signs, and then classifying these to the correct classes. This dataset includes 48 training images, which are divided to 4 different classes equally. These classes are no parking, triangle, walkway, and pedestrian crossing traffic signs. The dataset also includes 10 different unlabeled testing images.

# Methods

This problem is multi-class classification problem, which means that the ML model is crated for classifying each image to correct class by the feature vector obtained from the image [6]. Therefore, the convolutional networks were chosen to this problem. These CNN's are regularized versions of multilayer perceptron's, which are fully connected networks. This means that neuron in one layer is connected to all neurons in the next layer. CNN architectures include three main types of layers, which are convolutional layer, pooling layer, and fully connected layer. These layers are connected in different ways depending on the architecture of the CNN. [7]

In this work three different CNN architectures were used. These networks were LeNet, AlexNet, and VGGNet. All these models were constructed as shown in the reference articles [1,2,3]. Although the hyperparameters in the networks were tuned to correspond the multi-class classification problem shown in this paper. Basically, LeNet is the simplest network including three convolutional layers, which are followed by the pooling layers. The activation function in these convolutional layers is rectified linear activation function (ReLu). After these layers, the fully connected layers with SoftMax activation follows as output layers. AlexNet's structure is like LeNet architecture, but it is more deeper including more layers with filters, stacked convolutional layers, and max pooling. All in all, AlexNet has 5 similar convolution layers and 3 fully connected layers as LeNet. The problem of AlexNet is that it comprises of too many hyper-parameters, with this deeper architecture. Finally, the VGGNet is the network that solves the problem of including too many hyper-parameters. The architecture of VGGNet is like other CNN networks in this paper, and it includes overall 3 convolutional layers with 3 convolutional layer blocks followed by max pooling layer.

These networks were implemented with Python library of TensorFlow, which are good tools to solve these kinds of problems [4, 8]. This is because, TensorFlow include good tools to implement easily different kind of neural networks, such as CNN. Building of these networks started to divide the classified training data to training and validation test with split of 80% to training set and 20% to test set. After this, corresponding CNN networks was built. All these different networks were optimized using Adam-optimizer, with learning rate of 0.001. The cross-entropy loss was used to measure the performance of a classification [6]. After this the network was iterated over 15 epochs, and the results were plotted to accuracy and loss figures. Finally, the accuracy of the model was calculated respect to the test sets.

# Results

The accuracy and loss over every 15 epochs of these three different networks are shown in figure 1. The figure 1 also, includes the accuracy and loss plots to the writer's own dataset with VGGNet, since it was best performing network.

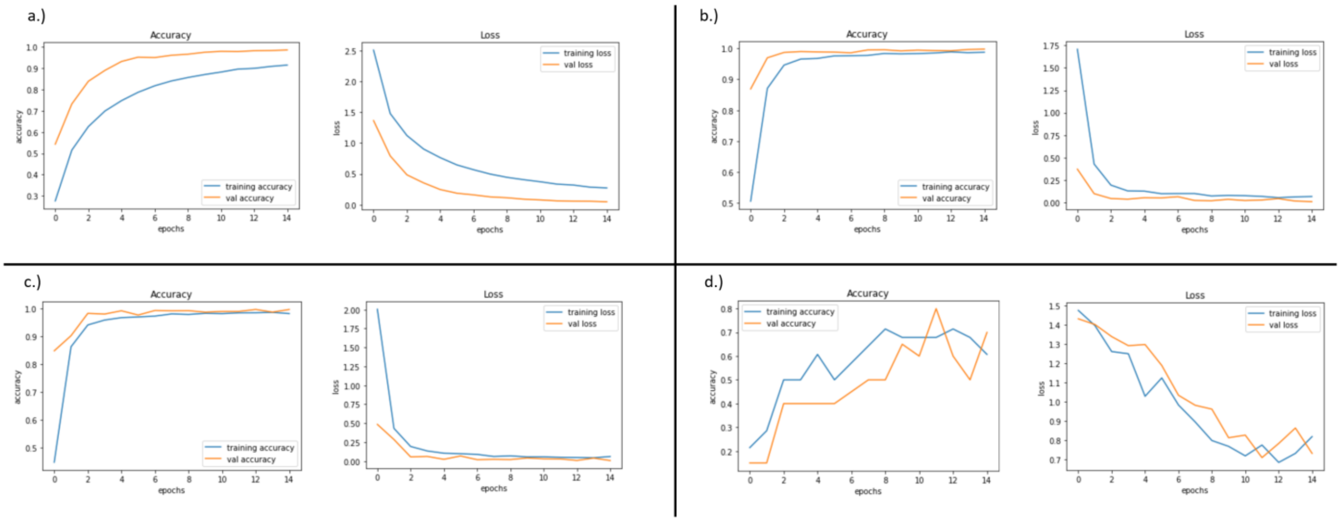


Figure 1. The accuracy and loss over every 15 epochs for training and validation data with a.) LeNet model b.) AlexNet model c.) VGG-16 model d.) VGGNet model with dataset created by writer.

As, one can see from the figure 1.b, and 1.c that the AlexNet, and VGGNet are performing similarly to each other. The LeNet (figure 1.a) architecture does not perform as well as AlexNet or VGGnet but the model functions similarly. The worst performing network is the VGGNet with writer's own dataset, as shown in figure 1.d. Overall, all these models appear to be quite well generalized and not overfit the training data. Table 1 shows the accuracies of these models with the test datasets.

Table 1. Accuracy of different networks with test dataset.

| Model | LeNet | AlexNet | VGGNet | VGGNet (writer's dataset) |
|---|---|---|---|---|
| Accuracy | 0.936 | 0.970 | 0.973 | 0.894 |

The accuracy of VGGNet is best as shown in the table 1. Also, it is important to see that the accuracy of AlexNet is very close to the VGGNet. LeNet, and VGGNet with writer's own data performed very similarly. Little bit worser performance of LeNet can be explained by the fact that it is quite simple network when compared to other networks. On the other hand, the lower accuracy of VGGNet with writer's own data can be explained by the fact that the dataset is so small (only 48 images in training set) compared to other networks.

# Conclusion

In this paper, three different convolutional neural network architectures, LeNet, AlexNet, and VGGNet, were studied to predict correct class of different traffic signs. All these three different networks performed well, as one can see from figure 1.a, 1.b, and 1.c. The best out of these three networks was the VGGNet, which was expected since it solves the AlexNet's problem of including too many hyper-parameters and is more precise than simpler LeNet. The worst performing network was VGGNet with writer's own dataset. Also, this behavior was expected since the dataset was so small, only including 12 images in each 4 classes. For future development, there is need to collect higher dataset of Finnish traffic signs since all reference traffic signs can be founded from Väylävirasto [9]. After this writer could install a camera with some microcomputer running with VGGNet to older car in order to recognize different traffic signs. Overall, the performance of LeNet, AlexNet, and VGGNet were good for classification of different traffic signs.

# References


1.) LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 86(11). 2278-2324. (1998).

2.) Krizhevsky, A., Sutskever, I., & Hinton, G. E. *Imagenet classification with deep convolutional neural networks*. Advances in neural information processing systems, 25, 1097-1105. (2012).

3.) Simonyan, K., & Zisserman, A. *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556. (2014).

4.) 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. (http://www.sciencedirect.com/science/article/pii/S0893608012000457) (read: 26.3.2021)

5.) Writer's image data and code templates for LeNet, AlexNet, and VGGNet architectures.
https://drive.google.com/drive/folders/1Wvc5NoJDzg5tMl4-mssGzPi6j6UlVHaq?usp=sharing
(Note: this google account is anonymous, so the owner's name is random.)

6.) A. Jung. Machine Learning. The Basics. 2021. mlbook.cs.aalto.fi.

7.) Satılmış, Y., Tufan, F., Şara, M., Karslı, M., Eken, S., & Sayar, A. CNN based traffic sign recognition for mini autonomous vehicles. In International Conference on Information Systems Architecture and Technology (pp. 85-94). (2018). Springer, Cham.

8.) Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*. arXiv preprint arXiv:1603.04467. (2016).

9.) Väylävirasto. *Suomen liikennemerkit QGIS-kuvakirjastona.* (read: 28.3.2021)