**Road Traffic Sign Recognition System**

Akhil Namburi

**Abstract**

Traditionally, standard computer vision methods were employed to detect and classify traffic signs, but these required considerable and time-consuming manual work to handcraft important features in images. Instead, by applying deep learning to this problem, we create a model that reliably classifies traffic signs, learning to identify the most appropriate features for this problem by itself. In this project, we show how we can create a deep learning architecture that can identify traffic signs with accuracy. The capability of recognition of a neural network increases with increasing the training accuracy. In order to justify the effectiveness, different test patterns of the signs are used to verify the system.

**Objective**

Traffic sign classification has become a mature area with the increasing focus on autonomous driving research. Notable research work exists on detection and classification traffic signs for advanced driver assistance systems. Most of those works attempted to address the challenges involved in real life problems due to scaling, rotation, blurring etc using computer vision and machine learning algorithms. Here, in this project we are implementing the same using Convolutional Neural Networks. The main challenge in this project is to reduce the effect of spatial transformations of the images using Tensorflow model and also implementing batch normalization technique. So, by working on this project, we are confident of building a CNN for any provided image dataset.

**Architecture**

We used convolutional neural network for our project. Convolutional neural networks are biologically inspired multi-stage neural network architecture that learns the invariant features automatically. Each stage consists of a convolution layer, non-linear transform layer, spatial pooling layer. The spatial pooling layer deceases the spatial information and acts like a complex cells in visual cortex. Generally, a gradient descent based optimizer is used for training and updating each filter to minimize loss function. The output of all the layers is fed to the classifier for improving the accuracy of classification.

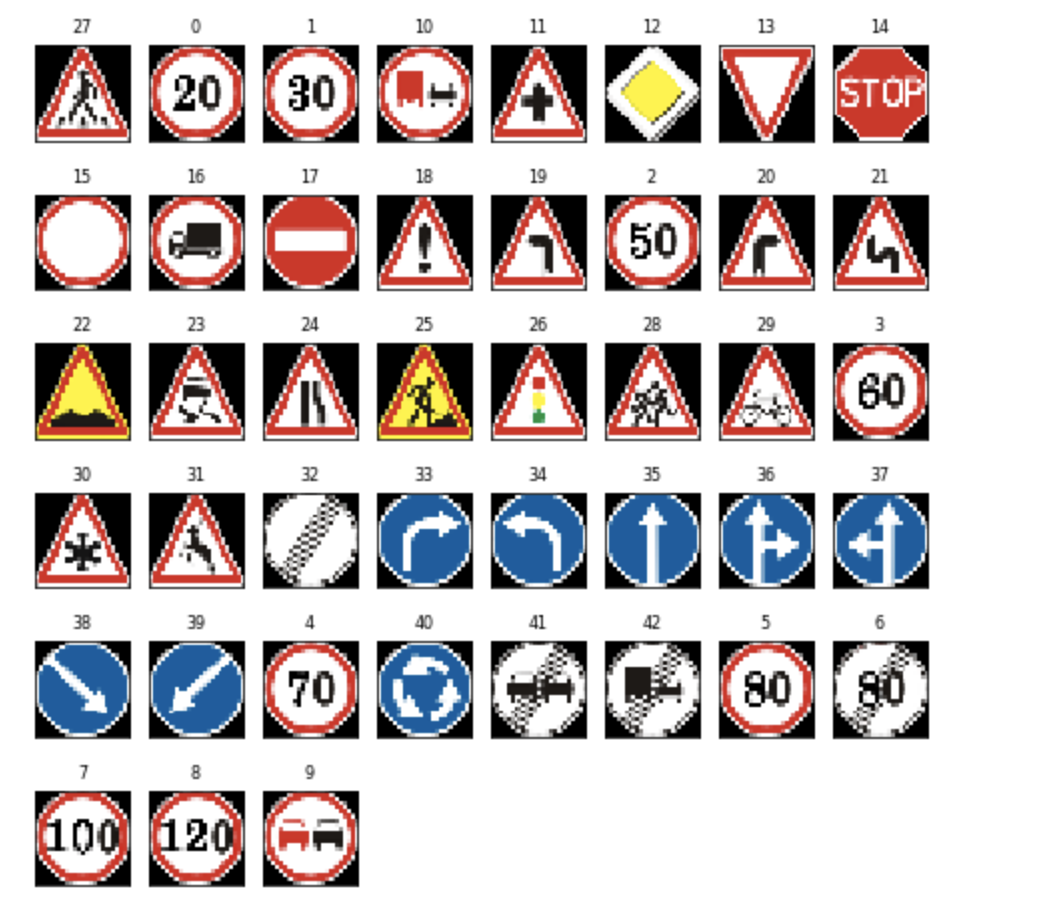
The dataset we are using is split into training, test and validation sets, with the following characteristics:

* Images are 25 (width) x 25 (height) x 3 (RGB color channels)
* Training set is composed of 31367 images
* Validation set is composed of 7842 images
* Test set is composed of 12630 images
* There are 43 classes

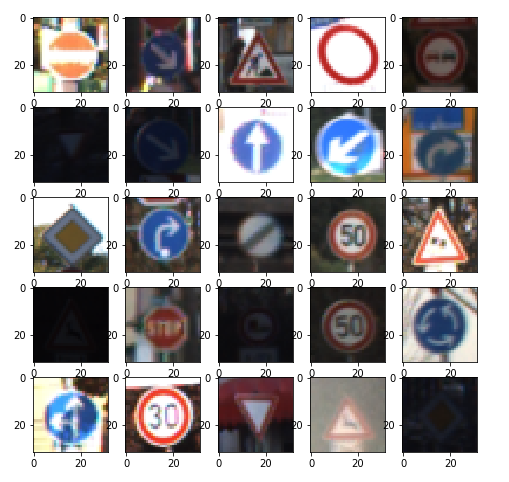
A screenshot of a cell phone

Description automatically generated

43 classes with class labels are shown in the below image.



The below are a few random sample images from the training set. In 5\*5 rows and columns.



The network is composed of 3 hidden layers with batch normalization, learning rate, step decay and dropout being implemented on each of the layers after activation, using techniques like ReLU or Softmax. The first layer is a convolutional layer in which the number of filters is 32 with kernel 5 x 5 and activation function as ReLU. The second layer is similarly a convolutional layer with the number of filters being 64 and the kernel size of 3 x 3 using the activation function as ReLU. The third layer is exactly similar to the second layer. The fourth layer is a fully connected layer and connects to 256 nodes. The fifth layer is similarly a fully connected layer and is also the final layer with input layer of size 256 and producing 43 results. This output layer uses Softmax as the activation function.

**Discussions**

We used CNN model Architecture which contains 3 Convolutional Layers and 2 Fully connected Layers. We added different sizes of filters over 5 Convolutional layers and after 10 epochs with Adam optimizer and with a batch size of 128 and dropout value 0.25, we were able to achieve the desired accuracy for the model. In order to prevent overfitting, we used a regularization technique called Dropout.

**Results**

The proposed network was trained and tested using the machine learning libraries, also using Adam optimizer and learning rate step decay techniques. In order to prevent overfitting, we used regularization techniques like Dropout. We were constantly working to generate a better model in terms of its accuracy, so we have been constantly changing the sizes of filters and using different dropout probabilities for different layers and also considering learning rates while performing batch normalization. The final model comprised of 5 layers, out of which there are 3 are hidden layers. There are two convolutional layers and a fully connected layer among those hidden layers. The end model results are as follows:

* Training set accuracy of 98.12%
* Validation set accuracy of 99.60%
* Test set accuracy of 96.18%

A close up of a map

Description automatically generated

**Conclusion**

In this paper, we worked on how deep learning can be used to classify the traffic signs with high accuracy, employing a variety of pre-processing techniques like adjusting the contrast and also performing image normalization and regularization techniques like dropout. We built a configurable code and made sure the accuracy spiked up with every epoch. We encountered overfitting and used regularization techniques to refine the model. Our model clocked an accuracy of 98.12% on training data and 96.18% on test data and 99.60% on validation data.

We enjoyed working on this project and gained practical knowledge using Tensorflow techniques and regularization techniques like batch normalization. Moreover, we had to be through with the previous research and the papers published on similar topics in order to gain a keen understanding of the model and the techniques being implemented. This project not only made us work practically build a model, but also refined our knowledge on deep learning.

**References**

[1] Wong, A., M. J. Shafiee, and M. St. Jules. 2018. “MicronNet: A Highly Compact Deep Convolutional Neural Network Architecture for Real-Time Embedded Traffic Sign Classification.” IEEE Access 6: 59803–10.

[2] Temel, D., M. Chen, and G. AlRegib. 2019. “Traffic Sign Detection Under Challenging Conditions: A Deeper Look into Performance Variations and Spectral Characteristics.” IEEE Transactions on Intelligent Transportation Systems, 1–11.

[3] Sermanet, Pierre, and Yann LeCun. 2011. “Traffic Sign Recognition with Multi-Scale Convolutional Networks.” In IJCNN, 2809–13.

[4] Haloi, Mrinal. 2015. “Traffic Sign Classification Using Deep Inception Based Convolutional Networks.” arXiv [cs.CV]. arXiv. <http://arxiv.org/abs/1511.02992>.

[5] Abdi, Lotfi, and Aref Meddeb. 2017. “Deep Learning Traffic Sign Detection, Recognition and Augmentation.” Proceedings of the Symposium on Applied Computing - SAC ’17. <https://doi.org/10.1145/3019612.3019643>.

[6] Cireşan, Dan, Ueli Meier, and Juergen Schmidhuber. 2012. “Multi-Column Deep Neural Networks for Image Classification.” arXiv [cs.CV]. arXiv. <http://arxiv.org/abs/1202.2745>.

[7] Arcos-García, Álvaro, Juan A. Álvarez-García, and Luis M. Soria-Morillo. 2018. “Deep Neural Network for Traffic Sign Recognition Systems: An Analysis of Spatial Transformers and Stochastic Optimisation Methods.” Neural Networks: The Official Journal of the International Neural Network Society 99 (March): 158–65.

[8] Lim, Kwangyong, Yongwon Hong, Yeongwoo Choi, and Hyeran Byun. 2017. “Real-Time Traffic Sign Recognition Based on a General Purpose GPU and Deep-Learning.” PloS One 12 (3): e0173317.

[9] Yasmina, Djebbara, Rebai Karima, and Azouaoui Ouahiba. 2018. “Traffic Signs Recognition with Deep Learning.” 2018 International Conference on Applied Smart Systems (ICASS). https://doi.org/10.1109/icass.2018.8652024.