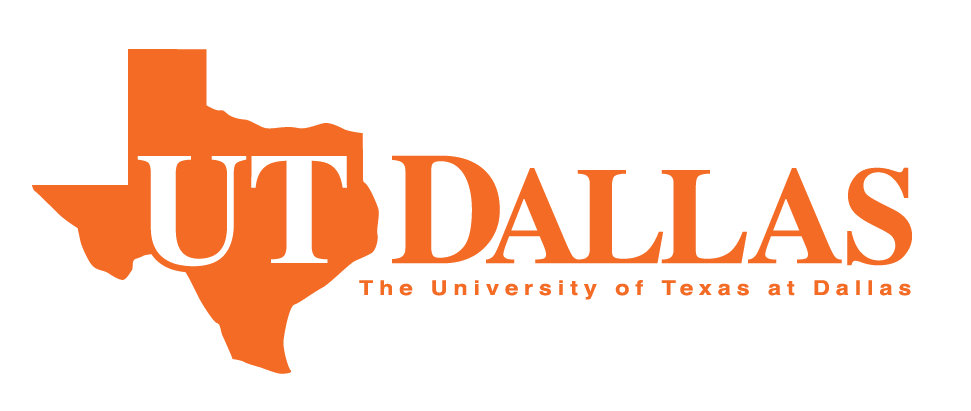
Project ReportTraffic Sign Classification using CLAHE  
: A Deep Learning Approach

short line  
  
  
 Research Assistant: Thouseef Syed  
 M.S. Applied Cognition & Neuroscience  
 School of Behavioural & Brain Sciences  
 The University of Texas, Dallas  
   
 Supervisor: Dr. Richard Golden  
 Head of Department  
 Applied Cognition & Neuroscience  
 School of Behavioural & Brain Sciences  
 The University of Texas, Dallas  


# Introduction

The world is full of objects. This report deals with the recognition of traffic signs. As far as travelling is concerned the traffic signs play a major role in orchestrating the flow of vehicles and pedestrians on the road. Object detection and object recognition have been around and are used interchangeably. They both are similar techniques for identifying objects, but they vary in their execution. Object detection is the process of finding instances of objects in images. In the case of deep learning, object detection is a subset of object recognition, where the object is not only identified but also located in an image. This allows for multiple objects to be identified and located within the same image. Therefore, traffic sign recognition is performed using Convolutional layers and based on the adaptation method, the results are produced.

# The Objective

The main objective is to perform traffic sign recognition in challenging environments. Firstly, the dataset is trained, validated and tested. Traditional testing parameters are used to validate the results.

# The Dataset

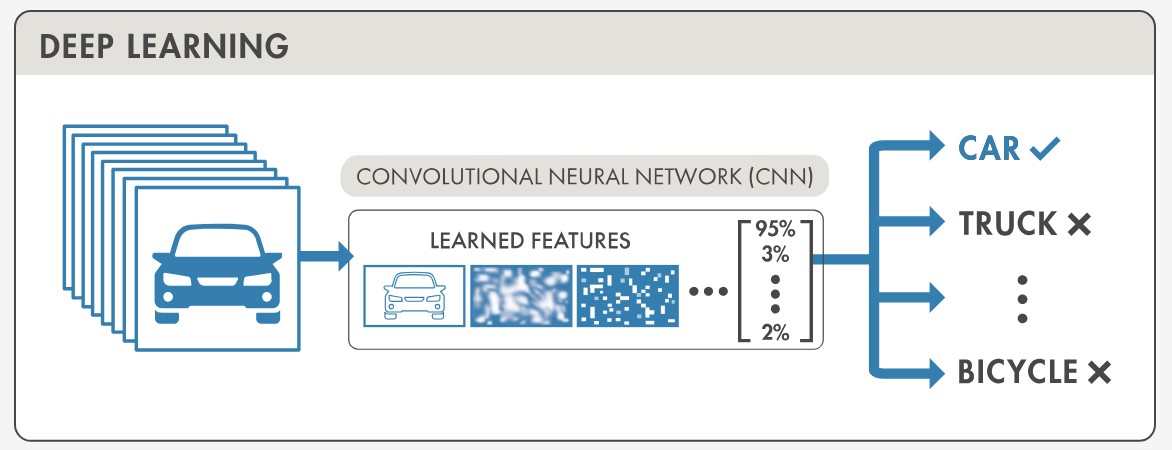
The dataset used is the [**German Traffic Sign Recognition Benchmark (GTSRB)**](https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign). This dataset is used to train and build our very own traffic sign classifier. It consists of 43 traffic sign classes and contains approximately 50,000 images.



# Recognition

**Object recognition:** Object recognition is a computer vision technique for identifying objects in images or videos.

* It is a key output of deep learning and machine learning algorithms. When humans look at a photograph or watch a video, we can readily spot people, objects, scenes, and visual details.
* The ultimate goal is to train a computer to visualize the environment and perform classification.



Traffic Sign Recognition has two parts:

* **Localization:** Detect and localize where in an input image/frame a traffic sign is located.
* **Recognition:** Take the localized ROI and actually recognize and classify the traffic sign.  
  \*Deep learning object detectors can perform localization and recognition in a single forward-pass of the network



# Convolutional Neural Network

Typical Convolutional Layer: For the purpose of image classification a **Convolutional Neural Network (CNN) i**s used

The neural network consists of the following layers:

1. Convolution layers
2. Rectifier Linear Unit   
   (Activation Function)
3. Max Pooling Layer

Finally, for image classification a **Fully Connected layer** is implemented.  
  
**Convolution layer:**

* The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image.
* Traditionally, the first Conv-Layer is responsible for capturing the **Low-Level features** such as **edges, color, gradient orientation**, etc.
* Therefore with additional layers, the architecture adapts to the **High-Level features** and extracts them accordingly.

**Rectified Linear Unit :**

ReLu is a non-linear activation function that is used in multi-layer neural networks or deep neural networks. This function can be represented as:



where x = an input value

According to equation 1, the output of ReLu is the maximum value between zero and the input value. An output is equal to zero when the input value is negative and the input value when the input is positive. Thus, we can rewrite equation 1 as follows:



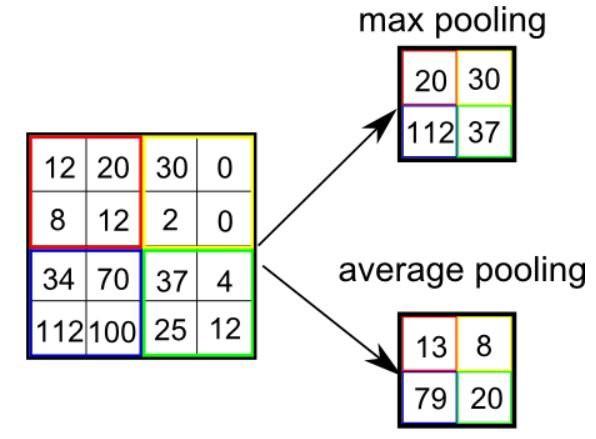
where x = an input value

**Pooling Layer :**

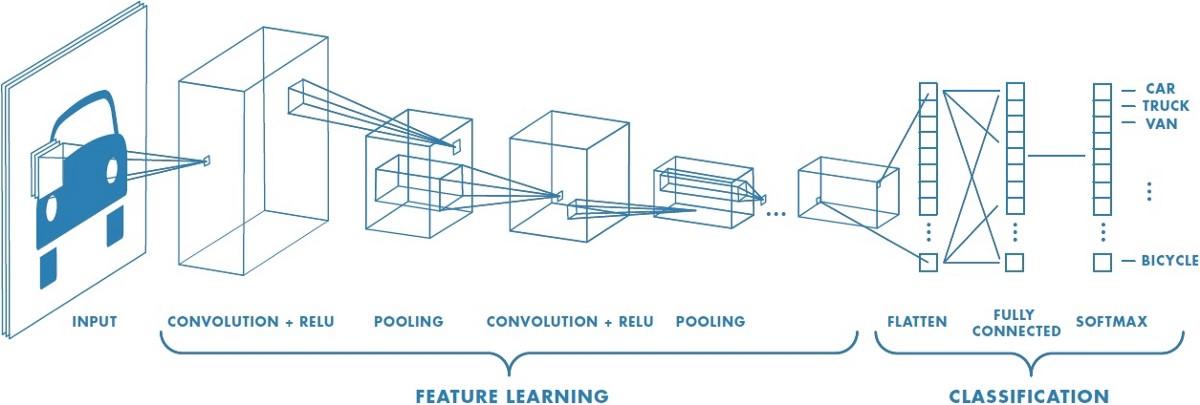
* The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. It is used to decrease the computational power.
* Furthermore, it is useful for extracting dominant features which are rotational and positional invariant,

There are two types of Pooling:

* **Max Pooling** returns the maximum value from the portion of the image covered by the Kernel.
* On the other hand, **Average Pooling** returns the average of all the values from the portion of the image covered by the Kernel.



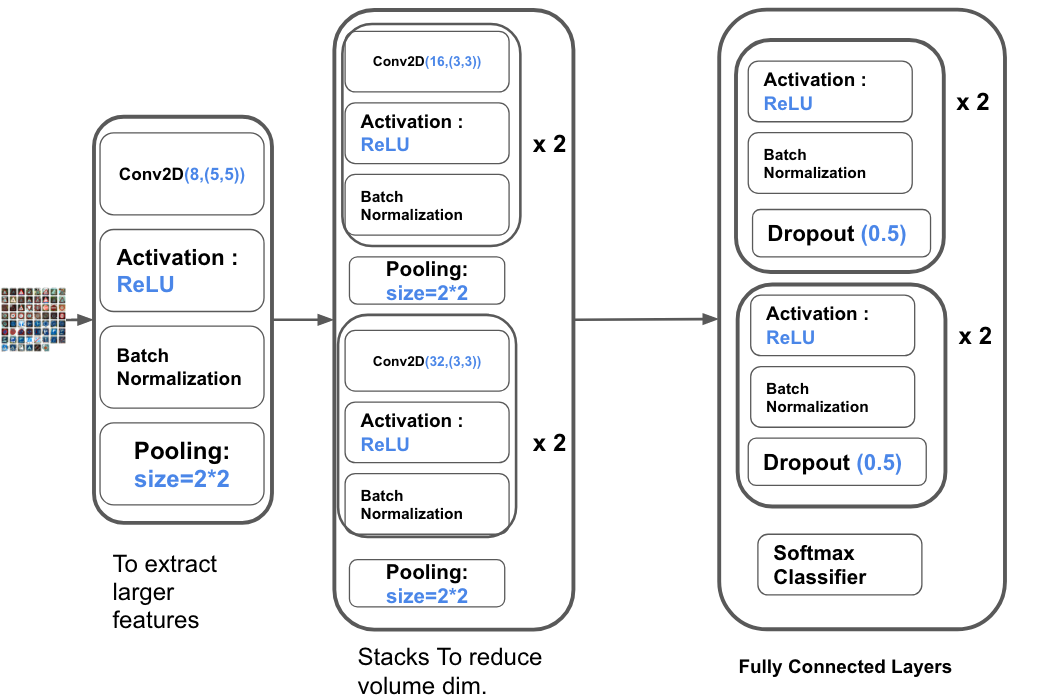
**Convolutional layer + Fully connected layer**



# Network Architecture

The network architecture of the **traffic sign classifier** is given as follows:

* First set of layers consist of a convolutional layer, ReLu layer, Batch Normalization and Pooling layer of size 2x2. This first set of layers are used to extract larger features that exist in the image.
* Second set of layers consists of two stacks of convolutional layer, ReLu layer, Batch Normalization and Pooling layer of size 2x2. This set of layers aid reducing the volume dimensionality.
* Finally, towards the end we have 2 fully connected layers along with Dropout and softmax classifier. Dropout is applied as a form of **regularization** which aims to prevent overfitting. The result is often a more generalizable model.



# Training

* First process is to load our training and testing split from the GTSRB dataset
* Followed by preprocessing the images for better recognition of the input image
* Next, the model is trained with close to 39,000 images.
* The model’s accuracy is evaluated using a set of parameters
* Finally the model is serialized to disk so we can later use it to make predictions on new traffic sign data.

# Preprocessing the data

Contrast limited adaptive histogram equalization **(CLAHE)** is used to improve the visibility level of foggy image or video. Many of the traffic signs that need to be recognized are of course detected but are misclassified due the noise existing in the image. Now, this noise or distortion is due to the environmental conditions like poor lighting,fog,mist,rain,etc.We can automatically improve image contrast by applying this algorithm. Therefore, using CLAHE we can improve the contrast of our traffic sign images:



# Results:

The results below are attached after training and testing the dataset. The model trained on **39,000** images and tested on nearly **12,000** images. It was trained for 30 epochs, evaluating the training loss, accuracy, validation loss and the validation accuracy. Therefore, the corresponding traffic signs were evaluated on parameters like precision, recall, f-1 and support.  
**Tools used:** Python Deep Learning libraries were used to implement the project.

2020-05-07 00:12:20.511943: I tensorflow/core/platform/cpu\_feature\_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2

[INFO] training network...

Epoch 1/30

612/612 [==============================] - 65s 107ms/step - loss: 2.5107 - accuracy: 0.3277 - val\_loss: 3.8145 - val\_accuracy: 0.1037

Epoch 2/30

612/612 [==============================] - 66s 109ms/step - loss: 1.2860 - accuracy: 0.5891 - val\_loss: 0.8165 - val\_accuracy: 0.7301

Epoch 3/30

612/612 [==============================] - 66s 108ms/step - loss: 0.8851 - accuracy: 0.7151 - val\_loss: 0.4981 - val\_accuracy: 0.8371

Epoch 4/30

612/612 [==============================] - 66s 109ms/step - loss: 0.6771 - accuracy: 0.7800 - val\_loss: 0.4374 - val\_accuracy: 0.8591

Epoch 5/30

612/612 [==============================] - 67s 109ms/step - loss: 0.5374 - accuracy: 0.8273 - val\_loss: 0.2828 - val\_accuracy: 0.9089

Epoch 6/30

612/612 [==============================] - 69s 113ms/step - loss: 0.4552 - accuracy: 0.8528 - val\_loss: 0.3179 - val\_accuracy: 0.9010

Epoch 7/30

612/612 [==============================] - 66s 108ms/step - loss: 0.3945 - accuracy: 0.8729 - val\_loss: 0.4852 - val\_accuracy: 0.8471

Epoch 8/30

612/612 [==============================] - 66s 108ms/step - loss: 0.3463 - accuracy: 0.8901 - val\_loss: 0.4838 - val\_accuracy: 0.8553

Epoch 9/30

612/612 [==============================] - 66s 108ms/step - loss: 0.3042 - accuracy: 0.9026 - val\_loss: 0.2943 - val\_accuracy: 0.9101

Epoch 10/30

612/612 [==============================] - 66s 108ms/step - loss: 0.2902 - accuracy: 0.9072 - val\_loss: 0.3902 - val\_accuracy: 0.8835

Epoch 11/30

612/612 [==============================] - 66s 108ms/step - loss: 0.2614 - accuracy: 0.9161 - val\_loss: 0.3241 - val\_accuracy: 0.9002

Epoch 12/30

612/612 [==============================] - 73s 120ms/step - loss: 0.2423 - accuracy: 0.9239 - val\_loss: 0.2501 - val\_accuracy: 0.9231

Epoch 13/30

612/612 [==============================] - 66s 109ms/step - loss: 0.2201 - accuracy: 0.9311 - val\_loss: 0.2497 - val\_accuracy: 0.9324

Epoch 14/30

612/612 [==============================] - 68s 110ms/step - loss: 0.2110 - accuracy: 0.9354 - val\_loss: 0.2422 - val\_accuracy: 0.9288

Epoch 15/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1937 - accuracy: 0.9391 - val\_loss: 0.2083 - val\_accuracy: 0.9377

Epoch 16/30

612/612 [==============================] - 67s 110ms/step - loss: 0.1913 - accuracy: 0.9407 - val\_loss: 1.3573 - val\_accuracy: 0.7012

Epoch 17/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1807 - accuracy: 0.9434 - val\_loss: 0.2318 - val\_accuracy: 0.9296

Epoch 18/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1748 - accuracy: 0.9460 - val\_loss: 0.1635 - val\_accuracy: 0.9525

Epoch 19/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1608 - accuracy: 0.9495 - val\_loss: 0.3264 - val\_accuracy: 0.9099

Epoch 20/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1582 - accuracy: 0.9508 - val\_loss: 0.4142 - val\_accuracy: 0.8927

Epoch 21/30

612/612 [==============================] - 66s 109ms/step - loss: 0.1525 - accuracy: 0.9531 - val\_loss: 0.4402 - val\_accuracy: 0.8745

Epoch 22/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1522 - accuracy: 0.9543 - val\_loss: 0.6831 - val\_accuracy: 0.8247

Epoch 23/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1412 - accuracy: 0.9565 - val\_loss: 6.6964 - val\_accuracy: 0.2070

Epoch 24/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1362 - accuracy: 0.9577 - val\_loss: 1.4984 - val\_accuracy: 0.6606

Epoch 25/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1260 - accuracy: 0.9603 - val\_loss: 0.4197 - val\_accuracy: 0.8886

Epoch 26/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1282 - accuracy: 0.9602 - val\_loss: 2.7532 - val\_accuracy: 0.5317

Epoch 27/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1252 - accuracy: 0.9615 - val\_loss: 0.1803 - val\_accuracy: 0.9480

Epoch 28/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1279 - accuracy: 0.9605 - val\_loss: 0.4663 - val\_accuracy: 0.8773

Epoch 29/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1181 - accuracy: 0.9630 - val\_loss: 0.5160 - val\_accuracy: 0.8751

Epoch 30/30

612/612 [==============================] - 66s 108ms/step - loss: 0.1172 - accuracy: 0.9633 - val\_loss: 0.1836 - val\_accuracy: 0.9488

[INFO] evaluating network...

precision recall f1-score support

Speed limit (20km/h) 0.87 1.00 0.93 60

Speed limit (30km/h) 0.90 0.99 0.94 720

Speed limit (50km/h) 0.96 0.97 0.96 750

Speed limit (60km/h) 0.95 0.93 0.94 450

Speed limit (70km/h) 1.00 0.95 0.97 660

Speed limit (80km/h) 0.93 0.90 0.91 630

End of speed limit (80km/h) 0.90 0.97 0.94 150

Speed limit (100km/h) 0.98 0.97 0.97 450

Speed limit (120km/h) 0.97 0.96 0.96 450

No passing 0.99 0.97 0.98 480

No passing for vehicles over 3.5 metric tons 0.99 0.97 0.98 660

Right-of-way at the next intersection 0.88 0.96 0.92 420

Priority road 0.99 0.99 0.99 690

Yield 0.99 1.00 1.00 720

Stop 0.96 1.00 0.98 270

No vehicles 0.94 0.98 0.96 210

Vehicles over 3.5 metric tons prohibited 0.97 1.00 0.98 150

No entry 1.00 0.96 0.98 360

General caution 1.00 0.74 0.85 390

Dangerous curve to the left 0.61 0.58 0.60 60

Dangerous curve to the right 0.96 0.88 0.92 90

Double curve 0.83 0.64 0.73 90

Bumpy road 0.92 0.83 0.87 120

Slippery road 0.67 0.96 0.79 150

Road narrows on the right 0.85 0.93 0.89 90

Road work 0.92 0.99 0.95 480

Traffic signals 0.93 0.97 0.95 180

Pedestrians 0.67 0.47 0.55 60

Children crossing 0.78 0.99 0.87 150

Bicycles crossing 0.87 0.90 0.89 90

Beware of ice/snow 0.83 0.65 0.73 150

Wild animals crossing 0.99 0.97 0.98 270

End of all speed and passing limits 0.98 0.92 0.95 60

Turn right ahead 0.98 1.00 0.99 210

Turn left ahead 1.00 1.00 1.00 120

Ahead only 0.99 0.99 0.99 390

Go straight or right 0.94 1.00 0.97 120

Go straight or left 1.00 0.95 0.97 60

Keep right 1.00 0.98 0.99 690

Keep left 1.00 0.99 0.99 90

Roundabout mandatory 0.91 0.98 0.94 90

End of no passing 0.91 0.80 0.85 60

End of no passing by vehicles over 3.5 metric tons 0.95 0.87 0.91 90

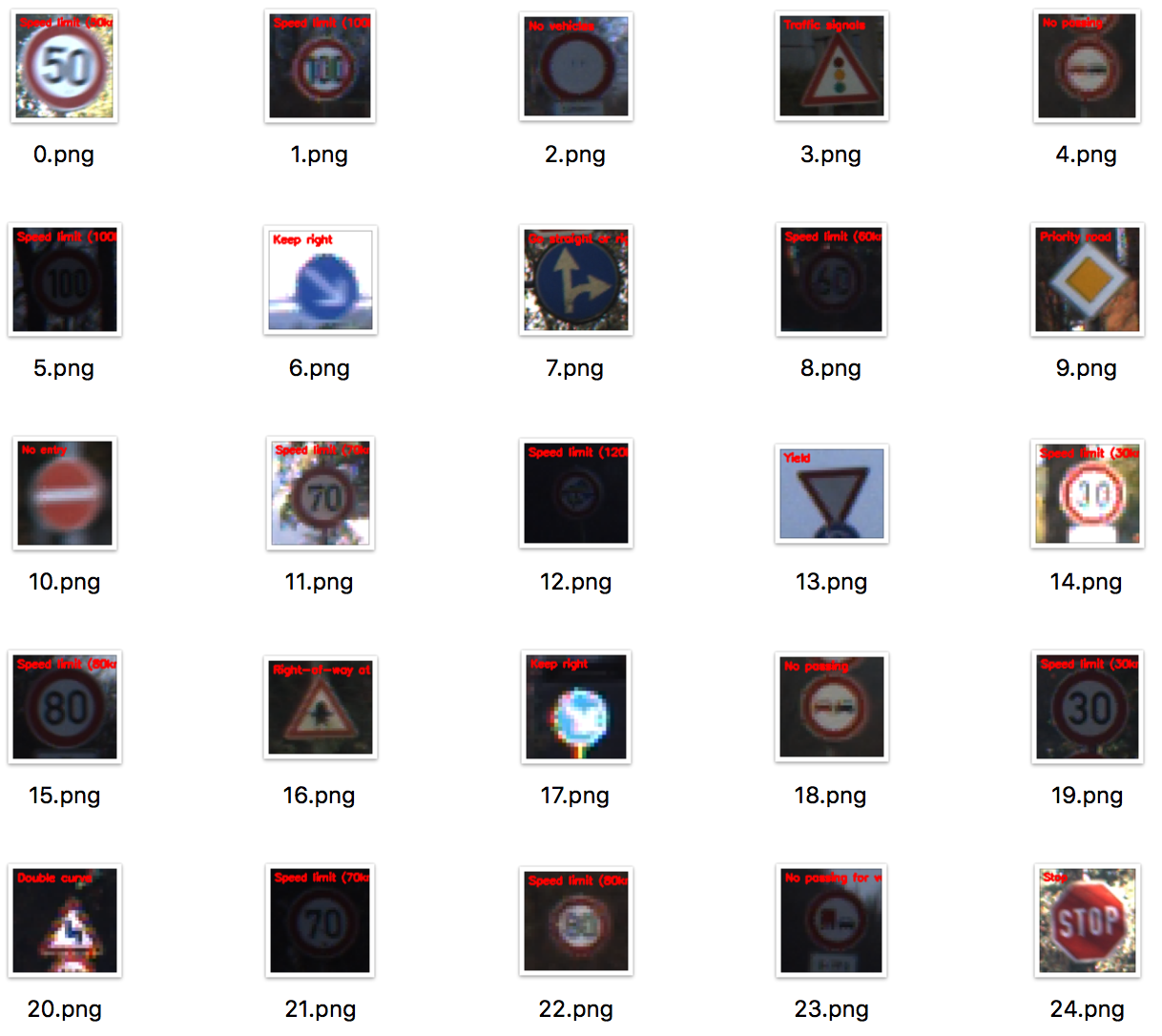
accuracy 0.95 12630

macro avg 0.92 0.92 0.92 12630

weighted avg 0.95 0.95 0.95 12630

It was observed that on the third iteration, the model was trained well and achieved better accuracy of around **95%.**

# Prediction



* Firstly, load the model from disk.
* Secondly, load the sample images from disk.
* Preprocess the sample images in the same manner as we did for training.
* Followed by, passing our images through our traffic sign classifier.
* Obtain our final output predictions  
    
  The model was trained with and without CLAHE (Contrast Limited Adaptive Histogram Equalization) method and the results varied for different iterations with the same conditions.The traffic sign in focus is the stop sign.

|  | **Output** | **Plot w/ CLAHE** | **Plot w/ CLAHE** |
| --- | --- | --- | --- |
| 1. | Chart |  |  |
| 2. | Chart |  |  |
| 3. | Chart |  |  |
| 4. | Chart |  |  |
| 5. | Chart |  |  |

Finally, a standard error was calculated for both the conditions (with & without CLAHE) and it was observed that using the method really helped the model learn better and perform well during prediction.  
  


# References

G. Yadav, S. Maheshwari and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, New Delhi, 2014, pp. 2392-2397, doi: 10.1109/ICACCI.2014.6968381.

A. de la Escalera, L. E. Moreno, M. A. Salichs and J. M. Armingol, "Road traffic sign detection and classification," in *IEEE Transactions on Industrial Electronics*, vol. 44, no. 6, pp. 848-859, Dec. 1997, doi: 10.1109/41.649946.

S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing and C. Igel, "Detection of traffic signs in real-world images: The German traffic sign detection benchmark," The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, 2013, pp. 1-8, doi: 10.1109/IJCNN.2013.6706807.

A. Mogelmose, M. M. Trivedi and T. B. Moeslund, "Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey," in IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, pp. 1484-1497, Dec. 2012, doi: 10.1109/TITS.2012.2209421.

A. Møgelmose, D. Liu and M. M. Trivedi, "Detection of U.S. Traffic Signs," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 6, pp. 3116-3125, Dec. 2015, doi: 10.1109/TITS.2015.2433019.