Writeup

November 30, 2017

1 Traffic Sign Recognition

1.1 Writeup

The aim of this project is to develop an algorithm using deep learning convolution neural network to classify traffic signs. The project invols the following steps.

The goals / steps of this project are the following: * Load the data set (see below for links to the project data set) * Explore, summarize and visualize the data set * Design, train and test a model architecture * Use the model to make predictions on new images * Analyze the softmax probabilities of the new images * Summarize the results with a written report

1.1.1 Rubric Points

1.2 Here the project reqirements rubric points are considered individually and described in detail how the implementation is done.

1.2.1 Project Files

The project files includes the python project code written in jupyter notebook, input dataset stored in .picle format, traffic sign pictures downloaded from internet. * Here is a link to the project code

1.2.2 Data Set Summary & Exploration

1. Dataset Summary The input data contains three sets of data called the training, validation and test. Only the training dataset is used for training the neural network. The proformance of trained network is validated using the validation dataset. Based on the performance of the network measured from the training and validation set the parameters of the network are tuned to acheive maximum possible performance. In the following sections it is explained in detail how the parameters of the network are tuned. Finally after tuning of parameters the network is tested with test dataset.

The summary of the dataset are calculated using python native features and then the dataset is visualized using matplotlib library. The basic summary of the dataset follows:

- The size of training set is 34799
- The size of the validation set is 4410
- The size of test set is 12630

- The shape of a traffic sign image is (32, 32, 3)
- The number of unique classes/labels in the data set is 43

2. Exploratory visualization of the dataset. The images in the dataset are grouped according its class and the no of images in each class are plotted as a bar chart in the following picture.

The firt thing that can be noted is that the number of images in the training dataset are greater than validation and test dataset. Large amount of training data is useful because the network learns better when it is trained with large dataset. It should also be noted that the classes such as ′speed limit 50 km/hr′ have large amount of data compared to class such as "Dangerous Curve to the Left". The network might have difficulty in classifying the traffic signs which have less data. In such cases it is better to augument the data for such classes. But in this project this is not done as the network acheived satisfactory performance.

The datasets have 43 different traffic signs one image in each of the traffic signs are shown in the picture below.

1.2.3 Design and Test a Model Architecture

1. Preprocessing A raw image in RGB format is represented in form of an array of size m x n x 3 where is m,n are number of pixes along width and height of the image and 3 represents three channels red,blue and green. Each value in this array ranges from 0 to 255 for an 8 bit image. The gradient desent algorithm works better if the input dataset is normalized i.e. the image has equal variances in both horzontal and vertical directions and zero mean.

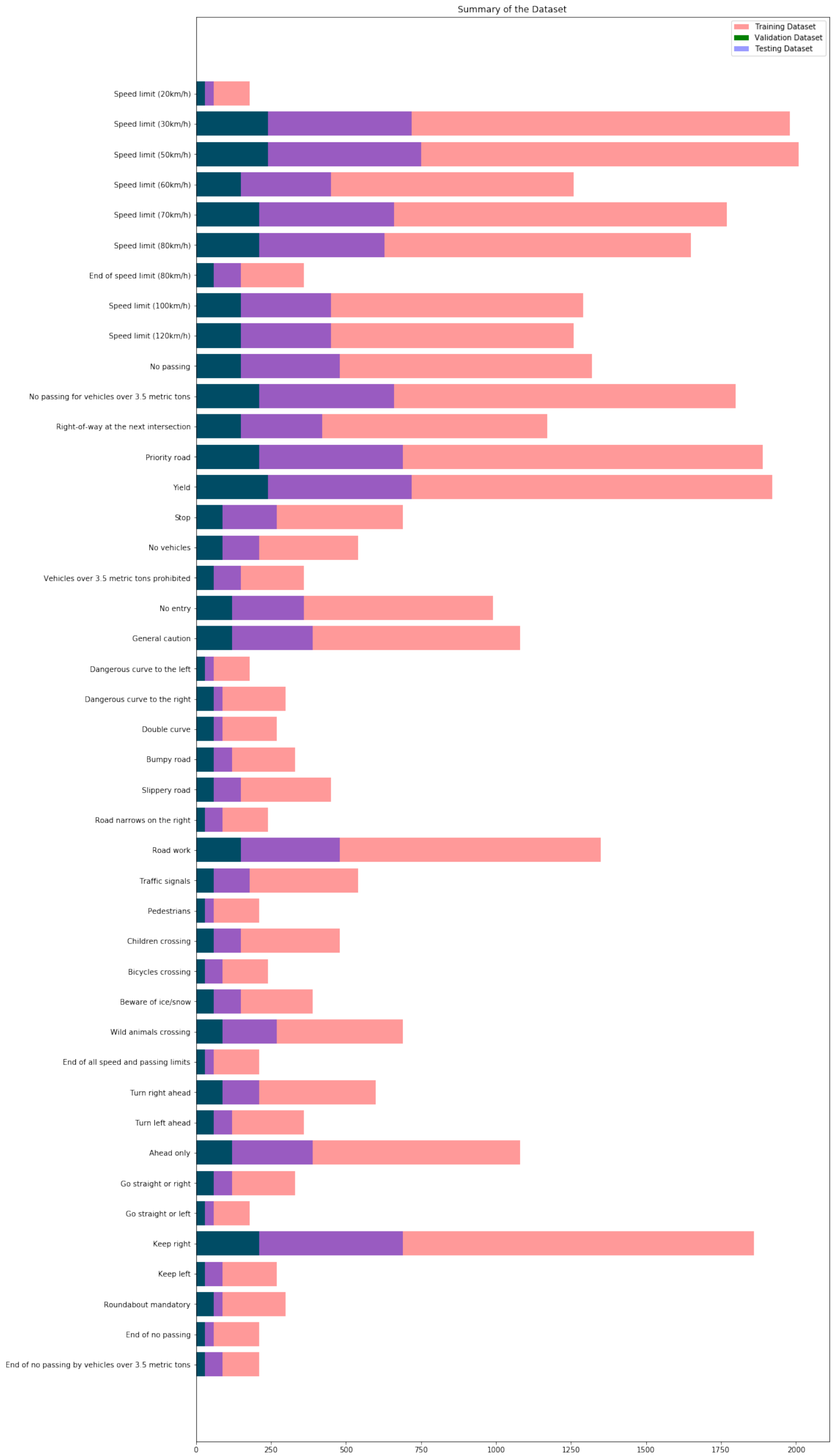
In this project a simpler approach called min-max scaling is used. This approach scales the input values from 0 to 255 in a 8 bit image to a values in a given range. The following formula is used to do min-max scaling.

Min-Max Scaling: $X' = a + \{\frac{(X - X_{\text{min}})(b-a)}{X_{\text{max}} - X_{\text{min}}}\}$

In the above forumla a and b are choosen output range chosen in the project as 0 and 1. X_min and X_max are the minimum and maximum values of the input data. This formula it simpler because the mean and variance are avoided and it acheives good performance. The picture below shows the images before and after normalization.

This project feeds RGB coloured image to the network as the convolution neural networks are able to handle 3 channel images and also the colour information can be useful in classifying the traffic signs as some signs are in different colours.

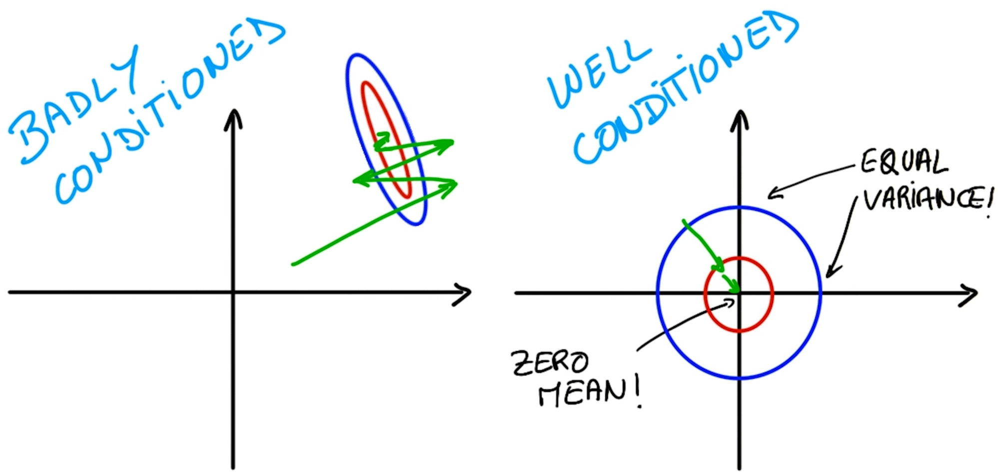
2. Model Architecture The LeNet architecture is used in this project. The LeNet architecture is a multi layed neural network architecture which is popularly used to classify handwritten character recognition. The LeNet architecture contains two sets of convolution layer and maximul pooling layer alternatevely followed by three fully connected layer.The convolution and fully connected layer uses rectified logic units(Relu) for activation.Finally the output of the final is feed to softmax function to calculate the probablities of the classes. The following image and the table shows the LeNet architecture used in this project.


Summary of the Dataset
Training Dataset
Validation Dataset
Testing Dataset
Speed limit (20km/h)
Speed limit (30km/h)
Speed limit (50km/h)
Speed limit (60km/h)
Speed limit (70km/h)
Speed limit (80km/h)
End of speed limit (80km/h)
Speed limit (100km/h)
Speed limit (120km/h)
No passing
No passing for vehicles over 3.5 metric tons
Right-of-way at the next intersection
Priority road
Yield
Stop
No vehicles
Vehicles over 3.5 metric tons prohibited
No entry
General caution
Dangerous curve to the left
Dangerous curve to the right
Double curve
Bumpy road
Slippery road
Road narrows on the right
Road work
Traffic signals
Pedestrians
Children crossing
Bicycles crossing
Beware of ice/snow
Wild animals crossing
End of all speed and passing limits
Turn right ahead
Turn left ahead
Ahead only
Go straight or right
Go straight or left
Keep right
Keep left
Roundabout mandatory
End of no passing
End of no passing by vehicles over 3.5 metric tons
0
250
500
750
1000
1250
1500
1750
2000
No of Image in each class

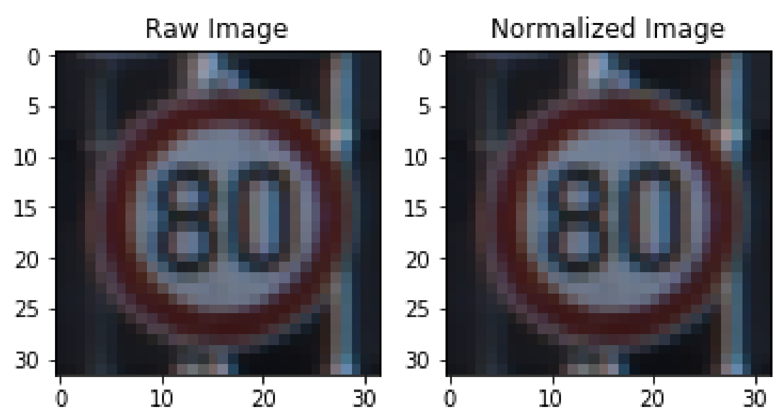
alt text


Stop
Dangerous curve to the right
Bicycles crossing
Roundabout mandatory
Go straight or right
Speed limit (50km/h)
Ahead only
Speed limit (60km/h)
Slippery road
Dangerous curve to the left
Yield
Traffic signals
No passing
Speed limit (70km/h)
Speed limit (80km/h)
No entry
Speed limit (30km/h)
Keep right
End of all speed and passing limits
Children crossing
Right-of-way at the next intersection
Road narrows on the right
General caution
End of no passing
Wild animals crossing
End of speed limit (80km/h)
Speed limit (100km/h)
No vehicles
Keep left
Vehicles over 3.5 metric tons prohibited
Pedestrians
Go straight or left
Turn right ahead
Priority road
Turn left ahead
End of no passing by vehicles over 3.5 metric tons
Beware of ice/snow
Speed limit (120km/h)
Road work
Speed limit (20km/h)
No passing for vehicles over 3.5 metric tons
Double curve
Bumpy road

alt text



alt text



alt text

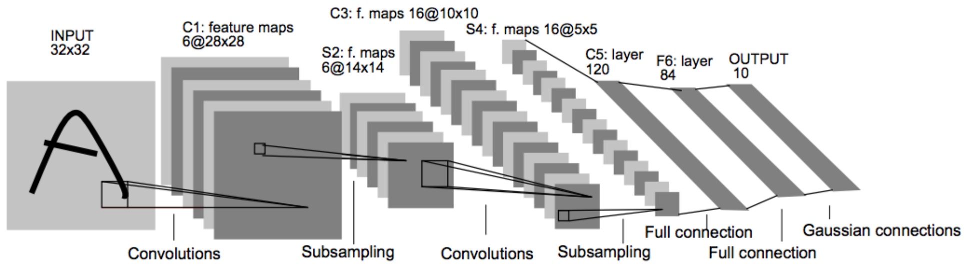
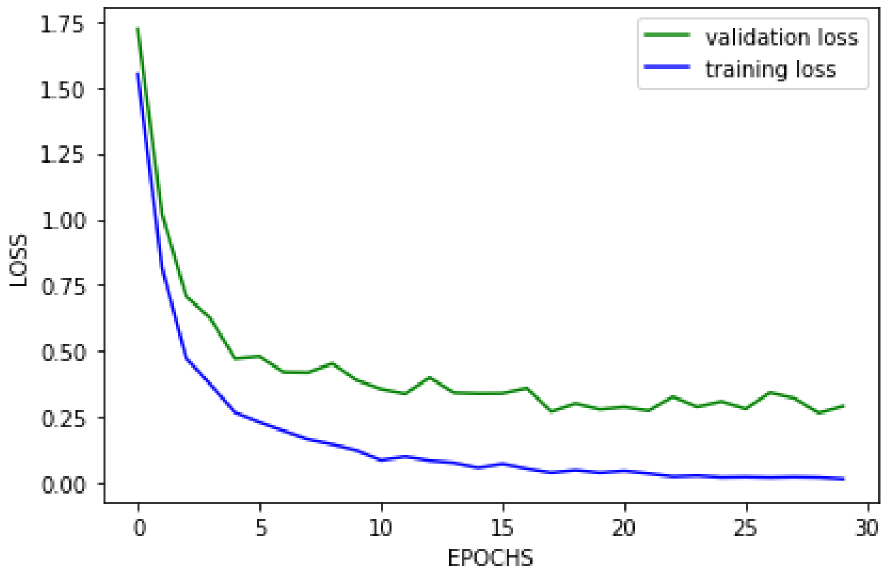


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

source: Yann Lecunn

Layer	Description
Input	32x32x3 RGB image
1st Convolution	5x5x3x6 filter, 1x1 stride, valid padding, outputs 28x28x6
RELU	Activation
Max pooling	2x2 filter, 2x2 stride, outputs 14x14x6
2nd Convolution	5x5x6x16 filter, 1x1 stride, valid padding, outputs 10x10x16
RELU	Activation
Max pooling	2x2 filter, 2x2 stride, outputs 5x5x16
Flatten	Reshapes the input array into a vector/row matrix,output 5x5x16 = 400
Fully connected	Flattened array is connected to 120 hidden nodes, weights 400x120,bias 120
RELU	Activation
Fully connected	Hidden layer 84 nodes, weights 120x84,bias 84
RELU	Activation
Fully connected	Output layer nodes = no. of classes(43), weights 84x43,bias 43
Softmax	Activation to calulate probablities of the classes

3. Model Training and Solution Approach The model is trained using gradient decent approach to minimize the cross entropy loss between the softmax predition from the network and the labels from the input dataset.For the optimizer Adam optimizer is used,which is an extension of the stocastic gradient desent algorithm. The adam algorithm is computationally more efficient than the stocastic gradient desent algorithm. The adam optimizer does not keep the learning rate constant instead it computes the adaptive learning rates of each of the parameters based on the first and second order moments of the gradients. This makes the optimizer more efficient. The details of the optimizer is out of scope here. The model is trained by feeding the input datasets in batches, calculating the cross entropy loss and adapting the weights of the network based on the learning rate. The whole dataset is feed through the network several times called EPOCHS in order to train the network better. This leads to several hyper parameters learning rate, batch size and EPOCHS which have to be tuned in order to acheive high accuracy. During the traiging of the model validation dataset is feed through the the to evaluate the accuracy of the network's predictions.The learning rate affects the way the network learns, higher learning rate means that the weights are changed in bigger steps this might make the network to overshoot from the taget and the loss may not be



alt text

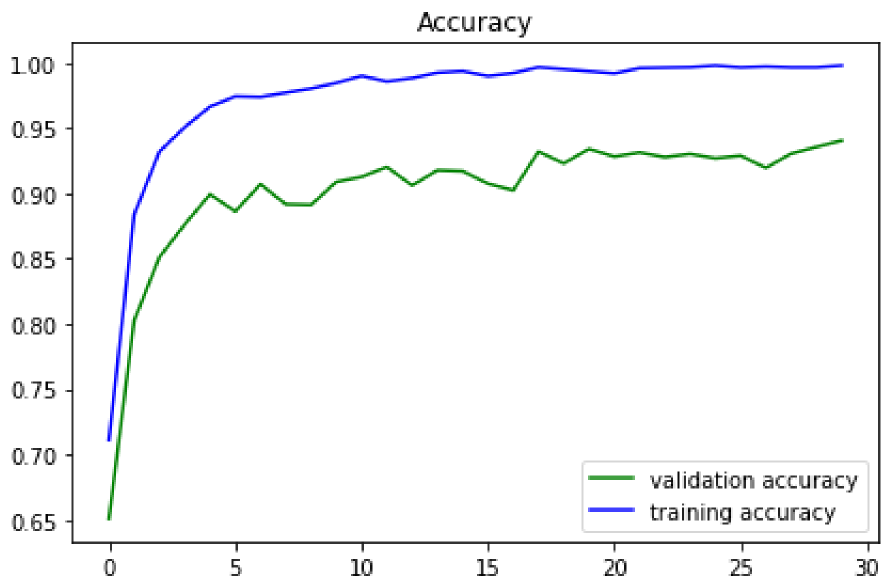
minimized. Lower learning rate reduces the size of steps the weights are changed and it makes the optimization to converge better. The batch size also plays similar role as the learning rate as it is propotional to the number of training steps. If the batch size is higher the number of training steps is lower and the network may not be trained to get best possible result. Lowering the batch size increases the number of iterations in the learning process and thus trains the network learn better. Like the batch size EPOCHS is also propotional network's number of learning steps. Higher the EPOCHS the networks learns better but this can cause the network to overfit the training dataset. To make to network classify the data with high accurary the parameters learning rate, batch size and EPOCHS were chosen carfully by repeated iterations.

Since the training data is used several times in order to train the network it lead to a problem of overfitting. The network was able to classify the training data with high accuracy but the it could not classify the validation data with the same accuracy. To avoid this problem dropout regularization technique is introduced to the trianing process, which makes the network more robust. The droput function was applied on the networks prediction before being fed to the cross entropy function. The dorpout function randomy drops some of the data in the networks prediction and doubles the remaining values in order to maintain the total probablity of the prediction. This forces the network to become more robust. After application of dropout the network was able to classify the validation dataset with higher accuracy and closer to the accuracy of training dataset. The dropout operation adds one more parameter to the network called the keep probabplity parameter which ranges from 0 to 1. The keep probablity represents the probabablity of an element to be kept in the output of the dropout function.

Finally the following parameter values: * Learning Rate = 0.0005 * Batch Size = 100 * EPOCHS = 40

The final model results were: * training set accuracy of 0.998 * validation set accuracy of 0.939 * test set accuracy of 0.915

The loss and accurary during the learning process:



alt text

alt text

1.2.4 Test a Model on New Images

Some images of the german traffic signs were downloaded from the internet and these data were feed to the network to be classified. The network was able to classify 9 out of the 12 signs correctly which is an accuracy of 0.75.

Traffic signs downloaded from the internet:

Prediction made by the network:

The following table shows the top 5 probablities for each of the traffic signs:

Image No	Label	Highest Probability	2nd Highest	3rd Highest	4th Highest	5th Highest
1	Ahead only	Ahead only	Roundabout Mandatory	Turn Right Ahead	Go Straight or Right	Turn Left Ahead

Image No	Label	Highest Probability	2nd Highest	3rd Highest	4th Highest	5th Highest
		1	0.000	0.000	0.000	0.000
2	No entry	No entry	Yield	Bicycles Crossing	No Passing for Vehicles over 3.5 Metric Tons	Slippery Road
		0.589	0.250	0.150	0.008	0.001
3	Priority road	Priority road	End of all speed and passing limits	Traffic Signals	End of no passing	No entry
		1	0.000	0.000	0.000	0.000
4	Right-of-way at the next intersection	Right-of-way at the next intersection	Bicycles crossing	Speed limit(80km/h)	Beware of ice/snow	Speed limit (30km/h)
		1.000	0.000	0.000	0.000	0.000
5	Roundabout mandatory	Roundabout mandatory	General caution	Right-of-way at the next intersection	Children crossing	Keep right
		0.999	0.001	0.000	0.000	0.000
6	Slippery road	Speed limit(20km/h)	No entry	Dangerous curve to the right	End of speed limit(80km/h)	Speed limit(30km/h)
		0.958	0.034	0.004	0.002	0.000

Image No	Label	Highest Probability	2nd Highest	3rd Highest	4th Highest	5th Highest
7	Speed limit(30km/h)	Speed limit(30km/h)	Speed limit(50km/h)	Go straight or right	Keep right	End of all speed and passing limits
		0.937	0.062	0.000	0.000	0.000
8	Speed limit(60km/h)	Speed limit(50km/h)	Speed limit(60km/h)	Speed limit(80km/h)	Speed limit(30km/h)	Go straight or right
		0.718	0.282	0.000	0.000	0.000
9	Speed limit(70km/h)	Speed limit(20km/h)	Bicycles crossing	Ahead only	Speed limit(30km/h)	Go straight or right
		0.589	0.373	0.033	0.005	0.000
10	Stop	Stop	Speed limit(30km/h)	Speed limit(80km/h)	Priority road	Speed limit(50km/h)
		0.985	0.014	0.001	0.000	0.000
11	Turn right ahead	Turn right ahead	Right-of-the-way at the next intersection	Beware of ice/snow	Road narrows on the right	Double curve
		0.623	0.377	0.000	0.000	0.000
12	Yield	Yield	Speed limit(30km/h)	No vehicles	No passing	Speed limit(60km/h)
		1.000	0.000	0.000	0.000	0.000

The network was not able to classify the traffic signs Slippery Road, Speed Limit 60km/h and Speed Limit 70km/h. Only around 500 images of slippery road traffic sign was used during the training, may the training dataset with more images could be used to train the network or data augumentaion could be used to improve the networks ability to classify this sign. If we look at the top probablities for the speed limit signs the network is not able to classify the digits in the traffic signs properly. For traffic signs speed limit 60 km/h it predicts as 70km/h and for speed limit 70km/h it predicts as 20km/h speed limit. This suggest that the network can be trained more to improve the classification of similar images. To make the algorithm work better the traffic sign recognition can be done in two steps, in the first step the network should be trained to classify the type of road sign such as speed limits, cautions, directions etc.Then in the second step the network

alt text

should classify the specific sign such as speed limit 50km/h, 70km/h etc.