Traffic Sign Classifier

Problem Statement:

Coming up with a model that can predict validation data set with 93% or more accuracy. Training a model of German traffic signs and training, validation, and test data are provided. The model is trained on set of 7 traffic signs and show the top 5 predictions.

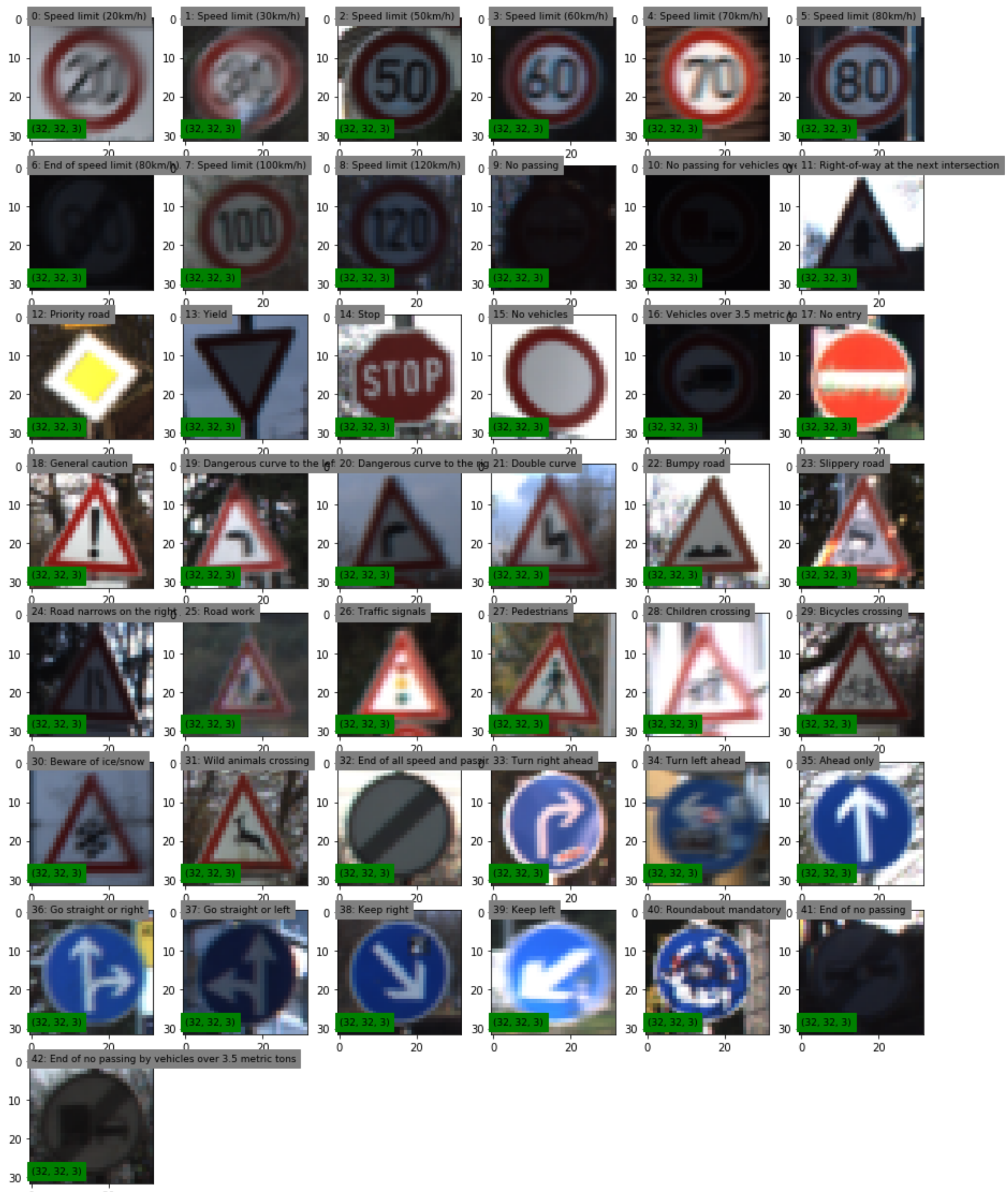


Figure1: 43 Unique German Traffic Signs

Data Exploration:

There are 43 unique signs and randomly selected signs shown above in Figure1.

Design and Test a Model Architecture:

Reaching expected validation accuracy (0.93 or higher):

What was the first architecture was tried and why was it chosen?

Initially started with the original LeNet model and the model specifications are shown on Table1. The model was tested with 20 epochs but validation accuracy lower than 0.90 then I started increasing depth and adding more fully connected layers. The initial model used for base lining.

Table1: LeNet Model Details

Layer	Filter	Strides	Padding	Kernel	Shape
Input					32x32x3
Convolution	5,5,1,6	1,1,1,1	Valid		28x28x3
Pooling		1,2,2,1		1,2,2,1	14x14x6
Convolution	5,5,24,16	1,1,1,1	Valid		10x10x16
Pooling		1,2,2,1		1,2,2,1	5x5x16
Fully Connected					400
Fully Connected					168
Fully Connected					84
Fully Connected					43

What were some problems with the initial architecture?

The main reason for the initial architecture was not providing good validation accuracy.

How was the architecture adjusted and why was it adjusted?

Alteration to the provided original LeNet model was to increase depth, multiply by 4, and adding additional fully connected layer. Table1, LeNet Enhanced Model specifications given as follow

Table2: Enhanced LeNet Model

Layer	Filter	Strides	Padding	Kernel	Shape
Input					32x32x3
Convolution	5,5,1,24	1,1,1,1	Valid		28x28x3
Pooling		1,2,2,1		1,2,2,1	14x14x24
Convolution	5,5,24,64	1,1,1,1	Valid		10x10x64
Pooling		1,2,2,1		1,2,2,1	5x5x64
Fully Connected					1600
Fully Connected					480
Fully Connected					168
Fully Connected					84
Fully Connected					43

Which parameters were tuned? How were they adjusted and why?

For the model architecture only depth and additional fully connected layer added because validation accuracy produced result around 0.93 about 20 epochs.

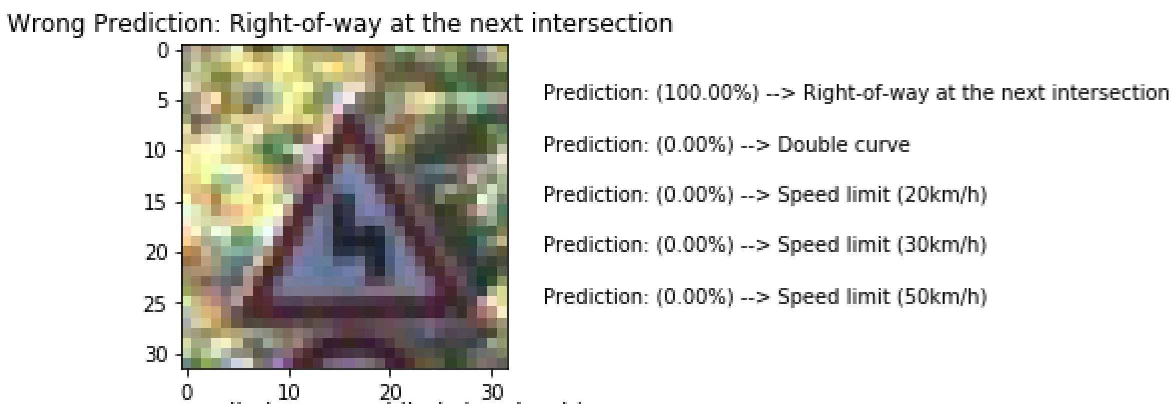
What are some of the important design choices and why were they chosen?

Increasing the depth by 4 made the training model much slower with tradeoff of better validation accuracy.

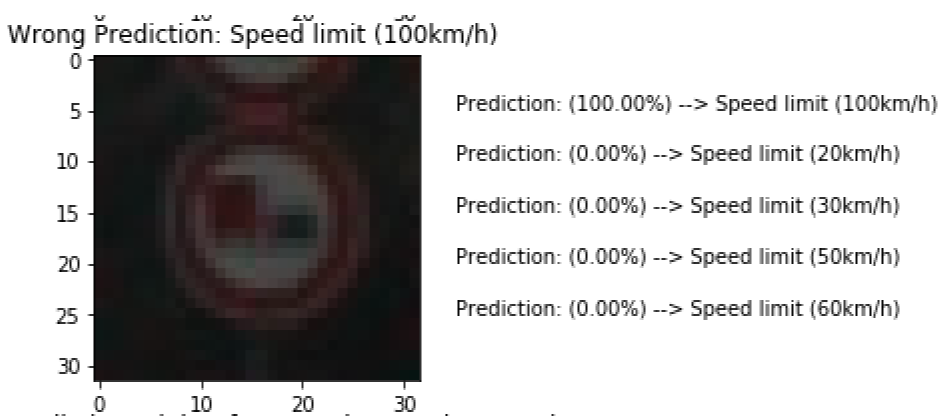
Test a Model on New Images:

Images selected on the web were different sizes which all of them bigger are bigger sizes than 32x32. Some of the images are dark and some are bright that is my concern that might lead to a wrong prediction. Also, with scaling down to 32x32 some useful image information is not retained.

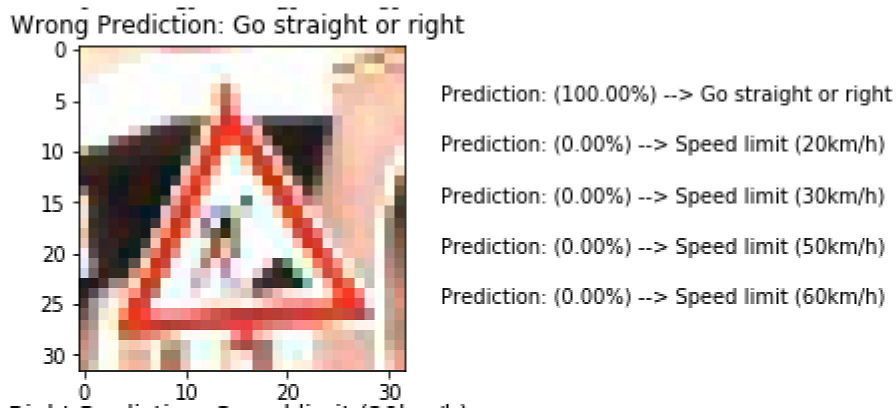
For instance, given image below predicted wrong that might be due to the brightness of the image with 100% confident level although the second probability choice is correct but 0% confident level.



Another example of image being dark also was classified wrong and none of the top 5 probabilities made a correct prediction.



Lastly, due to scaling image was lost crucial information and image shown below.



Test set accuracy is 93.9% and accuracy is only 43% on the new images that indicates that the model is overfitting.

Image Modifications:

1- Color Space:

Number of color spaces has been trained with but Gray and YUV are the best accuracy among all the color spaces. YUV space gave a slightly better accuracy.

2- Image Normalization

Normalization has a huge impact on the accuracy. For instance, validation data training accuracy jumped from 85% to 91% after normalizing the image data.

3- Image Rotation

To make the model more robust and avoid over fitting rotation images up to +/- 15 degree but it had a negative impact on the validation accuracy so rotation or any other augmentation was avoided during training.

Generic Improvements:

1- Learning Rate:

Dividing learning rate by 10 it made the model more stable but it took longer to train over 95% validation accuracy.

2- Epochs:

With small numbers such as 20 and 50 didn’t satisfy for a stable model because it was indicated in learning curves that there was still room to improve. When epochs set to 350 it was seen that training accuracy reached to the limit and validation accuracy was stable enough to stop at 350.