#**Traffic Sign Recognition**

##Writeup Template

###You can use this file as a template for your writeup if you want to submit it as a markdown file, but feel free to use some other method and submit a pdf if you prefer.

Build a Traffic Sign Recognition Project

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

Rubric Points

###Here I will consider the rubric points individually and describe how I addressed each point in my implementation.

###Writeup / README

####1. Provide a Writeup / README that includes all the rubric points and how you addressed each one. You can submit your writeup as markdown or pdf. You can use this template as a guide for writing the report. The submission includes the project code.

You're reading it! and here is a link to my project code

###Data Set Summary & Exploration

####1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

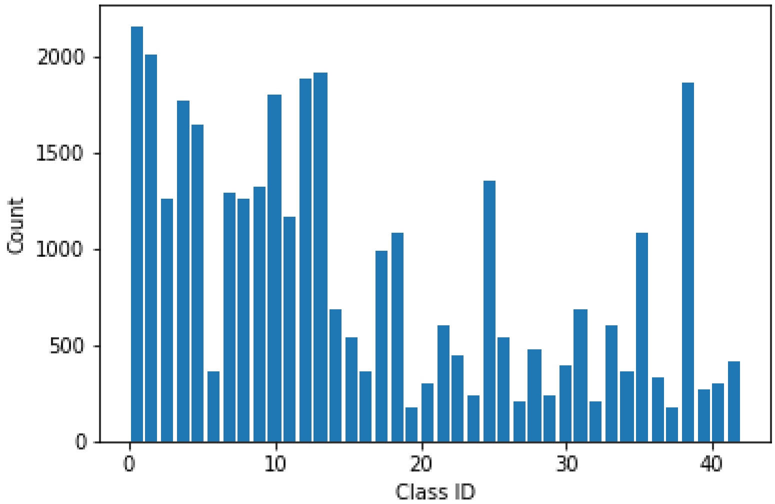
I used the pandas library to calculate summary statistics of the traffisigns data set:

- 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 (34799, 32, 32, 3)
- The number of unique classes/labels in the data set is 43

####2. Include an exploratory visualization of the dataset.

Here is an exploratory visualization of the data set. It is a bar chart showing how the data distributes.

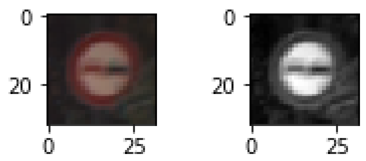


###Design and Test a Model Architecture

####1. Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques? Consider including images showing the output of each preprocessing technique. Pre-processing refers to techniques such as converting to grayscale, normalization, etc. (OPTIONAL: As described in the "Stand Out Suggestions" part of the rubric, if you generated additional data for training, describe why you decided to generate additional data, how you generated the data, and provide example images of the additional data. Then describe the characteristics of the augmented training set like number of images in the set, number of images for each class, etc.)

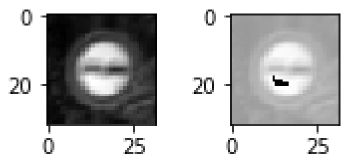
As a first step, I decided to convert the images to grayscale because I7d like to make the image more simple than the color one.

Here is an example of a traffic sign image before and after grayscaling.



As a last step, I normalized the image data because I can deal with the data at ease.

Here is an example of an original image and an augmented image:



The difference between the original data set and the augmented data set is the following

1. The training data set

The shape before the augmentation is (34799, 32, 32, 3).
The shape after the augmentation is (34799, 32, 32, 1).

2. The validation data set

The shape before the augmentation is (4410, 32, 32, 3).

The shape after the augmentation is (4410, 32, 32, 1).

3. The test data set

The shape before the augmentation is (12630, 32, 32, 3).

The shape after the augmentation is (12630, 32, 32, 1).

####2. Describe what your final model architecture looks like including model type, layers, layer sizes, connectivity, etc.) Consider including a diagram and/or table describing the final model.

My final model consisted of the following layers:

Layer	**Description**
Input	32x32x1 grayscale image
Convolution 3x3	1x1 stride, valid padding, outputs 28x28x6
RELU	
Max pooling	2x2 stride, outputs 14x14x6
Convolution 3x3	1x1 stride, valid padding, outputs 10x10x16
RELU	

Layer	Description
Max pooling	2x2stride, outputs 5x5x16
Flatten	Output 400
Fully connected	Output 120
RELU	
Fully connected	Output 84
RELU	
Fully connected	Output 43

####3. Describe how you trained your model. The discussion can include the type of optimizer, the batch size, number of epochs and any hyperparameters such as learning rate.

To train the model, I used an AdamOptimizer.

I set the batch size as 128, number of epochs as 30, and learning rate as 0.001.

####4. Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93. Include in the discussion the results on the training, validation and test sets and where in the code these were calculated. Your approach may have been an iterative process, in which case, outline the steps you took to get to the final solution and why you chose those steps. Perhaps your solution involved an already well known implementation or architecture. In this case, discuss why you think the architecture is suitable for the current problem.

My final model results were:

- training set accuracy of ?
- the highest validation set accuracy of 0.934.
- test set accuracy of 0.912.

If an iterative approach was chosen:

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

  I chose the color image data set as the input at first.

- What were some problems with the initial architecture?

  The accuracy didn't increase so much.

- How was the architecture adjusted and why was it adjusted? Typical adjustments could include choosing a different model architecture, adding or taking away layers (pooling, dropout, convolution, etc), using an activation function or changing the activation function. One common justification for adjusting an architecture would be due to overfitting or underfitting. A high accuracy on the training set but low accuracy on the validation set indicates over fitting; a low accuracy on both sets indicates under fitting.

  I didn’t change the activation function and didn’t use the generalization.

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

  I increased number of epochs from 10 to 30 to train the pipeline more.

- What are some of the important design choices and why were they chosen? For example, why might a convolution layer work well with this problem? How might a dropout layer help with creating a successful model?

  They’re important that the depth of the first input is 1 and I increased number of epochs.

If a well known architecture was chosen:

- What architecture was chosen?
- Why did you believe it would be relevant to the traffic sign application?
- How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?

###Test a Model on New Images

####1. Choose five German traffic signs found on the web and provide them in the report. For each image, discuss what quality or qualities might be difficult to classify.

Here are five German traffic signs that I found on the web:


The first image might be difficult to classify because it gets leaned a bit.

The second image might be difficult to classify because it gets leaned a bit.

The third image might be difficult to classify because it gets offset from the center of the image a bit.

The fourth image might be difficult to classify because it gets leaned a bit.

The fifth image might be difficult to classify because it's small a bit and gets offset from the center of the image.

####2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set. At a minimum, discuss what the predictions were, the accuracy on these new predictions, and compare the accuracy to the accuracy on the test set (OPTIONAL: Discuss the results in more detail as described in the "Stand Out Suggestions" part of the rubric).

Here are the results of the prediction:

Image	Prediction
Bicycles crossing	Slippery road
Bumpy road	Slippery road
Children crossing	Speed limit (50km/h)
Speed limit (60km/h)	Speed limit (50km/h)
Turn left ahead	General caution

The model was able to correctly guess 0 of the 5 traffic signs, which gives an accuracy of 0%. This is less than the accuracy on the test set of 0.912.

####3. Describe how certain the model is when predicting on each of the five new images by looking at the softmax probabilities for each prediction. Provide the top 5 softmax probabilities for each image along with the sign type of each probability. (OPTIONAL: as described in the "Stand Out Suggestions" part of the rubric, visualizations can also be provided such as bar charts)

The code for making predictions on my final model is located in the 35th cell of the Ipython notebook.

For the first image, the model detected this is a slippery road sign (probability of 0.22). However the image isn't slippery road sign. The model contain a stop sign. The top five soft max probabilities were

Probability	Prediction
.22	Slippery road
.19	Bicycles crossing
.09	Speed limit (80km/h)
.08	Dangerous curve to the right
.04	No passing for vehicles over 3.5 metric tons

For the second image, the model detected this is a slippery road sign (probability of 0.22). However the image isn't slippery road sign. The model contain a stop sign. The top five soft max probabilities were

Probability	Prediction
.59	Slippery road
.19	Dangerous curve to the left
.17	Right-of-way at the next intersection
.13	No passing
.10	No passing for vehicles over 3.5 metric tons

For the third image, the model detected this is a slippery road sign (probability of 0.22). However the image isn't slippery road sign. The model contain a stop sign. The top five soft max probabilities were

Probability	Prediction
.15	Speed limit (50km/h)
.11	Speed limit (30km/h)
.06	Wild animals crossing
.06	Keep right
.03	Speed limit (80km/h)

For the forth image, the model detected this is a slippery road sign (probability of 0.22). However the image isn't slippery road sign. The model contain a stop sign. The top five soft max probabilities were

Probability	Prediction
.26	Speed limit (50km/h)
.17	Speed limit (80km/h)
.12	Speed limit (30km/h)
.04	Speed limit (60km/h)
.03	Speed limit (20km/h)

For the fifth image, the model detected this is a slippery road sign (probability of 0.22). However the image isn't slippery road sign. The model contain a stop sign. The top five soft max probabilities were

Probability	Prediction
.21	General caution
.11	Speed limit (30km/h)
.08	Roundabout mandatory
.07	Right-of-way at the next intersection

Probability	Prediction
.04	Road work

(Optional) Visualizing the Neural Network (See Step 4 of the Ipython notebook for more details)

####1. Discuss the visual output of your trained network's feature maps. What characteristics did the neural network use to make classifications?