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 use len() to calculate summary statistics of the tra

- The size of training set is ?

The size of training size is 34799

- The size of the validation set is ?

The size of training size is 4410

- The size of test set is ?

The size of training size is 12630

- The shape of a traffic sign image is ?

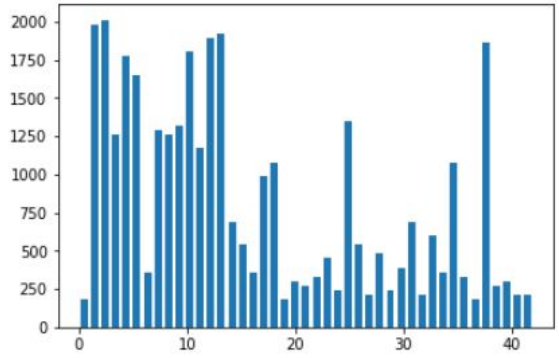
The shape of training size is 32X32X3

- The number of unique classes/labels in the data set is ?

The number of unique classes/labels 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 training data is distributed



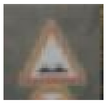
Design and Test a Model Architecture

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 for grayscale, the image would only have 1 channel thus, no extra confusion would be introduced into the learning process.

Also, I normalized the image data because normalizing data would helps in speeding up the training and improving the performance of training.

Here is an example of the traffic sign image before and after being processed.



Before processed

After processed

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
1	Input	32X32X1 Grayscale image
2	Convolution 5x5	1x1 stride, valid padding, outputs 28x8x6
3	Relu	
4	Max pool	2x2 stride, outputs 14x14x6
5	Convolution 5x5	1x1 stride, valid padding, outputs 10x10x16
6	Relu	
7	Max pool	2x2 stride, outputs 5x5x16
8	Convolution 2x2	1x1 stride, valid padding, outputs 1x1x400

9	Relu	
10	Flatten layers from 9	(1x1x400 ->400)
11	Dropout layer	
12	Fully connected layer	400->200
13	Dropout layer	
14	Fully connected layer	200->120
15	Dropout layer	
16	Fully connected layer	120->80
17	Dropout layer	
18	Fully connected layer	80->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 LeNet for the most part that was given, I added an additional convolution layer and also fully connected layer. I also used AdamOptimizer with a learning rate of 0.0009. The number of epochs I use are 30 and the number of batch size was 100. For dropout probability, I used 0.5.

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 0.97
- validation set accuracy of 0.97
- test set accuracy of 0.943

If a well known architecture was chosen:

- What architecture was chosen?

I used the similar LeNet architecture given in the course

- Why did you believe it would be relevant to the traffic sign application?

The LeNet architecture is a very good convolutional architecture. It has series of Convolutional layers and Pooling layers and series of fully connected layers. Because it is a convolutional neural network architecture. It is very efficient for classifying traffic sign with larger amount of data.

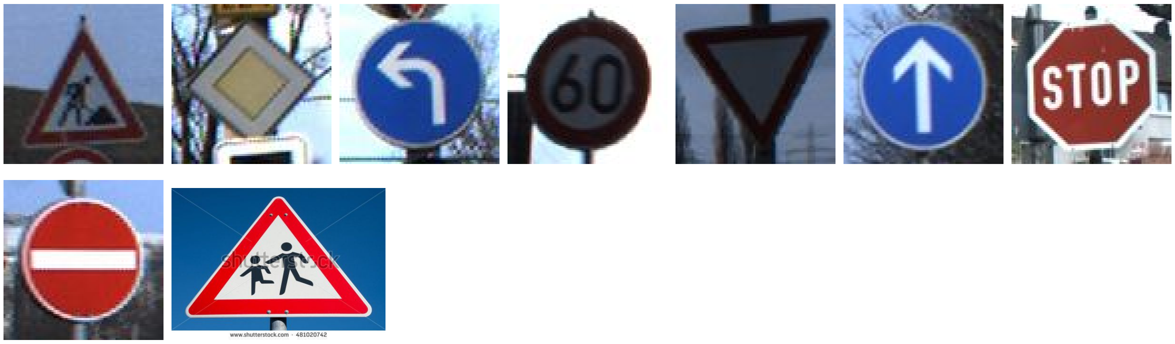
- How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?

From the results, It shows that all training, validation and test accuracy is above 94%, which is showing that the model works 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:



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
Road Work	Road Work
Priority	Priority
Turn Left ahead	Turn Left ahead

Speed limit (60km/h)	Speed limit (60km/h)
Slippery Road	Slippery Road
Yield	Yield
Ahead only	Ahead only
Stop	Stop
Children crossing	Children crossing

The model was able to correctly guess 7 of the 7 traffic signs, which gives an accuracy of 100%.

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)

For all the images, the model is very sure that they are the correct signs.

For example: For the first image, the model is 100% sure that it is a work road sign.

Probability	**Prediction**
1	Work Road
0	Dangerous curve to the right
0	Go straight or right

0	General caution
0	Ahead only

Below is the softmax probabilities for all signs: