Data Set Summary & Exploration

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

The code for this step is contained in the third code cell of the IPython notebook.

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

The size of training set is ?
39209
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. Include an exploratory visualization of the dataset and identify where the code is in your code file.

The code for this step is contained in the fourth code cell of the IPython notebook.

Here is an exploratory visualization of the data set. It is a sample image chosen randomly from the training data set.

Design and Test a Model Architecture

1. Describe how, and identify where in your code, you preprocessed the image data. What tecniques 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.

The code for this step is contained in the fifth code cell of the IPython notebook.

As a first step, I decided to convert the images to grayscale because Converting images to gray scale reduces the total size of the dataset by 66% by going from 3 color channels to just 1. This will improve the speed of the algorithm without sacrificing much information.

expanding the dimensions of the images from (32x32) to (32x32x1), which is required for 2D convolutions in TensorFlow.

centering the image by subtracting the mean and divide by standard division

2. Describe how, and identify where in your code, you set up training, validation and testing data. How much data was in each set? Explain what techniques were used to split the data into these sets. (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, identify where in your code, and provide example images of the additional data)

The code for splitting the data into training and validation sets is contained in the sixth code cell of the IPython notebook.

To cross validate my model, I randomly split the training data into a training set and validation set. I did this by using train test split function provided by SKlearn kit.

My final training set had (29406) number of images. My validation set and test set had (9803) and (12630) number of images.

3. Describe, and identify where in your code, 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.

The code for my final model is located in the eighth cell of the ipython notebook.

My final model consisted of the following layers

Layer	Description	Type	Description
Layer 1	Input: 32x32x1. Output: 14x14x6	Convolution, Max pooling	Filter 5x5, Valid padding, Strides 1x1
Layer 2	Input: 14x14x6 Output: 5x5x16	Convolution, Max pooling	Filter 5x5, Valid padding, Strides 1x1
Flatten	Input: 5x5x16 Output: 400	Flatten	
Layer 3	Input: 400 Output: 120	Fully Connected Layer	
Layer 4	Input: 120 Output: 84	Fully Connected Layer	
Layer 5	Input: 84 Output: 43	Fully Connected Layer	

4. Describe how, and identify where in your code, 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.

The code for training the model is located in the ninth cell of the ipython notebook.

To train the model, I used an Adam Optimizer with default settings, batch size of 128, epochs of 10, Hyperparameters: mean = 0, sigma = 0.1, Learning rate = 0.001 and dropout of 0.5 to avoid over fitting .

5. Describe the approach taken for finding a solution. 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.

The code for calculating the accuracy of the model is located in the twelve cell of the Ipython notebook.

My final model results were:

training set accuracy of ?
90%
validation set accuracy of ?
96%
test set accuracy of ?
92%
What architecture was chosen?
lanet-5 architecture with dropout 0.5, and L2 regularization.
Why did you believe it would be relevant to the traffic sign application?
because it is simple and train the data faster than any other architectures it isn't as accurate as others but it was efficient to me
How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?
as i mentioned above it is not as accurate as other architectures but it was efficient to me.

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. Identify where in your code predictions were made. 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).

The code for making predictions on my final model is located in the 14th cell of the Ipython notebook.

Here are the results of the prediction:

Image	Prediction
Yield	Yield
Work Zone	No Entry
Stop	Stop
No U Turn	Ahead Only
No Entry	No Entry

The model was able to correctly guess 3 of the 5 traffic signs, which gives an accuracy of 60%.