My project goes according to the following,

Loading the pickled data (Cell 1):

Where I loaded the files (train, valid and test) features and labels.

Show Basic summery of the data (Cell 2):

Where I printed very basic info about the dataset.

- Number of training and testing images
- Number of classes
- Image size

A Random Sample Visualization (Cell 3):

Where I showed some pictures (40) of the dataset labeled with their classes.

Preprocessing (Cell 4):

In my submission preprocessing consists of two steps:

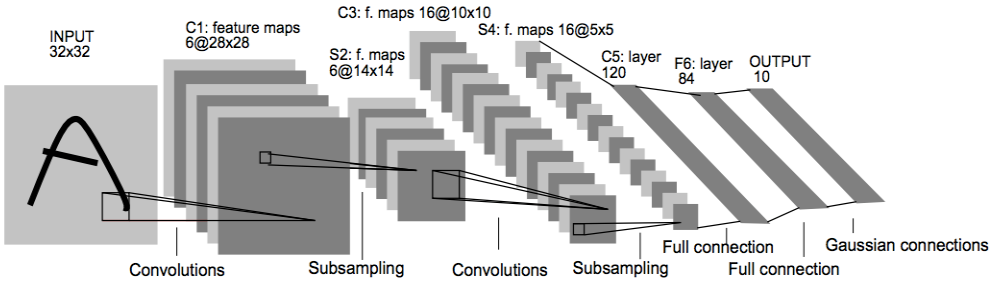
1- Convert to Grayscale, it helps to reduce training time of the network.
2- Normalize data to the range (-1:1), this was done by subtracting 128 and dividing by 128 as suggested in the lessons to provide around 0 mean.

Training, validation and Testing:

I used SciKit Learn train_test_split function to create a validation portion of the training set, it was 20% of the dataset that are used as data validation.

Model Architecture:

My model architecture was as follows:



Layer No.	Type	Input	Output
1	Convolutional	32x32x1	28x28x6
	Activation (Relu)		
	Pooling (max_pool)	28x28x6	14x14x6
2	Convolutional	14x14x6	10x10x16
	Activation (Relu)		
	Pooling (max_pool)	10x10x16	5x5x16
	Flatten	5x5x16	400
3	Fully Connected	400	120
	Activation (Relu)		
	Dropout		
4	Fully Connected	120	84
	Activation (Relu)		
	Dropout		
5	Fully Connected	84	43

Training the model:

Variable	Value
Batch size	100
Epochs	60
Learning rate	0.001
Mu	0
Sigma	0.1
Dropout keep probability	0.5

My approach:

I started with the Lenet implementation as suggested in the template and it turned out to work well, then I played around with the above parameters, sometimes the training time was very long when the learning rate was lower than 0.001 without noticeable improvements (at least for me).
When the test set accuracy reached 94.4%, I settled on the above values.

The new images test:

I searched for 10 new images resized then to 32x32 and tested them, the model got 60% of them right.

Using the model's SoftMax probabilities:

I used k=3 to see the top 3 most probable classes, and this was the output.