Project write up - Traffic Sign Classification

Dataset Summary:
As shown in Cell 2 , there are 39209 training images and 12630 test images

Exploratory visualization:
Cell 3 displays a random image from the training dataset

Preprocessing:
I attempted multiple preprocessing methodologies:

1. Converting images to gray-scale. This didn't make intuitive sense to me as the traffic sign colors denote some significance. Nonetheless I attempted with gray-scale visualization and got bad results. Accuracy was around 70%
2. Converting images to gray-scale and normalizing the images. This didn't work well either. Accuracy was around 70%.
3. Used the raw images with shuffling. This worked best, I got an accuracy of 95%

Model Architectures:
I followed the LeNet architecture and didn't make any modifications to the network layer. Here is my architecture. This is shown in cell 6.

Convolution (5,3,6)
Pooling (2,2)
Convolution (5,6,16)
Pooling (2,2)
Fully Connected(400,120)
Fully Connected(120,84)
Fully Connected Final (84,43)

Training:
I chose a 80 : 20 Traning set to Validation set split
I chose the following Hyper parameters -

- Learning Rate : 0.001
- Batch Size : 128
- Number of Epochs : 1000

I first trained the model with 10 epochs and obtained an accuracy of 96%
I then increased the number of epochs to 1000 and played around a bit with learning rate. 0.002, 0.003 etc but didn't find any significant improvement in accuracy which was hovering around 98%.

Solution Design:
I just stuck with the LeNet design as it is proven to work. I could have played around with the network a little by adding a few more conv layers but I am not convinced how that would help increase the accuracy.

New images:

I chose 6 random German traffic sign images on the web. They are pretty hard to classify as there are water-marks on those images. The cropping might help in alleviating the watermark issue.

Performance on new images:
The model gave the output for 3 of the 5 images correctly.
This is shown in Cell 13. The ordering is messed up a bit though.

Model Certainty - Log probabilities:
By looking at the top 5 log probabilities the model is certain about the classes as all the top probabilities are 1. Doesn't seem right. The model has a test data accuracy of 60% while the training accuracy is almost 98%. Seems to indicate that the model has overfit the training data and is terrible with new data.