1. Data preprocessing

The dataset preprocessing consisted of:

A. Converting to grayscale, which helps to reduce training time, while does not remove nay essential information.

B. Normalizing the data to the range (-1,1)

C. Data augmentation: random translation, scaling, warping, brightness. These techniques were used to generate more images to further balance the classes of the training set for better training.

D. One-hot encoding was used to binarized the 43 labels to be consistent with the softmax logits output of the cnn model.

2. Setup of the training, validation and testing data

A. the given dataset has already been splitted into training, validation and testing data.

B. However, the distribution of each class is not uniform. So data augmentation was used to rebalanced the classes a little bit to reduce biases towards classes with more samples. More specifically, classes with less than 800 samples were boosted to 800.

3. Model architecture

1. 5x5 convolution (32x32x1 in, 28x28x6 out)
2. ReLU
3. 2x2 max pool (28x28x6 in, 14x14x6 out)
4. 5x5 convolution (14x14x6 in, 10x10x16 out)
5. ReLU
6. 2x2 max pool (10x10x16 in, 5x5x16 out)

7. Flatten layer (5x5x6 in, 1x1x400 out)
8. ReLu
9. Fully connected layer (400 in, 120 out)
10. ReLu
11. Fully connected layer (120 in, 84 out)
12. ReLu
13. Fully connected layer (84 in, 43 out)

4. Model training hyper-parameters

Adam optimizer was used with

batch size: 100
epochs: 60
learning rate: 0.0009

5. Approach to take in coming up with a solution to this problem

A. First try to determine the learning rate from 0.001 to 0.00001

B. Different batch sizes were tried from 32 to 128, and finally 100 was chosen

C. After the learning rate and batch size were determined, and the accuracy kept increasing, tried to increase the epochs to train the model for longer time to achieve higher accuracy.

D. After training for 60 epochs, a validation accuracy of 0.937 was achieved.

6. Choose five candidate images of traffic signs and provide them in the report. Are there any particular qualities of the image(s) that might make classification difficult.

A. 8 images were chosen randomly from the web for testing.

*B. In order to get a good predictive result, these images were cropped into 32*32 size with the signs occupy the central part of the image.*

C. Then they were preprocessed by converting to gray-scale and normalizing.

7. Is your model able to perform equally well on captured pictures when compared to testing on the dataset?

The model predicted correct on all the 8 images. Although we cannot expect 100% accuracy on far more images. The correctness on these 8 images is a good sign that we have a good predictive model in hand for use.

8. Use the model's softmax probabilities to visualize the **certainty** *of its predictions,* tf.nn.top_k *could prove helpful here.*

```
TopKV2(values=array([[  1.00000000e+00,   7.06080492e-23,
2.12079193e-29],
       [  9.96401668e-01,   3.59558081e-03,   2.70966279e-06],
       [  1.00000000e+00,   9.33653654e-13,   7.59376039e-18],
       [  1.00000000e+00,   0.00000000e+00,   0.00000000e+00],
       [  1.00000000e+00,   4.20873086e-10,   8.24014088e-13],
       [  1.00000000e+00,   2.05218871e-14,   7.60531063e-15],
       [  1.00000000e+00,   3.64615504e-10,   1.19007108e-19],
       [  1.00000000e+00,   7.42192557e-11,   4.38634734e-22]], dtype=float32),
indices=array([[12, 40, 26],
       [24, 29, 25],
       [34, 38, 30],
       [38,  0,  1],
       [ 1,  2,  6],
       [ 3, 31,  5],
       [11, 30, 23],
       [18, 26, 11]], dtype=int32))
```

From the result we can see, the predictive model seems to be pretty sure about its prediction for the 8 testing images. The top guesses all have a very high probabilities very close to 100%. This low variance also indicates the effectiveness of the model.