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 used the numpy library to calculate summary statistics of the traffic signs data set. The result is shown as below:
   The size of training set is 34799.
   The size of the validation set is 4410.
   The size of the test set is 12630.
   The shape of a traffic sign image is 32x32x3.
   The number of unique classes in the data set is 43.
2. Include an exploratory visualization of the dataset.
   As shown in the jupyter notebook, I plotted all the unique 43 classes. Besides, I also plotted the histograms (one for each data set) which are shown below.
   The histograms for the training data set, test data set and validation data set are shown below. These plots show how data is distributed across the labels. From these plots, it can be observed that which label has the largest number of samples and which label has the smallest number of samples. Also, we can see that for all data set, the distributions of the data are unbalanced.

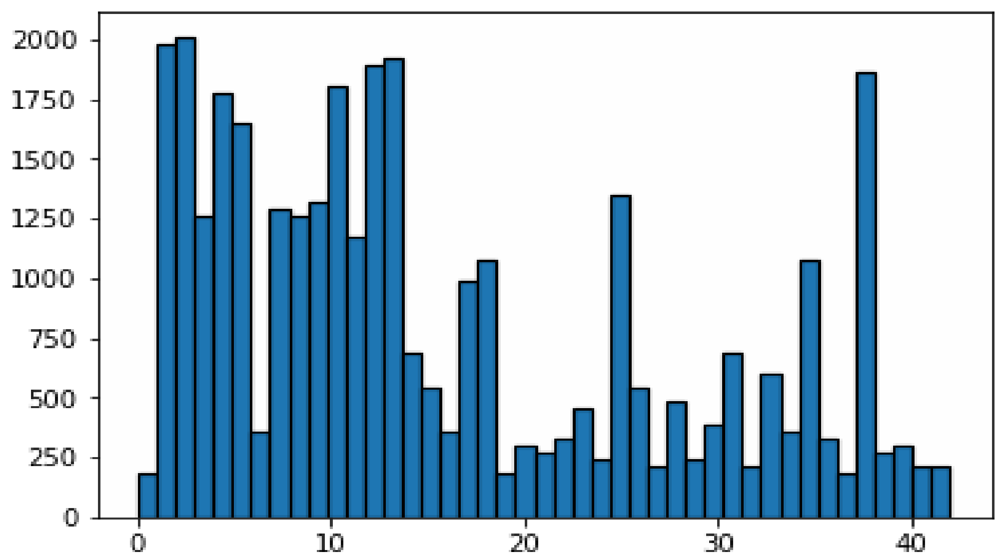


Figure 1: The histogram of the training data set.

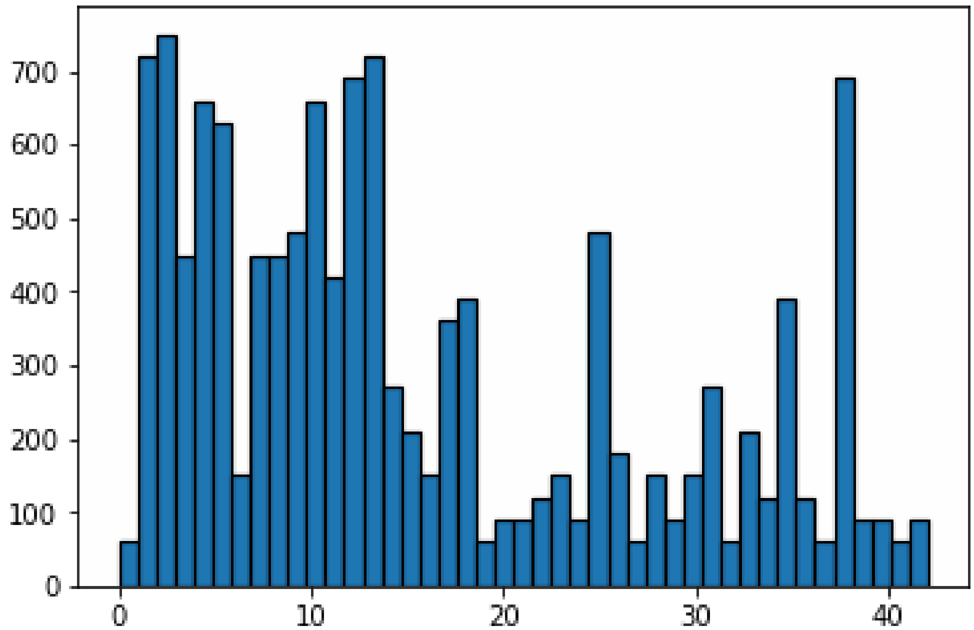


Figure 2: The histogram of the test data set.

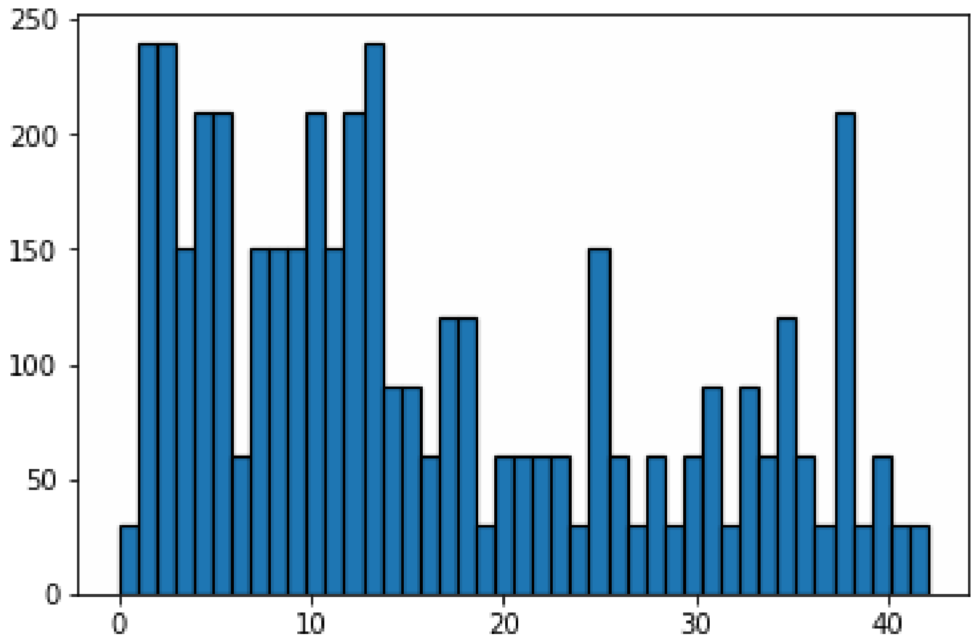
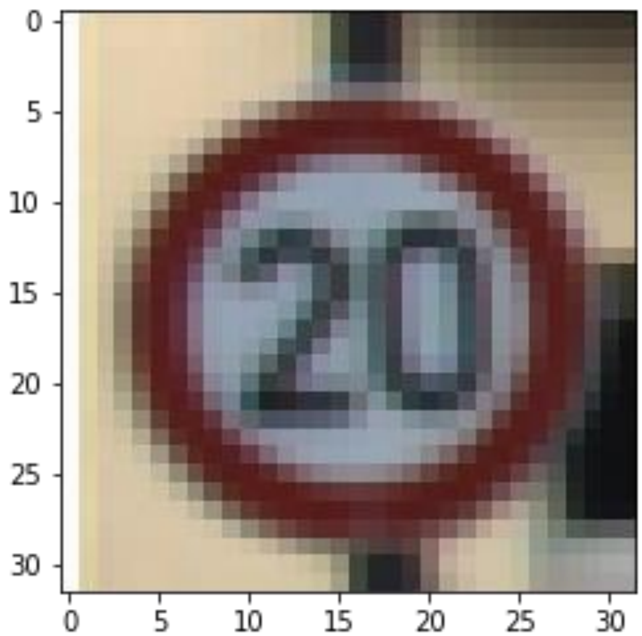
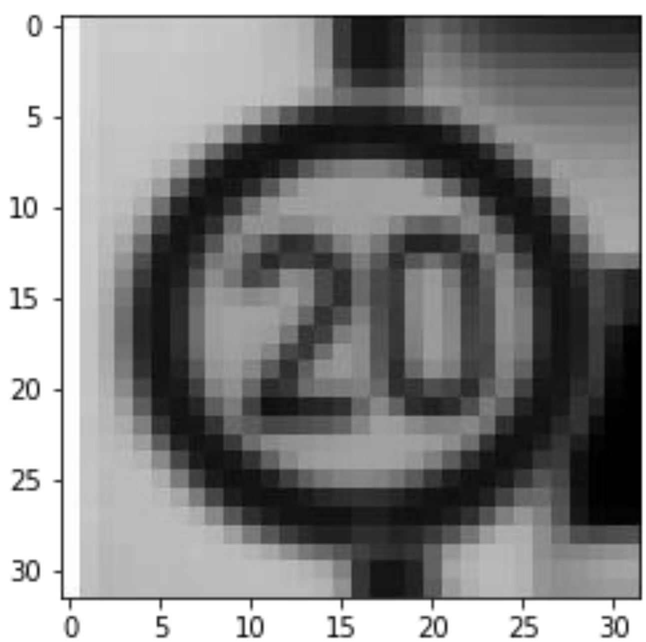


Figure 3: The histogram of the validation data set.

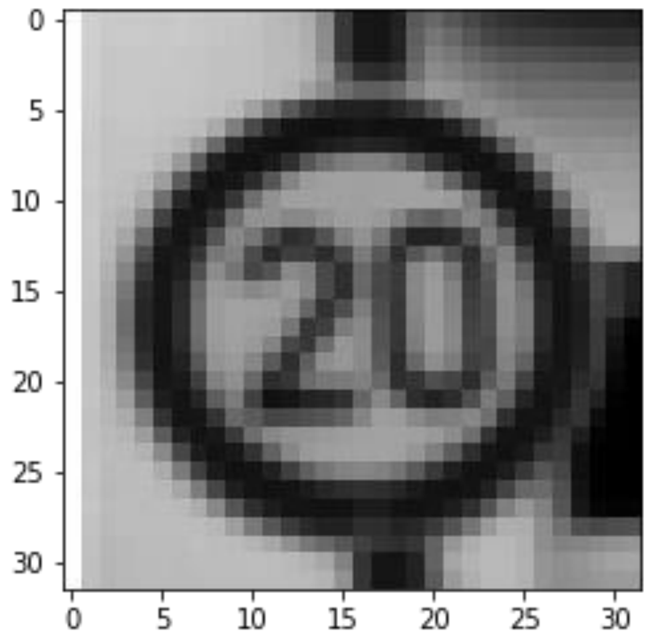
Design and Test a Model Architecture

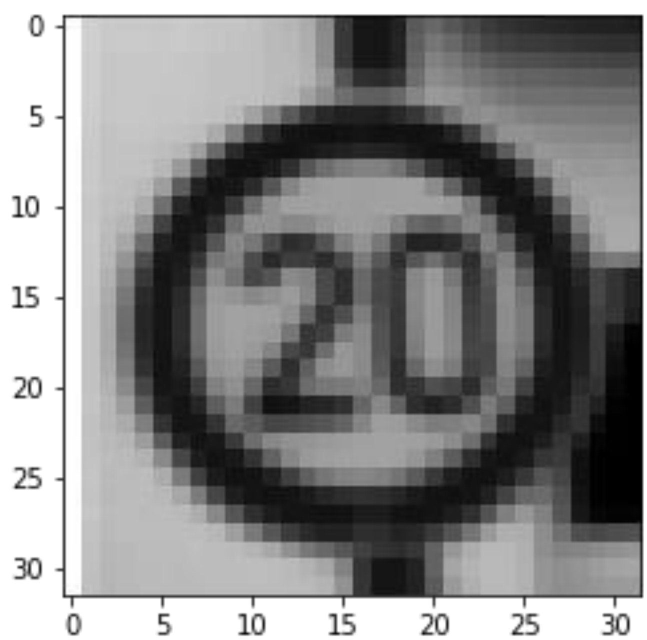
1. Describe how you preprocessed the image data.
   First, I decided to convert the images to grayscale since this can result in higher accuracy classification.
   Here is an example of a traffic sign image before and after converting to grayscale. (The top image is before converting to grayscale and the bottom image is after converting to grayscale)



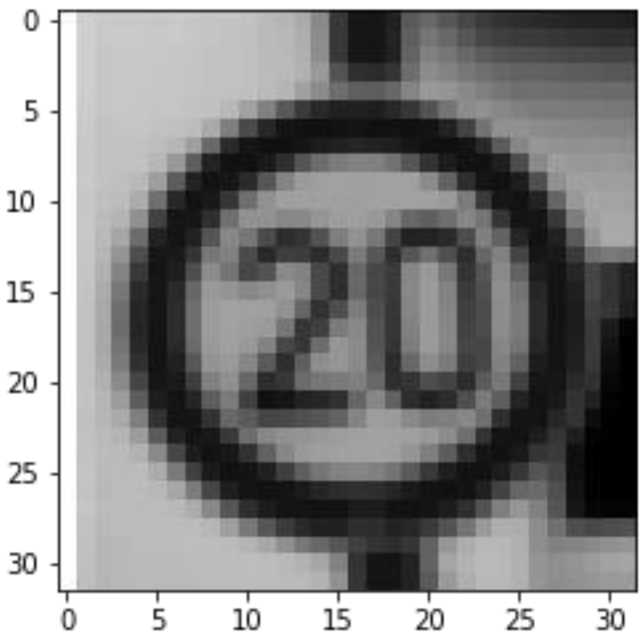


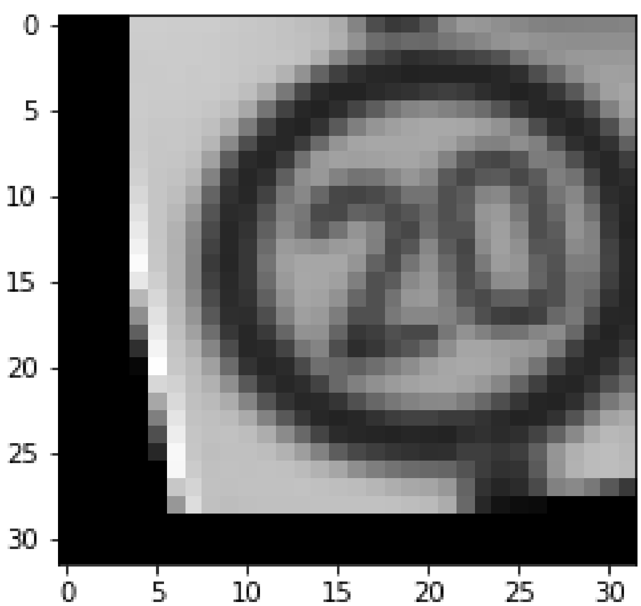
Then I normalized the images to the range between 0 and 1. The image must be normalized so the image data has zero mean and equal variance. Without normalization, the validation accuracy will be extremely low. So, the image must be normalized. Here is an example of traffic sign image before and after normalization. (The top image is before normalization and the bottom image is after normalization)





Then I choose to general additional data. From the data exploratory visualization, it can be concluded that the training data set is unbalanced. Therefore, I think I need to generate additional sample data so that the validation accuracy can be increased. To add more data to the training data set, I apply random rotation and random translation techniques for data augmentation. These two techniques are import image processing techniques and can help to reduce overfitting and increase the validation accuracy. Below is an example of traffic sign image before and after data augmentation. (The top image is before data augmentation and the bottom image is after data augmentation). From the two images below, it can be observed that the augmented image is slightly rotated, and the image size is slightly reduced.





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.
   The final model consisted of the following layers:

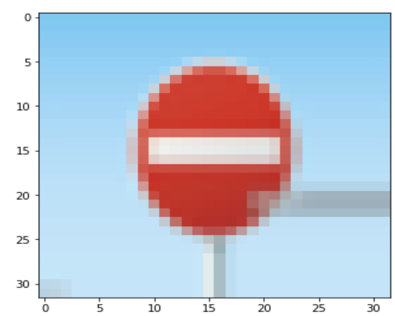
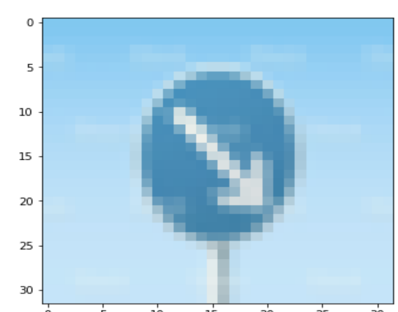
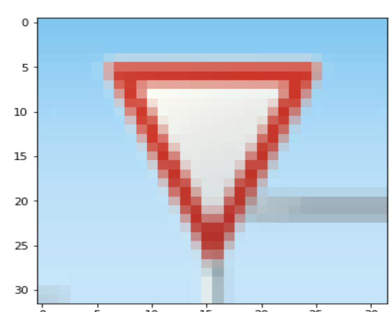
Layer	Description
Input	32x32x1 image
Convolution 5x5	Convolution layer
RELU	activation
Max pooling	Pooling layer
Convolution 3x3	Convolution layer
RELU	activation
Convolution 3x3	Convolution layer
RELU	activation
Convolution 3x3	Convolution layer
RELU	activation
Max pooling	Pooling layer
Flatten layer	
Fully connected layer 1	
RELU	activation
Drop out layer	regularization
Fully connected layer 2	
RELU	activation
Drop out layer	regularization
Fully connected layer 3	
Softmax	Return logits and then apply softmax on the logits

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 the Adam Optimizer and set the batch size equal to 128. The number of epochs is 10 and learning rate is 0.001.
4. Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93.
   My final model results were:
   The validation accuracy was 0.990 which is shown in 18th code cell.
   The test accuracy was 0.950 which is in 19th code cell.
   At first, I choose the model with only 3 convolution layers and each with one drop out layer, 2 max pooling layers, and 3 fully connected layers. The reason I choose this model because it is a typical and simple letnet cnn network and I add three drop out layers to prevent overfitting. However, after I run this model, the validation accuracy is very low. So, I choose to add one more convolution layer to make my model deeper and increase the depth of the filter k. This turns out to increase the validation accuracy significantly. But if I increase one more convolution layer, the validation accuracy seems to remain the same. Finally, I decide to include 4 convolution layers.
   Then I add the drop put layers between the fully connected layers and delete the drop out layers between the convolution layers. This also increase the validation accuracy greatly.

Also, I adjust the learning rate and the number of epochs. But it does not effect the validation accuracy too much. Therefore, I choose to not change the learning rate and number of epochs. Finally, I find out the reason why the model can not achieve the validation accuracy of 93. This is due to improper normalization. First, I just subtract the data by 128 and then divided by 128. But this method does not work for my model. Then I choose to normalize the data to the range between 0 and 1 and this results in a validation accuracy of 0.990.

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.

I think the first image may be hard to classify since it is a hexagonal. The shape is complex, so it might be hard for the classifier.

The speed limit 30km/h sign might be hard to classify since the classifier might not be able to distinguish between the speed limit signs (60 km/h, 30km/h etc). Those traffic signs look almost the same except the numbers are different. So, the classifier might not be able to identify the sign correctly.

The classifier might not be able to distinguish between the keep right and keep left sign because they look similar.

The no entry sign and the yield sign are easy for classifier since they have simple shapes.

2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set.

Image	Prediction
Speed limit (30km/h)	Speed limit (30km/h)
Keep right	Keep right
No entry	No entry
Yield	Yield
Stop	Priority road

The model achieved the test accuracy of 0.8. This was consistent with the prediction result. The prediction result showed that the classifier got 4 correct out of the 5 traffic signs which was a pretty good result.

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.

The first image speed limit 30 km/h

Softmax probabilities	prediction
0.98	Speed limit 30 km/h
0.016	Speed limit 20 km/h
0.0000014	Speed limit 50 km/h
0.00000031	End of speed limit 80 km/h
0.000000021	Speed limit 80 km/h

The model is very certain that this is a speed limit 30 km/h sign.

The second image keep right

Softmax probabilities	prediction
1	Keep right
0.0000000000016	Ahead only
0.0000000000012	Dangerous curve to the right
0.00000000000075	Beware of ice/snow
0.00000000000042	Turn left ahead

The model is 100% sure that this is a keep right sign.

The third image No entry

Softmax probabilties	prediction
1	No entry
0.00000000000063	Turn left ahead
0.00000000000000071	End of no passing by vehicles over 3.5 metric tons
0.000000000000000035	Stop
0.000000000000000020	Go straight or left

The model is 100% sure that this is a no entry sign.

The fourth image Yield

Softmax probabilities	prediction
1	Yield
0.000000000000023	Stop
0.000000000000000011	Speed limit (60 km/h)
0.000000000000000000022	End of speed limit (80km/h)
0.0000000000000000000075	No vehicles

The fifth image Stop

Softmax probabilities	Prediction
0.99	Priority road
0.00012	Roundabout mandatory
0.00011	Turn left ahead
0.000024	Road work
0.000022	End of no passing by vehicles over 3.5 metric tons

The model is very uncertain that this is a priority road sign, but this is a stop sign. The model fails to guess correctly this time. This proves that the stop sign is hard to classify to some extent.