Vinh Nghiem

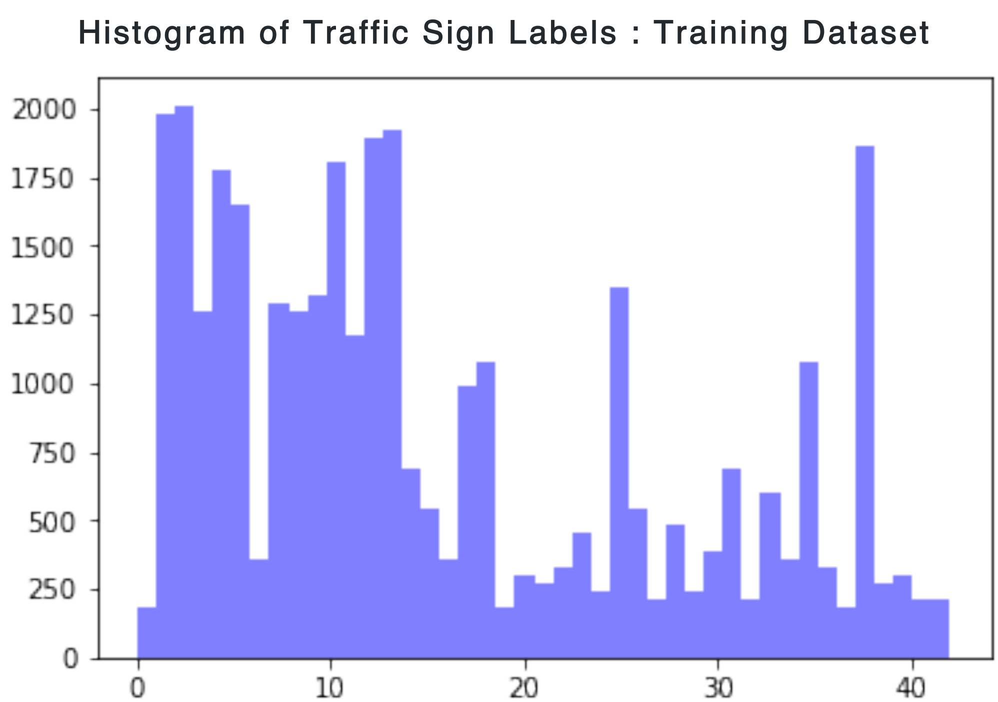
Project 2

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.

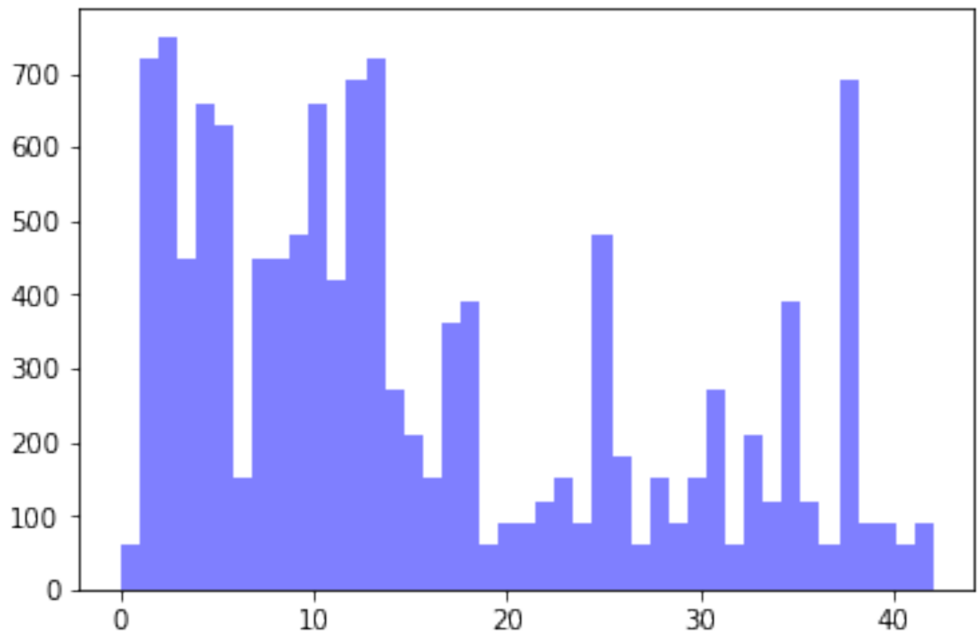
I used the numpy shape attribute to calculate summary statistics of the traffic signs data set:

- The size of training set is 34,799
- The size of the validation set is 4,410
- The size of test set is 12,630
- The shape of a traffic sign image is (32,32,3)
- The number of unique classes/labels in the data set is 43

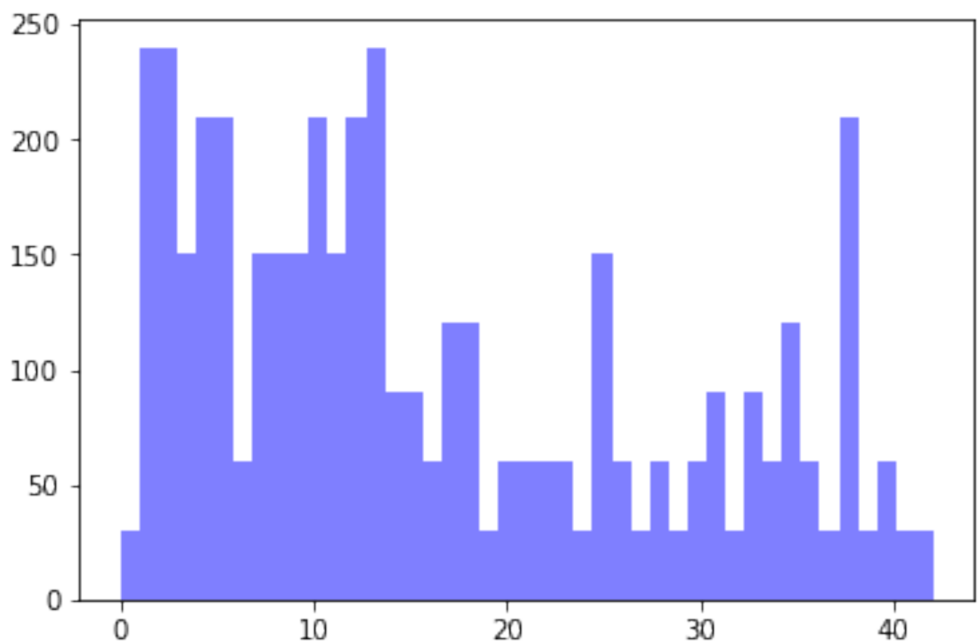
2. Include an exploratory visualization of the dataset.



Histogram of Traffic Sign Labels : Validation Dataset



Histogram of Traffic Sign Labels : Test Dataset



Design and Test a Model Architecture

1. Describe how you preprocessed the image data. What techniques 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.

I normalized the images so that all image values are scaled to the range of 0.1 and 0.9. The purpose of this normalization is for numerical stability, so that computations in the forward pass or back propagation algorithms do not result in large deviations from the true answer.

I initially converted the images to grayscale but got smaller accuracy for the validation set, so I kept the images in color. I believer keeping the images in color, with 3 channels, adds valuable information in distinguishing street signs.

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.

My final model consisted of the following layers:

Layer	Description
Input	32x32x3 RGB image
Convolution 5x5	1x1 stride, valid padding, outputs 28x28x6
RELU	
Dropout	0.8 keep probability during training
Convolution 5x5	1x1 stride, valid padding, outputs 24x24x16

Layer	Description
RELU	
Dropout	0.8 keep probability during training
Convolution 5x5	1x1 stride, valid padding, outputs 20x20x40
RELU	
Dropout	0.8 keep probability during training
Fully connected	
RELU	
Dropout	0.8 keep probability during training
Fully connected	
RELU	
Dropout	0.8 keep probability during training
Fully connected	
RELU	
Dropout	0.8 keep probability during training
Softmax	

Layer	Description

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 an Adam Optimizer that minimizes mean cross entropy. The batch size was 128 and the learning rate 0.001. The number of epochs was chosen to be 20, the number of epochs where validation accuracy started to decrease, in order to avoid overfitting.

4. Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93. 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.

My final model results were:

- training set accuracy of 99.9%
- validation set accuracy of 97.4%
- test set accuracy of 94.9%

If an iterative approach was chosen:

- What was the first architecture that was tried and why was it chosen?

  The first architecture chosen was LeNet. It was chosen arbitrarily as a starting point and because LeNet was successful as an image classifier.

- What were some problems with the initial architecture?

  Images are first converted to grayscale in the original LeNet, whereas color is actually valuable in distinguishing traffic signs. Here, the images were left in 3 color channels.

Also, the original Lenet architecture did not preprocess the images with normalization. Here, images were normalized in the range of 0.1 to 0.9 to increase numerical stability.

Lastly, LeNet uses max pooling layers after every convolution, which is not effective at reducing overfitting as dropout layers. Here, maxpool layers were replaced with dropout layers with keep probabilities of 0.8 during training.

- How was the architecture adjusted and why was it adjusted?

  Dropout for every layer was also added, and replaced the previous LeNet maxpool operation. The dropout seemed to result in reduced overfitting compared to maxpool. This alone improved validation accuracy from 93% after normalization of images to 95%

  An extra hidden layer was added to the Lenet architecture, consisting of convolution, followed by RELU, followed by dropout (with 0.8 keep probability during training). The new convolution had a patch of 5x5 with output an output of 40 channels. This layer was added in order to extract more features. This improved validation accuracy from 95% to 97%

- Which parameters were tuned? How were they adjusted and why?

  The number of epochs was tuned to 14. This was found to be the point at which validation accuracy started to decrease and the number of epochs stopped here in order to avoid overfitting.

- What are some of the important design choices and why were they chosen?

  The most important important design decisions were :

  1) Convolution layers. This allowed adjacent pixel images to share weights and allowed recognition of the same object translated across the image. Also, the convolution layers allowed subsequent feature maps to become deeper so that more features could be recognized at that hidden layer.
  2) RELU activation with multiple hidden layers. This allowed for nonlinear decision boundaries.
  3) Dropout layers. This allowed for reduction of overfitting.
  4) Normalization of images with range between 0.1 and 0.9. This allowed for increase in numerical stability.

5) Keeping three color channels in the input images. This allowed for high accuracy, since color carries valuable information that helps distinguish among objects.

- 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:

 Right of Way at next intersection (11). This may be hard to classify because of the ambient clouds.

 Children Crossing (28). This may be hard to classify because of the trees in the background. Also, the sign is being viewed from an angle and not face on.

Stop (14). This may be hard to classify because of the nonuniform background.

 Priority Road (12). This may be hard to classify because of the building in the background and also the center of the sign looks like there is a graphic even though I think that is just a result of pixilation when reducing the image size to 32x32.

 Road Work (25). This may be hard to classify because of the glare on the sign. Also, there are some clouds in the background.

Image	Prediction

Right of way At next intersection	Right of way At next intersection
Children Crossing	Children Crossing
Stop	Stop
Priority Road	Priority Road
Road Work	Road Work

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

Here are the results of the prediction:

The model was able to correctly guess 5 of the 5 traffic signs, which gives an accuracy of 100%. This compares to the 94.9% accuracy on the test set. The 100% accuracy was very surprising to me, given that the images had some irregularities that I believed would make accurate prediction challenging, but also impresses me on the power of convolutional deep neural networks.

3. 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 code for making predictions on my final model is located in the 11th cell of the Ipython notebook.

For the first image:

Probability	Prediction

Probability	Prediction
1.0	Right of way
0	Speed Limit (20km/h
0	Speed Limit (30km/h
0	Speed Limit (50km/h
0	Speed Limit (60km/h

For the second image :

Probability	Prediction
1.0	Children crossing
0	Speed Limit (20km/h
0	Speed Limit (30km/h
0	Speed Limit (50km/h
0	Speed Limit (60km/h

For the third image :

Probability	Prediction
1.0	Stop sign

Probability	Prediction
0	Speed Limit (20km/hr
0	Speed Limit (30km/hr
0	Speed Limit (50km/hr
0	Speed Limit (60km/hr

For the fourth image :

Probability	Predictio
1.0	Priority Ro
0	U-turn
0	Yield
0	Bumpy Ro
0	Slippery Ro

For the fifth image :

Probability	Prediction
1.0	Road Work
0	Speed Limit (20km/hr

Probability	Prediction
0	Speed Limit (30km/h
0	Speed Limit (50km/h
0	Speed Limit (60km/h

I was extremely surprised by the top-5 predictions for each image. I am getting 100% prediction on the correct image and 0% for the next four predictions, for every image. Something has to not be right, but I double checked the code and the procedure, then triple checked, then quadruple checked, and more, it looks like everything is sound. So, I am in disbelief, but at the same time, I am extremely impressed by the performance of deep learning and convolutional networks .