Project: Traffic Sign Recognition

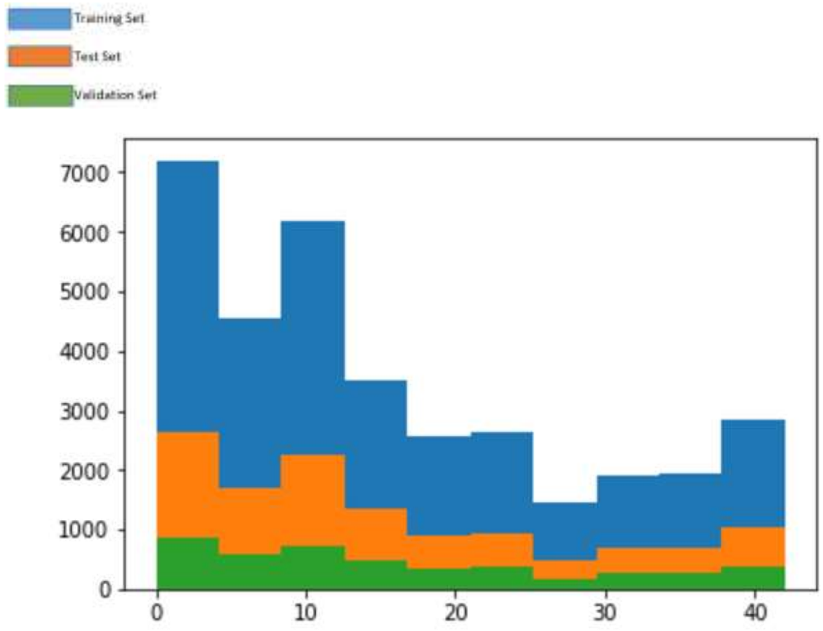
The goals / steps of this project are the following:

- Load the data set (see below for links to the project data set)
- Explore, summarize and visualize the data set
- Design, train and test a model architecture
- Use the model to make predictions on new images
- Analyze the softmax probabilities of the new images
- Summarize the results with a written report

Data Set Summary & Exploration

1. Used the pandas library to calculate summary statistics of the traffic signs data set:
   - The size of training set is 34799
   - The size of the validation set is 4410
   - The size of test set is 12630
   - The shape of a traffic sign image is 32x32x3
   - The number of unique classes/labels in the data set is 43

2. Exploratory distribution of 43 classes' traffic signs in training, validation and test set and plotted histogram.

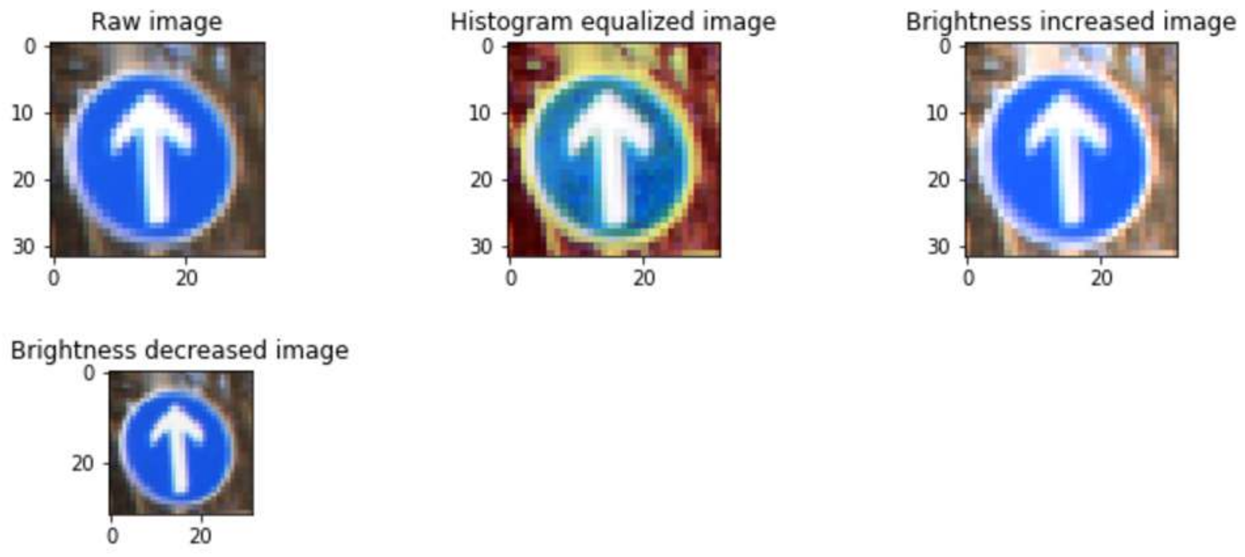


Histogram

In all the sets, there are good number of images for all 43 classes of traffic signs. Distribution of images look similar in all sets (train,valid,test). Though some traffic signs have lot of images than others. As a consequence, the traffic signs which has less number of images may not be learned properly.

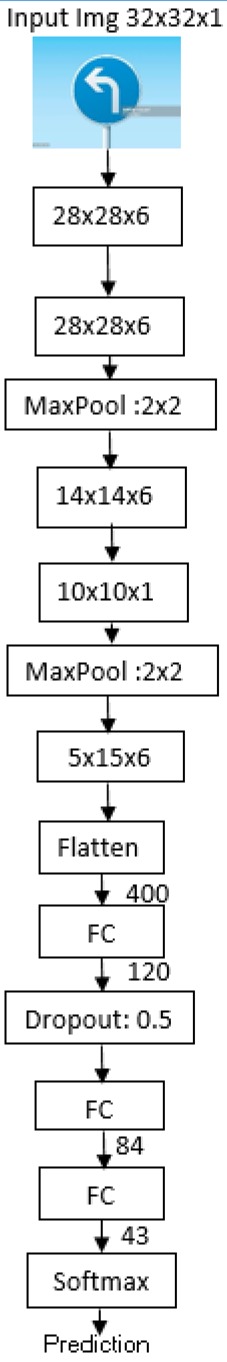
Design and Test a Model Architecture

1. No normalization was performed, because would like to keep the data as original as possible. However augmented the training set by applying below transformations on the image.
    1. Histogram equalized images
        a. Having histogram equalized images increases the brightness and contrast of the images. Hence the model can recognize the images with different contrasts.
    2. Brightness increased images
    3. Brightness decreased images
        a. In real world, the brightness of the traffic signs vary based on the lighting. Hence decided add images with increased and decreased brightness. Assumption is that the model can train to recognize signs in varying lighting conditions (low,bright,etc).



Original data set is of size 34799. After appling the above mentioned operations on each of the images final augmented data set is of size
34799*4 = 139196

2. The final model consisted of the following layers:



3. To train the model, AdamOptimizer is used which optimizes softmax cross-entryopy calculated from softmax of logits and one-hot encoding. The following hyperparamet configuration are used:
EPOCHS = 15
BATCH_SIZE = 128
dropout = 0.5

4.The accuracy calculated from the model as follows

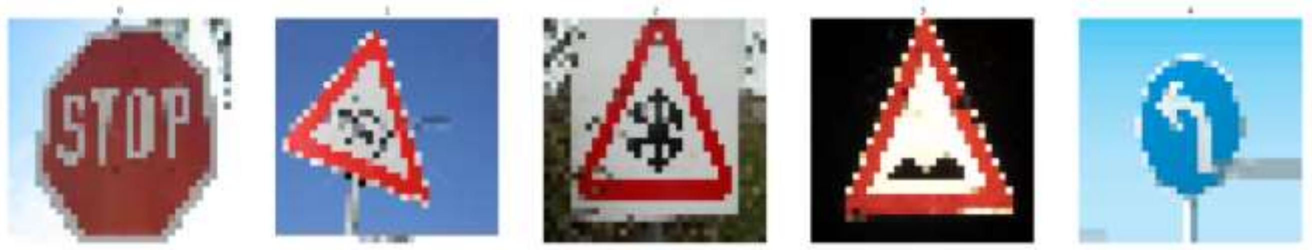
- training set accuracy of 95.2%

- validation set accuracy of 95.2%
- test set accuracy of 93.4%

Chose an architecture based on LeNet architecture explained in the previous module of this course. Since this network have convolution layers, max pooling and fully connected layers it is good for image recognition. I kept the filter size and internal layers' size same as the LeNet architecture. The architecture is adjusted in the final output layer which contain 43 outputs to represent 43 traffic sign classes. Also added dropout of 0.5 in the first fully connected layer in the network. It helps in avoiding overfitting and learning feature-maps of the image. Without dropout I got accuracy of less than 90% only. Tried with few other dropouts 0.4, 0.6 but then the accuracy got low due to under fitting. Hence finalized dropout of 0.5.

Test a Model on New Images

1. Here are five German traffic signs used to test the model:



- First image(Stop sign) is has some noise in the top right. That may pose some challenge.
- Second image(Bicycle crossing): The image is distorted too much when its size is scaled down. And also the angle of the image is not straight.
- Third image(Beware of ice and snow): Image has lot of distortion caused by scaling down the image size. But main difficulty would be due to unnecessary part in the background of traffic sign board. Also there are some leaves in front of the Traffic sign which might pose difficulty in recognizing the actual sign.
- Fourth image(Bumpy road): The space surrounding the sign is fully black. I don't think it would pose a problem, nevertheless something to watch out for.
- Fifth image(Turn left ahead): Image is similar to "Ahead only" sign and also image is shifted to right instead of centered.

2. **Output of cell[20] from the ipython notebook:**

```
image 0
Predicted: Stop
Actual: Stop
image 1
Predicted: Slippery road
Actual: Bicycles crossing
image 2
Predicted: Right-of-way at the next intersection
```

Actual: Beware of ice/snow
image 3
Predicted: Bumpy road
Actual: Bumpy road
image 4
Predicted: Ahead only
Actual: Turn left ahead

The model was able to correctly guess 3 of the 5 traffic signs, which gives an accuracy of 60%. This is not matching with test set accuracy of 94%. However the images were not correctly cropped and had some noise in the images. The model can be improved further to preprocess the image to cropout unnecessary details and smooth out any noises in the image.

3. The code for making predictions on my final model is located in the 21st cell of the Ipython notebook.

For the first image, the model is very sure that this is a stop sign (probability of 0.99), and the image does contain a stop sign.
The top five soft max probabilities were

Image - Stop
0.99747 Stop
0.00179 Yield
0.00053 No entry
0.00011 Speed limit (60km/h)
0.00005 General caution

For Bicycles crossing image, the model didn't recognize at all. I suspect it may be because there were n ot enough images to train it.
Image - Bicycles crossing
0.99919 No entry
0.00033 Stop
0.00010 Priority road
0.00008 Yield
0.00006 Road work

For Beware of ice/snow also the model didn't recognize at all. I suspect it may be because there were n ot enough images to train it.
Image - Beware of ice/snow
0.70886 Road narrows on the right
0.21813 Double curve
0.04221 Right-of-way at the next intersection
0.02937 Pedestrians
0.00044 Bicycles crossing

For Bumpy road, the model is reasonably sure (0.66) that it is Bumpy road sign.

<u>Image - Bumpy road</u>

0.66003	Bumpy road
0.33969	Bicycles crossing
0.00024	Road work
0.00003	Wild animals crossing
0.00000	Road narrows on the right

For Turn left ahead the model is reasonably sure (0.64) that it is Turn left ahead

<u>Image - Turn left ahead</u>

0.64129	Turn left ahead
0.35871	Keep right
0.00000	Go straight or right
0.00000	Dangerous curve to the right
0.00000	Ahead only