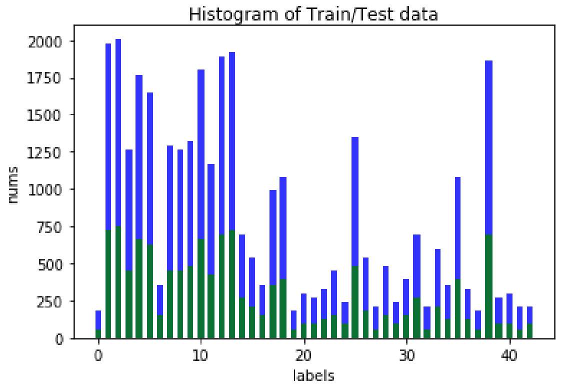
lProject: Traffic_Sign_Classifier
Name: Dongdong Wu

1.Data Set Summary & Exploration
training data / test data is provided. the result of data summary as below:

```
"Number of training examples = 34799 # size of training data
Number of testing examples = 12630 # size of test data
Image data shape = (32, 32, 3) # shape of data
Number of classes = 43 # labels number in training data"
```

the code is on "Provide a Basic Summary of the Data Set Using Python, Numpy and/or Pandas" part
labels Chart about train / test data



2.Model Architecture and Arguments Setting

a. since the data is already 32*32 format, the preprocess only do shuffle for training and test data set, no translation / rotation / scaling
shuffling is for guarantee test accuracy after updating parameters, normalization might help because some image histogram contrast is limited which can be considered as improvement

b. i split shuffled training images set with batch_size = 128 which is the same for validation and test set. The code is on 9th part cell
I pass images into model one batch by one batch continuously.
the final test image is from GTSRB data set. the final test image is not always 32*32 shape and ppm format.
I need load them by skimage lib, and resize them to 32*32 by numpy. the code is on 11th part cell.

c. based on what learned, I use Lenet architecture to build my CNN model. since this project is just for classification without location identification, and test image size is also quite small, so Lenet should be good enough

version1:
Architecture:

```
[ Layer              Description  ]
 Input               32x32x3 RGB image
 Convolution         5x5,1x1 stride, valid padding, outputs 28*28*6
```

```
RELU                YES
Max pooling         2x2 stride, outputs 14x14x6
Convolution_2       5x5,1x1 stride, valid padding,outputs 10*10*16
RELU                YES
Max pooling         2x2 stride, outputs 5x5x16
Fully connected     mean = 0, sigma = 0.1, input 400, output 120
Fully connected     mean = 0, sigma = 0.1, input 120, output 84
Fully connected     mean = 0, sigma = 0.1, input 84,  output 10
Softmax             output = 10

Optimizer = AdamOptimizer, Learning rate = 4*1e-4
Batch_Size = 128
Epoch = 100

Result :
      training set accuracy = unrecorded
      validation set accuracy = 0.89
      test set accuracy = 0.887
```

version 2:

A. Since result is about 0.9, then we need improve accuracy to above 0.93. the updates on:

1. do contrast normalization on batch raw images which will help detecting same label with different image contrast.

2. add two convolution layers before current convolution layers which use SAME padding to guarantee same output size.

B. Result improved:

validate set accuracy from 0.89 -> 0.94 ~ 0.95

test set accuracy = 0.94

3. Test result Analysis

final test images is from GTSRB data set in which images format is ppm, and the size is not 32 * 32.

Before the test, I need load ppm file by skimage and resize them to 32*32 by numpy

I pass 5 images to model, the result:

Version 1:

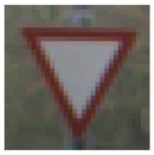
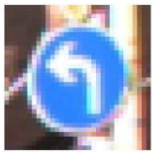
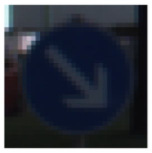
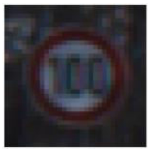
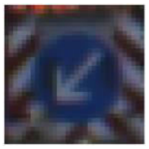
```
TopKV2(values=array([ 0.29774481,  0.07255951,  0.05254821,
0.04730011,  0.04410096],
[ 0.26439881,  0.11065474,  0.04948991,  0.04643316,  0.04405318],
[ 0.4554905 ,  0.04427201,  0.0392456 ,  0.03168693,  0.02614531],
[ 0.16883691,  0.10111474,  0.09836857,  0.07256201,  0.06892448],
[ 0.21717823,  0.0776408 ,  0.07398154,  0.05089254,  0.04290496]),
indices=array([31, 19, 21, 23, 29],[4, 1, 2, 0, 7], [13, 38, 41, 36,  1], [4, 7,
1, 5, 2],[ 8, 15,  9,  7,  3], dtype=int32))
```

Version 2:

1. add functions to evaluating image brightness / contrast, brightness just to get mean value of image contrast is use simple function to calcualte effective width of histogram

the code is on part 51:

a. test image

1.  2.  3.  4.  5. 

b.Image Quality Information:

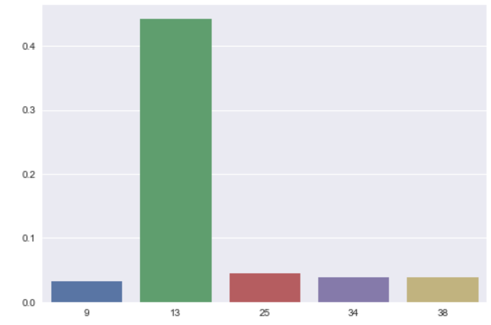
brightness	contrast
84.4046223958	171
114.790364583	236
54.859375	114
31.0026041667	47
117.527994792	237

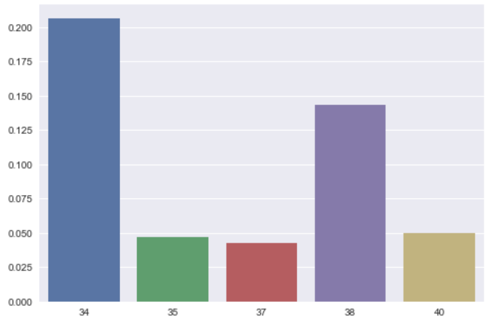
since i add contrast normalization functions before training, so the contrast difference should not effect that much

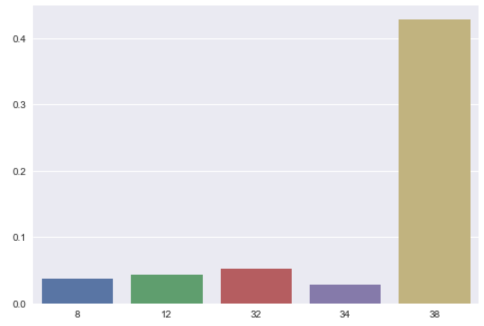
2. test result of new test image

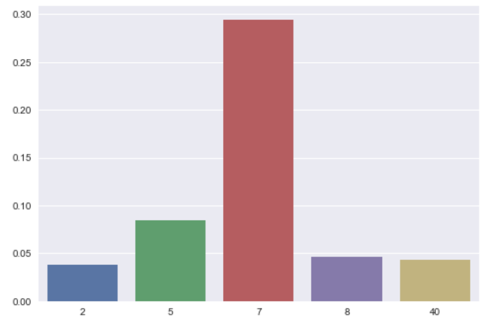
```
TopKV2(values=array([ 0.44195893,  0.0447876 ,  0.03804527,  0.03787285,
0.03265823],
[ 0.20635591,  0.1434537 ,  0.05008421,  0.04680218,  0.04237425],
[ 0.42858192,  0.05239325,  0.04333019,  0.03751702,  0.0289826 ],
[ 0.29404119,  0.08428058,  0.04591336,  0.04283585,  0.03809851],
[ 0.30415782,  0.05906209,  0.04737229,  0.03950538,  0.03904079],
dtype=float32), indices=array([13, 25, 38, 34,  9],
[34, 38, 40, 35, 37], [38, 32, 12,  8, 34], [ 7,  5,  8, 40,  2], [39, 37, 33,
4, 40], dtype=int32))
```

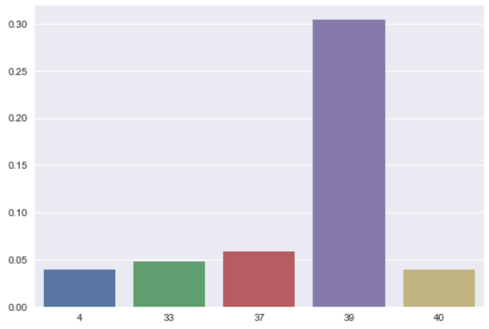
Histogram Chart

img1: 

img2: 

img3 :


img4:


img5:


3. Analysis about the result.

the final test sample is just 5 images as same as request, so the accuracy is higher which = 1 comparing to test images, however when i use 100 random picked images as test sample, the accuracy down to 0.994, and keep down to 0.963 if use 1000 images. I did not test all images since that is too big for whole GTSRB data set.
the accuracy measurement is on part of Analyze performance