SDCND - Project 2
Akshay Khole
akshay@reversehack.in

Dataset Exploration

Dataset summary:

The training and validation set are both normalized. The training set is also augmented by adding images that are pre-processed. In the pre-processing steps, I have created a image copy that is an equalised histogram of the original image. This creates an image that is brighter and well contrasted than original image.

This way, the training set has original images of not so good quality and color adjusted images that are more clear. This will help the model learn both kinds of images. The dataset size effectively doubles at this point.

Finally we are also normalizing all the images of training and validation set.

Exploratory Visualization:

Printing random images from the training dataset, printing images before and after pre-processing helps inspect how pre-processing is affecting the images.

Model Architecture:

The model architecture is pretty much out of the box of what we learnt in the lessons. I did not need to change any internals except ones recommended by David Silver in the project introduction videos.

Model Training:

To train the model, we feed the images to the LeNet Architecture and use the Adam Optimizer to converge to optimal solution. We also use cross entropy to train the data with different permutation of parts of the training set. This way the model has no bias towards any specific data slices for training and testing/validating.

Testing on new images:

The folder “p2-test-images” contains 5 test images downloaded from the internet. We have displayed the prediction, top 5 softmax probabilities and also displayed the images. The yield sign image is repeated because one is a clean image and one is from german google maps street view.

The output is printed here again:

```
--------------------------------------------------------------------
./p2-test-images/1.jpg

Prediction:
End of all speed and passing limits

Top 5 Softmax Probabilities are:
TopKV2(values=array([0.02672832,  0.02638547,  0.02612905,  0.02590962,  0.02583173],
dtype=float32), indices=array([32, 20, 26, 24, 31], dtype=int32))
--------------------------------------------------------------------
./p2-test-images/2.jpg

Prediction:
Yield

Top 5 Softmax Probabilities are:
TopKV2(values=array([ 0.03067694,  0.0303033 ,  0.03014619,  0.02993043,  0.02789136],
dtype=float32), indices=array([13, 21,  9, 33, 22], dtype=int32))
--------------------------------------------------------------------
./p2-test-images/3.jpg

Prediction:
Speed limit (50km/h)

Top 5 Softmax Probabilities are:
TopKV2(values=array([ 0.03793728,  0.03666491,  0.03582141,  0.03535096,  0.03477382],
dtype=float32), indices=array([ 2, 34, 11,  5, 26], dtype=int32))
--------------------------------------------------------------------
./p2-test-images/4.jpg

Prediction:
Go straight or left

Top 5 Softmax Probabilities are:
TopKV2(values=array([ 0.03025315,  0.02702978,  0.02655631,  0.02651535,  0.02607049],
dtype=float32), indices=array([37,  2, 12, 24, 42], dtype=int32))
--------------------------------------------------------------------
./p2-test-images/5.jpg

Prediction:
Vehicles over 3.5 metric tons prohibited

Top 5 Softmax Probabilities are:
```

```
TopKV2(values=array([ 0.02696596,  0.02628897,  0.02623263,  0.025883  ,  0.02575901],
dtype=float32), indices=array([16, 22,  3, 37, 23], dtype=int32))
```

As we can see the model isn’t very certain about it’s prediction. The reason could be that the training set isn’t training enough to include images of all variations. Also, it could be possible that the images I downloaded from the Internet are not suiting the how the model is trained against out data set. This results in a performance of 0% on new images. I am still trying to find ways to increase this %. Slack and mentors are helping a lot :) Hopefully I get to a solution soon and don’t get too wrapped up with P3!