UDACITY

PROJECT

Traffic Sign Classification

A part of the Self-Driving Car Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

2 SPECIFICATIONS REQUIRE CHANGES

Dear Student,

This was a very good attempt on this assignment.Overall you are already there in terms of achieving results.
All the best!!

Reviewer

Reviewer

Extra Important Material:

https://github.com/HFTrader/DeepLearningBook
http://www.fast.ai/
http://yerevann.com/a-guide-to-deep-learning/
https://medium.com/@vivek.yadav/dealing-with-unbalanced-data-generating-additional-data-by-jittering-the-original-image-7497fe2119c3#.obfuq3zde
https://medium.com/@vivek.yadav/improved-performance-of-deep-learning-neural-network-models-on-traffic-sign-classification-using-6355346da2dc#.tq0uk9oxy

Batch size discussion ::
http://stats.stackexchange.com/questions/140811/how-large-should-the-batch-size-be-for-stochastic-gradient-descent

Adam optimizer discussion::
http://sebastianruder.com/optimizing-gradient-descent/index.html#adam'

Dropouts ::
https://pgaleone.eu/deep-learning/regularization/2017/01/10/anaysis-of-dropout/

Files Submitted

The project submission includes all required files.

Dataset Exploration

The submission includes a basic summary of the data set.

The submission includes an exploratory visualization on the dataset.

Well tried!!

- An image has been displayed:

Suggestion

See these links for various ways of visualizing data;
Basic Data Plotting With Matplotlib
pylab_examples example code: histogram_demo_extended.py

Design and Test a Model Architecture

The submission describes the preprocessing techniques used and why these techniques were chosen.

Well done!!!

- Good job with grayscaling and normalization. : ✅

SUGESSTIONS

- For further improvements, here are some preprocessing techniques that have proven to work on this dataset:
- Can try random shuffling.
- Try Augmenting your training dataset with rotation, distortion, scaling ,translation and cropping images with varied contrast/brightness or normalized histograms) can improve the robustness of the model and give better results
- If images are not very bright, can do pre-processing like histogram equalization help in enhancing.

- Kindly make sure to give an explanation of each preprocessing technique.
- All the best !!! 👍

The submission provides details of the characteristics and qualities of the architecture, including the type of model used, the number of layers, and the size of each layer. Visualizations emphasizing particular qualities of the architecture are encouraged.

Well done here!!!!

- The characteristics and qualities of the final architecture like convolution layers used, number of layers, number of filters, relu,pooling, flattening and the number of fully connected layers have been adequately justified.✅

The submission describes how the model was trained by discussing what optimizer was used, batch size, number of epochs and values for hyperparameters.

Well tried here!!!! 👍

- You have only trained and evaluated your model and mentioned an optimiser user(AdamOptimiser) .Few more things needed to be added in your writeup.
- Below are the suggestions that can further help you to attempt this section well :

```
1)   Optimiser= ?
2) EPOCHS = ?        what  will happen if the epochs increases or decreases?
3)BATCH_SIZE = ?what  will happen to the accuracy if the batch size increases or decreases ?
4)Learning rate=?
```

```
5)Dropout probability=?
6)Hyperparameters: mu =?, sigma = ?, dropout = ?
added for dropout regularization, setduring the training of the classifier and for evaluation"
```

All the best!!!

The submission describes the approach to finding a solution. Accuracy on the validation set is 0.93 or greater.

- Please discuss your approach. Also validation accuracy has to be minimum 0.93 to pass this section. You can try data augmentation as generating additonal data also helped many student in improving accuracy.

Suggestions:

- While discussing your approach you can think on this question:

1. You can also discuss how did you choose the optimzer?
2. You can also discuss how did you tell a convolutional layer is well suited for this problem?
3. You can also discuss how did you choose the particular activation ?
4. You can also discuss how did you tune the hyperparameter?

    1. If a well known architecture was chosen:

        - What architecture was chosen?
        - Why did you believe it would be relevant to the traffic sign application?
        - How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?

- Check this link, it can help you:

https://medium.com/@gruby/convolutional-neural-network-for-traffic-sign-classification-carnd-e46e95453899

https://chatbotslife.com/german-sign-classification-using-deep-learning-neural-networks-98-8-solution-d05656bf51ad

https://chsasank.github.io/keras-tutorial.html

Test a Model on New Images

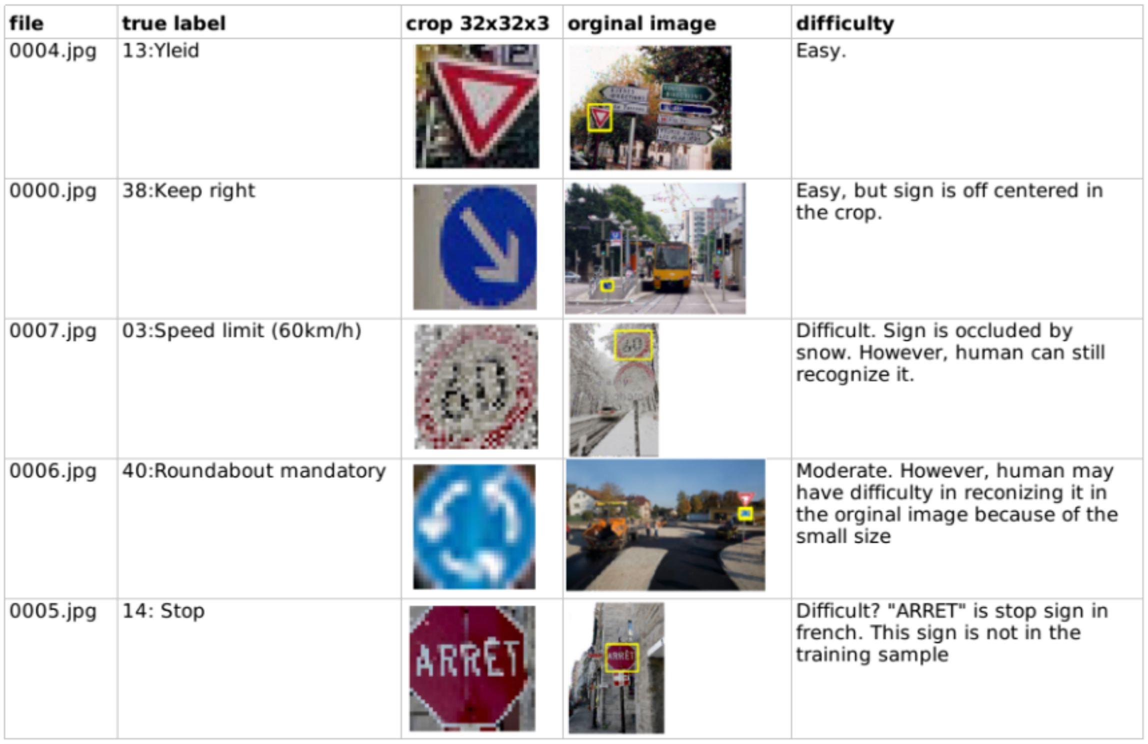
The submission includes five new German Traffic signs found on the web, and the images are visualized. Discussion is made as to particular qualities of the images or traffic signs in the images that are of interest, such as whether they would be difficult for the model to classify.

- All the candidate images chosen are appropriate.✅
- Please try to identify any single characteristics of the new images that might make it difficult for the model to classify.

Required

- Please still try to discuss the characteristics of any new image that might be it difficult for the model to classify. For example

```
        1.The Contrast of the image.
        2.The Angle of the traffic sign.
        3.Image might be jittered.
        4.The training data set does not include this traffic sign.
        5.Background Objects.
```



file	true label	crop 32x32x3	orginal image	difficulty
0004.jpg	13:Yleid			Easy.
0000.jpg	38:Keep right			Easy, but sign is off centered in the crop.
0007.jpg	03:Speed limit (60km/h)			Difficult. Sign is occluded by snow. However, human can still recognize it.
0006.jpg	40:Roundabout mandatory			Moderate. However, human may have difficulty in reconizing it in the orginal image because of the small size
0005.jpg	14: Stop			Difficult? "ARRET" is stop sign in french. This sign is not in the training sample

Figure.12: Test images from the internet

The submission documents the performance of the model when tested on the captured images. The performance on the new images is compared to the accuracy results of the test set.

Well done!!

- A fair comparison has been made between the prediction accuracy of the model on the captured images and those on the testing set.
- Acccuracy has been discussed here.

```
"The model was able to correctly guess 5 of the 5 traffic signs, which gives an accuracy of 100%."
```

The top five softmax probabilities of the predictions on the captured images are outputted. The submission discusses how certain or uncertain the model is of its predictions.

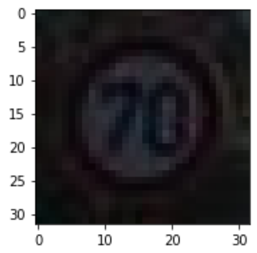
- Great job printing the softmax probabilities.✅
- The certainty of the model's predictions have been discussed well.✅

Suggestions

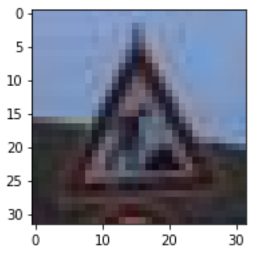
- Here is more example visualizing the softmax probabilities that you can check:

EXAMPLE: 1:

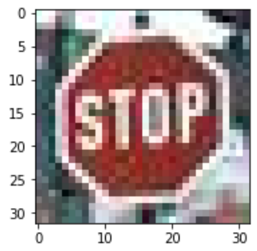
```
   tf.reset_default_graph()
  with tf.Session() as sess:
  top5 = sess.run(tf.nn.top_k(tf.nn.softmax(tf.constant(machine_answers))
   K=5))
  signnames = pd.read_csv('signamescsv')
  fig = plt.figure(figsize=(15,10)
 for i, indices in enumerate(top5,indices):
 a=fig.add_subplot(3,2,i+1)
     plt.imshow(images[i])
 for j,index in enumerate (indices):
     plt.text(35, (j*6)+2,"{}.{}".format(j+1, signames['SigName'](index]))
     plt.text(38, (i*6)+4.5,"{:.2%}" .format(top5.values[i][j]))
```



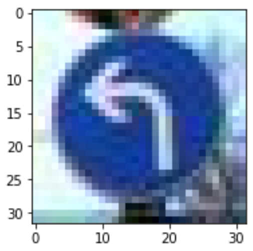
```
1. Speed limit (70km/h)
   99.97%
2. Speed limit (30km/h)
   0.03%
3. Speed limit (100km/h)
   0.00%
4. Roundabout mandatory
   0.00%
5. Keep left
   0.00%
```



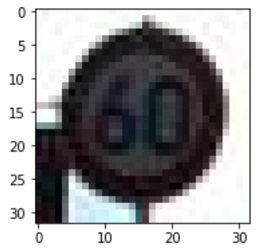
```
1. Road work
   100.00%
2. Beware of ice/snow
   0.00%
3. Wild animals crossing
   0.00%
4. Bumpy road
   0.00%
5. Bicycles crossing
   0.00%
```



```
1. Stop
   100.00%
2. Yield
   0.00%
3. No entry
   0.00%
4. Turn right ahead
   0.00%
5. Go straight or right
   0.00%
```



```
1. Turn left ahead
   100.00%
2. Keep right
   0.00%
3. Ahead only
   0.00%
4. End of no passing
   0.00%
5. No passing
   0.00%
```

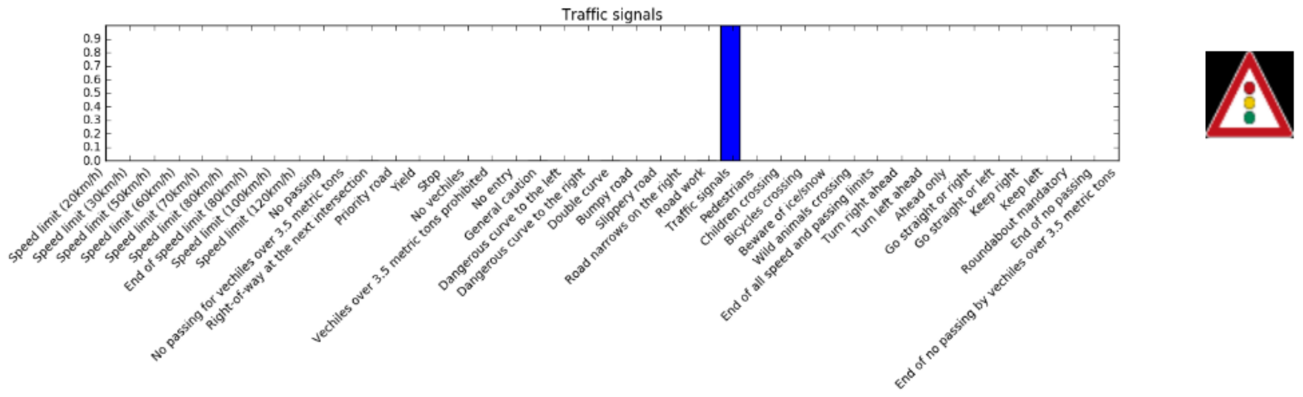


```
1. Speed limit (60km/h)
   99.91%
2. Speed limit (80km/h)
   0.09%
3. Speed limit (50km/h)
   0.00%
4. Ahead only
   0.00%
5. No vehicles
   0.00%
```

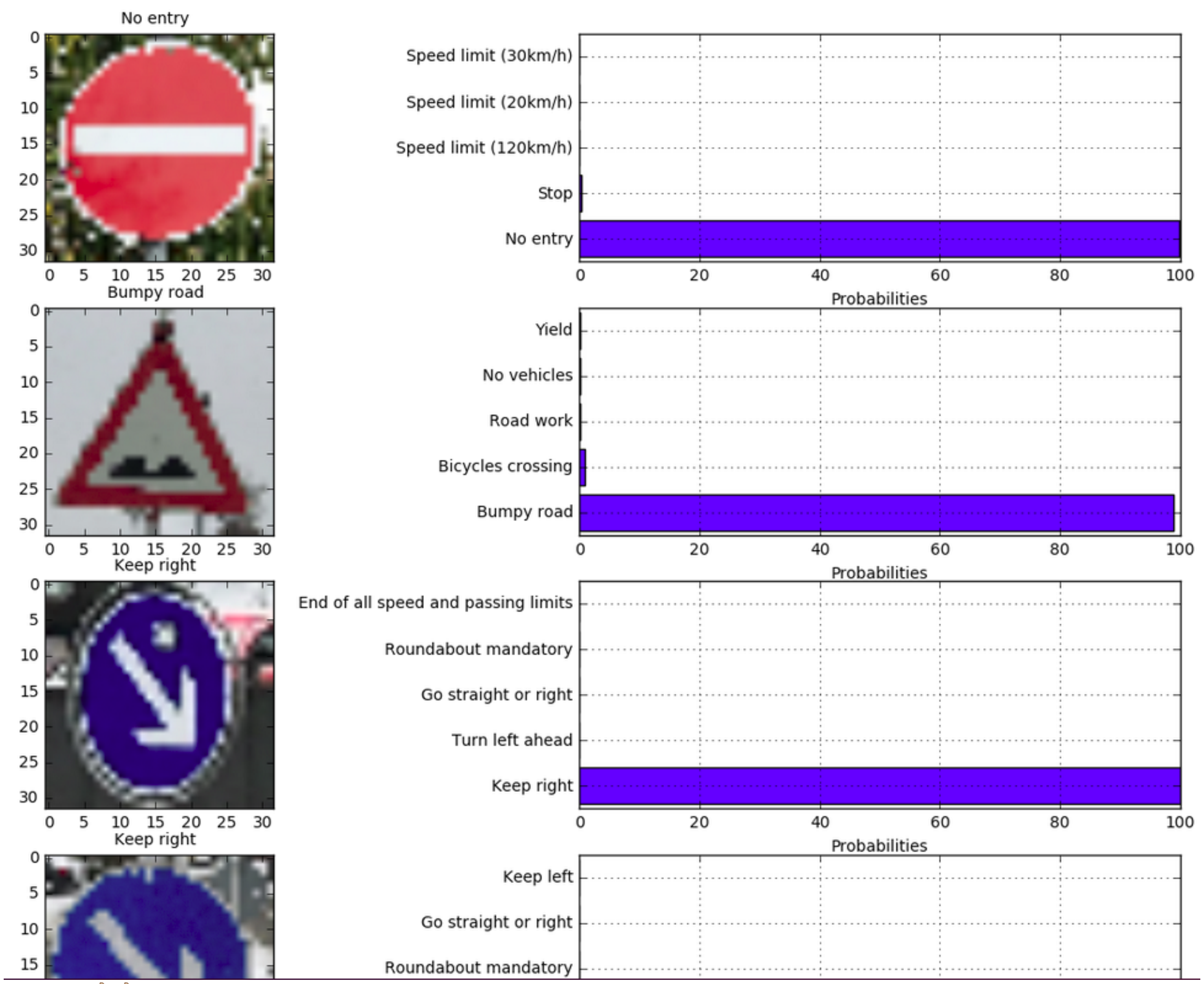
EXAMPLE: 2:

```
### Visualize the softmax probabilities here.
### Feel free to use as many code cells as needed.

for i, (labels, probs, candidate) in enumerate(zip(top_predictions.indices, top_predictions.values, candidates)):
    fig = plt.figure(figsize=(15, 2))
    plt.bar(labels,probs)
    plt.title(top_labels[i][0])
    height = candidate.shape[0]
    plt.xticks(np.arange(0.5, 43.5, 1.0), get_label_map('signnames.csv', np.unique(y_train)),  ha='right', rotation=45)
    plt.yticks(np.arange(0.0,1.0,0.1), np.arange(0.0, 1.0, 0.1))
    # Change the numbers in this array to position your image [left, bottom, width, height]
    ax = plt.axes([.75,0.25, 0.5, 0.5], frameon=True)
    ax.imshow(candidate)
    ax.axis('off')  # get rid of the ticks and ticklabels

plt.show()
```

And this is an example output from the code above:



EXAMPLE



Keep it up!!! 👍👍

RESUBMIT

DOWNLOAD PROJECT