Traffic Sign Classification

Lewis Collum

Updated: February 12, 2020

Started 02/09/2020

Downloading the German Traffic Sign Recognition Benchmark, GTSRB

The following downloads the dataset, unzips it, and moves the internal directories around (and changes their name) to match my prefered directory convention.

```
dataset=GTSRB_Final_Training_Images.zip
signNames=signnames.csv
dataDirectory="data"

function main() {
    downloadGtsrbDataset
    downloadSignNames
    configureDataDirectory
    echo "DONE!"
}

function downloadGtsrbDataset() {
    if [[ ! -d "$dataDirectory" ]]; then
        wget https://sid.erda.dk/public/archives/daaeac0d7ce1152aea9b61d9f1e19370/$dataset
        unzip $dataset -d "$dataDirectory"
        rm -f $dataset
    fi;
}

function downloadSignNames() {
    wget https://raw.githubusercontent.com/udacity/CarND-Traffic-Sign-Classifier-Project/master/"$sign
}

function configureDataDirectory() {
    mv "$signNames" "$dataDirectory"

    if [[ -d "data/GTSRB/Final_Training/Images" ]]; then
        mv "data/GTSRB/Final_Training/Images" "data/train"
        rm -rf "data/GTSRB"
    fi;
}

main
```

Common Variables

common.py

```
import pandas

signNames = pandas.read_csv("../data/signnames.csv")['SignName'].values

trainPath = "../data/train"
modelPath = "./model"
imageSize = 30
```

Class Names

```
import common
for i, name in enumerate(common.signNames):
    print(i, name)
```

```
 0 Speed limit (20km/h)
 1 Speed limit (30km/h)
 2 Speed limit (50km/h)
 3 Speed limit (60km/h)
 4 Speed limit (70km/h)
 5 Speed limit (80km/h)
 6 End of speed limit (80km/h)
 7 Speed limit (100km/h)
 8 Speed limit (120km/h)
 9 No passing
 10 No passing for vehicles over 3.5 metric tons
 11 Right-of-way at the next intersection
 12 Priority road
 13 Yield
 14 Stop
 15 No vehicles
 16 Vehicles over 3.5 metric tons prohibited
 17 No entry
 18 General caution
 19 Dangerous curve to the left
 20 Dangerous curve to the right
 21 Double curve
 22 Bumpy road
 23 Slippery road
 24 Road narrows on the right
 25 Road work
 26 Traffic signals
 27 Pedestrians
 28 Children crossing
 29 Bicycles crossing
 30 Beware of ice/snow
 31 Wild animals crossing
 32 End of all speed and passing limits
```

```
 33 Turn right ahead
 34 Turn left ahead
 35 Ahead only
 36 Go straight or right
 37 Go straight or left
 38 Keep right
 39 Keep left
 40 Roundabout mandatory
 41 End of no passing
 42 End of no passing by vehicles over 3.5 metric tons
```

Load Images using keras.preprocessing

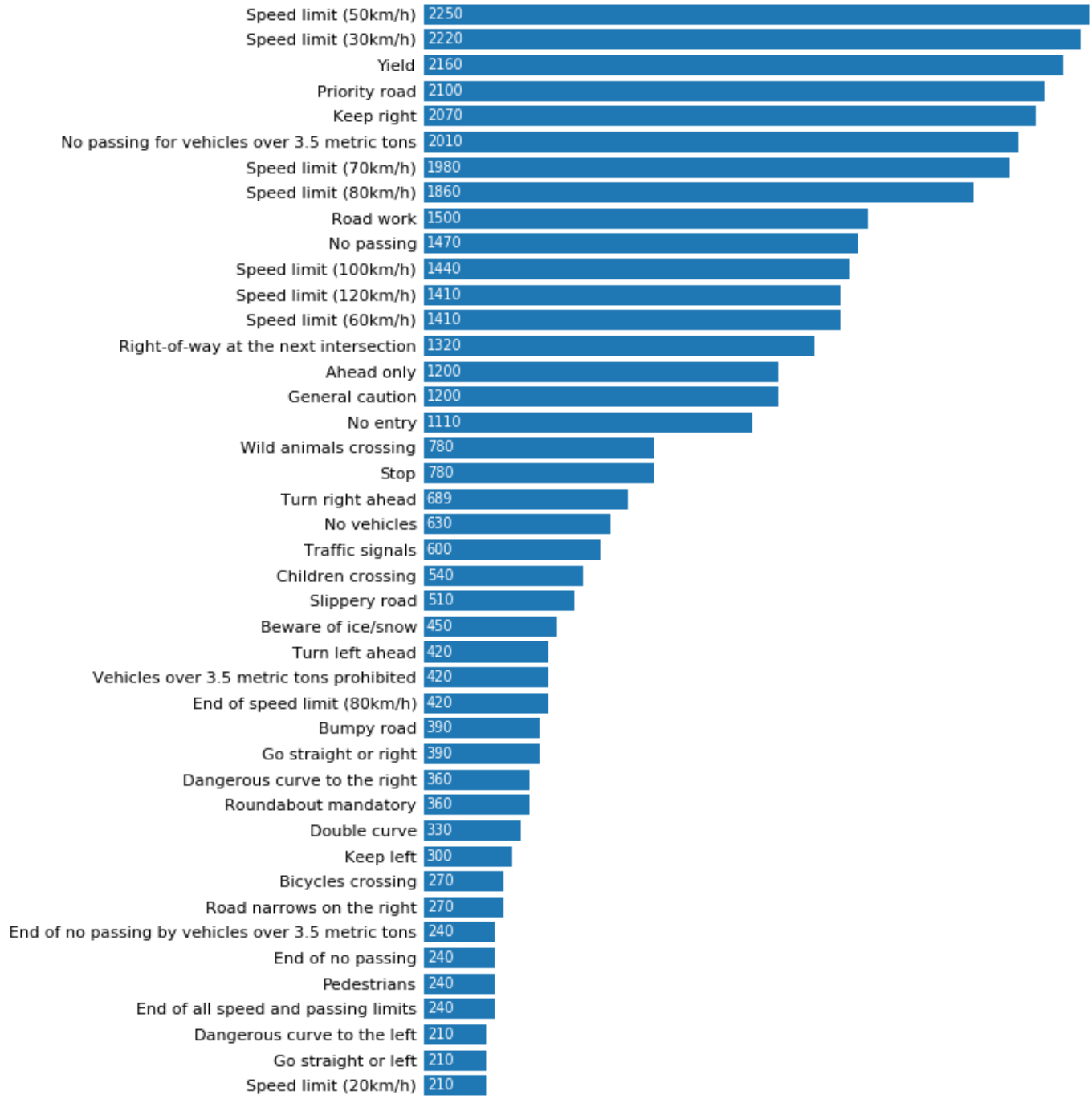
batch.py

```
from keras.preprocessing.image import ImageDataGenerator
import numpy

import common

size = 32

batchGenerator = ImageDataGenerator(rescale=1./255)
batch = batchGenerator.flow_from_directory(
    directory = common.trainPath,
    batch_size = size,
    shuffle = True,
    target_size = (common.imageSize, common.imageSize))

images, labels = batch.next()

classCount = len(labels[0])

def classAt(index):
    return numpy.where(labels[index] == 1)[0][0]

sampleClasses = batch.labels
sampleSize = batch.n
```

Training Set Distribution of Classes

```
import numpy
import matplotlib.pyplot as pyplot

import common
import batch

distribution = numpy.zeros(batch.classCount)
for i in batch.sampleClasses:
    distribution[i] += 1
```

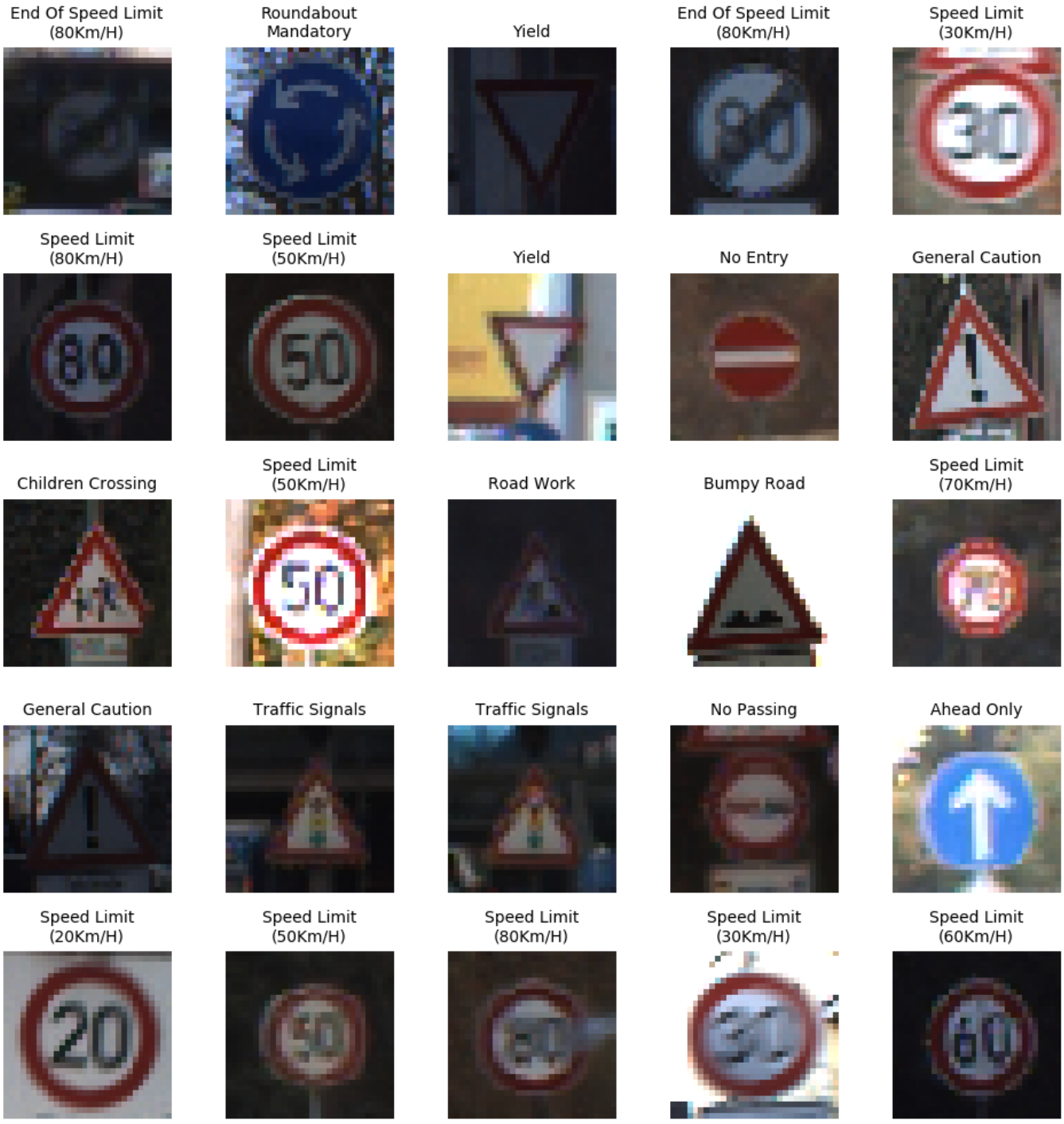
```
sortedIndices = distribution.argsort()
distribution = distribution[sortedIndices]
signNames = common.signNames[sortedIndices]

pyplot.figure(figsize=(8, 0.2*len(signNames)))
bars = pyplot.barh(signNames, distribution, align='center')
pyplot.yticks(fontsize=8)

pyplot.box(False)
pyplot.gca().get_xaxis().set_visible(False)
pyplot.gca().tick_params(length=0)

for bar in bars:
    print(bar.get_height())
    pyplot.gca().text(
        bar.get_x() + 10,
        bar.get_y() + bar.get_height()/2,
        str(int(bar.get_width())), va='center', color='white', fontsize=7)

pyplot.tight_layout()

pyplot.savefig('../figure/trainingSetDistributionOfClasses.png')
```



View a Batch

```
import matplotlib.pyplot as pyplot
from textwrap import wrap

import common
import batch
```

```
columns = 5
rows = 5

pyplot.figure(figsize=(2*columns,2*rows))
pyplot.subplots_adjust(hspace = .6)

for n in range(min(columns*rows, batch.size)):
    ax = pyplot.subplot(rows, columns, n+1)
    pyplot.imshow(batch.images[n])
    title = common.signNames[batch.labels[n] == 1][0].title()
    wrappedTitle = "\n".join(wrap(title, 18))
    pyplot.title(wrappedTitle, fontsize=10)
    pyplot.axis('off')

pyplot.tight_layout()
pyplot.savefig('../figure/batchView.png')
```



Resources

Tutorial

- https://towardsdatascience.com/recognizing-traffic-signs-with-over-98-accuracy-using-deep-learning-86737aedc2ab
- https://github.com/kenshiro-o/CarND-Traffic-Sign-Classifier-Project
- https://github.com/kenshiro-o/CarND-Traffic-Sign-Classifier-Project/blob/master/Traffic_Sign_Classifier.ipynb

Dataset

- https://sid.erda.dk/public/archives/daaeac0d7ce1152aea9b61d9f1e19370/published-archive.html
- http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset

Loading Images

- https://www.tensorflow.org/tutorials/load_data/images