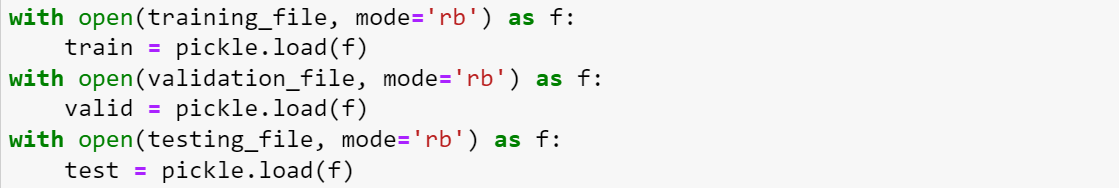
Report 4 - Build a Traffic Sign Recognition Classifier

Jack 1930026143

1. Load the data

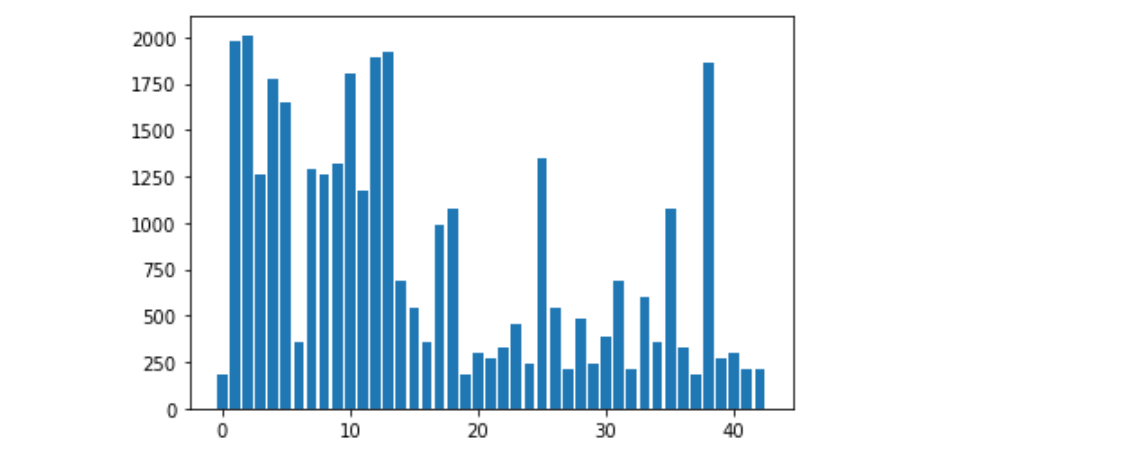
The data is which stores objects as files on disk and we can get the training set, test set and validation set. If we want to load this type of file, we should use the pickle package.



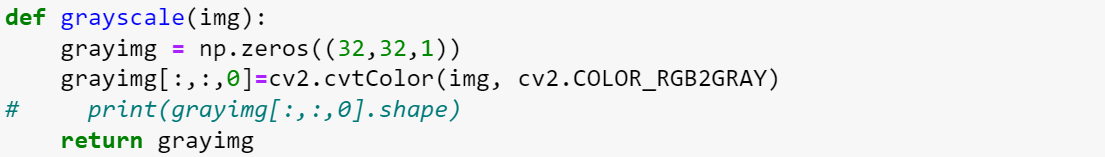
1. Dataset summary & Exploration
2. For all the dataset, they have two attributions: features and labels. 'features' is a 4 dimensions array containing raw pixel data of the traffic sign images, sample number width, height, and RGB channels).

'labels' is a 1D array containing the label/class id of the traffic sign.

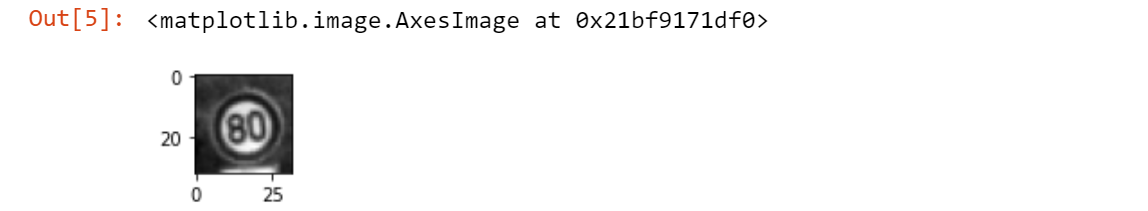
1. For training set, it has 34799 samples/ images, and the images shape is . For test set, it has 12630 images and validation examples is 4410.
2. Then we can check the distribution of sample class in training set:



1. Implement and Test a Model Architecture
2. Pre-processing: normalize the training data and change them to the grey images.



The original data is colorful images with RGB (3) channels, then we use a OpenCV library to convert RGB color images into gray scale images, which contain only one color channel.



1. Set the TensorFlow vision and hyper-parameter for the model, we use the vision of TensorFlow is , the epochs are 10 and batch size is 128.
2. Build Model: According to the paper, we can know the architecture of this model, it has 2 convolution layers, 1 pooling layer and 3 fully connection layers.

Layer 1: Convolutional layer. Input Output

Then the output should connect to a max pooling layer whose kernel’s : batch size is 1 and channel is 1.

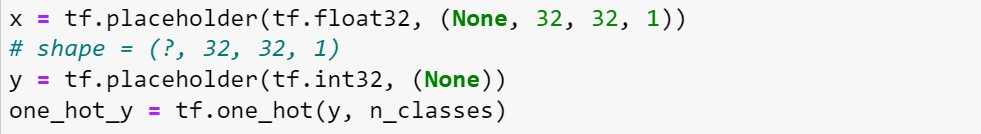
Layer2: Convolutional layer which is same as the above one. Then just apply an activation function and output the result for this layer.

Layer3: After flattening unfolds in one dimension , the vector entered into the fully connected layer, which does a linear transformation. before applying an activation function and output the result for this layer.

Layer4: Fully connected layer. I.

Layer5: Lastly, fully connected layer. I since it id the number of classes, which does a linear transformation.

1. Sets the vector form of model inputs and outputs.



is a placeholder which the first dimension will update each time, y is one-hot code used into the classification.

1. Create the optimizer and assign the loss function. For loss function, we use the cross entropy: The first step is to do a SoftMax on the output of the last layer of the network to get the probability of falling into each category, and then do a cross entropy between the output vector and the actual label of the sample. Then apply the loss function to the and set the learning rate to 0.01.
2. Model Evaluation
3. Train the model by training set, we can set the number of epochs be 10 or even more, but not too small or too large since it may lead to underfitting or overfitting.

We use the statements to control the execution of, and output files. Here we train 80 epochs and the accuracy 92.8%.

1. Test a Model on the New Images

Load the images and test from a directory and check the labels of the five images:



Eventually, the label the model predicts is which can search from the to find the traffic meaning. However, 2 images are error

1. , it will return the values and indices (class ids) of the top 5 predictions:

* What is the accuracy of your model?

The accuracy of the model is around 93% but may improve to higher.

* How to increase the accuracy of your model?

1. adjust the epochs number and the learning rate.
2. Add the dropout layer which means that we can discard a certain percentage of parameter updates to avoid overfitting.
3. Adjust the model structure just like add a more convolution layer, change the size of the kernel or the dimension of the hidden. We can use the grid search method to get the best choice.

* Can you use the model to classify other objects as well? What is the workflow?

Yes. We can use this model to sentiment analysis of sentences which is essentially a classification problem. First, we can use a vocab table to embed the sentence tokens. Then we get the vector dictionary. Then we can input the layer into the model to classify which sentiment it belongs to. Then other operation just like the above task.

* Can you use other models to classify the traffic signs? How to use new models?

Yes, we can use the (convolution neural network) model to finish the task, it is very suitable to deal with the images data and it is very similar to the model.

We can find that it has two convolution-pooling layers and some full connection layers.

After training the model we can get the possibility of each label, which means that we can finish the images classification.