Pablo Sauras Perez Reflection.

Term 1 – Project 2 – Traffic Sign Recognition

Data Set Summary & Exploration

1. Provide a basic summary of the data set

I used numpy and pandas to create the dataset summary:

Training Set Size	34799
Validation Set Size	4410
Test Set Size	12630
Shape of Traffic Sign Image	(32, 32, 3)
Number of unique classes	43

2. Include an exploratory visualization of the dataset.

The **first part of the dataset visualization** consists on the visualization of one random traffic sign from the training set, as it is shown in the following figure.

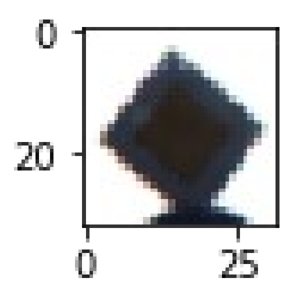

Figure 1: Traffic sign from training set. Priority road

The **second part of the dataset visualization** consists on the visualization of the number of classes in the datasets. Similar to providing histograms. In this way, I can see that some classes in the training set are under-represented. Based on this assessment, I can add more classes to the training set. In the following way:

→ I determine the class that is most represented, and then the ratio of the other classes with respect to this one. Classes with ratio less than 0.5 will be additionally added to the training set. The number of times that I will add those classes is determined based on how many times I have to add the classes to reach a ratio of 0.5.

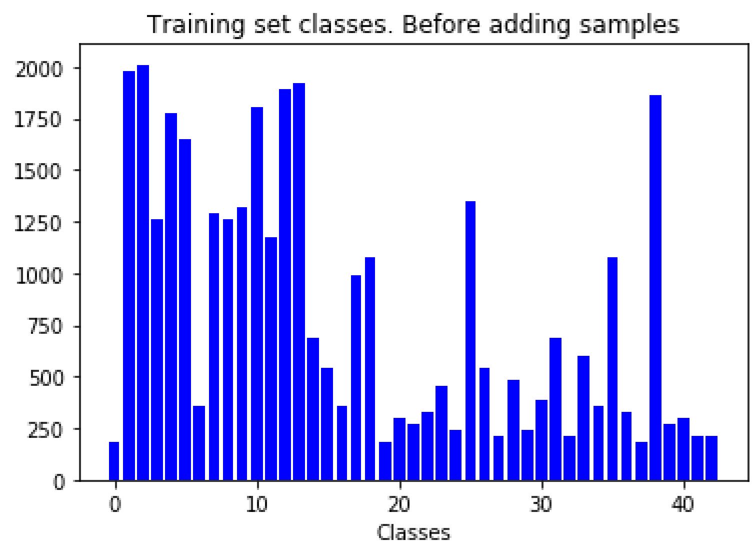

Figure 2: Distribution of training set before adding more classes

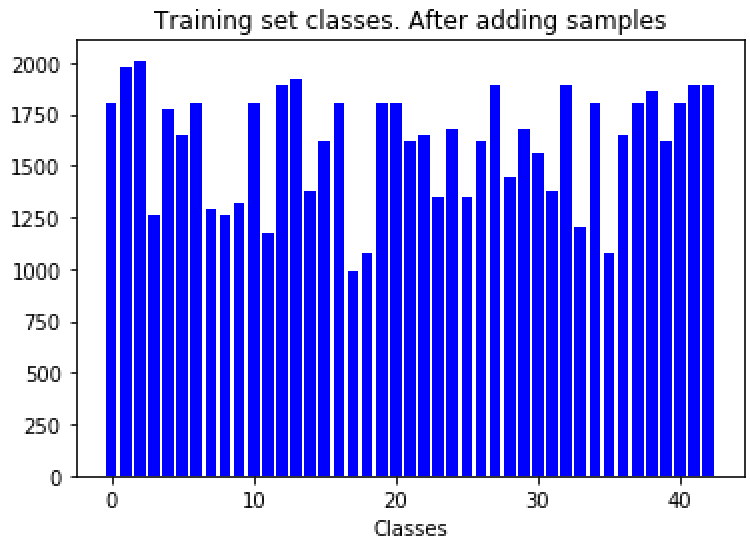

Figure 3: Distribution of the training set after adding more classes.

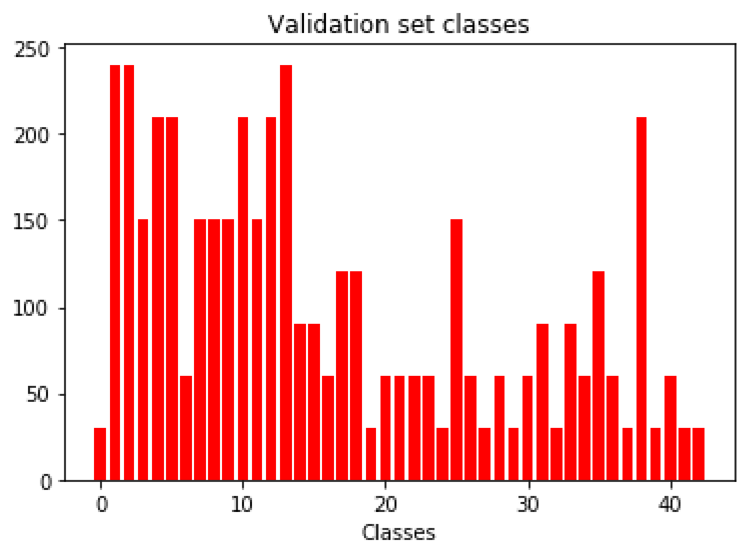

Figure 4: Validation set classes distribution

Figure 5: Test set classes distribution

Design and Test a Model Architecture

1. Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques? Consider including images showing the output of each preprocessing technique. Pre-processing refers to techniques such as converting to grayscale, normalization, etc.

For all the sets I converted the images to **grayscale and normalized** them. In this way, the images are within the same intensity values range.

2. Describe what your final model architecture looks like including model type, layers, layer sizes, connectivity, etc.) Consider including a diagram and/or table describing the final model.

My model is based on the **LeNet** model seen in Lesson 8 “Convolutional Neural Networks”

My final model consisted in the following layers

	Layer	Input	Stride	Padding	Output
1	Convolution 5x5	32x32x1 Gray-scale Image	1x1	VALID	28x28x6
	ReLu				
	MaxPool	28x28x6	2x2	VALID	14x14x6
2	Convolution 5x5	14x14x6	1x1	VALID	10x10x16
	ReLu				
	MaxPool	10x10x16	2x2	VALID	5x5x16
	Flatten	5x5x16			400
3	Fully connected	400			120
	ReLu				
4	Fully Connected	120			84
	ReLu				
5	Fully Connected	84			43

3. Describe how you trained your model. The discussion can include the type of optimizer, the batch size, number of epochs and any hyperparameters such as learning rate.

To train the model I used:

Optimizer	Adam Optimizer
Learning Rate	0.001
Epochs	55
Batch Size	128

4 . Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93. Include in the discussion the results on the training, validation and test sets and where in the code these were calculated.

My final model results were:

Training Set Accuracy	1.00
Validation Set Accuracy	0.95
Test Set Accuracy	0.93

As mentioned before, the **LeNet** Architecture was chosen. As **LeNet** performs well in character recognition, it may classify as well traffic signs, as the most relevant features of them can be treated as "characters".

After training the model the following loss and validation pots are obtained:

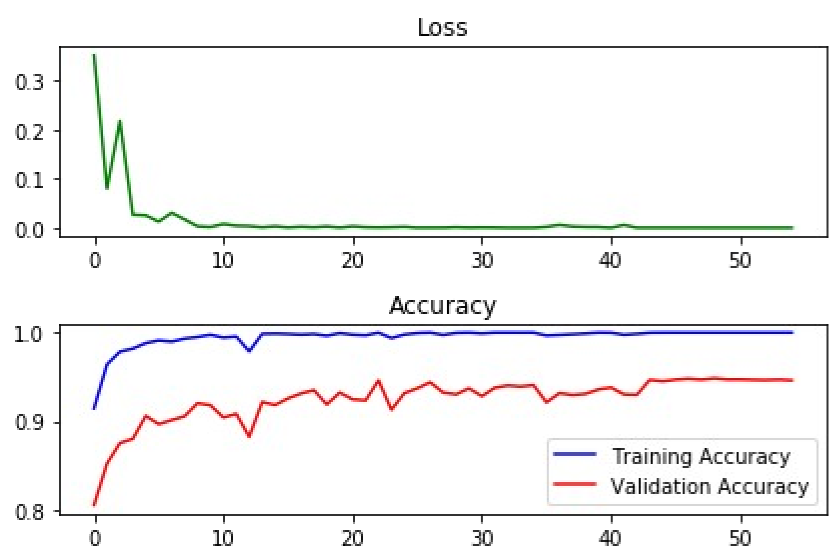

Figure 6: Loss and Accuracy plots

As it can be seen, the model loss decays over the epochs. Around **epoch 45** the loss is constant. Thus, the model "stops" learning. I decided to keep **55 epochs** to make sure that the loss was constant.

The same happens to the accuracy validation. We can see that around **Epoch 45** the validation reaches a constant value of **0.95**, which is a higher validation accuracy than the minimum required.

Once the model was trained, the **accuracy of the test** set was **0.93**.

Test a Model on New Images

1. Choose five German traffic signs found on the web and provide them in the report. For each image, discuss what quality or qualities might be difficult to classify.

Here are five German traffic signs that I found on the web:

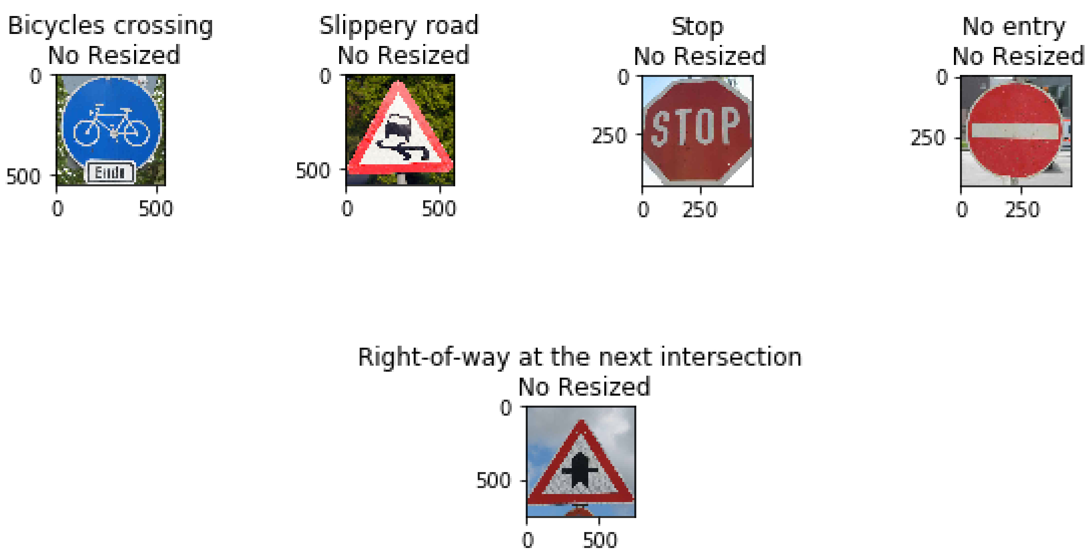

Figure 7: New test set images

All images were cropped so that the ratio was 1:1. Then, the images where resized to 32x32 and normalized in the same way as the previous sets.

As a side note, the first attempt of getting this images was done without cropping them. Then, they were resized so that the size was Ax32. After that, the dimension some padding of 0s was added to ‘A’ so that the image was 32x32. This approach failed to recognize the images (Accuracy 0.000). Thus, my conclusion is that this LeNet model will not work well in a real environment, where the traffic signs are not necessarily in the center of the image an the image ration is not 1:1.

Seeing the new images, I predict that “Bicycle Crossing” and “Right of way at the next intersection” may have challenges being recognized, as they have some elements at the bottom of the image.

2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set. At a minimum, discuss what the predictions were, the accuracy on these new predictions, and compare the accuracy to the accuracy on the test set

Here are the results of the prediction:

Image	Probability	Prediction
Stop	0.85	Turn right ahead
Right of way at the next intersection	1	Right of way at the next intersection
No entry	1	No entry
Slippery Road	0.99	Right of way at the next intersection
Bicycles Crossing	0.97	Roundabout mandatory

The model was able to correctly guess 2 of the 5 traffic signs, which gives an **accuracy of 40%,** which is lower than the 93% that wqas achieved previously.

3. Describe how certain the model is when predicting on each of the five new images by looking at the softmax probabilities for each prediction. Provide the top 5 softmax probabilities for each image along with the sign type of each probability.

As a side note, before testing the classification of the new images, I shuffled the sets.

For the Image **Stop,** the prediction **fails** and it is relatively certain of the prediction**.** I obtained the following top 5 softmax probabilities:

Probability	Prediction
0.85	Turn right ahead
0.13	Ahead only
0.02	Yield
0.00	Speed limit (30km/h)
0.00	No vehicles

For the Image **Right-of-way at the next intersection,** the prediction is **correct** and it is very certain of the prediction**.** I obtained the following top 5 softmax probabilities:

Probability	Prediction
1	Right-of-way at the next intersection
0	Beware of ice/snow
0	Speed limit (20km/h)
0	Speed limit (30km/h)
0	Speed limit (50km/h)

For the Image **No entry,** the prediction is **correct** and it is very certain of the prediction**.** I obtained the following top 5 softmax probabilities:

Probability	Prediction
1	No entry
0	Traffic signals
0	Roundabout mandatory
0	Beware of ice/snow
0	Speed limit (20km/h)

For the Image **Slippery road,** the prediction **fails** and it is very certain of the prediction**. I** obtained the following top 5 softmax probabilities:

Probability	**Prediction**
0.99	Right-of-way at the next intersection
0.00...	Double curve
0.00..	Priority road
0.00..	Beware of ice/snow
0.00..	Speed limit (80km/h)

For the Image **Bicycles crossing,** the prediction **fails** and it is very certain of the prediction**. I** obtained the following top 5 softmax probabilities:

Probability	**Prediction**
0.98	Roundabout mandatory
0.01	General caution
0.00..	Speed limit (30km/h)
0.00..	Turn right ahead
0.00..	Traffic signals