# Traffic Sign Recognition Report

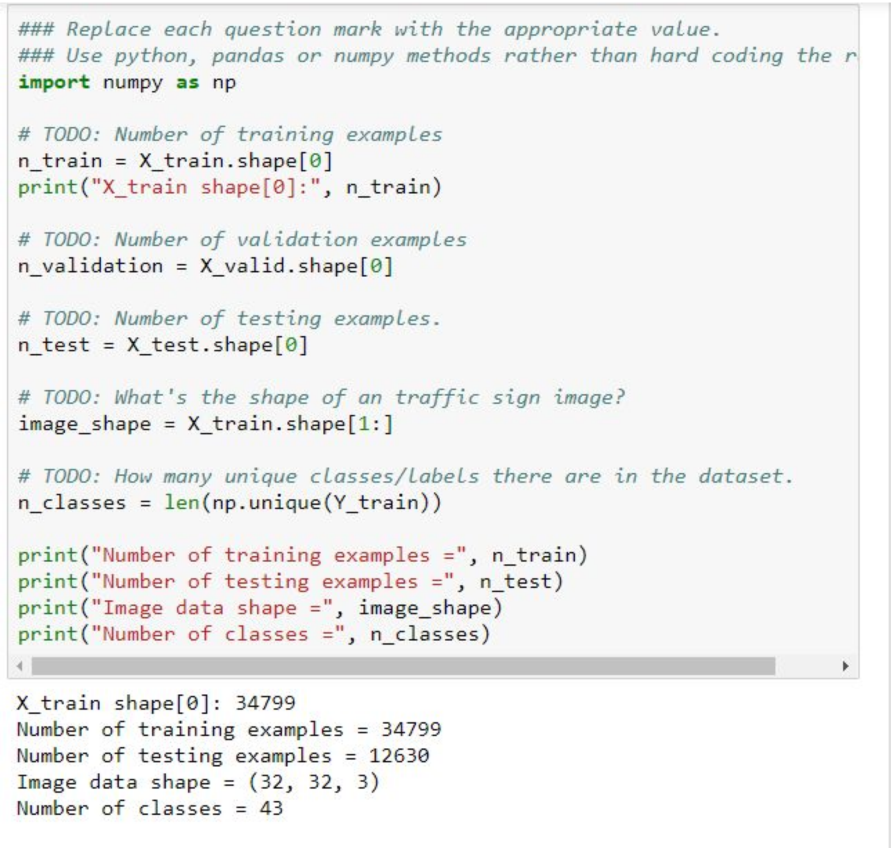
## Build a Traffic Sign Recognition Project

The goals / steps of this project has been fulfilled as requested. Data set has been loaded and visualised. Convolutional neural network has been trained, validated and tested where 96.3 % accuracy has been achieved on validation level. Architecture has been tested on new images where 80 % accuracy was achieved.

## Data Set Summary & Exploration

### 1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

I used the numpy library to calculate summary statistics of the traffic signs data set:



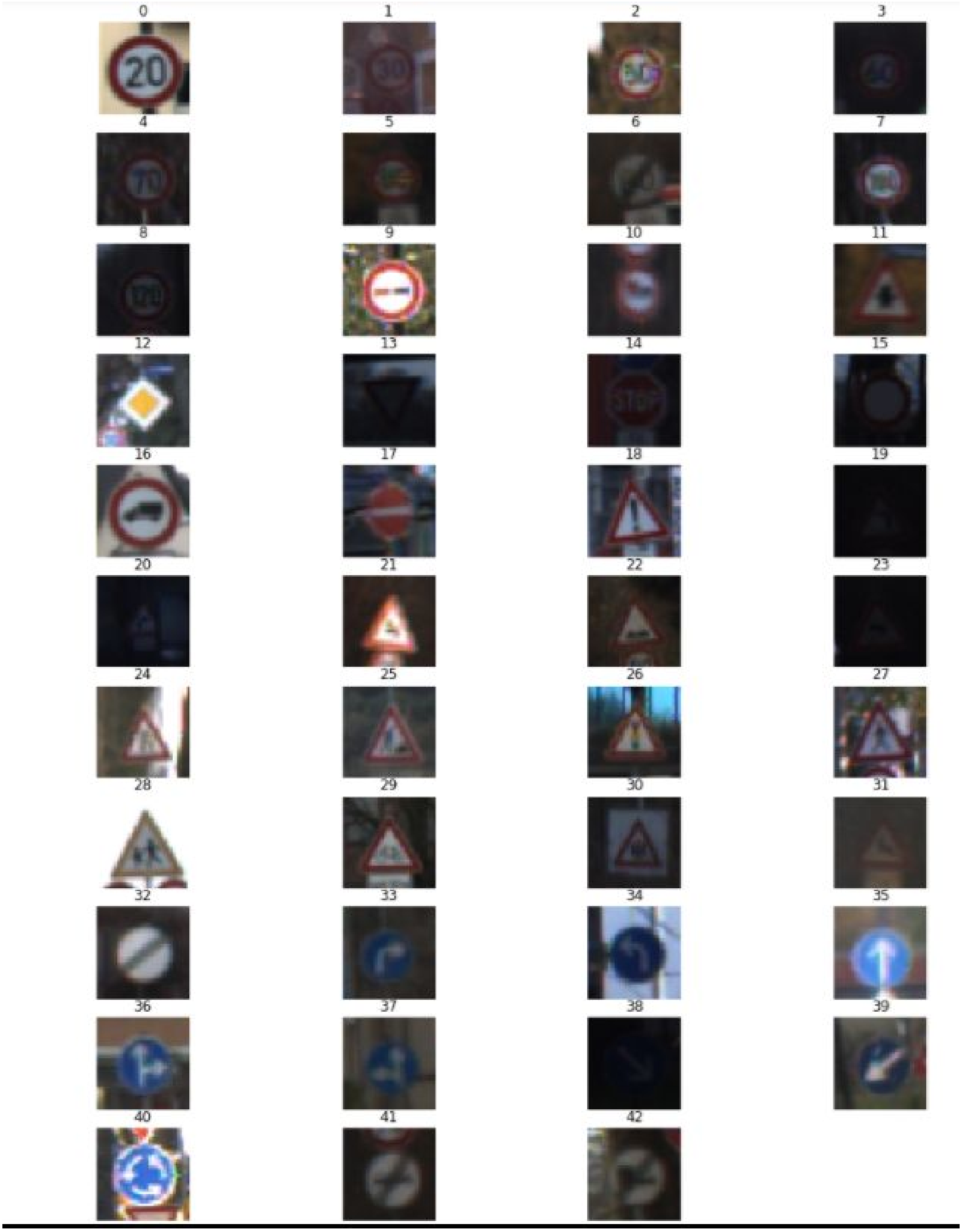
```
### Replace each question mark with the appropriate value.
### Use python, pandas or numpy methods rather than hard coding the r
import numpy as np

# TODO: Number of training examples
n_train = X_train.shape[0]
print("X_train shape[0]:", n_train)

# TODO: Number of validation examples
n_validation = X_valid.shape[0]

# TODO: Number of testing examples.
n_test = X_test.shape[0]

# TODO: What's the shape of an traffic sign image?
image_shape = X_train.shape[1:]

# TODO: How many unique classes/labels there are in the dataset.
n_classes = len(np.unique(Y_train))

print("Number of training examples =", n_train)
print("Number of testing examples =", n_test)
print("Image data shape =", image_shape)
print("Number of classes =", n_classes)
```

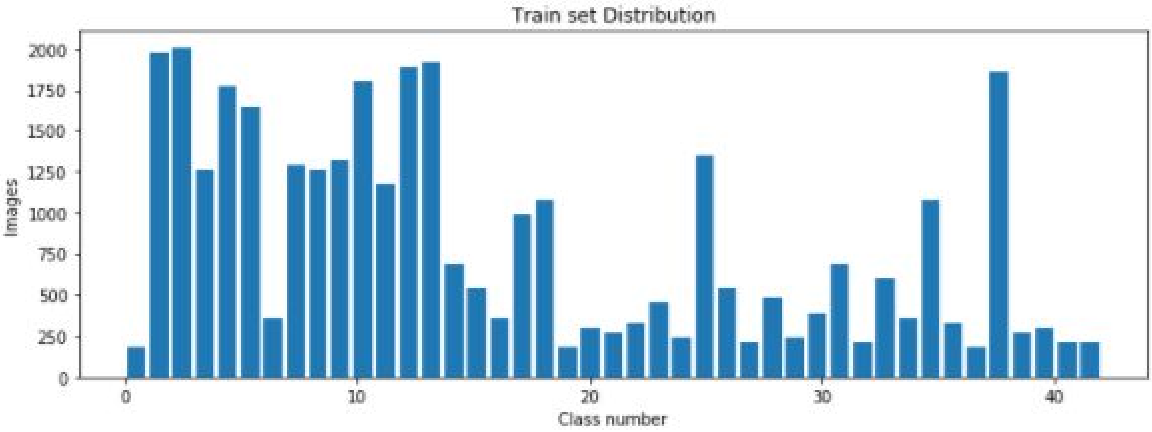
```
X_train shape[0]: 34799
Number of training examples = 34799
Number of testing examples = 12630
Image data shape = (32, 32, 3)
Number of classes = 43
```

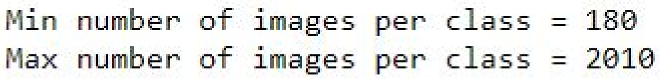
### 2. Include an exploratory visualization of the dataset.

I have used matplotlib in order to visualise the dataset.

Also I have plotted a summary of the distribution of images per class within the training data with its maximum and minimums.







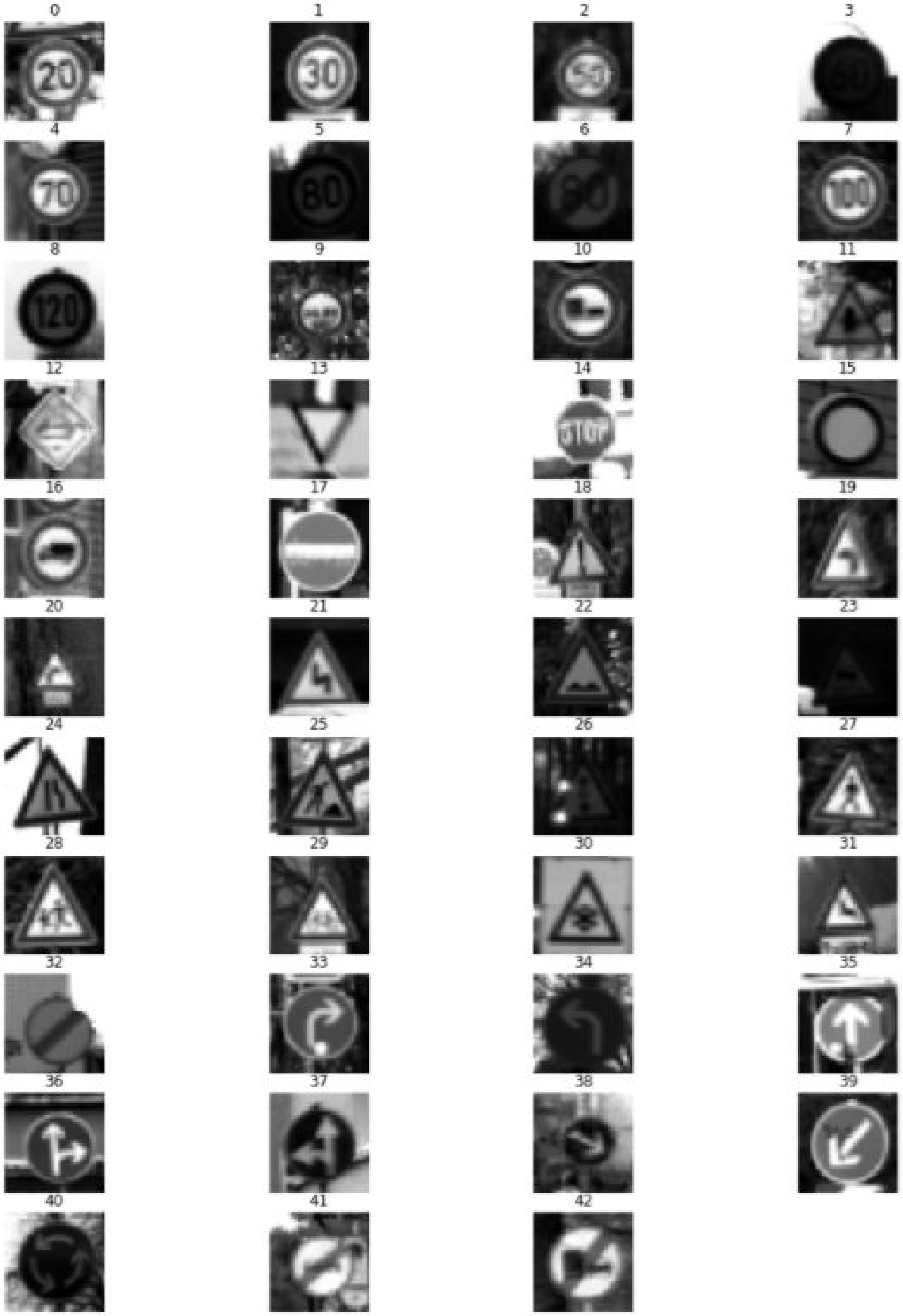
```
Min number of images per class = 180
Max number of images per class = 2010
```

## 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. (OPTIONAL: As described in the "Stand Out Suggestions" part of the rubric, if you generated additional data for training, describe why you decided to generate additional data, how you generated the data, and provide example images of the additional data. Then describe the characteristics of the augmented training set like number of images in the set, number of images for each class, etc.)

In my preprocessing section, I have performed two steps and more specifically transformation to grayscale and normalisation of the images.I have used the grayscale conversion from RGB as it reduces the computation effort and it is increasing the classification accuracy as stated here. The normalisation function is expanding the dynamic range of the images which also increases the classification accuracy.

Provided below is the pre-processing effect on the same training images as above.



**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 final architecture has this format:

| Layer | Description |
|---|---|
| Input | 32x32x1 grayscale normalised image |
| Convolution 5x5 | In 32x32x1, out 28x28x6, 1x1 stride and Valid padding |
| RELU | |
| Max Pooling 2x2 | In 28x28x6, out 14x14x6, 2x2 stride and Valid padding |
| Convolution 5x5 | In 14x14x6, out 10x10x16,1x1 stride and Valid padding |
| RELU | |
| Max Pooling 2x2 | In 10x10x16, out 5x5x16, 2x2 stride and Valid padding |
| Flatten | In = 5x5x16. Out = 400 |
| Fully Connected | In 400, Out 120 |
| RELU | |
| Dropout | |
| Fully Connected | In 120, Out 84 |
| RELU | |
| Fully Connected | In 84, Out 43 |

Using this architecture, I was able to achieve approximately 96 % validation accuracy. This CNN architecture is potentially the  one created by the LeNet Lab.

**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.**

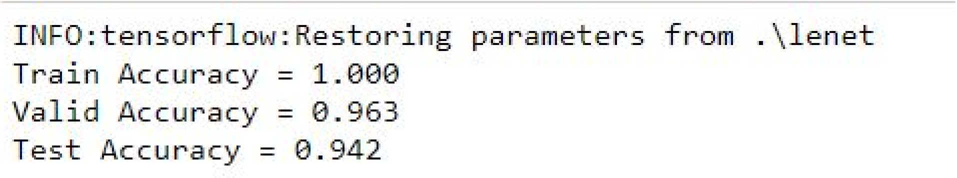
I used the Adam optimiser methodology.

The additional parameters are:

- Epochs = 60
- Batch Size = 128
- Learning rate = 0.0008
- Sigma = 0.1
- Mu = 0

**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. Your approach may have been an iterative process, in which case, outline the steps you took to get to the final solution and why you chose those steps. Perhaps your solution involved an already well known implementation or architecture. In this case, discuss why you think the architecture is suitable for the current problem.**

My final model results were:



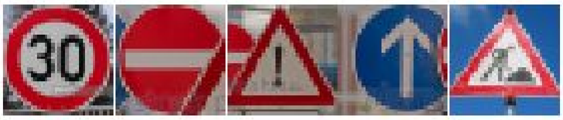
```
INFO:tensorflow:Restoring parameters from .\lenet
Train Accuracy = 1.000
Valid Accuracy = 0.963
Test Accuracy = 0.942
```

I have chosen to use the LeNet Lab architecture. I have chosen to use it as a starting point. The performance that it provided me with the preprocessing functions, was above the expected threshold.  I have tuned the learning rate and the numbers of epochs. I have also tested with different Batch sizes.

## 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.**

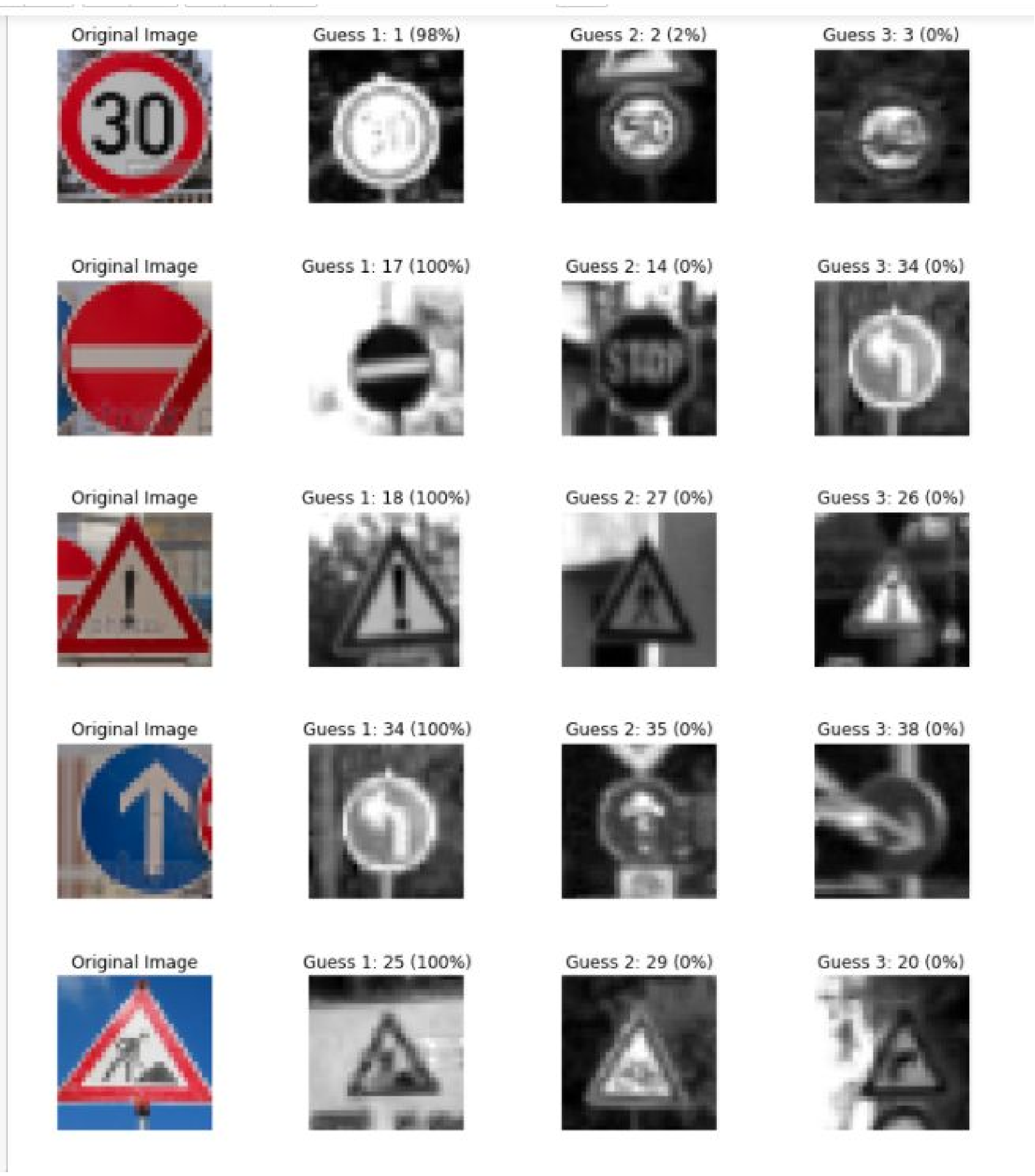
Here are five German traffic signs that I found on the web:



I believe the 2,3 and 4th image are bit confusing as there are multiple signs overlaying. I have reached 80 % accuracy and the only wrongly classified sign is number 4 the ahead only sign.

**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 (OPTIONAL: Discuss the results in more detail as described in the "Stand Out Suggestions" part of the rubric).**

Here are the results of the prediction for all the images:


Original Image
Guess 1: 1 (98%)
Guess 2: 2 (2%)
Guess 3: 3 (0%)
Original Image
Guess 1: 17 (100%)
Guess 2: 14 (0%)
Guess 3: 34 (0%)
Original Image
Guess 1: 18 (100%)
Guess 2: 27 (0%)
Guess 3: 26 (0%)
Original Image
Guess 1: 34 (100%)
Guess 2: 35 (0%)
Guess 3: 38 (0%)
Original Image
Guess 1: 25 (100%)
Guess 2: 29 (0%)
Guess 3: 20 (0%)