Traffic Sign Recognition

Writeup Report

By Liang Hu

The goals / steps of this project are the following:

- Load the data set (see below for links to the project data set)
- Explore, summarize and visualize the data set
- Design, train and test a model architecture
- Use the model to make predictions on new images
- Analyze the softmax probabilities of the new images
- Summarize the results with a written report

Here is a link to my project code.

Data Set Summary & Exploration

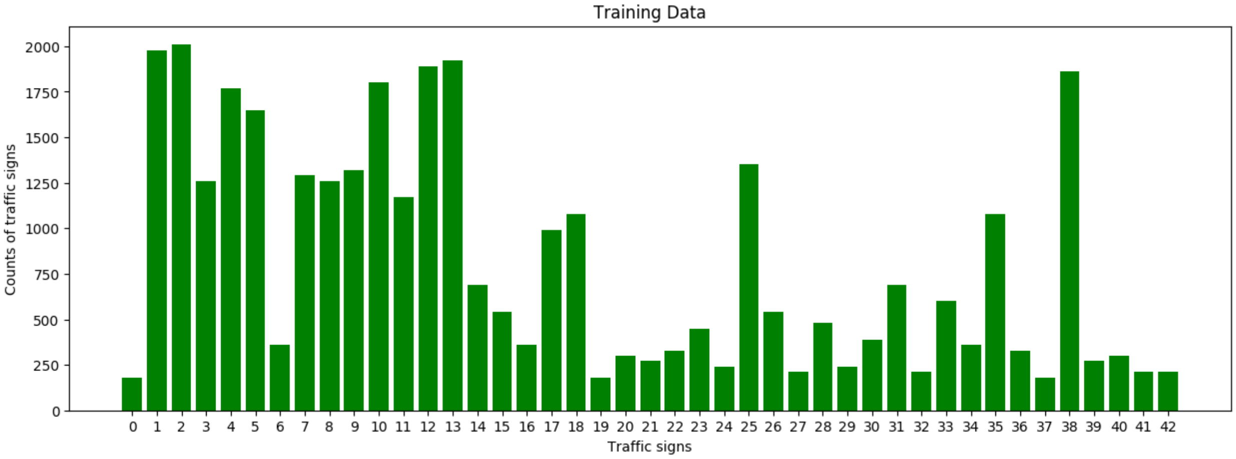
1. Basic summary of the data set.

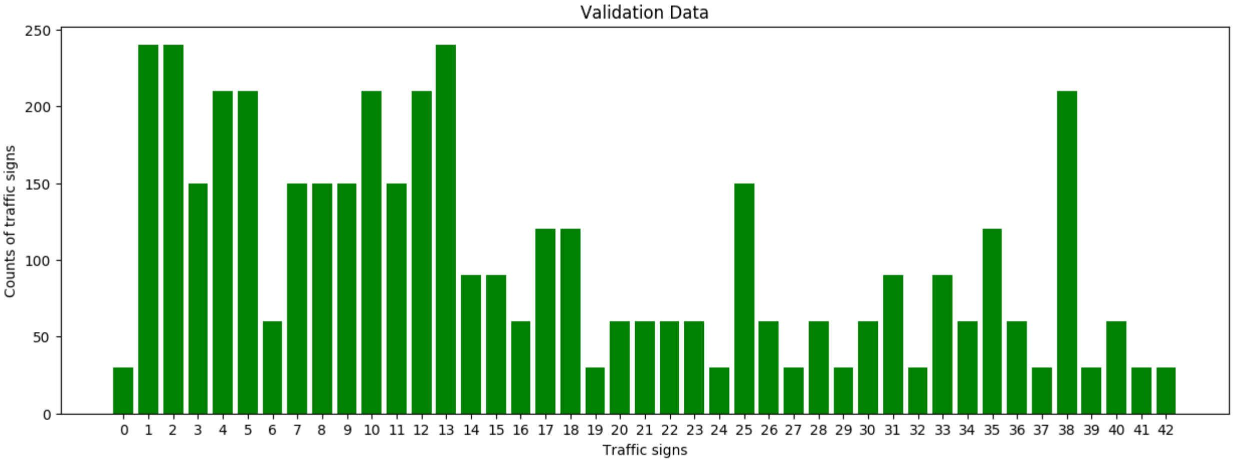
I used the numpy library to calculate summary statistics of the traffic signs data set:

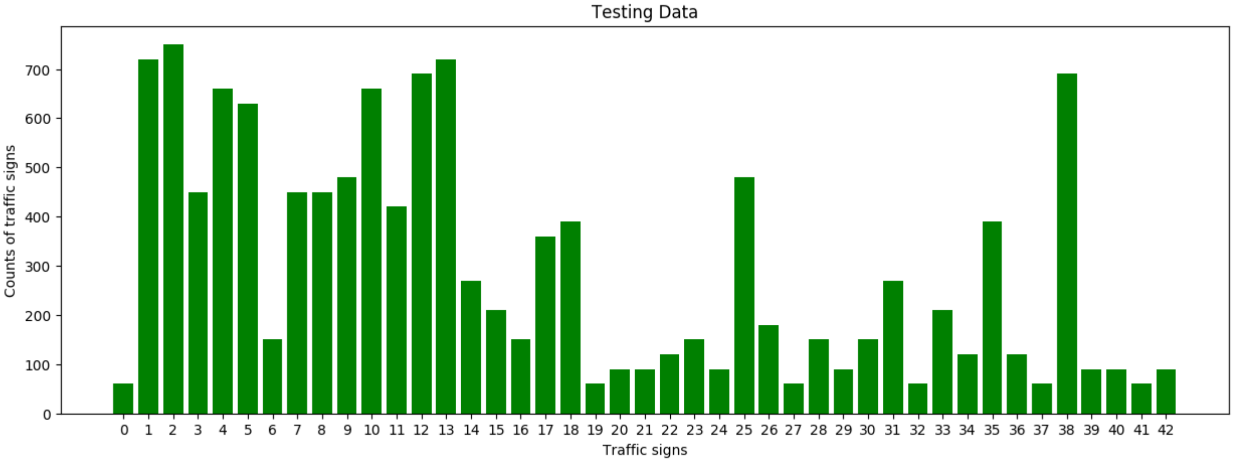
- The size of training set is 34799.
- The size of the validation set is 4410.
- The size of test set is 12630.
- The shape of a traffic sign image is (32, 32, 3)
- The number of unique classes/labels in the data set is 43.

2. An exploratory visualization of the dataset.

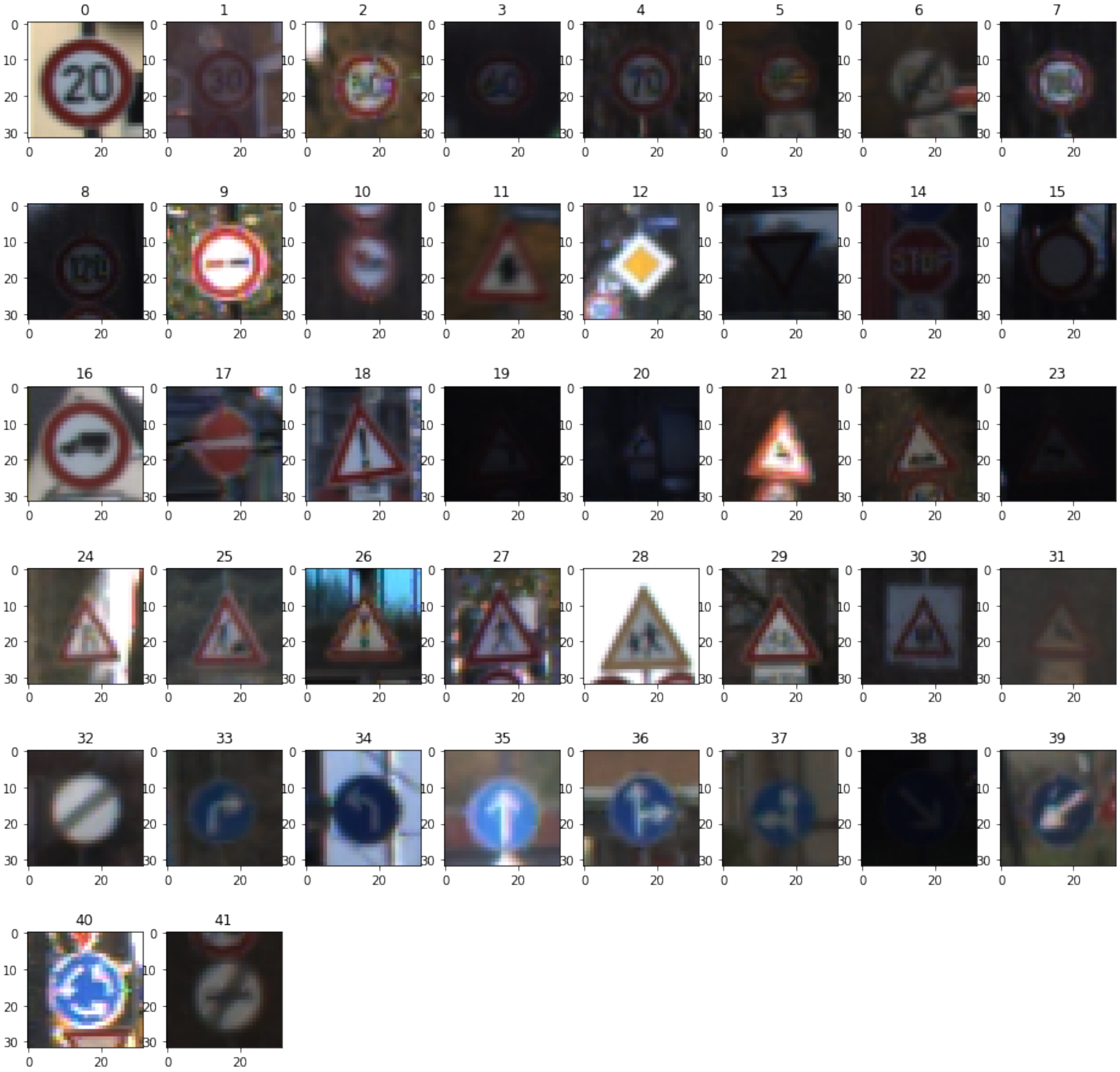
Here is an exploratory visualization of the data set. I plotted 3 bar charts showing how the training data, validation data, and testing data distribute by the traffic sign class, respectively.


Training Data
Counts of traffic signs
0
250
500
750
1000
1250
1500
1750
2000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
Traffic signs


Validation Data
Counts of traffic signs
0
50
100
150
200
250
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
Traffic signs


Testing Data
Counts of traffic signs
0
100
200
300
400
500
600
700
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
Traffic signs

I also showed an image for each unique traffic sign, to whether what they look like. Image title is the traffic sign class id.

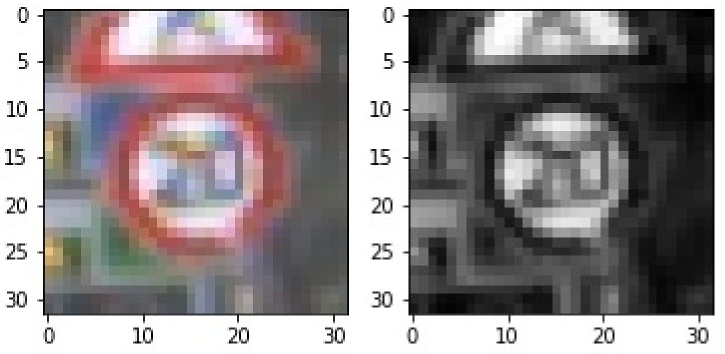


Design and Test a Model Architecture

1. Ways to preprocess the image data

First, I decided to convert the images to grayscale because image shape is the main factor to determine a traffic sign rather than color. Also, reducing image channel from 3 to 1 will make the neural network be trained more quickly.

Here is an example of a traffic sign image before and after grayscaling.



Second, I normalized the image data because normalization will also acceleration the model training.

2. Final model architecture

My final model consisted of the following layers:

Layer	Description
Input	32x32x1 grayscale image
Convolution 5x5	1x1 stride, valid padding, outputs 28x28x6
RELU	
Max pooling	2x2 stride, outputs 14x14x6
Convolution 5x5	1x1 stride, valid padding, outputs 10x10x16
RELU	
Max pooling	2x2 stride, outputs 5x5x16
Flatten	outputs 400
Fully connected	input 400 output 120
Fully connected	input 120 output 84
Fully connected	input 84 output 43
Softmax	

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 the AdamOptimizer() to minimize the cross entropy. The learning rate is set as 0.0009. The batch size is 128. I used epochs of 60 to train the model.

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.

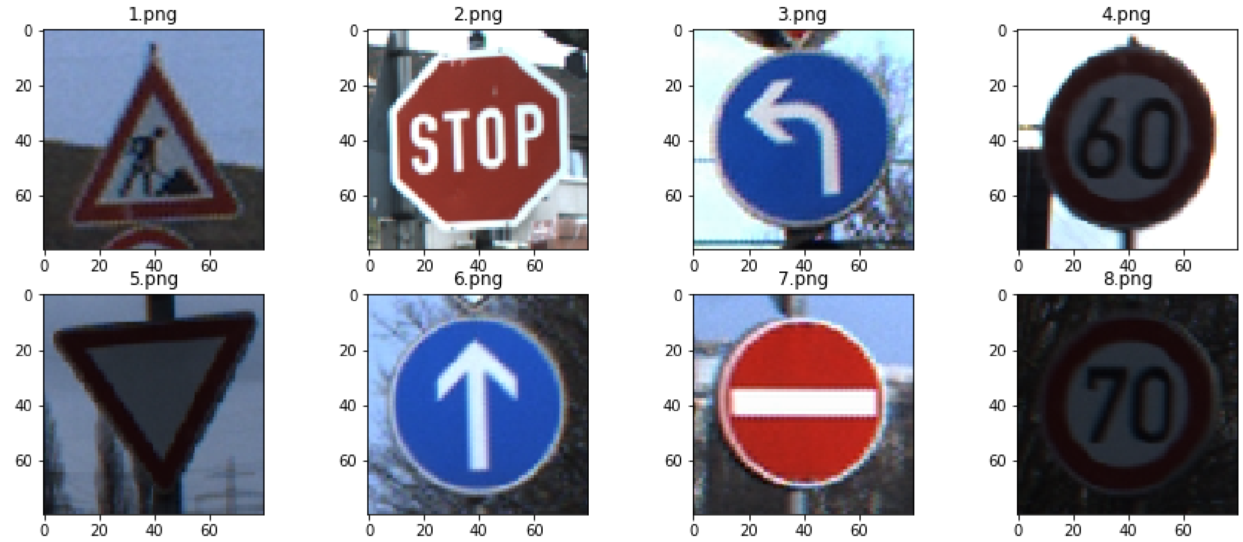
The training process is in code cell [14]. My final model results were:

- training set accuracy of 1.000 (code cell [17])
- validation set accuracy of 0.942 (code cell [14])
- test set accuracy of 0.925 (code cell [15])

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 8 German traffic signs that I found on the web:



The last image might be difficult to classify because it is relatively dark.

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:

Image	Prediction
25 Road work	25 Road work
14 Stop	14 Stop
34 Turn left ahead	34 Turn left ahead
3 Speed limit (60km/h)	3 Speed limit (60km/h)
13 Yield	13 Yield
35 Ahead only	35 Ahead only
17 No entry	17 No entry
4 Speed limit (70km/h)	4 Speed limit (70km/h)

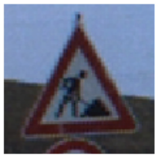
The model was able to correctly guess 8 of the 8 traffic signs, which gives an accuracy of 100%. This compares favorably to the accuracy on the test set of 0.925.

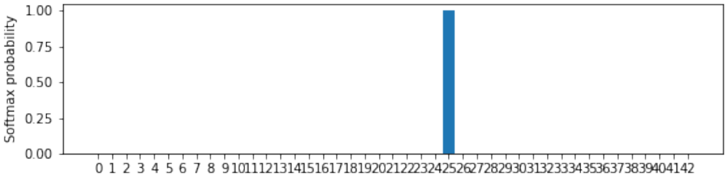
3. Describe how certain the model is when predicting on each of the five new images by looking at the softmax probabilities for each prediction. Provide the top 5 softmax probabilities for each image along with the sign type of each probability. (OPTIONAL: as described in the "Stand Out Suggestions" part of the rubric, visualizations can also be provided such as bar charts)

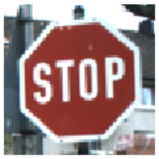
The code for making predictions on my final model is located in the 23th cell of the Ipython notebook.

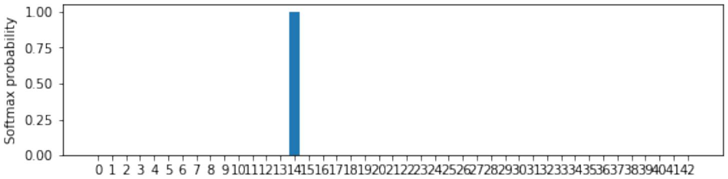
The 8 test traffic signs were predicted really well, as seen the below table and figure.

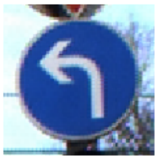
Probability	Prediction
1.0	25 Road work
1.0	14 Stop
1.0	34 Turn left ahead
1.0	3 Speed limit (60km/h)
1.0	13 Yield
1.0	35 Ahead only
1.0	17 No entry
0.9999	4 Speed limit (70km/h)

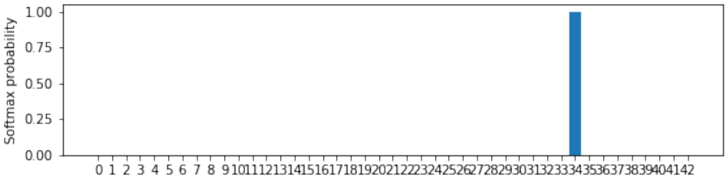


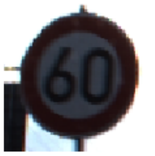

Softmax probability
1.00
0.75
0.50
0.25
0.00
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42

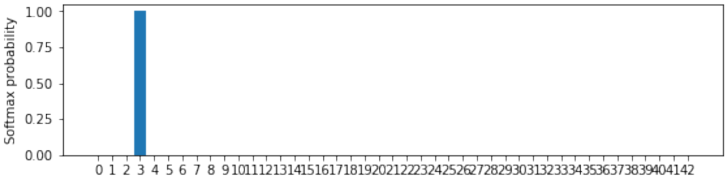

STOP

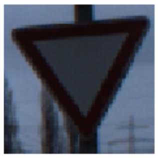

Softmax probability
1.00
0.75
0.50
0.25
0.00
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42

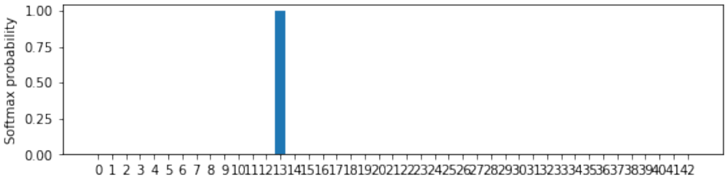


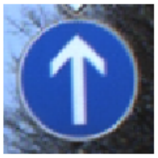

Softmax probability
1.00
0.75
0.50
0.25
0.00
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42

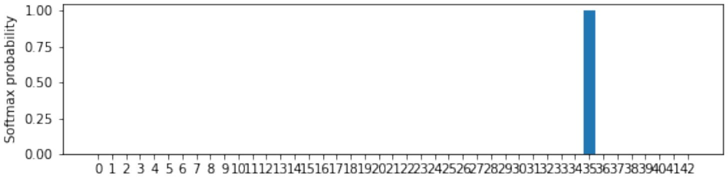

60

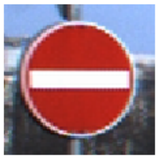

Softmax probability
1.00
0.75
0.50
0.25
0.00
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42

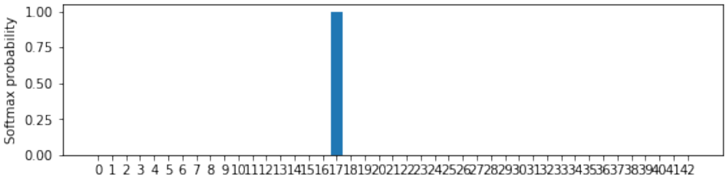


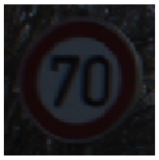

Softmax probability
1.00
0.75
0.50
0.25
0.00
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42

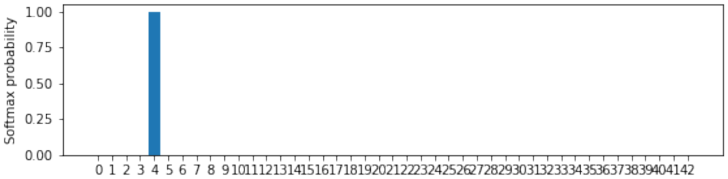



Softmax probability
1.00
0.75
0.50
0.25
0.00
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42




Softmax probability
1.00
0.75
0.50
0.25
0.00
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42


70


Softmax probability
1.00
0.75
0.50
0.25
0.00
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42