CLASSIFYING STREET SIGNS USING CONVOLUTIONAL NEURAL NETWORKS

Paul Woody

CLASSIFYING STREET SIGNS USING CONVOLUTIONAL NEURAL NETWORKS

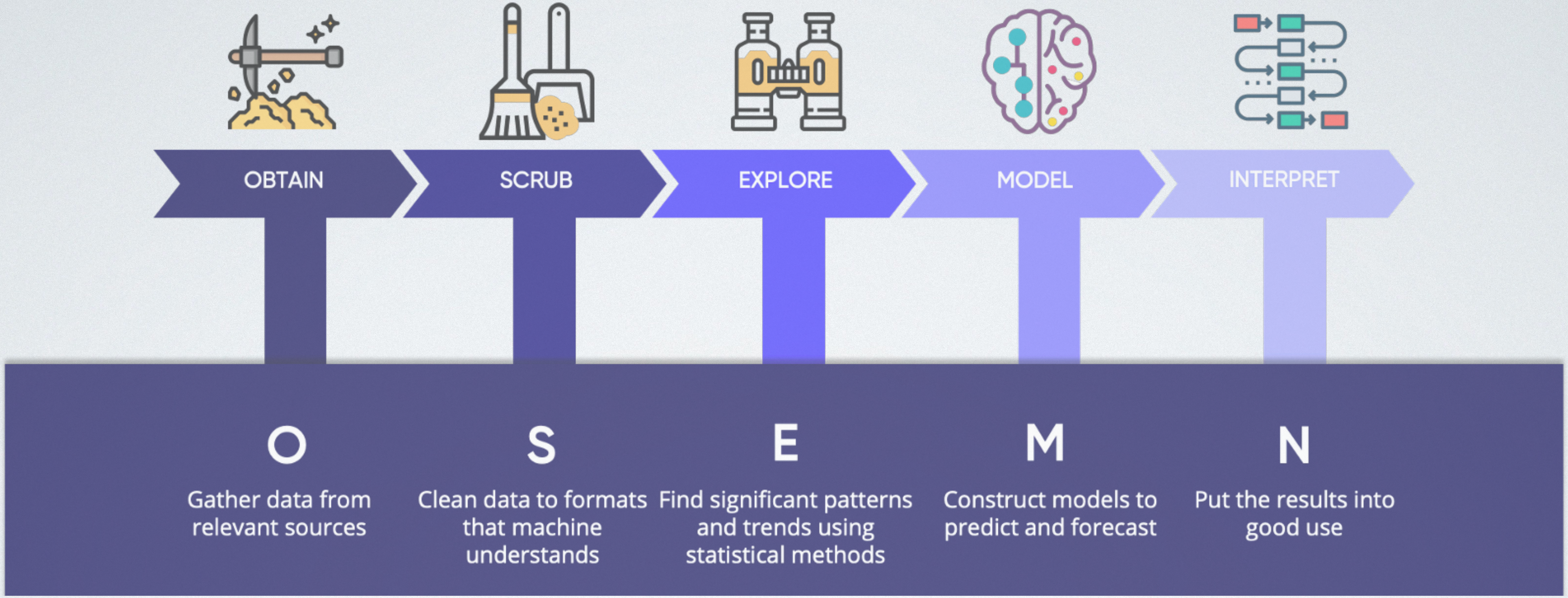
Significance

EDA

Models

Conclusions

METHODOLOGY

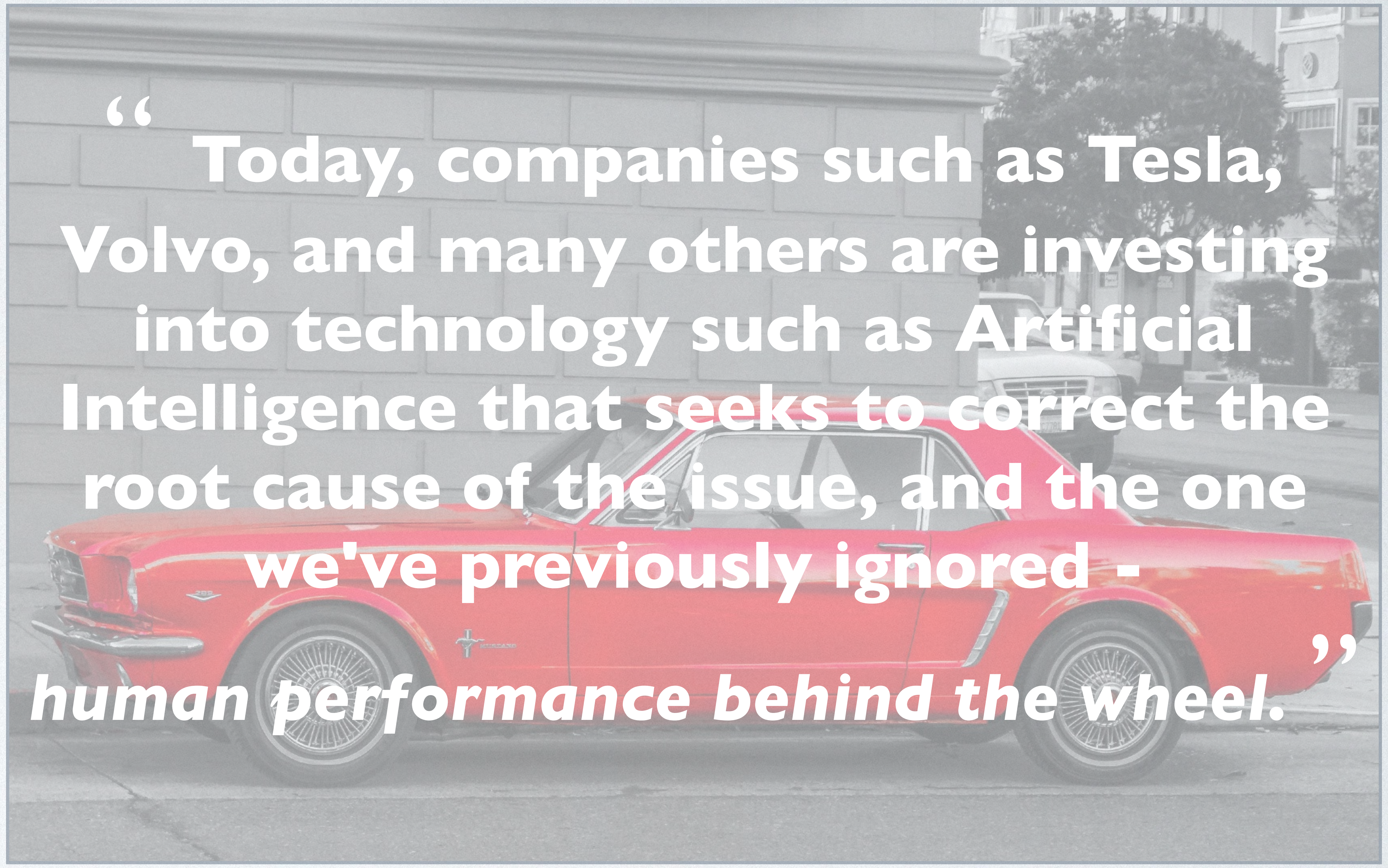


Perform exploratory data analysis to identify meaningful trends.
Use models and hypothesis tests, when appropriate, to evaluate significance.

SIGNIFICANCE

According to the National Safety Council, approximately **40,000 people died** in automotive accidents in the United States alone in 2018. In fact, there were a total of ~500 deaths resulting from plane crashes recorded globally in 2018 - that's **80 times** fewer deaths when compared to car crash fatalities **in the US only**.

While there are many factors that likely contribute to this drastic difference, the difference in mortality remains, in my opinion, pretty staggering.



BACKGROUND

Perhaps the most exciting advancement in automative technology since the invention of the car itself is the birth of the self-driving, or autonomous car. Once the topic of Science Fiction, Convolutional Neural Networks (CNNs) and other modeling techniques have pushed the bounds of "possible" into the realm of human imagination. There are 3 major challenges we must overcome in order to make self driving technology possible. Before we "hand over the keys", we must be able to do the following, very quickly:

Keys to driverless cars

BACKGROUND

Perhaps the most exciting advancement in automative technology since the invention of the car itself is the birth of the self-driving, or autonomous car. Once the topic of Science Fiction, Convolutional Neural Networks (CNNs) and other modeling techniques have pushed the bounds of "possible" into the realm of human imagination. There are 3 major challenges we must overcome in order to make self driving technology possible. Before we “hand over the keys”, we must be able to do the following, very quickly:

1. Obtain the data

sensors, cameras

2. Process the Data

AI - our focus here

3. Act on the Data

drive the car

IMAGE PRE-PROCESSING

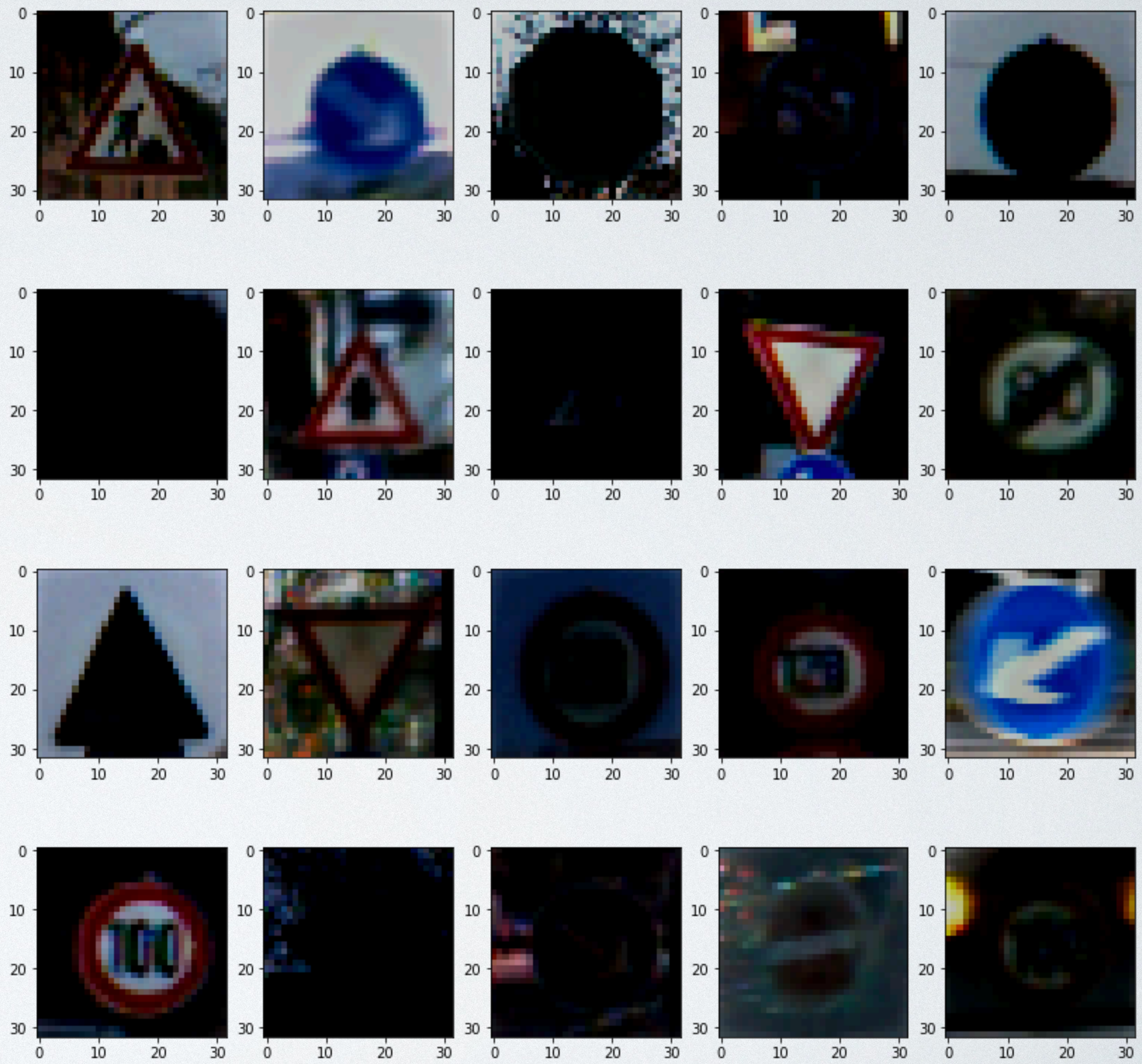
- 100,000 images of traffic signs saved to a pickle file
- Image rotation, standardization, resizing
- Class distribution equalized
- Split into training and test sets

EXPLORATORY DATA ANALYSIS

Dataset Info:

```
Number Training Examples: 86989
Number of Test Samples: 12630
Image Shape: (32, 32, 3)
Number of Target Classes: 43
```

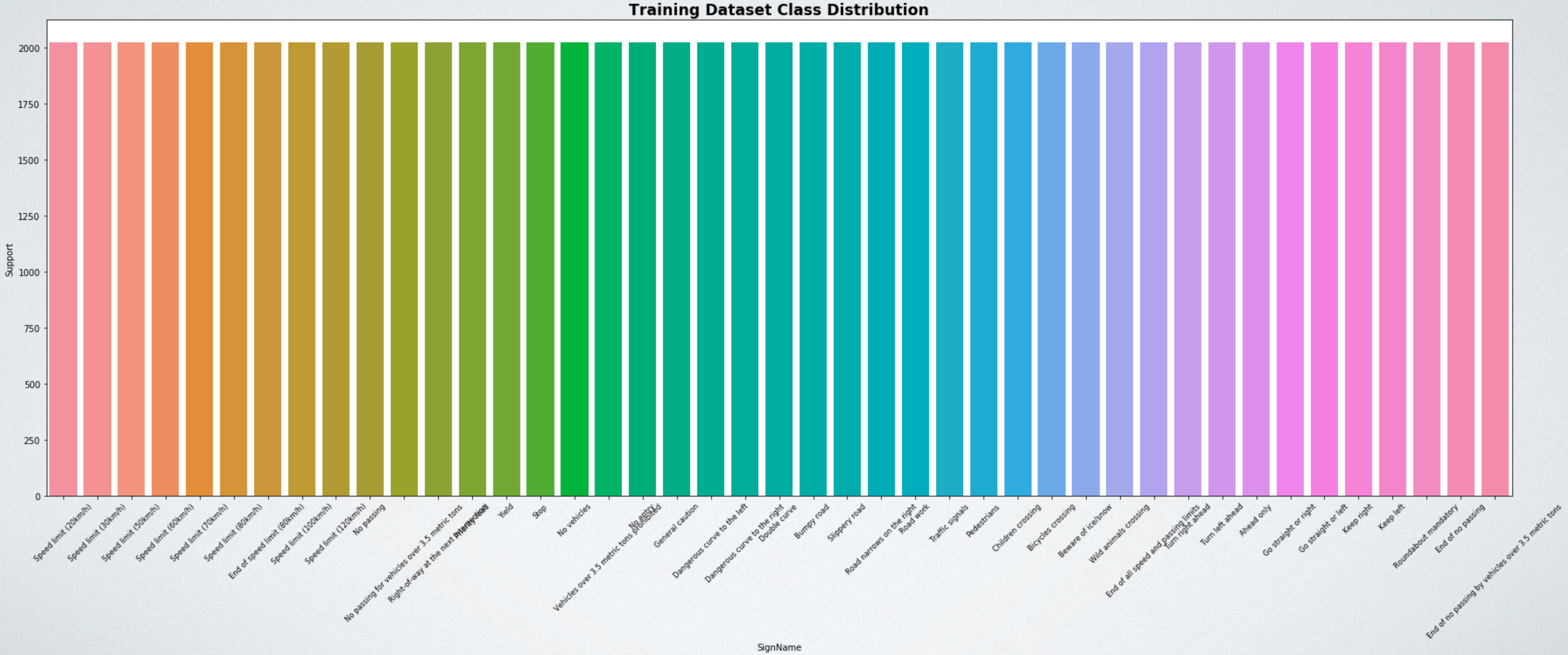
	ClassId	SignName
0	0	Speed limit (20km/h)
1	1	Speed limit (30km/h)
2	2	Speed limit (50km/h)
3	3	Speed limit (60km/h)
4	4	Speed limit (70km/h)
5	5	Speed limit (80km/h)
6	6	End of speed limit (80km/h)
7	7	Speed limit (100km/h)
8	8	Speed limit (120km/h)
9	9	No passing
10	10	No passing for vehicles over 3.5 metric tons
11	11	Right-of-way at the next intersection
12	12	Priority road
13	13	Yield
14	14	Stop
15	15	No vehicles
16	16	Vehicles over 3.5 metric tons prohibited
17	17	No entry
18	18	General caution
19	19	Dangerous curve to the left
20	20	Dangerous curve to the right
21	21	Double curve
22	22	Bumpy road
23	23	Slippery road
24	24	Road narrows on the right
25	25	Road work
26	26	Traffic signals
27	27	Pedestrians
28	28	Children crossing
29	29	Bicycles crossing
30	30	Beware of ice/snow
31	31	Wild animals crossing
32	32	End of all speed and passing limits
33	33	Turn right ahead
34	34	Turn left ahead
35	35	Ahead only
36	36	Go straight or right
37	37	Go straight or left
38	38	Keep right
39	39	Keep left
40	40	Roundabout mandatory
41	41	End of no passing
42	42	End of no passing by vehicles over 3.5 metric ...



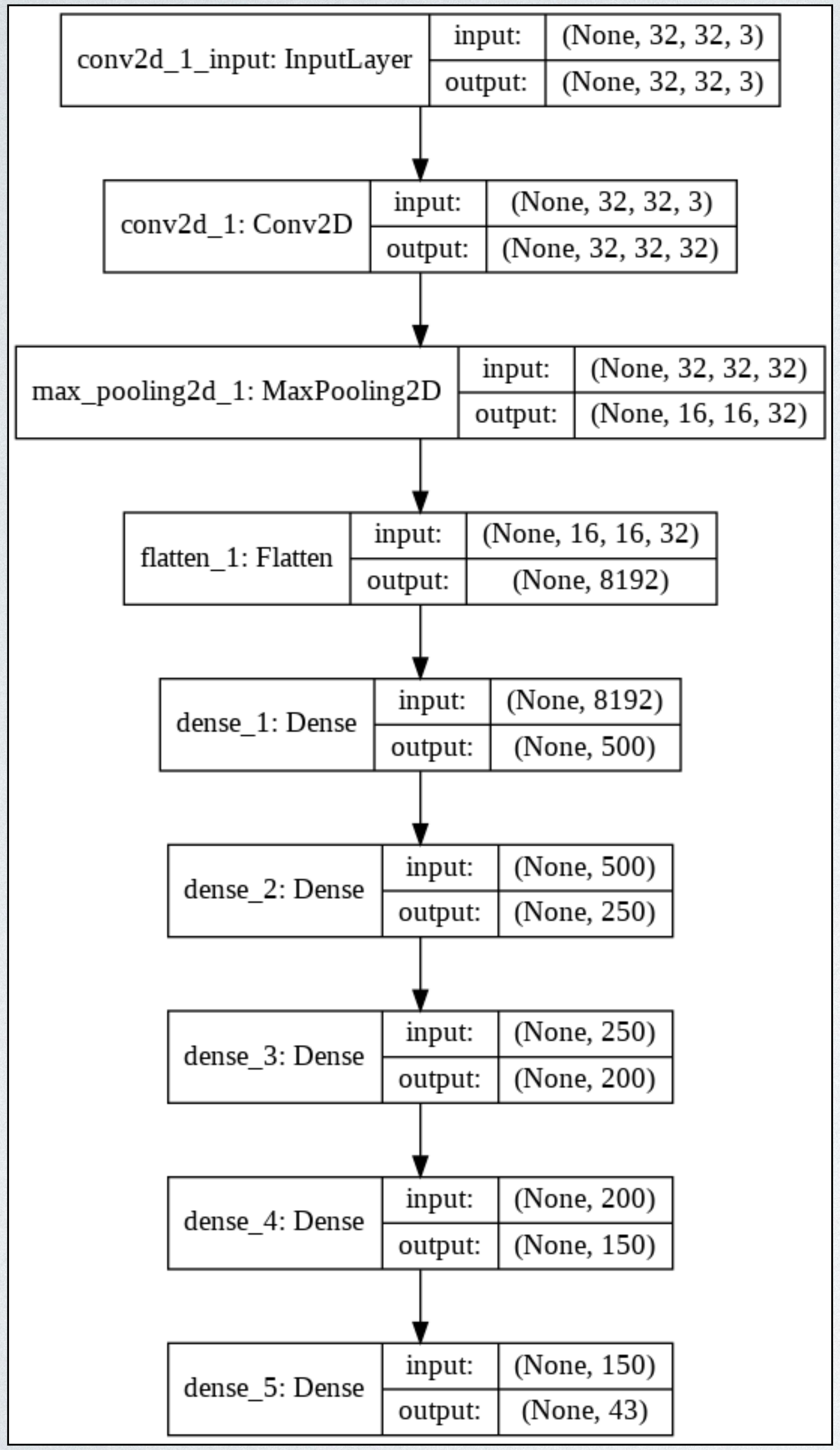
Random Sample from 10 different target classes

EXPLORATORY DATA ANALYSIS

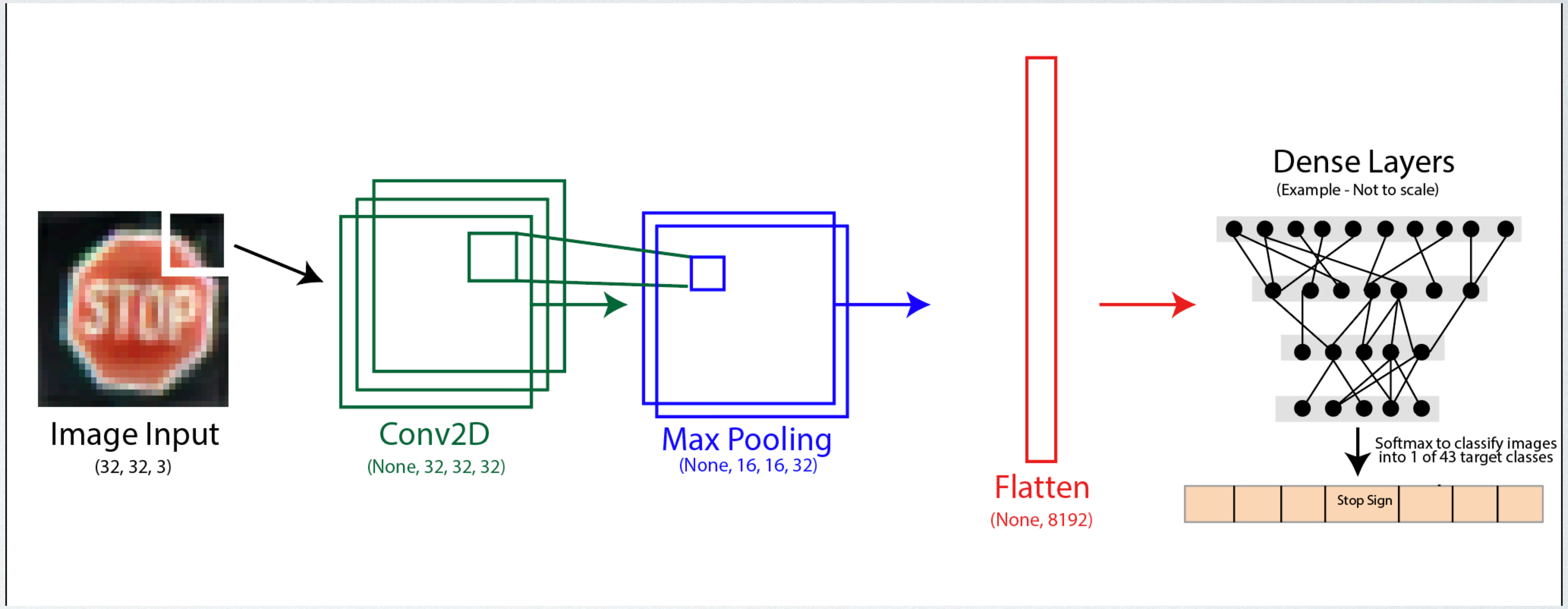
Training Dataset Class Distribution:



MODELING - BASELINE MODEL



Model Summary



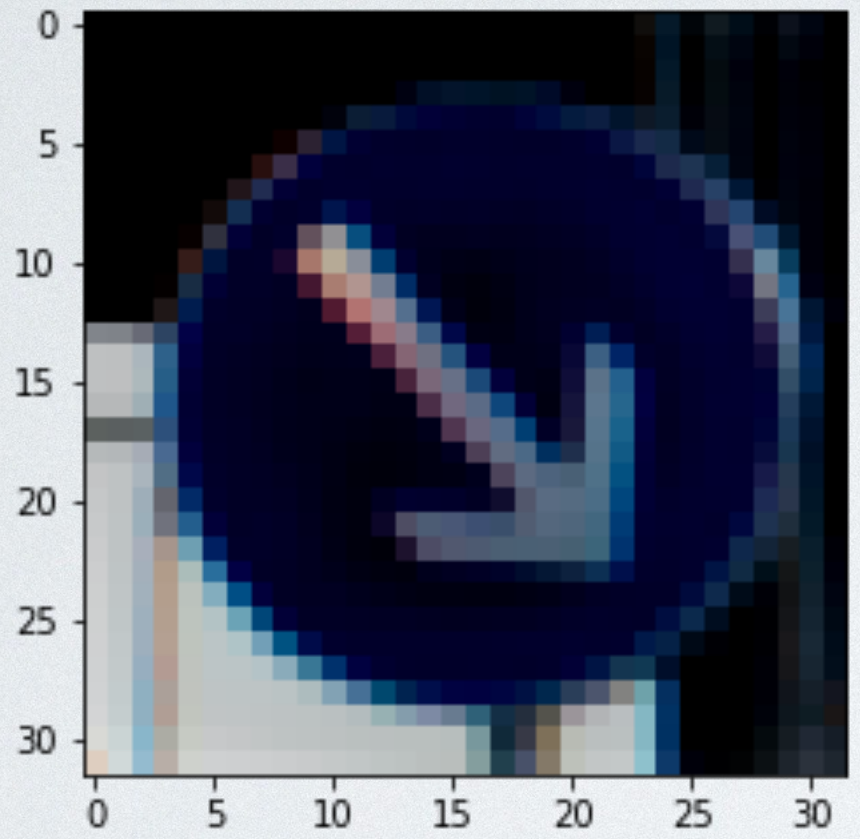
With the help of the figure above, summarizing our very basic CNN, we'll briefly discuss what is happening in each layer of our baseline CNN.

The first layer applied to the input image is a Convolutional layer. In this step, we apply a convolutional kernel to our image, effectively sliding a smaller 3x3 pixel filter over the input image, evaluating their dot products.

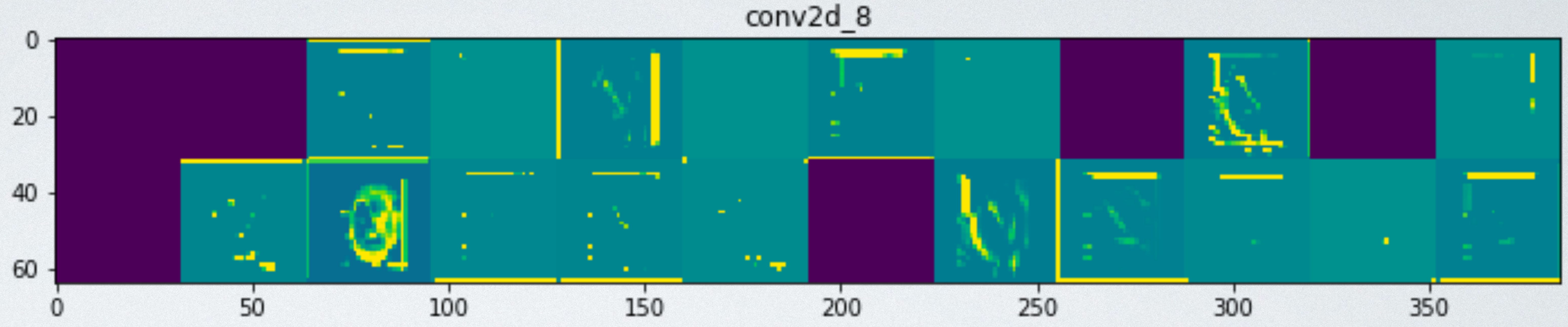
Next, a pooling layer is applied to reduce the dimensionality (and therefore parameters) of model input using a sliding 2x2 filter, selecting the maximum value for each "window" or "chunk" of our parsed image input.

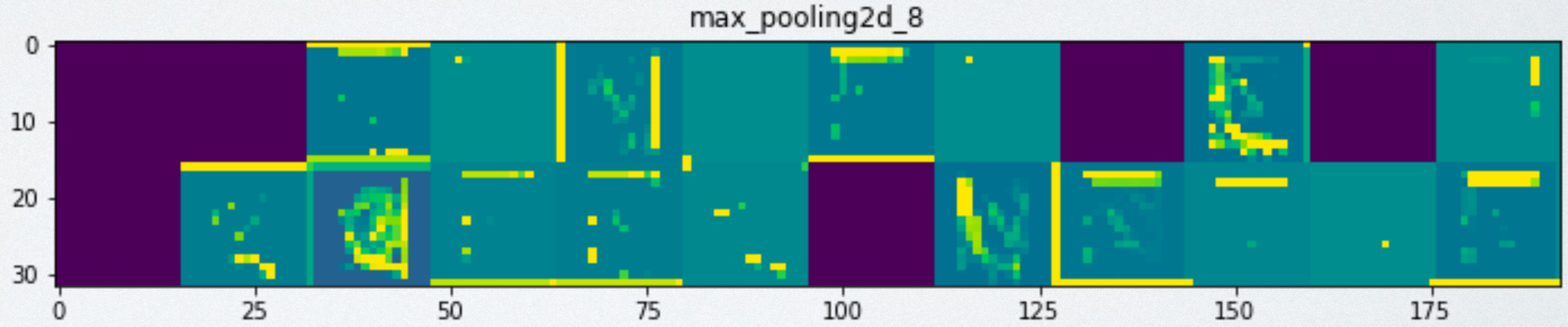
Finally, in order to massage our image input into a vectorized format compatible with the dense layers we'll use to classify our images we flatten the image input into an array.

BASELINE MODEL - INTERMEDIATE ACTIVATIONS

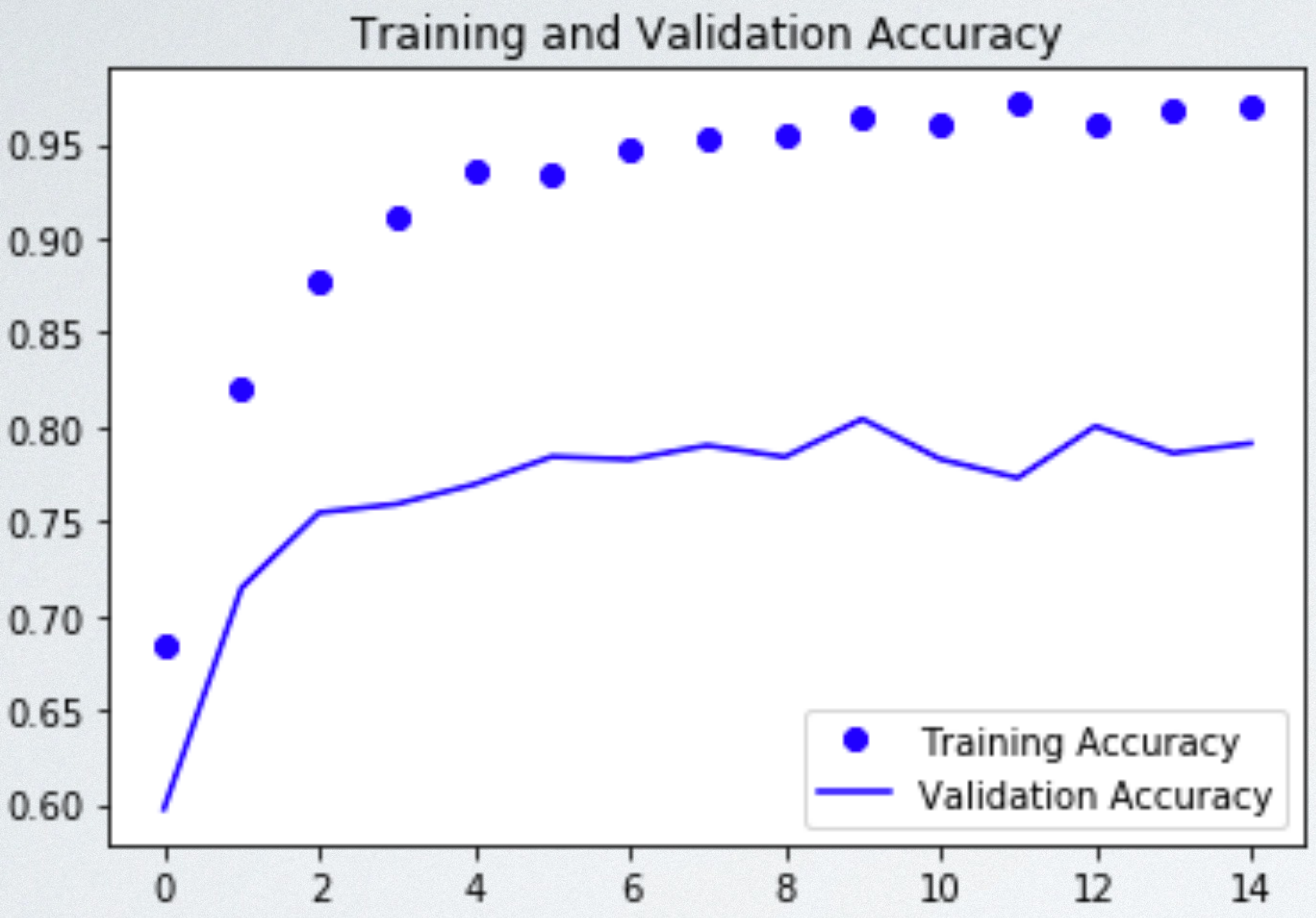


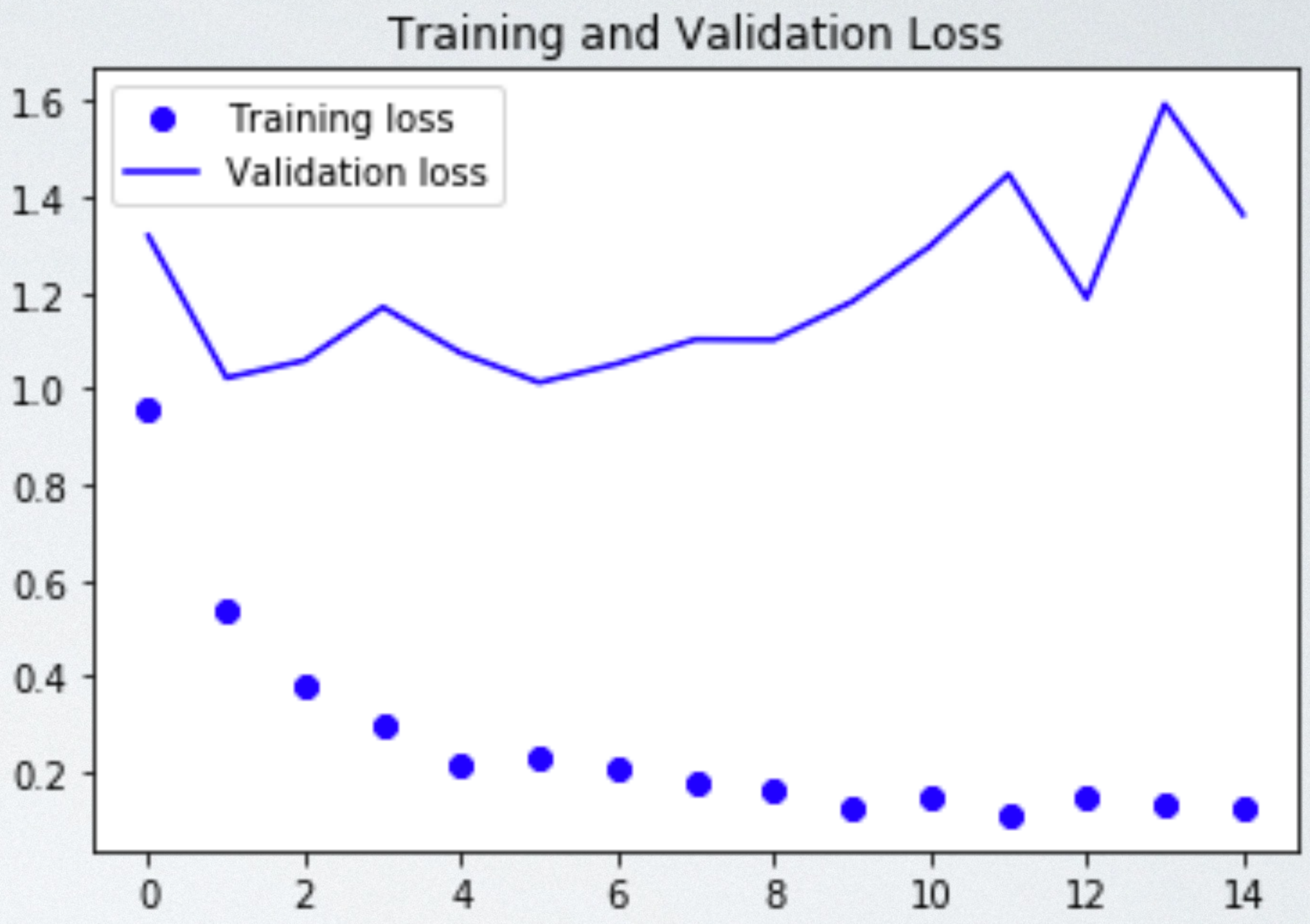
Example image from dataset





BASELINE MODEL - RESULTS

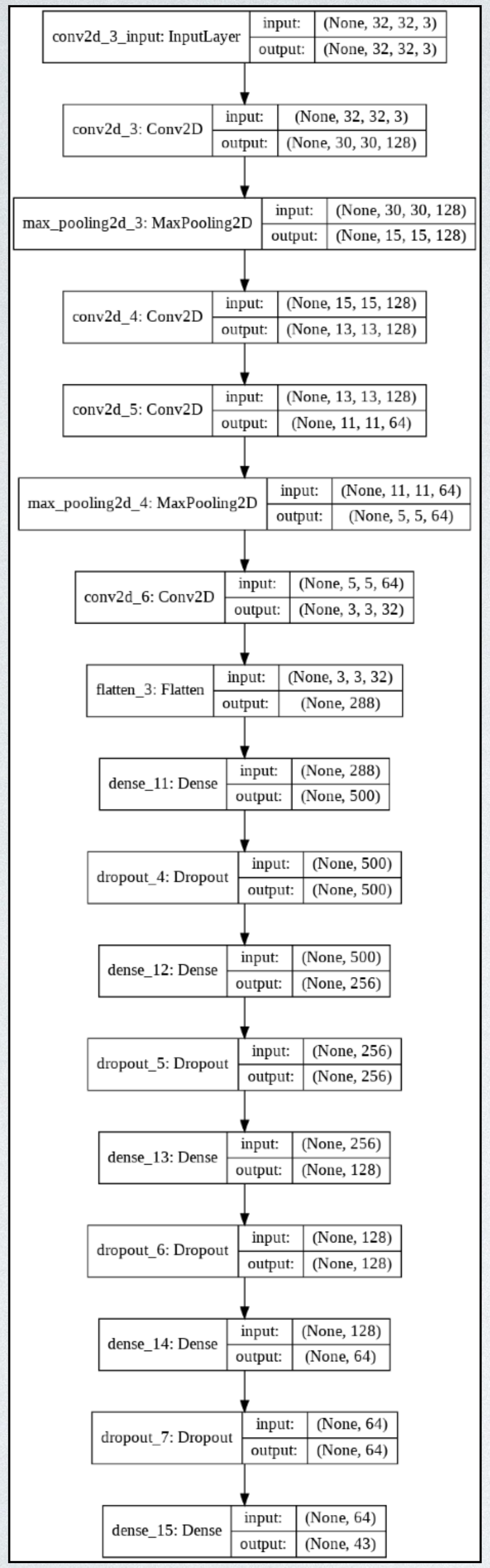




Over 15 epochs, training accuracy and loss quickly improve, with diminishing returns past the 10th epoch.

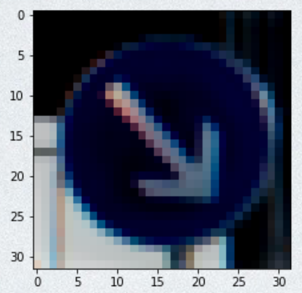
Validation accuracy and loss perform significantly worse, signifying our base model is overfit to our training data.

MODELING - TUNED MODEL

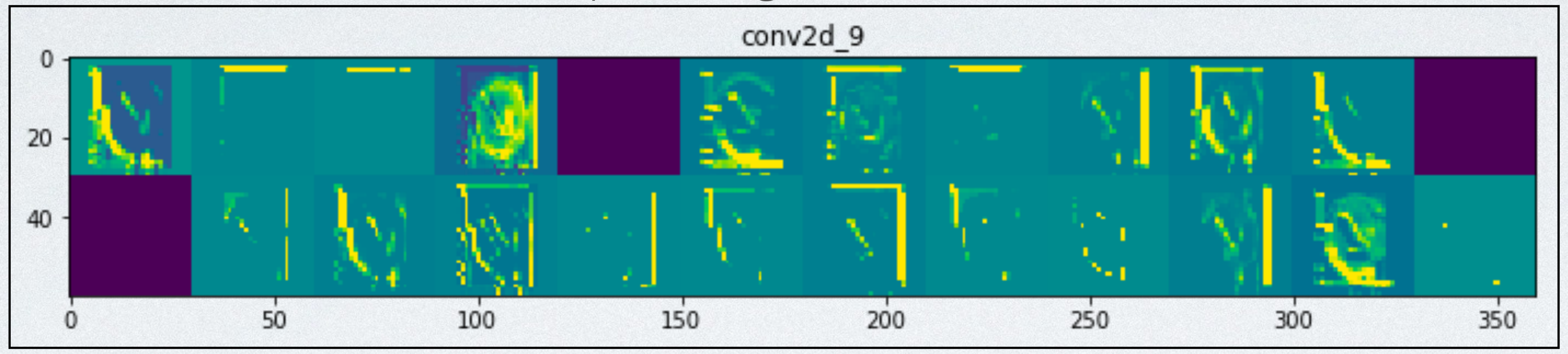


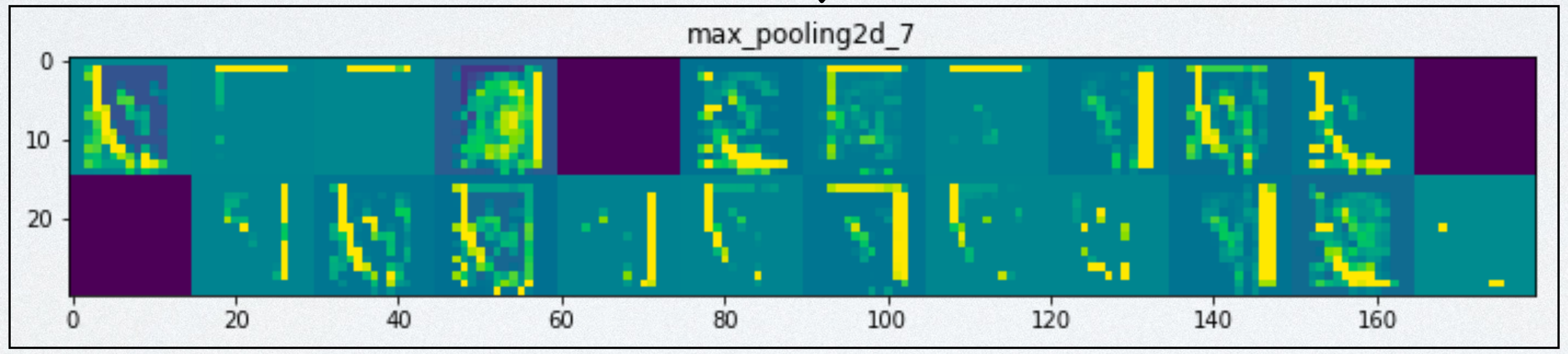
Model Summary

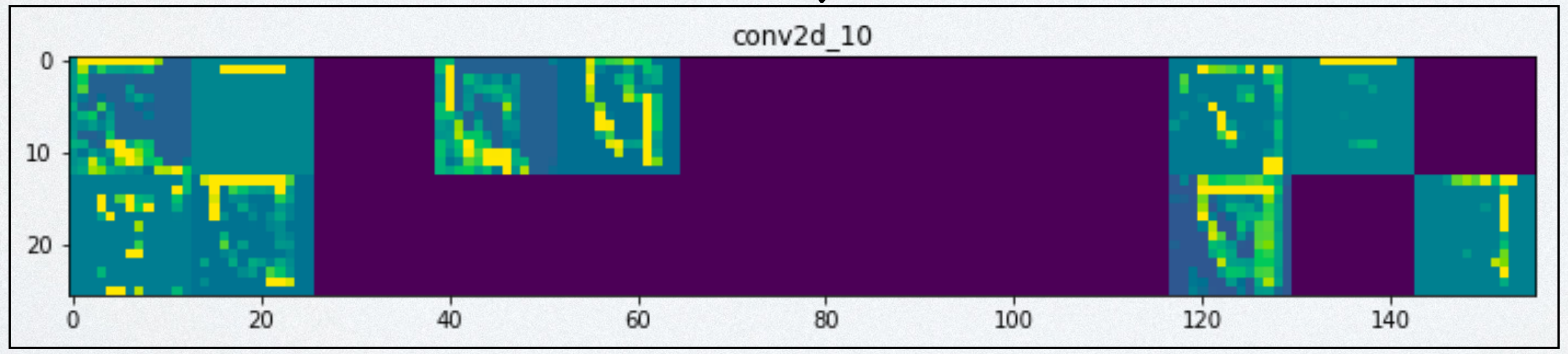
Intermediate Activations

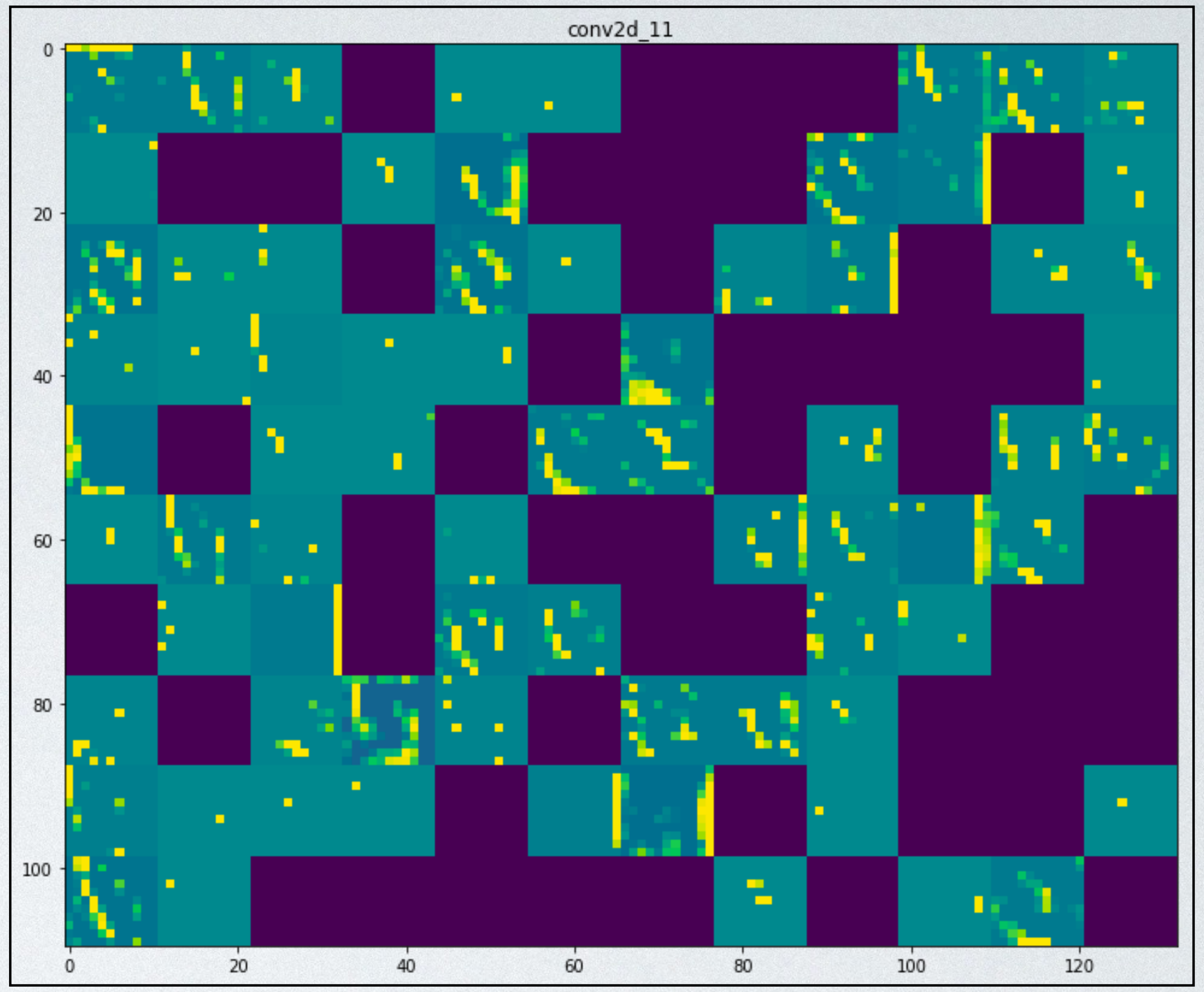


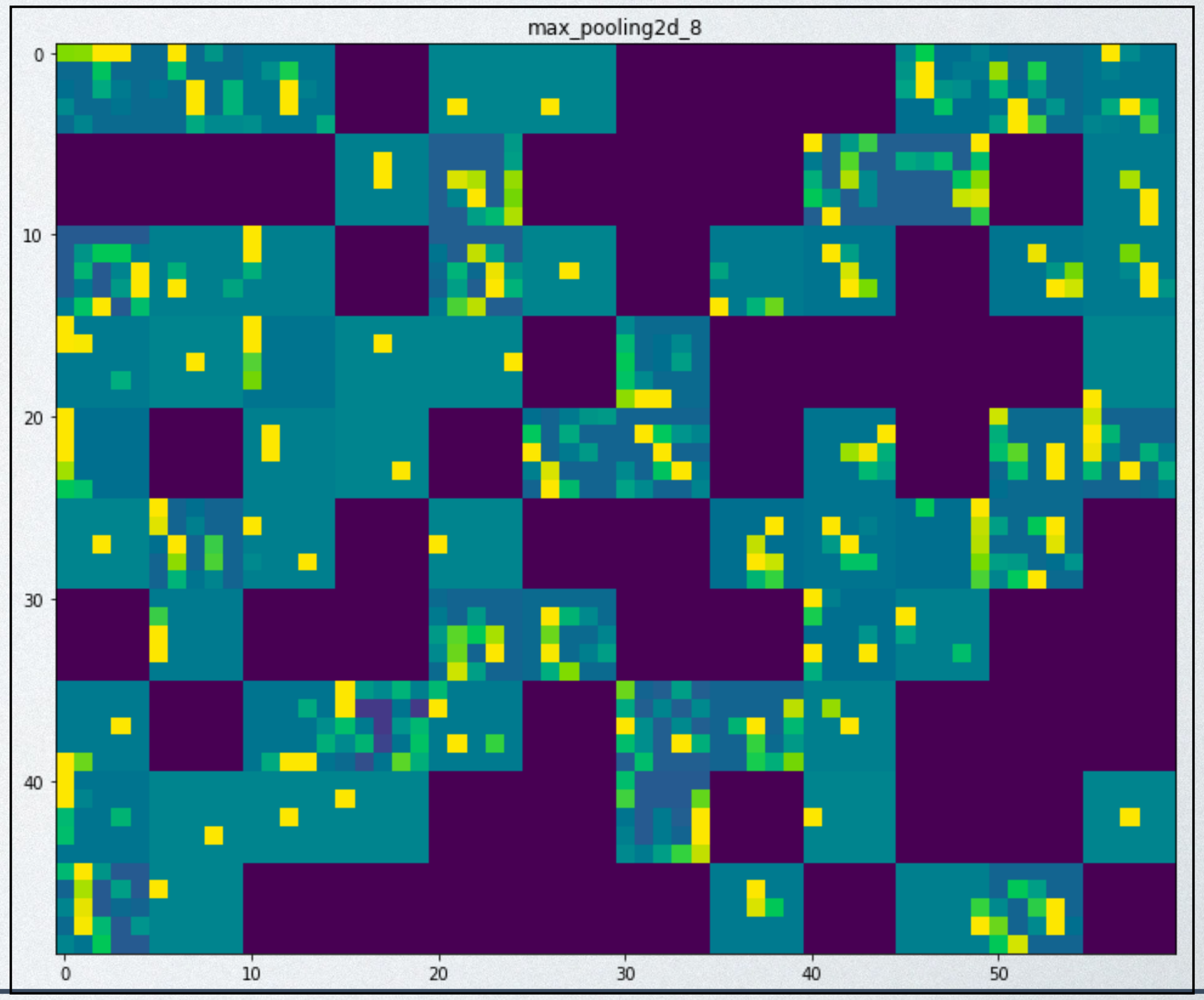
Example image from dataset



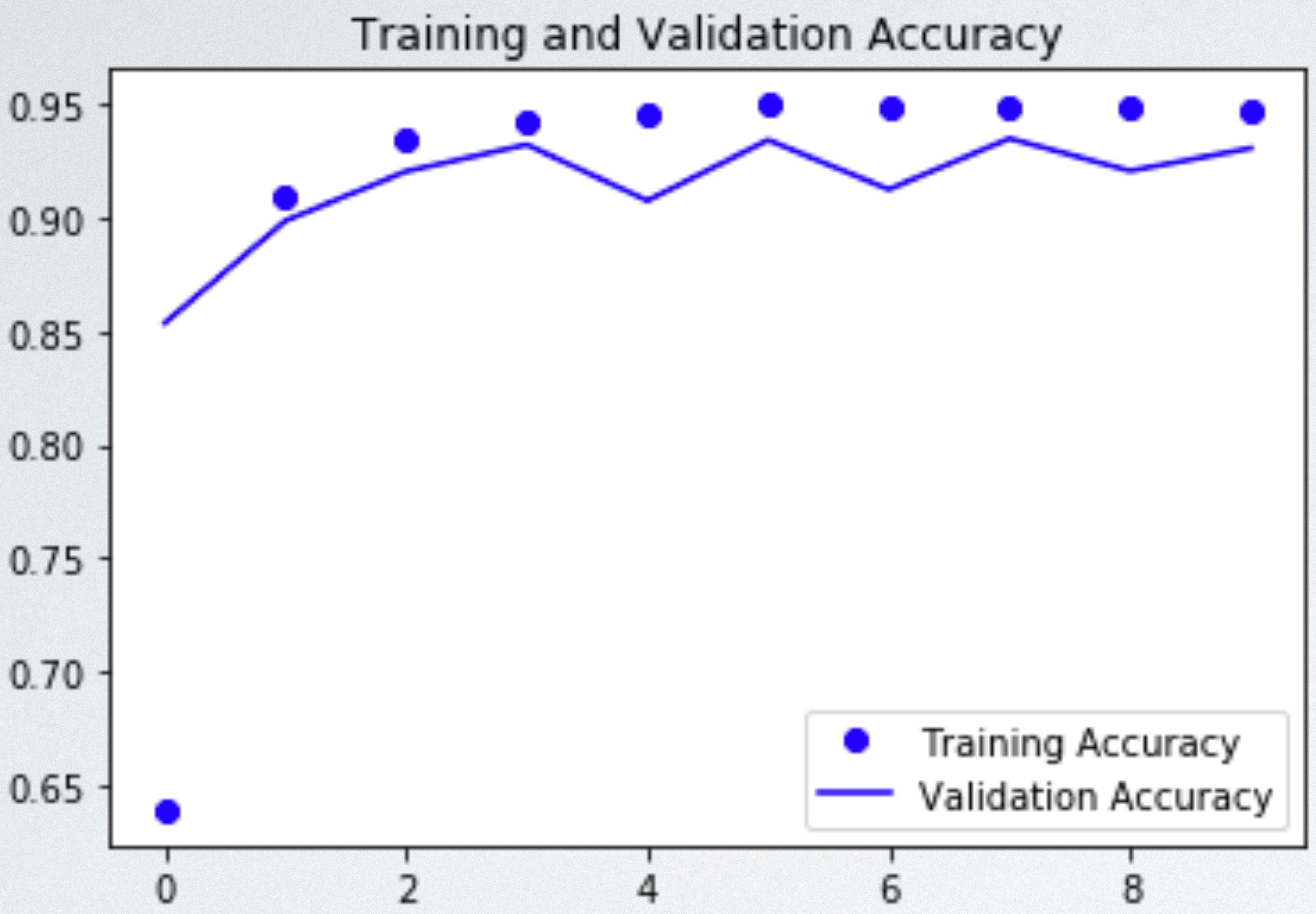


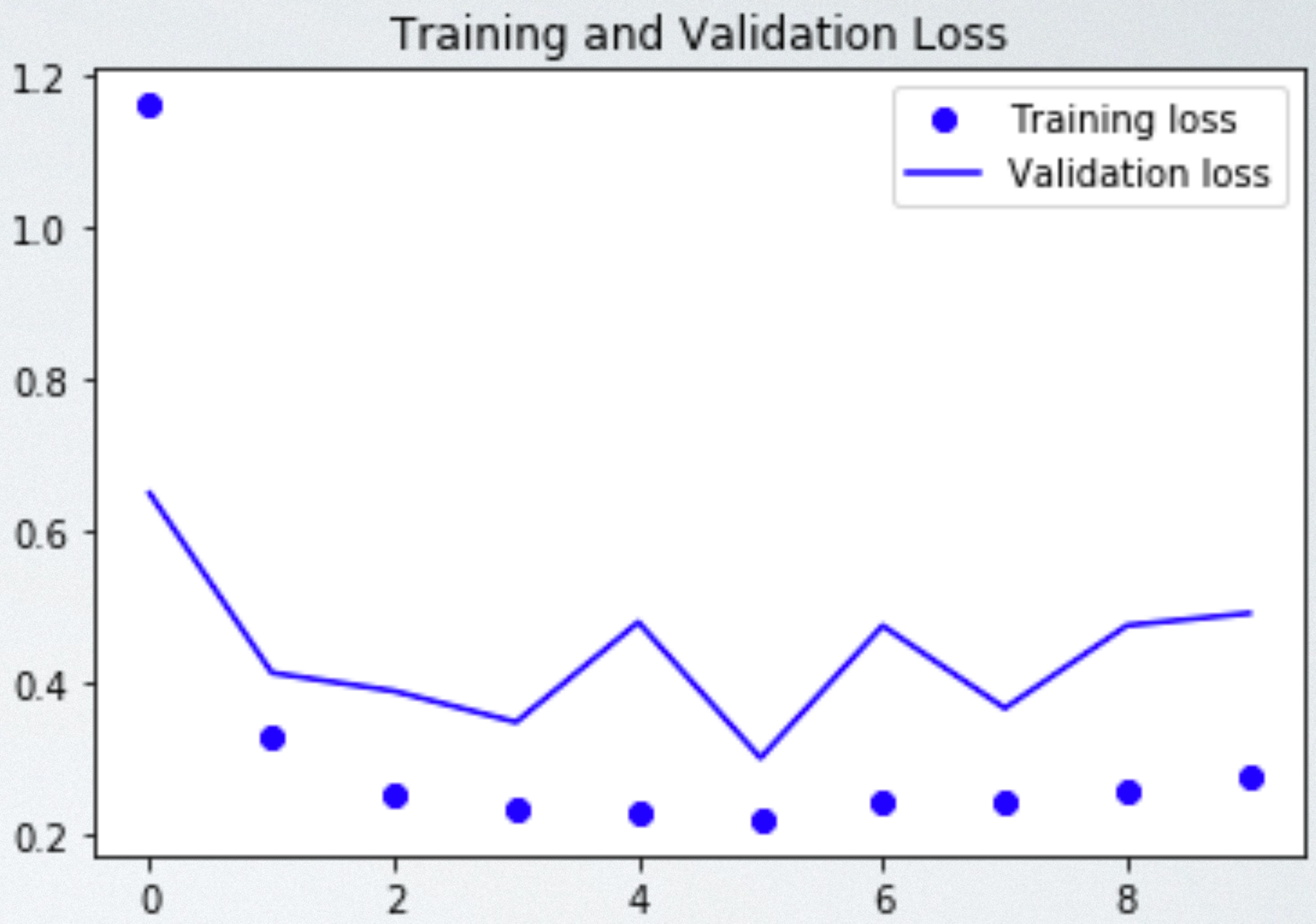






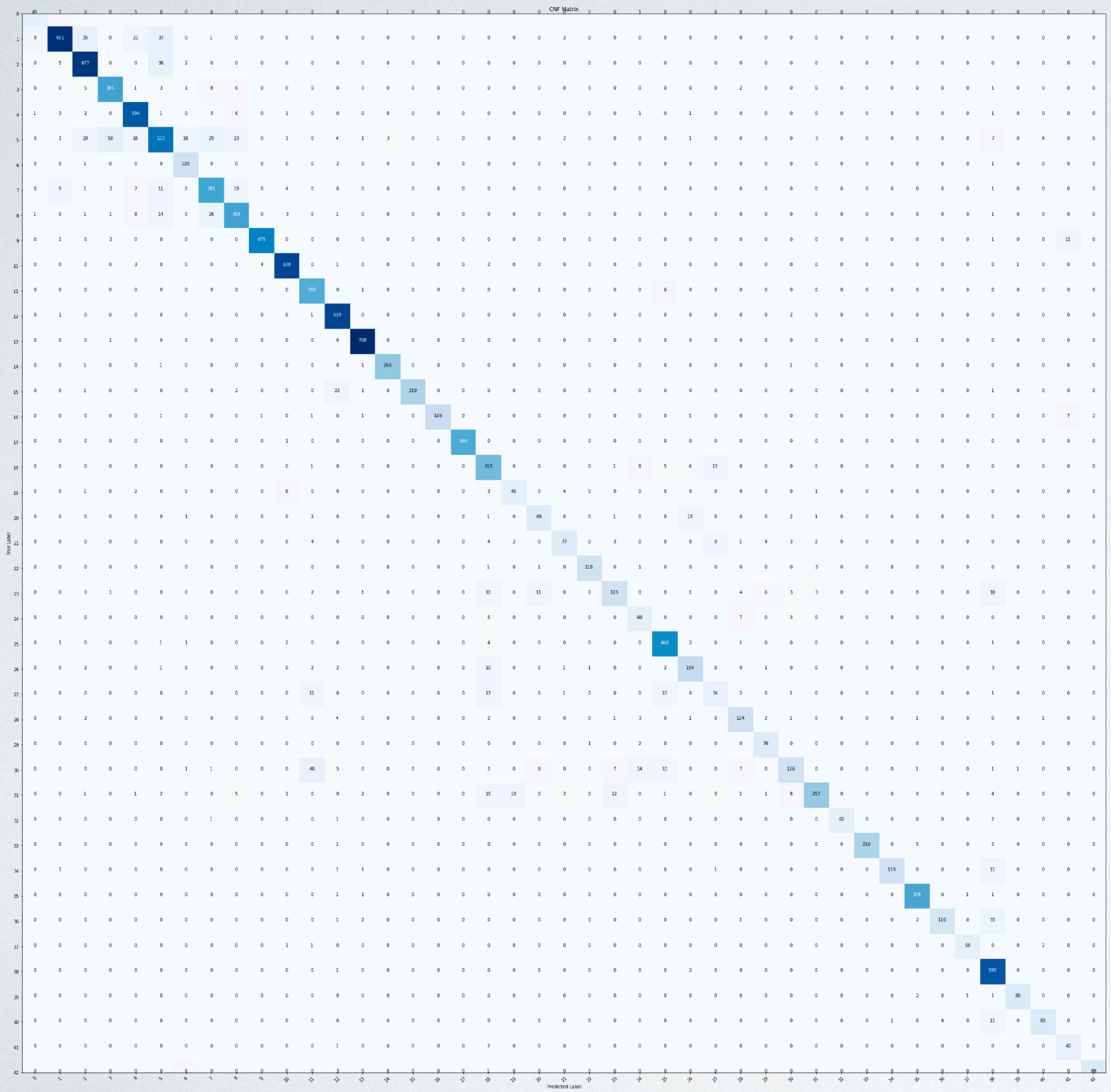
TUNED MODEL - RESULTS



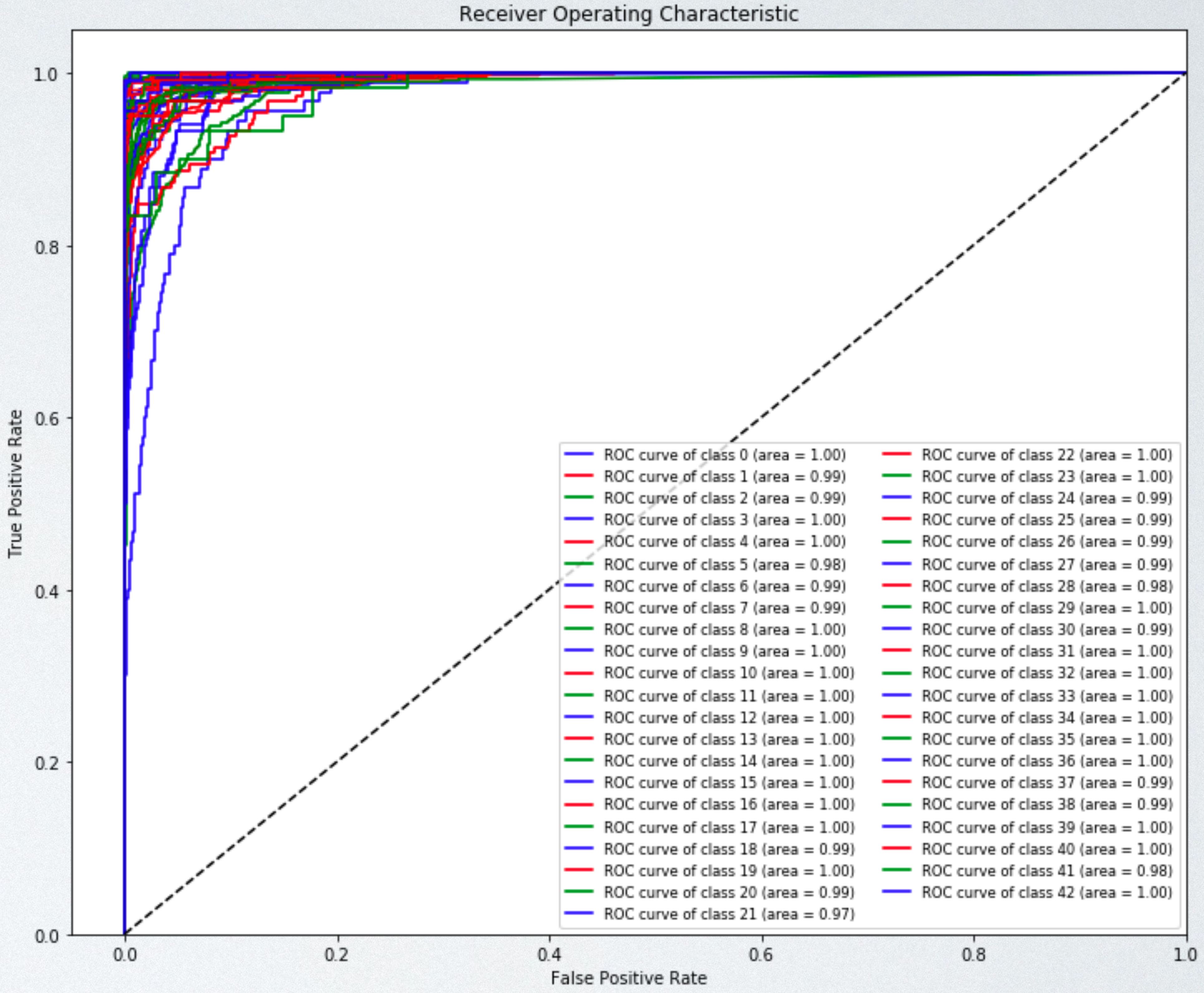


accuracy			0.91	12630
macro avg	0.88	0.89	0.88	12630
weighted avg	0.92	0.91	0.91	12630

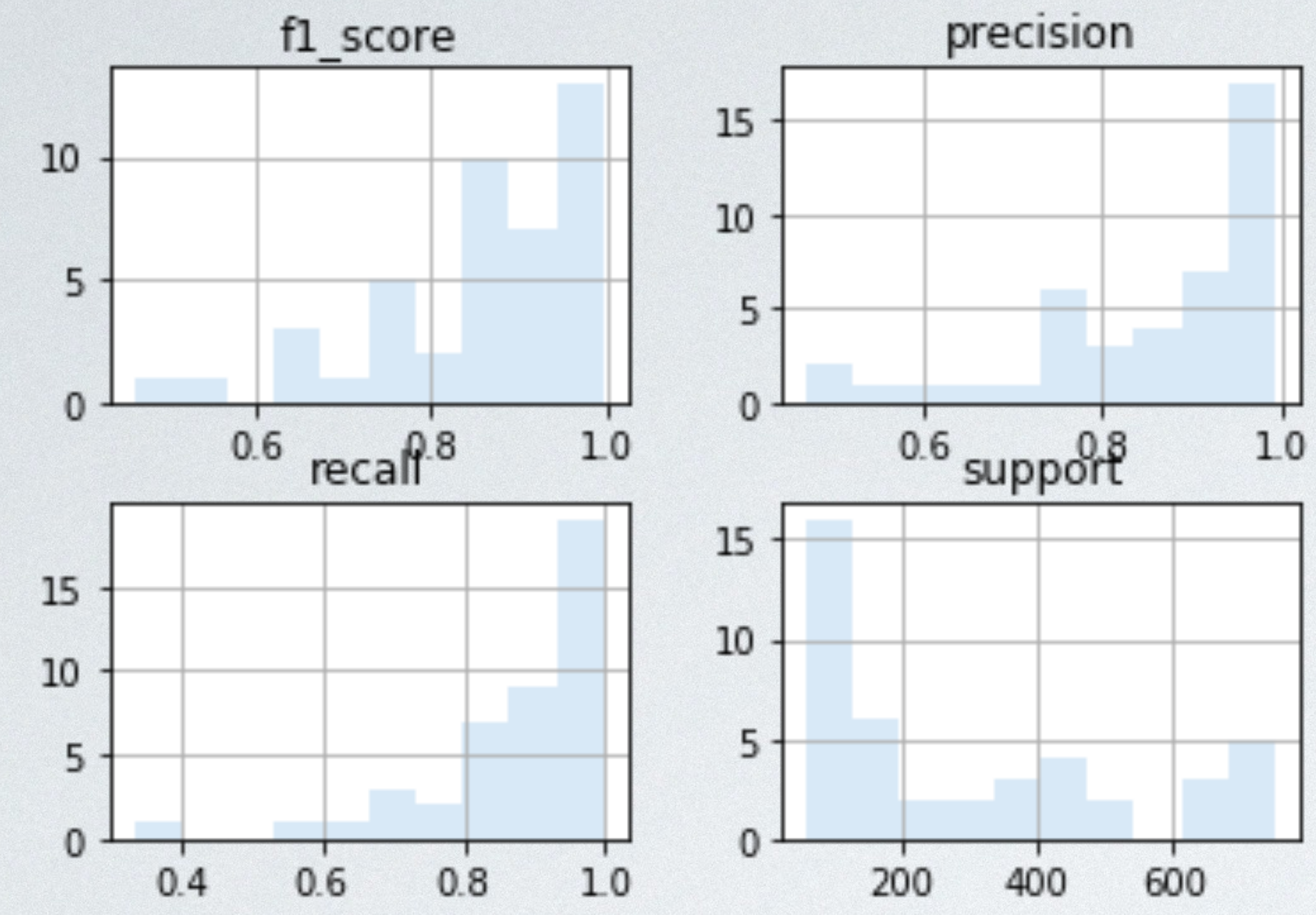
TUNED MODEL - RESULTS

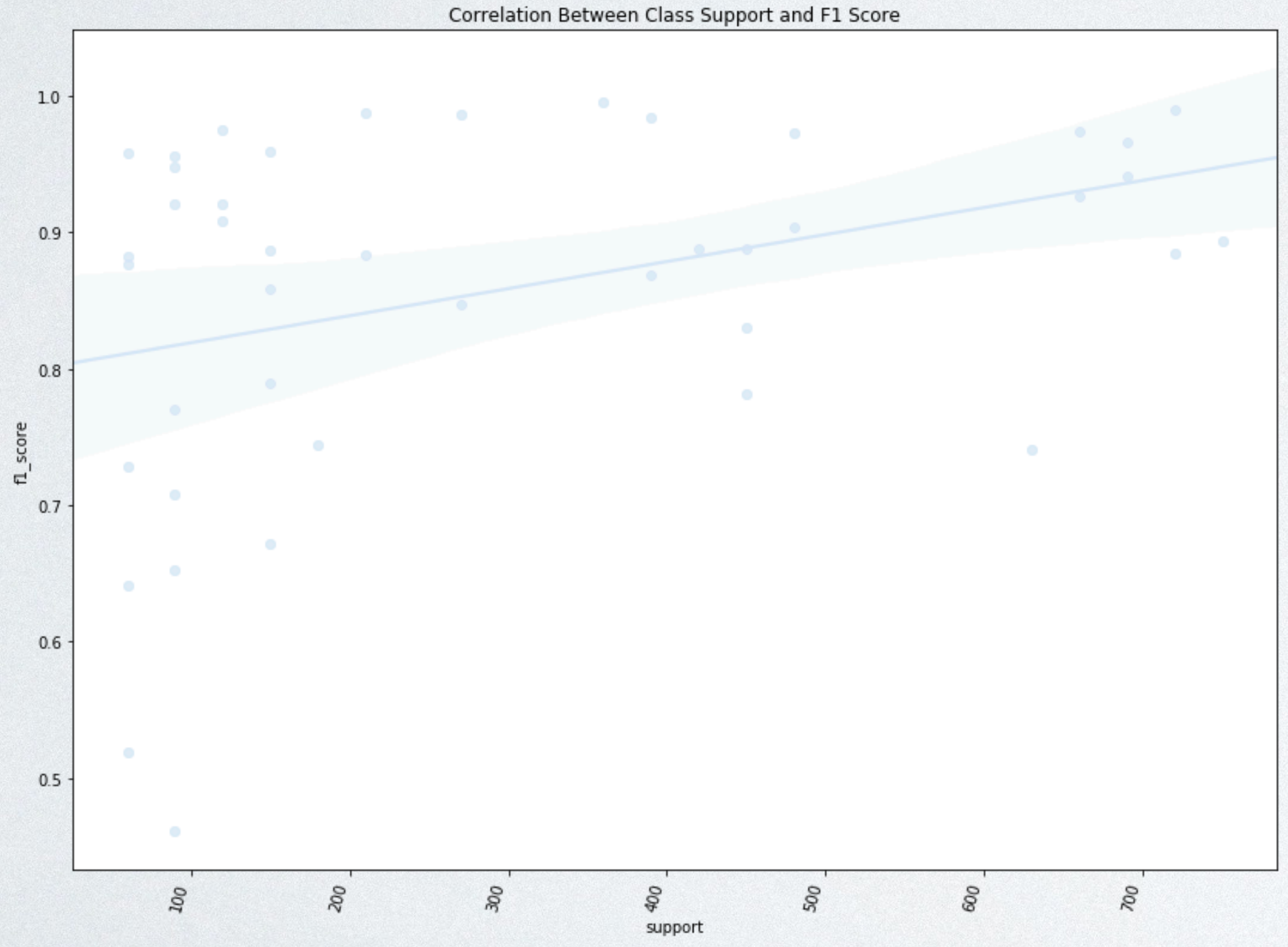


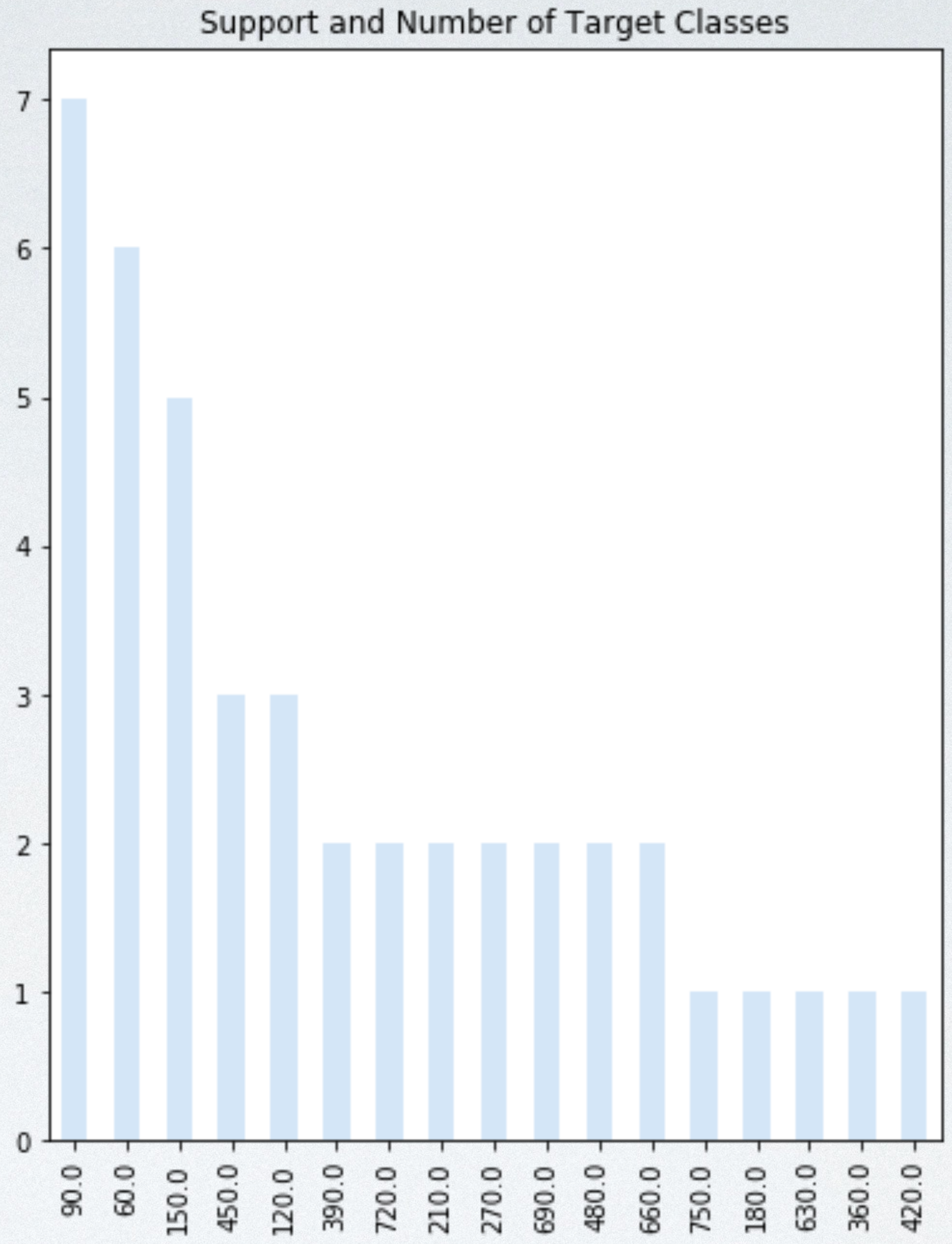
Avg F1: 0.91

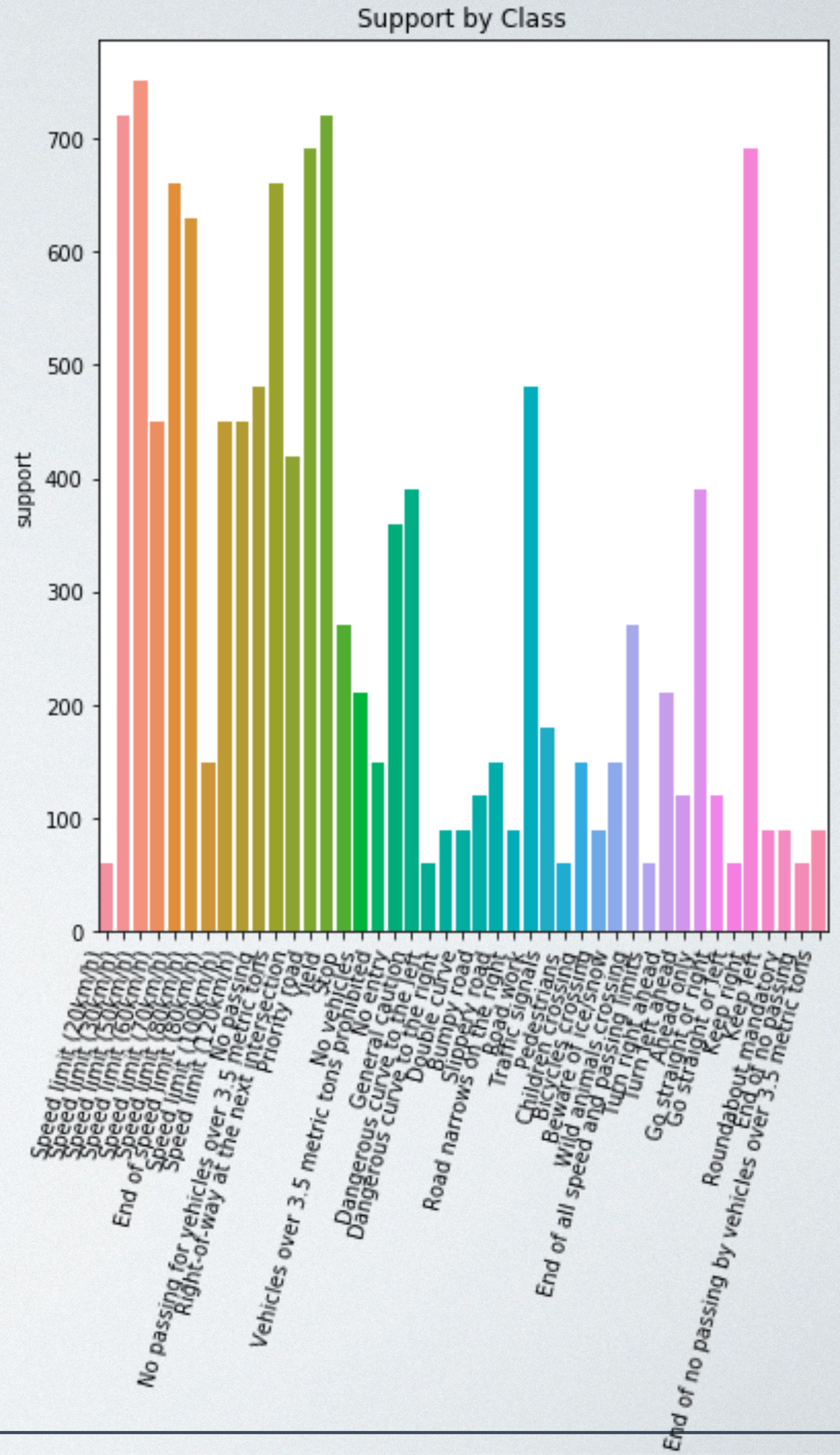


TUNED MODEL - RESULTS









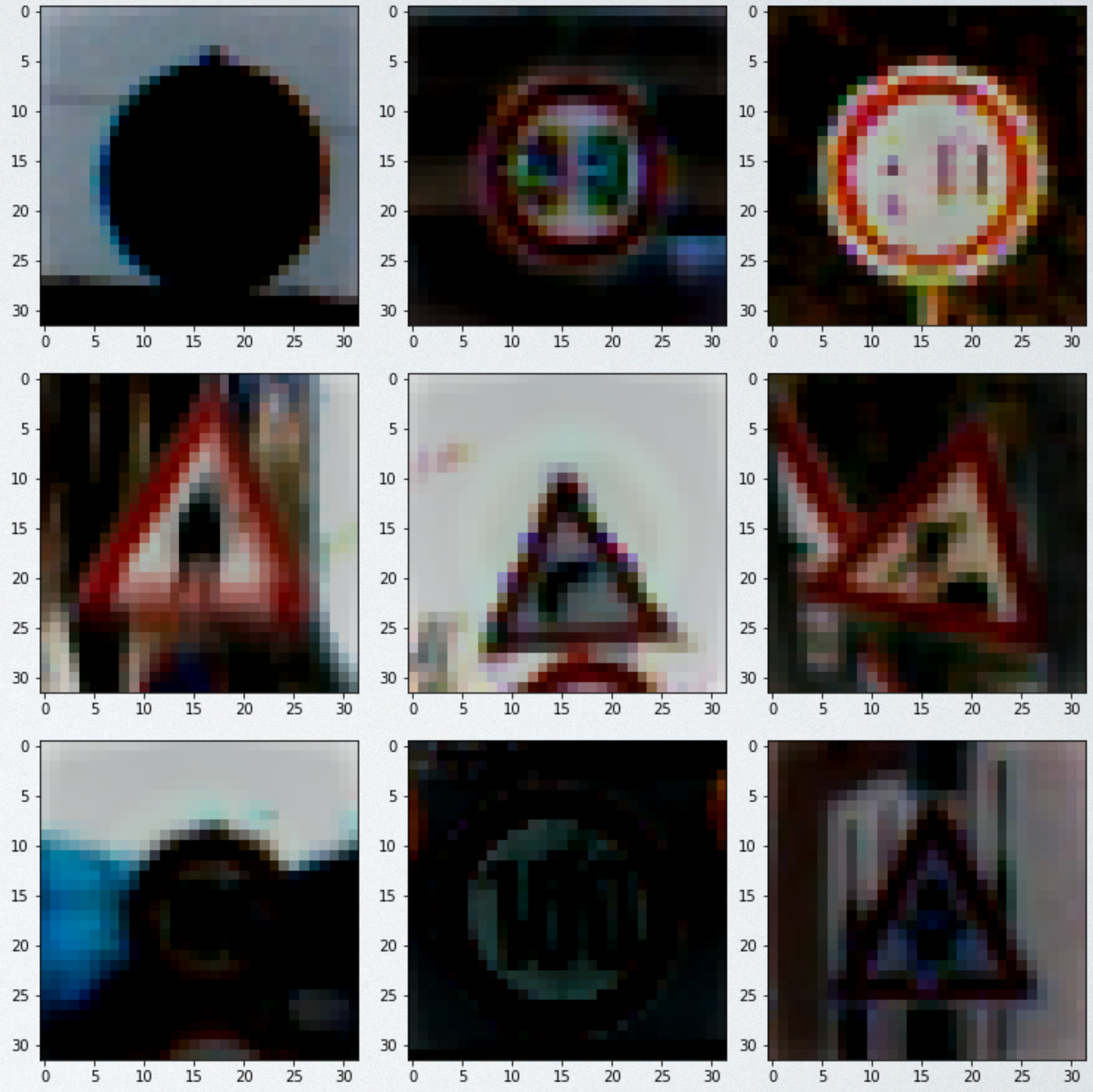
TUNED MODEL - WORST PERFORMERS

	precision	recall	f1_score	support	SignName
27	0.466667	0.583333	0.518519	60.0	Pedestrians
30	0.741935	0.613333	0.671533	150.0	Beware of ice/snow
19	0.741379	0.716667	0.728814	60.0	Dangerous curve to the left
24	0.603774	0.711111	0.653061	90.0	Road narrows on the right
23	0.845161	0.873333	0.859016	150.0	Slippery road

The classification report above shows that the model performed worst when classifying images of the following signs:

- Pedestrians
- Beware of ice/snow
- Dangeroud curve to the left
- Road narrows on the right
- Slippery Road

Examples of misclassified images:



CONCLUSIONS:

- Most if the images that were misclassified were either very dark or very low quality
- Higher quality images would likely result in model improvement, but would also greatly increase training time
- Additional pre-processing steps on misclassified images (such as increasing image brightness), would likely improve model performance
- Results underscore the importance of ethical concerns in implementation of automated driving and as well as general image classification processes

FUTURE IMPROVEMENTS:

- Attempt additional image pre-processing steps for images that appear to be very blurry or dark
- Score "importance" of each sign by weighting critical signs, to effectively "err on the side of caution"
- Apply transfer learning in order to utilize existing, very finely tuned CNNs to aid in classification

THANK YOU