EE454 Project 1 Report

Sean Curran Stephen Durofchalk Zachary Feldman
Ryan Wails

September 23, 2015

a Project Summary

b Procedural Approach

c Experimental Observations

c.1 Intermediate Output

Intermediate output generated by the CNN is included in the appendix attached to this report. Each figure shows the concatenated output of each layer as a gray scale image.

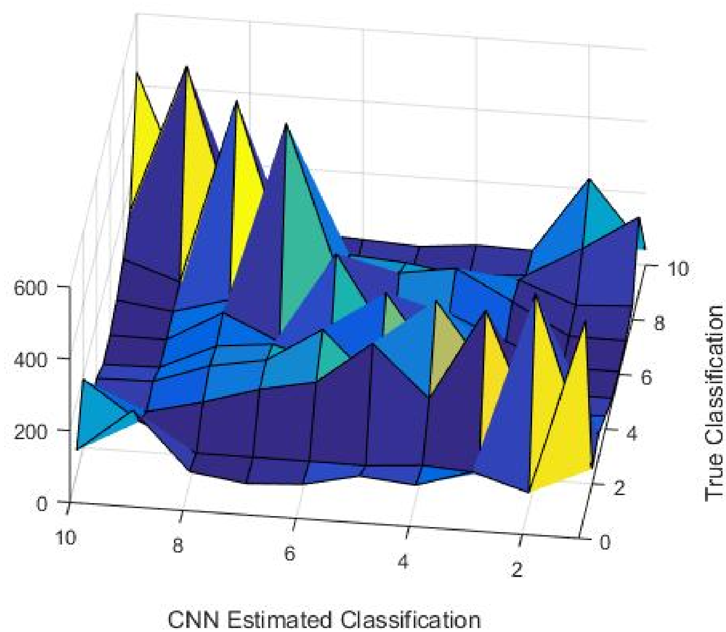
d Performance Evaluation

d.1 Results

The following results were the outcome of classifying images as the class associated with the highest probability in the probability vector.

	CNN Estimated Classification	1	2	3	4	5	6	7	8	9	10
True Classification											
1		531	41	65	37	10	8	18	38	210	42
2		40	519	9	26	10	7	19	29	111	230
3		87	8	386	117	97	70	104	88	25	18
4		39	18	127	325	45	136	186	89	13	22
5		53	6	270	69	259	38	162	114	22	7
6		19	7	151	222	49	281	111	125	20	15
7		10	7	120	125	93	23	557	33	9	23
8		32	7	73	98	77	94	54	533	13	19
9		192	84	35	44	7	8	10	16	542	62
10		69	191	23	41	4	9	30	68	127	438

$$\text{Accuracy} = 43.7\%$$



Discussion:

The surface plot above demonstrates how well the CNN classifies images. There is a very strong response on the diagonal which represents the correct classifications. What is interesting is that this matrix is somewhat diagonal. This demonstrates that if images in a class of images C_1 are being confused for images in a class of images C_2, then images in C_2 are also being confused for images in C_1. A lot of this confusion is entirely logical; for example, trucks and automobiles are frequently confused for each other. Another great example of confusion are cats and dogs are frequently confused for one another. We are again very impressed with how well the CNN because it not only detects similarities between images within a class, but it detects similarities amongst images spanning multiple classes.

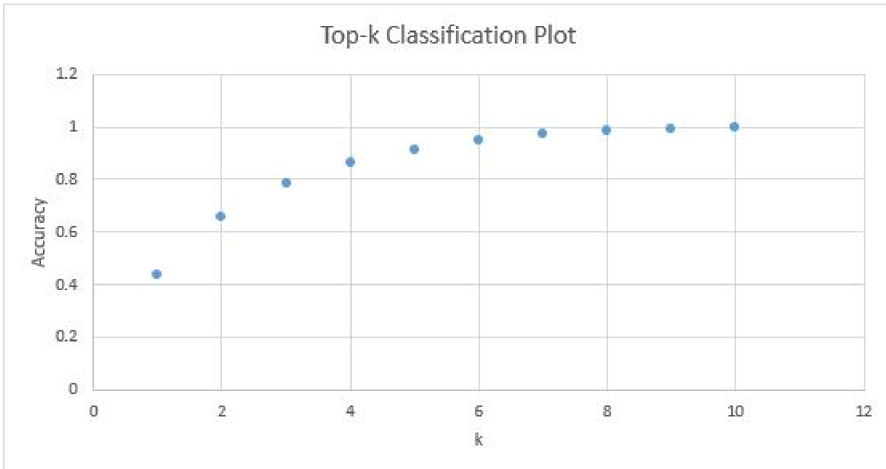
Note: The above results were produced using NeuralNet function located in 'NeuralNet.m'.

d.2 Looking at the Top-k Highest Probabilities

The following subsection looks at how frequently an image's true class appears in one of the top-k ranked classes, as determined by probability scores sorted from highest to lowest.

K	Accuracy
1	0.4371
2	0.6591
3	0.7864
4	0.8626
5	0.9135
6	0.9496
7	0.971
8	0.9847
9	0.9942
10	1

Note: Accuracy is the percentage of times that the correct object class appeared in the top-k ranked classes, as determined by probability scores sorted form highest to lowest.



Discussion: We were very surprised to see the top-k classification plot follow an exponential distribution. This demonstrates that the probability estimations provided by the CNN are accurate. We expected the probability estimations to be an arbitrary metric, but looking at the logarithmic nature of the plot it is clear that the probabilities are very accurate.

Note: The above results were produced using the topk function located in 'topk.m' and NeuralNet function located in 'NeuralNet.m'.

e Further Exploration

e.1 Test Cases

To test the robustness/accuracy of the training outside of the given data set, additional test images were gathered and fed into the CNN. Images were rescaled to thumbnail size (32x32x3) with the Matlab command

```
thumbnail = imresize(img, [32 32]);
```

110 images total were collected; 100 fell into the preexisting image classes, 10 contained scenes not falling into any preexisting class. Running these new 10 images through the CNN yielded the following results:

Image #	Image Contents	CNN Classification
Image 1	Tree in Field	Horse (8)
Image 2	Winter Mountain Scene	Ship (9)
Image 3	Ryan and Zach	Bird (3)
Image 4	Steve	Frog (7)
Image 5	Sean	Cat (4)
Image 6	Desktop Computer	Automobile (2)
Image 7	Football on Field	Deer(5)
Image 8	Robert Collins	Truck (10)
Image 9	Old Main	Bird(3)
Image 10	Penn State Logo	Truck (10)

Note: All of the exploratory images we used were either taken by us or were taken from Google Images.

e.2 Reclassification

An attempt was made to distinguish these 10 images from the rest of the data set (i.e. reclassify these 10 images as unknown). Let V be the vector of output probabilities from the CNN for each image. Let C be the classification of the image producing probability vector V. Originally, we have

$p_i \in V$
$p_{max} = \max(p_1, p_2, ..., p_{10})$
$C = i$ given $p_i = p_{max}$

To reclassify the images we use

$p_{max} = \max(p_1, p_2, ..., p_{10})$
$P = \{p_1, p_2, ..., p_{10}\} \setminus \{p_{max}\}$
$p_{2max} = \max(P)$

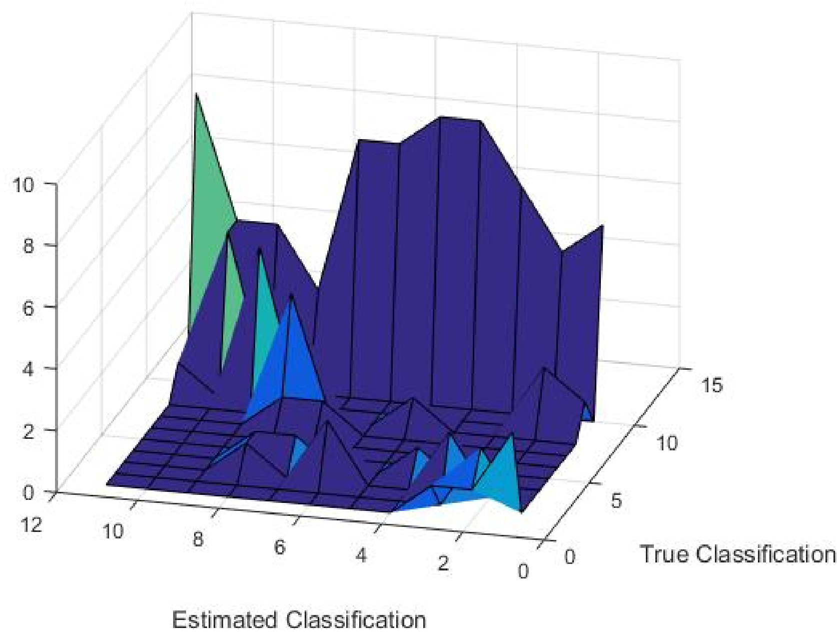
$$\begin{cases} C = i \text{ given } p_i = p_{max} & \text{for } p_{max} - p_{2max} >= 0.25 \\ C = 11 & \text{otherwise} \end{cases}$$

Less precisely, the difference between the two peak responses was thresholded by .25; so, if there were two strong responses, the image is reclassified as unknown.

e.3 Results After Reclassification

In the confusion matrix below, classes 1 through 10 are the same classifications. Class 11 identifies an unknown image class.

	CNN Class	1	2	3	4	5	6	7	8	9	10	11
True Class												
1		3	0	0	0	0	0	0	0	1	0	6
2		1	2	0	0	0	0	0	0	2	0	5
3		1	0	2	0	0	0	0	0	0	0	7
4		0	0	0	1	0	0	0	0	0	0	9
5		0	0	0	0	0	0	0	1	0	0	9
6		0	0	2	0	0	0	0	0	0	0	8
7		0	0	0	1	0	0	1	0	0	0	8
8		0	0	1	1	0	0	1	4	0	0	3
9		0	0	0	0	0	0	0	0	5	0	5
10		0	0	0	0	0	0	0	0	0	5	5
11		0	0	0	0	0	0	0	0	1	0	9



Accuracy = 29.09%
The reclassification metric does well in replacing the 10 new images in the unknown category, but also places images that were previously correctly classified in the unknown category. Other metrics tested were

1. Thresholding on the number of classes that had a response above 0.1 (the mean value for random classification)

2. Taking the spatial derivative of the probability vector; this metric actually produces zero-mean Gaussian noise with a very small standard deviation.

f Member Contributions

Overall the contributions made form each team member were pretty even. Whenever the team met to work on the project most of the team members were in attendance. Everyone who was in attendance at each team meeting attempted to work on a different piece of the project to ensure that we were making steady progress. The coding was not split up evenly as one team member did not attend the initial meeting and as a result he missed most of the actual coding and basically all of the design decisions. We attempted to split the report up evenly, but as the deadline approached scheduling meetings became difficult and some individuals were unavailable to work on the report. In conclusion, the work was not split up evenly among our four group members, but everyone attempted to contribute.

Appendices

A Intermediate Output

Figure 1: Original Image

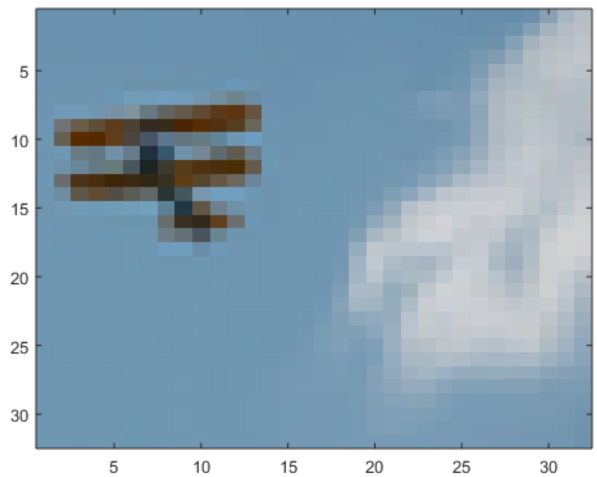


Figure 2: Layer 1 Output

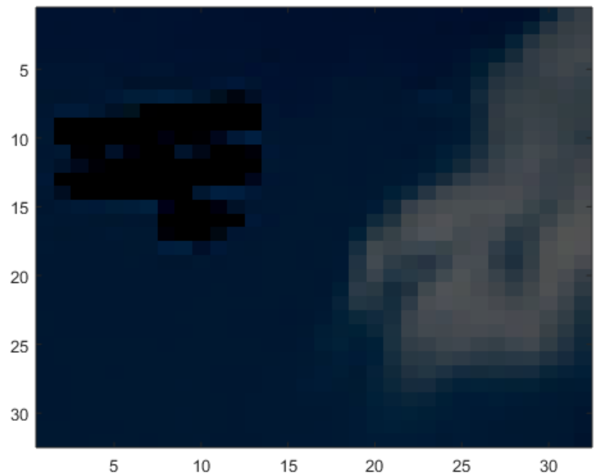


Figure 3: Layer 2 Output

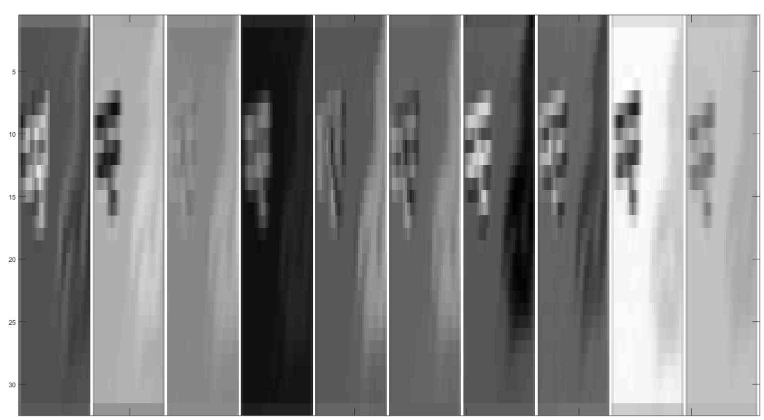


Figure 4: Layer 3 Output

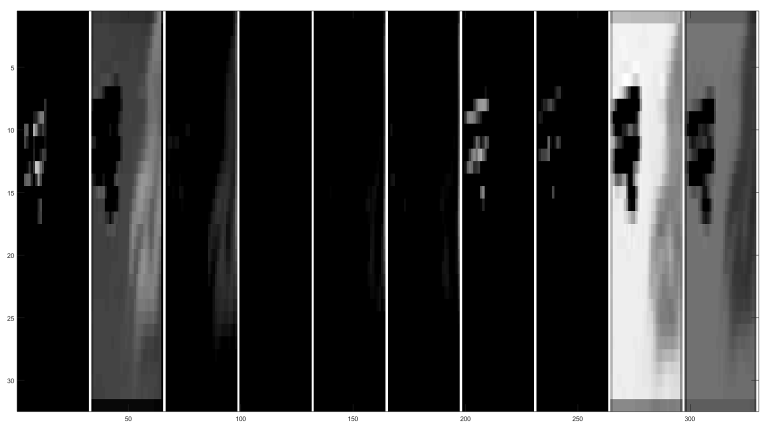


Figure 5: Layer 4 Output

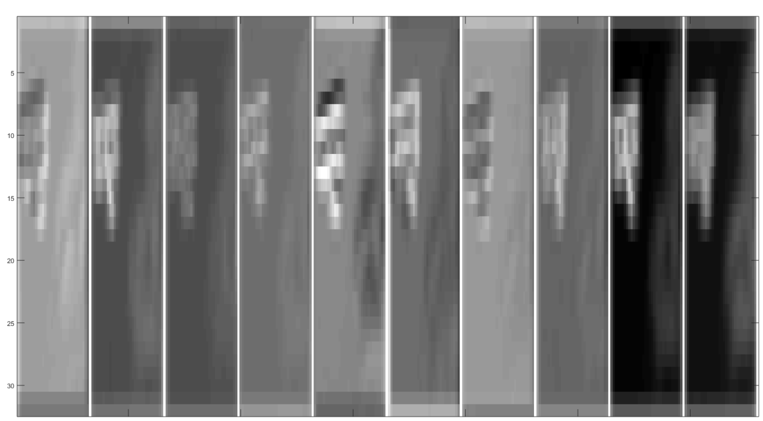


Figure 6: Layer 5 Output

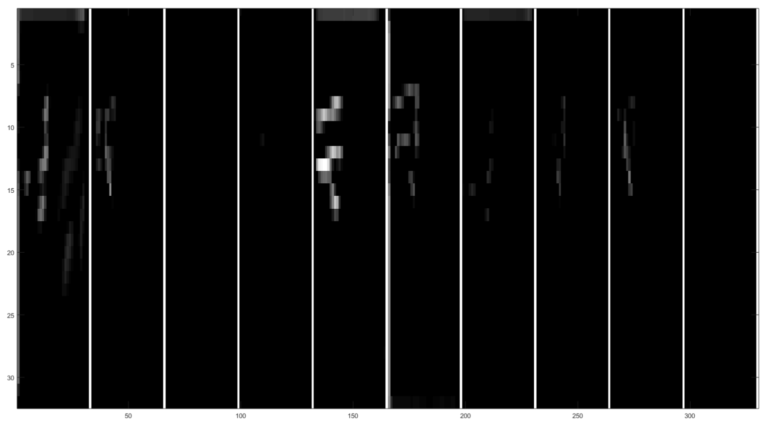


Figure 7: Layer 6 Output

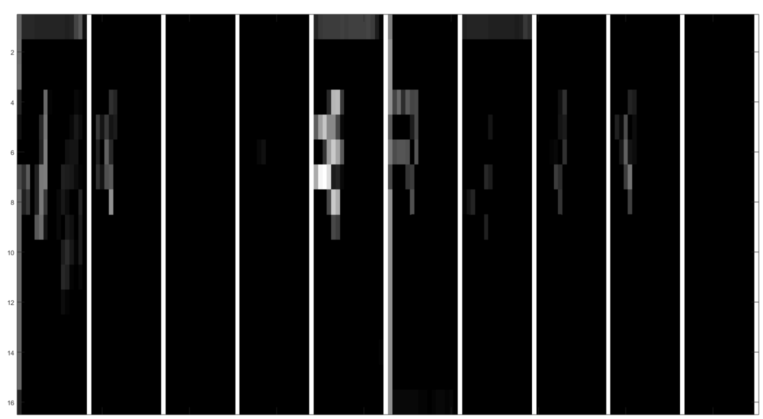


Figure 8: Layer 7 Output

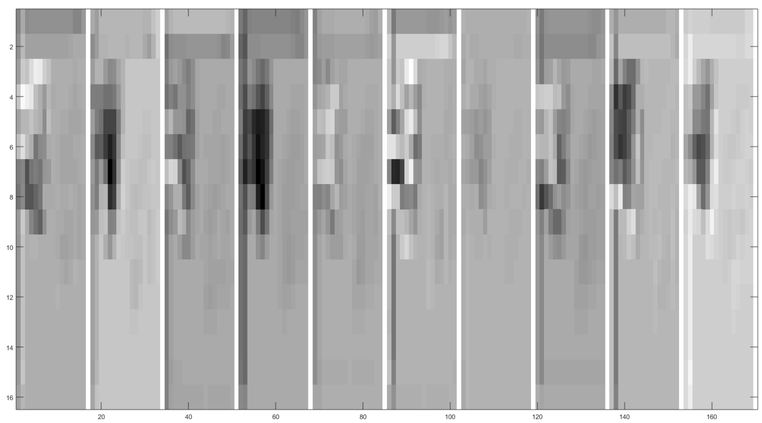


Figure 9: Layer 8 Output

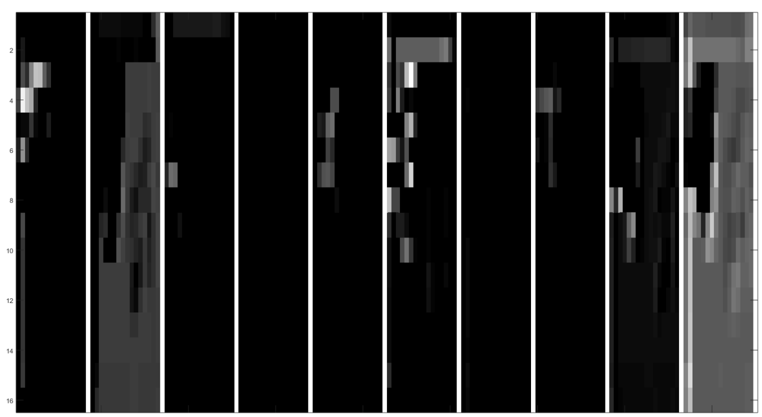


Figure 10: Layer 9 Output

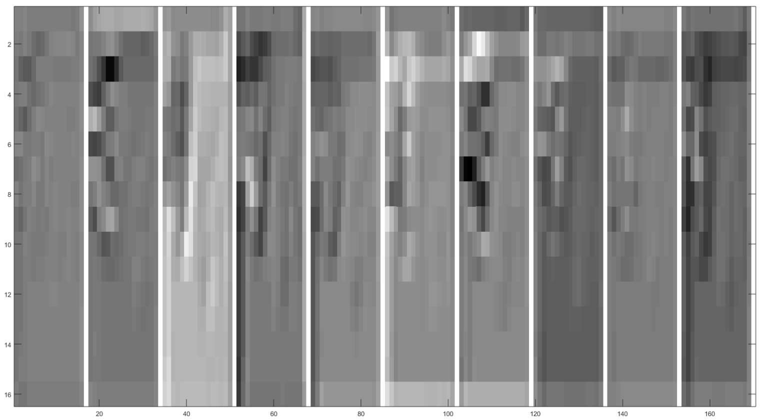


Figure 11: Layer 10 Output

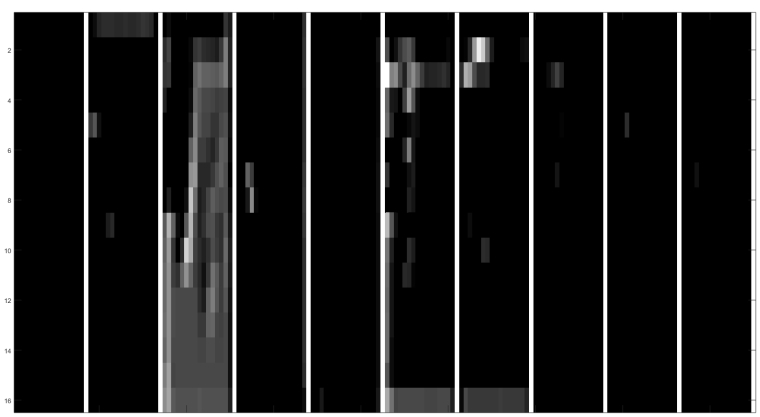


Figure 12: Layer 11 Output

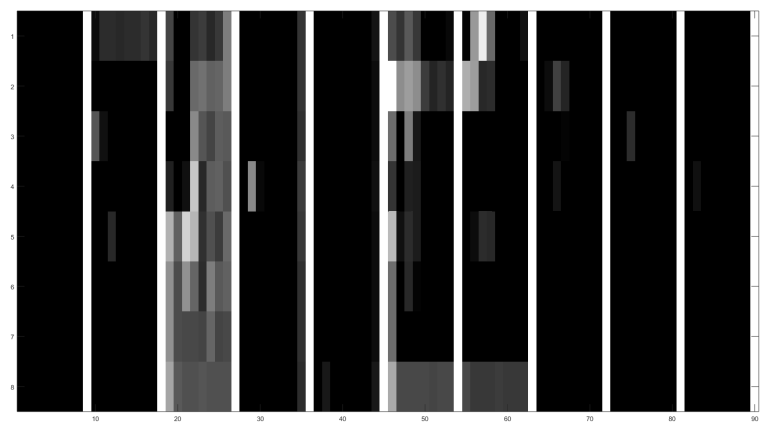


Figure 13: Layer 12 Output

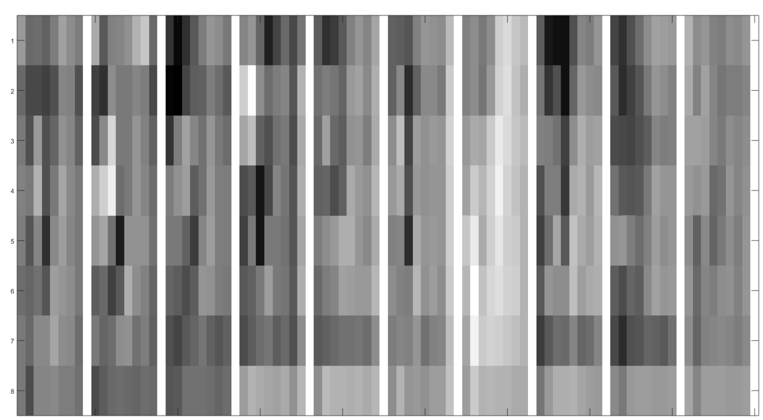


Figure 14: Layer 13 Output

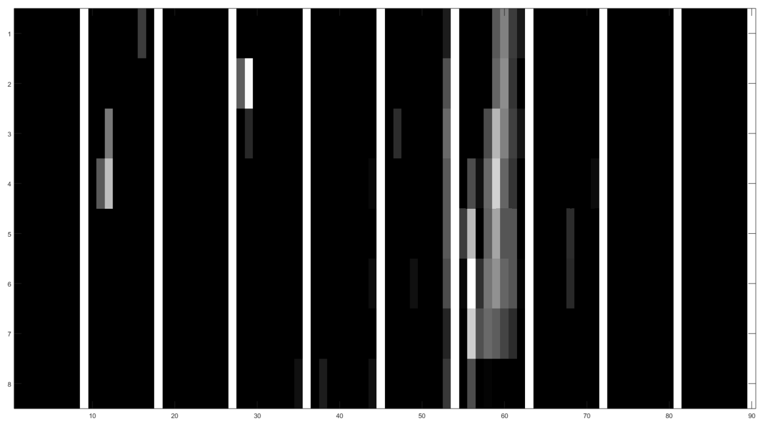


Figure 15: Layer 14 Output

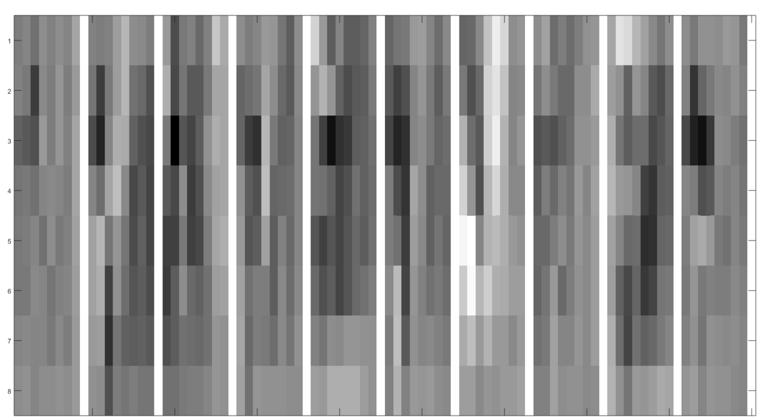


Figure 16: Layer 15 Output

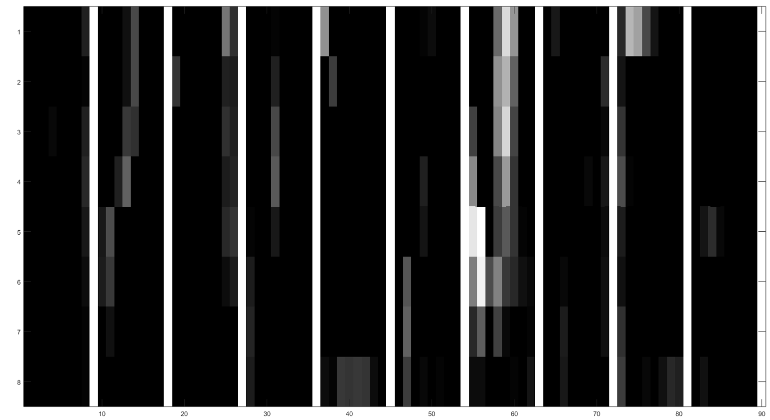


Figure 17: Layer 16 Output

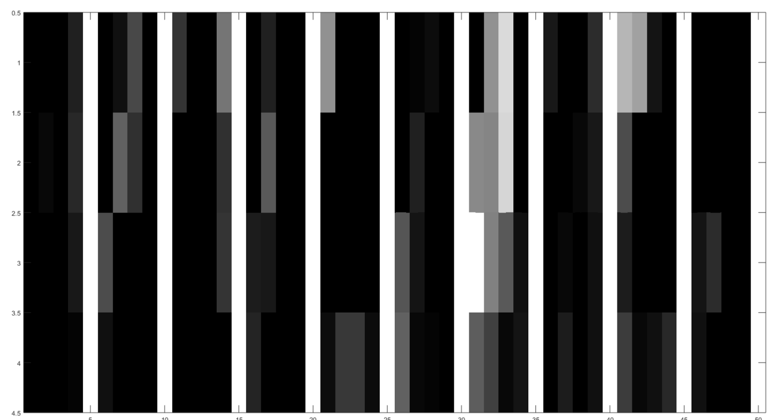


Figure 18: Layer 17 Output

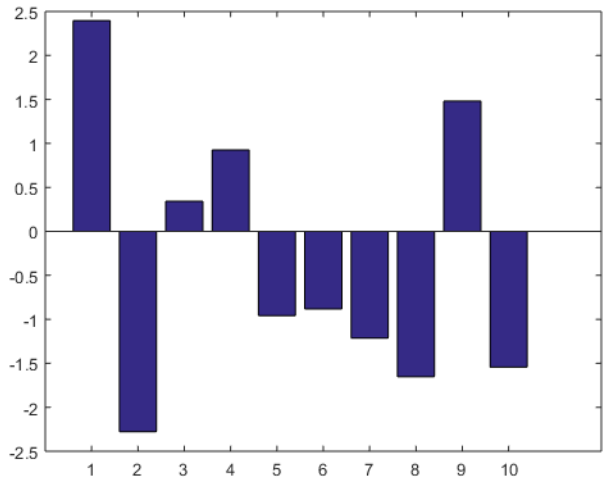


Figure 19: Layer 18 Output