Interim Technical Report

Hamish Sams

December 2018

Abstract

This report is based on comparing depths usefulness in video saliency prediction and describes results of already completed methods as well as future ideas to improve the models along with the general progress of the research project.

Contents

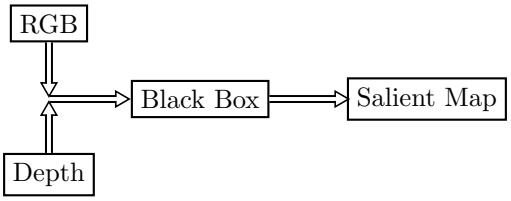


Figure 1: System Diagram

Acronyms

CNN Convolutional Neural Network. 1, 3, 5

FFN Feed-Forward Network. 7, 8, 13, 14

GAN Generative Adversarial Network. 1, 4, 5

HVS Human Visual System. 2, 3, 5

MVD Multi-view video plus depth. 3

1 Project Description

The basis of this project is mimicking the Human Visual System (HVS) for a given video sequence to predict how eye-catching(salient) each region is to create a saliency map. As a method of potentially improving saliency accuracy 3D data is used as an input as shown in Figure: 1. The complexity of this project comes from designing the black-box shown to create an accurate saliency map without the system becoming so complex and data heavy that it cannot compute within reasonable time. The project is therefore limited by "Reasonable computing power" which encompasses storage, ram, processing speed, time and power consumption. The idea of using 3D to improve the quality of saliency modelling was supplied by my supervisor, apart from that everything including how to go about solving the problem has been through my decision's. The majority of code has been written by myself except from the Generative Adversarial Networks. This project aims to see what the impact of 3D video on saliency modelling is. This technology could be implemented in compression algorithms to keep salient sections high definition. The results of these tests could be used to determine whether the extra cost of recording in 3D is worth it in terms of salience modelling. The overall effect of this upon implementation is reducing the data-flow required for similar quality video, this could make higher data-rate systems require less power and size saving money. This is unlikely to effect companies negatively as all systems are still required but on a smaller scale.

1.1 Project Specification

The project specification has changed due to the use of a different method being used to meet the project aims. This is because a neural net solution is now being used over an algorithmic approach. I felt this was a more appropriate solution as the algorithm would be mimicking a neural net. This

may lead to less understanding of what specific traits attract HVS attention, but should lead to significantly more accurate results.

1.2 Aims

The aims of the project are mainly untouched due to the wide definition. The third objective however is different due to a change in how the project is being solved. This change simply removes the limitation of spatial and temporal saliency methods being the only way to go about the project.

- Explore methods of state of the art salience modelling to predict/mimic the HVS.
- Research and develop systems of accurate video saliency.
- Review a range of salience models using data to validate model accuracy.
- Compare accuracy of saliency map prediction with and without depth.

1.3 Objectives

Unfortunately a small change in direction of the project has caused most objectives to become redundant. This however gives us an opportunity to improve and develop our objectives to be more useful.

- Construct a Neural net that can mimic the HVS by using pre-existing eye-tracking data .
- Measure the effectiveness of the neural net reaching >90% accuracy compared to eye tracking data.
- Use Multi-view video plus depth (MVD) data as a method to improve data accuracy.

2 Background Theory

Many different theory's and principles are involved in this project at a much deeper understanding than can be explained here within reason. This means many different topics may be explained to a basic level leaving a lot open to further research. Historically salience has been a psychology and biological field used to understand how the brain sees and understands images.[1] Here a range of cats were used to measure brain activity when exposed to shapes in the famous Hubel and Weisel experiment. The way the brain functioned showed a hierarchical style of vision using simple cells to view edges and rotation which fed into more and more complex cells until the brain could classify the image as shown over-simplified in Figure: 2. But as computing power has improved these same theories have been used to generate human vision like systems.

2.1 CNNs

The CNN was based directly off functioning the same way as the Hubell hierarchy works [2] the over-simplified diagram can be seen in Figure: 3. In a CNN generally an input image goes through many stages of convolution, relu and pooling before the feature maps become layers of outputs which are Gaussian connected to measure the fit of the output leading to a statistical output of how accurate the answer is.

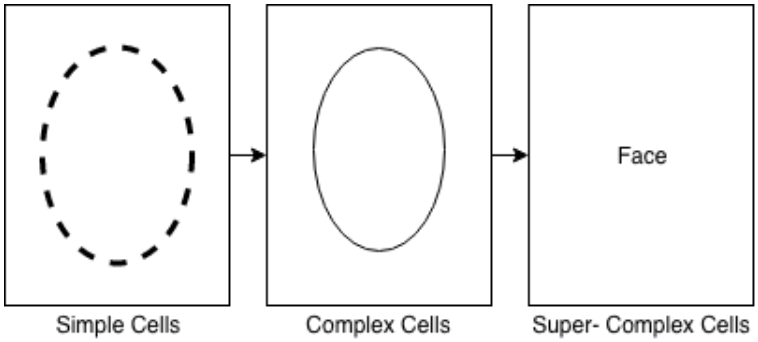


Figure 2: Hubell hierarchy

2.2 Scene analysis algorithm

Around the same time however the work of Itti and Koch [3] set a benchmark for image saliency. Here feature maps were used from the image and normalized into conspicuity maps used to make a salience map through a winner takes all system. The work here simply uses spatial characterization meaning that a single image is analyzed. Spatial methods can be used on video too but the saliency of each frame has no input based off previous of future frames. Temporal methods are the opposite where saliency is based off previous and future frames to target objects against the default flow in the frame. A bottom up approach was used which in salience means detecting what is passively eye-catching like a painting on an empty wall. The opposite of this would be what is eye catching for a specific task such as looking for cars and pedestrians while driving which is called a top-down approach.

2.3 GANs

A more modern invention is that of the GAN [4]. This is a new type of neural net made up of two separate neural nets (the generator(G) and the detective)D) where a generator reads an input to create an output. This output from the forger is then compared with what the output should be by the detective net as shown in Figure 4. The losses from the generator and the detective push against each-other until the forger is a master and the detective, despite also being a master, cannot tell the difference between the real data and the forged data. This leads to a forger trained such that it can create the output desired from the input data near identical to that of which it is trained on.

2.3.1 Image to Image GAN

Despite GANs being fairly new they are already being used frequently to solve problems previously impossible or hard to achieve. An example of this is Image to Image translation [5] ranging from training a neural net to color previously black and white images or films to making 3D maps from satellite images automatically.

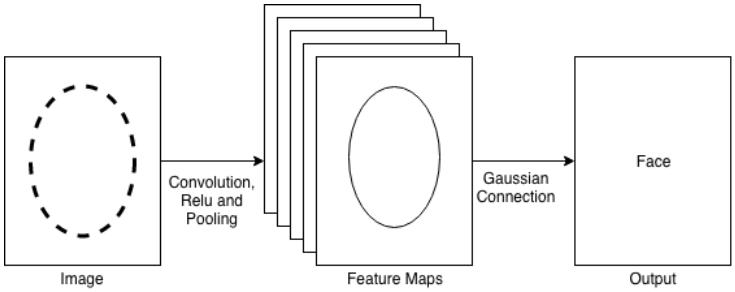


Figure 3: CNN

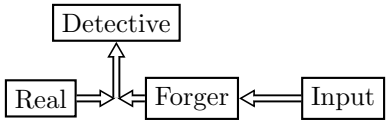


Figure 4: Generative Adverserial Network system diagram

3 Methodology

Initially to meet the original aims of the project a similar algorithmic approach as Itti and Koch [3] was to be used using bottom up pre-determined weighted feature maps but with both spatial and temporal methods to determine whether the filmed scene is more spatial or temporally orientated. To compare the algorithm with and without depth two separate algorithms must be designed, tweaked and tested which would take a long time to develop and test. As the human brain is by definition a neural net it would make sense to use a neural net to model the HVS.Initially the most basic form of neural net (feed-forward network) was to be used with 9 inputs and no hidden layers where each layer was a grayscale feature map of the original with the input pixel randomly weighted to the corresponding output pixel shown in Figure: 5. The system created attempts at the salience map which were compared to the measured eye-tracking data in the data set. The loss of the function was calculated by the mean absolute error ranging from 0-255. This system was tested and results given for runs ranging between 10 and 100 iterations. The results worked worse than desired and therefore a new method had to be proposed. This method took far too much computing power for such a basic net leading to mundane results that take far too long. As the HVS works much more like a CNN by building up from features compared to a pixel by pixel analysis. Due to the way CNNs output data a output node must be made for each pixel leading to a system that reads images like the HVS but creates images pixel by pixel and will take much longer than expected due to the huge number of outputs. Instead a GAN is used as inputs are taken like CNNs to classify features but also creates images as an output in a similar manner leading to a image easily read by the HVS. A GAN is used to learn and act as a neural net just like the HVS as well as reading and writing images similar to the HVS. As neural nets are used, large amounts of computer power are needed to teach the system especially as GANs are based around image generation. As of this, access to the university's "sharc" cluster is used to run the code. The code is written in python using the package PyTorch with Tensorflow [5]. This is the most advanced system yet proposed to deal with predicting saliency yet programmatically simple due to the re-use

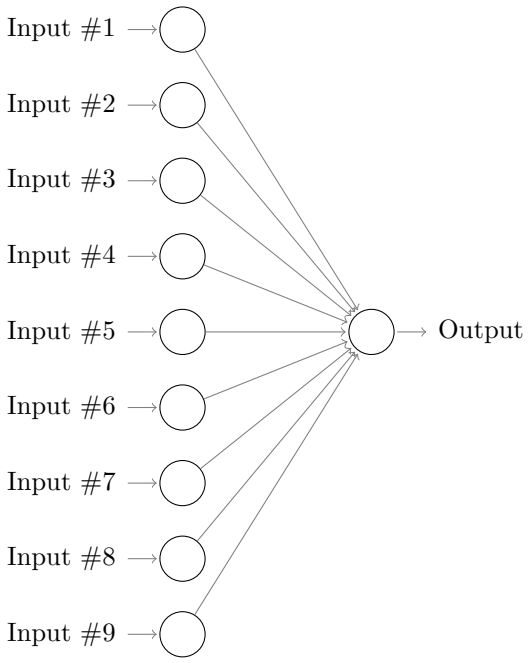


Figure 5: Proposed feed-forward neural net

of modern code needing only tweaks to re-size and join images for training. At-least two neural networks will be created, one with and one without depth to compare the saliency accuracy both being tested on the same data and number of epochs. Half of the data set [6] will be used to train the model with the other half left over to test the model. If possible with the computing power available many runs of each would be useful due to the randomness involved with neural networks. The method currently has been based loosely on a cycle where each idea is theorized, created, tested and then improved leading to more and more accurate results.

4 Results and analysis

Currently data has been taken for the feed-forward net which ran on the SHARC system four times successfully. This ran with 30 neural nets for each iteration to match the 30 CPU cores available. Each run ran a different amount of time before restarting. The longest run is shown in Figure: 6 as an error graph showing the worst, best and mean losses over the iteration count, all runs can be seen in Figures: 11,12 and 13. Over lower iterations the neural net seems to learn quickly and lowers the loss quickly as well as lowering error. Unfortunately after the first 10 iterations the loss starts to rise again slowly until around 85. This is likely due to the network training well for early frames but as soon as the frame style changes the network don't know how to deal with the changes. To test if the bad response was down to the overall design of the neural net compared to a bad run the results were superimposed in Figure 7. Here we see every run has almost identical peaks and values meaning the neural net simply isn't responding well to the problem. The lowest loss occurred in iteration 8, the saliency map for this can be seen in Figure 8. The image created

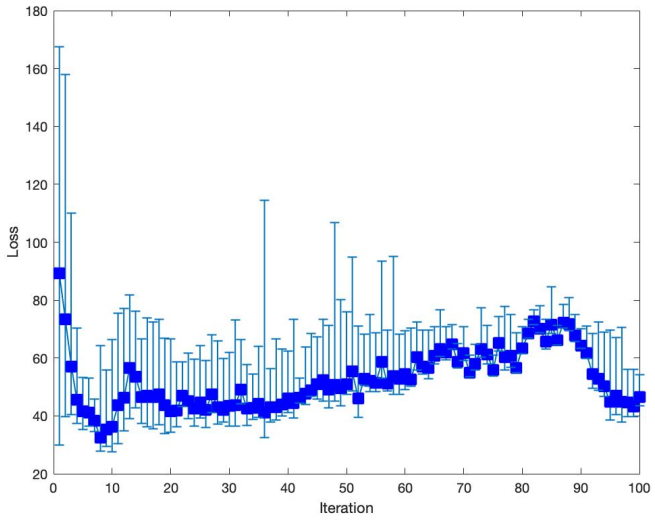


Figure 6: Longest run of the Feed-Forward Network (FFN)

		Predicted	
		1	0
Eye Tracking	1	60.03%	9.25%
	0	39.97%	90.75%

Table 1: Confusion matrix of the images in Figure: 8

resembles the measured saliency map but does not reach desired accuracy. To test more accurately how well the model fits the image a confusion matrix is used (Example in Table: 3). To make our data fit a confusion matrix both the saliency map and predicted map are forced into binary style saliency map by rounding all values to either 255 or 0 shown if Figure: 9. These maps are then compared to see how similar they are. The results are shown in Table: 1. For a high quality system the top left to bottom right diagonal should be a high percentage. Our system shows a high true negative rate meaning most non-salient areas are correctly assigned, however the true positive rate is very low at 60% accuracy. Upon inspection of the images this is based off (Figure: 9 it looks like there are two main salient regions with the prediction only picking up one leading to the almost 50/50 true positive rate.

5 Discussion and conclusion

Up to now in the project we have looked in depth at the basic pixel by pixel feed forward network. The results from this were not as good as they could have been nor were they within the >90% accuracy in the specification leading to the proposal of a new method to be researched and implemented. These results compared to existing state of the art systems [7] as well as baseline

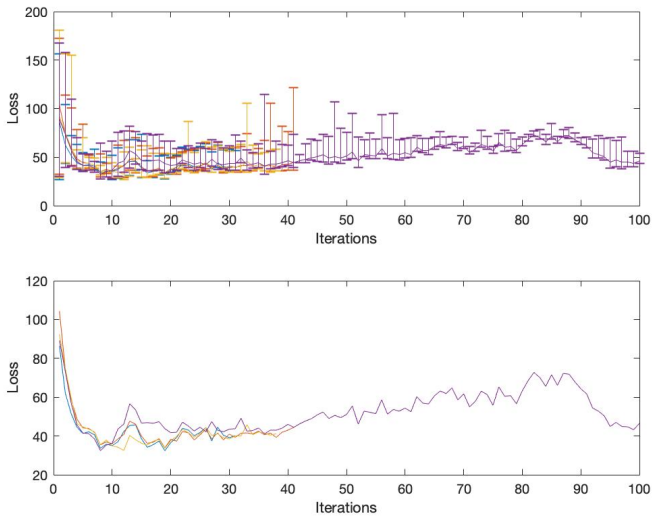
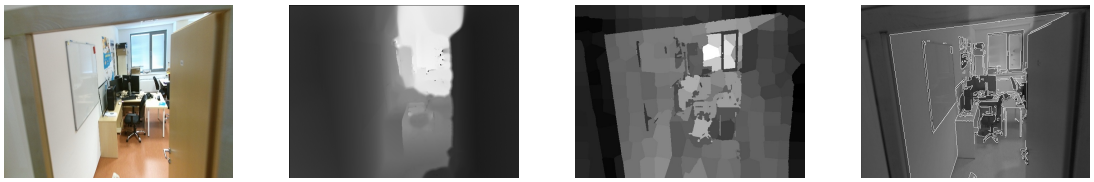
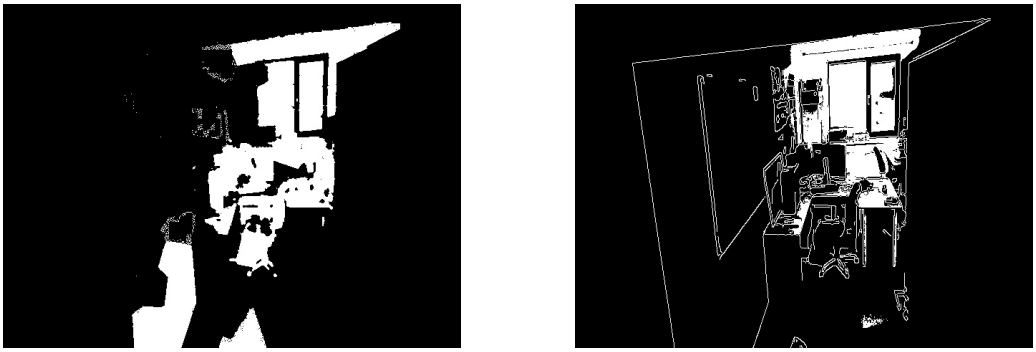


Figure 7: All mean runs of the FFNs



(a) RGB (b) Depth (c) Real Saliency (d) Predicted

Figure 8: Images of Loss 27.8 (11%) in run 1



(a) Real Saliency (b) Predicted Saliency

Figure 9: Binary saliency images

systems [3] shows this method is not use able to the same degree. Despite the results not being as accurate as possible many methods for quantifying and implementation of the project have been used allowing a better underlying understanding of the topic and analysis which should lead to a better plan, implementation and testing of future developments.

6 Milestone evaluation

Following the previous gantt chart shows I'm currently on schedule allowing for the next iteration to be again programmed in a agile fashion, tested and documented ready for the final IEEE article and symposium. The existing gantt chart has been adapted by removing previous deadlines and topic research from the future as all research has been completed and simply needs to be implemented. Funds have been easy to control as no paid hardware or software have been needed yet nor expeted to be used in the future.

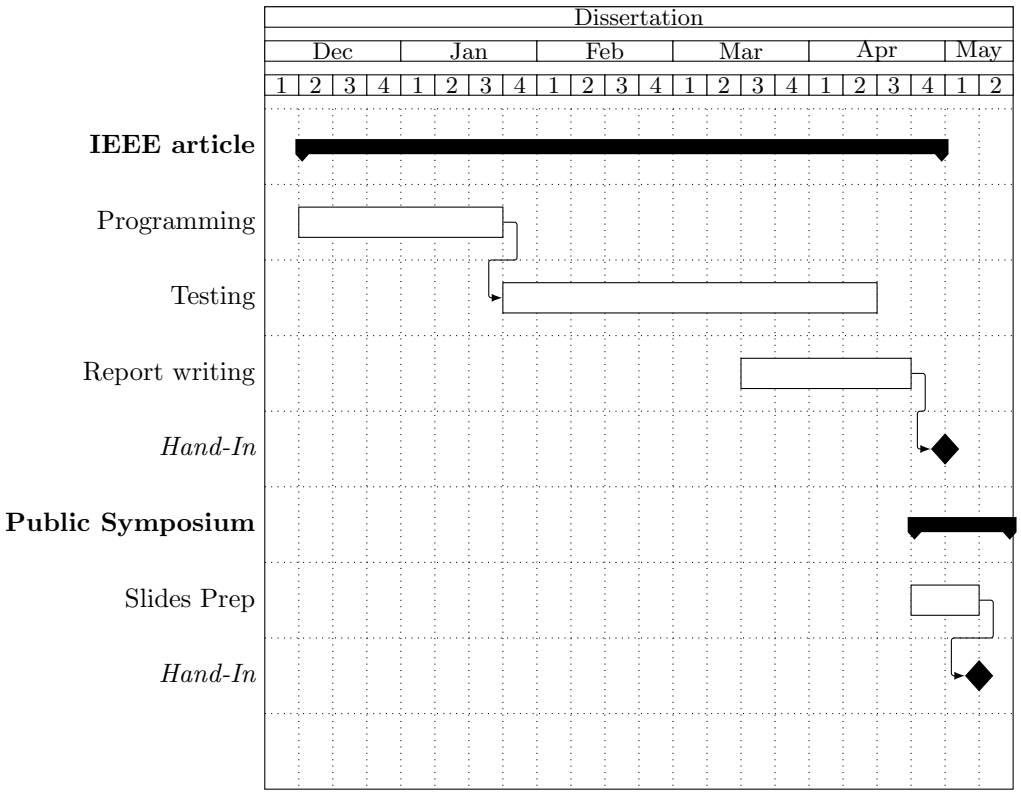


Figure 10: Gantt chart

7 References

References

[1] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[2] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, contour and grouping in computer vision*. Springer, 1999, pp. 319–345.

[3] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[6] V. Olesova, W. Benesova, and P. Polatsek, "Visual attention in egocentric field-of-view using rgb-d data," in *Ninth International Conference on Machine Vision (ICMV 2016)*, vol. 10341. International Society for Optics and Photonics, 2017, p. 103410T.

[7] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 733–740.

8 Appendix

Iteration	Run1			Run2			Run3			Run4		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
1	86.48	156.24	26.9	104.36	172.51	32.13	92.31	180.72	29.08	89.29	167.58	30.01
2	62.18	104.46	43.29	74.59	113.97	39.63	74.84	156.36	43.98	73.35	157.87	39.65
3	51.54	72.12	41.2	58.97	100.76	41.53	59.82	155.31	41.08	56.98	110.05	40.54
4	44.48	63.17	37.43	48.69	78.5	38.78	46.29	57.8	37.91	45.67	70.35	37.5
5	41.32	55.02	36.21	44.17	83.79	37.32	44.68	70	37.14	41.55	53.21	35.24
6	42.14	52.38	38.39	43.87	52.11	38.98	43.55	49.78	39.52	41.22	53.03	36.58
7	40.6	57.89	36.86	41.84	49.21	38.21	42.08	48.97	38.8	38.44	45.86	34.61
8	33.67	44.96	29.94	35.44	53.29	29.86	35.49	41.76	32.04	32.46	64.31	27.81
9	35.75	44.06	33.04	36.97	43.42	32.78	37.95	49.65	34.48	35.26	55.87	29.54
10	35.03	51.89	26.69	35.98	66.29	27.94	34.92	44.38	27.7	36.25	66.39	27.66
11	37.05	59.58	28.02	38.48	62.67	29.59	34.19	40.55	29	43.62	75.58	30.38
12	42.34	67.35	29.34	40.83	67.47	29.24	32.51	55.5	26.49	46.16	77.06	34.76
13	45.37	67.73	31.06	47.65	76.71	36.72	40.29	59.98	30.27	56.61	81.73	38.92

14	45.75	67.93	36.26	46.02	69.77	33.46	38.08	48.58	34.15	53.48	76.13	42.79
15	38.59	73.3	28.99	39.93	61.55	29.88	36.28	57.75	29.62	46.61	66.53	37.45
16	34.14	46.26	29.26	36.23	51.08	29.7	35.32	61.95	29.15	46.88	73.91	36.15
17	35.51	58.57	28.64	36.77	59.16	29.78	37.33	53.9	31.04	46.55	72.51	35.58
18	37.41	51.15	31.93	38.62	59.67	32.62	38.22	53.87	32.74	47.37	73.29	36.92
19	32.36	51.89	27.72	33.85	53.49	28.08	33.62	54.15	28.31	43.76	66.87	33.96
20	36.58	50.11	32.22	38.31	49.18	33.77	37.61	53.54	33.15	41.71	66.67	34.39
21	39.13	52.03	36.01	37.32	50.18	32.49	39.79	53.44	34.35	41.92	58.71	36.24
22	43.99	60.23	41.57	42.31	55.46	37.32	43.41	53.75	37.89	47.08	58.73	43.24
23	43.05	58.19	36.88	41.66	56.9	36.34	41.71	86.46	34.91	45.14	61.62	39.15
24	40.15	56.81	34.9	38	57.91	34.67	39.36	54.93	33.79	42.53	59.66	36.58
25	41.81	61.3	37.35	40.58	59.68	37.36	41.16	59.07	35.92	44.54	64.21	39.38
26	44.36	64.07	38.31	43.42	64.38	37.77	44.38	65.9	35.22	42.15	59.09	35.95
27	37.39	55.27	34.8	38.5	58.53	34.68	40.34	59.95	34.48	47.42	68.01	41.05
28	44.56	62.11	39.53	41.53	65.29	37.22	41.15	61.66	37.31	42.94	65.79	37.21
29	40.36	57.6	34.87	38.01	59.78	33.83	39.06	58.53	34.95	42.17	59.77	37.81
30	39.13	58.13	36.07	40.96	59.33	34.49	38.43	56.39	34.83	43.52	61.99	36.5
31	40.15	56.57	36.21	39.31	59.86	34.18	40.08	60.24	34.94	43.73	73.09	36.53
32				41.4	61.22	34.26	40	63.16	35.77	49.14	66.38	44.28
33				41.59	74.45	34.76	45.8	105.9	34.8	42.49	57.69	36.67
34				41.08	53.8	36.15	40.67	62.31	36.42	42.81	54.53	38.76
35				42.48	58.98	38.35	42.12	53.44	37.92	44.14	63.98	39.42
36				40.6	67.41	36.16	41.47	54.63	36.13	41.05	114.41	32.48
37				41.51	105.73	35.58	40.44	57.02	34.74	43.01	56.42	37.81
38				39.29	57.84	36.02	41.14	58.01	35.81	43	66.68	38.47
39				41.93	81.85	36.2				44.01	63.01	40.03
40				43.02	75.97	36.8				45.99	62.36	41.19
41				44.39	121.82	36.8				44.46	73.3	39.22
42										46.25	52.95	44.35
43										47.73	63.78	43.93
44										48.87	68.53	46.73
45										50.87	67.41	45.76
46										52.34	73.45	45.17
47										48.99	71.23	42.89
48										50.65	106.67	45.1
49										49.46	80.09	43.56
50										51.05	75.84	47.33
51										55.42	94.85	48.57
52										46.1	71.01	39.42
53										52.76	68.22	50.43
54										52.22	75.01	48.4
55										51.5	68.67	48.27
56										58.6	93.56	52.99
57										51.28	69.72	48.64
58										53.66	95.08	47.54
59										52.78	68.21	47.35

60		54.37	69.45	49.15
61		52.48	70.27	50.14
62		60.25	72.34	55.92
63		56.96	69.62	54.42
64		56.53	69.57	52.86
65		60.66	70.8	57.89
66		63.19	76.63	60.64
67		61.86	70.91	59.26
68		64.78	71.46	62.99
69		58.6	64.94	56.54
70		61.71	70.69	59.02
71		55	61.05	53.57
72		57.91	64.67	55.91
73		63.1	77.44	60.19
74		61.32	71.32	58.07
75		55.78	60.99	53.68
76		65.31	74.42	62.45
77		60.41	77.84	55.82
78		60.8	74.97	56.9
79		56.59	68.7	54.84
80		63.26	70.81	61.42
81		68.51	73.28	66.89
82		72.76	76.75	71.44
83		70.08	78.06	68.64
84		65.59	73.33	63.15
85		71.55	84.62	68.39
86		66.1	71.26	64.51
87		72.24	78.54	70.41
88		71.8	80.81	69.44
89		67.8	74.92	65.59
90		64.31	70.11	63.22
91		61.64	71.04	59.89
92		54.43	68.46	49.44
93		52.8	72.33	48.8
94		50.33	69	46.81
95		44.89	69.69	38.57
96		47	66.98	40.33
97		44.91	70.53	37.76
98		44.72	56.19	39.84
99		43.22	56.1	39.73
100		46.51	54.26	43.59

Table 2: Table of all feed forward neural net runs

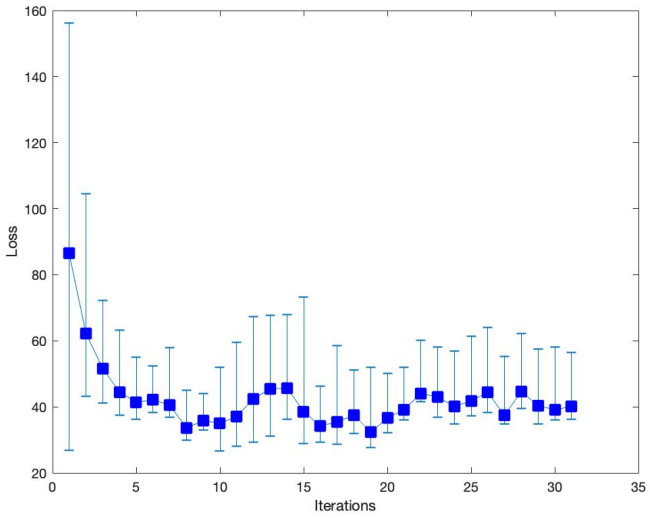


Figure 11: Run one of the FFNs

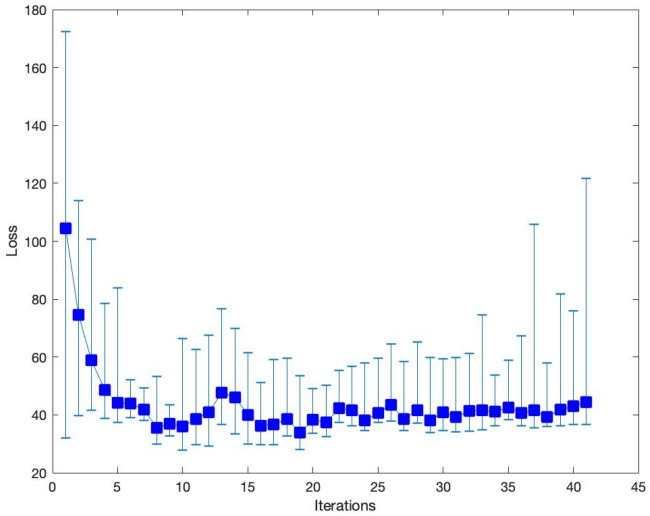


Figure 12: Run two of the FFNs

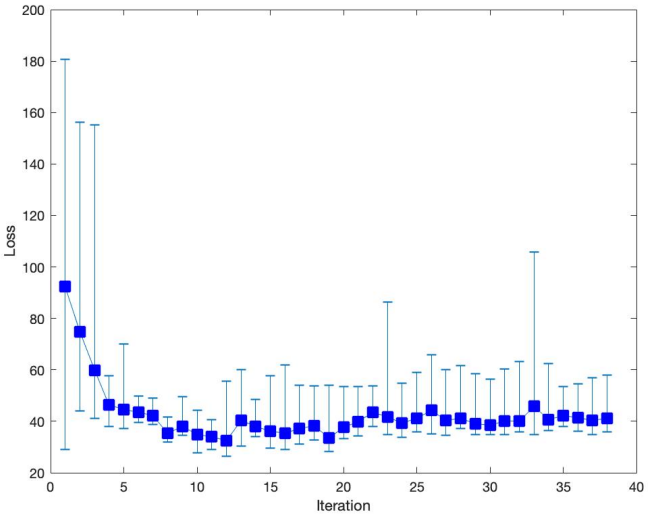


Figure 13: Run three of the FFNs

		Predicted	
		1	0
Eye Tracking	1	True Positive	False Negative
	0	False Positive	True Negative

Table 3: confusion matrix style