RBE/CS549 Computer Vision Homework 0 Report

Yijia Wu
School of Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts 01609
Email: ywu21@wpi.edu

I. Phase 1

The goal of Phase 1 is to implement a simplified Pb lite boundary detection pipeline.

A. Filter bank creation

The first step of Pb lite boundary detection is to filter the input image with a set of filters. I chose and implemented oriented Derivative of Gaussian (DoG) filters (Fig. 1), Leung-Malik filters (Fig. 2), and Gabor filters here (Fig. 3).

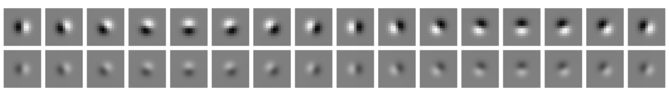
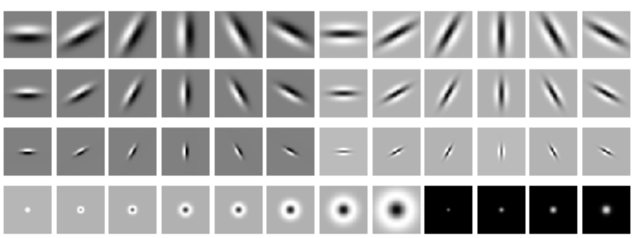
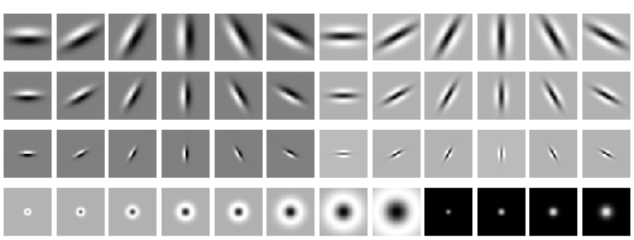


Fig. 1: DoG filter bank



(a) LM small filter



(b) LM large filters

Fig. 2: LM filters

B. Texton Map

To generate a Texton map, the first step is to combine all filter responses that are generated by convolving with each filter in the filter bank separately. Each pixel is then represented by a vector of the value of the same position in all filter responses. After that, pixel vectors in the map with all filter responses are clustered using the K-mean algorithm, which outputs the Texton map. The value of each pixel in the Texton map is the id number of the vector category that the same pixel is corresponding to in the vector map. Ten example raw images and Texton map of them are shown in Fig. 4 and Fig. 5.

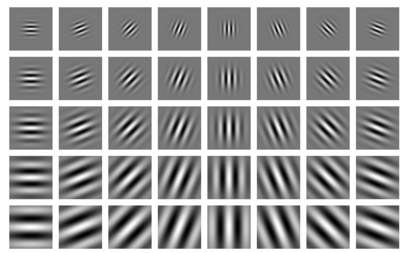


Fig. 3: Gabor filter bank

C. Brightness Map and Color Map

The initial maps of the Brightness map and the Color map are the grayscale of the input image and the original color image. The next step is the same as the clustering step when generating the Texton map. The Brightness and Color map of ten example images are shown in Fig. 6 and Fig. 7.

D. Texture, Brightness and Color Gradients

To obtain the gradients of the Texton, Brightness, and Color map, which shows the boundary between different clusters, half-disc masks are used to speed up the boundary detection. The generated half-disc mask bank is shown in Fig. 8. By convolving the map with a half-disc mask the mirrored version of it, we can compute the chi-square distance of each pixel. The sum of these distances are the Texture, Brightness, and Color Gradients.

E. Sobel and Canny baselines

Sobel and canny filters are two common filters that can be used for boundary detection. The sobel and canny filter responses are shown in Fig. 12 and Fig. 13 as the baseline of the final result.

Fig. 4: Initial image

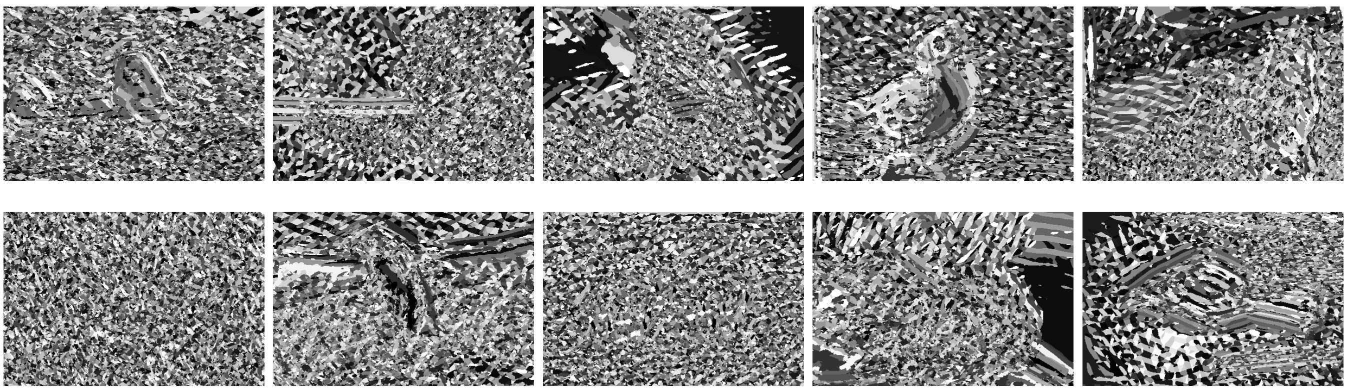


Fig. 5: Texton Map

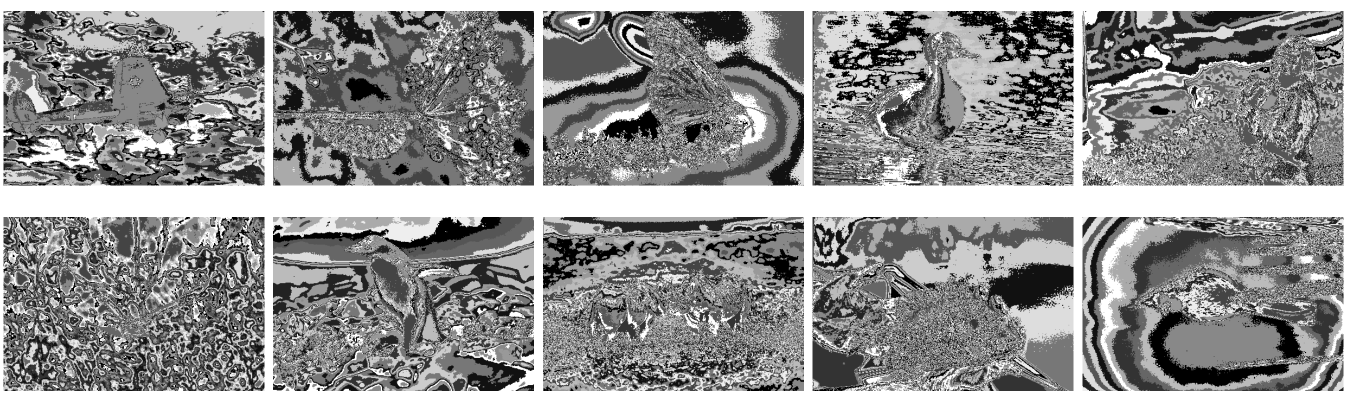


Fig. 6: Brightness Map

F. *Pb-Lite output*

The last step is the combination of the Texture, Brightness and Color Gradients, and sobel, canny baselines. The weight is chosen as 0.5. The final output is shown in Fig. 14

G. *Discussion and Conclusion*

The implemented Pb-lite output contains more boundaries than the one that is shown in the ground truth or the sobel and canny baseline. It is likely because of the filter parameter and texton map cluster number difference. Though it is still not good enough, this implementation is still better than two baselines as most boundaries are continuous and clear.

There are many reasons why the performances are improved. Compared with the sobel and canny baselines, the pb-lite method makes use of much more information. Unlike two filters which can only capture local information, the brightness and color map provides a global view of the image. The combination of multiple texture filters also helps with capturing the gradient information from a larger scale.

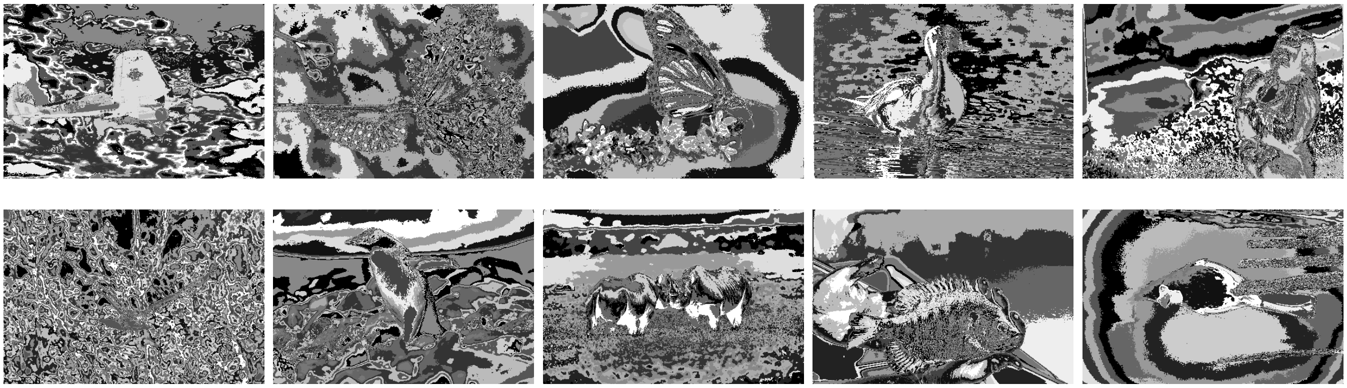


Fig. 7: Color Map

Fig. 8: HD mask

II. Phase 2

A. First neural network

The first neural network mainly contains three conv2d layers and three linear layers as shown in Fig. 18a. The optimizer is chosen as adam with a 0.001 learning rate. The batch size is 128. The number of parameters is 4937. The confusion matrices on the train data and test data are shown in Table. I and II.

B. Improved neural network

To improve classification accuracy, the size of each layer are doubled as shown in Fig. 18b. Data normalization, augmentation (rotate random angle within [-30, 30] degree or flip horizontally) and decaying learning rate are also added. The number of parameter of the new model is 9833. Though the training accuracy seems similar, the testing accuracy increases by 3%.

C. ResNet

A 34-layer ResNet is implemented by modifying the ResNet source code because of the time limit. The number of parameters is 33827.

TABLE I: Confusion matrix of the first neuron network on train data

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	4779	19	37	11	21	6	6	3	94	24
2	11	4918	0	3	0	1	5	0	11	51
3	127	10	4550	79	104	37	56	7	19	11
4	59	12	134	4159	64	292	190	18	35	37
5	33	8	155	81	4556	38	73	28	15	13
6	21	11	114	294	99	4273	109	43	16	20
7	13	15	115	45	24	16	4745	0	17	10
8	39	8	56	73	80	53	6	4658	6	21
9	68	51	8	9	4	1	7	0	4842	10
10	73	86	7	9	3	4	3	1	24	4790

TABLE II: Confusion matrix of the first neuron network on test data

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	817	20	24	20	19	5	8	6	50	31
2	18	884	4	5	4	0	6	2	21	56
3	73	6	653	61	71	48	53	15	11	9
4	35	12	84	538	51	150	79	16	9	26
5	24	5	97	60	677	29	53	40	10	5
6	23	5	53	155	35	637	42	34	5	11
7	11	11	47	37	29	15	830	4	12	4
8	24	5	39	45	82	38	0	741	6	20
9	52	41	5	7	12	3	3	2	853	22
10	50	83	6	10	0	5	5	6	19	816

TABLE III: Confusion matrix of the improved neuron network on train data

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	4986	1	5	0	1	0	1	1	4	1
2	1	4982	1	0	1	0	0	0	0	15
3	12	0	4946	7	15	8	8	3	1	0
4	1	0	11	4917	20	36	8	4	2	1
5	0	0	7	10	4965	4	7	7	0	0
6	1	0	4	59	14	4909	3	8	2	0
7	0	0	9	10	6	2	4971	0	0	2
8	0	0	1	11	5	9	0	4973	0	1
9	5	2	0	1	0	0	0	0	4991	1
10	0	7	1	1	0	1	1	1	2	4986

TABLE IV: Confusion matrix of the improved neuron network on test data

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	832	12	38	21	12	6	5	10	38	26
2	48	2	713	52	53	42	58	19	8	5
3	14	4	62	610	44	160	49	35	13	9
4	9	3	61	46	750	47	34	44	4	2
5	11	4	23	176	42	671	17	41	1	14
6	7	4	56	52	33	18	821	1	4	4
7	8	1	18	48	50	51	2	815	2	5
8	49	25	7	8	5	8	8	5	868	17
9	21	65	9	14	1	4	4	10	23	849

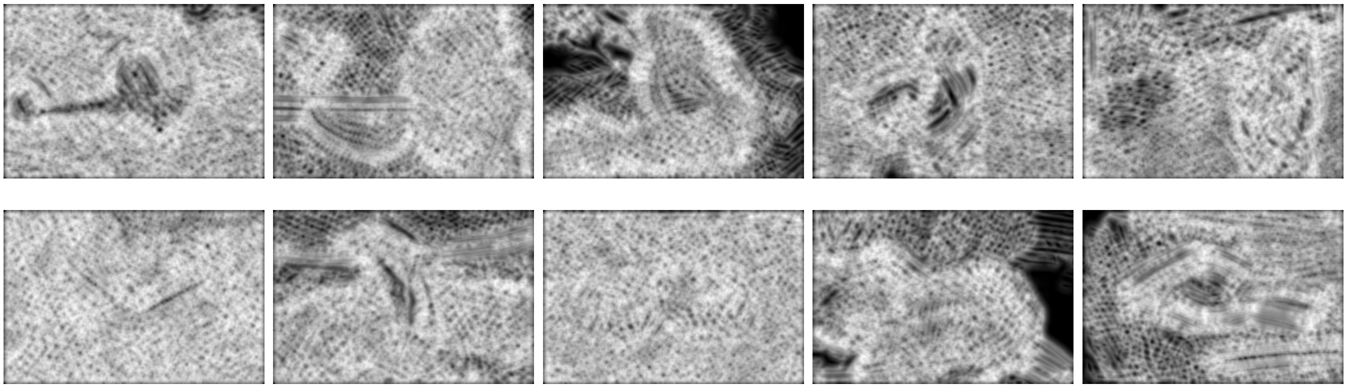


Fig. 9: Tg Map

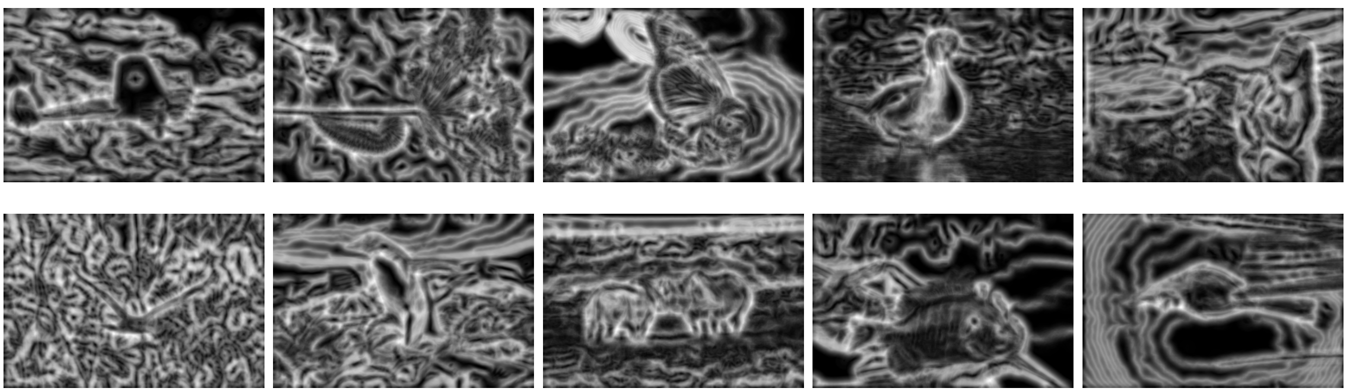


Fig. 10: Bg Map

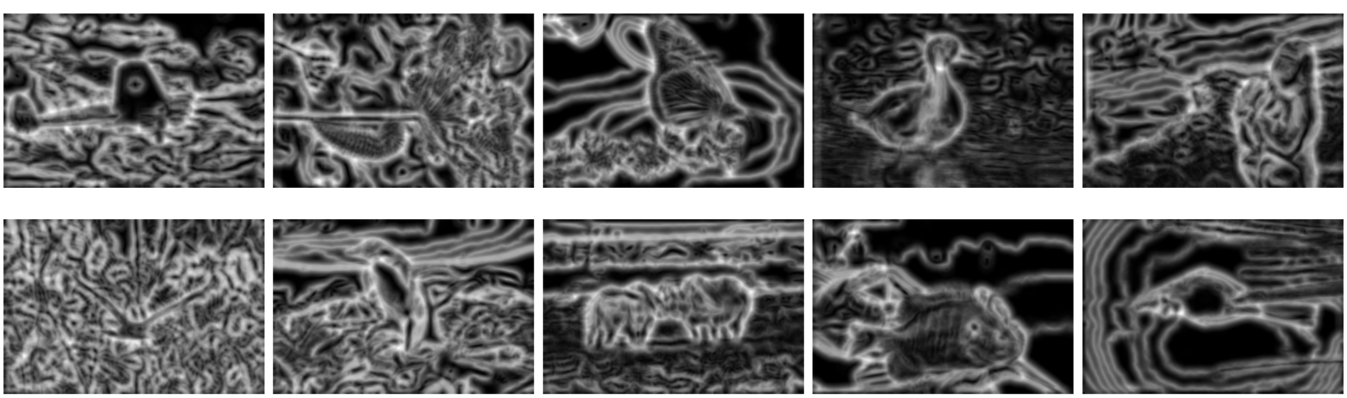


Fig. 11: Cg Map

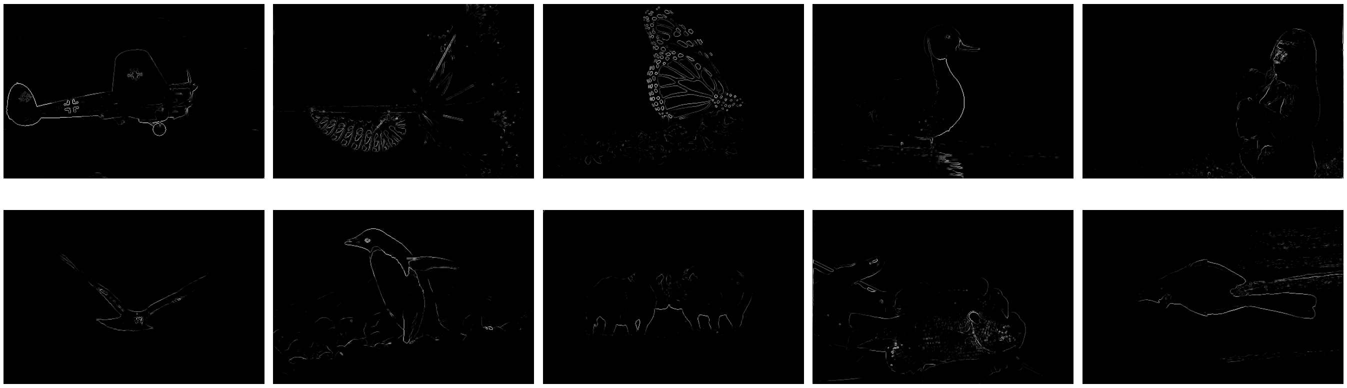


Fig. 12: Sobel Baseline

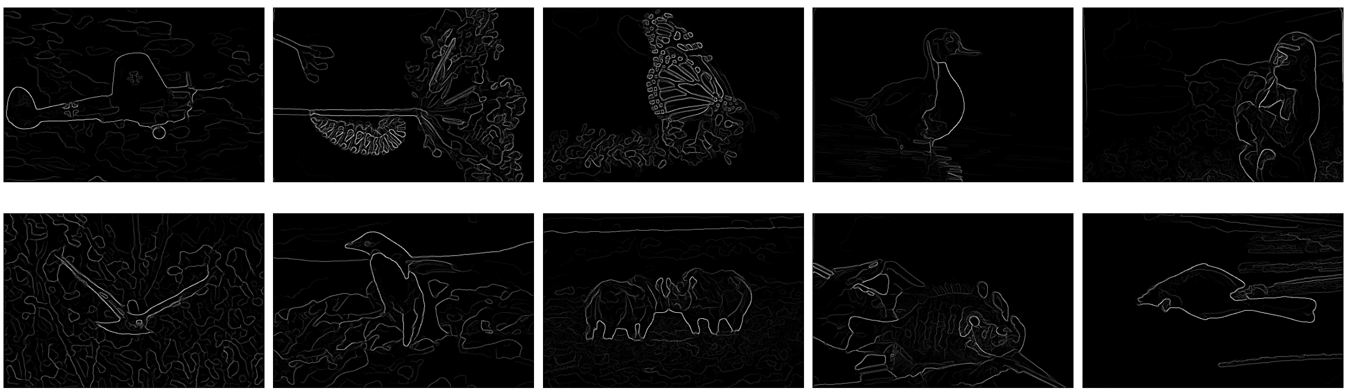


Fig. 13: Canny Baseline

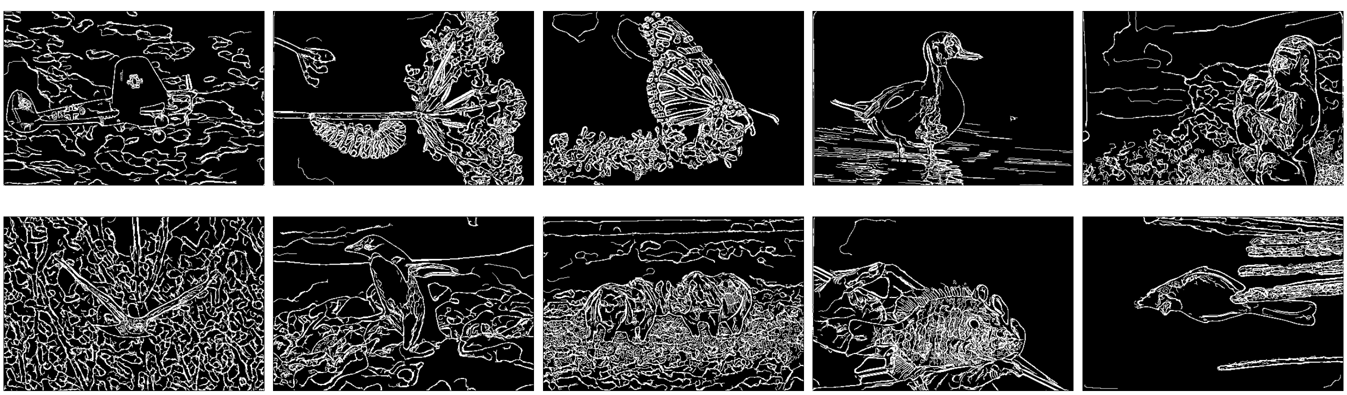


Fig. 14: PbLite

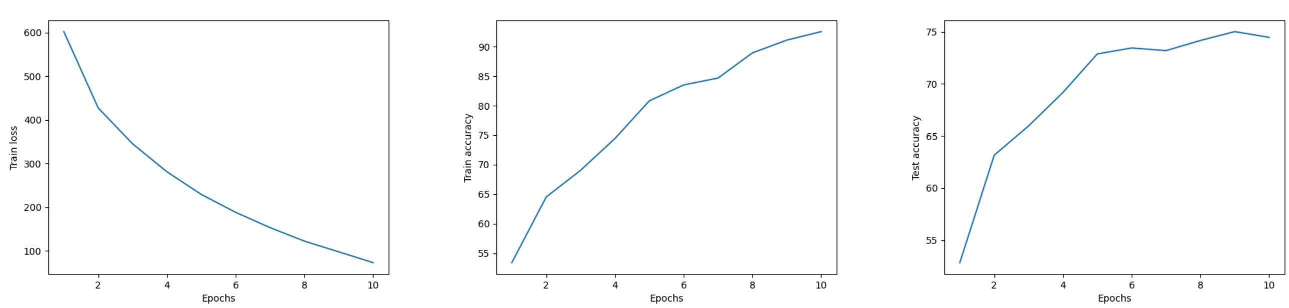


Fig. 15: First neuron network performance

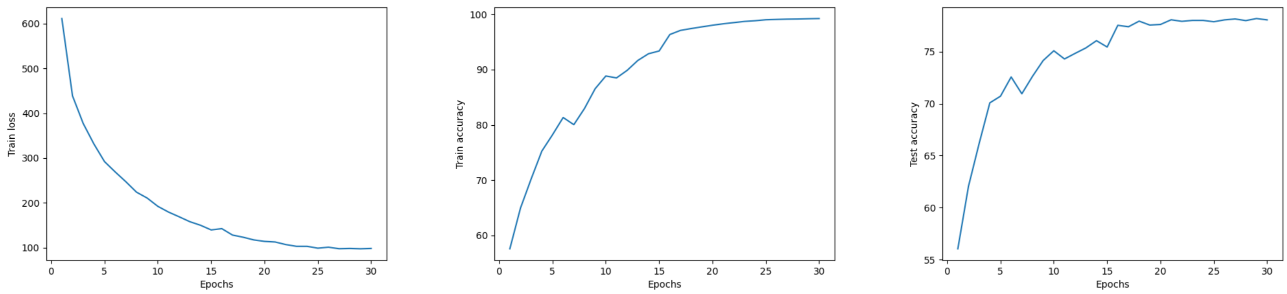


Fig. 16: Improved neuron network performance

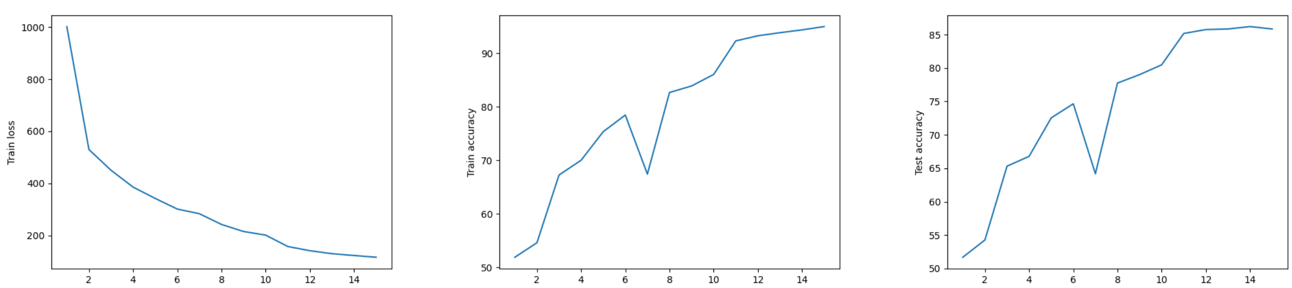
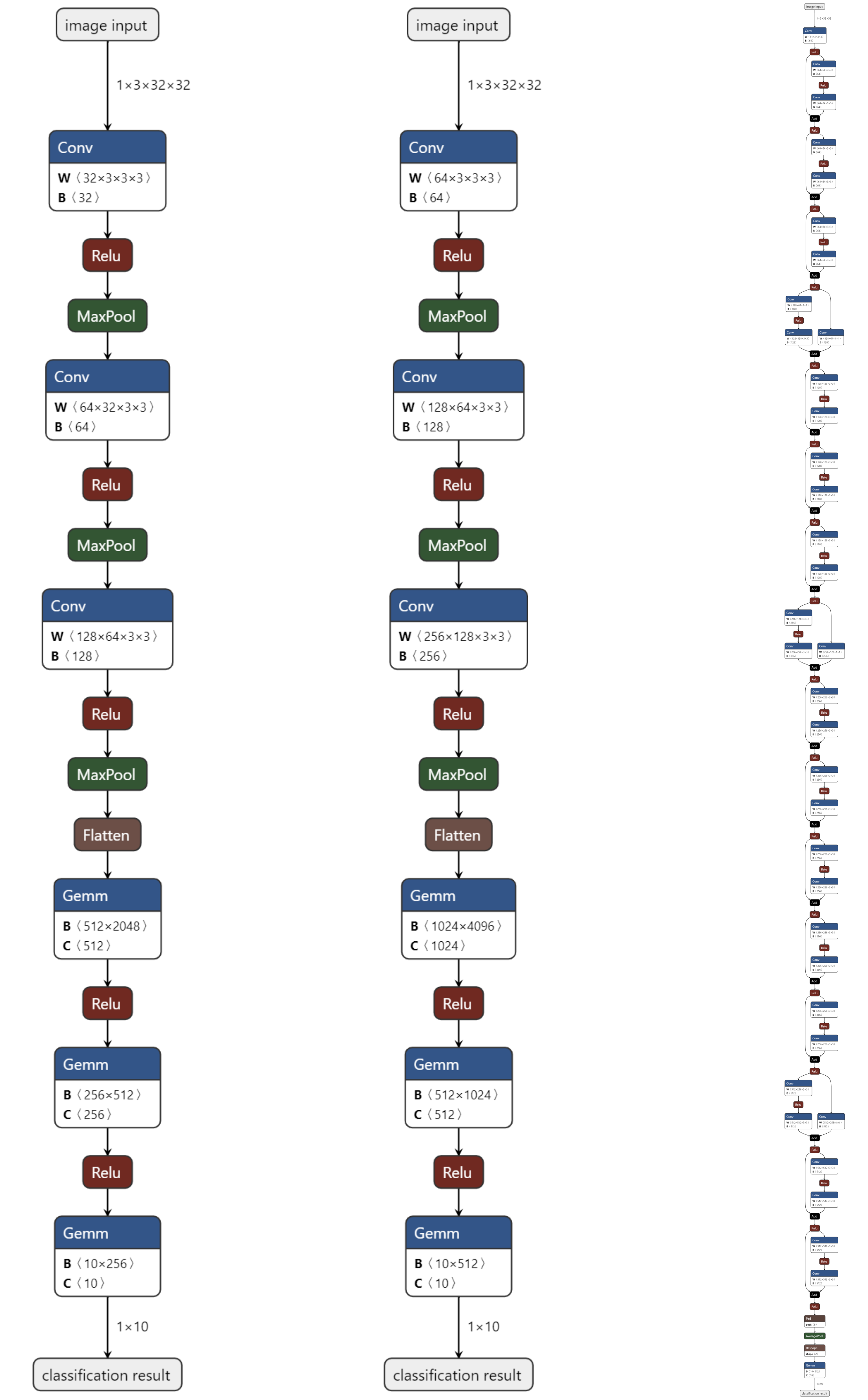


Fig. 17: ResNet performance



(a) First neuron network (b) Improved neuron network (c) ResNet34

Fig. 18: Neuron network architectures