Assignment 1

---

1 Part A

Best Parameters

The experiments performed are shown in tabular form :-
From the results below the best parameters are 4 convolution layers and 3 fully connected layers with kernel size 3 x 3 for 50 epochs and stride of 1 for convolution layers and stride of 2 for max pooling layers.
The model is below as follows :-

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()

        self.conv1 = nn.Conv2d(in_channels=3, out_channels=64, kernel_size=kernel_s)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)
        self.conv2 = nn.Conv2d(in_channels=64, out_channels=128, kernel_size=kernel_s)
        self.conv3 = nn.Conv2d(in_channels=128, out_channels=256, kernel_size=kernel_s)
        self.conv4 = nn.Conv2d(in_channels=256, out_channels=512, kernel_size=kernel_s)
        self.fc1 = nn.Linear(in_features=512, out_features=256)
        self.fc2 = nn.Linear(in_features=256, out_features=128)
        self.fc3 = nn.Linear(in_features=128, out_features=33)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = self.pool(F.relu(self.conv3(x)))
        x = self.pool(F.relu(self.conv4(x)))
        x = F.avg_pool2d(x, kernel_size=x.shape[2:])
        x = x.view(x.shape[0], -1)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

Table 1: Validation accuracy For Kernel of size 3 x 3.

Epoch	2 cov 2 fc	2 conv 3 fc	3 conv 3 fc	3 conv 4 fc	4 conv 3 fc
5	22%	17%	18%	15%	19%
10	26%	25%	28%	23%	27%
15	32%	30%	37%	38%	40%
20	35%	34%	40%	41%	48%
25	37%	36%	42%	48%	51%
30	39%	37%	44%	52%	51%
35	40%	41%	49%	51%	52%
40	40%	42%	52%	53%	55%
45	44%	42%	53%	52%	57%
50	45%	43%	54%	53%	59%
55	45%	44%	55%		59%
60	45%	45%	56%		58%
65	46%	44%	55%		51%
70	45%	47%	54%		56%
75	48%	46%	53%		54%
80	48%	47%	55%		59%
85	48%	48%	53%		59%
90	49%	48%	53%		59%
95	48%	48%	56%		52%
100	50%	48%	56%		58%

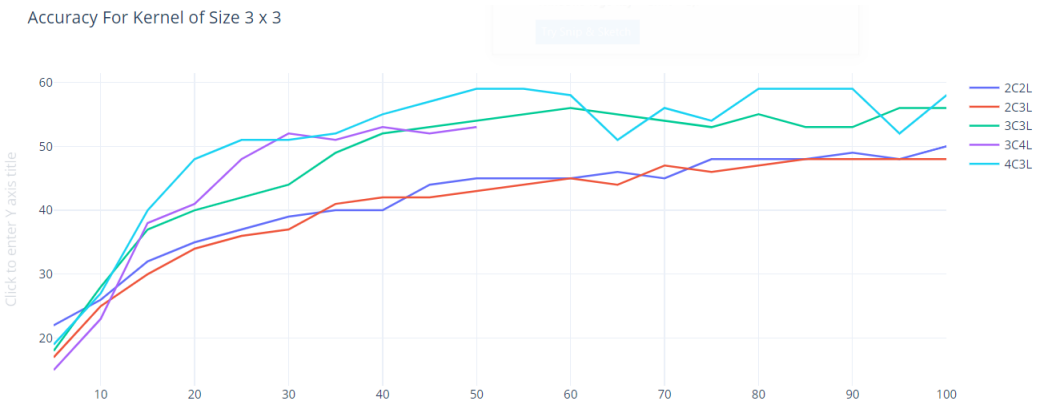


Figure 1: Validation accuracy for kernel size 3*3. (Here C-convolutional L-fully connected).

Table 2: Validation accuracy For Kernel of size 5 x 5.

Epoch	**2 cov 2 fc**	**2 conv 3 fc**	**3 conv 3 fc**	**3 conv 4 fc**	**4 conv 3 fc**
5	21%	20%	24%	19%	30%
10	30%	27%	34%	28%	40%
15	36%	33%	44%	35%	40%
20	40%	36%	48%	45%	40%
25	40%	38%	52%	50%	41%
30	44%	43%	53%	49%	43%
35	46%	44%	54%	53%	41%
40	46%	45%	56%	52%	46%
45	47%	47%	57%	52%	47%
50	49%	48%	56%	52%	47%
55	48%	49%	56%		
60	48%	51%	55%		
65	50%	49%	59%		
70	51%	50%	60%		
75	52%	50%	57%		
80	52%	51%	56%		
85	51%	53%	54%		
90	50%	52%	57%		
95	52%	52%	60%		
100	52%	52%	51%		

Table 3: Validation accuracy For Kernel of size 5 x 5.

Experiment	**Accuracy for 3x3 kernel**	**Accuracy for 5x5 kernel**
2 cov 2 fc	50%	53%
2 conv 3 fc	50%	52%
3 conv 3 fc	56%	50%
3 conv 4 fc	51%	54%
4 conv 3 fc	56%	46%

Table 4: Experiments for strides and max pooling layer

	Experiment 1			Experiment 2			Experiment 3	
	Input channels	**Output Channels**		**Input channels**	**Output Channels**		**Input channels**	**Output Channel**
Conv1	3	64		3	64		3	64
Conv2	64	128		64	128		64	128
Conv3	128	256		128	128		128	128
Conv4	256	512		128	256		128	256
FC1	512	256		256	128		256	128
FC2	256	128		128	128		128	128
FC3	128	33		128	33		128	33
Conv1 stride	1	1		1	1		2	2
Conv2 stride	1	1		1	1		2	2
Conv3 stride	1	1		1	1		1	1
Conv4 stride	1	1		1	1		–	–
Max pool Kernel size	2	2		3	3		2	2
Val accuracy after 50 epochs	57%			51%			50%	

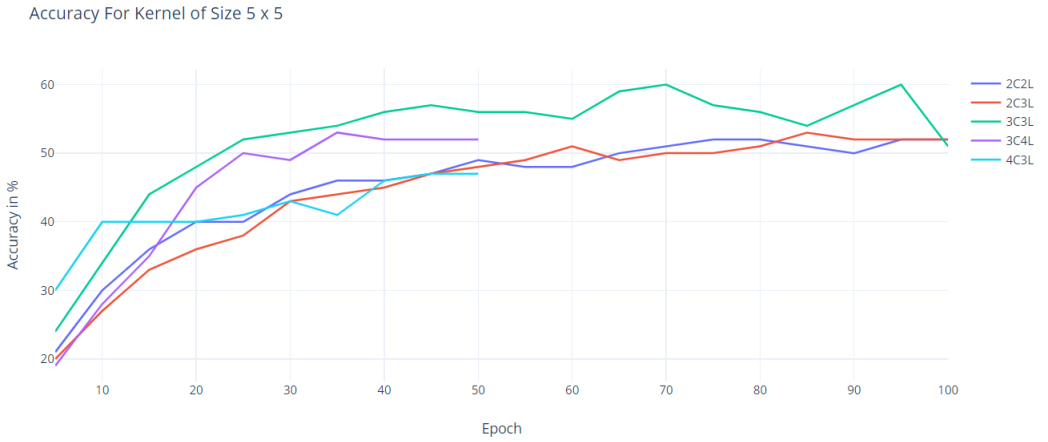
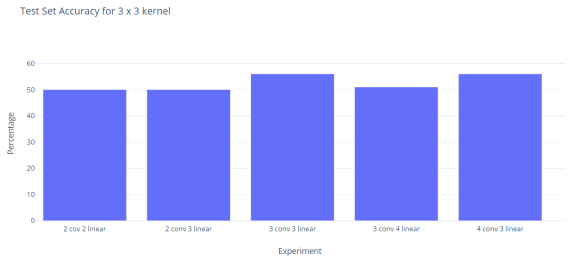
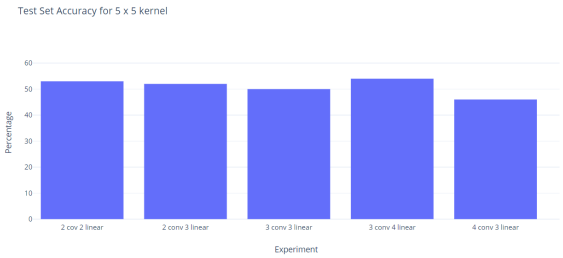


Figure 2: Validation accuracy for kernel size 5*5. (Here C-convolutional L-fully connected).



(a) 3 x 3 kernel



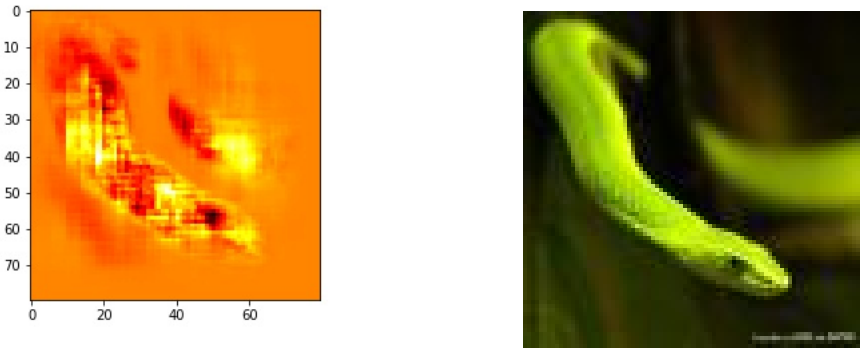
(b) 5 x 5 kernel

Figure 3: Test Accuracy for different experiments

2 Part B

2.1 Occlusion sensitivity experiment

For this experiment, a function was written to mask the image with window of size 5x5. So a 5x5 patch was slided on the displayed images and the corresponding confidence values were plotted as heatmap. The black colour is less confidence level and increases as the colour becomes yellow colour which means more confidence level.



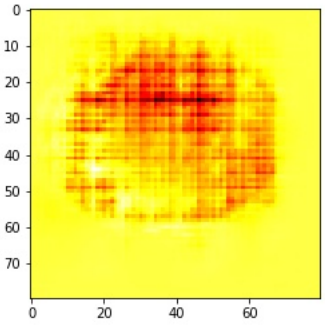
(a) Output (b) original

Figure 4: Green Mamba

(a) Output (b) original

Figure 5: Green Mamba

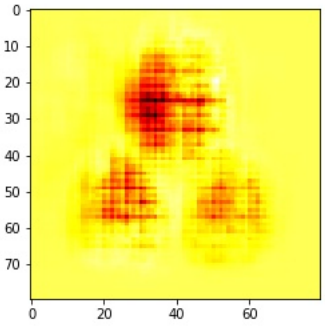
As we can see in the Figure 4 and 5, if the window is on the body of the snake especially near the head, there is a drastic drop in the confidence level of the prediction for that class.



(a) Output

(b) original

Figure 6: Orange

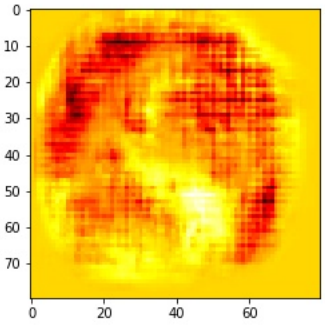


(a) Output

(b) original

Figure 7: Orange

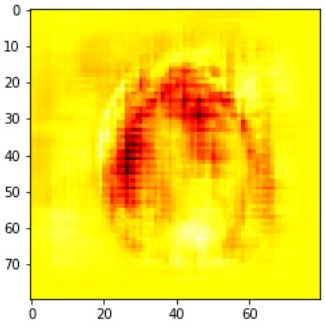
For Figure 6 and 7, we can see that when the window is on the body of orange, there is a drastic drop in the confidence levels of the predictions. The neural network is able to identify the three oranges in figure 7.



(a) Output

(b) original

Figure 8: Consomme

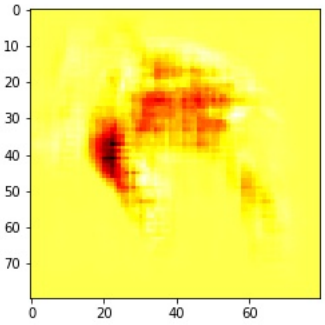


(a) Output

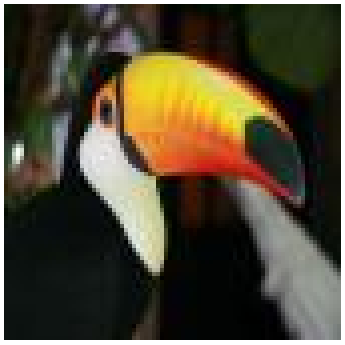
(b) original

Figure 9: Consomme

For figures 8 and 9, we can see that neural network is able to identify the circular portions of the bowl.



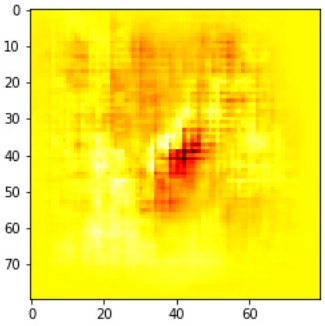
(a) Output



(b) original

Figure 10: Toucan

In figure 10, the confidence level drops drastically when the window is on the face of the bird. It seems that the neural network is able to identify the white patch on the neck and the orange beak of the bird.

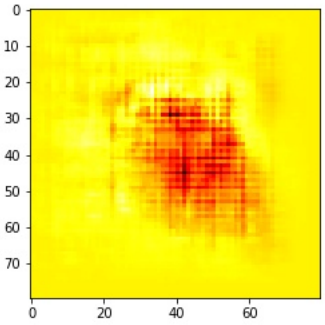


(a) Output

(b) original

Figure 11: Ladybug

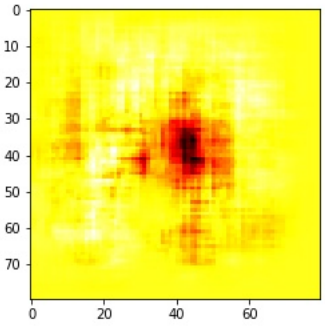
In figure 11, the neural network is not clearly able to identify the ladybug but there are some confidence level drop around its body.



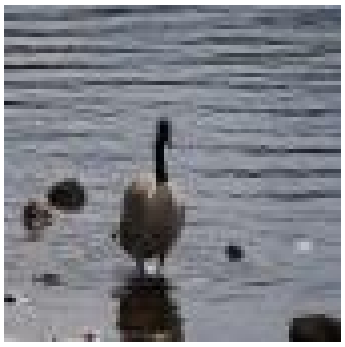
(a) Output

(b) original

Figure 12: Poodle



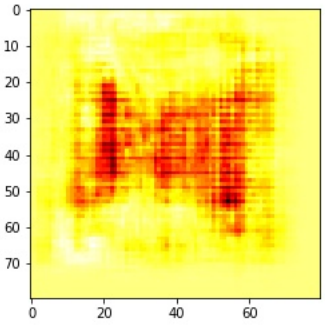
(a) Output



(b) original

Figure 13: Goose

For figure 12 and 13, the neural network is ably to identify the part where the primary object is located. There are drastic drops in confidence levels around the body of the poodle and goose.



(a) Output

(b) original

Figure 14: Vase

In figure 14, the neural network is not clearly able to identify the vase but there are significant confidence level drops around the body of the vase.

2.2 Filter Analysis

2.2.1 Filter Identification

Below are the image patches of the maximum responses of the filters in the corresponding layers. As the kernel size is 3x3, the layer one image patch will be 3x3. But due to max pooling in the subsequent layers, the size of the patches will increase.

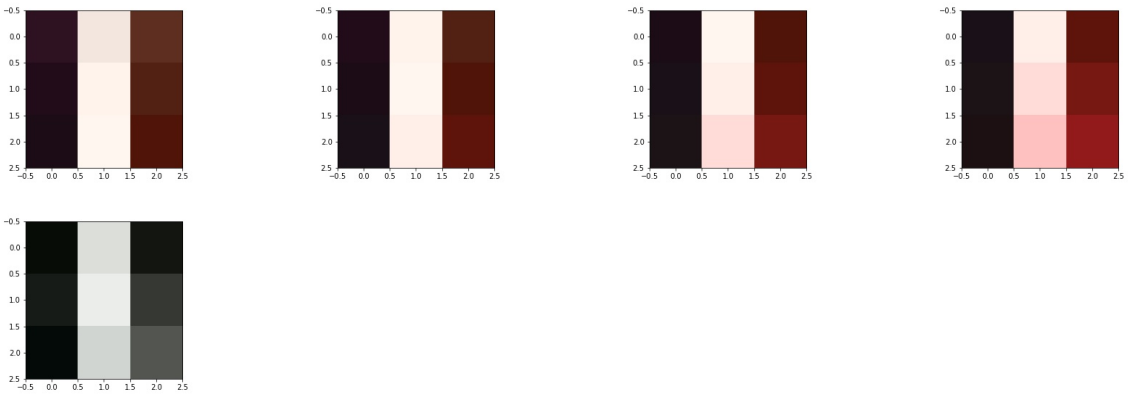


Figure 15: Layer 1 filter number 0

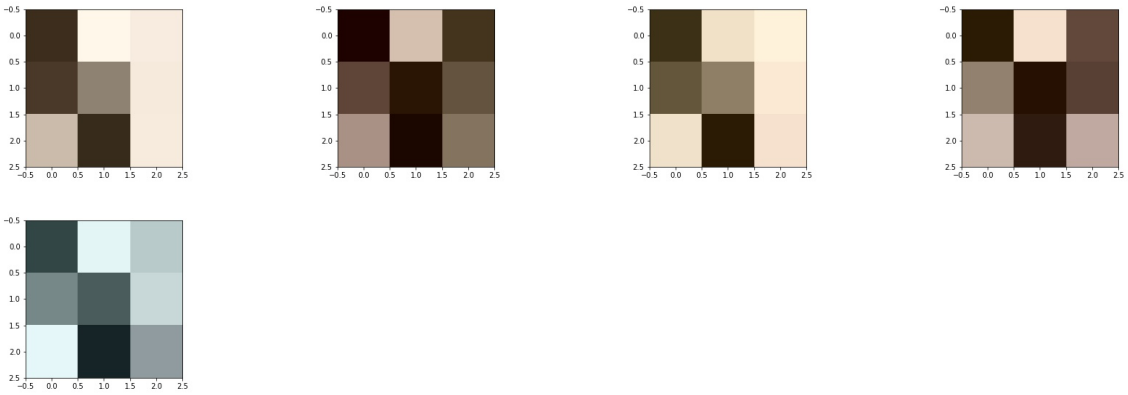


Figure 16: Layer 1 filter number 1

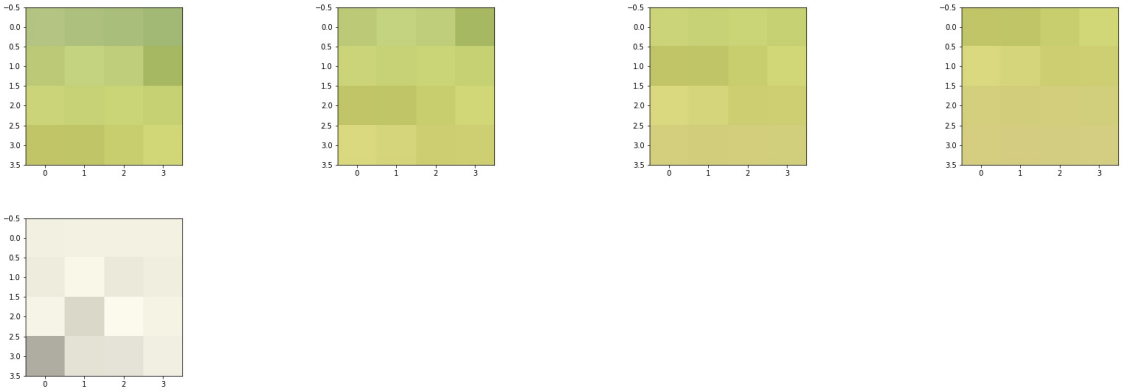


Figure 17: Layer 2 filter number 70

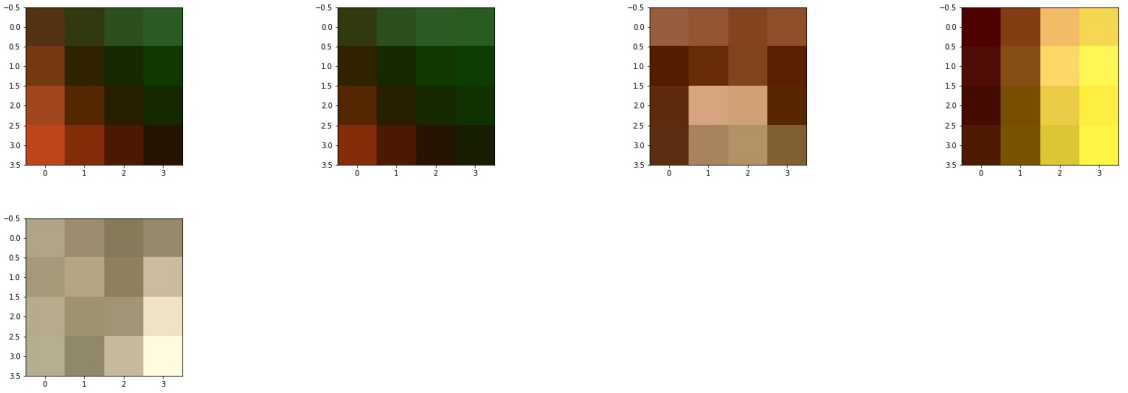


Figure 18: Layer 2 filter number 71

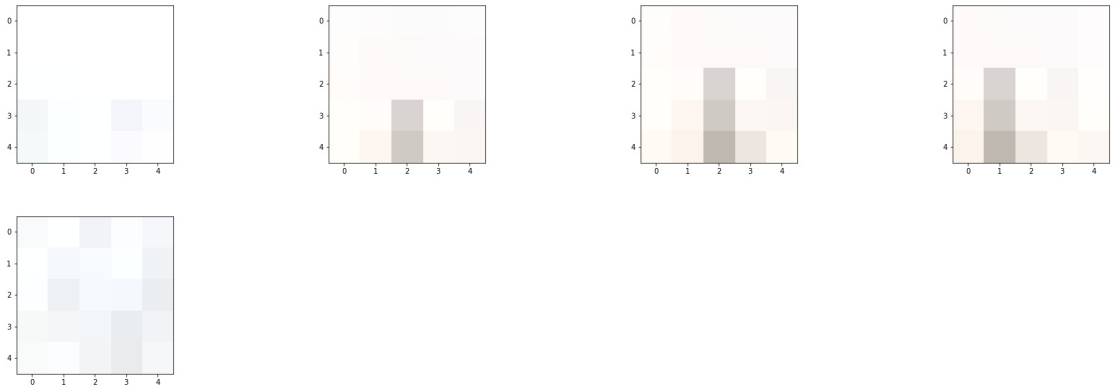


Figure 19: Layer 3 filter number 100

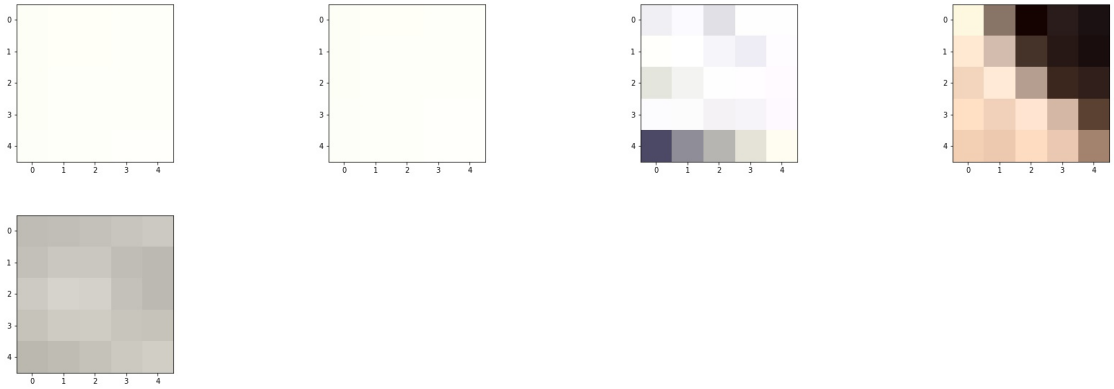


Figure 20: Layer 3 filter number 101

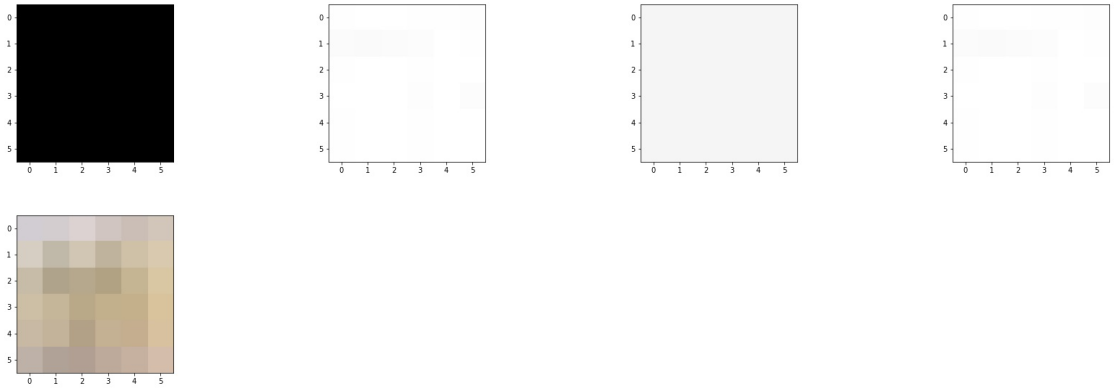


Figure 21: Layer 4 filter number 50

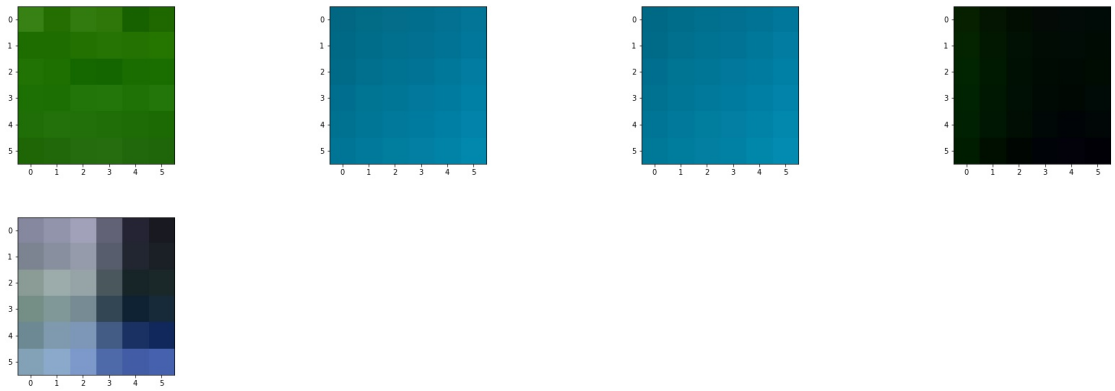


Figure 22: Layer 4 filter number 51

2.2.2 Filter Modifcation

Table 5: Validation accuracy For Kernel of size 5 x 5.

Class	Images misclassified after filter modification
african_hunting_dog	1
ant	3
ashcan	1
black_footed_ferret	0
bookshop	0
carousel	1
catamaran	1
cocktail_shaker	1
combination_lock	5
consomme	3
coral_reef	0
dalmatian	3
dishrag	0
fire_screen	1
goose	1
green_mamba	0
king_crab	2
ladybug	0
lion	1
lipstick	0
miniature_poodle	5
orange	0
organ	1
parallel_bars	0
photocopier	2
rhinoceros_beetle	2
slot	0
snorkel	0
spider_web	2
toucan	2
triceratops	2
unicycle	2
vase	6

As we see in the table, after the filters[0,1] in convolution layer 1, filters[70,71] in convolution layer 2, filters[100, 101] in convolution layer 3,filters[50,51] in convolution layer 4 were set to zero, the following images began to be misclassified.
The classes vase, miniature poodle and combination lock had the most misclassifications.

The last Image in Figure no. 16 corresponds to a patch in image of a vase. Since this was set to zero, we see 6 misclassifications in class vase.
Teh last Image in Figure no. 15 corresponds to a patch in image of a combination lock. As this was also set to zero, there are 5 misclassifications in the class combination lock.
There are three misclassifications in consomme class as the kernel corresponding to last images in figure no. 18 and Figure no. 21 may be an important factor in classifying consomme class as the patchs corresponds to images in consomme class.
There are 2 misclassification in King crab class with blue water in background. This may be due to the blue patches turned off in Figure 22.