CS6910 Fundamentals of Deep Learning Assignment 1 Analysis Report

ME18B182 Vasudev Gupta

October 11, 2020

Contents

1 Training Convolutional Neural Network

All the experiments presented in this section are done with a 3 CNN-layered, 2 fully-connected layered network for 10 epochs only (unless change is mentioned in the subsections) because of computational constraints. Later, final model is trained for large no of epochs. While measuring performance due to one of the parameter, rest of parameters are kept fixed.

1.1 Number of filters

As number of filters are increased beyond certain limit, there are more chances that model will learn just redundant features and hence model performance won't increase further on validation data. But if number of features are too less, then model will underfit on training data because it wouldn't be able to capture important features.

Table-1		
No of filters in 1st layer	Train Accuracy	Val Accuracy
768	0.393	0.2627
512	0.3589	0.2503
256	0.3253	0.2646
128	0.3197	0.2543
64	0.2985	0.2399
32	0.2712	0.2282
16	0.2502	0.209

Table-2		
No of filters in layer-2	Train Accuracy	Val Accuracy
768	0.3164	0.2573
512	0.3288	0.2571
256	0.308	0.2637
128	0.3179	0.2523
64	0.3003	0.2481
32	0.2939	0.2239
16	0.3197	0.2543

Table-3		
No of filters in layer-3	Train Accuracy	Val Accuracy
128	0.3798	0.2801
64	0.3865	0.2733
16	0.3485	0.2797
8	0.3331	0.2533
4	.2941	0.2491
2	0.2181	0.1872
1	0.2006	0.1405

Table-1 shows impact of no of filters in layer-1. No of filter in layer 2, 3 are 16, 16 respectively. **Table-2** shows impact of no of filters in layer-2. No of filter in layer 1, 3 are 128, 16 respectively. **Table-3** shows impact of no of filters in layer-3. No of filter in layer 1, 3 are 256, 256 respectively.

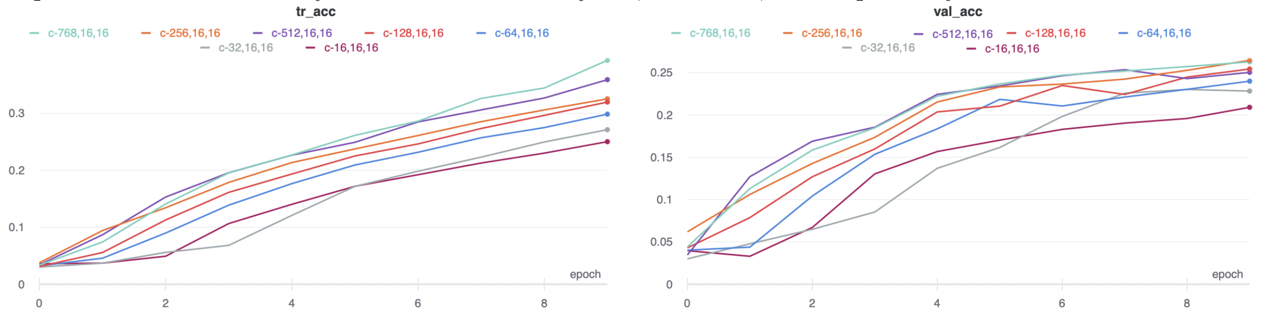


Figure shows impact on training (Left) and validation (Right) accuracy, when number of filters in 1st layer are increased from 16 to 768. **Left:** Its clear that as number of filters are increased, model performance is becoming better on training data and its getting overfitted on training data. **Right:** On validation data, increase in filters beyond 256 is leading to similar performance.

1.2 Kernel size

If kernel size is too small, then deeper network will be needed to capture global information within an image. Bigger kernel size will cover more receptive field and model will be learning global relations within earlier layers. But if kernel size is too large, then model will no longer be under constraint to have sparse connections and it will be very similar to dense network. Model with very large kernel size will be over-parameterized and unable to extract features from image while less kernel size will be neglecting global information. So, kernel size tuning can be very important to have better performance especially in shallow networks.

Table-1		
Kernel size	Train Accuracy	Val Accuracy
13	0.3225	0.2824
11	0.3358	0.2895
9	0.338	0.2663
7	0.3604	0.2827
5	0.3865	0.2733

Table-1 shows impact of kernel size in layer-1. Kernel size in layer- 2, 3 are kept 5, 5 respectively.

1.3 Stride

Stride decides movement of kernel over image and hence effects the size of feature map size. Stride should not be more than kernel size because else model will be missing some information. Having good value of stride will result in increase in training speed and less computational cost. If network is very deep, then we need to assert that stride is not resulting in very small feature map at the end of last CNN layer.

Table-1				
s1	s2	s3	Train Accuracy	Val Accuracy
1	1	2	0.2746	0.259
1	2	2	0.2367	0.2238
1	2	1	0.2619	0.2508
2	1	1	0.2261	0.2031

Table-1 shows impact of changing stride on model's performance. Above table suggests that stride should not be large atleast in initial layers. Having more stride can reduce the computational cost at the cost of accuracy.

1.4 Maxpooling

Maxpooling helps in making model invariant to small shift in pixel values. It can help to speedup training if stride is set to appropiate value. The main idea behind it is that we need only max responses at local levels.

Table-1		
Maxpool size	Train Accuracy	Val Accuracy
3	0.3326	0.2833
2	0.3313	0.2982
No Pool		

Table-1 shows impact of pool size. Performance slightly decreased when pool size is increased.

1.5 No of Convolutional layers

The more deeper the network is more complex function it can capture. In the final layers, model can focus on the features which are significantly differ in different samples and hence model can become better.

Table-1		
No of CNN layers	Train Accuracy	Val Accuracy
1	0.1654	0.1701
2	0.2219	0.2239
3	0.2764	0.2536
4	0.2692	0.2672
5	0.3095	0.3104
6	0.1567	0.1647

Table-1 shows impact of no of CNN layers.

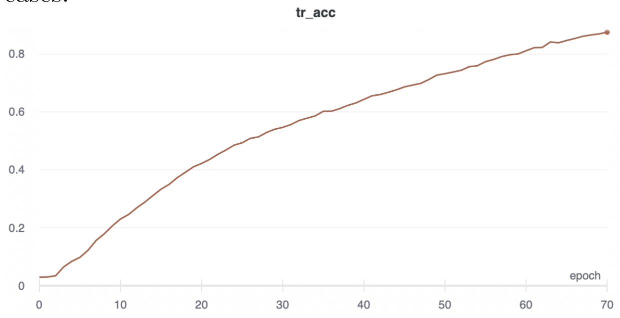
1.6 No of fully connected layers

Once the features are extracted, these layers do the classification task. Most of the parameters of the network, lies in these layers.

Table-1		
No of fc layers	Train Accuracy	Val Accuracy
1	0.2991	0.2954
2	0.3357	0.3498
3	0.243	0.2209
4	0.159	0.131

1.7 Training time (No of epochs)

Generally no of epochs required for model's convergence depends on its complexity and capacity. If model is over-trained (i.e trained for very large no of epochs) then, there are more chances that model will start memorizing the training data after some time and validation accuracy will start decreasing or remains same. This can happen since model is seeing same data again and again. So, early stopping can be helpful in those cases.



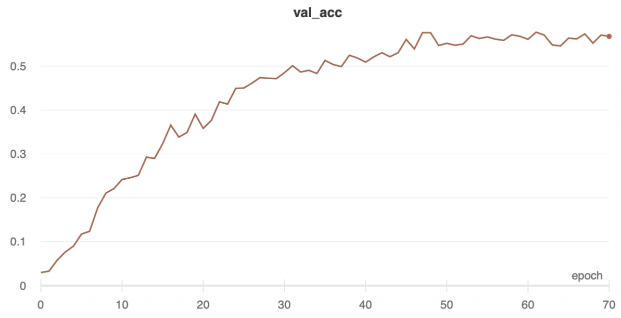


Figure above clearly shows that as no of epochs are increasing model is no longer becoming better on validation data, rather accuracy is remaining constant or decreasing slightly. But on training data, accuracy is still increasing although at decreasing rate. In above figure, some smoothing has been applied to original graph for easily observing pattern.

Table-1			
Epochs	Training Time (in mins)	Train Accuracy	Val Accuracy
10	29.517	0.2304	0.2419
20	39.8	0.4222	0.3582
30	50.117	0.5468	0.485
40	60.45	0.6434	0.509
50	70.8	0.7321	0.552
60	81.167	0.8114	0.561
70	91.55	0.8775	0.5676

Table-1 shows impact of no of epochs (or training time). Training time is larger for 1st 10 epochs as compared to next 10 because for the 1st epoch, system needs to load data from disk.

1.8 Effect of Learning Rate

Learning rate is one of the most important hyperparameter for deciding whether model will converge or never converge at all. If learning rate is too large, then model will diverge for sure and if its too small, then model can take infinite time to achieve good accuracy.

The best model is trained for 48 epochs on combination of train val data and achieved the accuracy of 59.77 on test dataset. Further, this model is used for experiments in part-B.

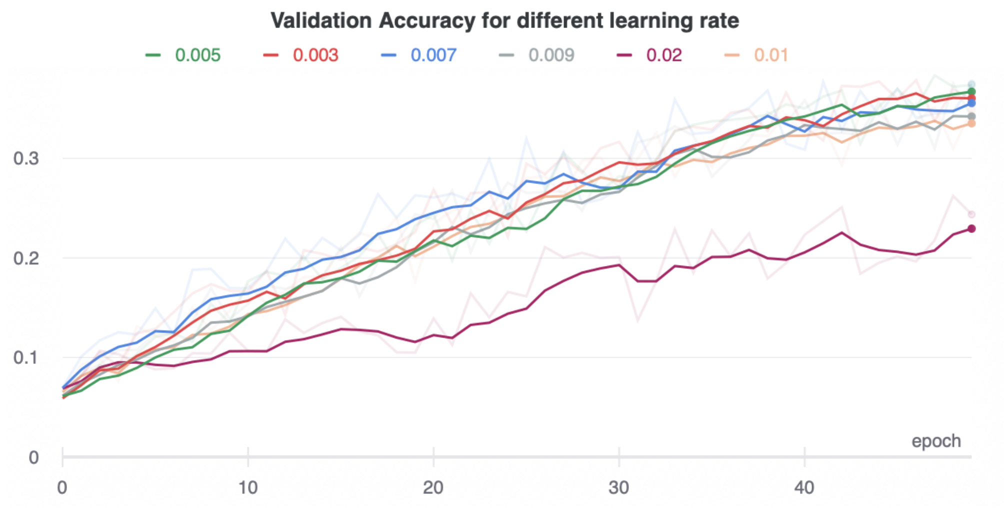


Figure 1: Figure shows effect of learning rate on validation accuracy. Learning rates are mentioned in figure itself.

2 Visualizing and Understanding Convolutional Neural Networks

2.1 Occlusion Test

2.1.1 Binary Filters

Binary filters are generated with 0 in place of patch of occlusion. Further these are multiplied with original images to get image with black occluded patch. Now this black occluded image is converted to gray-scale occluded image.

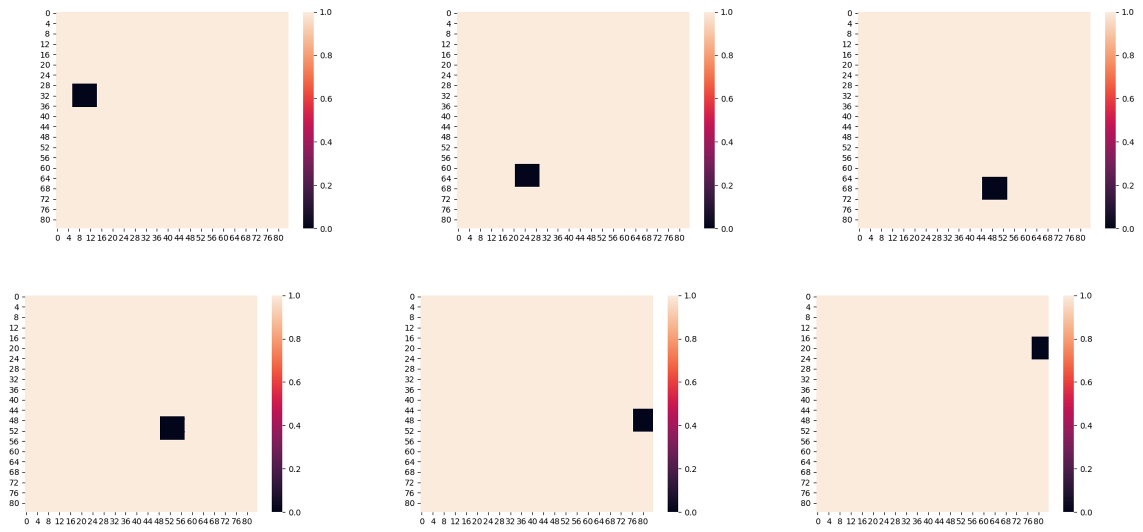


Figure shows the heatmaps of the binary-filter generated for matrix multiplications of these binary-filters and original image to output occluded image.

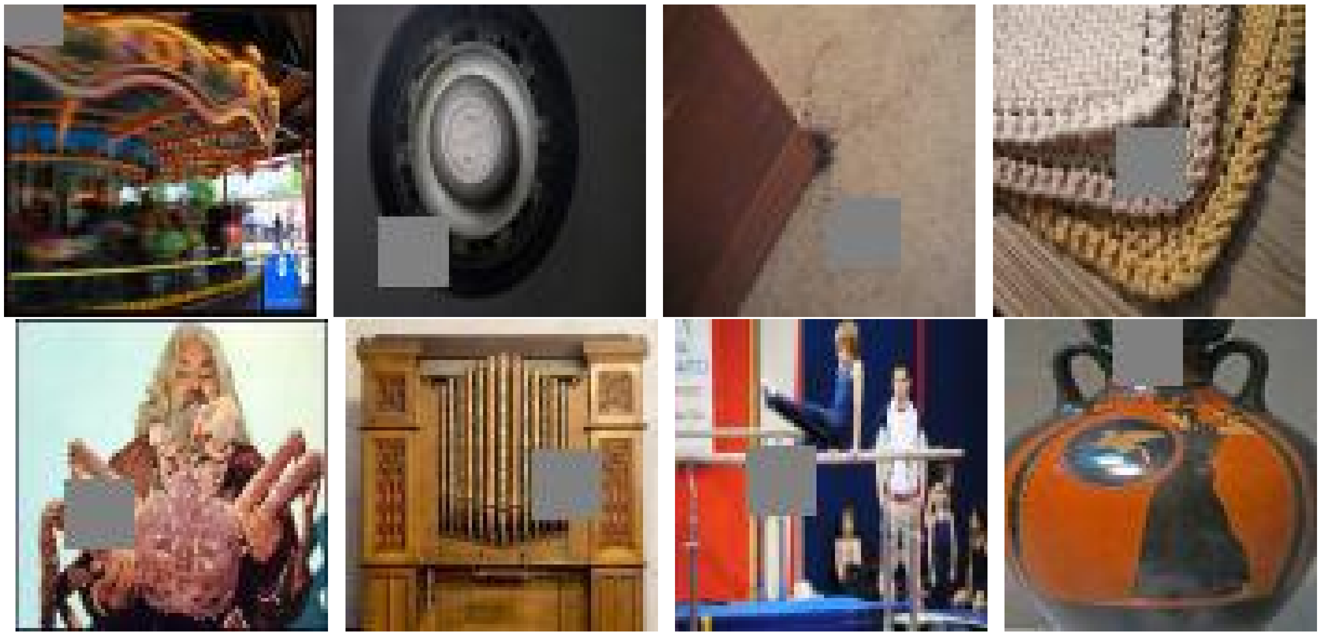
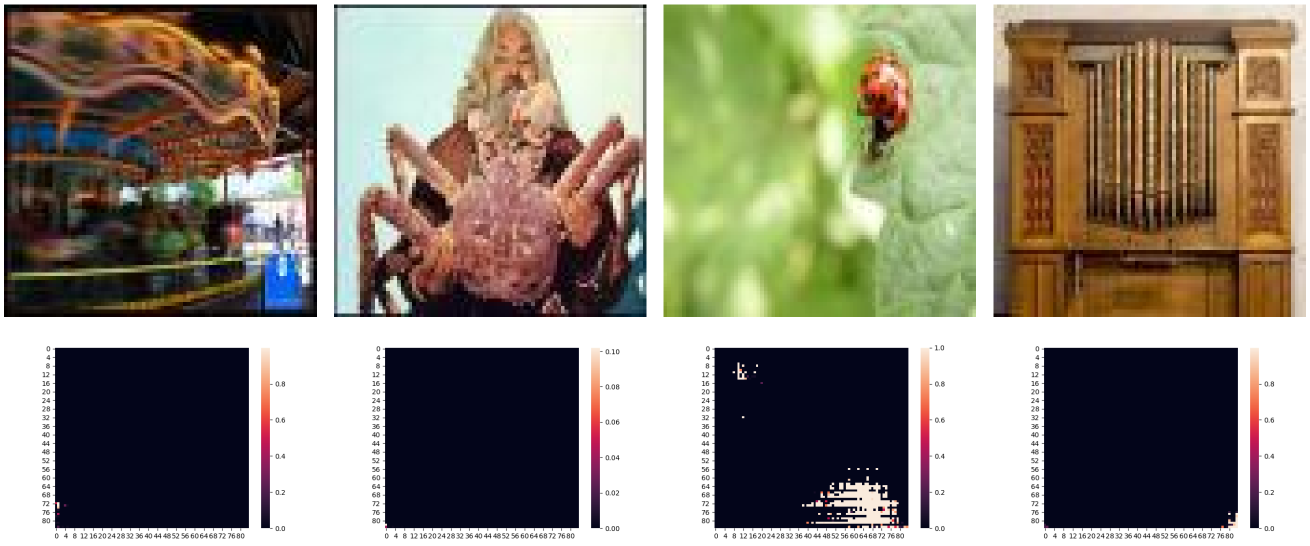
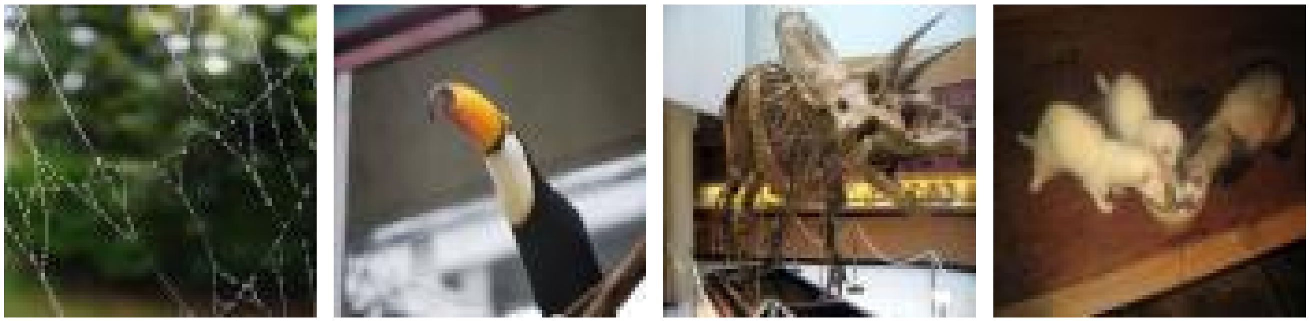
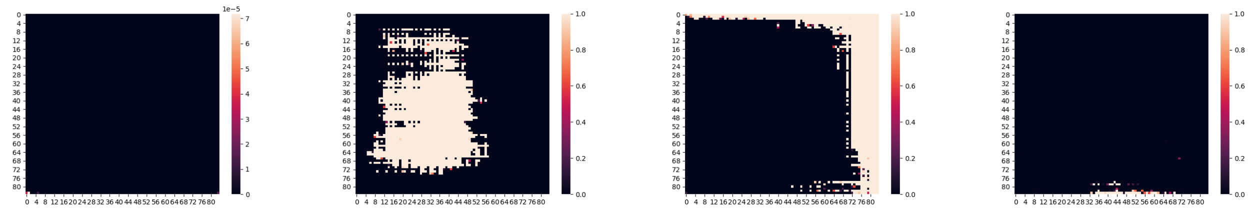
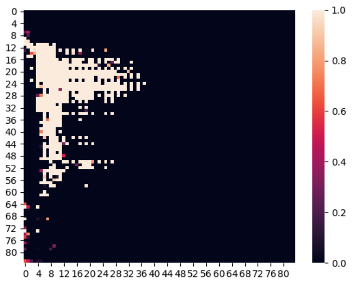


Figure shows occluded image which can be generated easily by matrix-multiplication of above filters and original image.

2.1.2 Confidence array



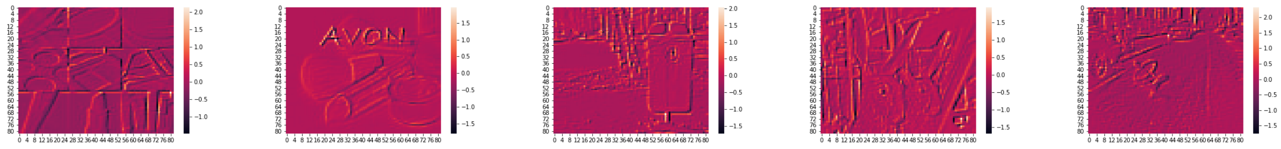


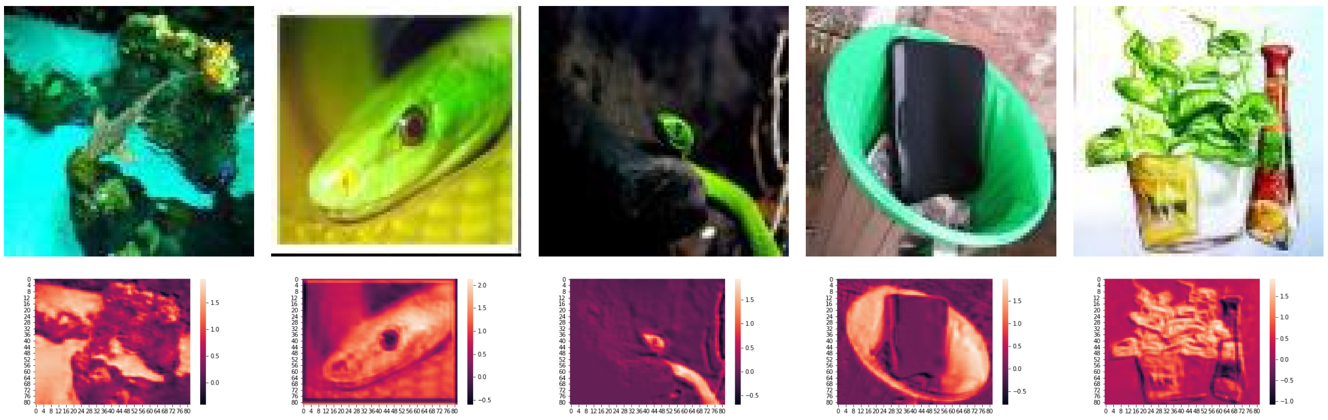
Images with their confidence arrays (below the image) are shown.

For the images, which are spread across complete images; confidence score is mostly 0 everywhere which suggests that its not easy for the model to classify images based on some pixels and rather all the pixels are important. But for the images which are more centered, classification score is independent of boundary pixels or pixels which are far from actual object.

2.2 Filter Analysis

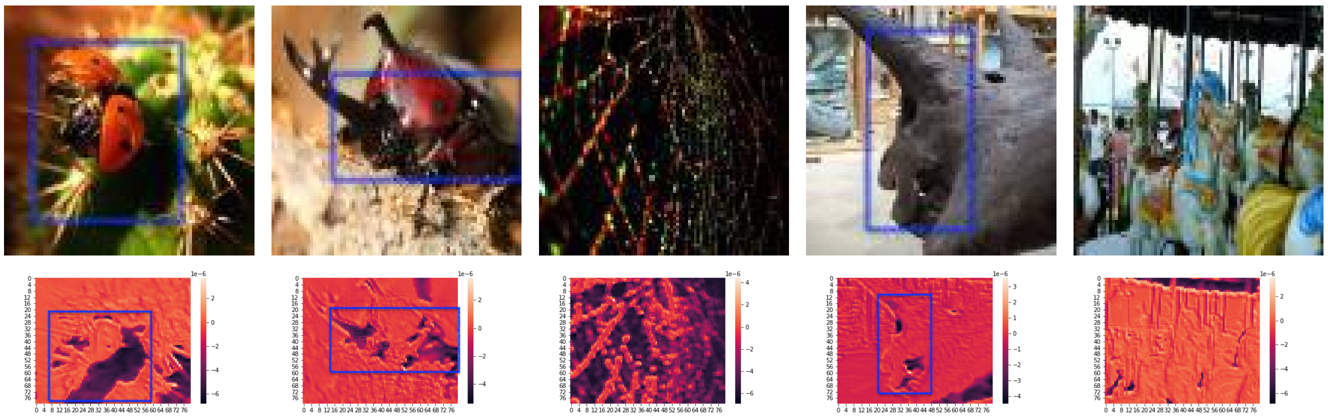
2.2.1 Responses from layer-1

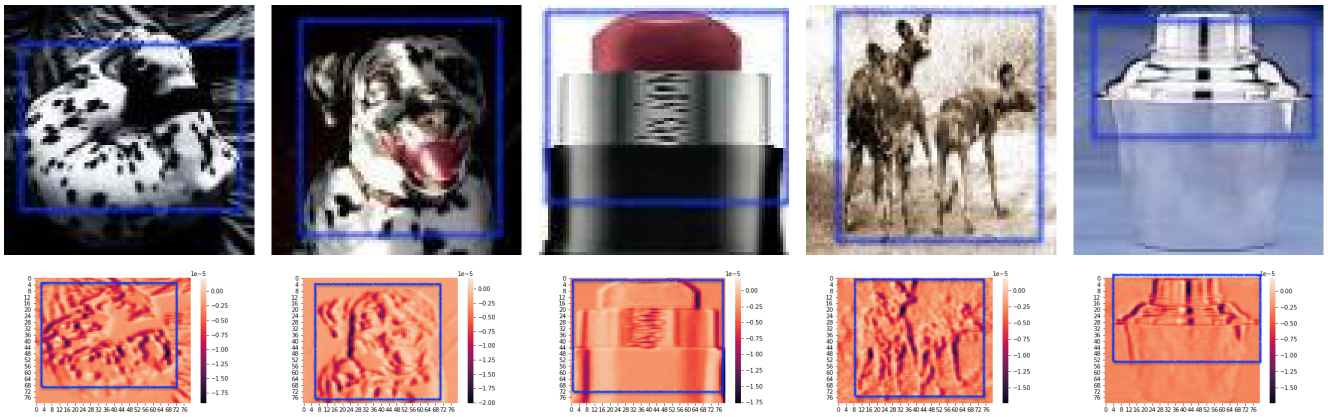




These are the responses from 2 filters from layers 1 of the network.

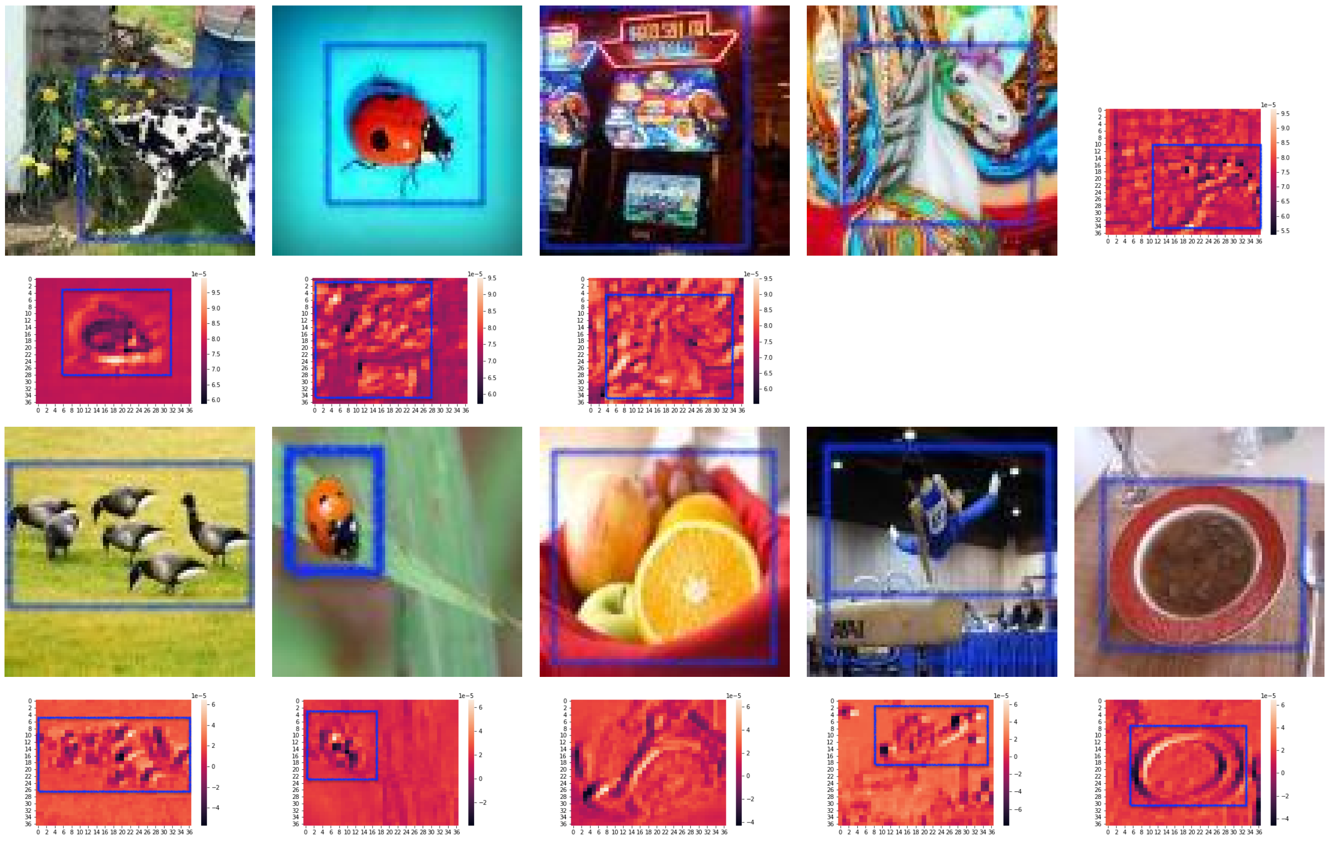
2.2.2 Responses from layer-2





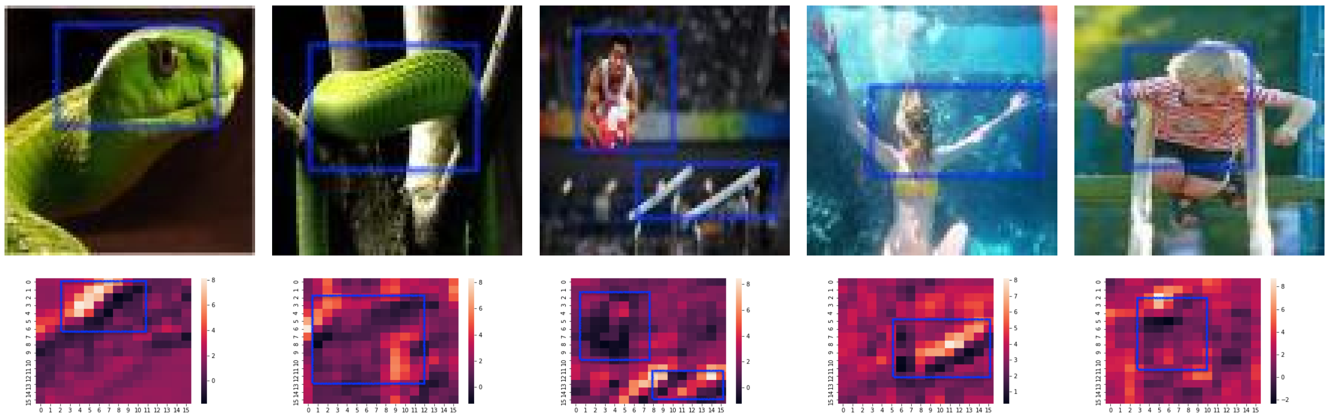
Above are the responses from 2 filters from layer-2. Original images and patches corresponding to max-responses are also show.

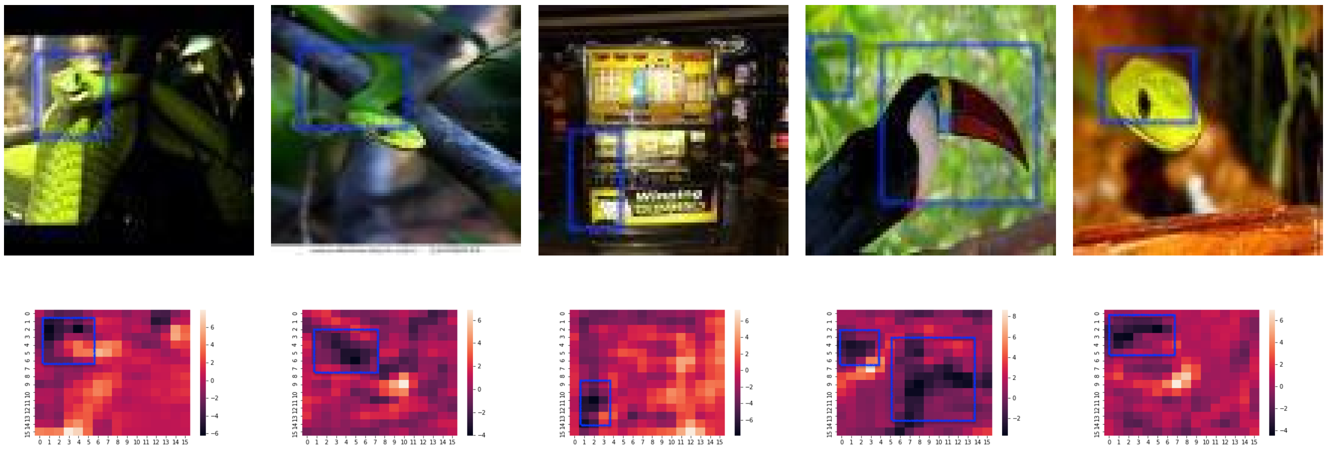
2.2.3 Responses from layer-3



Above are the responses from 2 filters from layer-3. Original images and patches corresponding to max-responses are also show.

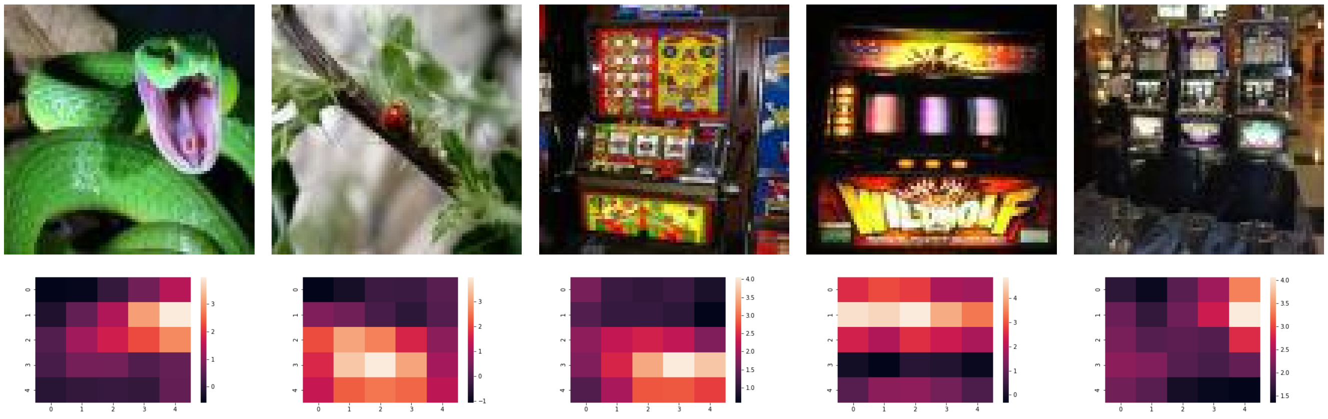
2.2.4 Responses from layer-4

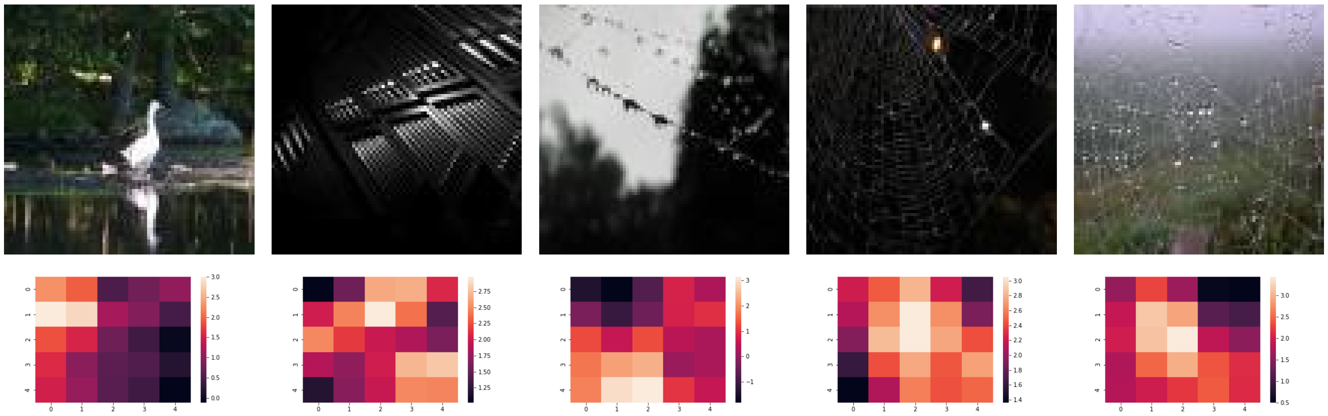




Above are the responses from 2 filters from layer-4. Original images and patches corresponding to max-responses are also show.

2.2.5 Responses from layer-5





Above are the responses from 2 filters from layer-5. Original images and patches corresponding to max-responses are also show.

In layer-1, model has just captured whatever information is there in original image as it is. It is just identifying the boundaries within the image. In layer-2, 3; model has actually started learning something. It is not focusing on complete image; rather a subset from original image. Layer-4 weights are further focusing on even smaller subset of image. Its looks like this layer is overfitted more on snake class. Layer-5 filters are very hard to interpret since they are very small as compared as compared to size of original image.

2.2.6 Modification in layer-1 filters

Filter-1	
Class No	frequency
21	12
29	5
16	5

Filter-2	
Class No	frequency
20	4
1	4
10	4

2.2.7 Modification in layer-2 filters

Filter-1	
Class No	frequency
1	1
4	1

Filter-2	
Class No	frequency
11	9
0	5
5	5
8	5

2.2.8 Modification in layer-3 filters

Filter-1	
Class No	frequency
29	2
10	2

Filter-2	
Class No	frequency
13	1
17	1

2.2.9 Modification in layer-4 filters

Filter-1	
Class No	frequency
23	3
1	2
4	2
27	2

Filter-2	
Class No	frequency
26	5
4	4
6	4
31	2

2.2.10 Modification in layer-5 filters

Filter-1	
Class No	frequency
4	17
2	10
22	10
7	7
19	7
24	5
26	4

Filter-2	
Class No	frequency
9	36
15	17
8	7
32	7
1	4
7	4

Above tables shows the frequency of the samples which were classified correctly by normal trained model but misclassified when one of the filter is zeroed out. Only few top classes are shown above.

Above tables shows clearly that last layers are more important than initial layers atleast for classification problems. It shows that initial layers captures the features are which are common in most of samples while the final layers focuses more on capturing features which can more different and helpful for classifying 2 images.

3 Summary and Conclusions

Deep models generally performs better as compared to shallower, if trained properly. Developing insights about what model is doing, can be really important to improve performance.