Introduction to Deep Learning CS6910

Chella Thiyagarajan N
ME17B179

Data Set Preview:

The mini-ImageNet dataset was proposed by Vinyals et al. for few-shot learning evaluation. Its complexity is high due to the use of ImageNet images but requires fewer resources and infrastructure than running on the full ImageNet dataset. In total, there are 33 classes with 600 samples of 84×84 color images per class.

The dataset is divided into 400 images/class for trainset, 100 images/class for validation set and 100 images/class for test set.

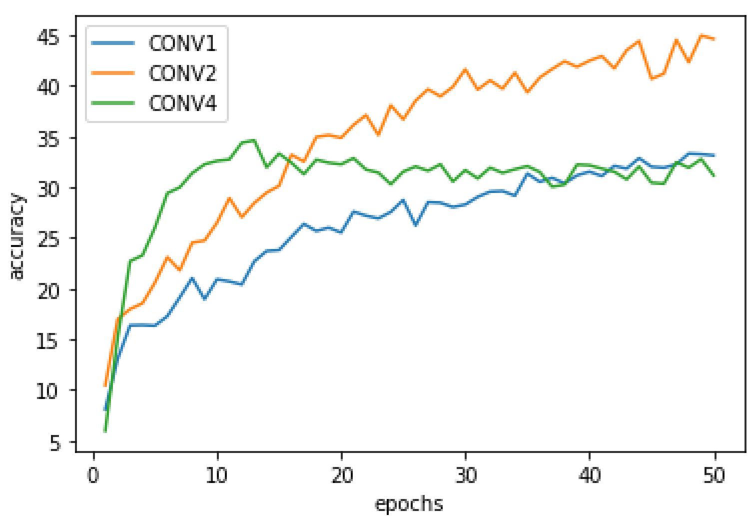
Part A

Aim: It is expected to experiment with various parameters of a typical convolutional network to come up with a study of the change in the training, validation, and test accuracies. Based on the inference of the study, one has to improve the performance of the model.

Convolution Layers:

This sub-section looks at the observations when the number of convolutional layers are changed from the given boilerplate architecture. Note that the other features of the model across all the layers are kept constant during the study. It has been experimented with 1,2 and 4 convolutional layers.

Results:



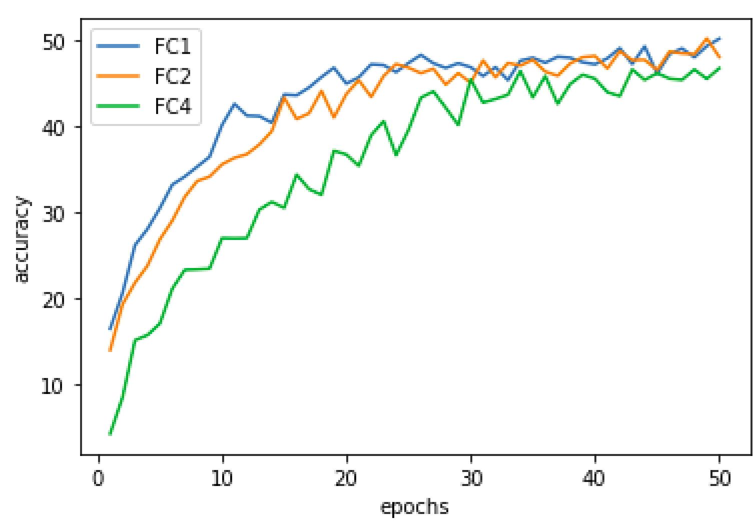
Model	Validation Accuracy
1 layer	33%
2 layers	44%
4 layers	29%

Inferences:

- 1 Convolutional layers are able to produce reasonable train and validation accuracy.
- 2 Convolutional layers seems to outperform 1 layer model
- 4 convolution layer model seem to perform better in training data. But on validation data it performed poorer than 2 convolutional layer model
- 4 layer model has shown poorer performance because the rate of improvement decreases with more layers. This is likely due to the heavy loss of information due to maxpool layers that are in between.

Fully Connected Layers:

This sub-section looks at the observations when the number of fully connected layers are changed from the given boilerplate architecture. Note that the other features of the model across all the layers are kept constant during the study. It has been experimented with 1,2 and 4 convolutional layers.

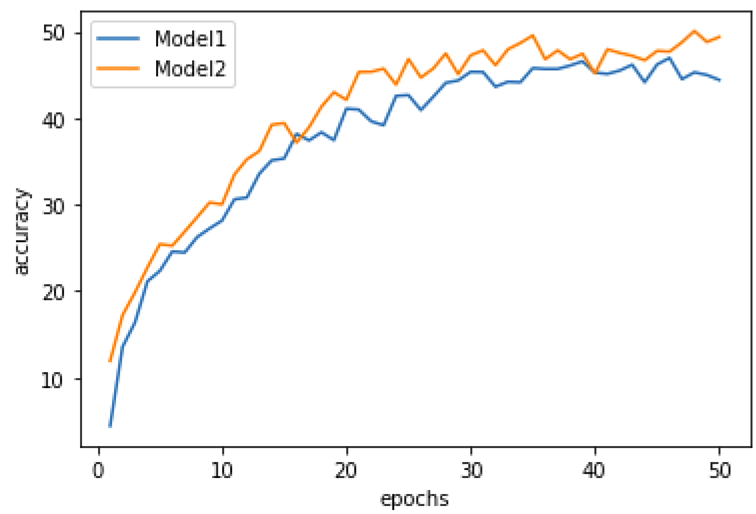


Inference:

Changing the number of fully connected layers does not have any significant effect on the accuracy of the model on the validation set. Therefore the number of fully connected layers does not affect the model's performance. Even training loss is not affected

Number of Filters:

This sub-section looks at the observations when the number of filters are changed from the given boilerplate architecture. Note that the other features of the model across all the layers are kept constant during the study. It has been experimented with 48 and 112 convolutional layers.



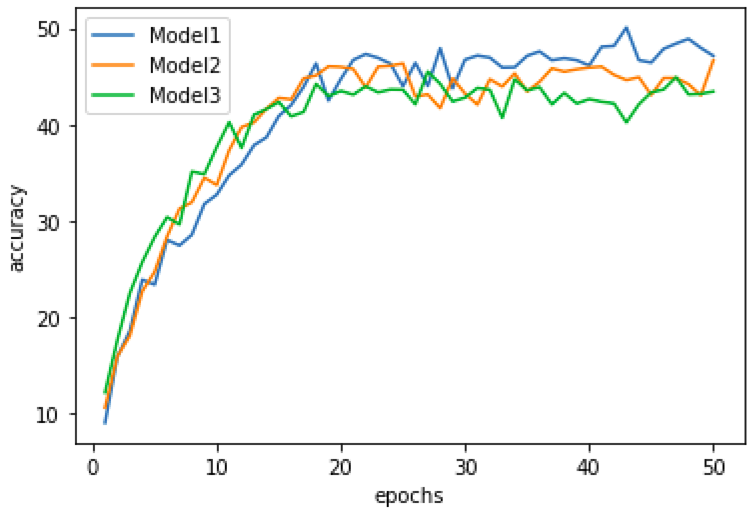
Model 1 corresponds to a conv model with 48 filters and Model 2 corresponds to a conv model with 112 filters

Inferences:

From the above plot we can conclude that increasing the number of filters in a convolution layer increases model's performance. More number of filters used in a convolution layer indicates more number of features are extracted from previous layers and hence the model learns to distinguish between classes using new correlations of image features that were not identified previously.

Max Pooling:

This sub-section looks at the observations when the stride of Max Pooling techniques are changed from the given boiler plate architecture. It was experimented with a stride of 2,3 and 4



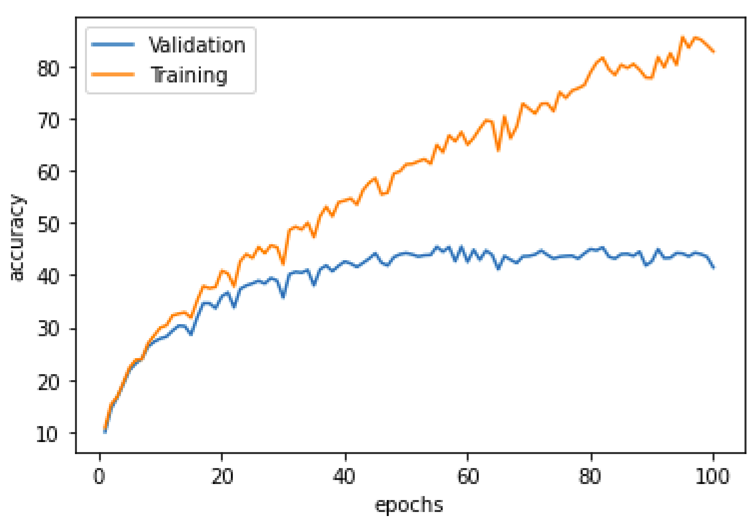
In the above Figure, Model 1 corresponds to stride=2, Model 2 corresponds to stride=3 and Model 3 corresponds to stride=4.

Inferences:

From the above plot we can conclude that as we increase stride size, accuracy decreases. From Observation: training loss, validation accuracy and test accuracy also supports the above fact that the performance of the model decreases with increase in stride size.

Number of Epochs:

This sub-section looks at the observations when the number of epochs are changed from the given boilerplate architecture. Note that the other features of the model across all the layers are kept constant during the study. It has been experimented with upto 100 epochs.

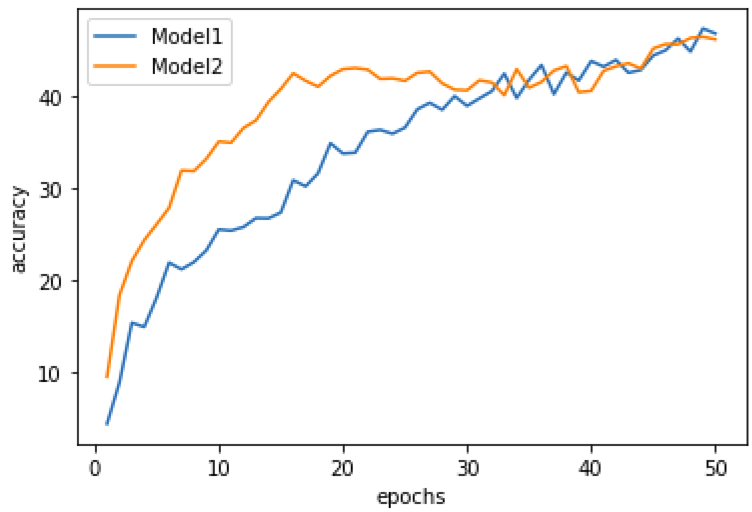


Inferences:

It is normal to expect that as the number of epochs increase training loss decreases and training accuracy increases. From the above plot we can observe that rate of increase of accuracy decreases with increase in epochs. Validation accuracy saturates after certain number of epochs and tend to even decrease due to the problem of overfitting as the train loss decreases significantly with increase in epochs.

Filter Size:

This sub-section looks at the observations when the size of Filters are changed from the given boilerplate architecture. Note that the other features of the model across all the layers are kept constant during the study. It has been experimented with 3x3 filter size and 5x5 filter size.



Model 1 corresponds to Filter size: 3x3 and Model 2 corresponds to Filter size 5x5

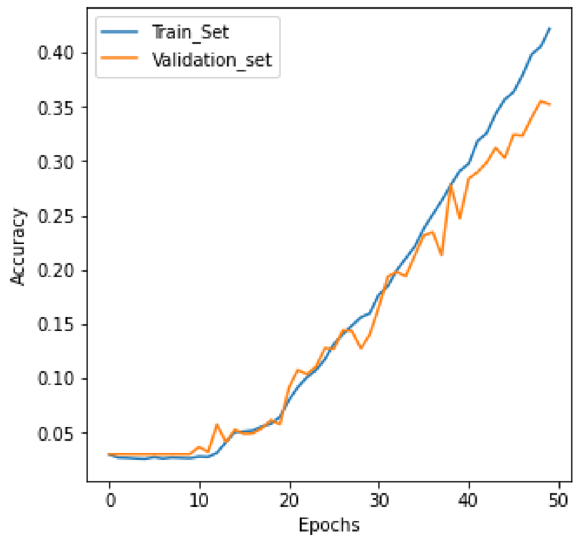
Inferences:

From the above plot we can observe that filter size of 5x5 achieves high accuracies within less number of epochs, whereas model with filter size 3x3 increases gradually. Given a reasonable high epoch number both tend to perform in the same level with filter 3x3 having slightly higher accuracies.

Padding:

This sub-section looks at the observations when the size of Padding are changed from the given boilerplate architecture. Note that the other features of the model across all the layers are kept constant during the study. It has been experimented with padding=0 and padding = 1.

Image1: Image2:



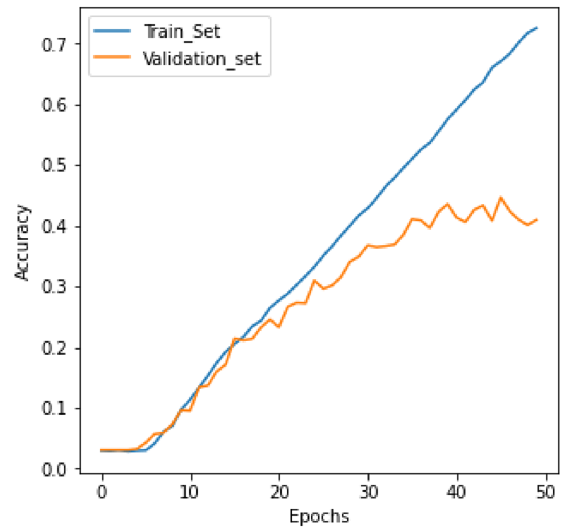


Image1 corresponds to the model with padding and Image2 corresponds to the model without padding

Inferences:

Model without padding has higher performance than model with padding, this may be because the object of interest is mostly located in the middle of the image and edges are not given much importance. We know padding captures edges, which is unnecessary in our dataset.

Training Time:

Training time has direct influence on the performance of the network. This is obvious from the fact that the number of optimizing steps taken increases with time. It is observed that the training accuracy increases to 100% as we train more. However, the testing accuracy saturates at a point and does not increase further.

Best Model:

So far we have observed a lot of models. The best out of this experiment turned out are two models.

Model A:
Convolution layers: 3
The final number of filters: 256
Max pool Kernel size: 2
Filter size: 5
Epochs: 50
Stride: 2
Fully connected layers: 3

Model B:
Convolution layers: 5
The final number of filters: 256
Max pool Kernel size: 3
Filter size: 3
Epochs: 50
Stride: 1
Fully connected layers: 3

On my Assignment1.ipynb file the above two models are defined as model2 and model4 respectively. Model 1 in the ipynb file is to explain how a crude model performs. Model 3 in the ipynb file is to explain how padding affects our model4.

Best Model A = model2 in Assignment1.ipynb
Best Model B = model4 in Assignment1.ipynb

Reason for misclassification by looking at a few images:

Looking at some of the images that are misclassified, we can find some of the feature common between a few of them. The model learned to classify images with blue or dark background as an aircraft carrier. The model failed to differentiate between a house finch and hourglass due to the wood background. Mostly model misclassifies as it predicts an image by background than true class features.

Conclusion:

It was experimented with a lot more parameters than was presented and only the most interesting ones were presented in the report. The experiment really proved to be a good learning curve. Now moving on to the next part we will use the above selected best model to perform experiments.

Part B: 2.1

Aim: How the confidence of the model varies when a part of the image is greyed out.

For this exercise I choose a 50X50 filter with stride = 1 and performed the test across 10 random images, visualize the prediction probability of its true class as a heat map that visually expresses the confidence of a patch.

In the following Images, the patches with low probabilities tells us, that particular part of image contributes to the prediction of the true class.

I have used model2 for the following experiment

Results:

Figure1:

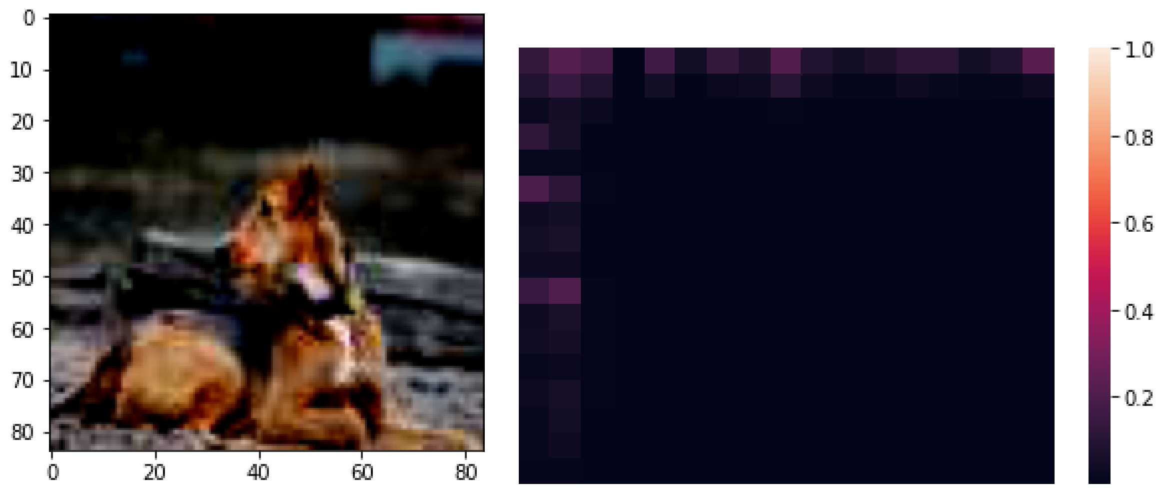


Figure 2:

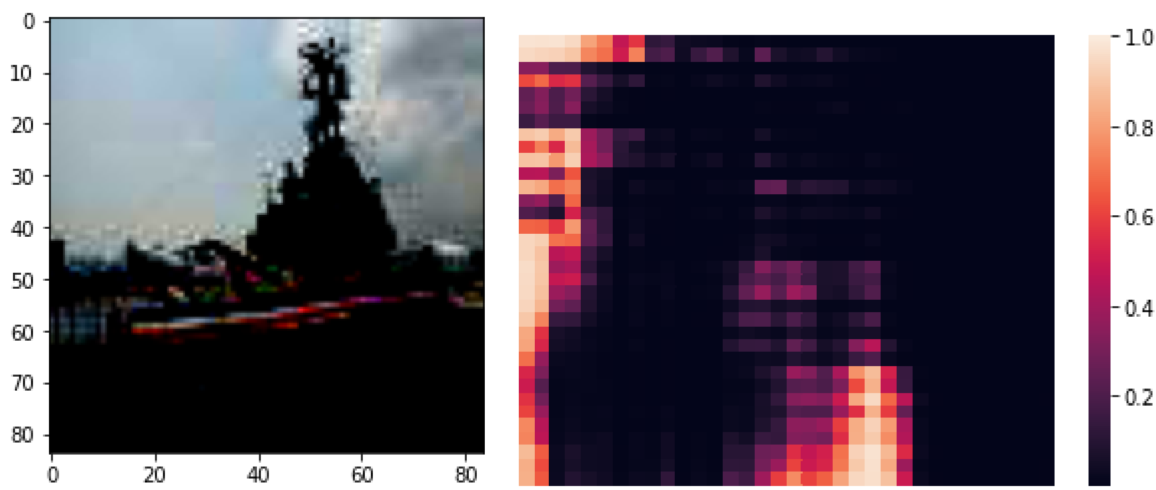


Figure 3:

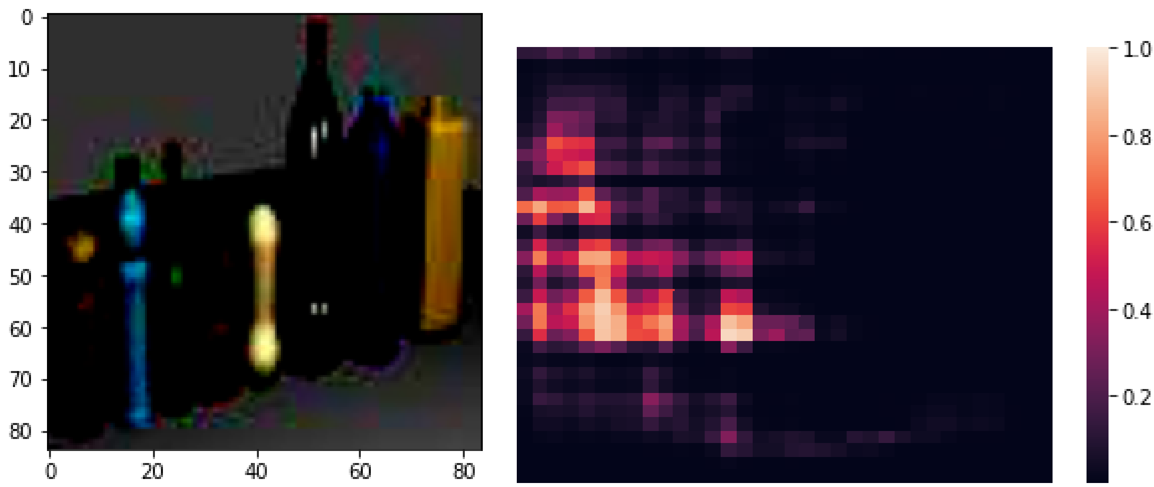


Figure 4:

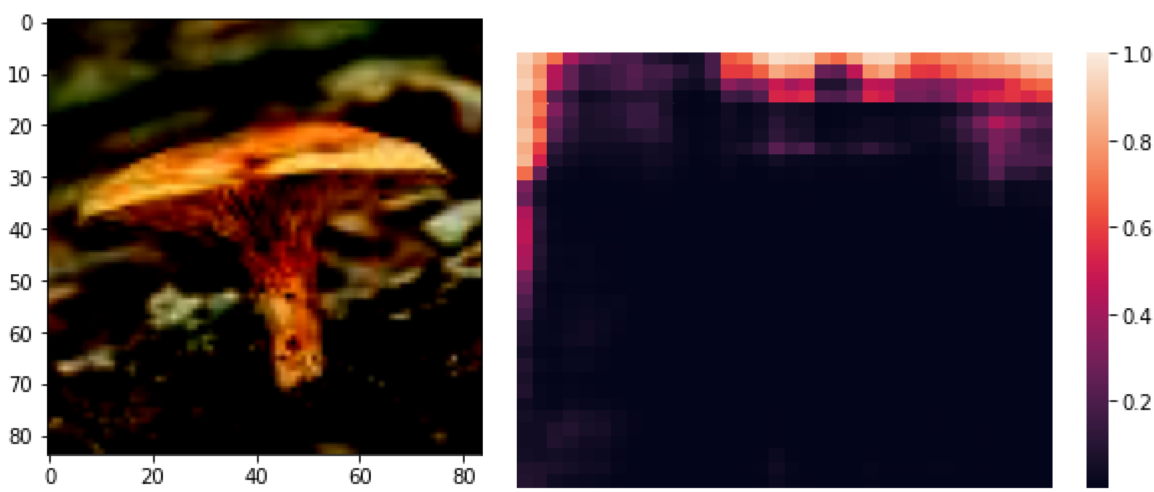


Figure 5:

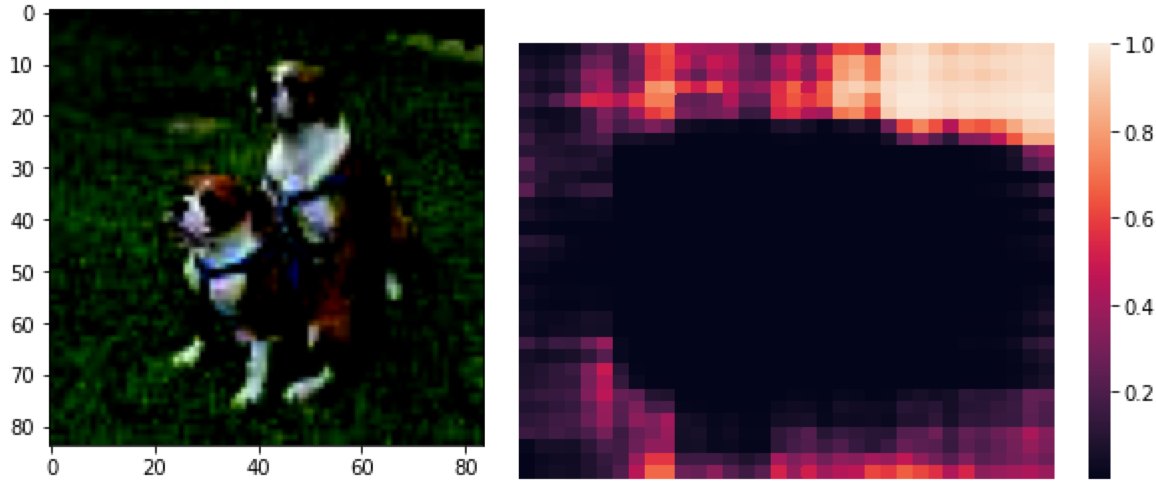


Figure 6:

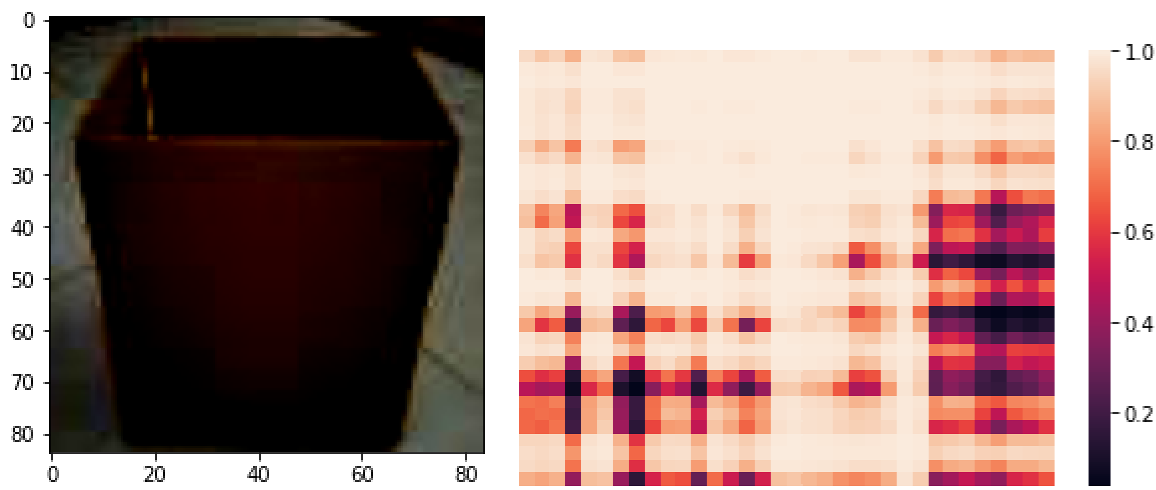


Figure 7:

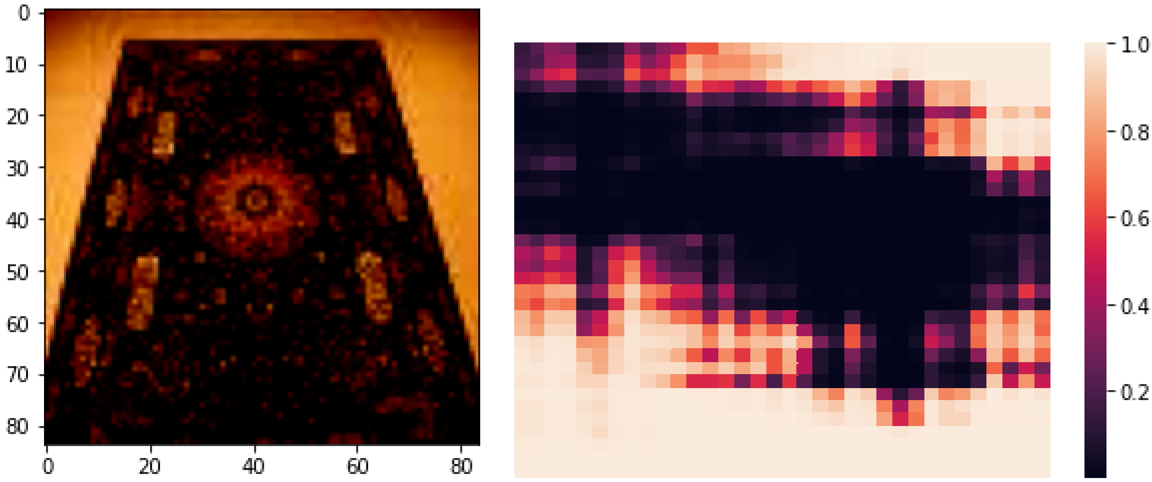


Figure 8:

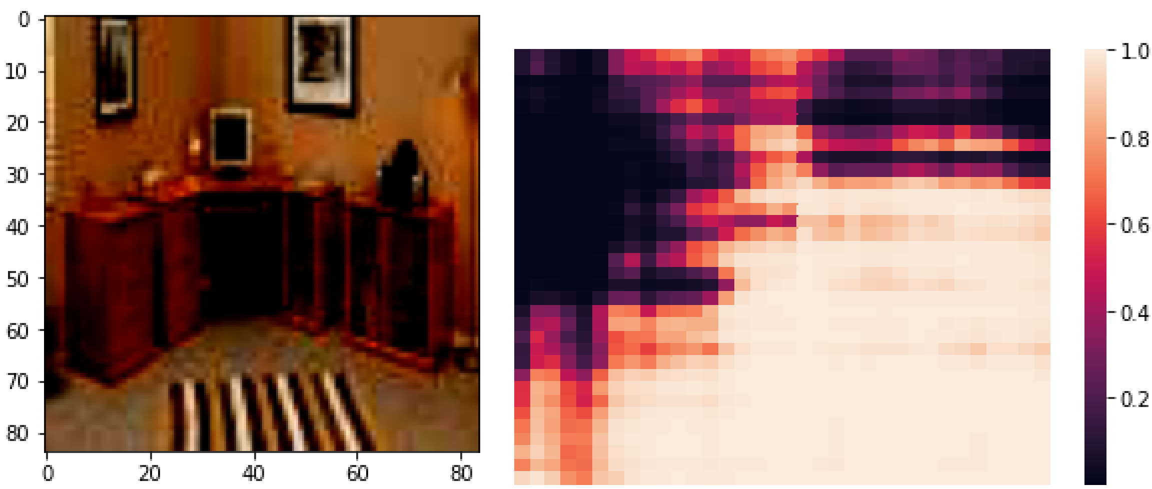


Figure 9:

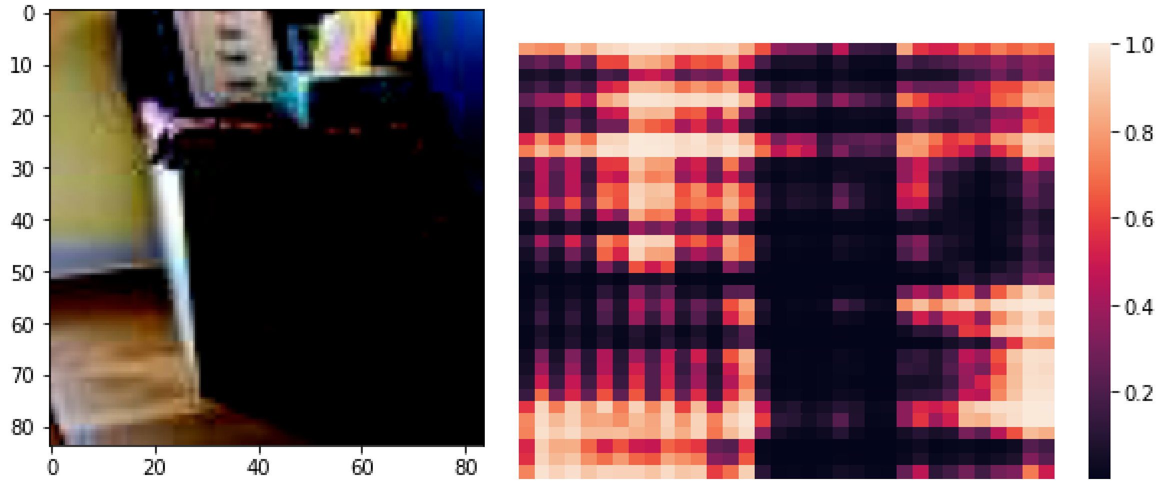
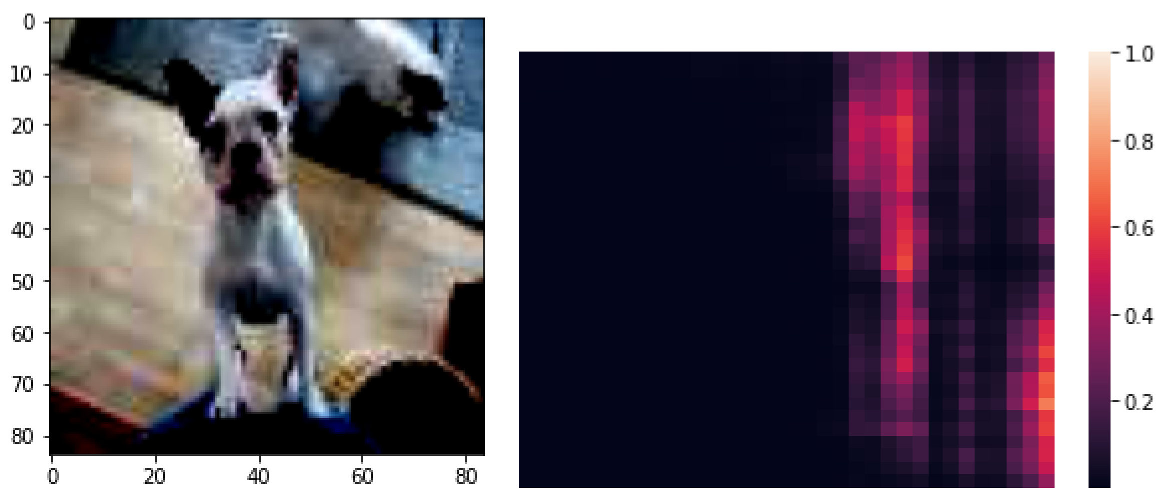


Figure 10:



Conclusion and Inferences:

The idea behind occlusion sensitivity experiment was really interesting to work upon.Ideally it should help in identifying which part of the image, the model sees the most to classify an object.

Summary of results of experiment using the given dataset and best accuracy model: Looking at the result 4, it seems that model considers all features from image for proper prediction and since almost entire image would be necessary for that, in corresponding heatmap most of the portion is in zero probability region, This may be caused due to larger occlusion patch size that I have chosen.

Figure 5 heatmap clearly points out the proximity or area covered by the dogs, this tells us that our model considers features of the true class to classify rather than classifying using random background cues.

Other than this it is difficult to conclude anything from other results.As we can see from images from dataset displayed above. Major reason for this is that given dataset quality is really poor. Some images are really poor in quality that human eye inspection itself is hard to give proper answer.Performing these experiments repeatedly multiple times, picking someother 10 images in random and doing the experiment also ended up in same result. Only in very few images we can realise the findings of heatmap and the free localising power that comes with CNN models.

Part B: 2.2.1

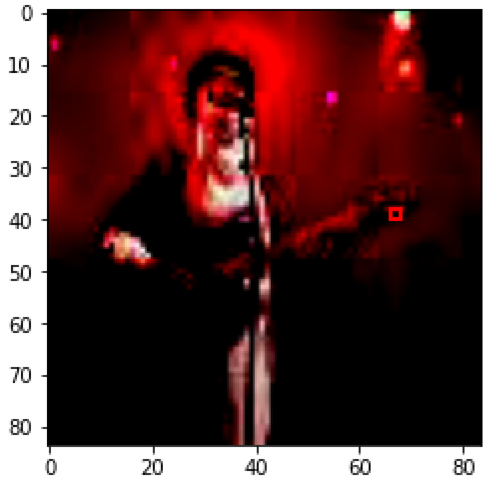
Overview:

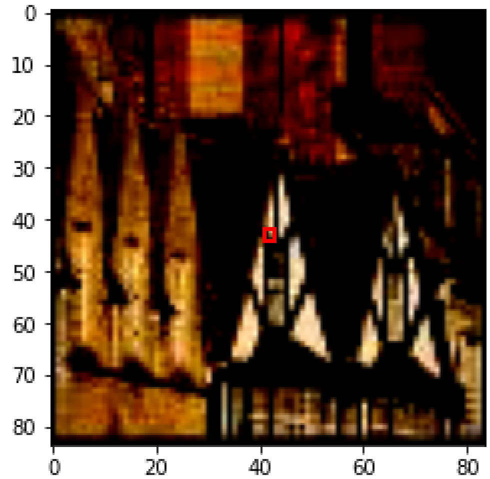
Model4 is chosen as our best model here. The final goal is to display the image patches corresponding to the maximum response of the chosen filters out of all the images. Let us look at the results of the experiment.

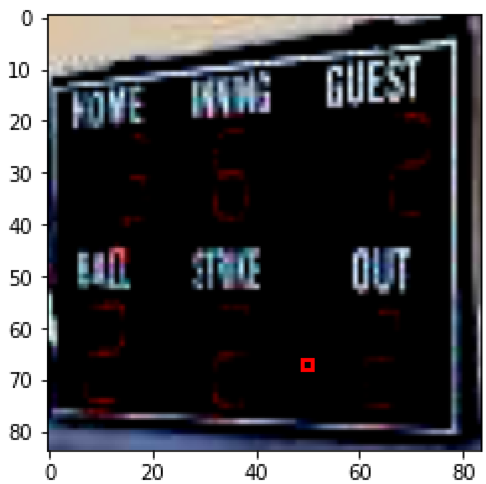
I have chose filter 3 and 12 from each convolutional layer

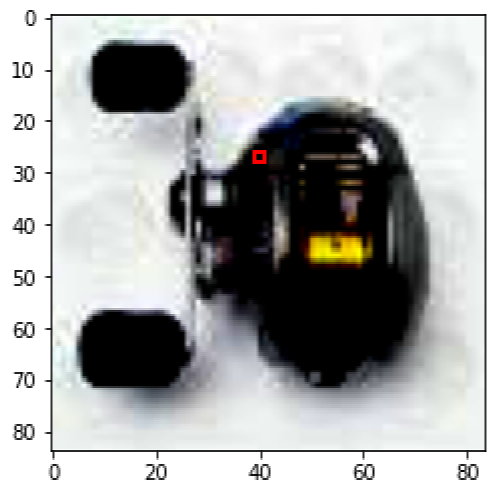
Results:

Layer1 and filter 3:



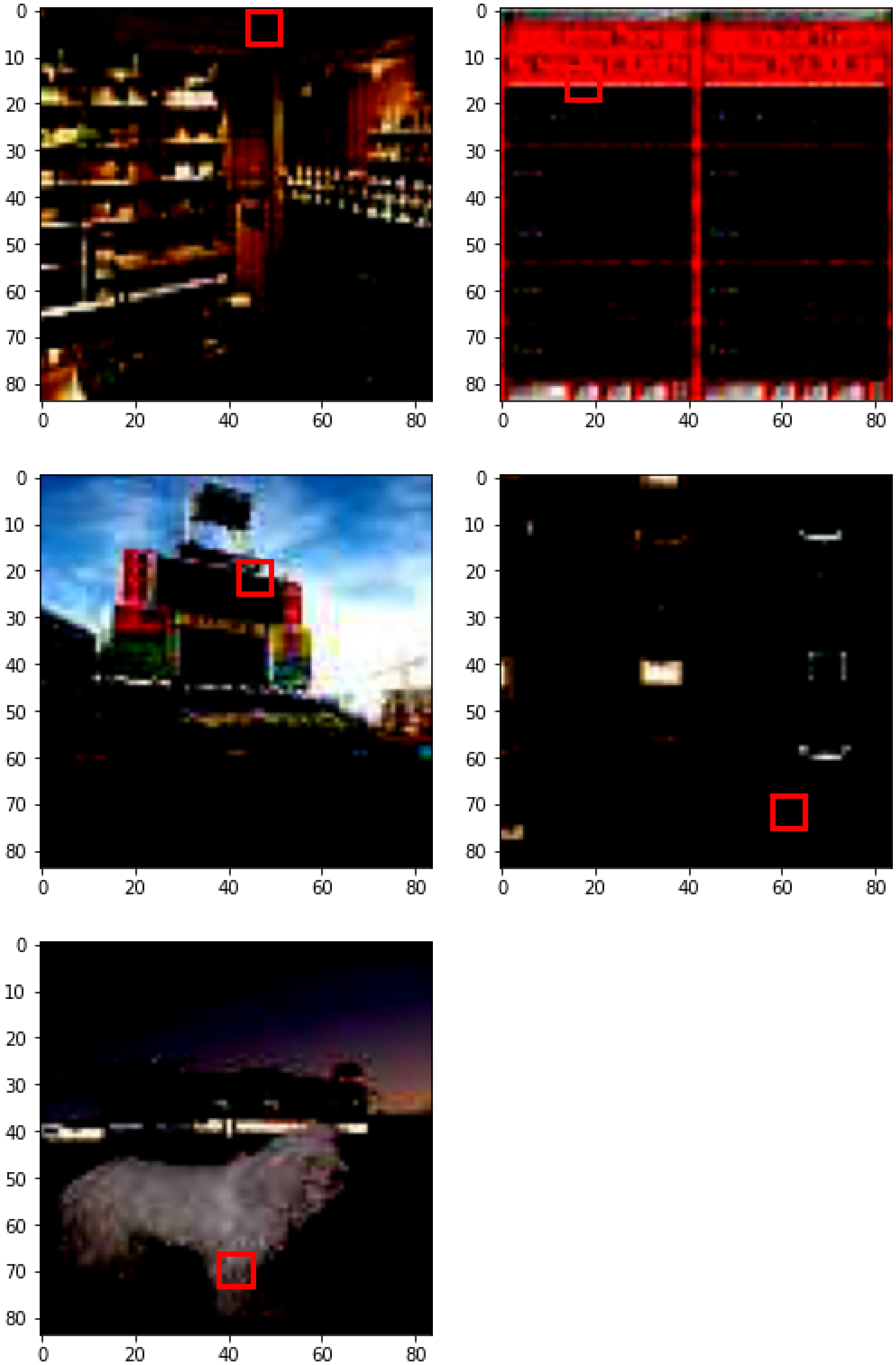






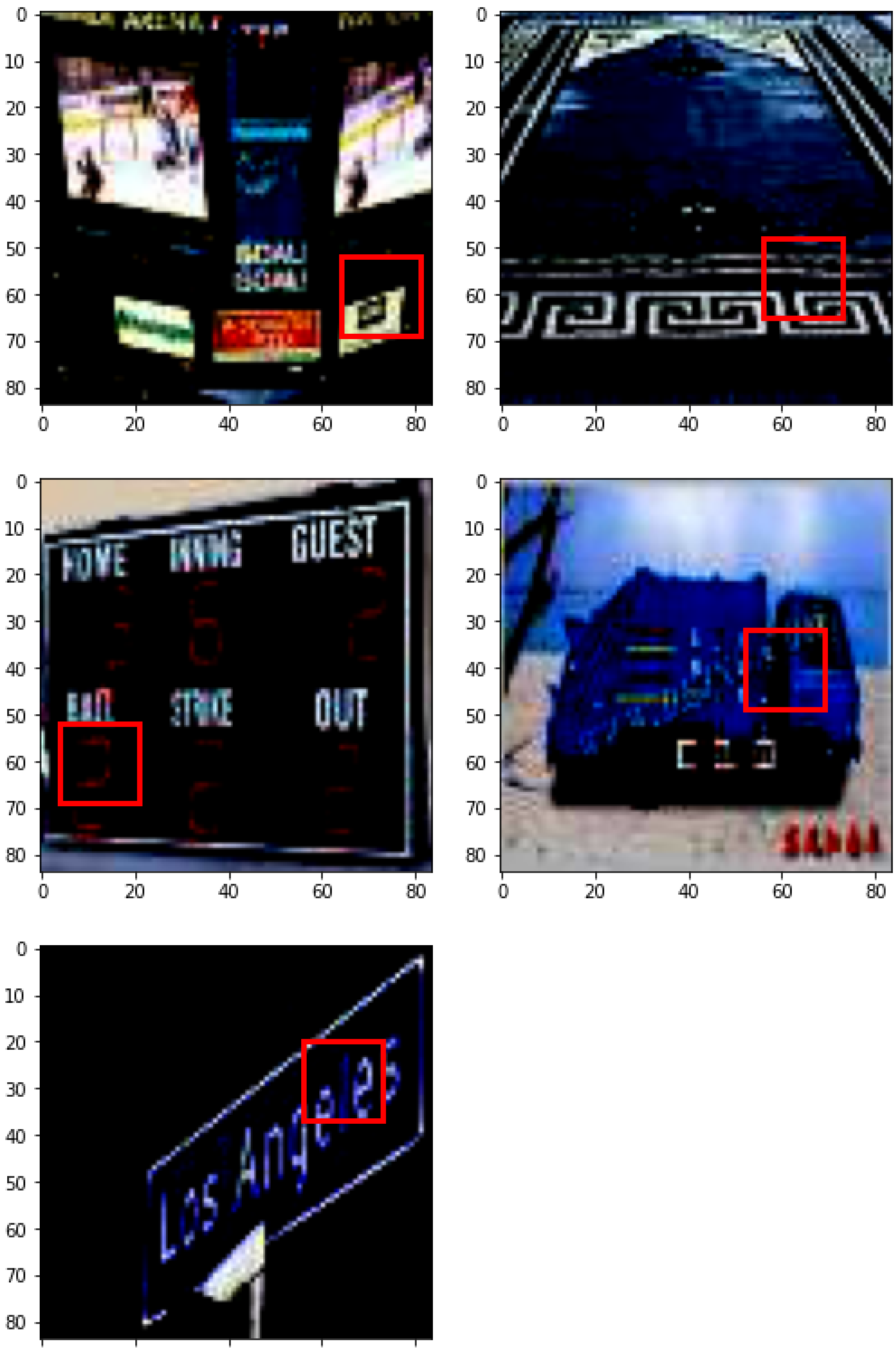
Layer1 and filter 12:

Layer2 and filter 3:

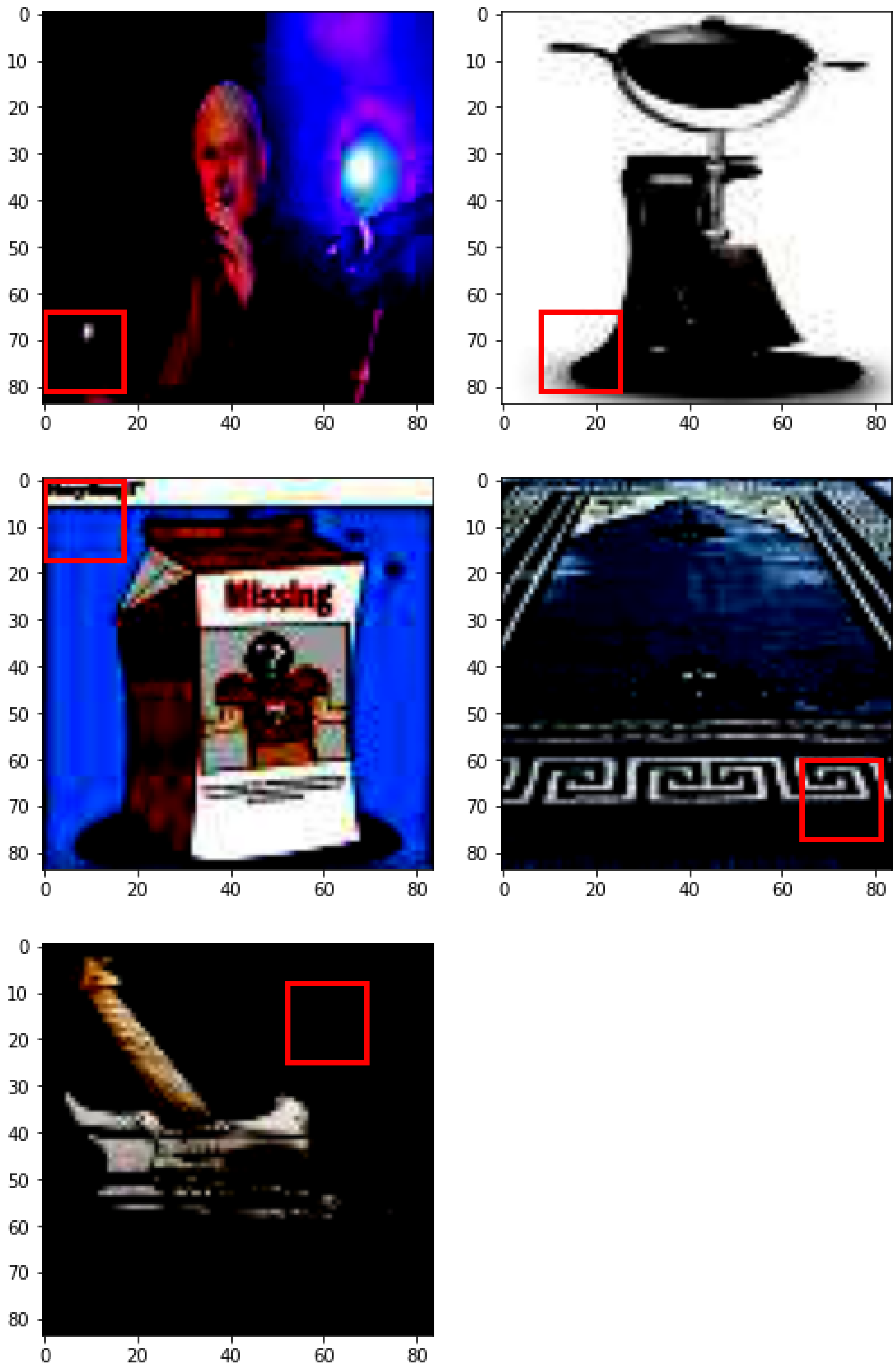


Layer2 and filter 12:

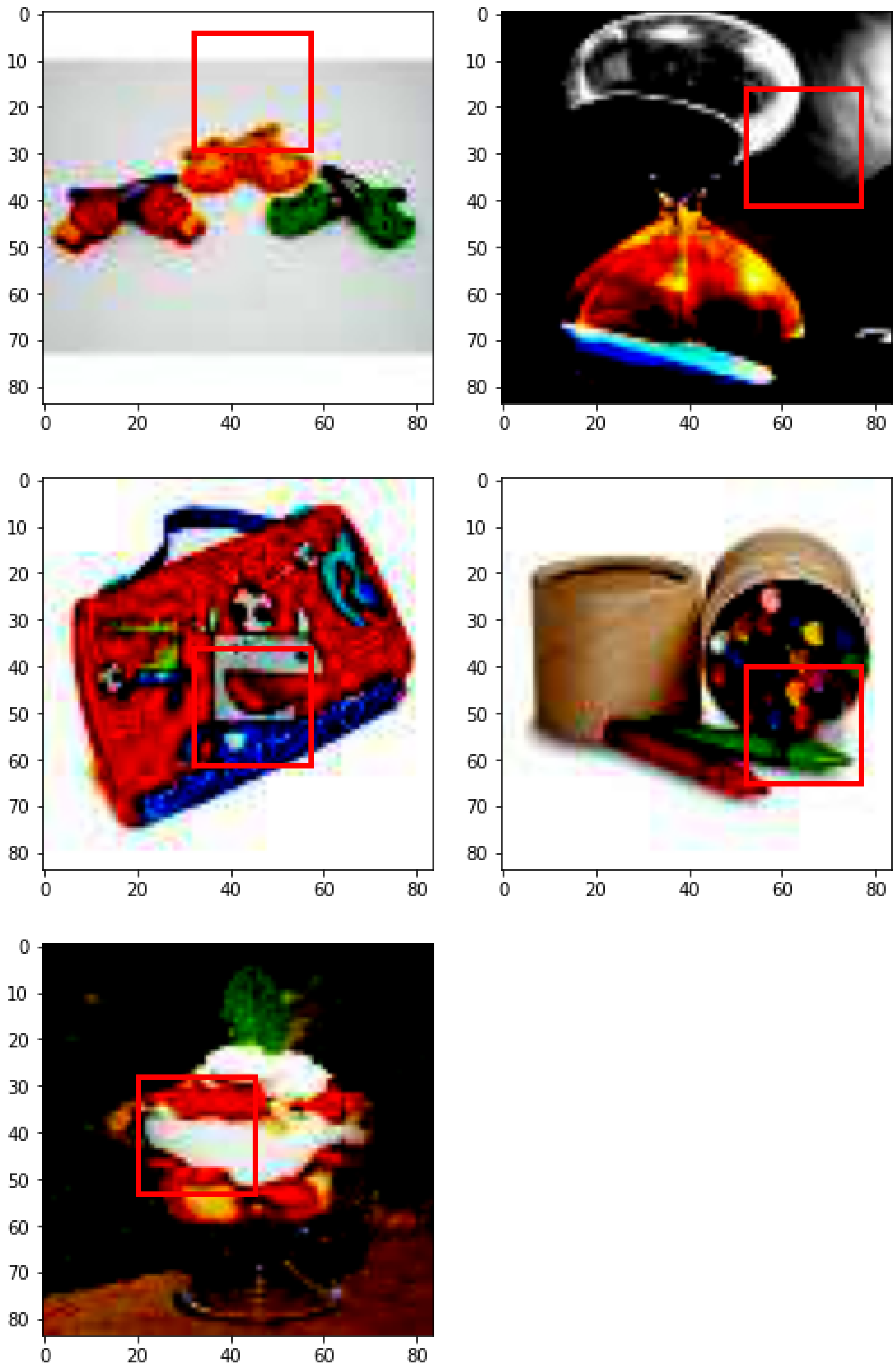
Layer3 and filter 3:



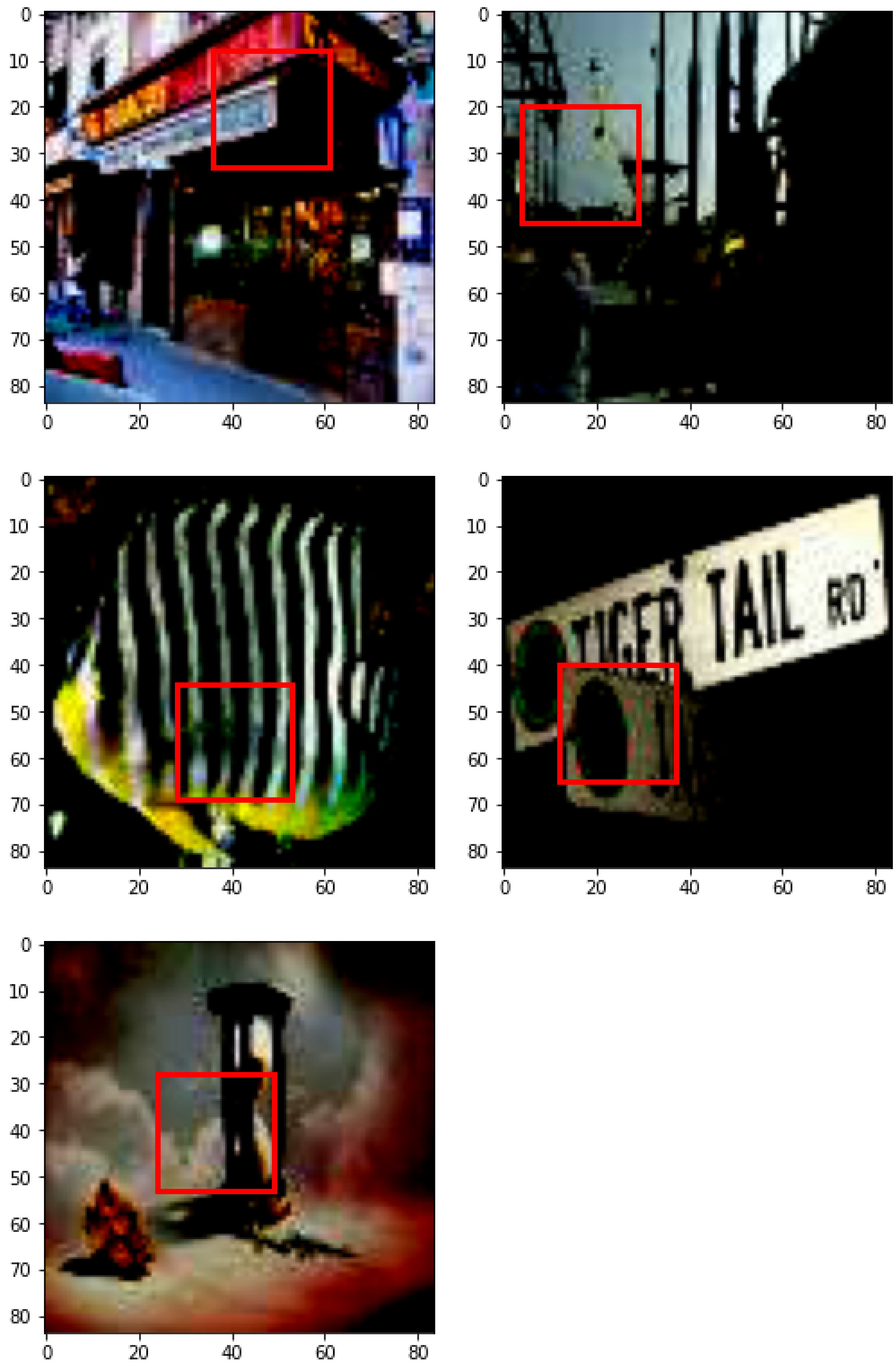
Layer3 and filter 12:



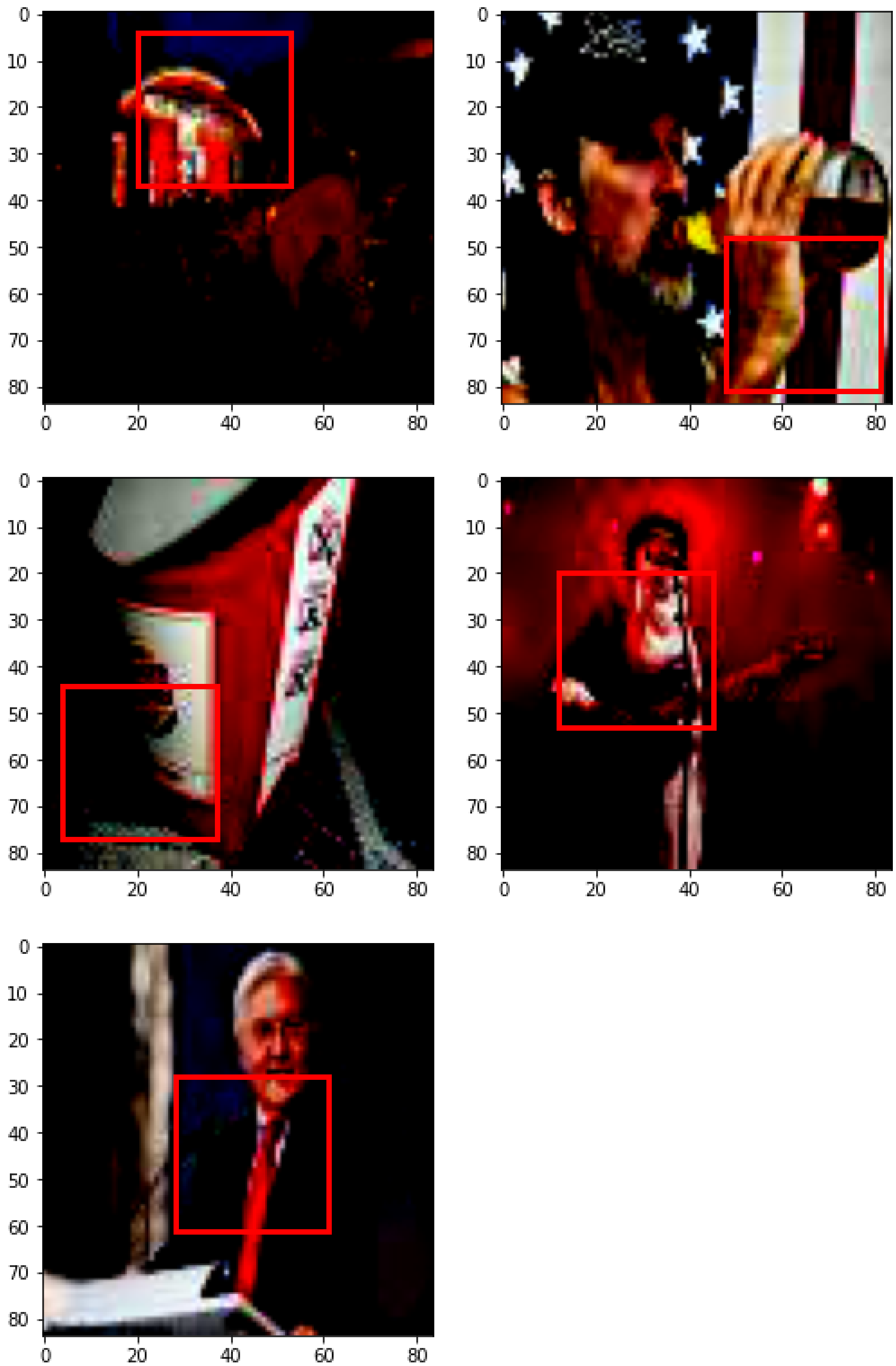
Layer4 and filter 3:



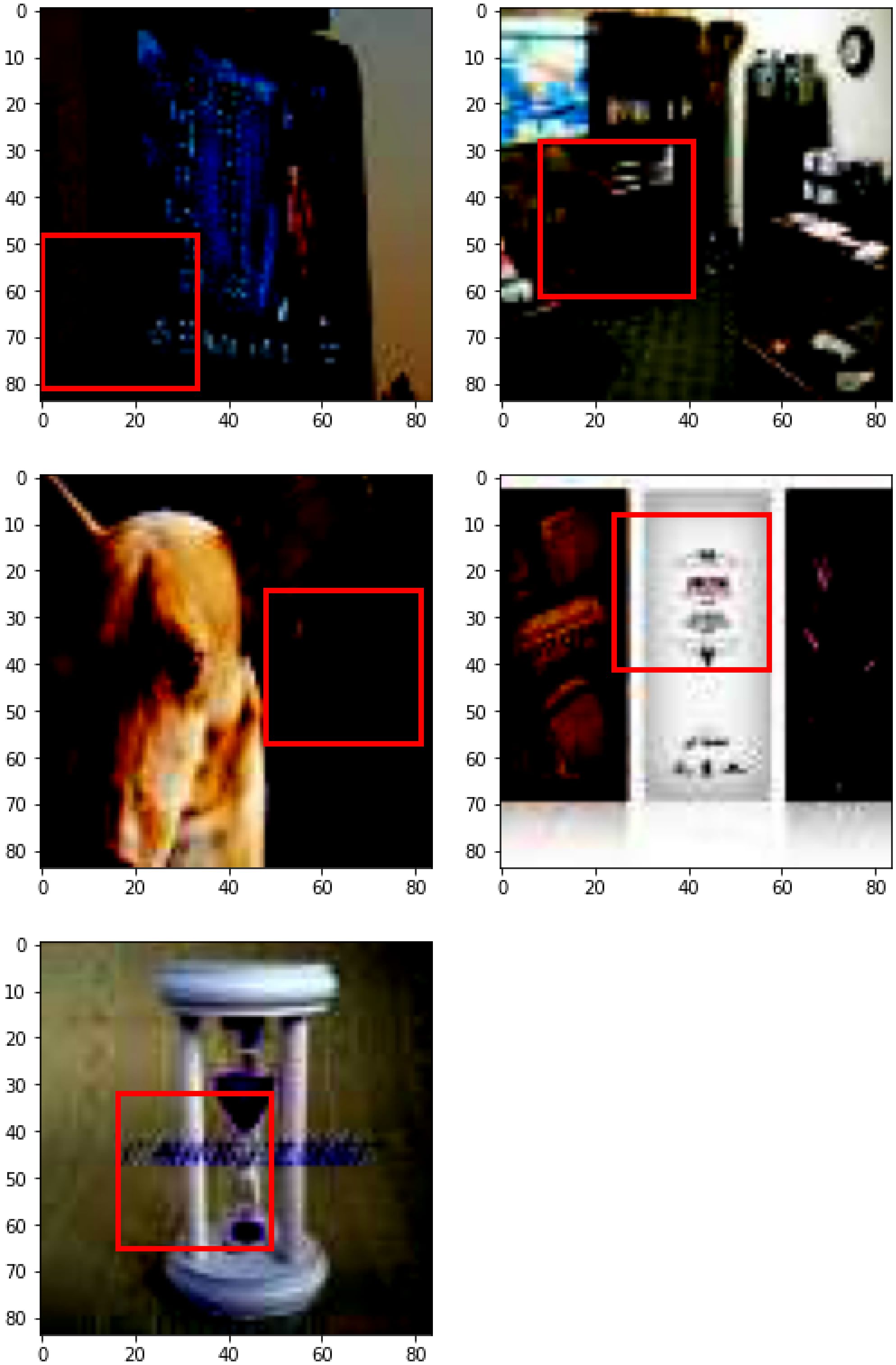
Layer4 and filter 12:



Layer5 and filter 3:



Layer5 and filter 12:



Conclusion:

The previous section gives an idea of the image patches corresponding to the maximum response of the filters and the corresponding full image.

The area looked by a filter increases as we go deeper as can be seen. The filters in the initial layers look for basic features and in the latter layers, it looks into complex features.

Part B: 2.2.2

Overview:

The same model that was used in the previous experiment is chosen which is our best model.First we observe class-wise accuracy on test dataset. We now set the weights of the random filters that was chosen to 0(thus switching them off) and again observe class-wise performance of model on the same test dataset.

Class wise Accuracy before weight change:

```
Accuracy of Ibizan_hound : 41 %
Accuracy of aircraft_carrier : 54 %
Accuracy of beer_bottle : 41 %
Accuracy of     bolete : 32 %
Accuracy of      boxer : 22 %
Accuracy of     carton : 52 %
Accuracy of       dome : 25 %
Accuracy of electric_guitar :  2 %
Accuracy of       file : 44 %
Accuracy of french_bulldog : 10 %
Accuracy of garbage_truck : 45 %
Accuracy of golden_retriever : 32 %
Accuracy of gordon_setter : 45 %
Accuracy of hair_slide : 16 %
Accuracy of  hourglass : 37 %
Accuracy of house_finch : 49 %
Accuracy of   komondor : 63 %
Accuracy of   malamute : 30 %
Accuracy of    meerkat : 45 %
Accuracy of pencil_box : 43 %
Accuracy of prayer_rug : 53 %
Accuracy of       reel :  9 %
Accuracy of rock_beauty : 71 %
Accuracy of scoreboard : 57 %
```

```
Accuracy of solar_dish : 40 %
Accuracy of      stage : 43 %
Accuracy of street_sign : 41 %
Accuracy of       tank : 46 %
Accuracy of  tile_roof : 40 %
Accuracy of tobacco_shop : 42 %
Accuracy of     trifle : 67 %
Accuracy of white_wolf : 53 %
Accuracy of       yawl : 61 %
```

Classwise Accuracy after weight change:

```
Accuracy of Ibizan_hound : 40 %
Accuracy of aircraft_carrier : 50 %
Accuracy of beer_bottle : 20 %
Accuracy of     bolete : 29 %
Accuracy of      boxer : 18 %
Accuracy of     carton : 53 %
Accuracy of       dome : 27 %
Accuracy of electric_guitar :  1 %
Accuracy of       file : 24 %
Accuracy of french_bulldog :  1 %
Accuracy of garbage_truck :  7 %
Accuracy of golden_retriever : 44 %
Accuracy of gordon_setter : 29 %
Accuracy of hair_slide : 13 %
Accuracy of  hourglass : 18 %
Accuracy of house_finch : 56 %
Accuracy of   komondor : 57 %
Accuracy of   malamute : 21 %
Accuracy of    meerkat : 55 %
Accuracy of pencil_box : 54 %
Accuracy of prayer_rug : 66 %
Accuracy of       reel :  5 %
Accuracy of rock_beauty : 69 %
Accuracy of scoreboard : 13 %
Accuracy of solar_dish : 24 %
Accuracy of      stage : 24 %
Accuracy of street_sign : 20 %
Accuracy of       tank : 31 %
Accuracy of  tile_roof : 40 %
Accuracy of tobacco_shop : 27 %
Accuracy of     trifle : 54 %
Accuracy of white_wolf : 60 %
Accuracy of       yawl : 49 %
```

Conclusion:

When we switch off filter the chosen 10 filters there is an abrupt change in accuracy of classes Yawl. Accuracy of classes aeroplane_carrier and trifle have also reduced but through very smaller margin. Since we observed the maximum activation of the chosen filters of deepest layer was corresponding to features of yawl and hourglass in image. Although some middle layer filters picked up features of aeroplane_carrier ,meerkat their result do not seem much affecting by switching off these filters. This indicates that model performance is strongly dependant of deepest layers' feature extracted than those at the top. Interestingly class wise accuracy prediction of prayer_rug have increased by a considerable amount. This might be because of the fact that since filters activating neurons for prediction of other classes are now zeroes, hence weightage of activation neurons of prayer_rug features would increase and hence the model predicts cat with higher probability for large number of images among which several would have been misclassified before.