CS6910: Programming Assignment 1

Vimarsh Sathia
CS17B046

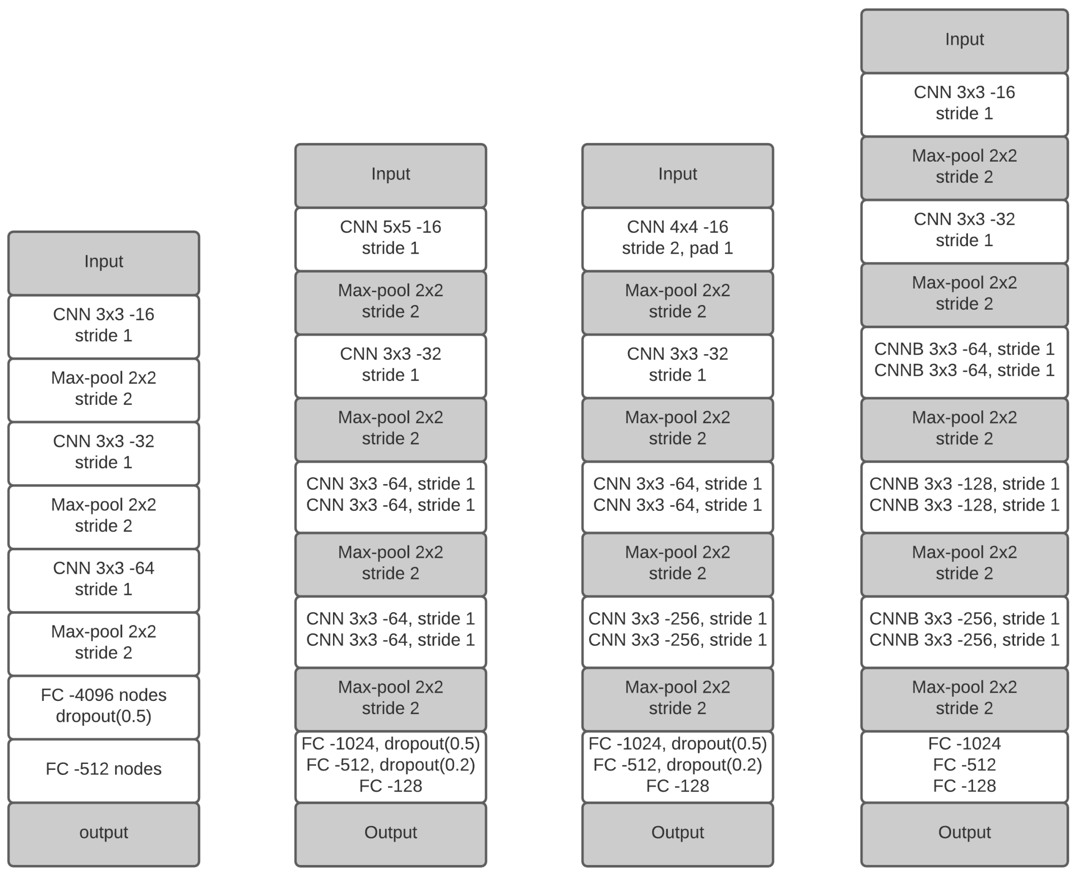
Indian Institute of Technology Madras — October 20, 2020

General Information

All image classification experiments were carried out on the subset of the CIFAR-10 dataset provided in split 2. Each input is a $3 \times 64 \times 64$ RGB image.
During the training phase in Part-A, batch size was taken to be 32 for all 4 models.

1 Part-A

For this part, 4 different models(Net1, Net2, Net3 and Net4) with different network parameters were trained. The code defining the 4 neural network modules can be found in `classifiers.py`. A brief summary of the network layers is shown in fig. 1.



(a) Net1 (b) Net2 (c) Net3 (d) Net4

Figure 1: Summary of all trained models. CNNB refers to batch normalization before applying ReLU

1.1 Training Results

The following accuracy results were obtained after training all 4 models, which were trained for a total of 50 epochs. Each model was checkpointed every 10 epochs, and the model with the best test accuracy was chosen as the final model for Part-B. The accuracies for all models are captured in table 1. The loss curve for the best performing model(Net 4) is present in fig. 2.

Table 1: Test and Validation Accuracy Chart(for all 4 models)

Model	epochs	validation accuracy(%)	test accuracy(%)
Net1	10	70.16	70.28
	20	75.88	74.64
	30	78.16	76.2
	40	76.92	76.68
	50	77.8	77.16
Net2	10	36.04	35.84
	20	63.12	61.44
	30	73.56	70.32
	40	76.16	73.16
	50	79.4	76.16
Net3	10	39.84	39.6
	20	61.2	59.4
	30	70.24	69.16
	40	75.84	74.44
	50	77.32	75.48
Net4	10	78.2	76.96
	20	81.4	79.92
	30	**83.4**	**82.24**
	40	83.84	82.2
	50	83.48	81.96

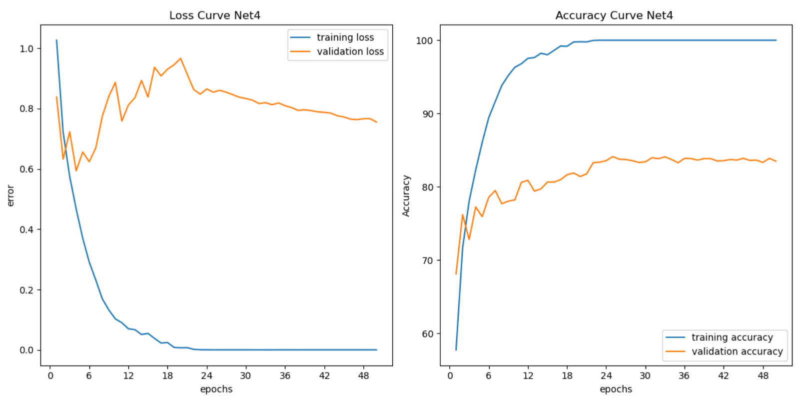


Figure 2: Loss curves for Net 4 (best model)

1.2 Training Inferences

From the figures in table 1 and the models in fig. 1, we can make the following conclusions about hyperparameter effects on performance:

1. **Convolutional Layers**: In general, an increase in the number of convolutional layers lead to better train and test accuracy. However, for networks with very large number of conv layers, the train speed is very slow(as evidenced in Net2 and Net3). This can be offset by applying batch normalization after every conv layer(as evidenced in Net4).

Table 2: Classwise accuracy on test dataset for Net4 - 30 epochs

Class	Accuracy(in %)
aeroplane	96
cat	65
deer	88
dog	70
frog	84

2. **No of filters/layer**: In general, having more number of filters in higher layers helped capture more detailed features, and led to a higher accuracy(as evidenced in Net3 and Net4)
3. **Stride**: Convolving with stride ≥ 1 seems to result in a reduced accuracy, as evidenced between Net2 and Net3. This happens because we miss out on features when increasing stride.
4. **Maxpooling**: Increasing no of maxpooling layers in shallow nets(Net1) leads to greater loss.

In fig. 3, we can see a small subset of the misclassified images in the dataset split. In most of the images, it is hard to classify the images even with the naked eye.

Figure 3: Some misclassified images

2 Part-B

For this part, all experiments are conducted with the weights learned after epoch 30 of Net4 from Part-A.

2.1 Occlusion sensitivity experiment

Occlusion experiments were carried out on 5 random samples from the dataset, with a window size of 10×10 and 20×20 respectively. Some example images are visualized in fig. 4.
From the heatmap outputs, we can infer that increasing the occlusion kernel size around the main features of the objects to be classified causes the probability of misclassification to increase(increase in blue shading).

Table 3: Filter Selection in Net4

Conv Layer(sub-block #)	Filter #s
1	2 , 13
2	11 , 23
3-1	9 , 49
4-1	54 , 98
5-1	5 , 171

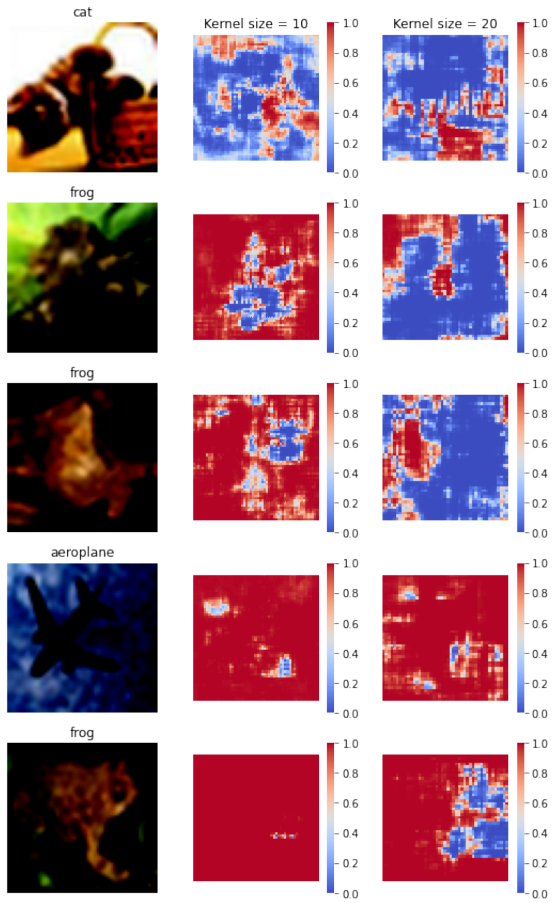


Figure 4: Images with probability heatmaps for kernels of size 10×10 and 20×20 for the occlusion sensitivity experiment

2.2 Filter Analysis

2.2.1 Filter Identification

In Net 4, the filters specified in table 3 were selected for further analysis. Some of these selected filters are visualized in fig. 5 to fig. 14.

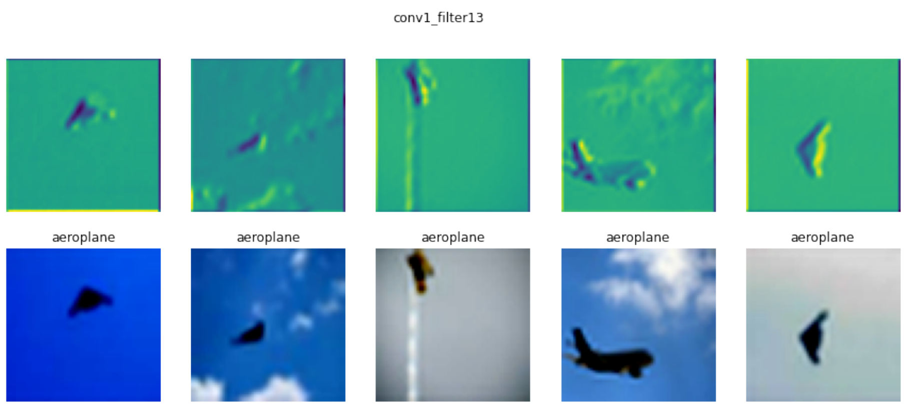


Figure 5: 13^{th} filter output corresponding to 1^{st} conv layer of Net4

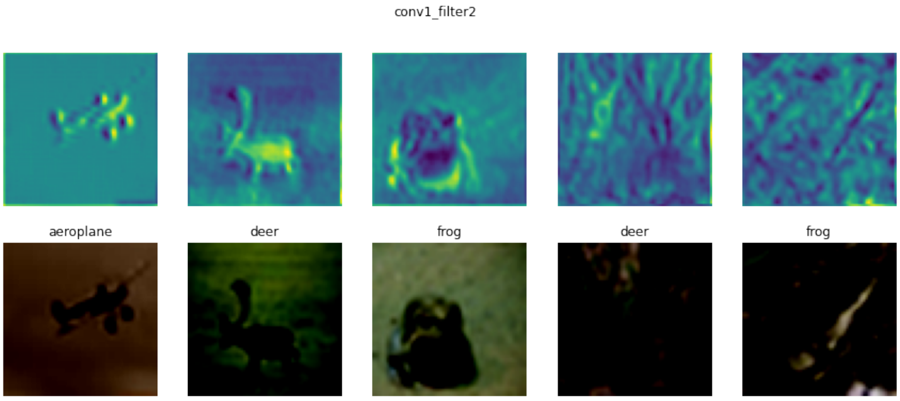


Figure 6: 2^{nd} filter output corresponding to 1^{st} conv layer of Net4

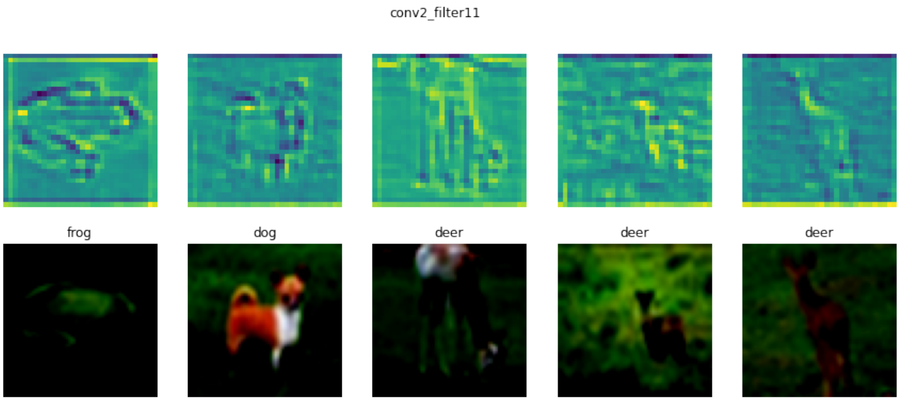


Figure 7: 12^{th} filter output corresponding to 2^{nd} conv layer of Net4

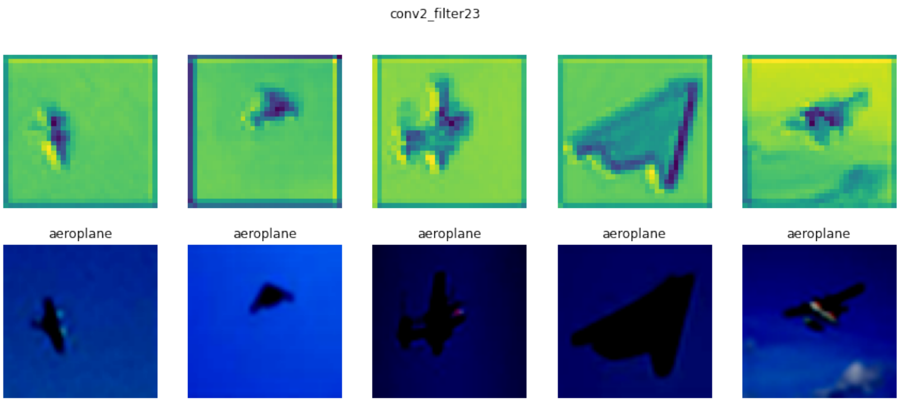


Figure 8: 23^{rd} filter output corresponding to 2^{nd} conv layer of Net4

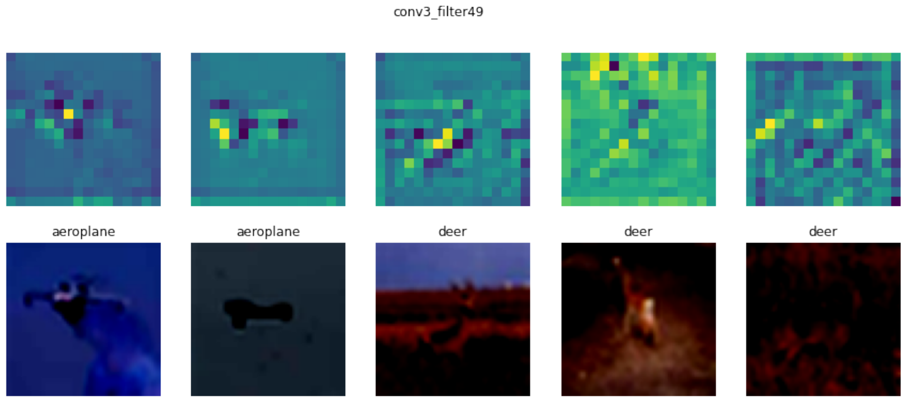


Figure 9: 49^{th} filter output corresponding to 3^{rd} conv layer of Net4

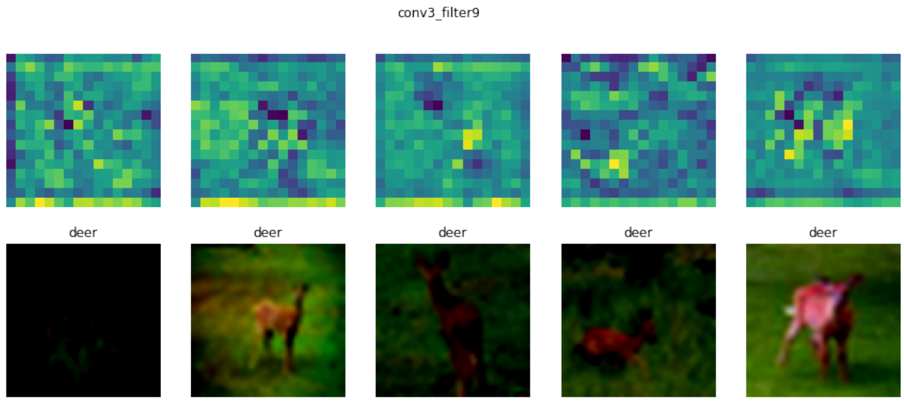


Figure 10: 9^{th} filter output corresponding to 3^{rd} conv layer of Net4

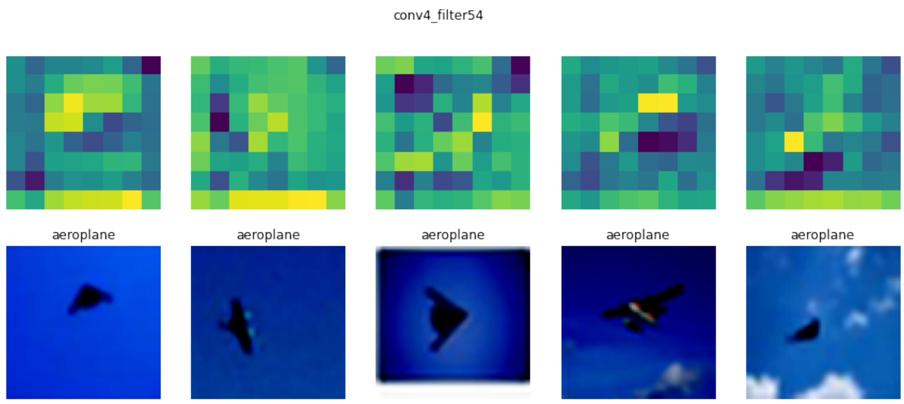


Figure 11: 4^{th} filter output corresponding to 54^{th} conv layer of Net4

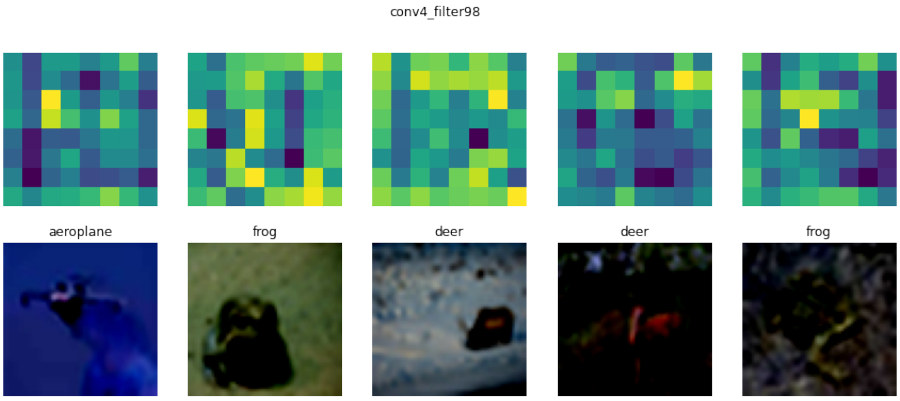


Figure 12: 4^{th} filter output corresponding to 98^{th} conv layer of Net4

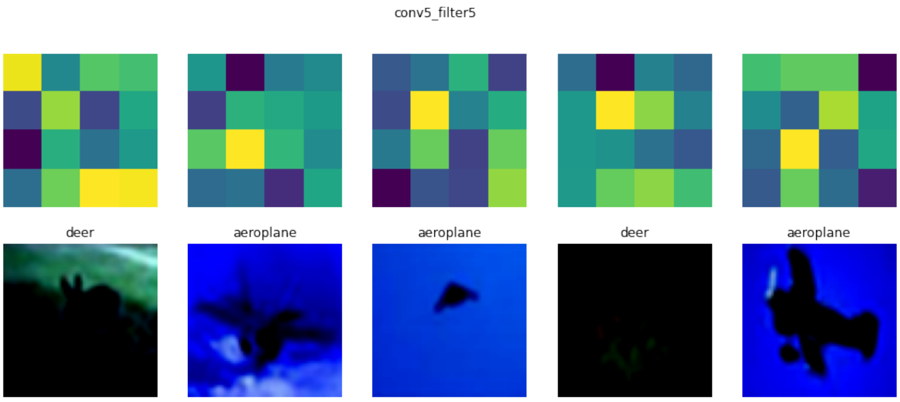


Figure 13: 5^{th} filter output corresponding to 5^{th} conv layer of Net4

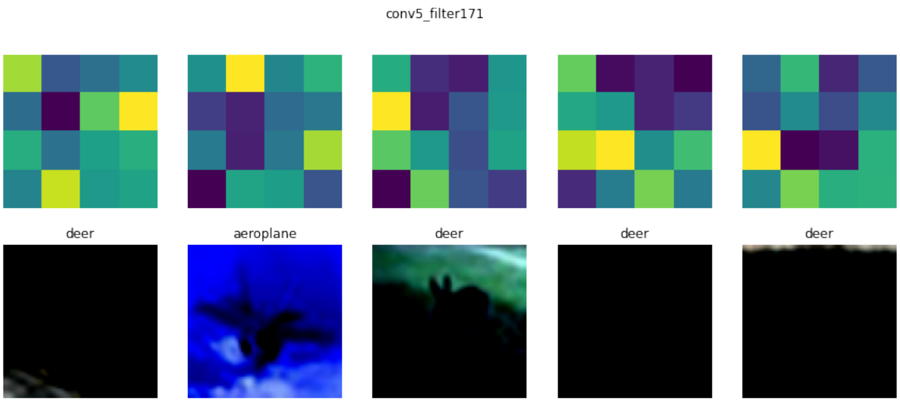


Figure 14: 171^{st} filter output corresponding to 5^{th} conv layer of Net4

2.2.2 Filter Modification

The stats of the images which mis-classify after switching off the weights of the filters specified in table 3 is captured in table 4

By viewing the values in table 4, we can roughly conclude that the net effect of the filters in table 3 is

Table 4: Misclassified image breakup after switching off filters present in table 3

Class	Misclassification count
aeroplane	7
cat	31
deer	18
dog	50
frog	12

to identify a common feature among cats, dogs and deers, which might be a form of quadrupedalism. fig. 15 shows some of the reclassified images after switching off the weights in table 3.

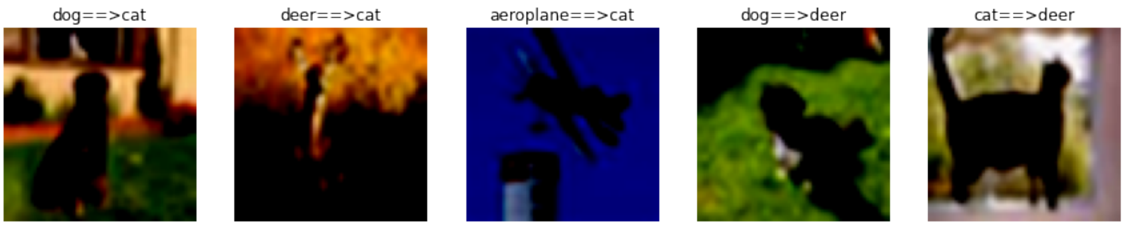


Figure 15: Some sample images reclasified after switching off weights