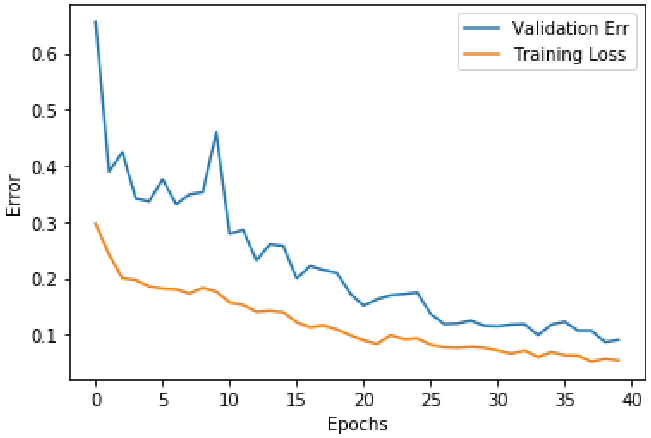
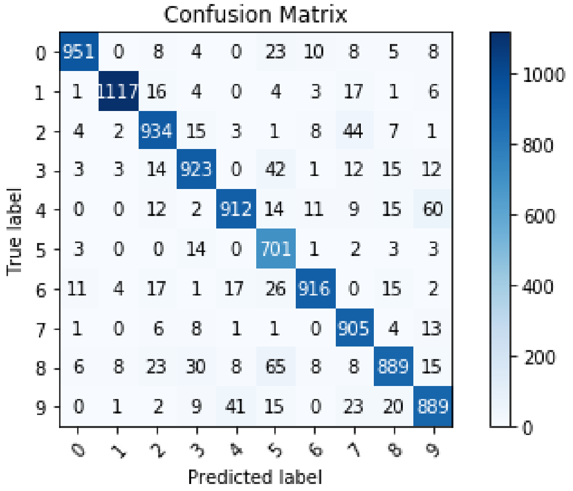
MNIST:

This model contained single Convo Layer with 5 filters of 9x9 and a max pool Layer with Fully Connected Layers having (500 → 256 → 128 → 10) neurons each with sigmoid activation except last which has SoftMax. It achieved **94% accuracy with 40 epochs and learning rate of 0.001**





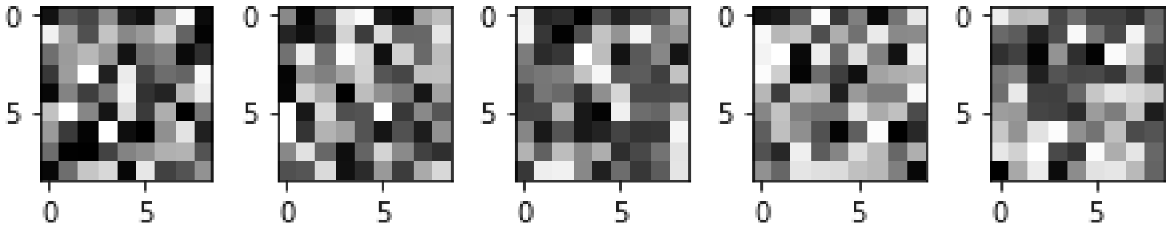


Figure 1: Before Training

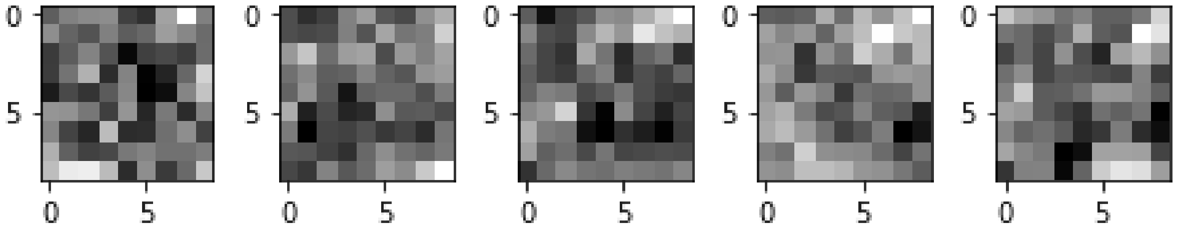
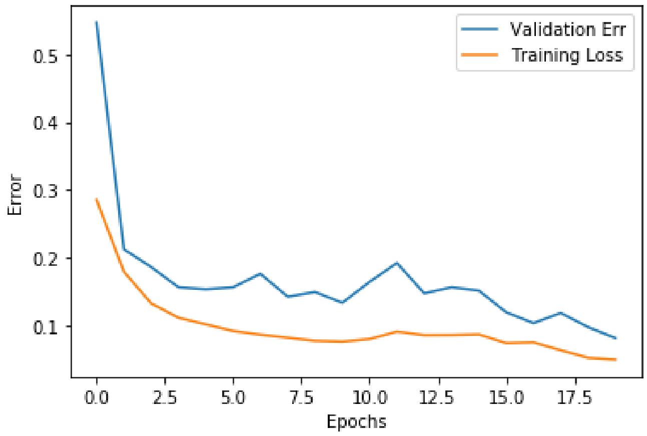
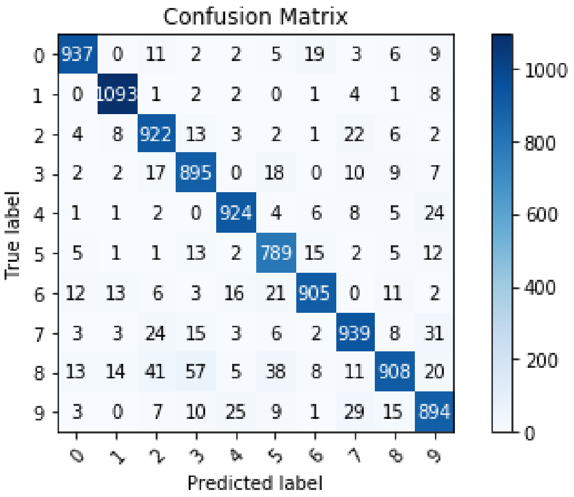


Figure 2: After Training

These trained filters don't match much with the filters used in Assignment 3 may be due to very few filters. Many filters may have merged into one during training.

This next model also has single Convo Layer with 10 filter of 9x9, a Max Pool Layer and fully connected layer of (1000 → 256 → 128 → 10) neurons. This model achieved **95% accuracy with 20 epochs and learning rate of 0.001.** Now in this some filters starting to look like the filters used in the Assignment 3, like the 7th filter below it is similar to the edge detecting filter in slanting position.





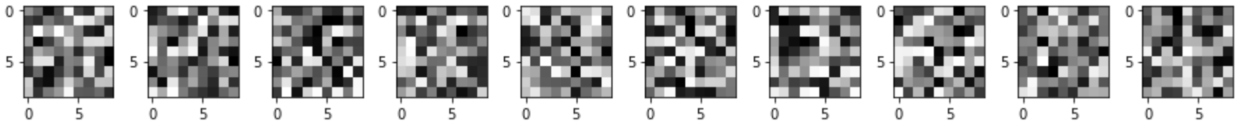


Figure 3: Before Training

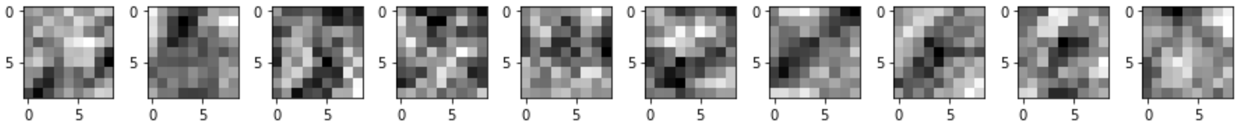


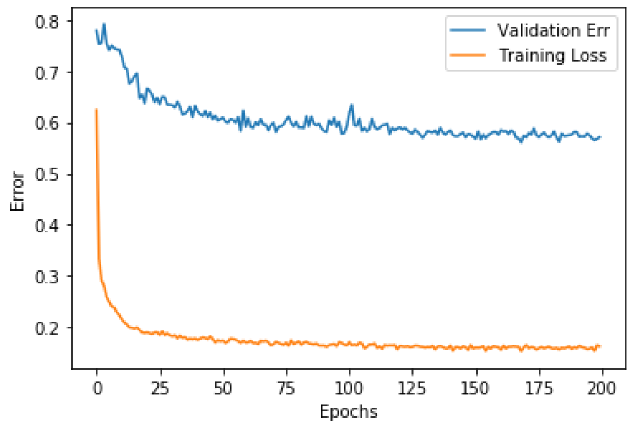
Figure 4: After Training

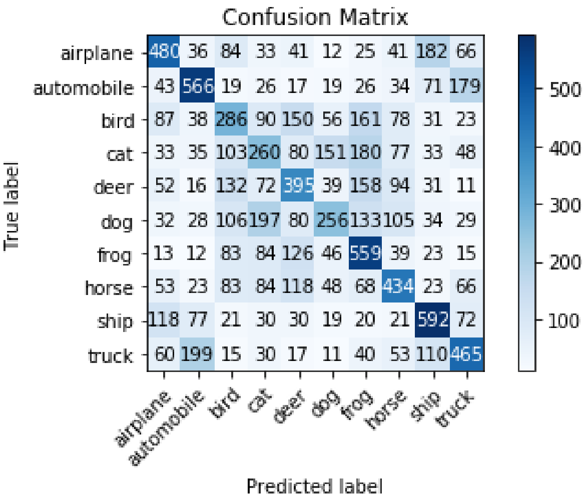
As It is seen that with double filters almost same accuracy was achieved in half the epochs and the loss curves also become smoother. I didn’t used whole dataset in each epoch rather I implemented a batch system in which each epoch only sees 1500 samples randomly selected from the data set due to lack of computation power.

CIFAR-10:

The model contained single Convo Layer with 48 filters of 3x3 and 3 individual channels, a Max Pool Layer and fully connected Layer of (10800 → 128 → 10) neurons

This attained **accuracy of 45% in 100 epochs** and then kind of just stuck there with **learning rate of 0.0001 for first 100 epochs** and then changed to 0.0005 for the next 100 epochs. The activation function in these are ReLu as it gave better results than sigmoid.





As I used all the channels of the filters individually so there they are trained separately and have different behaviors on different channels of the Input. These filters are seeming to be similar to the filters used in Assignment 3 but the size of smaller here. Many filters look like the single and double derivatives of the input with directions horizontal vertical and diagonal.

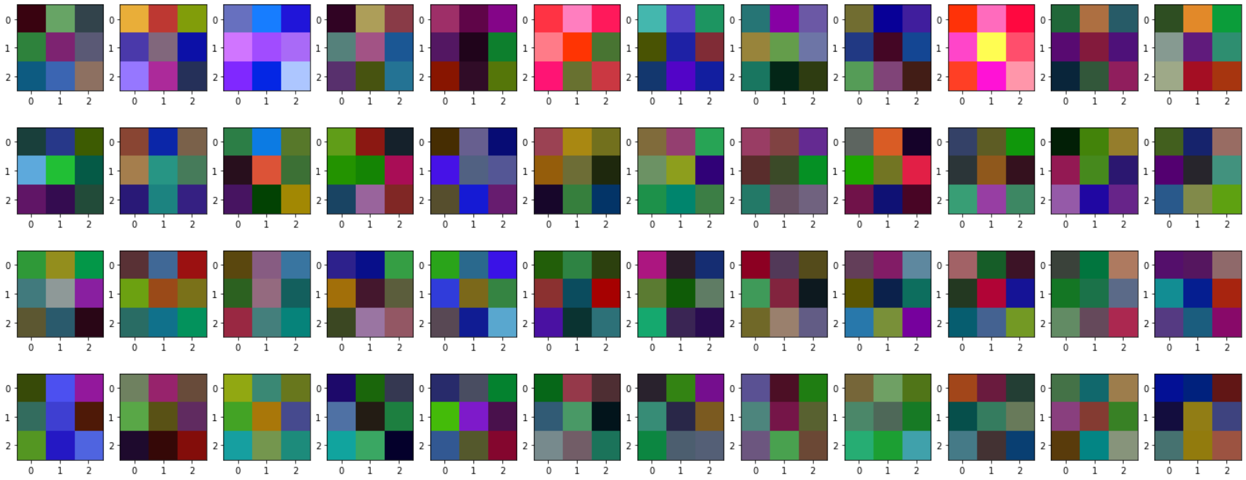


Figure 5: All Channels Combined

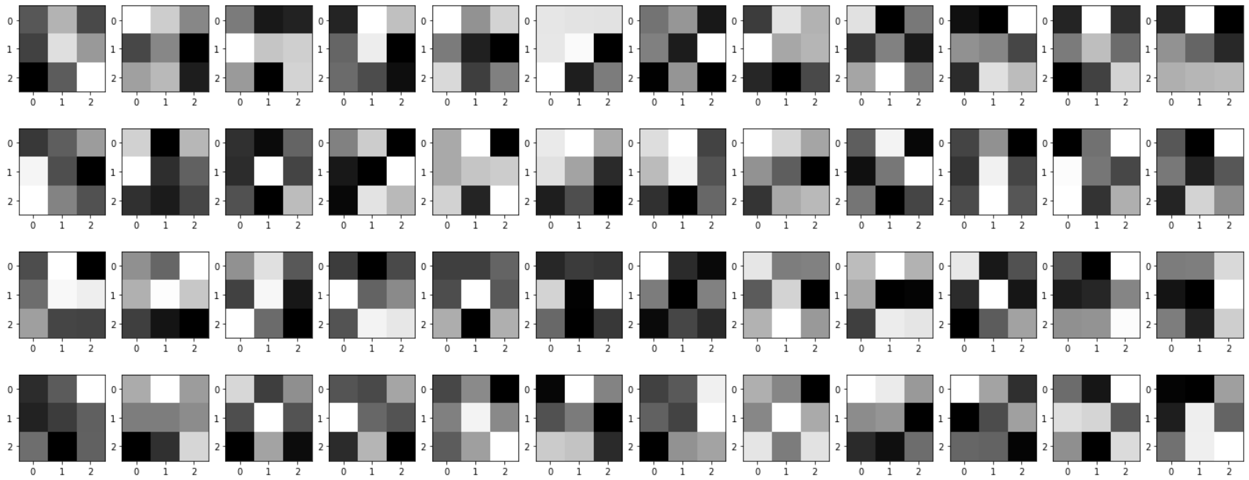


Figure 6: Channel 1

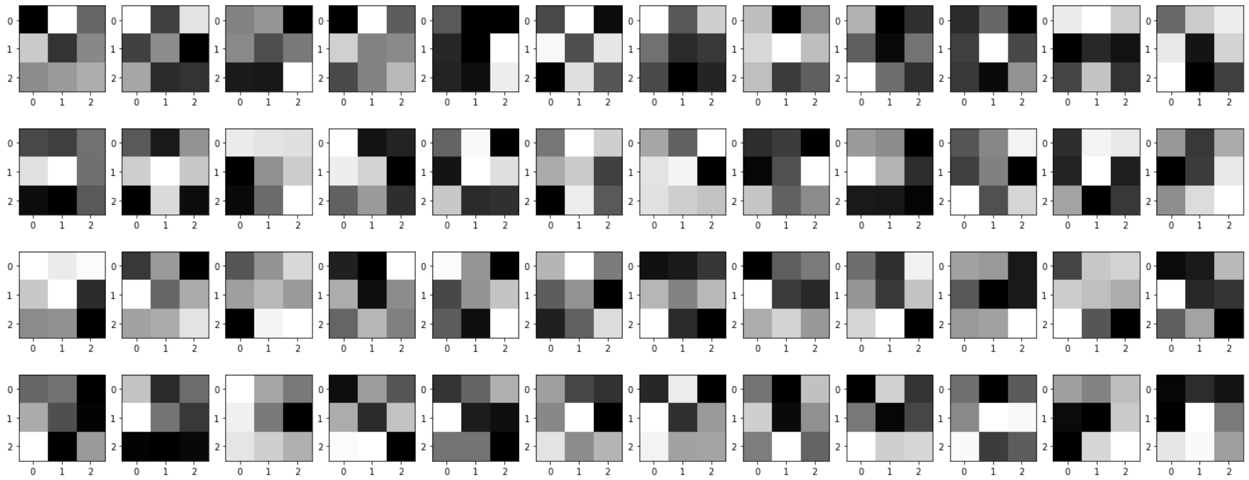


Figure 7: Channel 2

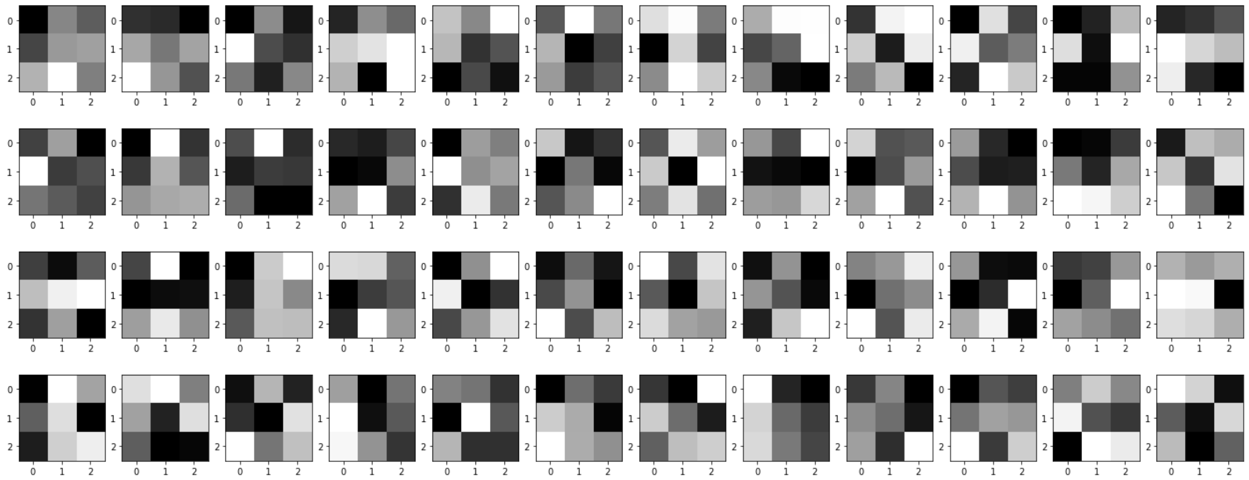
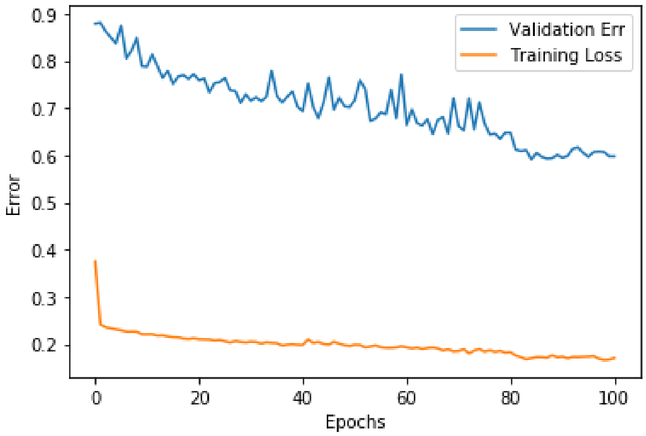
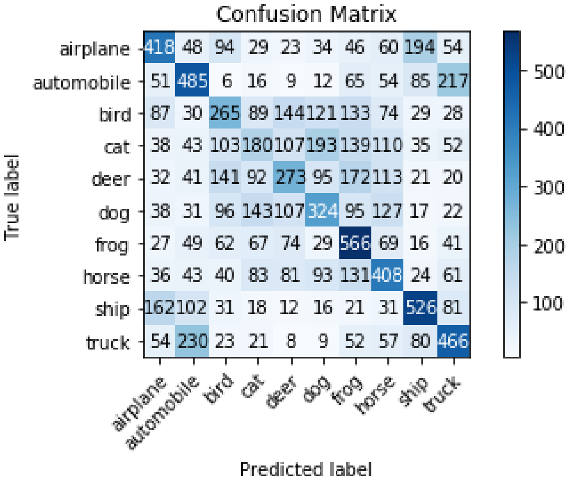


Figure 8: Channel 3

I also trained a model by using same filter over all the channels. This model also attained **accuracy of 45% in 100 epochs** with **learning rate of 0.0001.** Here the loss curve is steep at first but then the it decreases almost linearly. But the validation error oscillated very much. The confusion matrix is also almost similar.

This model contained two Convo Layer with 32 filters of 3x3 each same filter over all channels, a Max Pool Layer and fully connected Layer of (6272 → 128 → 10) neurons with ReLu and SoftMax.





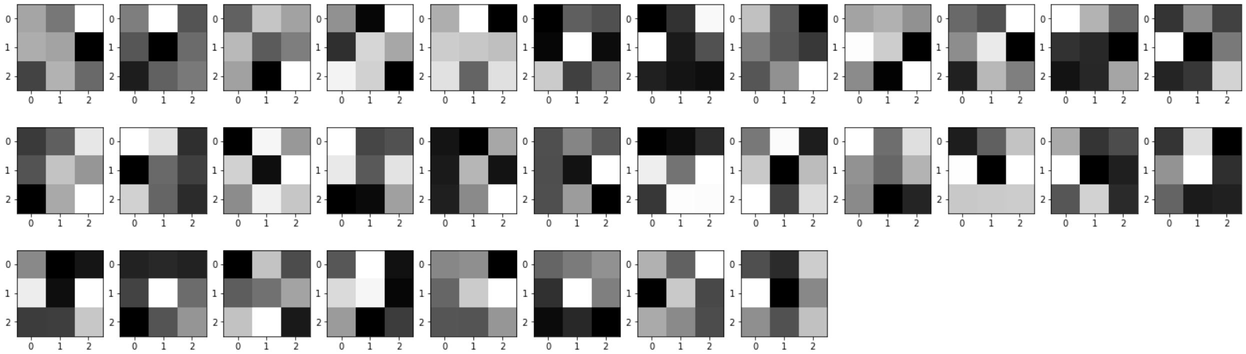


Figure 9: First Convo Layer

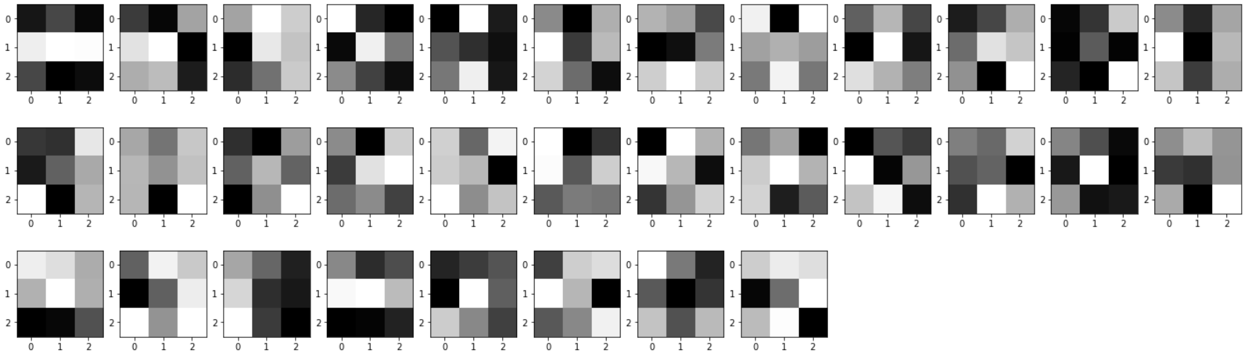
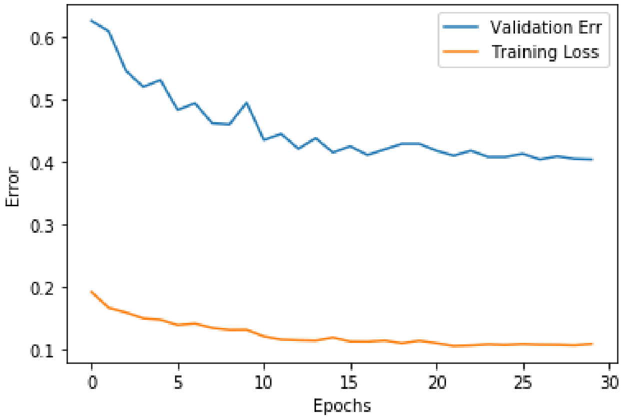


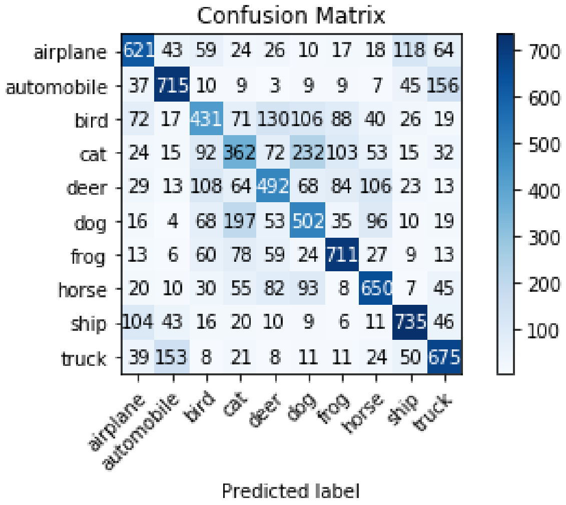
Figure 10: Second Convo Layer

The Second Layer seems to have much of the same filters from the filter bank like the first seems a second derivative filter and the 12th one seems a single derivative filter.

My guess behind the reason of accuracy getting stuck at 45% is may be the gradients become vanishing as they pass to the convo layers because I have seen that it is not actually stuck at 45% but it improves very slowly after 40% as you can see that about 40% accuracy in the first model was achieved in about 50-60 epochs but after that to reach 45% I had to make 100 epochs more. May be If Momentum or other optimizers were used It could increase the learning. And also, as I am not using the entire data set in each epoch It oscillates a lot and could cause the learning to slow down.

So, I created Another model with three convo Layers with 32 filters of 5x5 each and initialized the weights with gaussian random. And FC layer (112 →128 → 10) with relu activations. It gave me **59% accuracy in 30 epochs. The learning rate for first 10 epochs were 0.001 and then for next 15 epochs 0.0002 and then for next 10 epochs 0.00001.**





	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	621	43	59	24	26	10	17	18	118	64
automobile	37	715	10	9	3	9	9	7	45	156
bird	72	17	431	71	130	106	88	40	26	19
cat	24	15	92	362	72	232	103	53	15	32
deer	29	13	108	64	492	68	84	106	23	13
dog	16	4	68	197	53	502	35	96	10	19
frog	13	6	60	78	59	24	711	27	9	13
horse	20	10	30	55	82	93	8	650	7	45
ship	104	43	16	20	10	9	6	11	735	46
truck	39	153	8	21	8	11	11	24	50	675

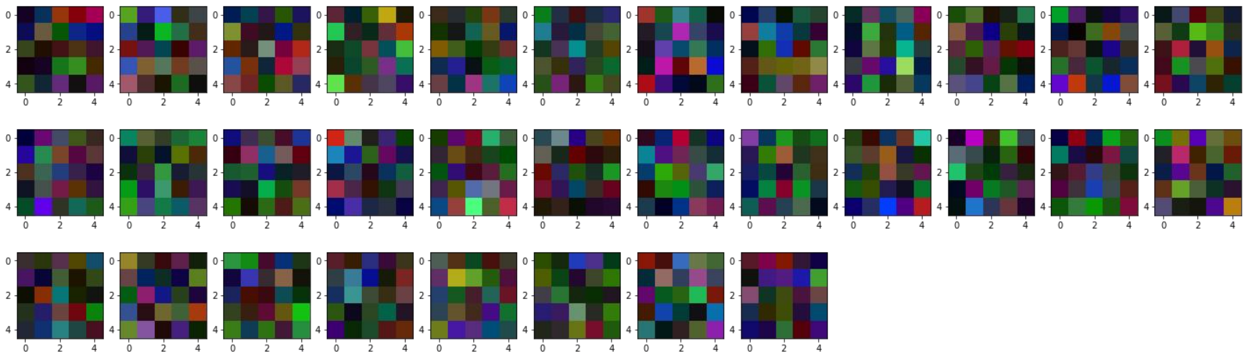


Figure 11: all channel combined

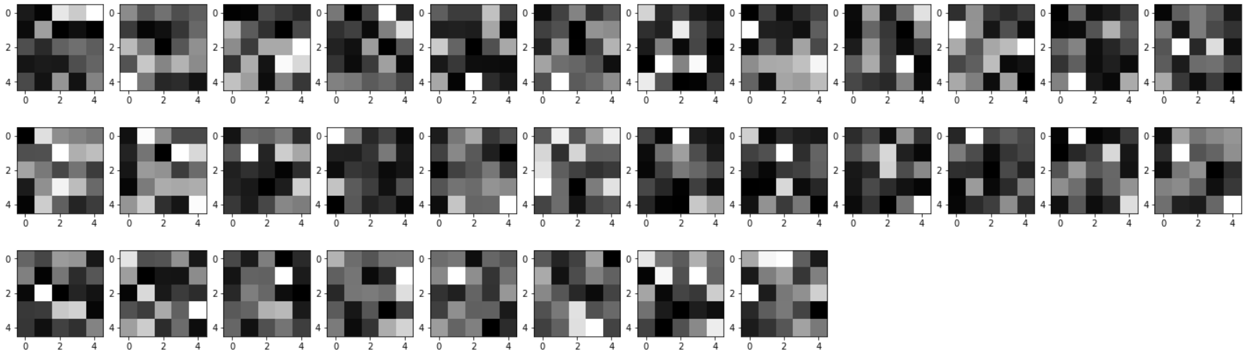


Figure 12: channel 1

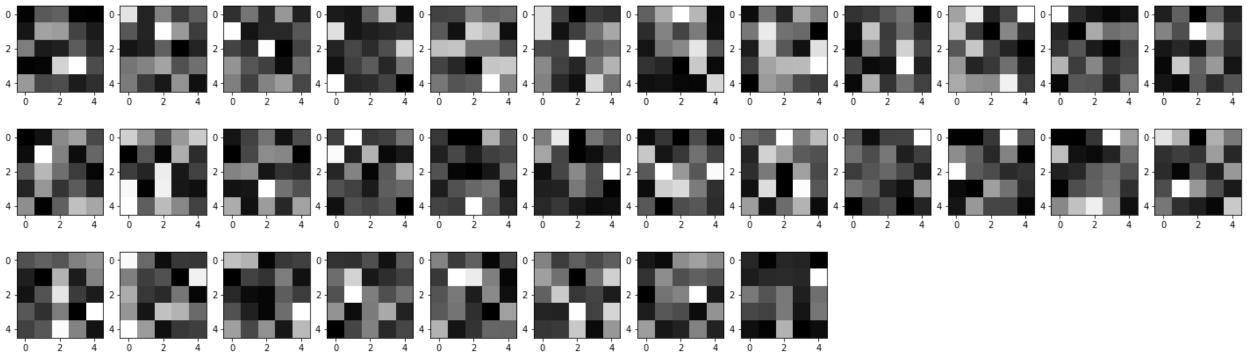


Figure 13: channel 2

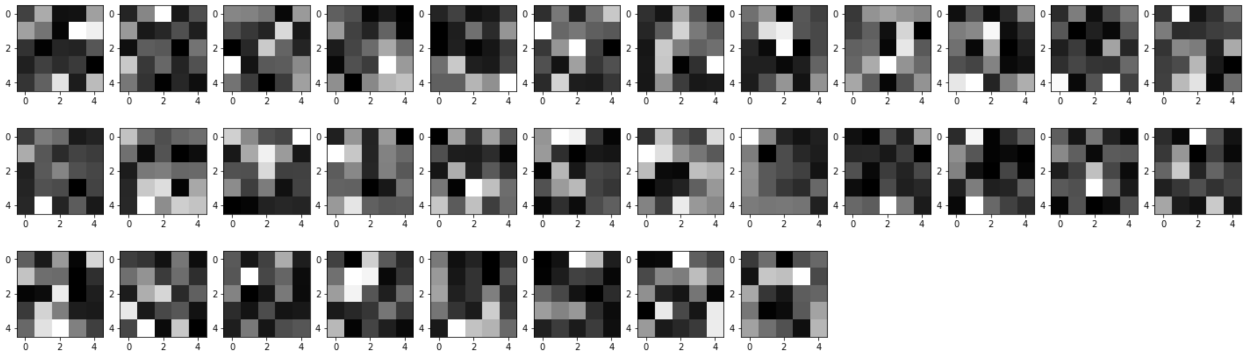


Figure 14: channel 3

So by increasing the receptive field of the filters I was able to get more accuracy.