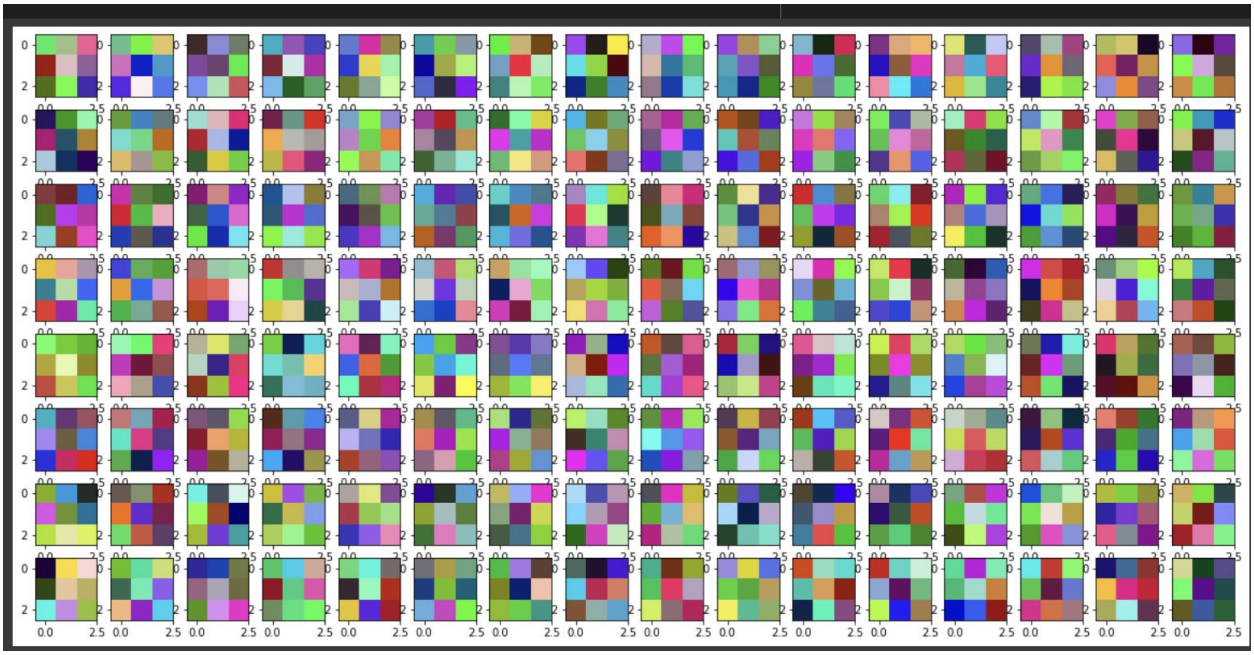
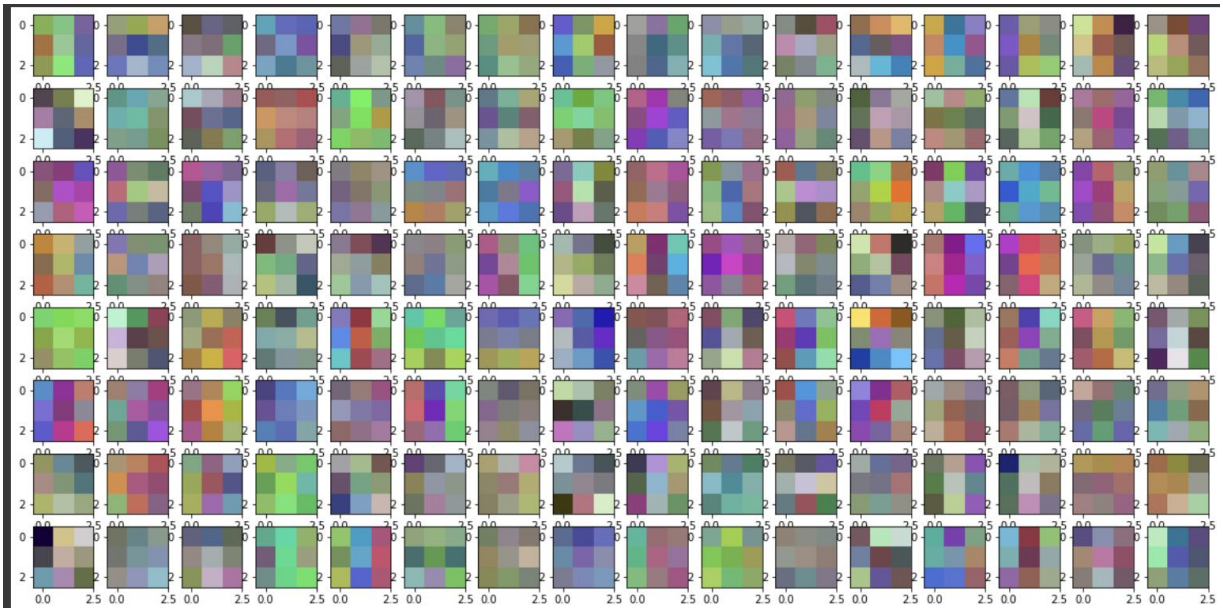
Q1 a) The final loss = 0.0891 (this has not been the smallest throughout all the epochs)
The final validation accuracy = 81.7 % (the highest accuracy in the whole training)
b) The number of parameters = 7678474
c) These are the filters before the training. What is distinct now is that all the pixels within filters are quite bright. Also, the collection of colors seems to follow the pattern of randomness meaning that there does not seem to be any correlation between pixels within filters.

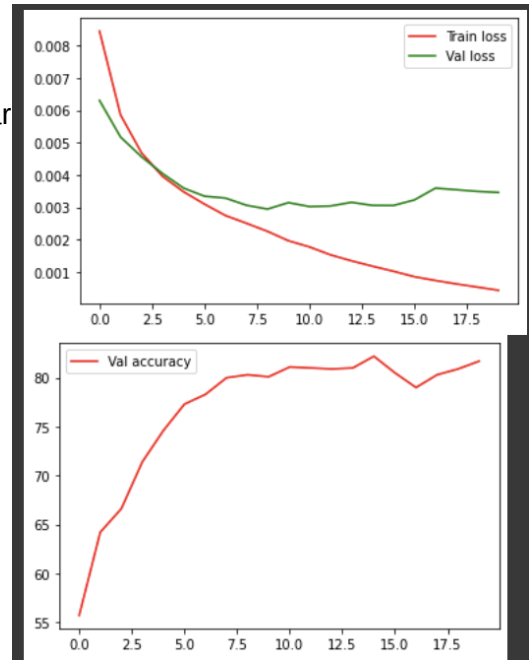


What is drastically different in the filters after the training is that almost all the pixels became more blurry. Also, it is possible to see lots of similar pixels within filters clearly indicating trend.

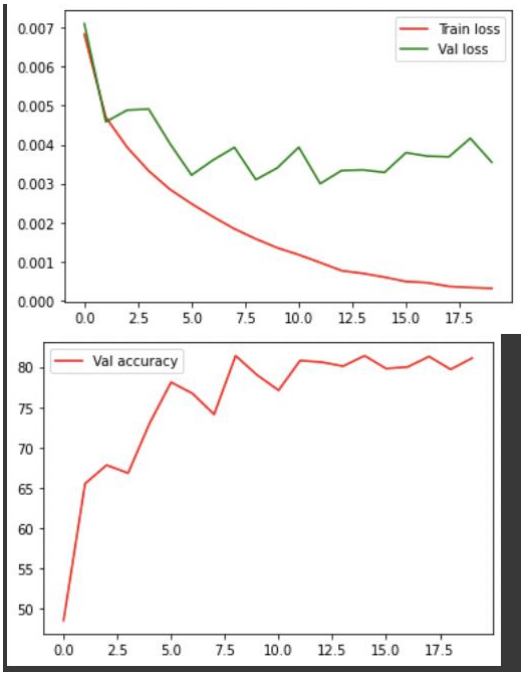


Q2

a) The graph to the right is a model without batch normalization. It can be seen here that after some time it starts to overfit and validation loss does not go down anymore. Furthermore, validation accuracy starts to oscillate and it is not clear if it will go down



In this graph with norm., it can be seen that validation loss oscillates extensively on average going down. In case of the validation accuracy, it seems to go up with a more clear pattern into increasing.



b) Without batch normalization:
With early stopping: the best validation accuracy
Without early stopping: the last validation accuracy is 81.7 %

Batch normalization:
With early stopping: the best validation accuracy is 85.0 %, epoch 49.
Without early stopping: the last validation accuracy is 81.1%

Question 3.

a) We performed several experiments to identify best hyper-parameters for data augmentation. We selected horizontal flip, random rotation, color jitter, and random crop. For 20 epochs, with RandomHorizontalFlip(p=0.5) and RandomRotation(50) we got **64.8%** accuracy, while with RandomHorizontalFlip(p=0.5), RandomRotation(50), ColorJitter(brightness=0.1, contrast=0.2, saturation=0, hue=0), RandomCrop(32, padding=4) we got **67.1%**.
b) We added Dropout between all conv blocks, and after experimenting with probabilities, and position in the model (before or after maxpooling layer) we got results worse in validation error, comparing on results of training on **20 epochs without data augmentation.**
But adding dropout layer only on 5th layer improved the result. The results can be seen in the table below:

Model	Validation accuracy
No dropout	82.8%
Dropout (between all conv) p=0.2	76.7%
Dropout (5th layer) p=0.2	85.9%
Dropout (5th layer) p=0.5	82.9%
Dropout (5th layer) p=0.3	83.3%

Final comparison (**50 epochs**):
Model after Q2 – 85.0%
Model after Q2 with Dropout and Augmentation – 83.0%

Summary: With Augmentation we got less score than without, adding the dropout on 5th layer helped us improve score by 0.9%

Question 4.

a) After completing the code we got accuracy of - 64.8%
b) Without augmentation (30 epochs):
Pretrained model without updating parameters (fine_tune=False) - 10.4%
Model without loading ImageNet weights (fine_tune=True) – 65%

With augmentation (30 epochs):

Pretrained model with finetuning with augmentation - 53.3%

Summary: augmentation here also made score less than without. The best score with VGG architecture was without loading the pretrained weights.