

HLCV EXERCISE 3 REPORT

AKSHAY JOSHI

2581346

s8akjosh@stud.uni-saarland.de

ANKIT AGRAWAL

2581532

s8anagra@stud.uni-saarland.de

SUSHMITA NAIR

2581308

s8sunair@stud.uni-saarland.de

Question 1: Implement Convolutional Network

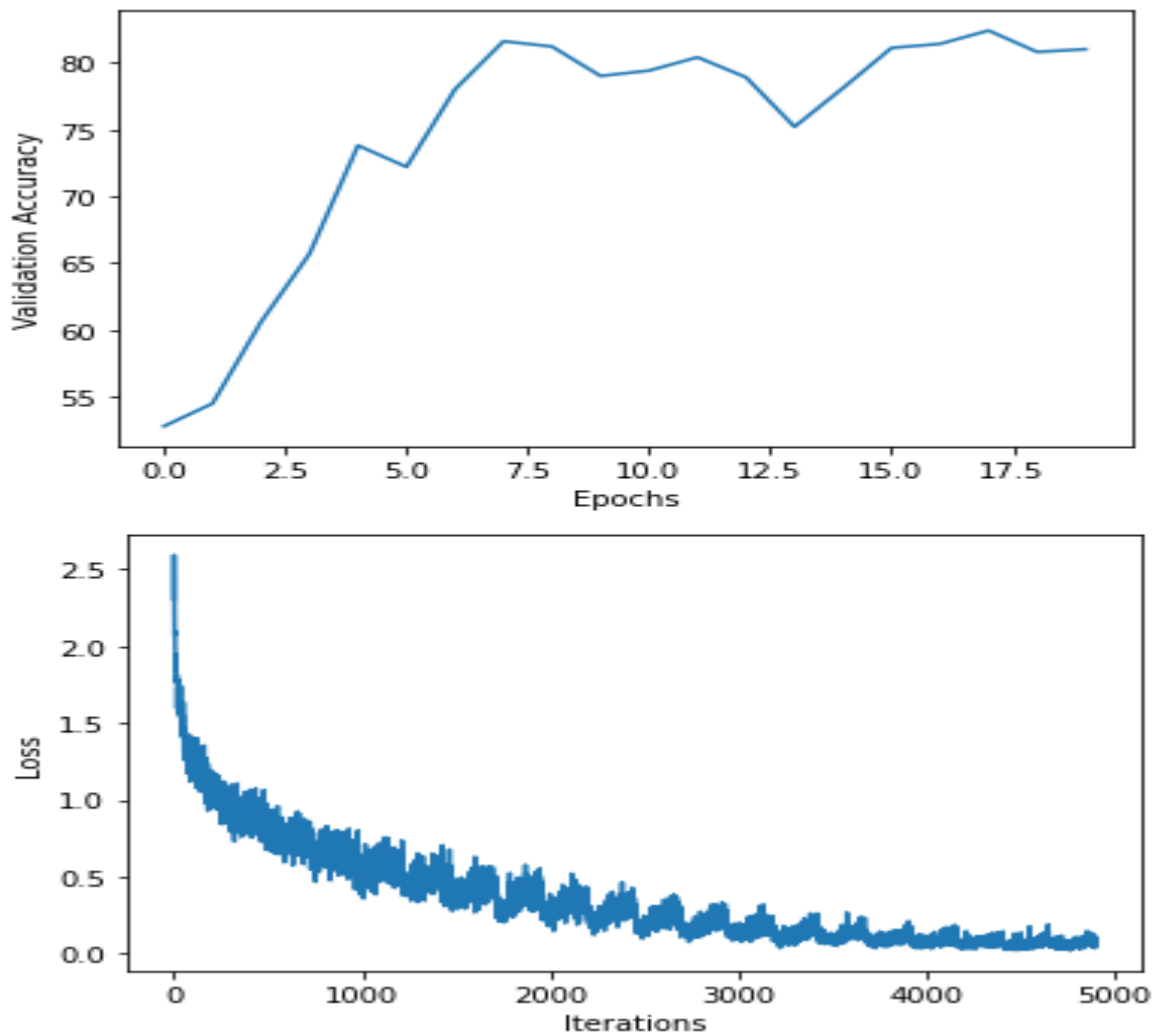
1. Report the training and validation accuracies

We have used **Kaiming Weight Initialization** method considering the ReLu activation used in the network. Because of this we have achieved slightly better results compared to Random Weight Init.

Validation Accuracy: **81.0 %**

Accuracy of the network on the 1000 test images: **81.69999999999999 %**

Plots to show the Accuracies & Loss of this model:

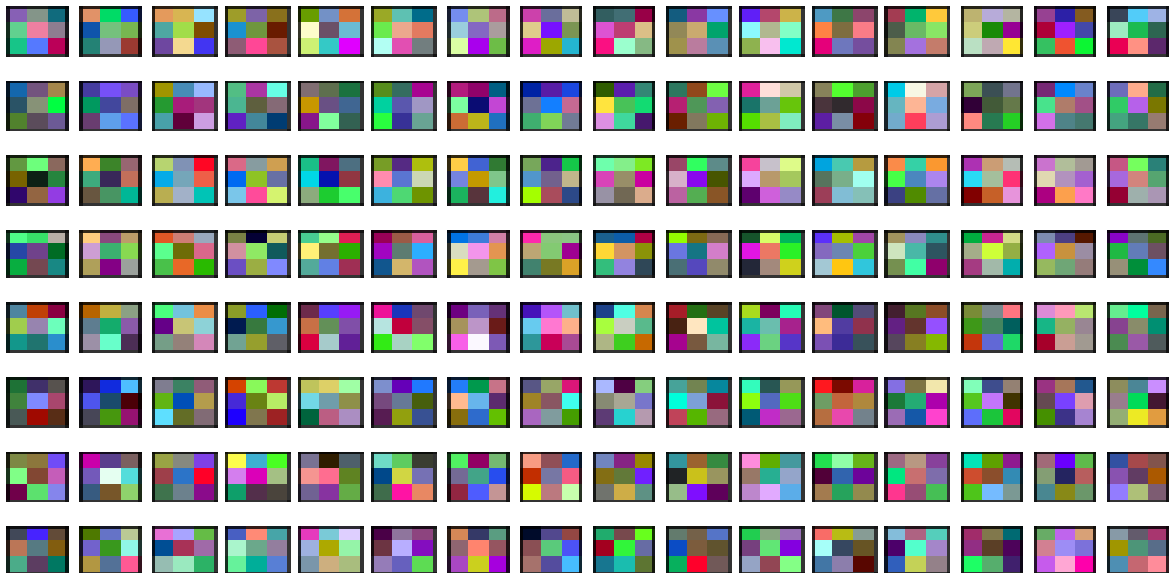


2. Report the number of trainable parameters of the model

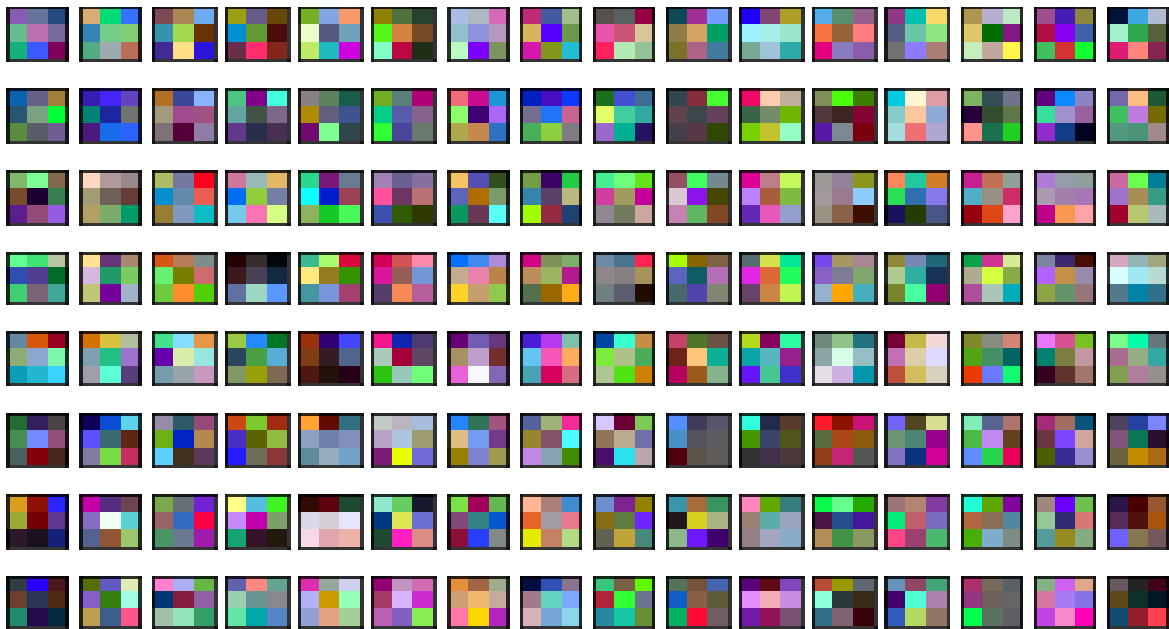
Number of trainable parameters = 7682826

3. Visualize & compare the filters before and after training

Before:



After:



It can be noticed that the filters after the network training process have slightly changed based on certain conditions/regions of the image on which our kernel/filter passes through.

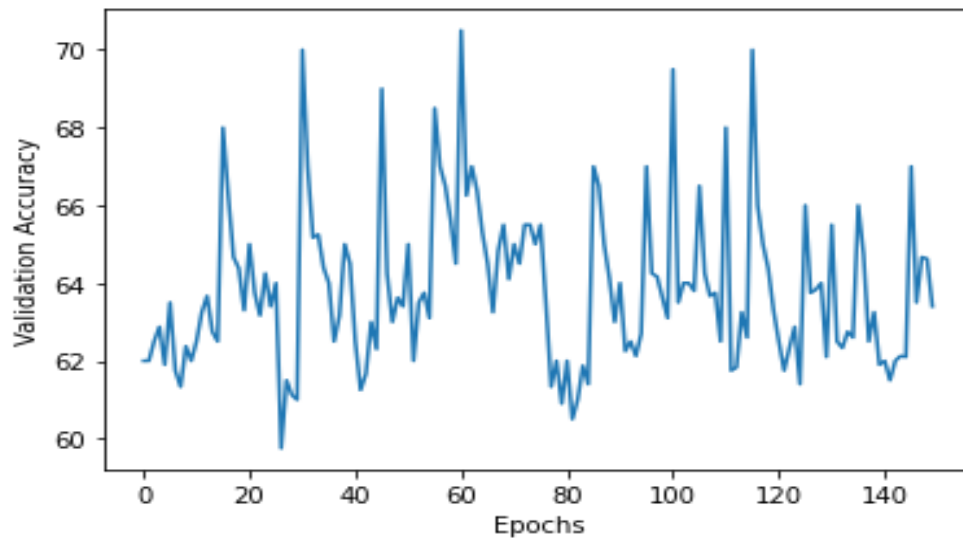
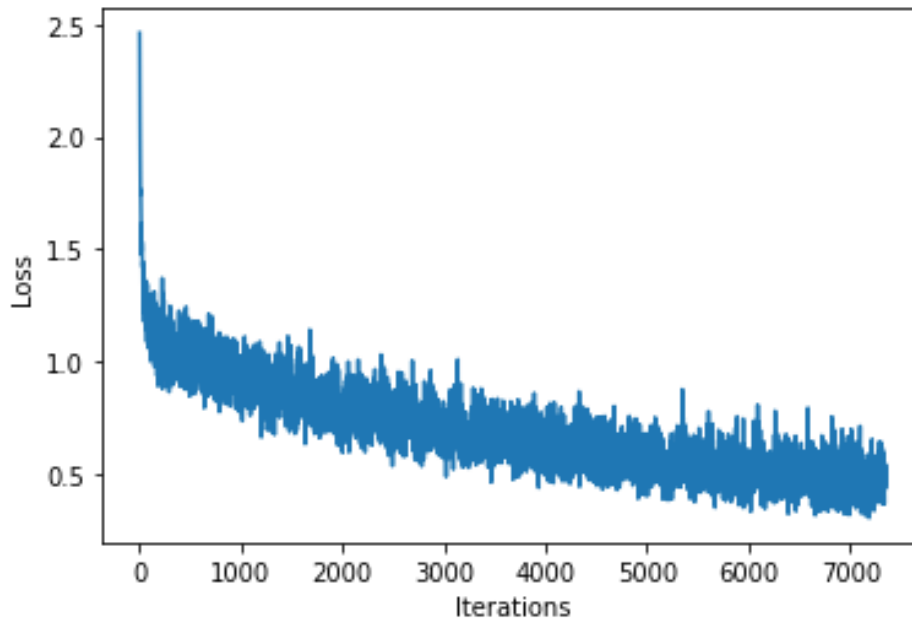
Some colors from the input are more pronounced and picked over other (Light intensity -> Dark in the output) and vice versa. Though it is sad that we can't

understand/see much about the edge detection or fine features detected by the network.

Question 4: Use pre-trained networks

1. Validation and testing accuracies of Pre-trained Model:

Validation Accuracy: **64.8%**



2. Compare the two models training curves, validation and testing performance

Please find the accuracy metrics for Model 2 (Fine-tuned) and Model 3 (Trained from scratch)

Test:

Accuracy of the best pre-trained network (using Early stopping) on the 1000 test images: **89.5 %**

Accuracy of the network with weights from random init on the 1000 test images: **88.9 %**

Validation:

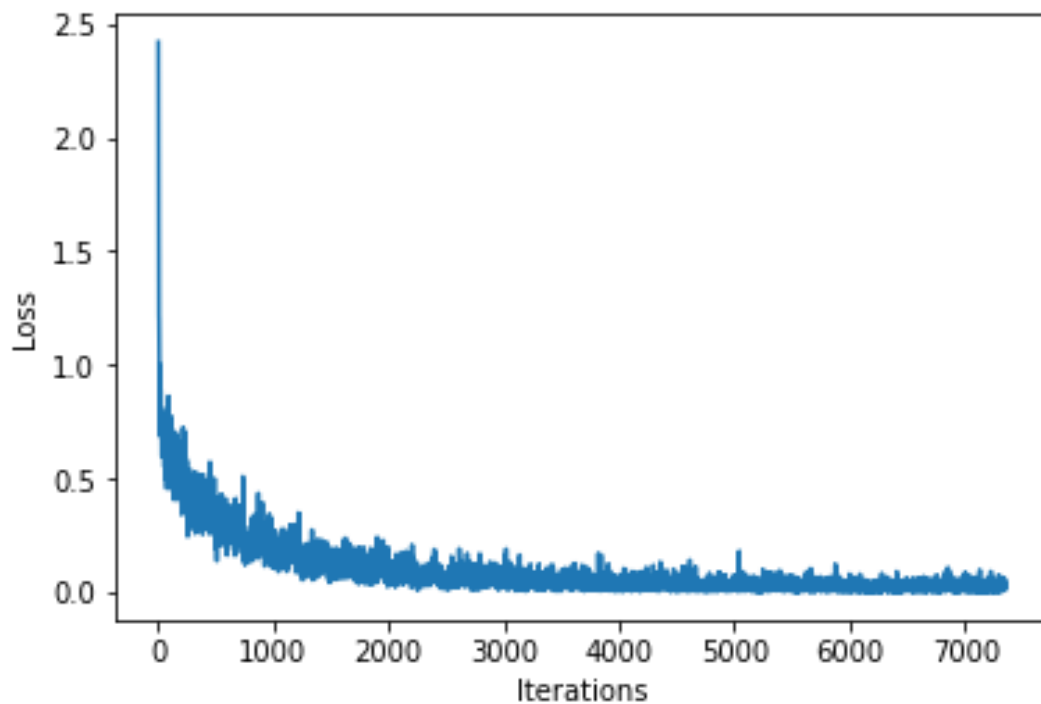
Accuracy of the best pre-trained network (using Early stopping) on the 1000 test images: **89.9 %**

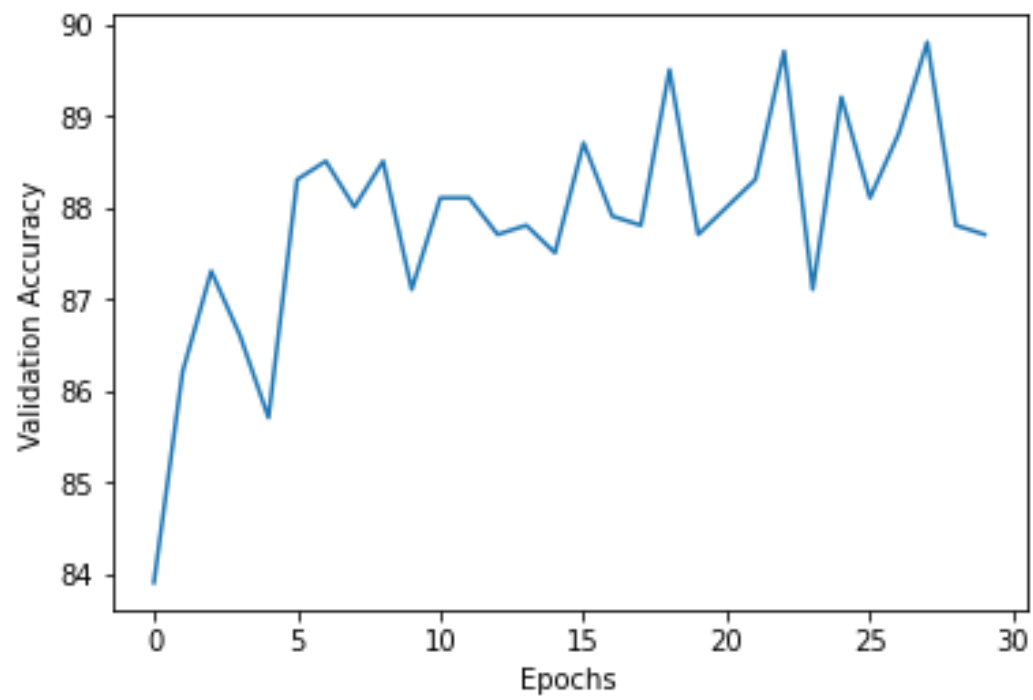
Accuracy of the network with weights from random init on the 1000 test images: **89.1 %**

It appears that the fine-tuned model is slightly out-performing the model which was trained from scratch in both Test and Validation accuracies

Model 2:

Test:





Val:

