Phang Teng Fone 1003296

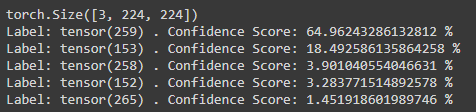
50.038 - Homework 3

**Question 1**

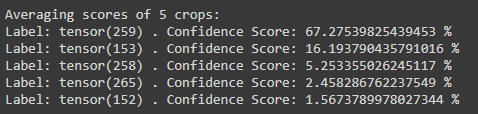
****

Dog.jpg

Out of the box ResNet101:



Data Augmentation with 5 crops and ResNet101:



Data augmentation helps us to have more data on top of the pre existing data available.

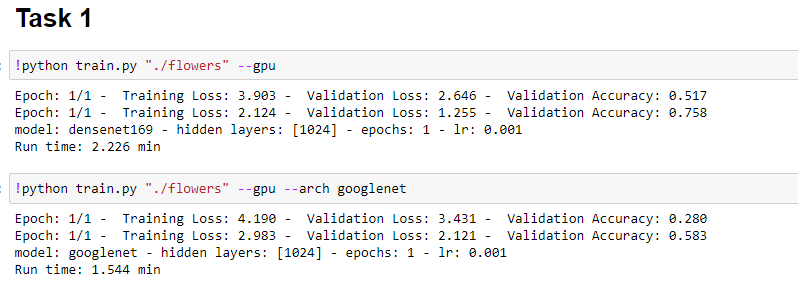
During test time, data augmentation can be used to predict on a n average of multiple views of the same input. It produces more robust results but needs to run the model n number of times which means more computational time.

During training time, data augmentation can be used to extend the training data available by creating instances which can be derived from the training data set.

Data augmentation are also noted to be an estimation of the image and might not reflect the full scale of the image during train time, as such, having an over augmented image on a single file might cause performance decrease instead of benefiting the network.

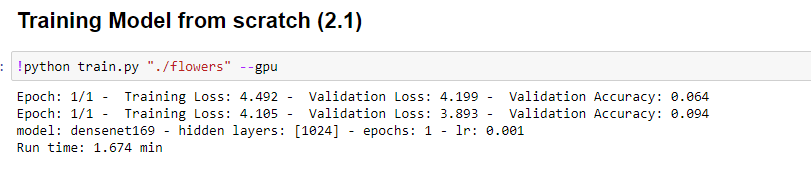
**Question 2**

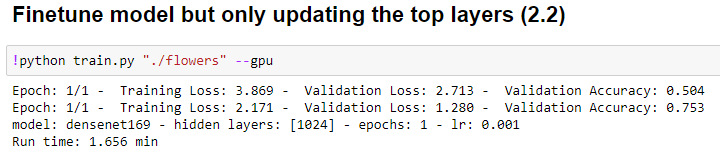
**Task 1**

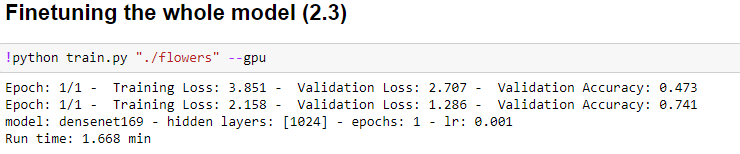


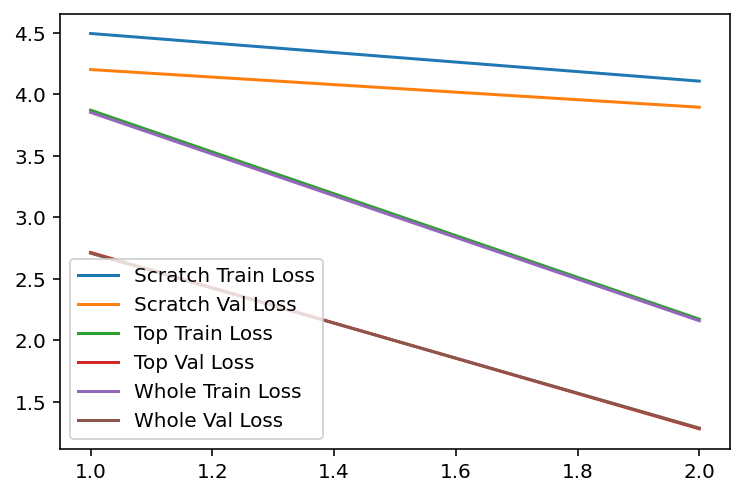
Comparing DenseNet169 to GoogleNet. Densenet 169 has a higher validation accuracy as compared to GoogleNet.

**Task 2**

****

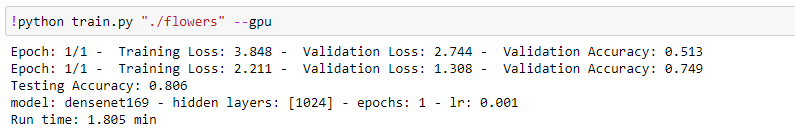
****

****

****

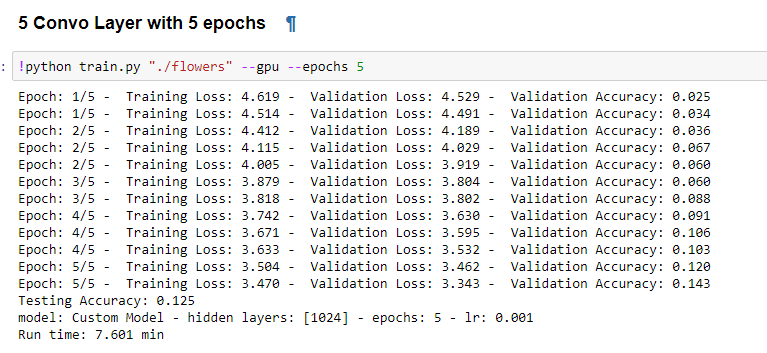
The training from scratch will take a long time to converge as all weights are not loaded preemptively as compared to the rest of the methods. For the whole fine tuning, it performs better than the top layer only fine tuning as theoretically, having to fine tune the entire preloaded weights should give a better performance.

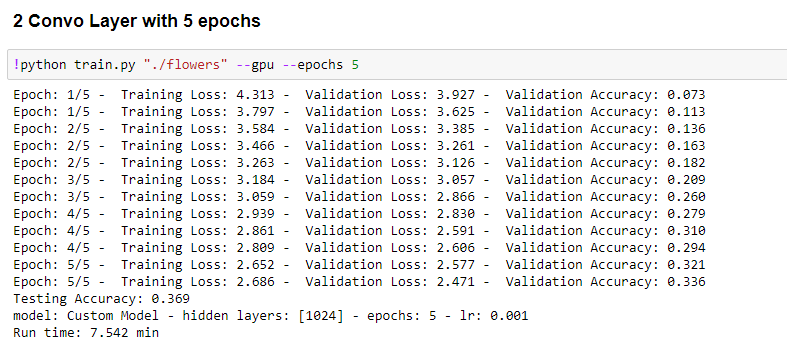
**Task 3**



**Task 4**

For this task, I took reference from Alex Net (<https://github.com/pytorch/vision/blob/master/torchvision/models/alexnet.py>) Sequential layers and atune it to this particular model, including weight initialization to obtain better results.





CNN with lesser convolutional layers performs better as the CNN layers are dependent on the number of data sets available. In our case, since the data set is small, lesser convolutionary lesser will perform better.