Name: Muhammad Aleem Javed
Roll #: 18L-1879

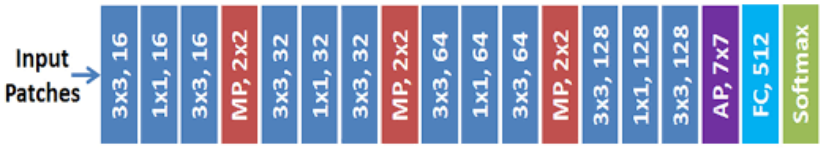
DermNet: Building a deep learning model that classifies skin images with samples of 8 common skin pathologies and carcinoma.

SECTION - I

CLASSIFY THE DISEASES USING BEST DEEP LEARNING ARCHITECTURE FOR CLASSIFICATION.

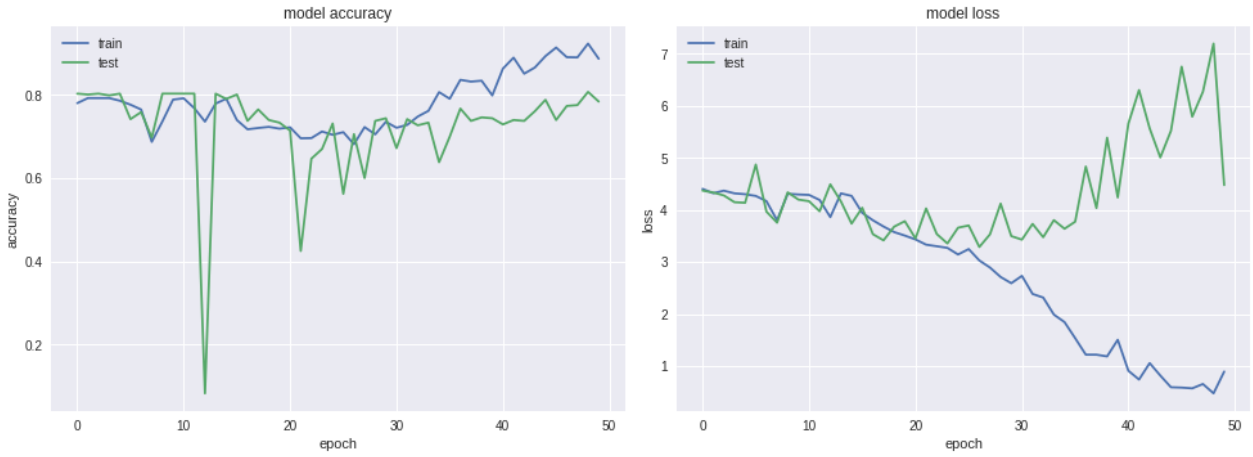
Model and their variations which are used, mentioned below
Class labels are from 0 to 6. {0,1,2,3,4,5,6}, because keras sparse categorial label starts considering class from 0.

Model # 1: Lesion Feature Network (LFN)



Attempt # 1:

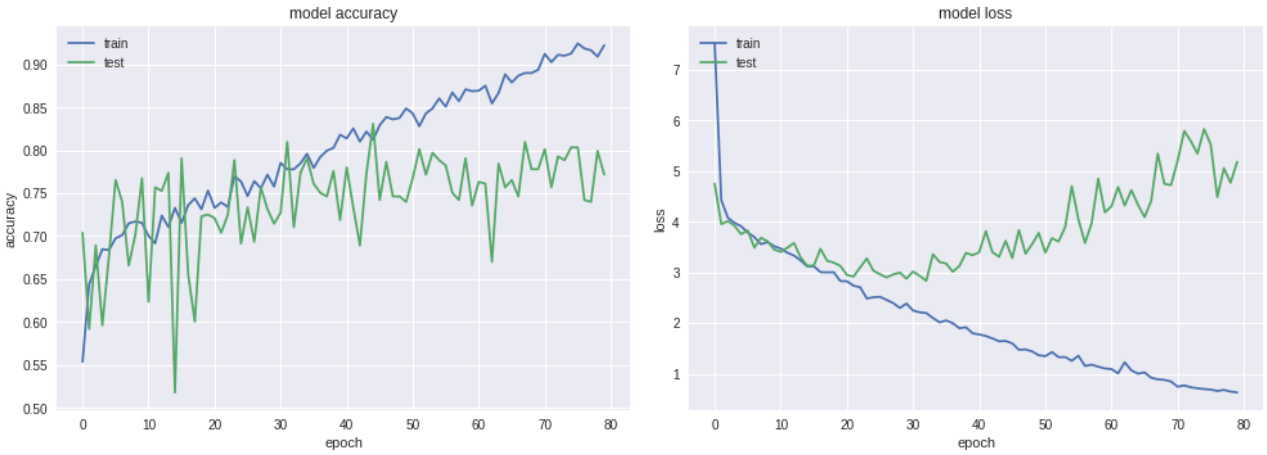
- **Imbalance**: in this attempt, issue of imbalance is solved by assigning particular weights to each class according to the nature of imbalance. Code for this, is given in jupyter notebook attached.
  Balanced Class Weights:
  {**5**: 1.0, **3**: 7.773223016171494, **2**: 7.952674105425165, **0**: 6.801250411319514, **1**: 7.05667463298054, **4**: 8.474374743747438, **6**: 8.405449369662465}
- **Optimizer**: Adam,
- **Input Shape**: (300,300,3).
  Data is resized using opencv function resize, with interpolation: INTER_LANCZOS4
- **Total Parameter**: 1368199
- **Train, Test:** Train on 1891 samples, validate on 473 samples.
  Validation split function of keras is used to extract validation data of 20%.
- **Epochs:** 50
- **Batch Size:** 32



Conclusion: high loss at the end represent the imbalance in test dataset, because we implement the particular weights to each class according to the imbalance, so high loss at the end mean classes with less sample is not being recognized by the cnn.

Attempt # 2:

- **Imbalance**: Same method as Attempt#1 (Model-1) is used to solve imbalance.
- **Optimizer**: Adam, (Learning Rate: 0.00001)
- **Input Shape**: (300,300,3).
  Data is resized using opencv function resize, with interpolation: INTER_LANCZOS4
- **Total Parameter**: 1368199
- **Train, Test:** Train on 1891 samples, validate on 473 samples.
  Validation split function of keras is used to extract validation data of 20%.
- **Epochs:** 50
- **Batch Size:** 16 (Smaller)



Conclusion: smaller batch with lower learning rate has the same large validation loss at the end due to weights assigned to loss function according to class imbalance, the oscillation is due to noise in the data

Attempt # 3:

- **Imbalance**: Same method as Attempt#1 (Model-1) is used to solve imbalance. class weighting differently
- **Optimizer**: Adamax
- **Input Shape**: (300,300,3).
  Data is resized using opencv function resize, with interpolation: INTER_LANCZOS4
- **Total Parameter**: 1368199
- **Train, Test:** Train on 1891 samples, validate on 473 samples.
  Validation split function of keras is used to extract validation data of 20%.
- **Epochs:** 25
- **Batch Size:** 16 (Larger)
- **Model Alteration:** One dropout layer is added.

Conclusion: Adamax is better than Adam, one more drop out and less oscillation, and less over fit. Loss remain the same at the end due to imbalance in test data set and large loss return for smaller dataset classes

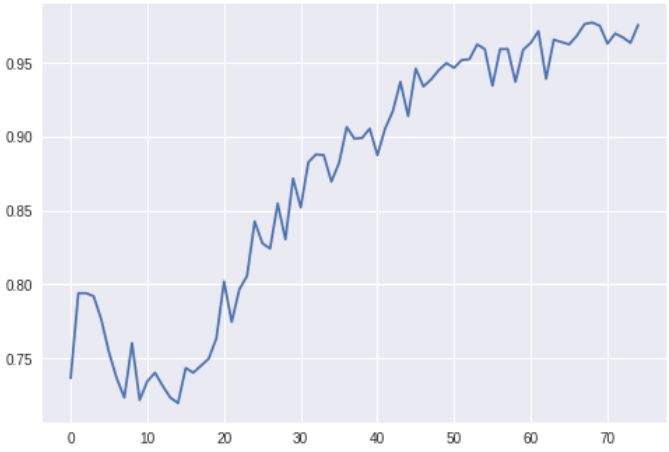
Attempt # 4:

- **Imbalance**: Class weights used in previous attempts used to balance training samples. ***Note,** this time to remove imbalance from test samples we used random sampling of equal size from all classes. Code given in the jupyter notebook attach

- **Optimizer**: Adam
- **Input Shape**: (300,300,3).
  Data is resized using opencv function resize, with interpolation: INTER_LANCZOS4
- **Total Parameter**: 1,630,855
- **Train, Test:** Train on 1891 samples, validate on 473 samples.
  ***Note:** Test and train are separated by code into different folders. Model is train separately only on train set and then tested using test_on_batch function available in keras.
- **Epochs:** 75
- **Batch Size:** 32
- **Model Alteration:** Two dense layers followed by 0.5 dropout.

flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 512)	1049088
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 512)	262656
dropout_2 (Dropout)	(None, 512)	0

- 1st When trained separately on only training data extracted separately (training accuracy)
- 2nd When trained on randomly selected equal samples from all classes. (test accuracy)



Conclusion: this experiment is done , while controlling imbalance in test dataset which was causing large loss at the end, randomly equally choosing samples from all classes causing oscillation and noise also playing its role.

SECTION - II

USE TRANSFER LEARNING ON RESNET, GOOGLENET TO RETRAIN SOME PART OF THE NETWORK

Model # 1: ResNet 50

Attempt # 1:

- **imbalance**: imbalance is removed by writing a python data generator for both train and test for picking equal and random data batch. after getting already trained model on imagenet dataset from keras.application. We freeze its layers except last 4 layers, added one dense of 256 and dropout of 0.40, parameter after,

```
Total params: 49,279,879
Trainable params: 26,746,887
Non-trainable params: 22,532,992
```

- **Image size**: 224x224x3 & 32 batch size
- **Train & Test**: we trained it separately and then test it using test_on_betch, to test it on balanced smaples generated from test data.
- **Epochs & batch**: 200 epochs , 32 was batch size

Conclusion: Our dataset was smaller and this model has very large parameters and start memorizing training dataset if you play with data un-augmented and randomly

Attempt # 2:

- **imbalance**: left last 10 layers for training, try to train on data by equaling selecting batch of 70 from test and train data sets dropout, dense layer of 128 is added at the end of 0.40, parameter after,

  train and test folder extracted again to remove biasness if any,

  ```
  Total params: 36,433,799
  Trainable params: 17,311,751
  Non-trainable params: 19,122,048
  ```

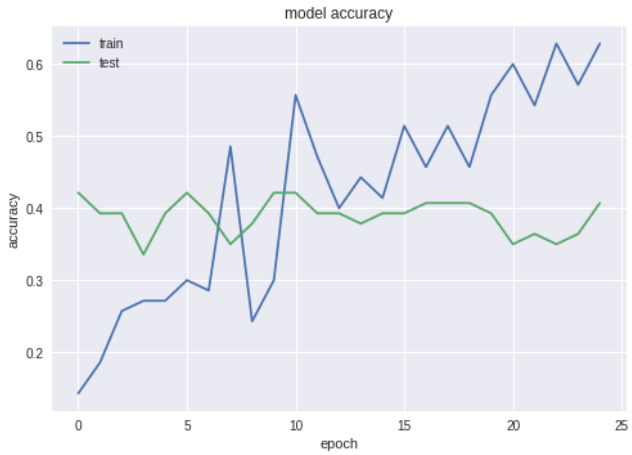
- **Image size**: 224x224x3 & 32 batch size
- **Train & Test**: we trained it separately and then test it using test_on_betch, to test it on balanced smaples generated from test data.
- **Epochs & batch**: 75 epochs , 21 was batch size

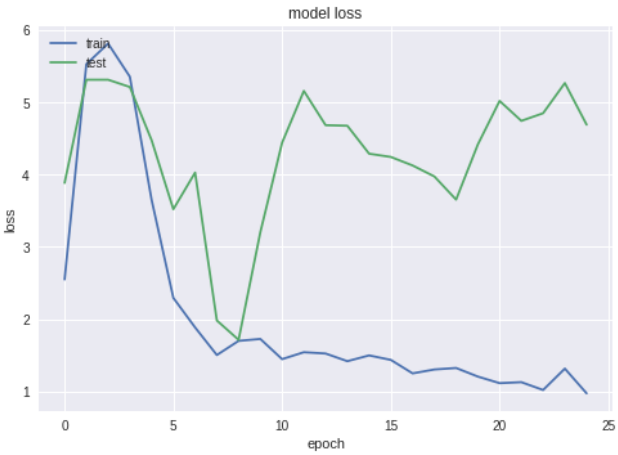
Conclusion: Test and train datasets extracted again from the original data set for random selection of samples.

Model # 2: GoogleNet Inception V3

Attempt # 1:

- **imbalance**: using data without augmentation, using equal and random selection of samples to test and train on, test and train re of equal size.
- **Image size**: 229,229,3 & 10 batch size & 15 epochs
- : drop 0.35 , dense 512, layer all off
- **Train & Test**: we trained it separately and then test it using test_on_betch, to test it on balanced smaples.





SECTION - III

USING GAN TO PRODUCING NEW DISEASE IMAGES

Models :

Searched multiple but unable to make them work,