**Report of Project 3 “DermNet” Deep Learning**

First Part

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

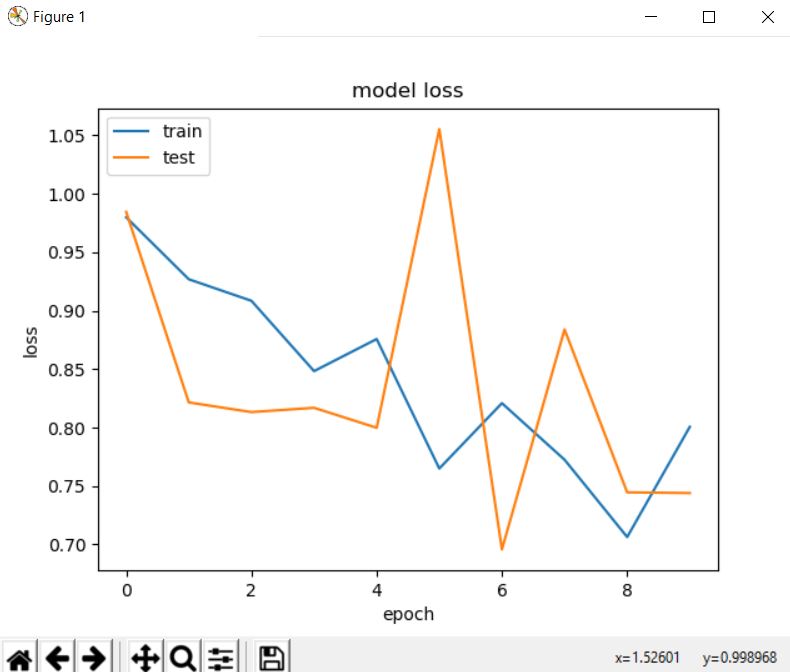
The Skin3000 data set containing dermatology images containing disease and normal skin data:

**Imbalanced data:** Typically refers to a problem with classification problems where the classes are not represented equally.

Class imbalance remove using Fit generator by making a train generator and a test generator which generate image on real time and provide to model fit generator for predictions.

1st Run

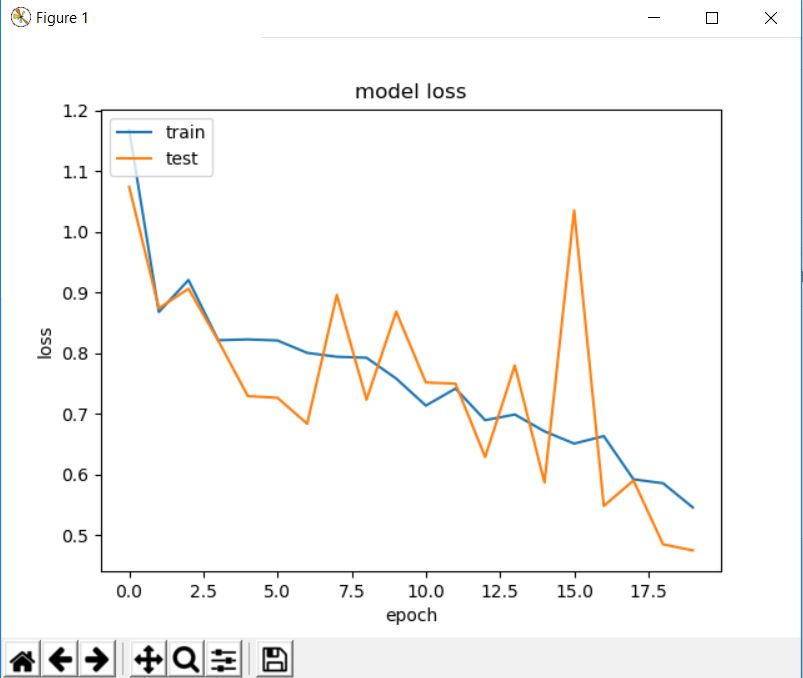
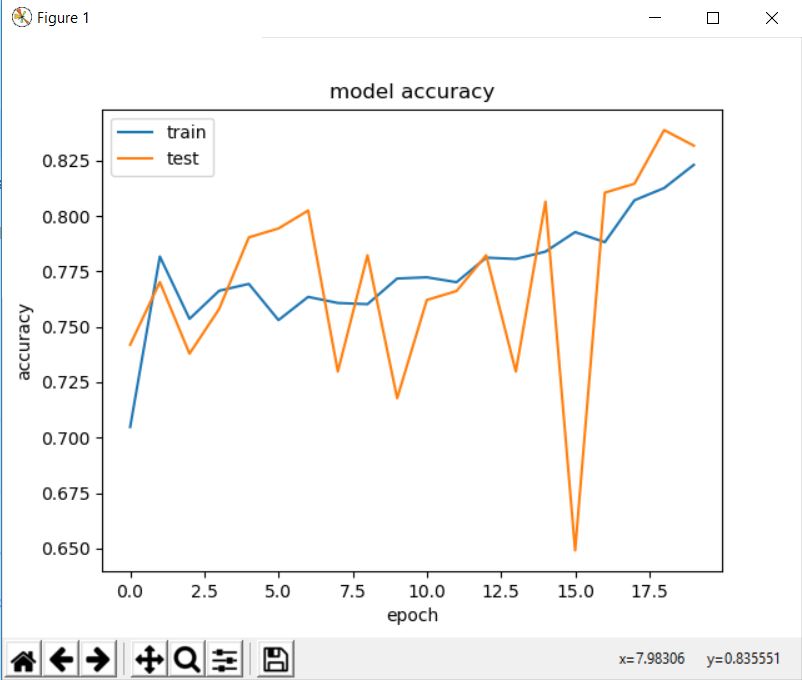
* Batch Size=20
* Number of Epoch= 10
* Total Samples=2423
* Num classes = 7
* Model =AlexNet self train
* Dropout=0.4  
  Dense256,activation="sigmoid"
* Con2d Layer=5
* MaxPooling=2



Conclusion: As the epoch is working the model training accuracy was jolting and test loss was not decreasing as we expected. So I decided to change my model with different parameters.

2cd Run

* Batch Size=30
* Number of Epoch= 20
* Total Samples= 2423
* Num classes = 7
* Model= Self trained model Include
* First add a Conv2D of kernel size=(3,3),filters=32,input\_shape=(128, 128, 3)
* Then add another Conv2D of kernel size=(3,3),filters=32,activation=relu with padding seme  
  here add a MaxPooling2D of pool size=(2,2),strides=(2,2)  
  Conv2D(kernel\_size=(3,3),filters=32,activation="relu",padding="same"))  
  Conv2D(kernel\_size=(5,5),filters=32,activation="relu",padding="same"))  
    
  Dropout=0.5  
  Dense256,activation="sigmoid"



Conclusion: Training accuracy is getting better as of the epochs is working and rest accuracy is also working fine. if we check the loss ,training loss the reducing as we expected and test loss is also reducing with sometime more variations.

**Second Part**

Use transfer learning on ResNet to retrain some part of the network

**Transfer Learning**

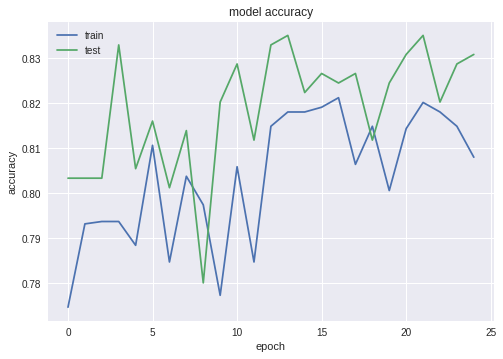
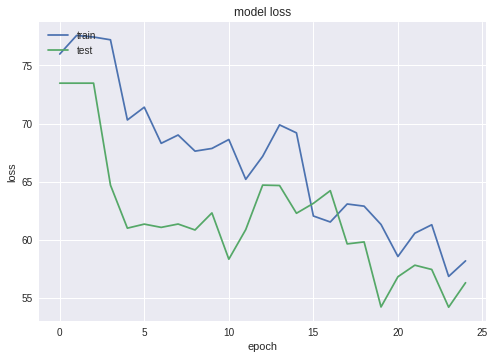
Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has al- ready been learned. ... The survey covers transfer in both inductive learning and reinforcement learning, and discusses the issues of negative transfer and task mapping in depth.

Transfer learning using ResNet50

* Batch Size=64
* Number of Epoch= 25
* Total Samples=2423
* Num classes = 7
* Model =ResNet50
* Input shape=224, 224, 3
* Freeze layer= all

Then add few own layers

* Dense=256, activation=**"relu"**
* Dropout=0.4

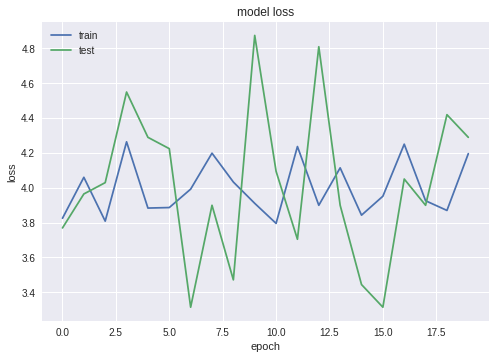


**Transfer learning using InceptionV3**

* Batch Size=20
* Number of Epoch= 20
* Total Samples=2423
* Num classes = 7
* Model =In
* Input shape=224, 224, 3
* Freeze layer= all

Then add few own layers

* Dense=512, activation=**"relu"**
* Dropout=0.5



Third Part

Use a generate adversarial network(GAN) such as [1] to generate new images for these diseases

**The Generator**

The input to the generator is a series of randomly generated numbers called **latent sample**. Once trained, the generator can produce digit images from latent samples.

generator = Sequential([  
 Dense(128, input\_shape=(224,)),  
 LeakyReLU(alpha=0.01),  
 Dense(784),  
 Activation('tanh')  
], name='generator')

### The Discriminator

The discriminator is a classifier trained using the supervised learning. It classifies whether an image is real (1) or not (0).

discriminator = Sequential([  
 Dense(128, input\_shape=(224,)),  
 LeakyReLU(alpha=0.01),  
 Dense(1),  
 Activation('sigmoid')  
], name='discriminator')

gan = Sequential([  
 generator,  
 discriminator  
])