Data Pre-Processing

In this thesis project we have decided to utilize Kaggle Eyepacs dataset which contains 35,000 plus images for diabetic retinopathy classification. In that dataset images of different dimensions, contrast, brightness, structures are present. So it becomes necessary to use various images processing techniques so that our data can become consistent. For making Kaggle Eyepacs dataset. Data pf eye as an images was collected from various source using different fundus cameras under various conditions for different patients. Resultantly the data becomes noisy and very different dimension in terms of image understanding. For feeding a image to a deep learning model they have to be properly cleaned and standardized Preprocessing is a step which is responsible for reducing complexity of training the model which results in increasing the overall models performance. Writing unique algorithm for individual data source is very difficult so after we collect an image we try to standardize it following a particular format or similar to the images collected from rest l sources so that the data becomes consistent. The following paragraphs describes briefs of various image processing techniques that we have followed in this thesis for training our model properly.

Data Imbalance

Class imbalance is a severe problem in many image processing classification tasks that is certain type of abnormality in human body appears in very limited set of images as compared to entire dataset. Usually two problems can be categorized out medical image classification problems, first is rare diseases appears in very less numbers of images which becomes very costly to annotate as well because of rarely available very good skilled medical experts. Secondly there is a big difference in the numbers of normal and diseased images. The normal or common class occupies very high ratio of total dataset. Sometimes the normal classes can also have large number of varieties thus large number of samples which makes it really difficult and exhausting to collect good variants in normal images. Usually our CNNs are feed forward based networks which utilizes back propagation algorithm for reducing errors in the model. What happens is if you feed more number of normal images the computer will get biased towards normal images and will get biased towards predicting diseased images as normal images. One of the important methods for reducing data imbalance in image processing is using oversampling on the minority classes. Using this we can create more data using some of the augmentation techniques as well which usually helps data scientists in the augmentation problem. In next section we will talk about data augmentation problems and solutions.

Modern Data Augmentation Techniques

Deep learning models have incredibly progressed while solving the discriminative tasks. This has been mostly possible because of modern deep neural architectures. A lot of modern efficient big data computational algorithms and architectures with powerful compute capabilities like NVidia tesla gpu. Sometimes when model performs well in training dataset but fails to perform similarly in testing dataset then it means model is overfitted. This can be considered as model is having very poor generalizing ability. One way a data scientist can find out your model is overfitting by plotting the training and validation graphs. Validation should be mutually exclusive of training data. While if both the graphs are following the same pattern that means the model is generalizing well. But if it’s the opposite that validation loss is over shooting then some kind of data augmentation is needed to solve the problem. Data augmentation approaches try to solve the problem of overfitting from the root which is modifying the training dataset. A very obvious assumption is made that if more information is conveyed to the training model that it can train more properly and its generalizing capability can be increased. An important point becomes is while doing data augmentation you have to keep in mind its label annotation should not change because that will lead to the generation of new data in a very poor way which might misled our model which in turn will reduce its accuracy. One of the very basic data augmentation technique can be based on basic image manipulation. For example brightness adjustment, contrast adjustment etc. Few of the basic data augmentation techniques are mentioned below.

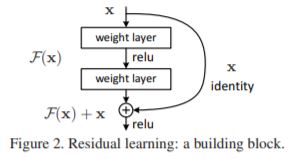
Flipping of the images.   
As we can see that pupil images are circular image are circular and both right and left side images exists, so flipping augmentation horizontally becomes very important in out flipping based augmentation techniques. This is one of the very easiest image augmentation to implement and have been proved very useful in many datasets for example Pascal voc, Cifar 10 and many others. And top of it, this is a label preserving transformation in Kaggle Eyepacs dataset.

Aspect Ration Preserving Restructuring  
Resizing the images is pretty much required while training of any deep learning models because a lot of deep learning models gives optimum results when images of optimum sizes are fed. Cropping is a very practical and easy to understand augmentation method for training the deep learning model. Sometimes if the image size is very big or its having lots of regions, which should not be focused while training a deep learning neural network based model then we should ideally delete those regions if possible using image cropping and if the size of the images is very high as compared to the required to the deep learning model then the image should be resized but aspect ratio of the image should be preserved so that we are accidentally not getting any new label for the same data that we have already labelled once with a different label. Sometimes people also employ zooming and cropping to generate different perspectives from the same images. But here we cannot randomly zoom and crop because data collection method using the fundus camera would be fixed for any given condition. So to remove the unnecessary part of the image which is actually redundant while training we will limit our cropping process to that.

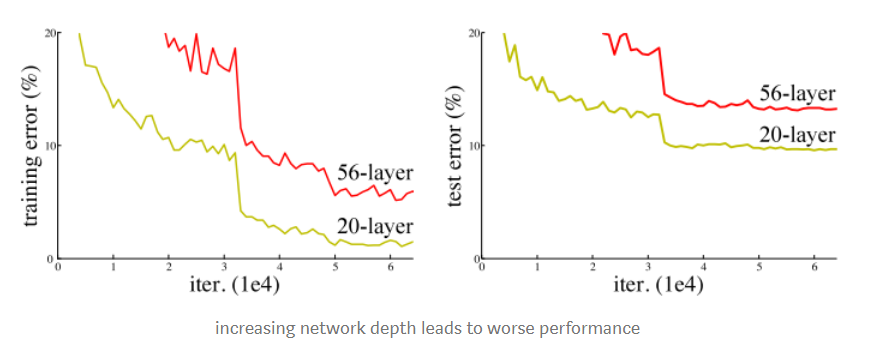
CNN

ResNET

It is sometimes very difficult to train very deep neural networks due to many shortcoming of the architectures before like exploding gradient and vanishing gradients problems. Therefore He Zhang presented a neural network architectures which involved a newly coined term residual learning, which made possible training of substantially deep neural architectures as compared to those which came before. A very important point was proved here which is on increasing depth of the model considerable gain in the accuracy can be achieved. When ResNET’s paper was first presented, authors were able to achieve 28 % more accuracy in COCO dataset as compared to any other model. Building block of residual learning is present in the figure below. As you can see that the activation from previous layer is carried forward and added to the next layers, overfitting and vanishing gradient becomes very very difficult and thus that is how it solves our getting bottleneck problem while training the model.



Universal Approximation theorem states that, if you give enough capacity to a neural network it can approximate any complex and easy mathematical function. Beleiving this statement to be true a lot of authors tried increasing the sizes of layers but they were often trapped under the traps of overfitting and vanishing gradient. After the arrival of alex net a lot of deep neural nets are exploring depth fantasies with vgg net reached 13 and 16 layers while google net ventures as depth as 20 to 22 layers.



Following curve shows that with increasing depths in neural network model overfits and validation error increases. Before ResNET a lot of trial and error methods were tried out to tackle the problem. Some of the notable mentions including adding an extra loss in between to supervise the increasing error rates but none of them were really successful in tackling the problem. The authors strongly believed that just stacking layers should not decrease performance, so they created identity mapping as to transfer activations from previous layers. This clearly means that if the models get deeper their errors would never be higher than the similar fashioned made shallow models.

DenseNET

Some of the recent developments in academia have shown that a deeper network can be valuable sometimes like Resnets. Before inception of Resnets people believed that beyond a particular depth the networks starts performing very badly and many problems like overfitting and vanishing gradients occurs. Densenets proved that close to inputs and close to outputs if we have shorter connections then networks work more efficiently as compared to any other permutation possible. Usually in Deep Neural Networks befor L convolutions used to have L connections , but Gao Liu provided proofs that with L(L+1)/2 connection between subsequent layers a neural network can be very effective a well.

Before the proposal of Densenets a cascade network structure was proposed in neural networks. Their major work focus was on multi layer perceptron where network should have been trained in layer by layer fashion. Although this was quite effective in some empirical tasks and very smaller datasets, it failed badly and needed many improvements if it was to be scaled for major modern datasets like imagenet and coco. For training more than 100 layers, end to end highway networks provided the first breakthrough. Even with 100s of layers inside the highway networks, those networks could be optimized very easily. This point was further verified by authors of ResNETs. Depth of network can be increased in many ways, one out of which is increasing the layer width. If performance in resnets has to be increased number of filters increasing can be an option. One of the wide networks which achieved good results were the fractal nets.

Densenets could have been modelled as a very deep or a very wide architectures but instead of doing so, dense nets authors tried to do feature reuse which resulted in more connected neural networks which are easier to train and deploy. By concatenation of feature maps from previous layers, we increase the amount of information flowing from layer by layer which makes the gradient flow very smooth and useful. As it can be seen that in inception nets as well, convolutional layer are concatenated but the operations are more effective when it comes to densenets.

One of the major reasons that dense convolutional layers which are used in densenets have higher accuracy is every layer receives additional supervision from previous layers using loss functions and shorter connections. Deep Densenets can be called to be called as “Deep Supervision” models. Densenets can be said to perform similar to deep supervision nets. Authors made sure that loss function and optimizer of densenets are less complicated than DSNs and loss functions of all layers are same. 