Predicting if disease is present in Chest X-rays with Convolutional Neural Networks

The ChestXnet Dataset was released in Fall of 2017 and gave the public access to over 100, 000 Chest Xrays image. The full dataset is available on <https://www.kaggle.com/nih-chest-xrays/data>, and the sample dataset of as well as a sample set of over 5000 Xrays is available at <https://www.kaggle.com/nih-chest-xrays/sample/data>. There is also a csv file in both those links that has the findings of the radiology reports. The authors of the dataset state the ‘Findings’ columns of the csv was derived using NLP from the radiology reports. They claim >90% accuracy. However, as this paper will demonstrate and corroborated by a well known blogpost by Dr. Oakden-Raynor <https://lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/> this dataset is not labeled correctly. There are serious issues with the accuracy of the labels. In fact Stanford’s famous paper “CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning”

<https://stanfordmlgroup.github.io/projects/chexnet/> used the same ChestXnet database, but had panel of radiologists read over them and relabel them to ensure accuracy. Despite this shortcoming there is still utility in using the dataset, least of which in demonstrating the importance of good data.

The Chest X-rays images are png files a size of 1024 x 1024 with 3 channels. Due to computation load factors, limitation in physical memory, and the fact that X-ray images are black and white I decided to reduce the size of the images and make them grayscale. I decided to reframe the problem statement as well. I used the sample dataset csv filed, created a dataframe using the python pandas module and parsed through the findings labels and one hot encoded the 15 labels corresponding to pathology. However, considering that many of these labels were incorrect, and that the classes were heavily imbalanced towards “No Findings” aka no abnormalities, I decided to change make this a binary classification problem, detecting if an xray shows an abnormality or not. I grayscaled, resized, and normalized the images to 256 x 256 and to 128 x 128 using opencv and numpy and dividing the value of every pixel by 255. I did not attempt to see measure the difference in aliasing options, and interpolation methods nor did I try and compare with other modules such as Pillow and SkImage. These are some further considerations which would be interesting for any project. I then compared the performance on multiple convoluted neural networks including a simple CNN with 2 convolutional layers each followed by poolings layers, then a third convolutional layer, a flatten layer, a dense layer, a dropout layer, and a final dense layer. I used a filter size of 2x2 with padding set to ‘same’ and max pooling with a kernel of 2x2 and 2 strides. I also used transfer learning and compared models with a base of MobileNet, Inception V3, VGG-16. The topology of these models are included below. I used them as a base and added a Gap2D layer, dropout layer, dense layer, dropout layer, and final dense layer. The activation function chosen was the sigmoid function, the loss function was binary cross entropy, and the optimizer was Adam optimizer.

In the future I’d preprocess the images by adjusting the contrast and slight rotating the xray images up to 25 degrees in either direction. Additionally I’d like to predict individual pathologies found in the xrays by using multiclass multi-output approach since many xrays can show multiple pathologies.