Capstone Project 2 Milestone Report
Detecting Pneumonia with Chest X-Rays

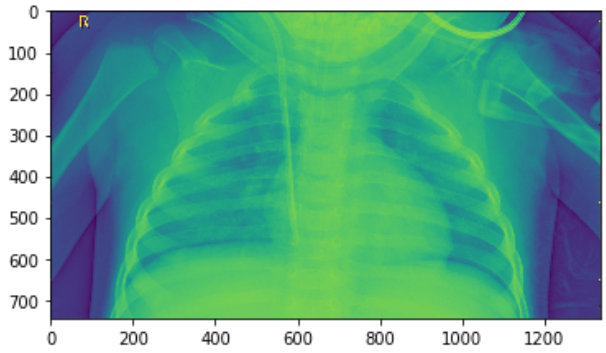
Pneumonia is an infection of the lungs that affects millions of people a year as one of the most common causes for admission to a hospital. To diagnose a patient with pneumonia, chest x-rays are typically taken to a highly trained specialist in order to diagnose the extent and location of the infection. If the process for such a common infection can be streamlined and made more efficient through automation and machine learning, patients would be able to receive treatment faster and doctors would be able to diagnose more patients.

We will be using data sourced from:
https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

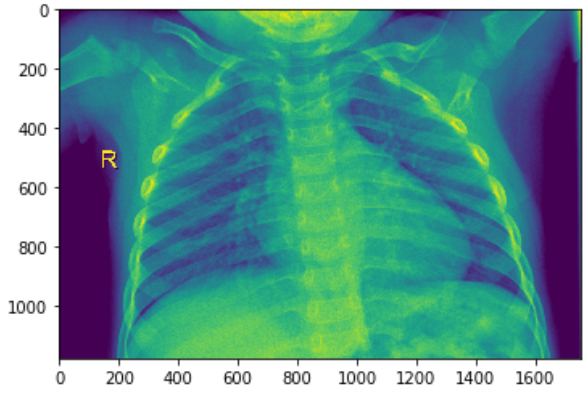
The images are grouped by doctors as being normal, or with pneumonia. The photos labeled as pneumonia are also labeled as bacterial or viral. Low quality or unreadable scans were also removed from the data set.

Let's first see some examples of our images labelled pneumonia and normal.

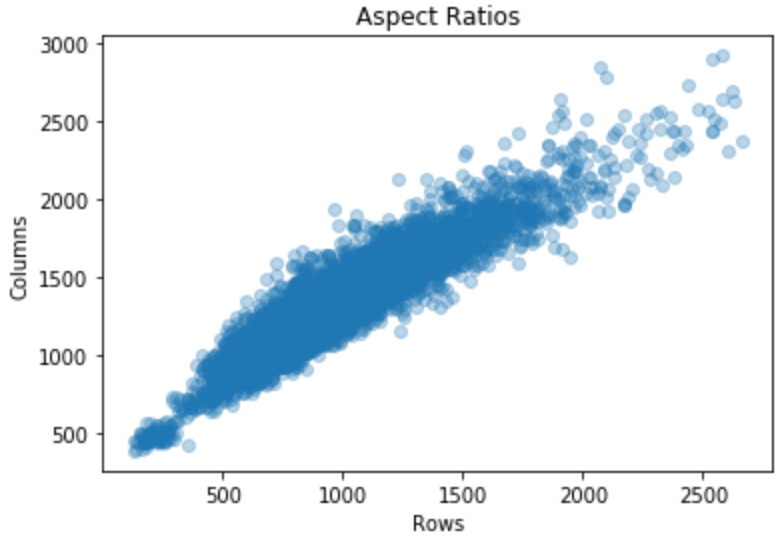
Pneumonia



Normal



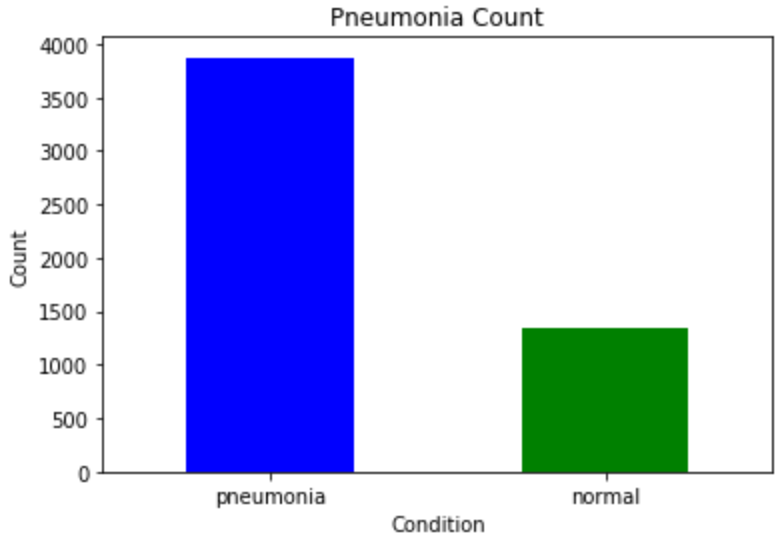
The first thing we note is that the images in the set all vary in size. Let's see what the sizes generally look like.



Mean Aspect Ratio = 1.44

Even though the image sizes vary, we can see that there is a linear relationship between the rows and the columns of an image. Will need to resize the images for our deep learning network, so it may be useful to maintain a similar aspect ratio so that we do not alter the images too much.

Next we should see how many instances of each category (pneumonia and normal) exist in our data set.



We can see that our data set is slightly imbalanced, having more pneumonia images than normal images. This suggests that we may need to be careful when selecting our evaluation metrics, and that simply using accuracy may not be sufficient. We will need to explore the use of recall, precision and F1 scores.

The data set is first shuffled and then loaded using DataImageGenerator with a validation split of 0.3. The images are then processed using the following deep learning model architecture.

```
model = Sequential()
model.add(Conv2D(32, (3,3), strides = (1,1), activation = 'relu',
input_shape = (image_size[0], image_size[1],3)))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(32, (3,3), strides = (1,1), activation = 'relu',
input_shape = (image_size[0], image_size[1],3)))
model.add(MaxPooling2D((2,2)))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(128, activation = 'relu'))
model.add(Dense(1, activation = 'sigmoid'))
```

Model was compiled using the ‘adam’ optimizer, with accuracy, f1 score, precision, and recall as our desired metrics.

After 10 epochs of training,

Acc = 0.9992
F1 = 0.9994

Evaluating our model on the test data,
Acc = 0.7756
F1 = 0.8434

This clearly indicates an overfit issue. The next steps to improve the model would be to experiment with adding weight regularization or some dropout layers. We will also try to improve the model by increasing its complexity by adding more layers and adjusting the parameters.

Another possible way to improve the model is to perform some data augmentation on the images using OpenCV. Some examples of this would be edge detection, erosion, or dilation.