Data Science Career Track

Capstone 2 -

Final Report

by Edward Franke

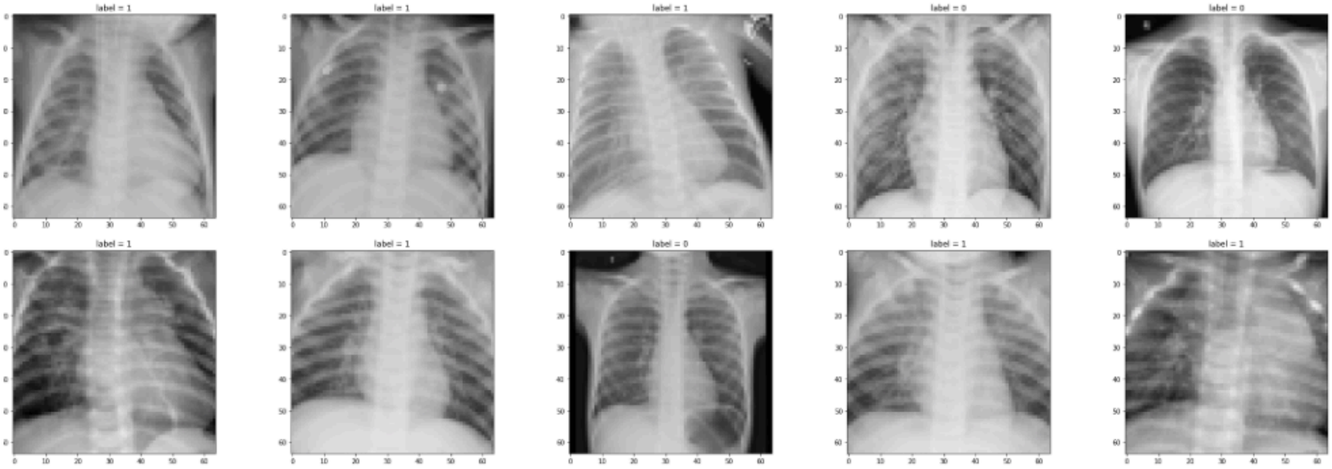
06/11/2019

Detect Pneumonia is chest x-rays Project – Final Report

EXECUTIVE SUMMARY:

The purpose for this project is to find a correlation connected to chest x-rays containing pneumonia that can separate them in real time compared to normal healthy chest x-rays. The dataset is a cleaned dataset from Kaggle.com. Keras (with Tensorflow as the backend) has discovered connections and methods to determine which x-rays have pneumonia with varying success.

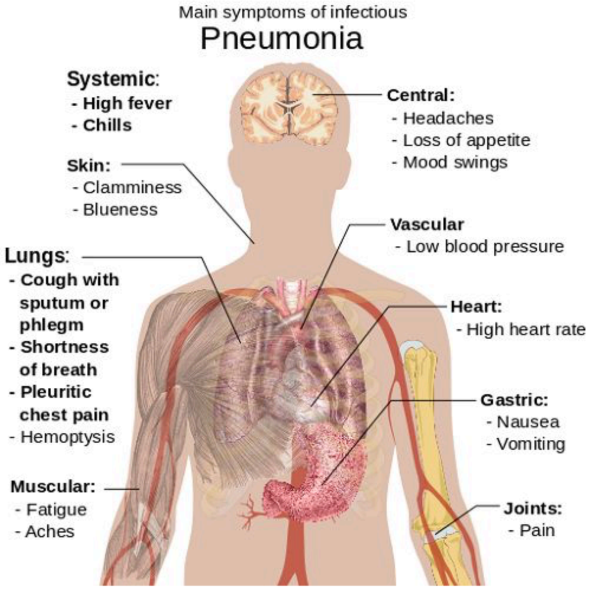
IDEA: A model to detect pneumonia is chest x-rays. (problem to solve)
CLIENT: Medical Professionals
REASON: Pneumonia is a very serious condition that has the potential for death. I have personal knowledge of how serious it can be. The sooner it can be detected, the better the chance for survival and less damage to the lungs. Machine Learning is now capable of detecting unhealthy faster and more accurate than trained medical professionals.
DATA: From Kaggle, 2 cleaned datasets with 5863 images and over 112,000 images.
SOLUTION: Create a model or analysis to discover what makes an x-ray image of the chest to have pneumonia and to capture this automatically.
DETAILS: See below.
DELIVERABLES: Code and a presentation outlining the discoveries.



BACKGROUND:

Pneumonia: details

Pneumonia is a lung inflammation caused by bacterial or viral infection, in which the air sacs fill with pus and may become solid. Inflammation may affect both lungs (double pneumonia), one lung (single pneumonia), or only certain lobes (lobar pneumonia).



Bacterial pneumonia. This type is caused by various bacteria. The most common is Streptococcus pneumoniae. It usually occurs when the body is weakened in some way, such as by illness, poor nutrition, old age, or impaired immunity, and the bacteria are able to work their way into the lungs. Bacterial pneumonia can affect all ages, but you are at greater risk if you abuse alcohol, smoke cigarettes, are debilitated, have recently had surgery, have a respiratory disease or viral infection, or have a weakened immune system.

Viral pneumonia. This type is caused by various viruses, including the flu (influenza), and is responsible for about one-third of all pneumonia cases. You may be more likely to get bacterial pneumonia if you have viral pneumonia.

Mycoplasma pneumonia. This type has somewhat different symptoms and physical signs and is referred to as atypical pneumonia. It is caused by the bacterium Mycoplasma pneumoniae. It generally causes a mild, widespread pneumonia that affects all age groups.

Other pneumonias. There are other less common pneumonias that may be caused by other infections including fungi.

Pneumonia: statistics

It is the leading cause of death due to infection in children younger than 5 years of age worldwide. Pneumonia and influenza together are ranked as the eighth leading cause of death in the U.S. Those at high risk for pneumonia include older adults, the very young, and people with underlying health problems. 2,400 children a day in 2015.There are 120 million episodes of pneumonia per year in children under 5, over 10% of which (14 million) progress to severe episodes. There were an estimated 880,000 deaths from pneumonia in children under the age of five in 2016. Most were less than 2 years of age.

Pneumonia: grading

The CURB-65 scores range from 0 to 5. Assign points as in the table based on confusion status, urea level, respiratory rate, blood pressure, and age. Clinical management decisions can be made based on the score, as described in the validation study below:

Score	Risk of Death	Disposition
0 or 1	1.5% mortality	Outpatient care
2	9.2% mortality	Inpatient vs. observation admission
≥ 3	22% mortality	Inpatient admission with consideration for ICU admission with score of 4 or 5

Pneumonia: Machine Learning

A new artificial intelligence algorithm can reliably screen chest X-rays for more than a dozen types of disease, and it does so in less time than it takes to read this sentence, according to a new study led by Stanford University researchers.

The algorithm, dubbed CheXNeXt, is the first to simultaneously evaluate X-rays for a multitude of possible maladies and return results that are consistent with the readings of radiologists, the study says.

Scientists trained the algorithm to detect 14 different pathologies: For 10 diseases, the algorithm performed just as well as radiologists; for three, it underperformed compared with radiologists; and for one, the algorithm outdid the experts.

(Source https://med.stanford.edu/news/all-news/2018/11/ai-outperformed-radiologists-in-screening-x-rays-for-certain-diseases.html)

Pneumonia: Personal experience

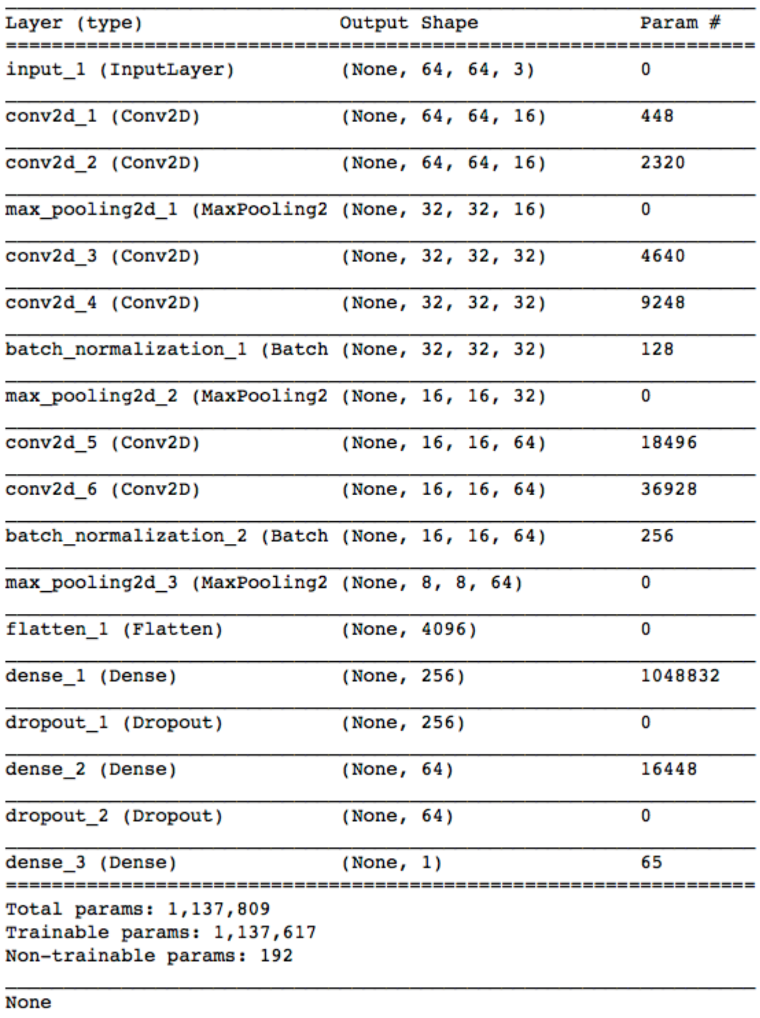
The author of this project had a personal experience with bilateral (both lungs) pneumonia combined with ARDS (acute respiratory distress syndrome) in early 2011. This experience combined with the interest of imaging were the reasons for choosing these datasets to use imaging machine learning.

DATASETS and PROCESSING

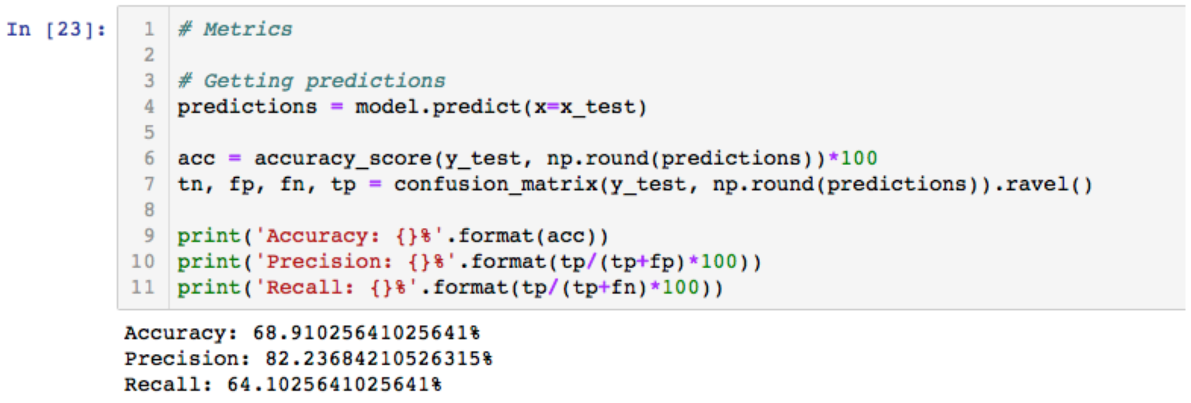
Pneumonia Dataset

This dataset has 5866 images in jpg format separated into separate “healthy” and “pneumonia” folders with applicable image files. Including 3,883 characterized as depicting pneumonia (2,538 bacterial and 1,345 viral) and 1,349 normal, from a total of 5,856 patients to train the AI system. The model was then tested with 234 normal images and 390 pneumonia images (242 bacterial and 148 viral) from 624 patients.

The images were loaded into Python and categorized into train, test, validation sets under “normal” or “pneumonia”. Because the dataset was cleaned prior, no cleaning was required. Also, no outliers were discovered. A model was created with Kera’s functional API, processed, and hyperparameters for fitting set.

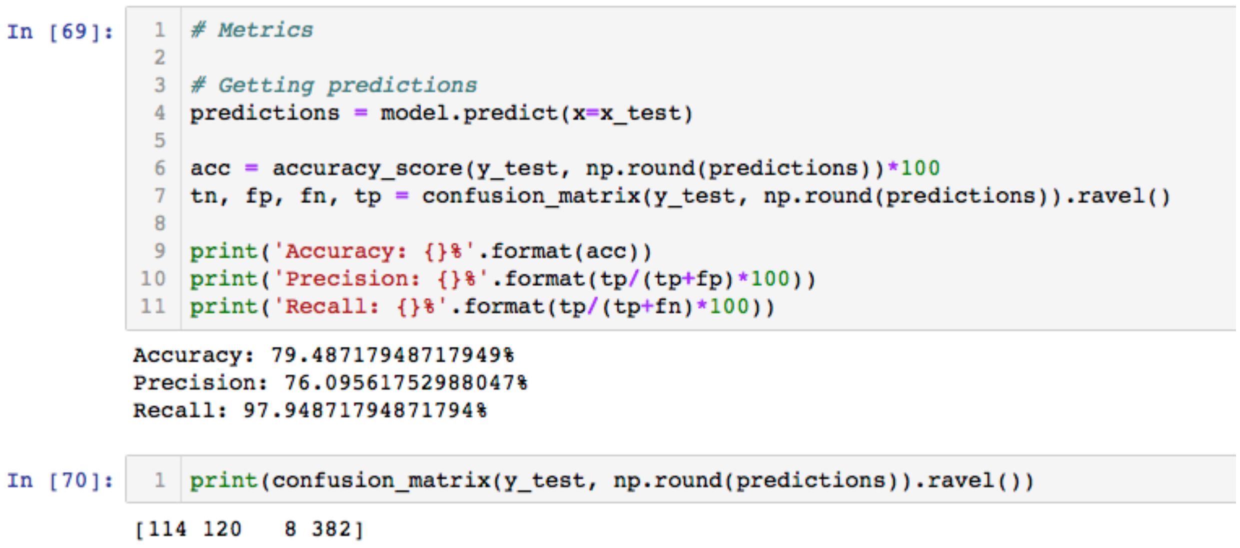


```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 64, 64, 3)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 64, 64, 16)        448
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 64, 64, 16)        2320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 32, 32, 32)        4640
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
batch_normalization_1 (Batch (None, 32, 32, 32)        128
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 16, 16, 64)        36928
_________________________________________________________________
batch_normalization_2 (Batch (None, 16, 16, 64)        256
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 4096)              0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               1048832
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                16448
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 65
=================================================================
Total params: 1,137,809
Trainable params: 1,137,617
Non-trainable params: 192
_________________________________________________________________
None
```



```
In [23]:
# Metrics

# Getting predictions
predictions = model.predict(x=x_test)

acc = accuracy_score(y_test, np.round(predictions))*100
tn, fp, fn, tp = confusion_matrix(y_test, np.round(predictions)).ravel()

print('Accuracy: {}%'.format(acc))
print('Precision: {}%'.format(tp/(tp+fp)*100))
print('Recall: {}%'.format(tp/(tp+fn)*100))
```

```
Accuracy: 68.91025641025641%
Precision: 82.23684210526315%
Recall: 64.1025641025641%
```



```
In [69]:
# Metrics

# Getting predictions
predictions = model.predict(x=x_test)

acc = accuracy_score(y_test, np.round(predictions))*100
tn, fp, fn, tp = confusion_matrix(y_test, np.round(predictions)).ravel()

print('Accuracy: {}%'.format(acc))
print('Precision: {}%'.format(tp/(tp+fp)*100))
print('Recall: {}%'.format(tp/(tp+fn)*100))
```

```
Accuracy: 79.48717948717949%
Precision: 76.09561752988047%
Recall: 97.94871794871794%
```

```
In [70]:
print(confusion_matrix(y_test, np.round(predictions)).ravel())
```

```
[114 120   8 382]
```

Changing the number of epochs greatly increased the accuracy of the results but also requires greater computer processing times. Because this is an unbalanced dataset (one class being dominate), the confusion matrix is the best method to determine the results of the model with the 8 false and 382 true classifications being acceptable.

About the model
Tensorflow (backend used by Keras)
Starting in 2011, Google Brain built DistBelief as a proprietary machine learning system based on deep learning neural networks. Its use grew rapidly across diverse Alphabet companies in both research and commercial applications. Google assigned multiple computer scientists, including Jeff Dean, to simplify and refactor the codebase of DistBelief into a faster, more robust application-grade library, which became TensorFlow. In 2009, the team, led by Geoffrey Hinton, had implemented generalized backpropagation and other improvements which allowed generation of neural networks with substantially higher accuracy, for instance a 25% reduction in errors in speech recognition. TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on

single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors. During the Google I/O Conference in June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned TensorFlow, of which only 5 were from Google.

About Keras

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel. In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling. Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep-learning models on clusters of Graphics Processing Units (GPU) and Tensor processing units (TPU).

Pros and Cons of Keras:

Easier to use and build model(pro)
Can use both Tensorflow and Theano backend(pro)
Less time consuming and reusability is excellent.(pro)
There have been complaints about performance issues with Tensorflow backend(con).

Features

rescale is a value by which we will multiply the data before any other processing. Our original images consist in RGB coefficients in the 0-255, but such values would be too high for our models to process (given a typical learning rate), so we target values between 0 and 1 instead by scaling with a 1/255. factor.

shear_range is for randomly applying shearing transformations

zoom_range is for randomly zooming inside pictures

horizontal_flip is for randomly flipping half of the images horizontally --relevant when there are no assumptions of horizontal asymmetry (e.g. real-world pictures).

fill_mode is the strategy used for filling in newly created pixels, which can appear after a rotation or a width/height shift.

Sample: one element of a dataset.

Example: one image is a sample in a convolutional network

Example: one audio file is a sample for a speech recognition model

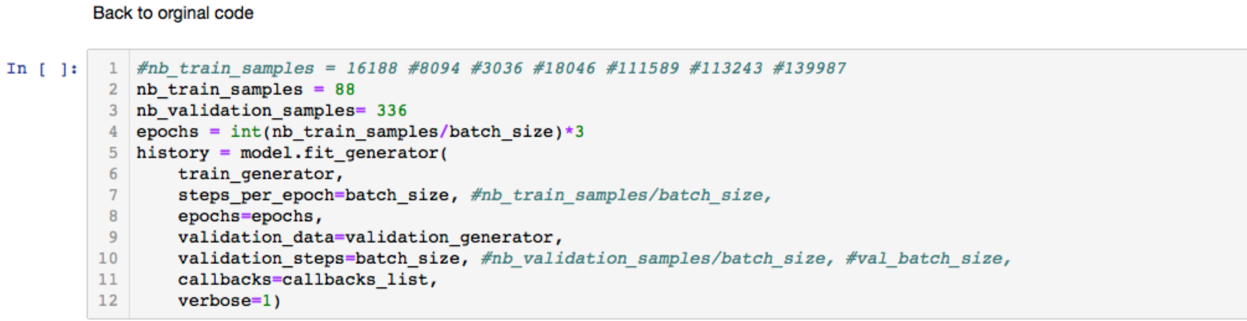
Batch: a set of N samples. The samples in a batch are processed independently, in parallel. If training, a batch results in only one update to the model.

A batch generally approximates the distribution of the input data better than a single input. The larger the batch, the better the approximation; however, it is also true that the batch will take longer to process and will still result in only one update. For inference (evaluate/predict), it is recommended to pick a batch size that is as large as you can afford without going out of memory (since larger batches will usually result in faster evaluation/prediction).
Epoch: an arbitrary cutoff, generally defined as "one pass over the entire dataset", used to separate training into distinct phases, which is useful for logging and periodic evaluation.
Tensorflow and Keras has discovered some connections.

NIH Dataset

This NIH Chest X-ray Dataset is comprised of 112,120 (1024 x 1024 resolution) X-ray images in png format with disease labels from 30,805 unique patients. To create these labels, the authors used Natural Language Processing to text-mine disease classifications from the associated radiological reports. The labels are expected to be >90% accurate and suitable for weakly-supervised learning. The dataset takes 45GB of disk space and is separated as follows: images_001.zip: Contains 4999 images, images_002.zip: Contains 10,000 images, images_003.zip: Contains 10,000 images, images_004.zip: Contains 10,000 images, images_005.zip: Contains 10,000 images, images_006.zip: Contains 10,000 images, images_007.zip: Contains 10,000 images, images_008.zip: Contains 10,000 images, images_009.zip: Contains 10,000 images, images_010.zip: Contains 10,000 images, images_011.zip: Contains 10,000 images, images_012.zip: Contains 7,121 images, README_ChestXray.pdf: Original README file, BBox_list_2017.csv: Bounding box coordinates. Note: Start at x,y, extend horizontally w pixels, and vertically h pixels, and Data_entry_2017.csv: Class labels and patient data for the entire dataset. There are 15 classes (14 diseases, and one for "No findings"). Images can be classified as "No findings" or one or more disease classes: Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural thickening, Cardiomegaly, Nodule Mass, and Hernia. Because the dataset was cleaned prior, no cleaning was required. Also, no outliers were discovered.

The code used for the Pneumonia dataset is ineffective on this dataset.

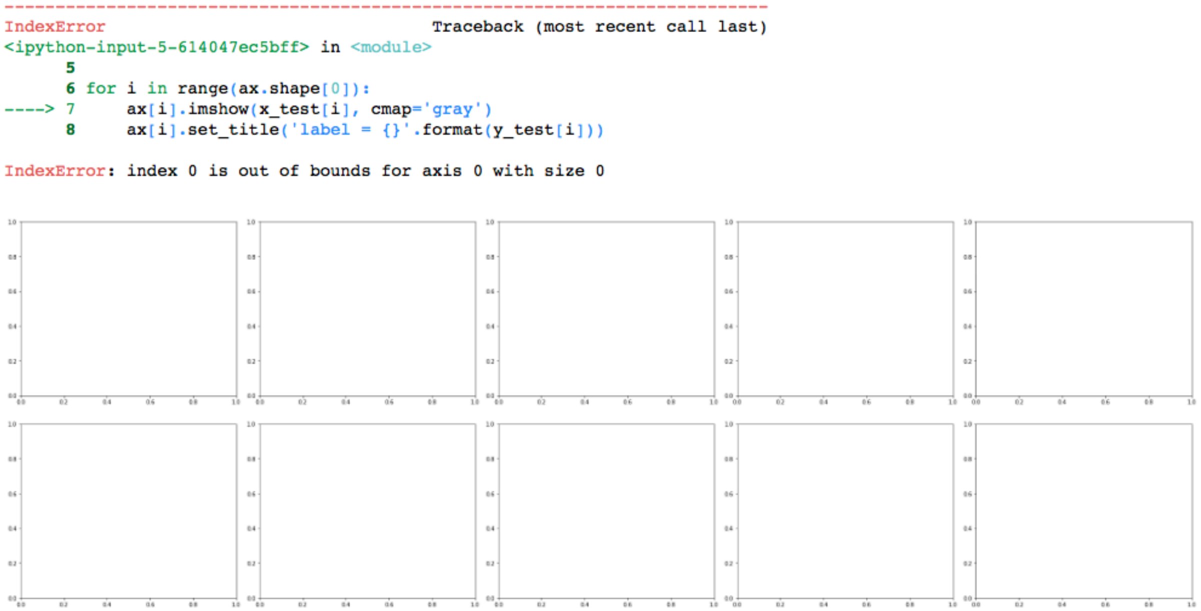


Back to orginal code

In [ ]:

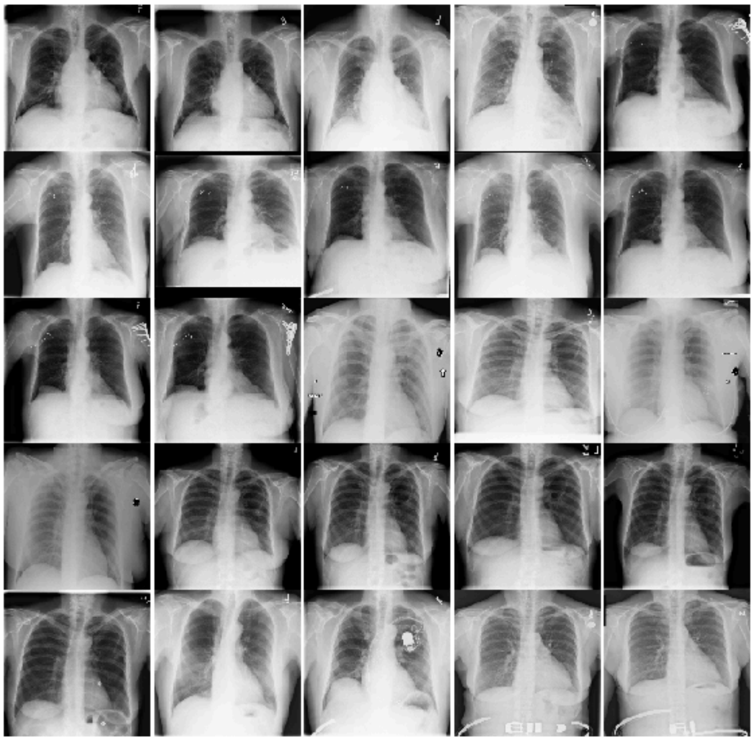
```
#nb_train_samples = 16188 #8094 #3036 #18046 #111589 #113243 #139987
nb_train_samples = 88
nb_validation_samples= 336
epochs = int(nb_train_samples/batch_size)*3
history = model.fit_generator(
    train_generator,
    steps_per_epoch=batch_size, #nb_train_samples/batch_size,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=batch_size, #nb_validation_samples/batch_size, #val_batch_size,
    callbacks=callbacks_list,
    verbose=1)
```

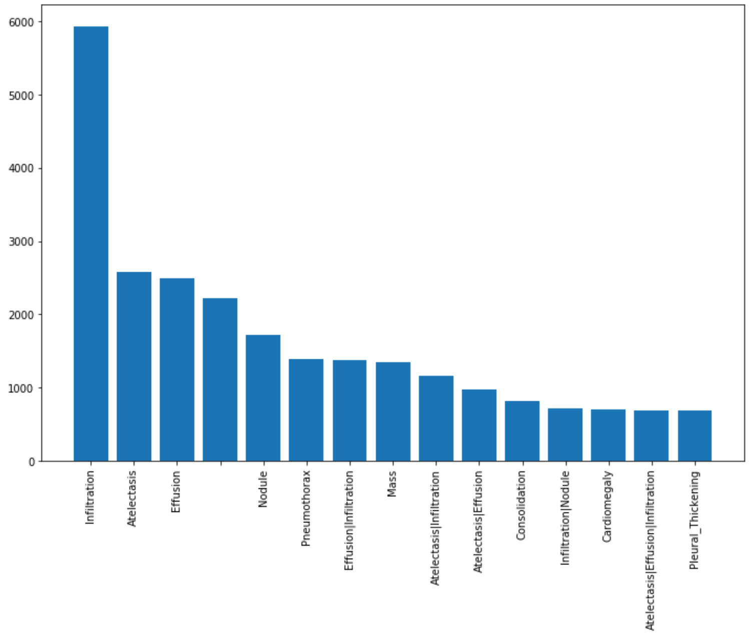
Normally it is impermissible to include error codes, but this screen capture shows the working code for the pneumonia dataset does not produce results for the NIH dataset. The exact cause is not known.



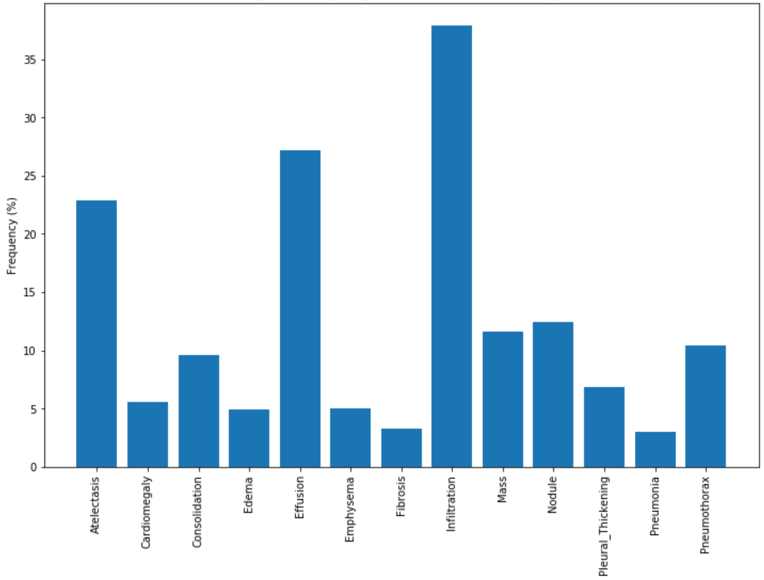
```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-5-614047ec5bff> in <module>
      5
      6 for i in range(ax.shape[0]):
----> 7     ax[i].imshow(x_test[i], cmap='gray')
      8     ax[i].set_title('label = {}'.format(y_test[i]))

IndexError: index 0 is out of bounds for axis 0 with size 0
```

Code written specifically for this dataset gathers labels from a csv file and is also able to analyze the images to determine the ailment being searched for. This code was originally set to discover Fibromyalgia but was modified to detect Pneumonia.

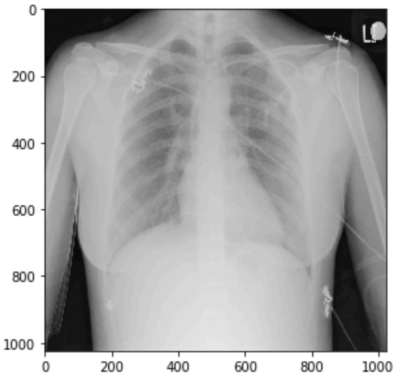


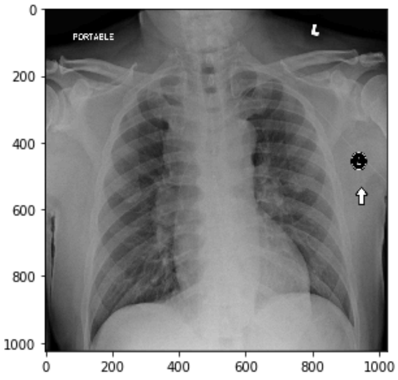


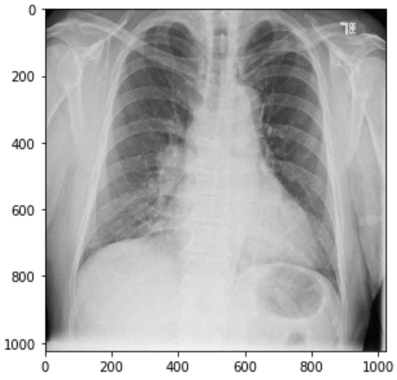
Adjusted Frequency of Diseases in Patient Group

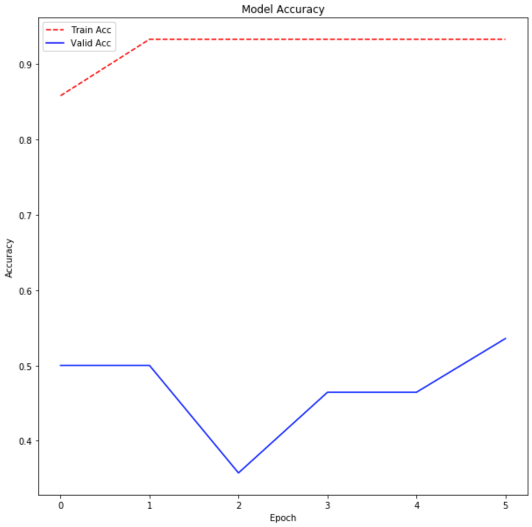


These graphs show the distribution per ailment from the csv file.

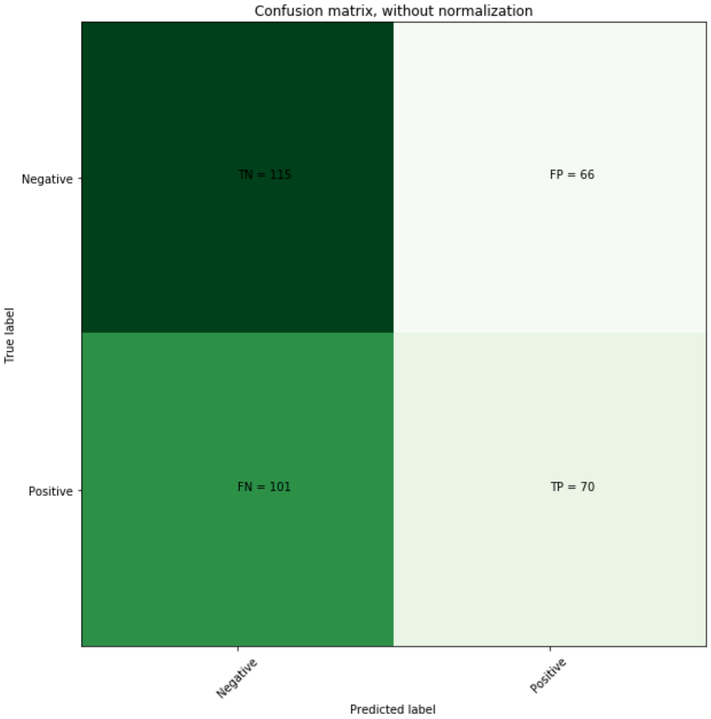








The model designed specifically for this dataset loads the images, upsamples the classes to match (to correct for the unbalanced condition), splits the images into train, test, and validation, and uses Keras (Tensorflow as the backend) to classify the images.


Confusion matrix, without normalization

RESULTS

The model discovered some connections. The settings need to be adjusted as false identification as pneumonia is 101 compared to true identification at 70. With the number of Epochs increased, the accuracy will also be increased but this will also require decent computer processing power due to the number of images being processed.

CONCLUSION

The models created for each dataset do work. With further tuning and adjusting, it is expected to become a very useful tool. Why the model and code did not work on the other dataset is currently unknown whether it's file size, image size, image format, color space for the image (RGB, CYMK, indexed color, …), bit size, or something else. Changing sample sizes and epochs are likely to increase the accuracy and reduce false identifications. With a computer with greater processing power, its likely these answers would be discovered.