Springboard Data Science Capstone 3 Project: Diagnosing Pneumonia from Chest X-ray Images

Thomas Stern

August 17th, 2022

Introduction:

Pneumonia is an infection of the lungs causing inflammation and fluid build up. These infections can be bacterial or viral in nature and are also further categorized by their acquisition. There is community acquired pneumonia and hospital acquired pneumonia. Oftentimes patients with respiratory complaints are rehospitalized only days later with pneumonia. This could be attributed to a hospital acquired infection or it could be that pneumonia is being underdiagnosed during the first hospitalization. Conventionally, a chest X-ray will be performed to rule out pneumonia. If pneumonia is not being seen during the patient's first hospitalization, maybe radiologists are missing something within these images.

Chest X-rays are a quick and inexpensive diagnostic tool used in emergency rooms and primary care all over the world. They have proven themselves to be relatively reliable but misdiagnosis is still a large problem compared to other radiographic techniques of this region (ie chest CT or Ultrasound). One meta-analysis from the Society of Critical Care Medicine found that chest X-rays have an overall sensitivity of 48%, whereas Ultrasound sensitivity was 95% and both had similarly high specificities of 92-94%. It seems that either the images are not showing pathology or radiologists are not able to effectively diagnose based on the images. The American Journal of Emergency Medicine puts the sensitivity of chest X-rays for pneumonia specifically to be between 38% and 76%. It is my hope that machine learning tools, such as convolutional neural networks, may be able to diagnose based on image data better than human experts.

The Data:

In this project I aim to make a model that performs at least as well as human experts using data from Guangzhou Women and Children's Medical Center. There are 5863 chest X-ray images of pediatric patients between 1 and 5 years old. Each image is labeled as Normal or Pneumonia. Images labeled Pneumonia were further classified into Bacterial/Viral categories. These X-rays were taken during routine visits and analyzed by 2 professional radiologists.

This data was relatively easy to clean/process as there was no missing information and all required features were contained within the image and the file name/directory. The images were housed in Train, Test and Val folders. There were 5216 images in the Train folder, divided into NORMAL (containing 1341 images) and PNEUMONIA (containing 3875 images) folders. The Test folder had 624 images, with 234 in the NORMAL folder and 390 in the PNUEMONIA folder. The Val folder contained only 16 images with 8 in each category. Upon first review of this data it seemed that the Val folder was much too small and the Train set was further broken down into a second Train and second Val set. I chose to make this 2nd Val set 10% the size of the Train set so that the Test and Val sets would add up to roughly 20% of the total data. Once the CNN model was created I would try both Train and Val sets to see which provided better results on the Test set.

Upon visualization of the diagnosis distribution, as seen in Figure 1, we can see the data is not balanced. There are significantly more images with pneumonia diagnosis than those without. It is unclear how this compares to the true distribution seen in all chest X-rays but this imbalance would need correction before using a CNN model either way. The Train set was around 74% pneumonia images and 26% normal. The Test set's distribution was slightly different with 63% pneumonia images and 37% normal. The Val set was 50/50 for each diagnosis.

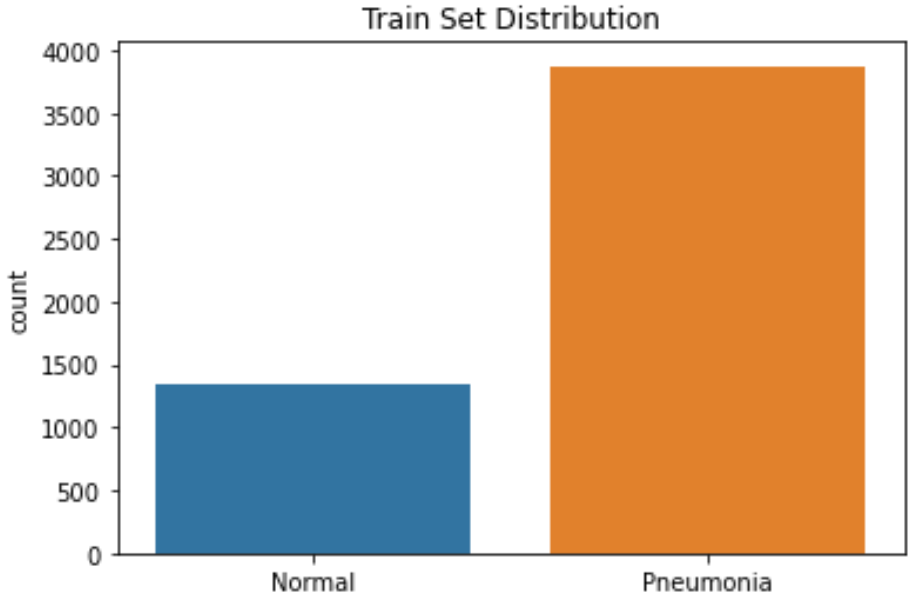


Figure 1: Comparing number of pneumonia cases with number of normal cases contained within the train set

Data Processing:

As seen in **Figure 2**, when 16 random images were pulled from the Train set, it was clear that each image had its own resolution, size and mean pixel values. These would have to be processed into uniform size and mean pixel value for the CNN model. All images were black and white with pixel values ranging between 0 and 255. CNN works best with smaller numbers so the image arrays were divided by 255 so that all image color maps would range from 0 to 1. The arrays were also reshaped so that every image dimensions were equal, with a value of 256 by 256. This size was chosen because it is a common image size used in neural networks yet is small enough to not slow down the model significantly.

Reshaping all these images results in different zoom values and some stretching/shrinking of image dimensions. The images were also each unique in their perspective of the patient, with some patients at an angle with respect to the X-ray machine. To best train a CNN model, ImageDataGenerator was used to create unique batches of images with different rotations, zoom, width shift and height shift. This creates more unique data where these arbitrary differences are not highly valued by the CNN model. These images were then randomly visualized to ensure no major loss in information was taking place. This function also normalized mean pixel value and their standard deviation to 0 and 1 respectively.

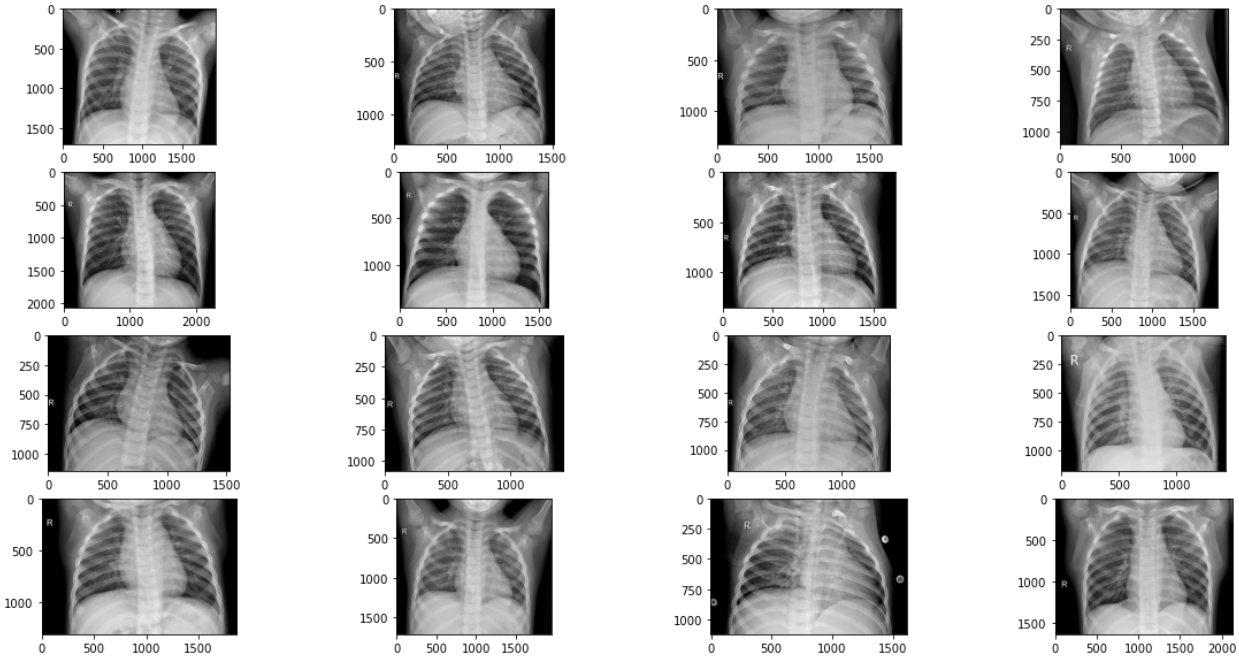


Figure 2: 16 randomly selected images from the Train set display variations in patient orientation

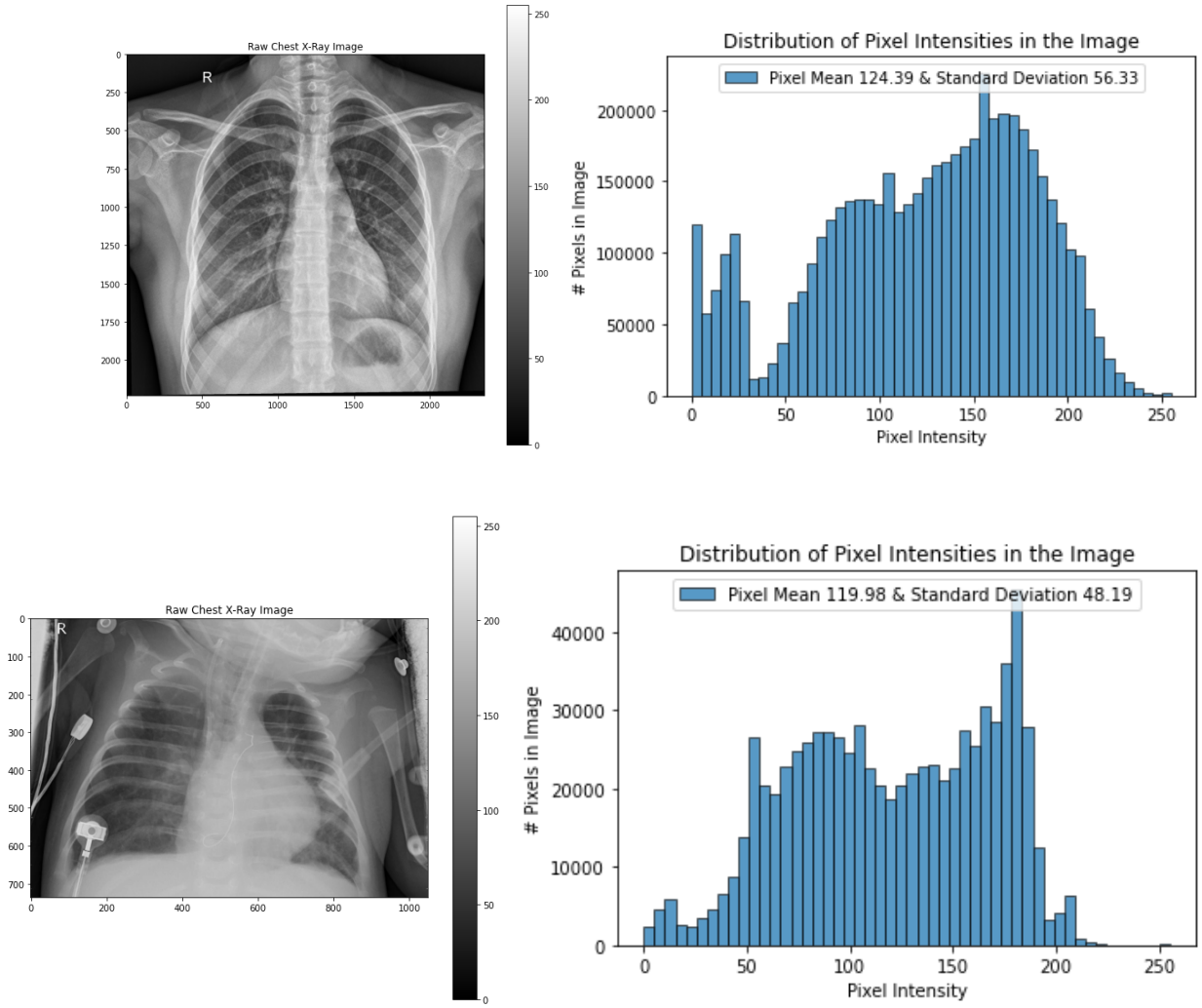


Figure 3: Randomly selected raw images show wide range in image size/dimensions as well as nonuniform pixel means and standard deviation of pixel values

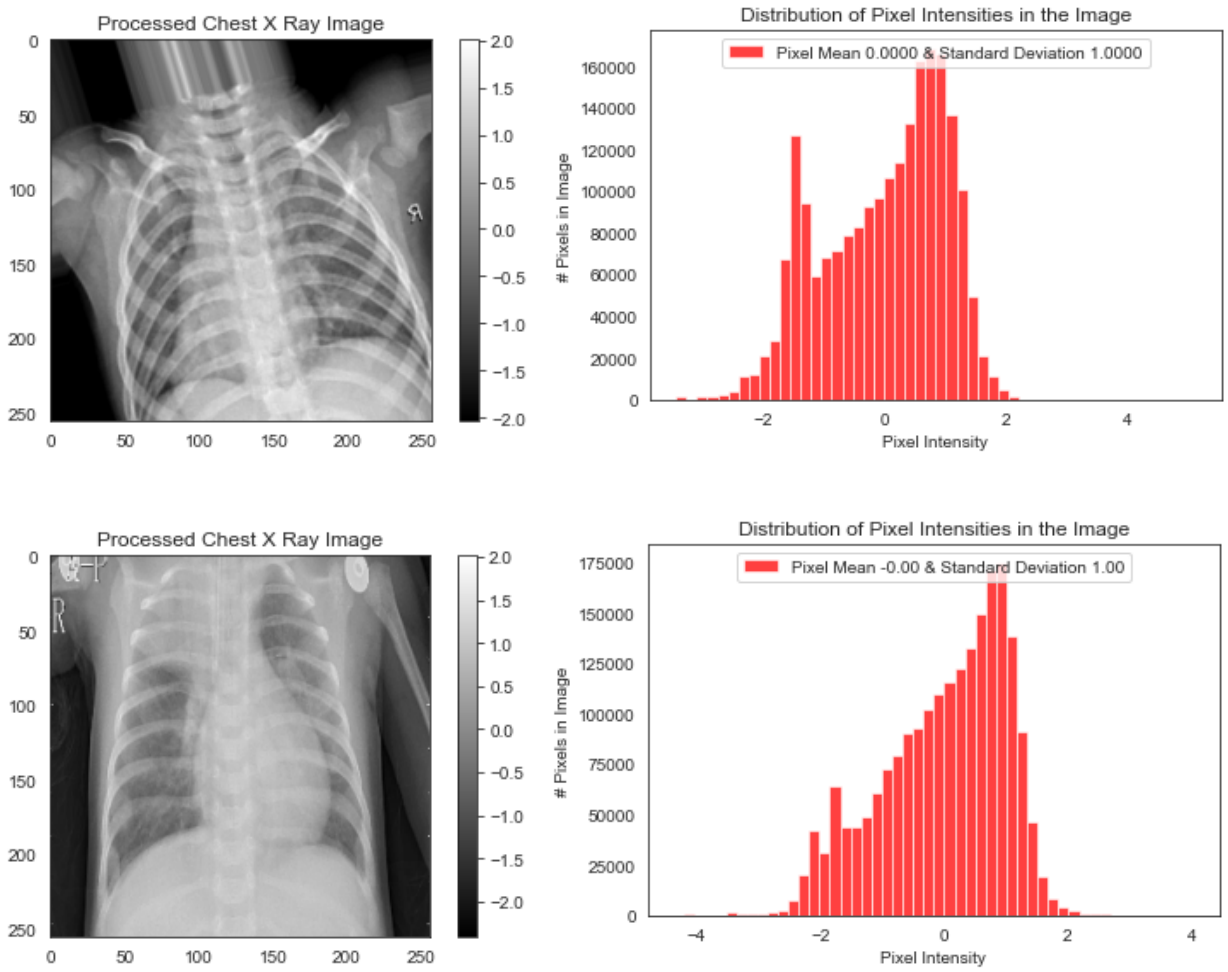
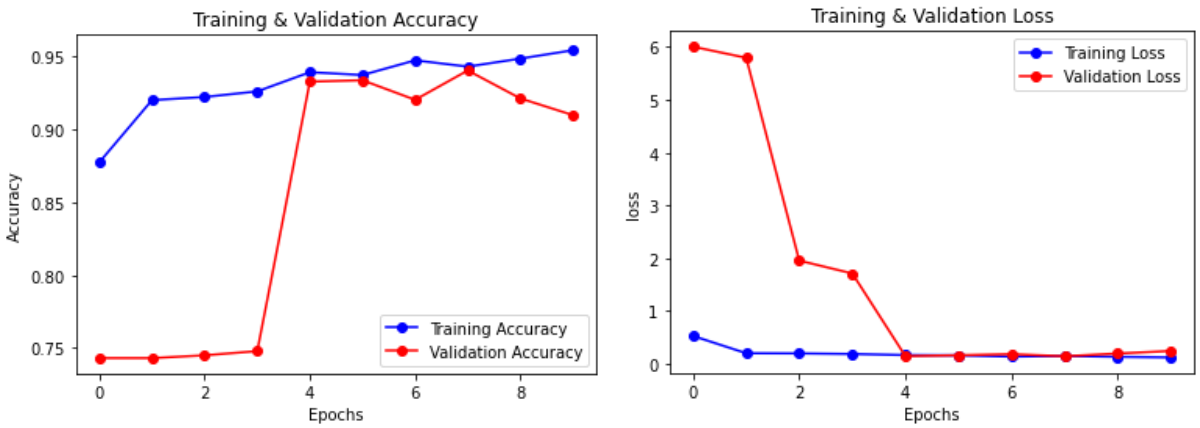


Figure 4: Random processed images show slightly altered images with normalized pixel mean and standard deviation.

Neural Network Model:

A CNN model was used as it is an excellent choice when considering image classification problems. This simple binary classification becomes much more complicated when dealing with images as complicated as an X-ray. There is so much room for variation from one image to the next and very few of these differences are diagnostic. A CNN model is able to separate out all these features found within the images and focus on those that have an impact on the final classification, while ignoring features that are irrelevant. These models are typically made up of a number of convolutional layers followed by pooling layers and end with fully connected, or dense, layers. The model is then compiled and fit to the training data. The validation set is used to continually test the model training to reduce overfitting. The Val data set that the images were initially housed in was only 16 images and did not have a similar class imbalance as the Train or Test sets. I decided to use the Test set as the validation set as it was a much more considerable percentage of total images and had a similar imbalance between classes.

My CNN model was made up of 5 convolutional/pooling layers with a number of dropouts to minimize overfitting. It was then fit to the training data which was created from a subset of the original train set. The validation data was created from the corresponding subset of the original train data. Two callbacks were created so that the model would alter parameters or stop if an increase/decrease in specific metrics was not seen in a set number of epochs. After fitting the model to the training data, the accuracy and loss of both the training set and validation set were analyzed. **Figure 5** shows these metrics change over a number of epochs. Ultimately we want accuracy for the model to be high for not only the train set but also for the validation and test sets. A high accuracy and low loss for both train and validation sets makes it less likely that the model is overfitting the data.



Model Evaluation:

This model displayed an accuracy of 0.88 and a loss of 0.32 on the test data

Accuracy and loss were the initial metrics in fine tuning the CNN model but our final metric would be sensitivity as this is where we see the most misdiagnosis. Sensitivity would be our most likely metric where we could beat human experts. Using this model to make predictions on the test set, it is possible to compare the label predictions with the true label of each image. **Figure 6** shows the confusion matrix for these predictions. It shows the number of false positives, false negatives, true positives and true negatives. With these numbers we can calculate sensitivity. Sensitivity is calculated by dividing the number of true positives by the number of false negatives plus number of true positives. Our initial goal was to beat the rather low, laying between 38-76%. Our model had a sensitivity of 83% which was far higher than sensitivity seen in the field.

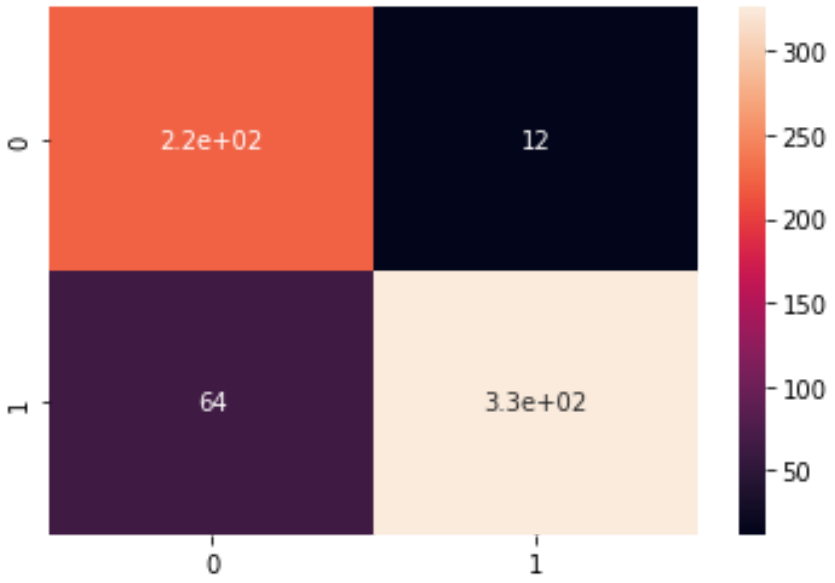


Figure 6: Confusion matrix shows number of true positives(330), true negatives(220), false positives(12) and false negatives(64)

Although sensitivity was the easiest metric to beat, it may not be the best summarizing metric for the model's performance. With this in mind, we also considered precision, recall and ROC_AUC scores. ROC_AUC seemed like the best summarizing metric and our model scored 0.966 which is extremely high.

	precision	recall	f1-score	support
0	0.78	0.95	0.85	234
1	0.96	0.84	0.90	390
accuracy			0.88	624
macro avg	0.87	0.89	0.87	624
weighted avg	0.89	0.88	0.88	624

Conclusion:

In conclusion it seems that CNN modeling can diagnose pneumonia with much higher accuracy and sensitivity than human experts. This leaves us with 2 possible solutions: start using CNN models in the field of medicine or start using different diagnostic tools that have higher sensitivity when observed by human experts.