# Machine Learning Engineer Nanodegree

## Capstone Proposal

Subramanian Vellaiyan
August 12th, 2018

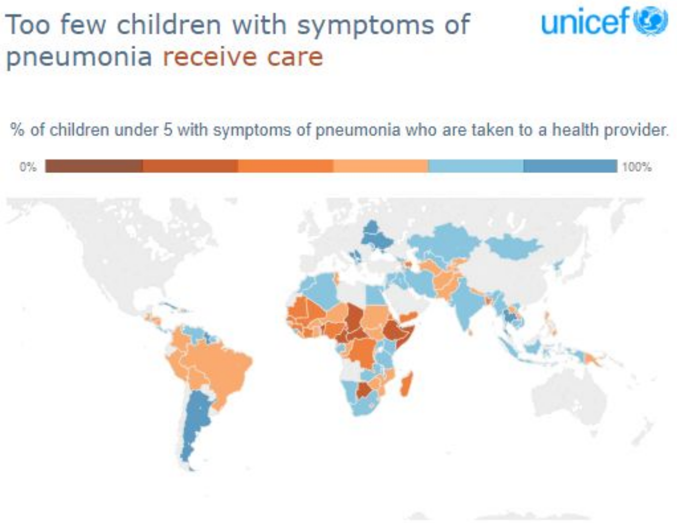
## Proposal

## Domain Background

The project proposal is Pneumonia Classification based on Chest X-Ray Images. The lungs fill with sacs of fluid and make breathing difficult. Pneumonia can affect any age group. Signs of pneumonia are a combination of respiratory symptoms, including ‘cough and fast or difficult breathing due to a chest-related problem’.

Top Stats:

- Pneumonia is the world’s leading cause of death among children under 5 years of age, accounting for 15% of all deaths of children under 5 years old.
- Pneumonia is the #1 most common reason for US children to be hospitalized.
- Viral pneumonias are the leading cause of hospitalization of infants


Too few children with symptoms of pneumonia receive care
unicef
% of children under 5 with symptoms of pneumonia who are taken to a health provider.
0%
100%

Motivation

- Chest X-rays are the most common imaging examination and if a machine can tell that there is a probability of Pneumonia before going to Doctor/Radiologist would help the person/guardian to take immediate measure.
- Radiologists to patient ratio might not be the same in all parts of the world. So a deep learning solution would help to detect the disease and send to the respective health provider at the earliest.
- Deep Learning/CNN - ability to solve real world problems. I am highly motivated to delve deep into this subject and understand an end-to-end process of image classification using CNN.

References

https://www.thoracic.org/patients/patient-resources/resources/top-pneumonia-facts.pdf

# Problem Statement

The problem statement is to classify a Chest X-Ray image into the below three levels

- Lung Opacity
- No Lung Opacity / Not Normal
- Normal

RSNA(Radiological Society of North America) has hosted a Pneumonia detection challenge in Kaggle. The data-set for this problem, has been taken from this competition.

# Datasets and Inputs

The input consists of train and test images in DICOM format. Each image has the patient id in the image name. Stage_1_detailed_class_info.csv file is taken as the input and it has been used to fetch images from train images folder.

## File descriptions

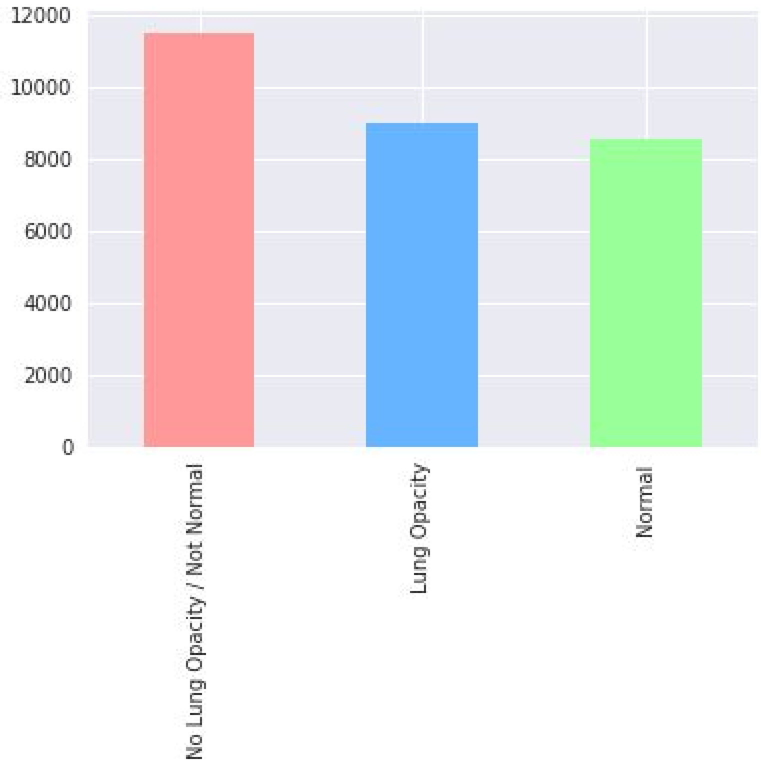
- stage_1_detailed_class_info.csv - provides detailed information about the type of positive or negative class for each image.
- Stage_1_train_images - Contains 25K train images with DICOM format
- stage_1_train.csv - the training set. Contains patient Ids and bounding box / target information.

## Data description

The input data is an image with size 1024x1024. The data is very huge and with limited resources, we might not be able to use all the data. But we will be able to use at least 10K images with effective resizing. There are three target classes

1. Lung Opacity
2. Not Normal No Opacity
3. Normal

Below is the distribution of the target classes:



There is no class imbalance but “Not norma”l cases are 10% higher than Lung Opacity and Normal cases. But this ratio seems to be a good mixture of data.

## Solution Statement

This is an image classification problem with the input image being a grayscale image. So the best solution is to build a combination of pre trained deep learning model and custom top layer using keras API with tensorflow backend. Note that the model should be able to process grayscale. Then the learning is transferred to the custom layer to adapt the output of the model as per the requirement which would be 3 levels in our case. Training will be on the entire network keeping weights fixed on the pre-trained network and take the best weights for the custom layer.

## Benchmark Model

The goal of the project is to classify whether a case is affected by Pneumonia or not. Also determine if it has signs showing that it has Pneumonia(cases where it is not normal). Identified cased would moved to a higher priority in the queue for treatment at health centers. A benchmark model would be to achieve at least 70% F-1 score so that both precision and recall is balanced. We don't want the model to classify all the cases as pneumonic and increase the queue of patients and on the other hand we don't want the model to classify all the cases as non pneumonic. This way we can build a model that balances between Precision and Recall.

## Evaluation Metrics

The core of understanding the performance matrix for a model is in the basic **confusion matrix**, which tracks the **true positives**, **true negatives**, **false positives**, and **false negatives**. We have a multi class model so let's take the example of opacity. In our case, a true positive is when we correctly predict a case as opacity, while a true negative is when we predict the normal or not normal. False positive is the reverse; when we predict opacity, but it does not occur. A false negative is when we predict no opacity, but then the case is a opacity.

### Accuracy

The most basic measure of model performance is accuracy. It simply tells you what percentage of all your predictions were correct.

*Formula = (true positive + true negatives) / total prediction*

However, accuracy can be misleading, because we are dealing with a relatively rare event. If we classify all the cases as either normal or not normal, we might still have 66.67% so we would consider this metric as the last important metric.
There are two other metrics - **Recall** and **Precision -** which are more useful in this case.

### Recall

Recall tells you how many of the total opacity cases were successfully predicted. If there are 10 opacity cases and we predict 7 of those, then we have 70% recall. Of course, if I predict opacity every time, then I will generate perfect recall, but very poor overall accuracy.

*Formula = (true positive) / (true positive + false positive)*

### Precision

Precision balances recall by telling of how many of the times which we predicted opacity, we were correct. So, if we predict opacity 10 times, but we only see opacity 7 times, then we have 70% precision.

*Formula = (true positive) / (true positive + false negative)*

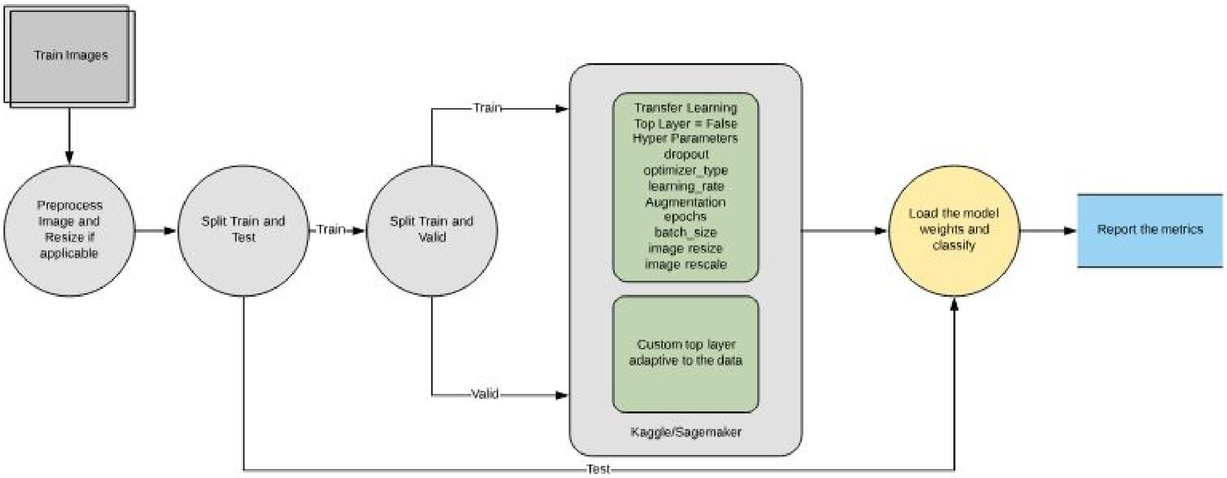
### F1

So we want to have both high recall and high precision.  Fortunately, there is another metric which combines these two called **F1**. A strong F1 score tells you that a model is generating both good recall and good precision.  That is, your model correctly predicts a large percentage of opacity and rarely predicts opacity incorrectly.

$$\mathit{Formula} = (1 + \beta^2)\ \mathit{x}\ (\mathit{precision\ x\ recall})\ /\ \beta^2\ \mathit{x\ precision\ x\ recall}$$

So as explained in the benchmark model, F1-score would be considered the prime metric along with Recall and Precision to evaluate the model for all the classes.

## Project Design



The above diagram shows the high level design of the project. So input will be a stratified sample of the train images. Then it is further split into train, test and valid. The train and valid images are passed onto the modelling section. The algorithm we are planning to implement hers is CNN and final model is a combination of pre-trained neural network and custom top layer to classify the three levels. A list of of hyperparameters are involved which can be configured before each run. Additionally data transformation like augmentation and image resizing can be configured before each run to see how it affects the model’s performance. Once the model with the weights are loaded, then it is used to classify the test data and report the metrics for evaluation.