

PNEUMONIA DETECTION Assignment

Interim Report



Team:

|  |  |
| --- | --- |
| **Name** | **E-mail** |
| Arup Kumar Mohanty | Arup.kumar1988@gmail.com |
| Mohit Gupta | Mohitgupta.mait@gmail.com |
| Ved Verma | Ved.verma.273306@gmail.com |
| Venkatesh Doddihal | Ds.venky1975@gmail.com |

Table of Contents

[**1.** **Summary of the problem statement, Data and findings** 4](#_Toc103201319)

[1.1. Problem Statement 4](#_Toc103201320)

[1.2. Project Objectives: 4](#_Toc103201321)

[**3.** **Pre-Processing** 9](#_Toc103201322)

[3.1. Pre-processing Methods 9](#_Toc103201323)

[3.2. **Pre-processing Applied** 9](#_Toc103201324)

[**4.** **Model and Model building** 11](#_Toc103201325)

[4.1. **Model Approach** 11](#_Toc103201326)

[4.2. **Model Evaluation Metrics** 11](#_Toc103201327)

[4.3. **Model creation** 11](#_Toc103201328)

[4.4. **Model Summary** 11](#_Toc103201329)

[**5** **Improve Model Performance** 12](#_Toc103201330)

[5.1 Approaches 12](#_Toc103201331)

[5.2 Summary 12](#_Toc103201332)

# **Summary of the problem statement, Data and findings**

## Problem Statement

Pneumonia is an infection in the lung, which requires review of a chest radiograph by highly trained specialists. Pneumonia shows up in a chest radiograph as an area of opacity. However, diagnosis of it can be complicated and much time and effort is spent by specialists in reviewing them. Chest radiograph is the most common performed diagnostic imaging study. Due to the high volume of chest radiography, it is very time consuming and intensive for the radiologists to review each image manually. As such, an automated solution is ideal to locate the position of inflammation in an image. By having such an automated pneumonia screening system, this can assist physicians to make better clinical decisions or even replace human judgement in this area.

## Project Objectives:

* To build a deep learning a pneumonia detection system, to locate the position of inflammation in an image.
* Use TensorFlow/Keras as the framework for building the model
* Read Medical images are stored in a special format called DICOM files (\*.dcm).
  1. **Data & Findings :**
* Details about the data and dataset files are given in below link,  
  <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data>
* The first step would be to examine the data available for this. The data is given in a zip file “rsna-pneumonia-detection-challenge.zip”, which contains the following items:
* A folder “stage\_2\_train\_images”: This folder contains all the training dataset chest radiograph DICOM images.
* A csv file “stage\_2\_train\_labels.csv”: This file contains the corresponding patientID images to the folder “stage\_2\_train\_images” and contains the bounding box of areas of pneumonia detected in each image along with a target label of 0 or 1 for pneumonia detected.
* A csv file “stage\_2\_detailed\_class\_info.csv”: This file contains the corresponding patientID images to the folder “stage\_2\_train\_images” and contains the target class labels of the images.
* A folder “stage\_2\_test\_images”: This folder contains all the test dataset chest radiograph DICOM images. We will not be using this set of images as they do not contain labels.
* A csv file “stage\_2\_sample\_submission.csv”: This file contains the corresponding patientID images to the folder “stage\_2\_test\_images”. We will not be using this set of file.

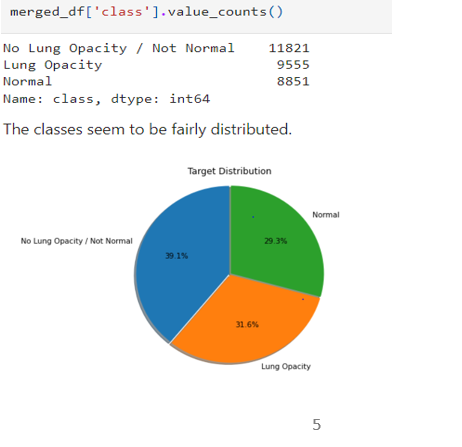
1. **Exploratory Data Analysis:**

**2.1. Approach:**

* To support the building of a neural network, the project will be done on **google Colab**.
* The first step is to unzip the zip file to open the above files to the google drive directory.
* Second is to verify the file format of the images provided, and they are all DICOM images in the “.dcm” file format. In order to open and read the DICOM images, we are using the **pydicom** library for that purpose. After that, the next step would be to inspect the csv files.

**2.2. Analysis:**

* Loading the “**stage\_2\_detailed\_class\_info.csv**” file into pandas dataframe, a quick glance reveals that it has only 2 columns:
* **patientId** – which refers to the patientId’s corresponding image name
* **class** – Target label of the patientId’s image



**Figure 1**

* As illustrate in Figure 1 , there are 3 types of classes: **Normal, Lung Opacity and No Lung Opacity / Not Normal**. The distributing is more or less even. The primary concern of the project would be to detect images with Lung Opacity, and the others would be in the same group labelling.
* Loading the “**stage\_2\_train\_labels.csv**” file into pandas dataframe, we can see that it has below fields:

**patientId** – which refers to the patientId’s corresponding image name

**x** - upper-left x coordinate of the bounding box

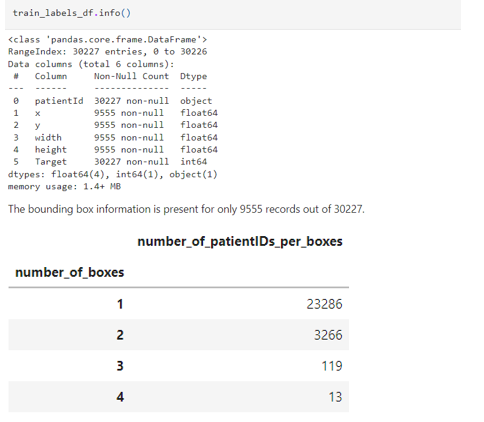
**y** - upper-left y coordinate of the bounding box

**width** – the width of the bounding box

**height** – the height of the bounding box

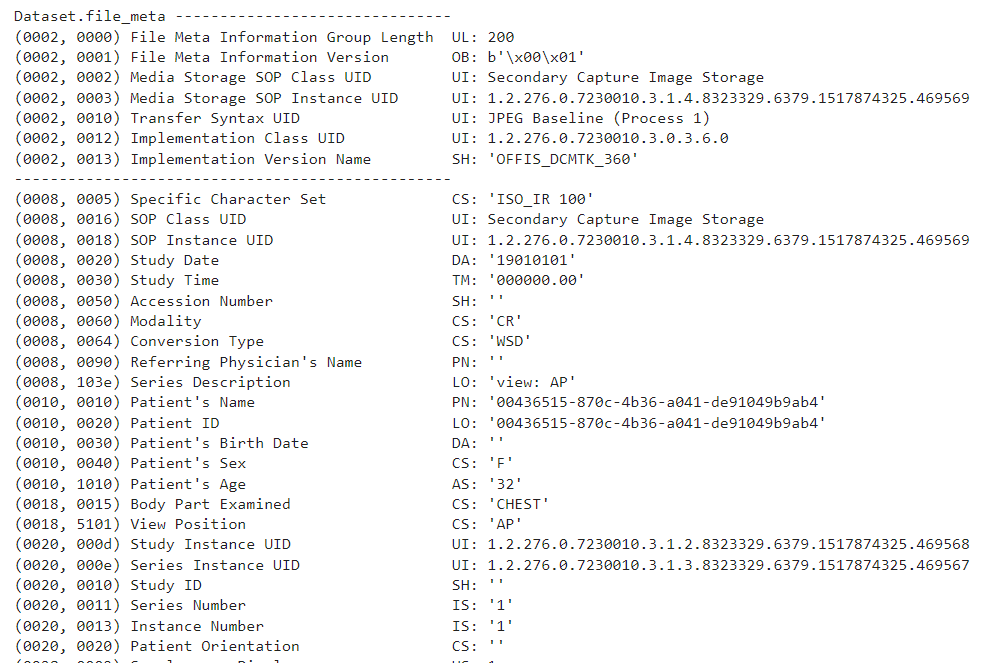
**Target** – binary target indicating if this image has evidence of pneumonia

* There is a total of 30,227 entries, no missing values with 9,555 images have bounding boxes. This corresponds to the data available “stage\_2\_detailed\_class\_info.csv” file.



**Figure 2**

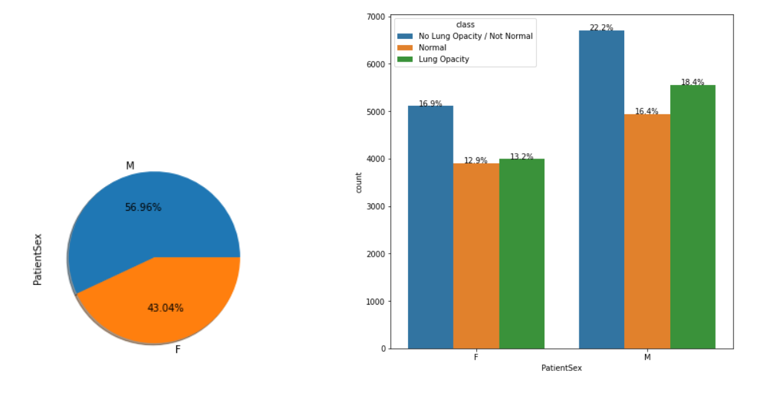
* There are multiple records for patients. Number of duplicates in patientID is 3,543.
* After merging both the csv files below are the observations.
* About 23,286 patientIds (~87% of them) provided have 1 bounding boxes while 13 patients have 4 bounding boxes. The reason is because each row records a single bounding box area of pneumonia detected. However, in a patient image, it might be the case of several bounding boxes area of pneumonia detected.
* Chest examinations with Target = 1 i.e. ones with evidence of Pneumonia are associated with Lung Opacity class.
* Chest examinations with Target = 0 i.e. those with no definitive evidence of Pneumonia are either of Normal or No Lung Opacity / Not Normal class.
* The next step is to read the images in the file “**stage\_2\_train\_images**”. Images provided are stored in DICOM (.dcm) format which is an international standard to transmit, store, retrieve, print, process, and display medical imaging information. We will make use of pydicom package here to read the images.

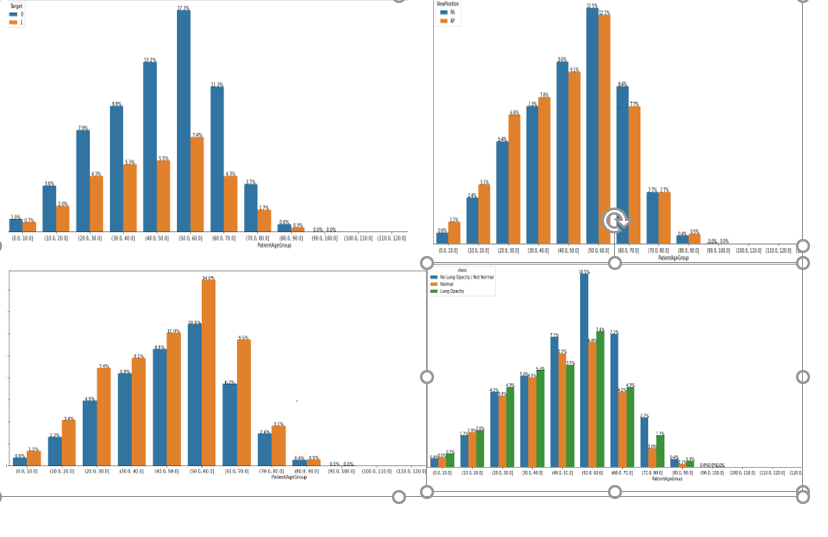


**Figure 3**

* From the above sample(figure 3) we can see that dicom file contains some of the information that can be used for further analysis such as **sex, age, body part examined , view position and modality**. Size of this image is 1024 x 1024 (rows x columns).
* To examine further we will merge the image features with the existing class data. This will help us understand distribution of age for those with evidence of lung opacity and those with no definite evidence of lung opacity.
* To understand distribution of male and female for those with evidence of lung opacity and those with no definite evidence of lung opacity
* Explore different view positions in the dataset
* Explore modality.
* We will pickle the file and do our analysis on the saved file.

**2.3 Visualization:**

* As we proceed further we will use different visualization techniques like univariate, multivariate analysis to discover patterns and anomalies in the data. 
* As shown above, There are 56.96% male patients and roughly 43% female patients. There are also more no of male patients having pneumonia compared to females.



* As illustrated above, a bivariate analysis of patient age group with features like viewposition, sex and target has below observations.

# **Pre-Processing**

This section describes the pre-processing steps applied to data before modelling. The images are in dicom format which contains lot of metadata along with pixel data. The pixel data needs to be extracted and converted to either jpg or png format.

# Pre-processing Methods

The following pre-processing methods can be applied to images.

* Conversion of image to jpg or png format.
* Find the number of channels in images and align to 1 or 3 channels.
  + Convert images to gray scale.
* Image resizing required as per base model requirements like 224\*224 for VGG16.
* Drop duplicate data.
* Set null values to 0 or drop the rows.
* Pixel normalization i.e scale the pixel values.
* Convert the pixel values to float.
* Image transformations like,
  + Rotate.
  + Flip vertically or horizontally.
  + Generate masks for the image.
  + Thresholding.
  + Erosion, Dilation.
  + Cropping.
  + Translation.
  + Noise addition. Etc.

# **Pre-processing Applied**

The data generators are used to pre-process the image. The images are resized to 224\*224 and processed in batches of 32.

Duplicate rows are dropped from merged data frame “train\_feature\_engineered”.

The total number records after dropping duplicates are 26684.

The distribution of target variable and classes are given below.

Distribution of target and classes

0 20672

1 6012

Name: Target, dtype: int64

No Lung Opacity / Not Normal 11821

Normal 8851

Lung Opacity 6012

Name: class, dtype: int64

The shape of data after split into train, test and validation are as below.

Shape of the dataframes:

TRAIN:(21348, 3)

VALID:(2668, 3)

TEST:(2668, 3)

The data distribution is proper across train, test and validation data set.

Distribution of target in the training set:

0 0.78

1 0.22

Name: Target, dtype: float64

Distribution of target in the validation set:

0 0.78

1 0.22

Name: Target, dtype: float64

Distribution of target in the test set:

0 0.77

1 0.23

Name: Target, dtype: float64

# **Model and Model building**

# **Model Approach**

# **Model Evaluation Metrics**

# **Model creation**

# **Model Summary**

# **Improve Model Performance**

## Approaches

## Summary