1. Overview

In this capstone project, the goal is to build a pneumonia detection system, to locate the position of inflammation in an image.

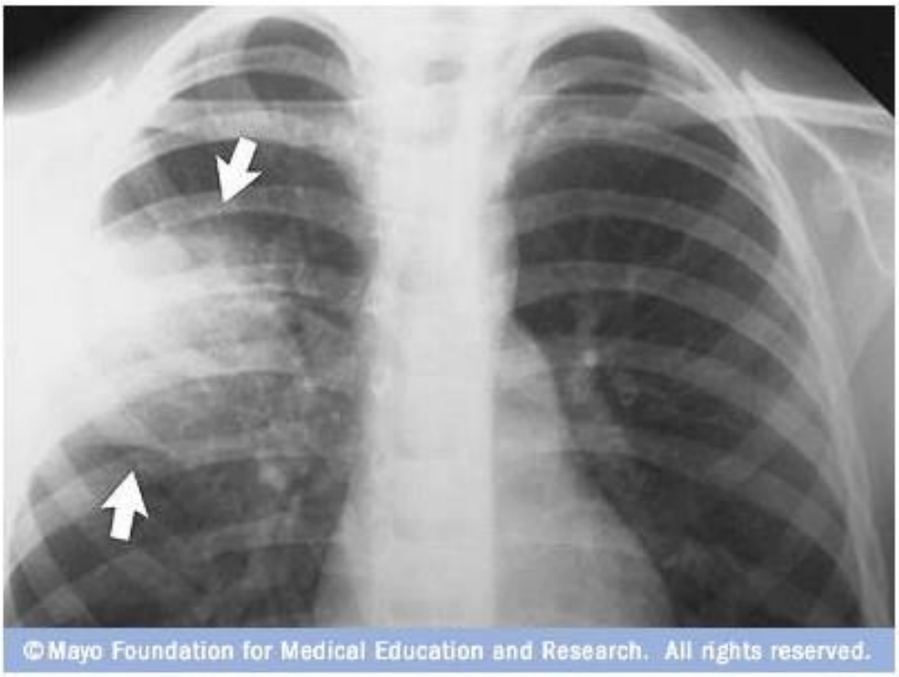
2. Introduction

2.1 Pneumonia: A respiratory disease

Pneumonia is a kind of lungs infection. It can be caused by bacteria, virus or fungi. In this infection air sacs or alveoli of the lungs fill up with fluid or pus.

2.2 Chest X-Ray

Chest x-rays (CXR) are a scan used to evaluate the lungs, heart and chest wall and can detect medical conditions such as: Pneumonia



The whitish part in the image indicate infection in the lung region.

3. Data

Data Source

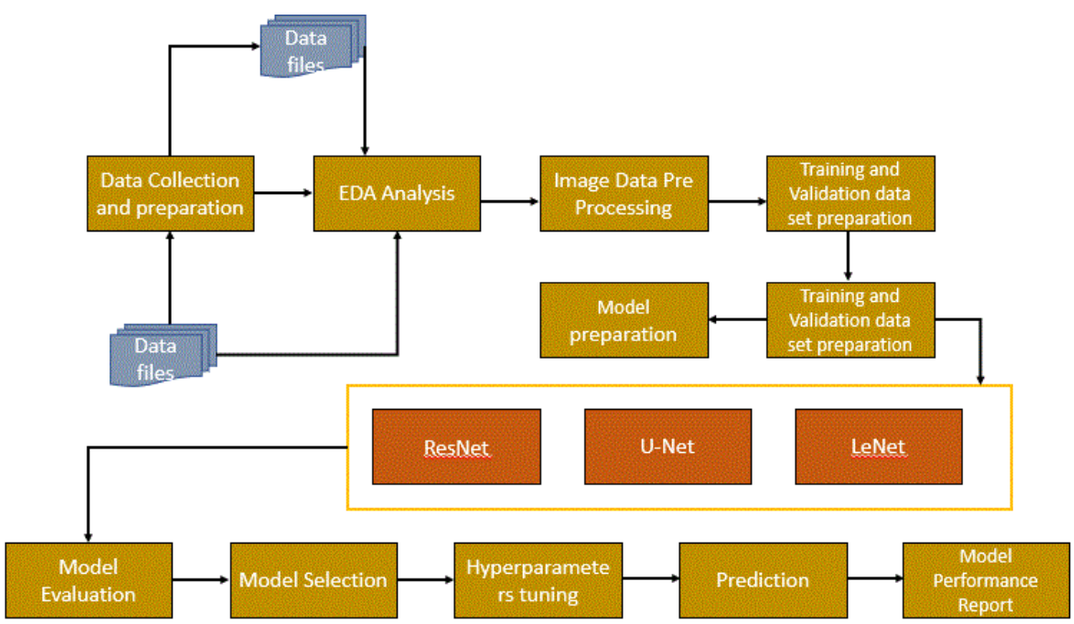
Data from ”RSNA pneumonia detection challenge” hosted by Kaggle, is used in this project. https://www.kaggle.com/c/rsna-pneumonia-detection-challengeDICOM Images

The image data is in Digital Imaging and Communications in Medicine (DICOM) format which includes images and image-related information.

4. Details of the Attributes

- patientId: Unique Identifier for a Patient.
- Age : Patient’s age in completed years.
- Gender: Patient’s gender is a categorical attribute with values of ’M’(male) ’F’(female)
- ViewPosition: This is a categorial attribute with values of ’PA’ and ’AP’
  - a. PA - PosteroAnterior: From back to front: When a chest x-ray is taken with the front against the film plate and the x-ray machine in back of the patient it is called an posteroanterior (PA) view.
  - b. AP - Anteroposterior: From front to back. When a chest x-ray is taken with the back against the film plate and the x-ray machine in front of the patient it is called an anteroposterior (AP) view.
- Class: Categorical attribute with the following values,
  - a. Lung Opacity
  - b. Normal
  - c. No Lung Opacity / Not Normal
- BoundingBox - These values denotes position of the lung opacity in the CXR image. There can be multiple opacities for a person
- Target - Target attribute denotes whether a patient is Pneumonia positive(1) or negative(0)

5. Process Flow



6. Exploratory Data Analysis (EDA)

Steps followed to perform EDA:
(Programming is done on google Collaboratory.)

Step 1. Downloaded data from Kaggle to google drive and unzipped there. Following files are extracted:

- stage_2_detailed_class_info.csv (contains class information)
- stage_2_sample_submission.csv (contains format of submission file)
- stage_2_train_images (folder containing DICOM files for training models)
- stage_2_train_labels.csv (contains target information)

Step 2. Installed pydicom package

Step 3. Loaded the file containing class information into dataframe class_df

Step 4. There are 30227 samples in the class_df dataframe. It has 26684 unique patient id and 3 unique classes(No Lung Opacity / Not Normal, Lung Opacity and Normal)

Step 5. Loaded the file containing target information into labels_df dataframe. It has five features - patientId, x, y, width, height, Target, where x and y are the upper-left x coordinate and upper-left y coordinate of the bounding box, respectively. width and height fields are the width and height of bounding box, respectively. Target is binary valued feature, indicates whether the sample has evidence of pneumonia.

Step 6. Missing value analysis:

- No missing value in class_df dataframe.
- 20672 missing values in labels_df dataframe.

The missing data in labels_df dataframe makes sense, as bounding box is only defined where Target value is 1

Step 7. From labels_df dataframe file we found 9555 records entries having pneumonia. Step 8. Classes are balanced

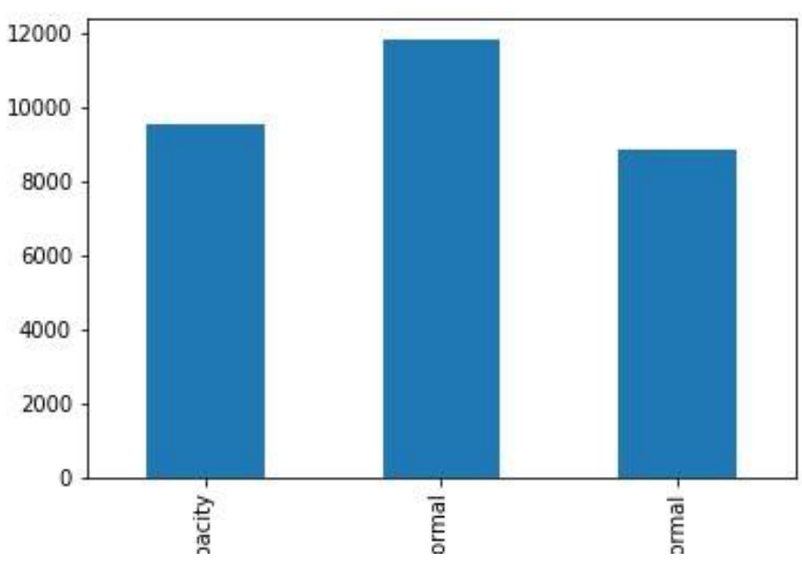


Figure 1

- 39% of records are of category ’No Lung Opacity / Not Normal’
- 31.6% of records are of category ’Lung Opacity’
- 29% of records are of category ’Normal’

Step 9. Checked and found that patientId column is same in both the dataframes class df and labels df

Step 10. Merged the two dataframes class df and labels df to label class df dataframe using Patient ID as the merge criteria.

Step 11.Found that for patients having pneumonia, the class value is

”Lung Opacity” Step 12.

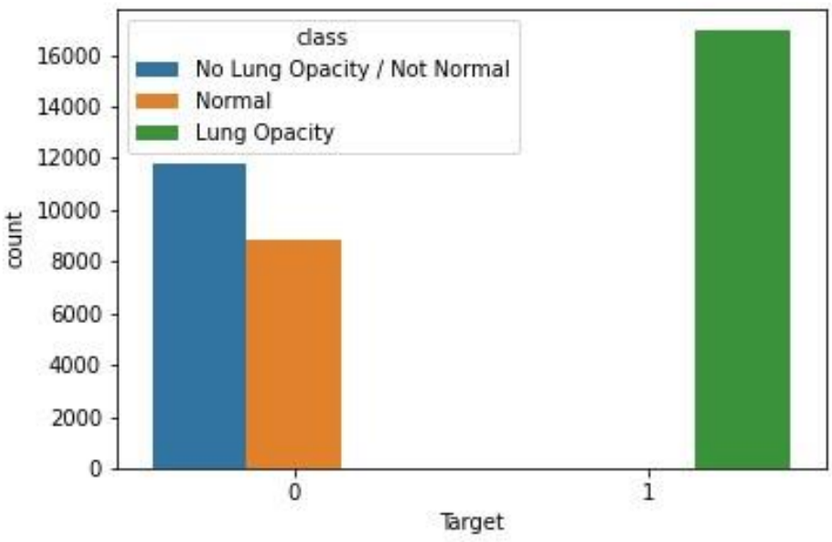


Figure 2

Plotted the number of examinations for each class detected, grouped by Target value. All entries with Target
= 1 is associated only with class ’Lung Opacity’. Entries with Target = 0 is associated with either class
’Normal’ or class ’No Lung Opacity / Not Normal’

Step 13. All DICOM files are read and metadata is saved to a seperate .csv file called as image metadata.csv and further this file is loaded to a dataframe called as image df.

Step 14. On inspecting the image df file we found following fields as relevant

- BodyPartExamined
- ConversionType
- Modality
- PatientSex
- ViewPosition

so other attributes are
removed

Step 15. BodyPartExamined, ConversionType and Modality have only one unique value. PatientSex and ViewPosition have 2 unique values. Therefore we dropped BodyPartExamined, ConversionType as well as Modality columns. And checked the unique value of PatientSex and ViewPosition cloumn

Step 16. We changed PatientID column of image df dataframe to patientId, so that we can join it with labels class df dataframe. The new dataframe formed is called as merged df.

Step 17. Viewed position(AP or PA) distribution

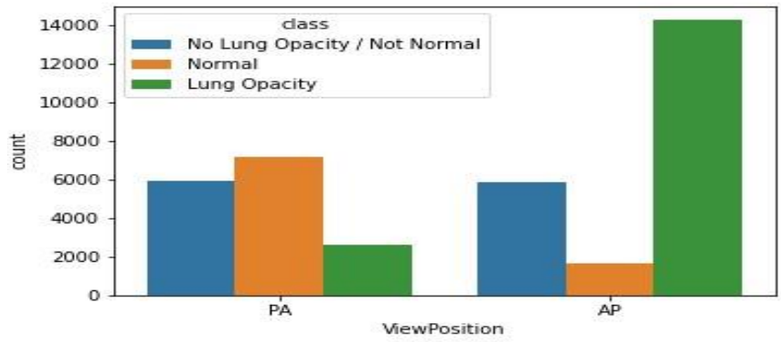


Figure 3

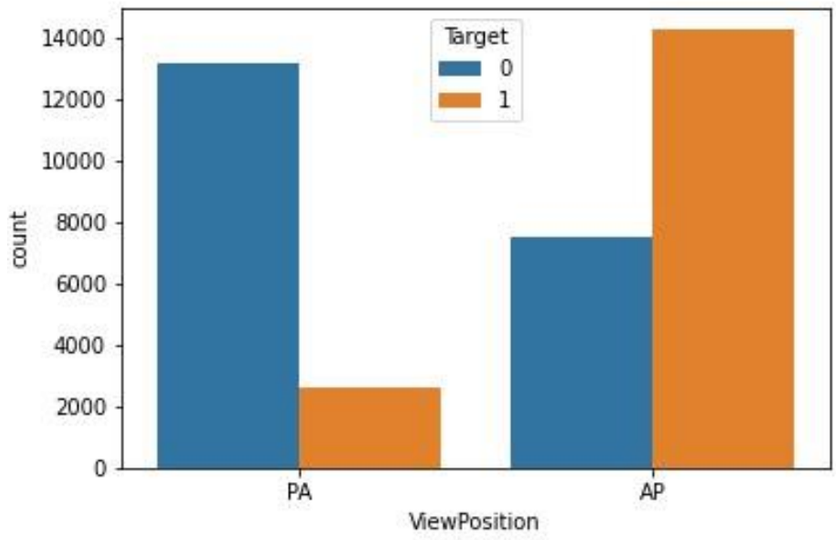


Figure 4

We observed that When ViewPosition is PA the, the number of records with Target = 0 are very high compared to records with Target = 1 and when ViewPosition is AP the, the number of records with Target
= 1 are very high compared to records with
Target = 0

Step 18. From distribution of patientsex attribute, it is observed that the attribute does not seem to have any effect on Target value

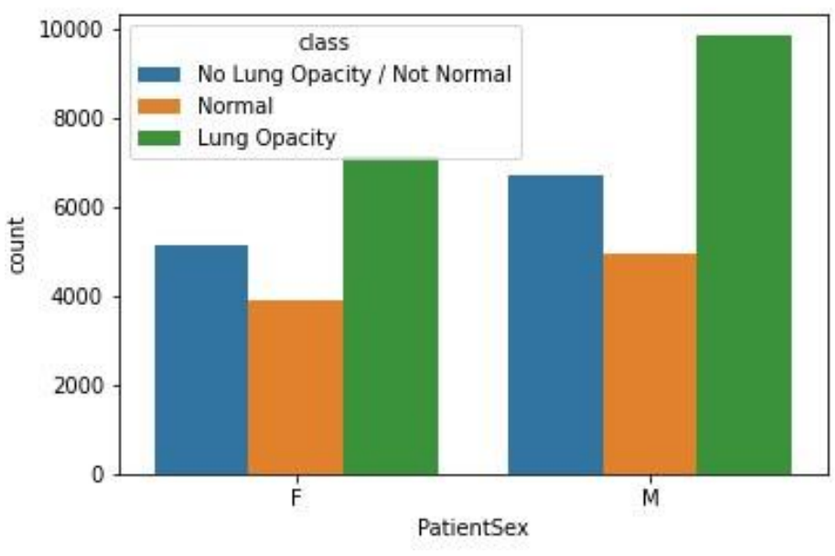


Figure 5

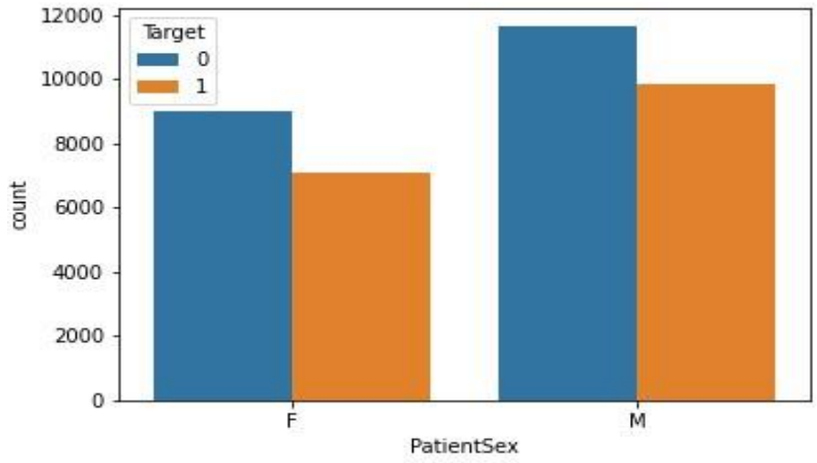


Figure 6

Step 19. From distribution of PatientAge attribute, it is observed that there are outliers

Step 20. patients are divided into five age groups. It is observed that most of the patients are adults

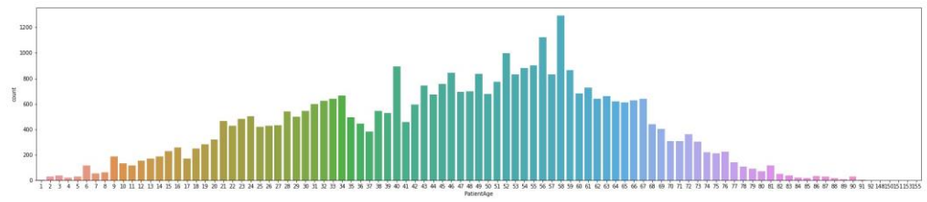


Figure 7

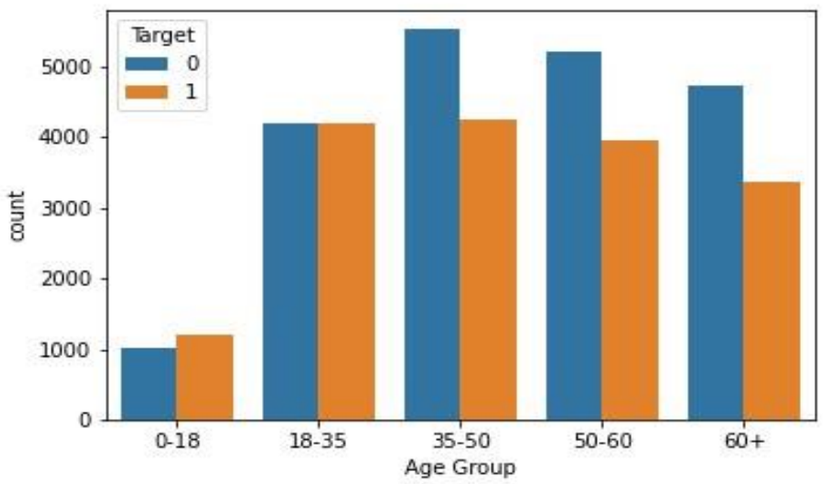
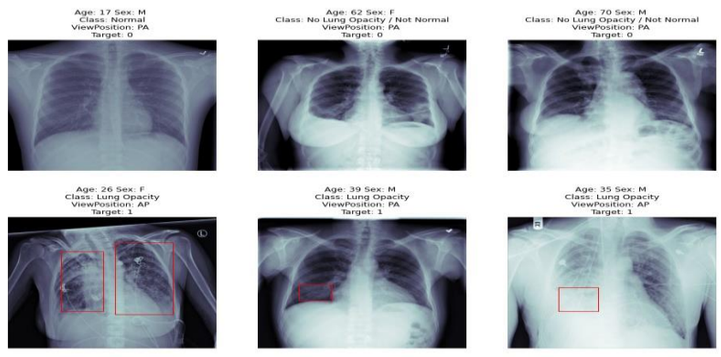


Figure 8

Step 21. Images having target 1 and target 0 are shown below.



Images are single channeled (Monochrome). and all images have same size i.e. 1024X1024

7. Inferences on EDA

After exploring the data, both the csv and DICOM data, we found that

1. Data Set is well balanced. 39% of records are of category 'No Lung Opacity / Not Normal'. 31.6% of records are of category 'Lung Opacity'. 29% of records are of category 'Normaal'
2. There are no real missing values in the csv data. The NaN values in stage_2_train_labels seem legit
3. Patients having neumonia have class value as "Lung Opacity"
4. In the training data, 22.53% records are having Pneumonia while 77.47% records does not have Pneumonia
5. Metadata present in dcm files does not seem to be of much importance for prediction
6. There is a good distribution of people from all ages in the data. The data shows that most number of patients in the sample are adults
7. PatientAge seems to have some incorrect data. We observed 5 outliers there
8. When ViewPosition is AP the, the number of records with Target = 1 are very high compared to records with Target = 0

8. Pre-Processing

8. 1 Dealing with Imbalanced Data

With reference Figure 1, classification of patient's images are more or less equally distributed, hence no specific strategy has been applied to overcome the imbalanced data scenarios.

8. 2 Image resize

The training images are single channeled (Monochrome) with fixed size but large in size (1024X1024). To achieve the better memory usage and time, images are normalized and resized to (256X256). For further visualization throughout project, both trained and validation deserialized images are rescaled to its original size for better clarity.

9. Model and Model building

9.1 Model based on ResNet Architecture

9.1.1 ResNet

To train the network model in a more effective manner, we herein adopt the same strategy as that used for DSSD (the performance of the residual network is better than that of the VGG network). The goal is to improve accuracy. However, the first implemented for the modification was the replacement of the VGG network which is used in the original SSD with ResNet. We will also add a series of convolution feature layers at the end of the underlying network. These feature layers will gradually be reduced in size that allowed prediction of the detection results on multiple scales. When the input size is given as 256 and 256, although the ResNet–101 layer is deeper than the VGG–16 layer, it is experimentally known that it replaces the SSD's underlying convolution network with a residual network, and it does not improve its accuracy but rather decreases it.

9.1.2 Model Block Architecture (based on ResNet Backbone)

We would create the convolutional neural network using ResNet blocks. The model consists of a down sampling block (consisting of Batch normalization, leaky ReLU, Convolutional 2D and max pool layer) and a ResNet block (consisting of batch normalization, leaky ReLU and

convolutional 2D layers). The down sampling and ResNet blocks are repeated based on configuration parameters of n_blocks and depth. In the end we up sample the data points to the same size as before in order to get the target mask.

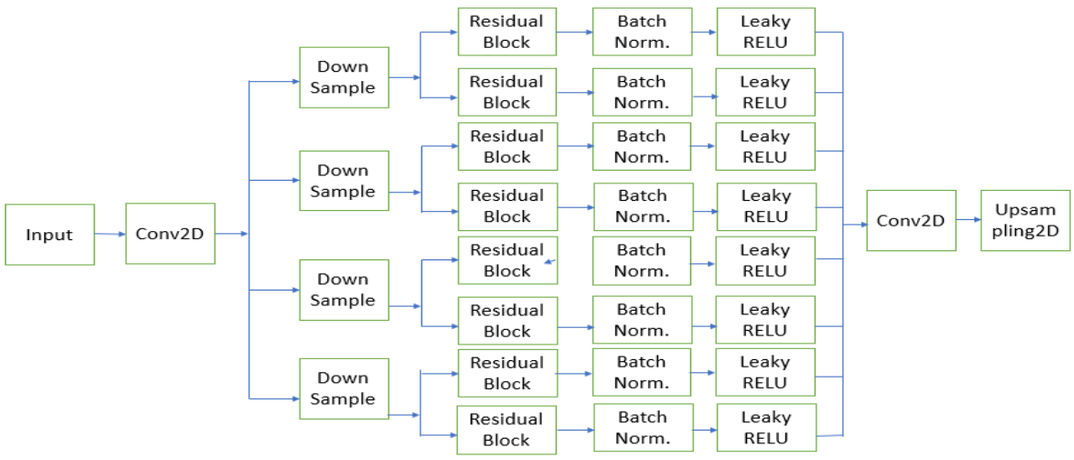


Figure 9

The block notation “DownSample” is again comprised of layers of BatchNormalization, LeakyReLU, Conv2D and MaxPool2D

Figure 10

The block notation “Residual Block” is again comprised of layers of BatchNormalization, LeakyReLU, Conv2D, BatchNormalization, LeakyReLU, Conv2D.

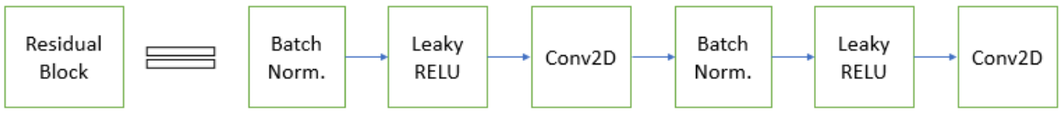


Figure 11

9.1.3 Loss Function

Neural networks are trained using an optimization process that requires a loss function to calculate the model error. We have defined IoU loss function to train the neural network which in turn use binary cross-entropy in order to minimize the model error. So, the model is compiled using below parameters…

- loss: use the loss function as mentioned above
- optimizers: use Adam optimizer
- metrics: accuracy (mean IoU)

9.1.4 Model Evaluation Metrics

Evaluation metrics are used to measure the quality of the statistical or machine learning model. Evaluating machine learning models or algorithms is essential for any project. There are many different types of evaluation metrics available to test a model. In this project context, we have used IoU (intersection over union) and Confusion matrix to express the model performance. The matrix IoU is used to as the model is primary build or trained to detect the object classified as “Lung Opacity” and its boundary box around the object. On the other hand, Confusion matrix has been used to evaluate the model performance with respect to ground truth.

9.1.5 Model performance

8.1.5.1 Loss Function evaluation

The trained model has been performed pretty well on validation dataset with accuracy closed to 97%. Model performance data of variation of loss and mean_iou has been captured for both training and validation dataset.

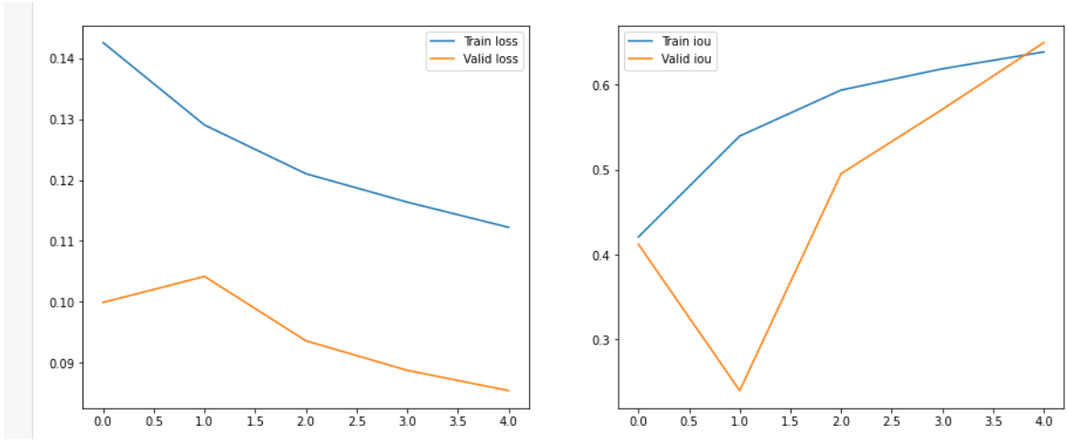
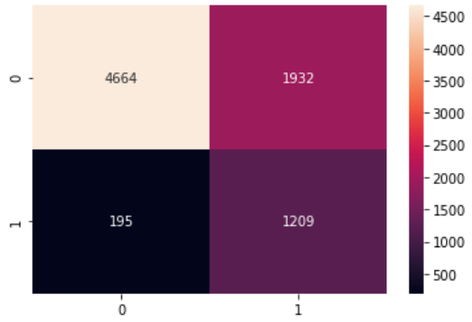


Figure 12

From the figure 12, we could interpret the model performance a bit underfit as training loss is higher than validation loss. This could be the reason of less availability of images classified as “Lung opacity” compared to images classified as “No Lung opacity/Not Normal” or “Normal”. The model performance could further be improved with Kfold/cross validation technique to overcome the underfit problem.

9.1.5.2 Confusion Matrix

The confusion matrix is captured the model performance on classification of images broadly into two categories “Lung Opacity” with target 1 and rest are with target 0. From figure 13, we could see that the F1 score for target 1 is 53% which gives an indication on further improvement of the model.



Below is the classification report based on the actual and predicted target values.

	precision	recall	f1-score	support
0	0.96	0.71	0.81	6596
1	0.38	0.86	0.53	1404
accuracy			0.73	8000
macro avg	0.67	0.78	0.67	8000
weighted avg	0.86	0.73	0.76	8000

AIML/.../Final classification and detection.ipynb

Figure 13

9.2 Model based on U-Net Architecture

9.2.1 U-net

The main idea is to supplement a usual contracting network by successive layers, where pooling operations are replaced by upsampling operators. Hence these layers increase the

resolution of the output. What's more, a successive convolutional layer can then learn to assemble a precise output based on this information.

9.2.2 Model Block Architecture (based on U-Net Backbone)

We would create the convolutional neural network using U-Net blocks. The model consists of a sequence of a convolutional 2D layers and MaxPooling followed upsampling layers

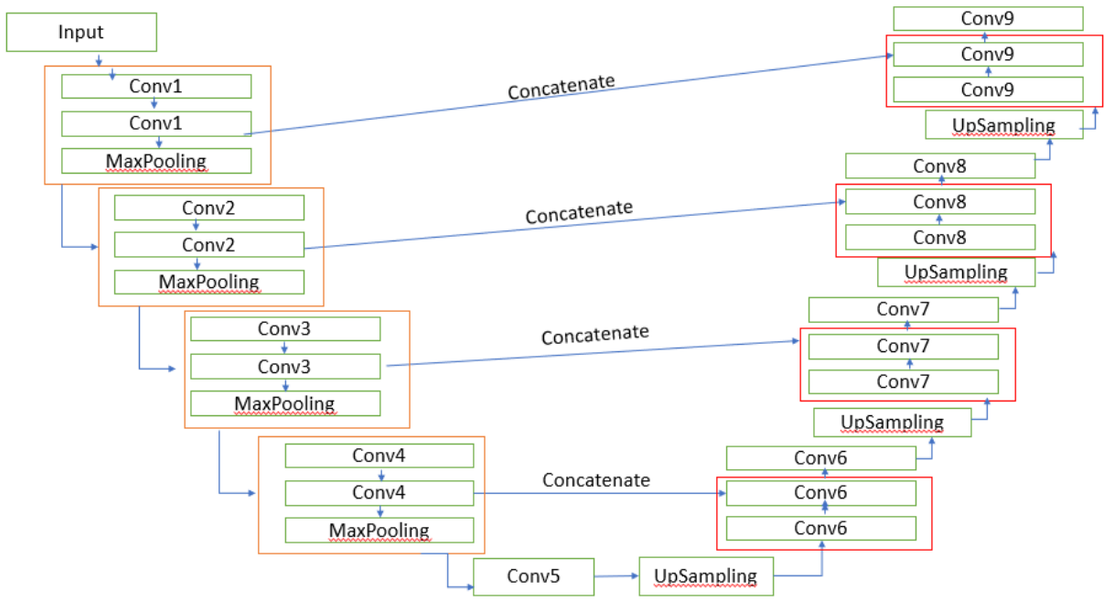


Figure 14

9.2.3 Loss Function

Neural networks are trained using an optimization process that requires a loss function to calculate the model error. We have defined IoU loss function to train the neural network which in turn use binary cross-entropy in order to minimize the model error. So, the model is compiled using below parameters…

- loss: use the loss function as mentioned above
- optimizers: use Adam optimizer
- metrics: accuracy (mean IoU)

9.2.4 Model Evaluation Metrics

Evaluation metrics are used to measure the quality of the statistical or machine learning model. Evaluating machine learning models or algorithms is essential for any project. There are many different types of evaluation metrics available to test a model. In this project context, we have used IoU (intersection over union) and Confusion matrix to express the model performance. The matrix IoU is used to as the model is primary build or trained to detect the object classified as “Lung Opacity” and its boundary box around the object. On the other hand, Confusion matrix has been used to evaluate the model performance with respect to ground truth.

9.2.5 Model performance

9.1.5.1 Loss Function evaluation

The trained model has been performed pretty well on validation dataset with accuracy closed to 96%. Model performance data of variation of loss and mean_iou has been captured for both training and validation dataset.

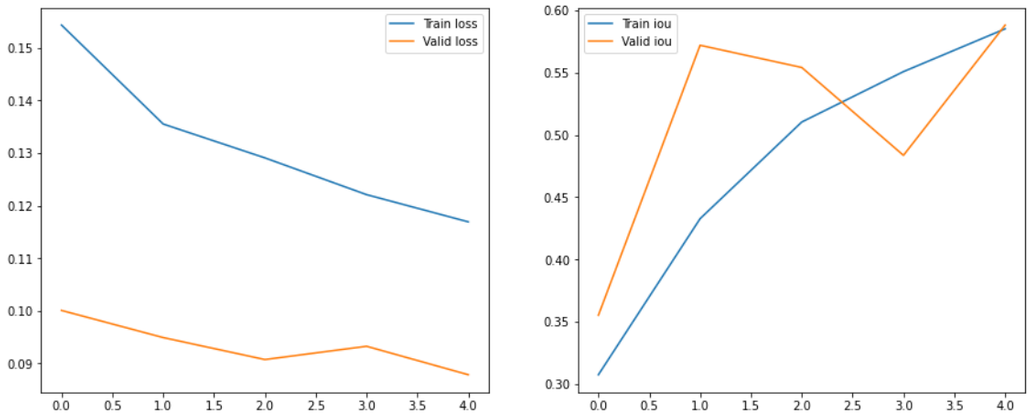
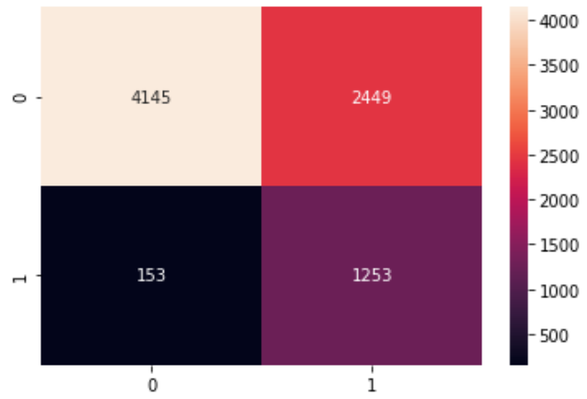


Figure 15

9.2.5.2 Confusion Matrix

The confusion matrix is captured the model performance on classification of images broadly into two categories “Lung Opacity” with target 1 and rest are with target 0. From figure 13, we could see that the F1 score for target 1 is 49% which gives an indication on further improvement of the model.



Below is the classification report based on the actual and predicted target values.

	precision	recall	f1-score	support
0	0.96	0.63	0.76	6594
1	0.34	0.89	0.49	1406
accuracy			0.67	8000
macro avg	0.65	0.76	0.63	8000
weighted avg	0.85	0.67	0.71	8000

Figure 16

10. Model Comparison Evaluation

We have trained our model on both ResNet and U-Net architecture and captured performance metrices on validation data respectively. Both the model performed nearly same and has many scopes for further improvement. However, the model built and trained upon ResNet architecture comparatively performed better in both image classification and boundary box identification.

11. Hyperparameters optimization

In machine learning/deep learning, we often come across the name “parameters” . In broadly, we can categories those parameters into two part namely implicit parameters and explicit parameters. The implicit parameters are those parameters which controls by integrated algorithm for a given problem context. Whereas the explicit parameters (commonly known as **Hyperparameters**) whose value is used to control the learning process for a given model. Controlling of such parameters often comes handy to optimize the model performance.

The example of a model hyperparameter is the topology and size of a neural network. Also, the example of algorithm hyperparameters are learning rate and batch size and often these hyperparameters are interconnected. Changes in one often demands the change in the other to get the effect.

Learning rate : The parameter "learning rate" is hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. Choosing the learning rate is challenging as a value too small may result in a long training process that could get stuck, whereas a value too large may result in learning a sub-optimal set of weights too fast or an unstable training process.

Batch size : The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters.

11.1 Model hyperparameter optimization

The problem statement of our project is a well known and available in Kaggle. So, we have consulted high performed solutions available in the Kaggle.

In our project, we have used well known neural network topology "Resnet" to define our model. So, we don't need to further hyper tune our model topology unless the Resnet topology is needed some optimization.

11.2 Algorithm hyperparameter optimization

We have tuned our model (with small dataset) with different learning rate (within 0.0 to 1). We can see that the model was able to learn the problem well with the learning rates 1E-1, 1E-2 and 1E-3, although successively slower as the learning rate was decreased. With the chosen model configuration, the results suggest a moderate learning rate of 0.001 results in good model performance on the train and validation sets.

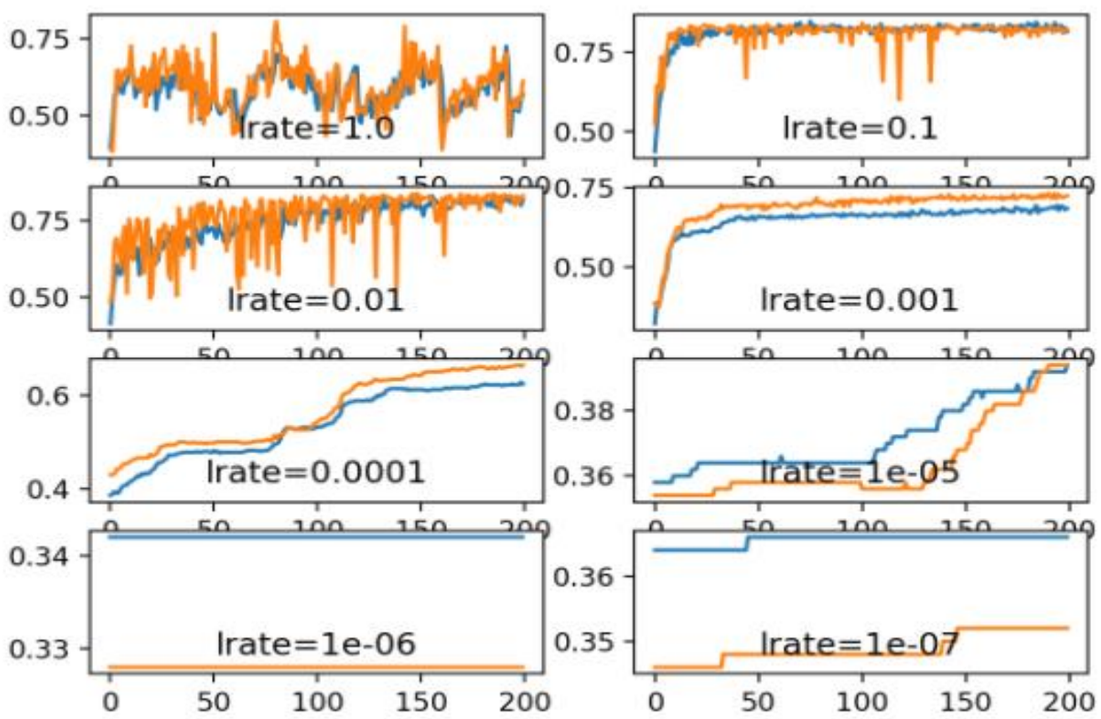


Figure 17

11.3 Model performance evaluation (After hyperparameter tuning)

We have applied the best learning rate 0.001 (with reference to figure 17) on our Reset topology model and captured the model performance visualization graph.

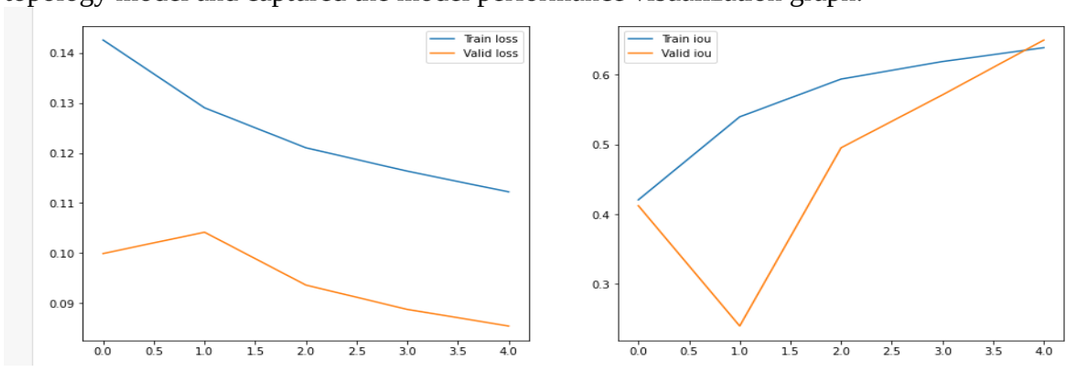
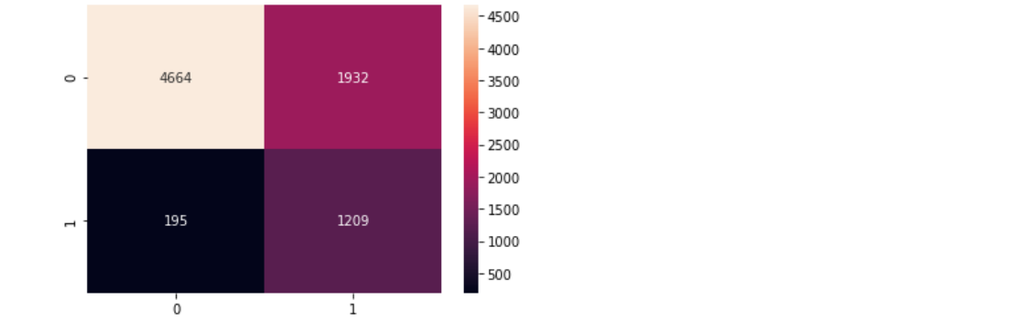


Figure: 18



Below is the classification report based on the actual and predicted target values.

	precision	recall	f1-score	support
0	0.96	0.71	0.81	6596
1	0.38	0.86	0.53	1404
accuracy			0.73	8000
macro avg	0.67	0.78	0.67	8000
weighted avg	0.86	0.73	0.76	8000

AIML/.../Final classification and detection.ipynb

Figure 19

12. Project Implications

Our project goal is to learn the computer vision technology for deep learning solution. We have applied the solution with the best of our knowledge capacity in this domain. We, as a remote team (group of 5 members), all are new to this domain and technology but enthusiastic, energized have tried the best within the limited infrastructure in this pandemic situation.

1. This solution can be used to identify the pneumonia detection in the area of medical fields, however there are huge improvement scopes in this project to use the model in the real world.
2. Level of confidence is moderate with the proposed solution but we are very confident along with better infrastructure and little more dedicated effort, we can definitely bring this model with high level of confidence.

13. Limitations

As we have taken up this project as toy (learning) project, it definitely has limitations if we think to apply for the real world…

1. The biggest limitation we felt throughout the project is the not available of adequate infrastructure which often we mitigate by adjusting the sample sets. But doing so, predominantly has major impact on model performance on real world. So, our project has too.
2. Our model is well performed on lung opacity area detection, however fall short on lung opacity classification. So, we need to further work on it to improve the same.