**Pneumonia Detection using Convolutional Neural Networks**

Table of Contents

[Problem Statement 2](#_Toc41837908)

[Methodology 3](#_Toc41837909)

[PHASE 1: Data Understanding & Preparation 3](#_Toc41837910)

[Dataset summary 4](#_Toc41837911)

[PHASE 2: Exploratory Data Analysis 6](#_Toc41837912)

[Class Distribution 6](#_Toc41837913)

[Univariate Analysis 7](#_Toc41837914)

[Bivariate Analysis 8](#_Toc41837915)

[Feature Engineering 10](#_Toc41837916)

[Missing Dataset 12](#_Toc41837917)

[PHASE 3: Model Selection and Building 13](#_Toc41837918)

[Convolutional Neural Networks 13](#_Toc41837919)

[Model Selection 13](#_Toc41837920)

[PHASE 4: Model Performance and Comparison 14](#_Toc41837921)

[Model Development 15](#_Toc41837922)

[Stage 1 15](#_Toc41837923)

[Model Performance Metrics 16](#_Toc41837924)

[Stage 2 17](#_Toc41837925)

[Model Performance Metrics 19](#_Toc41837926)

[Conclusion 19](#_Toc41837927)

[Deployment Strategy 20](#_Toc41837928)

[Infrastructure 21](#_Toc41837929)

[Limitations 23](#_Toc41837930)

[References 25](#_Toc41837931)

[Appendix 26](#_Toc41837932)

# Problem Statement

Pneumonia is an illness that disturbs the lung air sacs of an infected person. It is triggered by bacteria, fungi, or a virus that infects the air sacs of lungs that fill up with discharge fluids that leads to chills, fever, coughing with mucus, and breathing trouble among persons diagnosed with this disease. Children below five years of age and elderly patients with weak immune system are vulnerable to this type of diseases. Pneumonia has high mortality rate in children and remains a life-threatening disease now a days if not detected or diagnosed earlier. Radiography (CXR), CT-scan, or MRI is the common method to discover pneumonia. Medical personnel check the patient’s radiograph of their chest to determine if the patient is suffering with Pneumonia. In addition, the usual method for finding pneumonia is through medical history and laboratory results of the patient.

Radiograph of chest is penetrated through X-rays where the soft tissues produces a dark color and hard tissues like bones produces a bright color. Patients diagnosed with pneumonia shows the chest cavity signs of fluids filling the air sacs of lungs as for the radiograph picture appears brighter. Several abnormalities may be seen on lung cavities as brighter color may represent other abnormalities such as cancer cells, blood vessels swelling, and heart ailments. To validate the range and spot of an infected area of the lungs, chest x-rays is the utmost method.

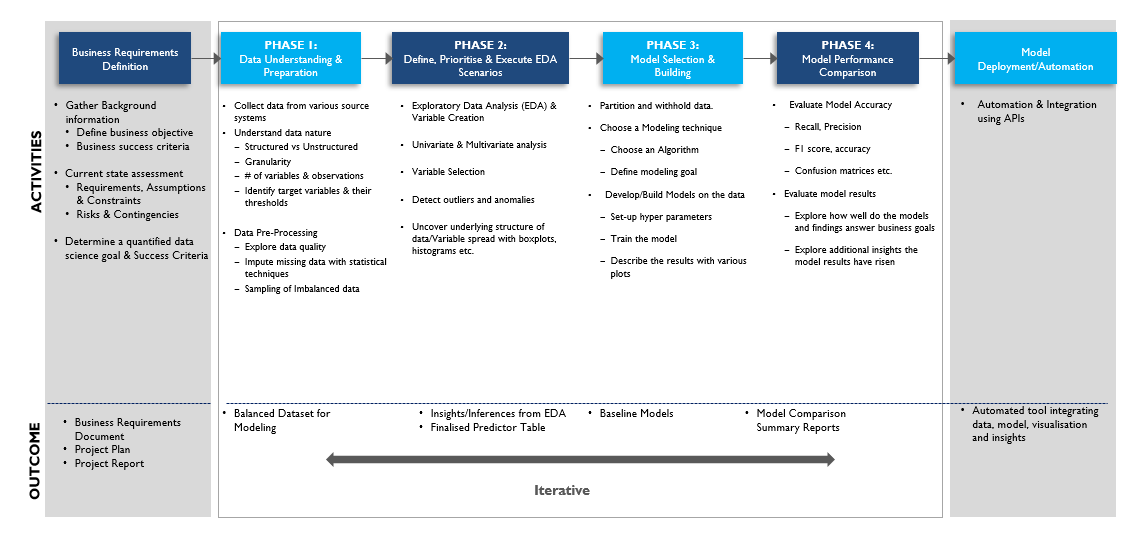
There are certain challenges faced by clinical experts during examination of Chest Radiograph (CXR). The patient could be suffering with other conditions such as fluid overload, bleeding, volume loss which makes diagnosis of pneumonia on CXR complicated. The positioning of the patient and depth of inspiration is important and can alter the appearance of the CXR. One of major challenge is lack of clinical experts to examine high volume of CXR images across hospital & lab facilities.

Powered by the advancement in Artificial Intelligence deep learning techniques the technological improvement today has reached new step closer in human intelligence. The deep learning has gained the ability in simulating the function of the human brain. It can recommend the solution to solve real-life problems. The deep learning by means of the convolutional neural networks has ability in obtaining significant characteristics in image classification tasks and provide promising results in medical image analysis. CNN architecture is capable in assisting the identification of some features from an image and use this feature to generate probabilities in classifying specific input.

The contribution of this study is to experiment with various deep learning models of CNN that can detect and classify pneumonia diseases efficiently thereby assisting clinical experts to make better decisions by unlocking relevant information hidden in the massive amount of data.

# Methodology

The proposed CRISP-DM (cross-industry process for data mining) methodology presents the structured approach to planning a data mining project. It is a robust and well-proven methodology and widely used in analytics model.



CRISP-DM methodology approach is applied in this specific study. The team followed the above phase wise approach to arrive at final findings. Within a phase, multiple tasks were carried out in parallel.

## PHASE 1: Data Understanding & Preparation

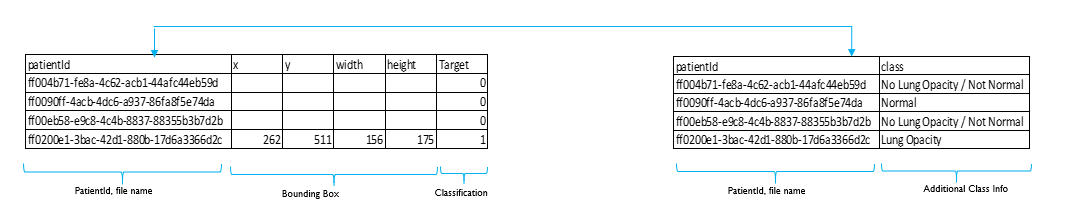
Team utilized the Radiological Society of North America (RSNA) dataset through the Kaggle RSNA Pneumonia Detection Challenge which contains 26,684 image data. The data set obtained follows DICOM ( Digital Imaging and Communications in Medicine) standard and it is grouped into two classes, pneumonia infected and normal with the dimensions of 1024 x 1024 pixels at maximum.

### Dataset summary

The below table summaries the data elements used in this project:

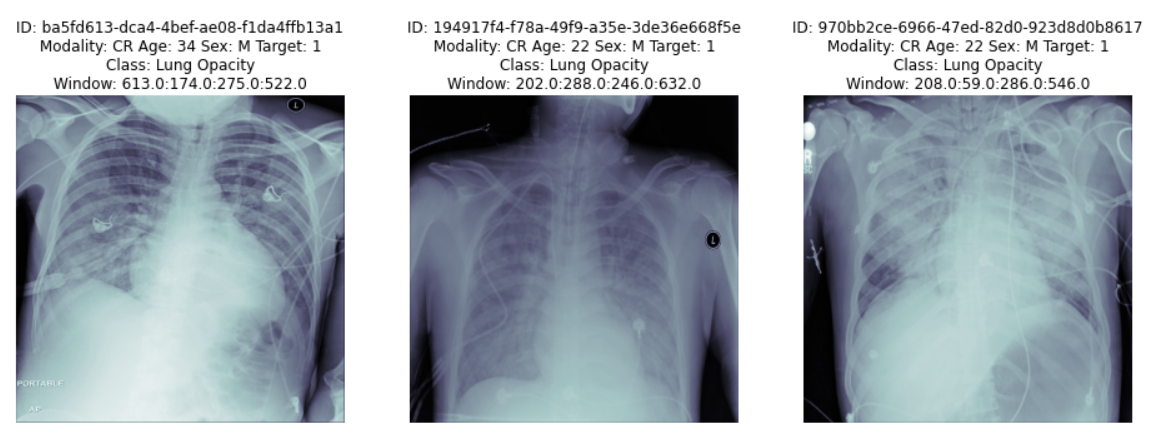
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **File** | **Record Count** | **Column Count** | **Understanding** |
| 1 | stage\_2\_train\_labels.csv | 30,227 | 6 | * Details providing classification of pneumonia and non-pneumonia images * PatientId (corresponding to a CXR DICOM image file name) * Bounding box coordinates providing affected area details * Variable to differentiate between pneumonia and non-pneumonia images |
| 2 | stage\_2\_detailed\_class\_info.csv | 30,227 | 2 | * Details providing classification of images into 3 different types of classes * There is a possibility that pneumonia was not present, nonetheless some type of abnormality on the was identified in the image |
| 3 | stage\_2\_train\_images | 26,684 |  | * Folder containing CXR DICOM images for model training * DICOM images contains tag (meta data) |
| 4 | stage\_2\_test\_images | 3,000 |  | * Folder containing CXR DICOM images for final model testing & submission |

The two files 1) Stage\_2\_train\_lables.csv 2) Stage\_2\_detailed\_calss\_info.csv are combined during EDA and model development using patientid as the key to between the two files. The DICOM images follow a naming convention where the file name represents patientid and thereby can be related to annotations and labels in csv files.



The images below depict the CXR images of patients where lung opacity is observed and are diagnosed with Pneumonia:



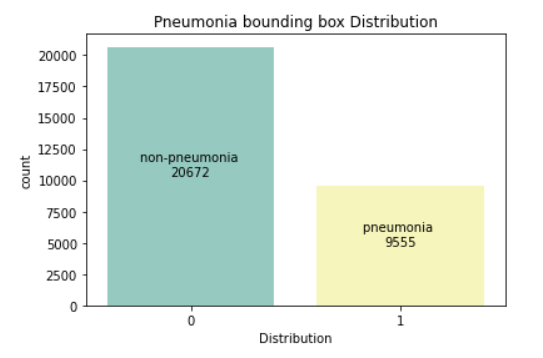


## PHASE 2: Exploratory Data Analysis

Team carried out exploratory analysis to get a better understanding of dataset. The objective of this phase is to class distribution, identify missing dataset and perform univariate and bi-variate analysis.

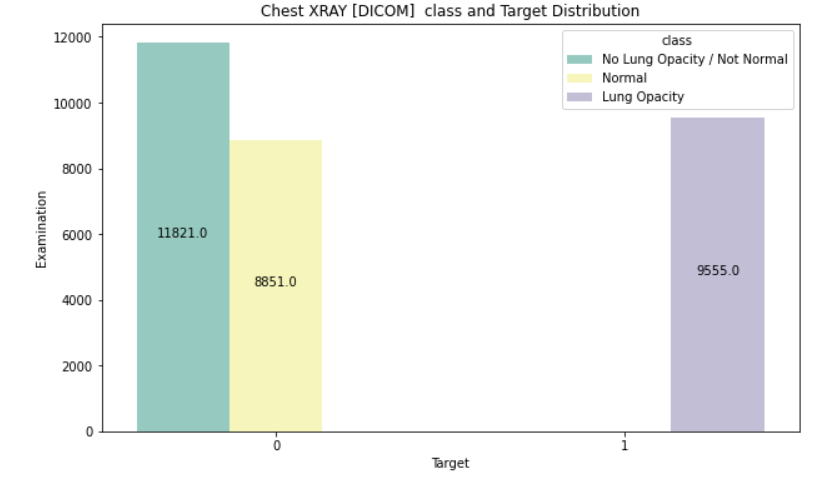
### Class Distribution

The total number rows in stage\_2\_train\_labels (30,227) is 11% higher than the total number of images (26,684). The dataset of 30,227 rows have 26,684 unique patientId’s which matches with the count of images. Further analysis indicates there are no missing images and each image is correctly represented by patientId column in labels csv file.



The entire dataset is represented by two classes:  
1) Target Class = 1 represents Pneumonia cases  
2) Target Class = 0 represents Non-Pneumonia cases

The overall distribution of classes is 31.61% (pneumonia) and 63.38% (non-pneumonia). A higher number of images with pneumonia cases may help the model to extract relevant features which can further improve model to classify the cases.



The non-pneumonia class comprises of two sub-classes 1) Normal 2) No Lung Opacity / Not Normal. For the purpose of model training both the above sub-classes are considered as part of training dataset. There is a further scope to perform a multiple class classification with 3 types of classes which is not accomplished as part of this study

### Univariate Analysis

Further visual analysis is carried out to validate bounding box annotations.

|  |  |
| --- | --- |
| Scatter plot of x,y coordinates | Scatter plot of x,y coordinates |
|  |  |

|  |  |
| --- | --- |
|  |  |

The X,Y coordinates plot below indicates that most of the annotations are correctly positioned (overlay) on top of the lung region with <1% outliers. The images with outlier annotations are not removed from training dataset.

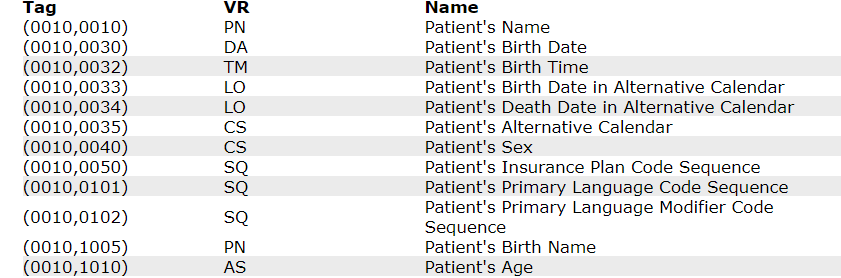
|  |
| --- |
| Bounding Box Distribution |
|  |

A plot of annotations indicates that bounding boxes (BBoxes) lie within the image range (1024 \* 1024) and are correctly positioned (overlay) on top of the lung region in the image.

### Bivariate Analysis

In order to analyze additional features about CXR image, DICOM reader library is used to extract tags from image. A DICOM data object consists of several attributes, including items such as name, ID, etc., and one special attribute containing the image pixel data.

The image below shows a sample of DICOM tags are included as part of the DICOM data object



There are many attributes which requires medical domain knowledge to comprehend and hence not considered as part of this experiment.  
In this study the attributes extracted from DICOM data object are Patient’s Age and Patient’s Sex along with view position.

A count plot by Age indicates that a higher percentage of patients (count wise) in the age group of 45-65. The mean average age of patient is around 46

|  |
| --- |
| Age Distribution of Patients diagnosed with Pneumonia |
|  |

.

The number of male patients is approx. 50% higher than female indicating that males are more prone to pneumonia disease. Also, the distribution of male/female seems consistent across all age groups.

|  |
| --- |
| Distribution by Age, Sex of Patients diagnosed with Pneumonia |
|  |

Based on the above observation, the following Null hypothesis has been constructed:

# Hypothesis test to check if Gender has an effect on Target class (Pneumonia, Non-Pneumonia)

Ho = "Gender has no effect on Predicted Class" # Stating the Null Hypothesis

Ha = "Gender has an effect on Predicted Class" # Stating the Alternate Hypothesis

crosstab = pd.crosstab(train\_class\_df['PatientSex'],train\_class\_df['class'])

chi, p\_value, dof, expected = stats.chi2\_contingency(crosstab)

if p\_value < 0.05: # Setting our significance level at 5%

print(f'{Ha} as the p\_value ({p\_value.round(3)}) < 0.05')

else:

print(f'{Ho} as the p\_value ({p\_value.round(3)}) > 0.05')

crosstab

The observation is that

**Gender has no effect on Predicted Class as the p\_value (0.57) > 0.05**

**Hence, Null Hypothesis is Accepted**

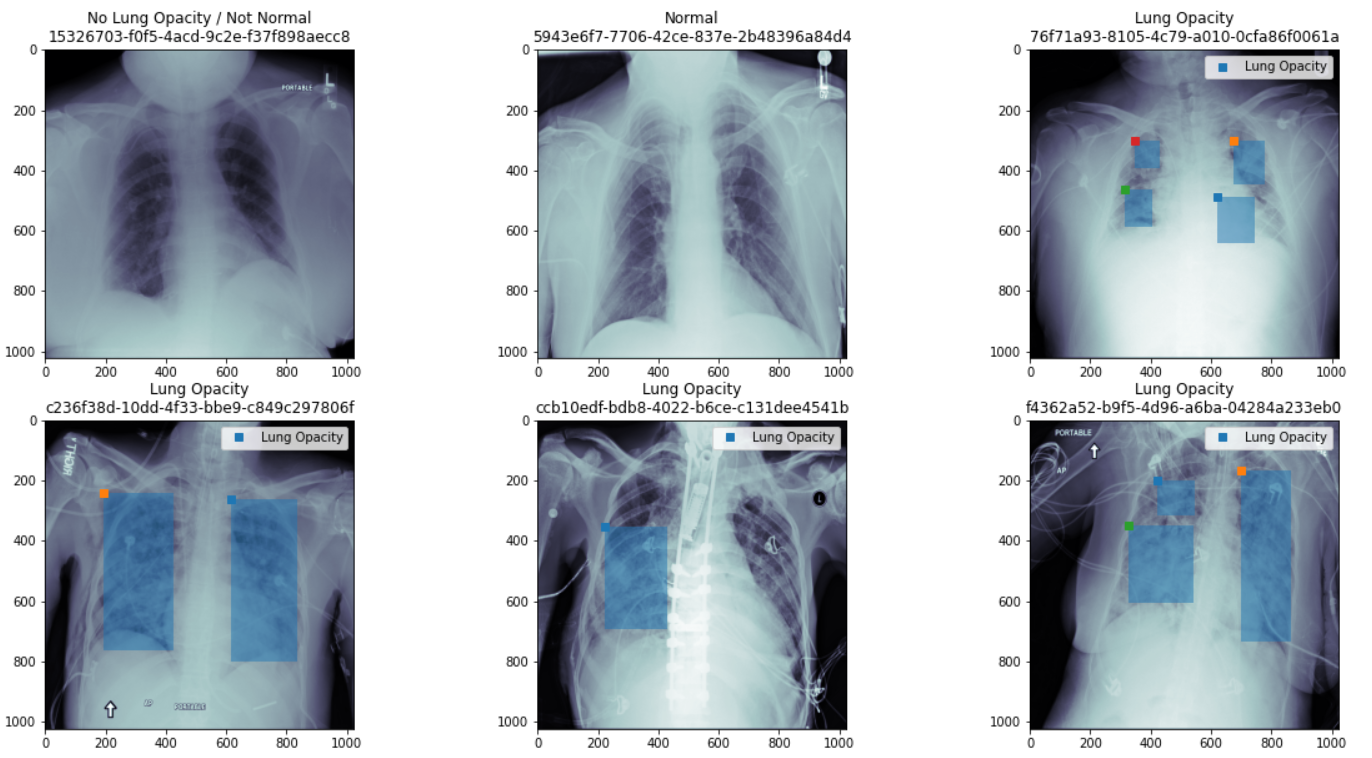
### Feature Engineering

As next step few additional features are derived based on the labels data. Since there is a difference in the count of images and labels, it indicates that certain images (patients) may have more than 1 BBoxes annotations (patches of pneumonia). There are approx. 13% of patient’s with more than 1 patch (evidence) of pneumonia.

|  |
| --- |
| Number of Patients’ with Pneumonia evidences (patches) |
|  |

The DICOM images below are few examples of various conditions in different patients. Traversing in a specific order from top to bottom and left to right:

1. No Lung Opacity / Not Normal : Non-Pneumonia
2. Normal : Non-Pneumonia
3. Lung Opacity : Patient suffering with Pneumonia condition and having 4 patches of lung opacity
4. Lung Opacity : Patient suffering with Pneumonia condition and having 2 patches of lung opacity
5. Lung Opacity : Patient suffering with Pneumonia condition and having 1 patch of lung opacity
6. Lung Opacity : Patient suffering with Pneumonia condition and having 3 patches of lung opacity



In order to arrive at size of each individual pneumonia patch (evidence), a new Bounding Box area feature is derived. The Bounding Box area distribution indicates that few annotations might be incorrect and should be treated as outliers (max 371184 pixel). However, for modeling exercise the team has considered all images.

|  |  |
| --- | --- |
|  |  |

### Missing Dataset

The labels dataset is analyzed to ensure there is are cases of missing data. All rows have patientId and Target classes defined.

|  |
| --- |
| Rows with Null Values for Patient and Class Attributes: |
|  |

A large number of rows contain null values for attributes such as width, height, x and y but further investigation reveals that these null values appear only when Target class = 0 which is a valid scenario.

|  |  |
| --- | --- |
| Rows with Null Values | Rows with Null Values where Target =1 |
|  |  |

Based on the above analysis, there is no evidence of missing data and the entire dataset is considered for model development.

## PHASE 3: Model Selection and Building

Proposal is to build  Convolutional Neural Network (ConvNet/CNN), a Deep Learning algorithm which can take in an input image, assign importance (learnable weights) to various aspects/objects in the image and be able to differentiate one from the other. The model building consists of experimental analysis of each model towards the detection and classification of pneumonia disease.

### Convolutional Neural Networks

Process of CNN is to detect and categorize images from learned features. CNN image classification takes an input image, processes it and classifies it under certain categories. In this study there are two classes 1) Pneumonia 2) Non-Pneumonia.

During model training and validation phase, each input image will pass through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply classification (softmax) function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

|  |
| --- |
|  |

### Model Selection

The approach is to experiment with several CNN architectures and then arrive at the most optimal model which can be used for classification purpose. A subset of well know CNN architectures listed below are used for model training and inference.

|  |
| --- |
|  |

The CNN architectures used for this study are:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Model | Top-1 Accuracy on ImageNet Validation dataset | Comments |
| 1 | MobileNetV2 | 74.7% |  |
| 2 | VGG19 | 74.5% |  |
| 3 | InceptionV3 | 78.8% |  |
| 4 | ResNet50 | 81.2% |  |
| 5 | Mask R-CNN |  | Uses ResNet50 / ResNet101 as backbone |

Reference : <https://paperswithcode.com/sota/image-classification-on-imagenet>

For each of the above model, a minimum of two operations are performed:

1. Transfer learning
2. Model Training

## PHASE 4: Model Performance and Comparison

The objective of working with multiple models is to make use of constructive feedback principle to arrive the best model. Build a model, get feedback from metrics, make improvements and continue until desirable accuracy is achieved. Evaluation metrics explain the performance of a model. An important aspect of evaluation metrics is their capability to discriminate among model results.

In this study the team has narrowed down on following two important metrics with specific weights assigned to each of the metric:

1. Reliability : F1 Score since False Negatives are critical, and the objective is not to incorrectly classify a pneumonia patient as Normal
2. Space Complexity : model size on disk (normalized between 0 -1)

A third metric, Inference Time (Prediction time) is an important metric and should be considered in final model evaluation.

# Model Development

The 26,684 images are split in 80:20 ratio and considered as training dataset and validation dataset. The training dataset provides the primary input for feature extraction and having extensive training data can produce strong features for good result. The validation dataset draws results to validate the effectivity of the model to work in actual circumstances. A fixed subset of 3000 images out of 26,684 images is considered as test dataset to arrive at the model reliability.

The input images are resized to 224\*224 dimensions for MobileNetV2, VGG19, InceptionV3 and ResNet50 models. The image depth is 3 for all models.

A stage wise approach is taken to first evaluate all CNN models using pre trained weights followed by training for bottom layers. The objective is to understand if there is an improvement in model performance post training and can these models be used based on their performance metrics.

In the second stage the intent is to evaluate more complex frameworks which deal with object instance segmentation since the end goal is to classify the images and predict the bounding boxes.

## Stage 1

The following CNN models are considered for training and evaluation based on performance metrics. The model approach is two class classification.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Model | Weights | Model Training | Hyper Parameter |
| 1 | MobileNetV2 | Imagenet | Classifier head (top layer) | Alpha = 1.4 |
| 2 | MobileNetV2 | Imagenet | Unfreeze bottom 3 layers | Alpha = 1.4 |
| 3 | VGG19 | Imagenet | Classifier head (top layer) |  |
| 4 | VGG19 | Imagenet | Unfreeze bottom 2 layers |  |
| 5 | InceptionV3 | Imagenet | Classifier head (top layer) |  |
| 6 | InceptionV3 | Imagenet | Unfreeze bottom 5 layers |  |
| 7 | ResNet50 | Imagenet | Classifier head (top layer) |  |
| 8 | ResNet50 | Imagenet | Unfreeze bottom 5 layers |  |

### Model Performance Metrics

The below metrics are achieved in a single iteration keeping batch size as 50 passing iterating over 5 Epochs.

The model performance metrics achieved are as below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Model | Transfer Learning (ImageNet weights) with classification layer | Model (layer) Training | Model Size on disk (MB) | Model Training duration (min) | Precision | Recall | F1 Score | Model Evaluation Metric |
| 1 | MobileNetV2 | 1) GolablAveragePooling  2) Dense, 2, softmax | None | 17.7 | 213 | 0.49 | 0.30 | 0.37 | 0.49 |
| 2 | MobileNetV2 | Bottom 3 layers | 17.7 | 294 | 0.69 | 0.30 | 0.42 | 0.53 |
| 3 | VGG19 | 1) Flatten  2) Dense , 4096, relu  3) Dense, 409, relu  4) Dense, 2, sigmoid | None | 558 | 231 | 0.22 | 0.64 | 0.32 | 0.32 |
| 4 | VGG19 | Bottom 2 layers | 558 | 269 | 0.55 | 0.89 | 0.68 | 0.61 |
| 5 | ResNet50 | 1) GolablAveragePooling  2) Dense, 2, sigmoid | None | 95 | 186 | 0.82 | 0.30 | 0.44 | 0.53 |
| 6 | ResNet50 | Bottom 5 layers | 95 | 185 | 0.61 | 0.49 | 0.54 | 0.61 |
| 7 | InceptionV3 | 1) GolablAveragePooling  2) Dense, 2, sigmoid | None | 85 | 135 | 0.63 | 0.24 | 0.35 | 0.46 |
| 8 | InceptionV3 | Bottom 5 layers | 88 | 105 | 0,60 | 0.29 | 0.39 | 0.49 |

Note\*: The threshold for Class =1 is > 0.4 for all of the above models  
The model training time is taken only for one iteration

Model Evaluation Metric is arrived taking into consideration two factors:

1. Model Size on Disk (the values are normalized between 0 -1) with 20% weightage
2. F1 Score with 80% weightage

Model Evaluation Metric = (20% \* Model Size on Disk) + (80% \* F1 Score)

As per the evaluation VGG19 (model 4) and ResNet50 (model 6) are the top 2 classification models. The models used in Stage 1 are mainly classifier models however the end objective is also to predict bounding boxes. R-CNN based frameworks are more suited for Region of Interest (RoI) use cases.Based on the above inference the decision is to move ahead with Mask R-CNN instance segmentation framework as it uses ResNet model as the backbone.

## Stage 2

Mask R-CNN framework is an extension of Faster R-CNN. Faster R-CNN is widely used for object detection tasks. For a given image, it returns the class label and bounding box coordinates for each object in the image. Mask R-CNN, in addition to the class label and bounding box coordinates for each object, will also return the object mask.

|  |
| --- |
| Mask R-CNN Framework |
|  |

Mask R-CNN framework works in 4 different steps:

1. Backbone Model

The first step is to take an image and extract features using the ResNet architecture. These features act as an input for the next layer.

1. Region Proposal Network (RPN)

RPN network is applied on the feature maps obtained in the previous step. This basically predicts if an object is present in that region (or not). In this step, the model predicts the regions or feature maps which contain some object.

1. Region of Interest (RoI)

The regions obtained from the RPN might be of different shapes, hence, framework applies a pooling layer and converts all the regions to the same shape. Next, these regions are passed through a fully connected network so that the class label and bounding boxes are predicted.

1. Segmentation Mask

For all the predicted regions, framework computes the Intersection over Union (IoU) with the ground truth boxes.

IoU = Area of the intersection / Area of the union

only if the IoU is greater than or equal to 0.5, framework considers it as a region of interest. Otherwise, it neglects that region. This step is performed for all the regions and only a set of regions for where the IoU is greater than 0.5 is selected.

Mask R-CNN framework is evaluated for following scenarios:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Backbone | Weights | Model Training |
| 1 | ResNet50 | Imagenet | Trainable Layers : All |
| 2 | ResNet50 | Imagenet | Trainable Layers : heads |
| 3 | ResNet101 | mask\_rcnn\_coco | Trainable Layers : All |
| 4 | ResNet101 | mask\_rcnn\_coco | Trainable Layers : heads |

### Model Performance Metrics

The below performance metrics are achieved in a single iteration with 5 Epochs.

The model performance metrics achieved are as below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Model | Transfer Learning | Model (layer) Training | Model Size  on disk (MB) | Model Training duration (min) | Precision | Recall | F1 Score | Model Evaluation Metric |
| 9 | Mask R-CNN | Resnet50 backbone, Image weights | All layers | 115 | 58 | 0.17 | 0.68 | 0.27 | 0.35 |
| 10 | Mask R-CNN | ResNet50 backbone, Imagenet weights | heads | 115 | 100 | 0.20 | 0.89 | 0.33 | 0.40 |
| 11 | Mask R-CNN | Resnet101 backbone, coco weights | All layers | 256 | 101 | 0.21 | 1 | 0.35 | 0.35 |
| 12 | Mask R-CNN | Resnet101 backbone, coco weights | heads | 256 | 105 | 0.22 | 0.11 | 0.15 | 0.19 |

Note\*: The threshold for Class =1 is > 0.98 for all of the above models  
The model training time is taken only for one iteration

Model Evaluation Metric is arrived taking into consideration two factors;

1. Model Size on Disk (the values are normalized between 0 -1) with 20% weightage
2. F1 Score with 80% weightage

Model Evaluation Metric = (20% \* Model Size on Disk) + (80% \* F1 Score)

## Conclusion

**As per the evaluation, Mask R-CNN (model 10) qualifies as the best the classification model. This model is used as the prediction engine for generating the final project submission file ‘project\_submission.csv’.**

|  |  |
| --- | --- |
| Predictions on Test images | |
|  |  |

# Deployment Strategy

The proposal is to make use of ensemble technique towards model deployment. Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model.

The approach involves narrowing down on at least three (odd number of models) top performing models based on model performance metrics and use majority voting mechanism for final prediction. Every model makes a prediction (votes) for each test instance and the final output prediction is the one that receives more than half of the votes.

The top three models based on experiments are:

1) Model 10 : Mask R-CNN

2) Model 4: VGG19

3) Model 6: ResNet50

|  |
| --- |
| Deployment Strategy : Voting Mechanism |
|  |

Technical deployment of AI model plays an important part in the overall scheme of things. Various models can be explored based on ease of use, prediction accuracy, inference latency etc. The following four options can be explored based:

* Deployment on devices such as mobile phones for ease of use
* Medical devices such as CXR
* Edge devices such as IOT gateways within medical facility premises
* Remotely on Cloud

# Infrastructure

The team utilized three different environments during the model building phase. The first choice has been using local desktops for initial piece of development working with smaller set of training images. The code was then moved to Google Colab (free version) to speed up the training process. The image files were loaded onto google drive with Colab Jupyter notebooks accessing the training images using drive mapping functionality. Few challenges were faced while training multiple models with larger number of images. Observed network disconnect between Colab environment and Google Drive resulting in freezing/hanging of notebook. The disconnects observed during model training step would require the entire training process to be re-run. The training step was a bit inconsistent from processing time perspective as similar model / training data set would take different timeframe for completion. Notebook hanging during training of full dataset was observed. Quite possible that this could be limitations of free version.

The code and dataset were then moved to AWS (paid version). Training dataset was loaded to S3 and AWS SageMaker notebook instances were used for model coding & execution. The code was modified to access files from S3 bucket using boto library. The initial part of the development was completed using a small size instance (ml.t2.medium). Once the code was stable, a larger instance type (ml.c5.rxlarge) was used for model training. Team’s experience with SageMaker has been satisfactory. Few challenges observed were around boto library which is too slow to iterate over list objects in a bucket, notebook instances timing out every 12 hours. Probably these challenges can be overcome by using different mechanism.

|  |
| --- |
| AWS SageMaker Notebook Instances |
|  |

|  |
| --- |
| AWS SageMaker Notebook Environment |
|  |

The team managed to read up some comparative differences between the two environments:

|  |  |  |
| --- | --- | --- |
| Comparative Study | Google Collab/Datalab | AWS SageMaker |
| Notebook Deployment | Notebook instance is launched via the Google cloud shell from the Google Cloud Console interface | AmazonSageMaker provides a one-click notebook deployment interface from Console |
| Customized Algorithms | Does not contain any pre-customized ML algorithms | The SageMaker custom algorithms have a variety of supervised, unsupervised and deep learning algorithms |
| Model Deployment | There is no direct way to handle the code deployment into production servers. the model built on this platform is packed into a Python module and deployed on Google CloudML | There is a provision for direct deployment of the trained ML models. The deployment is done to elastic compute infrastructure with high availability. |
| Automated Hyper-Parameter Tuning | Does not provide automated hyper-parameter tuning. HyperTune feature helps in automatic optimization of the ML model for an improved accuracy/minimized error. It provides this feature for TensorFlow models | It provides an option for automated hyper-parameter tuning on the ML model during the training period |
| Managed Compute Infrastructure | Offers a fully managed compute infrastructure. However, the Datalab instance is not auto-elastic. | Amazon SageMaker runs on a fully managed elastic compute server. Amazon SageMaker fully takes care of health checks, and outline infrastructure maintenance tasks via the built-in “Amazon CloudWatch monitoring and logging” service. |

# Limitations

This project explores the deep learning function in detecting pneumonia through computer vision using five convolutional neural network models. Our study enables to identify among the various models the best model to detect pneumonia.

Quite a few aspects can be revisited to enhance the outcome of these models. Some of the key areas are:

Exploratory Analysis:

* Investigate DICOM tags to understand better correlation between different attributes. Explore possibility of building different models based on evidences for e.g. characteristics of pneumonia patches may be different in children compared to adults

Pre-processing :

* The DICOM images can cropped in such a way that only area of interest (lungs portion) is provided as an input to CNN network

Model Hyper parameter tuning:

* Experiments can be conducted with different input sizes (resolution)  
  Increase number of epoch’s (currently constraint due to hardware capacity, only CPU based machines used to train the model  
  Learning rate  
  Patience Level  
  Batch size  
  Experiment with training multiple model layers

Model Development:

* Explore multi class models since the dataset can be split into 3 different classes 1) Normal 2) Pneumonia 3) Other Diseases

For future studies, adaptation of other convolutional neural network architectures like yolo, SSD architectures for pneumonia detection can be implemented. These observations will help medical work force in their decision making for a real-time application of the use of more accurate model in detecting pneumonia and discover the potential of diagnosing pneumonia using deep learning models.

# References

The team went through a lot of reading material on the web. Some of the notable sites referenced during project execution are listed below:

<https://paperswithcode.com/sota/image-classification-on-imagenet>

<https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d>

<https://arxiv.org/pdf/1801.04381.pdf>

<https://arxiv.org/ftp/arxiv/papers/2002/2002.09334.pdf>

<https://www.analyticsvidhya.com/blog/2020/04/build-your-own-object-detection-model-using-tensorflow-api/>

<https://www.analyticsvidhya.com/blog/2019/07/computer-vision-implementing-mask-r-cnn-image-segmentation/>

<https://medium.com/@alittlepain833/simple-understanding-of-mask-rcnn-134b5b330e95>

<https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088>

<https://www.geeksforgeeks.org/python-programming-language/>

<https://machinelearningmastery.com/how-to-perform-object-detection-with-yolov3-in-keras/>

# Appendix

Github link : <https://github.com/ds007kumar/Capstone_project>

List of files submitted as part of the project:

|  |  |  |  |
| --- | --- | --- | --- |
| # | EDA / Model related | File Name | Description |
| 1 | Prediction on Test Images | Final\_Prediction\_Submission.scv | Submission file |
| 2 | Summary | Pneumonia\_Detection\_Project\_Report\_Final\_Submission.pdf | PDF version of the presentation |
| 3 | EDA | Capstone\_PD\_EDA | Exploratory Analysis |
| 4 | MobileNetV2 | Capstone\_PD\_Mobilenet\_TL\_Fit | MobileNetV2 transfer learning |
| 5 | MobileNetV2 | Capstone\_PD\_Mobilenet\_TL\_MT\_Fit | MobileNetV2 model training |
| 6 | VGG19 | Capstone\_PD\_VGG19\_TL\_FIT | VGG19 transfer learning |
| 7 | VGG19 | Capstone\_PD\_VGG19\_TL\_MT\_FIT | VGG19 model training |
| 8 | ResNet50 | Capstone\_PD\_ResNet50\_TL\_FIT | ResNet50 transfer learning |
| 9 | ResNet50 | Capstone\_PD\_ResNet50\_TL\_MT\_FIT | ResNet50 model training |
| 10 | InceptionV3 | Capstone\_PD\_InceptionV3\_TL\_FIT | InceptionV3 transfer learning |
| 11 | InceptionV3 | Capstone\_PD\_InceptionV3\_TL\_MT\_FIT.ipynb | InceptionV3 model training |
| 12 | Mask R-CNN | Capstone\_PD\_MaskRCNN\_Resnet50\_Train-heads | Mask R-CNN, resnet50 backbone, imagenet weights and training ‘heads’ |
| 13 | Mask R-CNN | Capstone\_PD\_MaskRCNN\_Resnet50\_Predict\_heads | Test predictions with Mask R-CNN, resnet50 backbone, imagenet weights and training ‘heads’ |
| 14 | Mask R-CNN | Capstone\_PD\_MaskRCNN\_Resnet50\_Train\_AllLayers | Mask R-CNN, resnet50 backbone, imagenet weights and training ‘all’ layers |
| 15 | Mask R-CNN | Capstone\_PD\_MaskRCNN\_Resnet50\_Predict\_AllLayers | Test predictions with Mask R-CNN, resnet50 backbone, imagenet weights and training ‘all’ layers |
| 16 | Mask R-CNN | Capstone\_PD\_MaskRCNN\_coco\_Train\_heads | Mask R-CNN, resnet101 backbone, coco weights and training ‘heads’ |
| 17 | Mask R-CNN | Capstone\_PD\_MaskRCNN\_Coco\_Predict\_heads | Test predictions with Mask R-CNN, resnet101 backbone, coco weights and training ‘heads’ |
| 18 | Mask R-CNN | Capstone\_PD\_MaskRCNN\_coco\_Train\_AllLayers | Mask R-CNN, resnet101 backbone, coco weights and training ‘all’ layers |
| 19 | Mask R-CNN | Capstone\_PD\_MaskRCNN\_Coco\_Predict\_AllLayers | Test predictions with Mask R-CNN, resnet101 backbone, coco weights and training ‘all’ layers |
| 20 | Mask R-CNN | Capstone\_PD\_MaskRCNN\_Resnet50\_Predict\_heads-FinalSubmission | Final submission file using Mask R-CNN, resnet50 backbone, imagenet weights and training ‘heads’ |