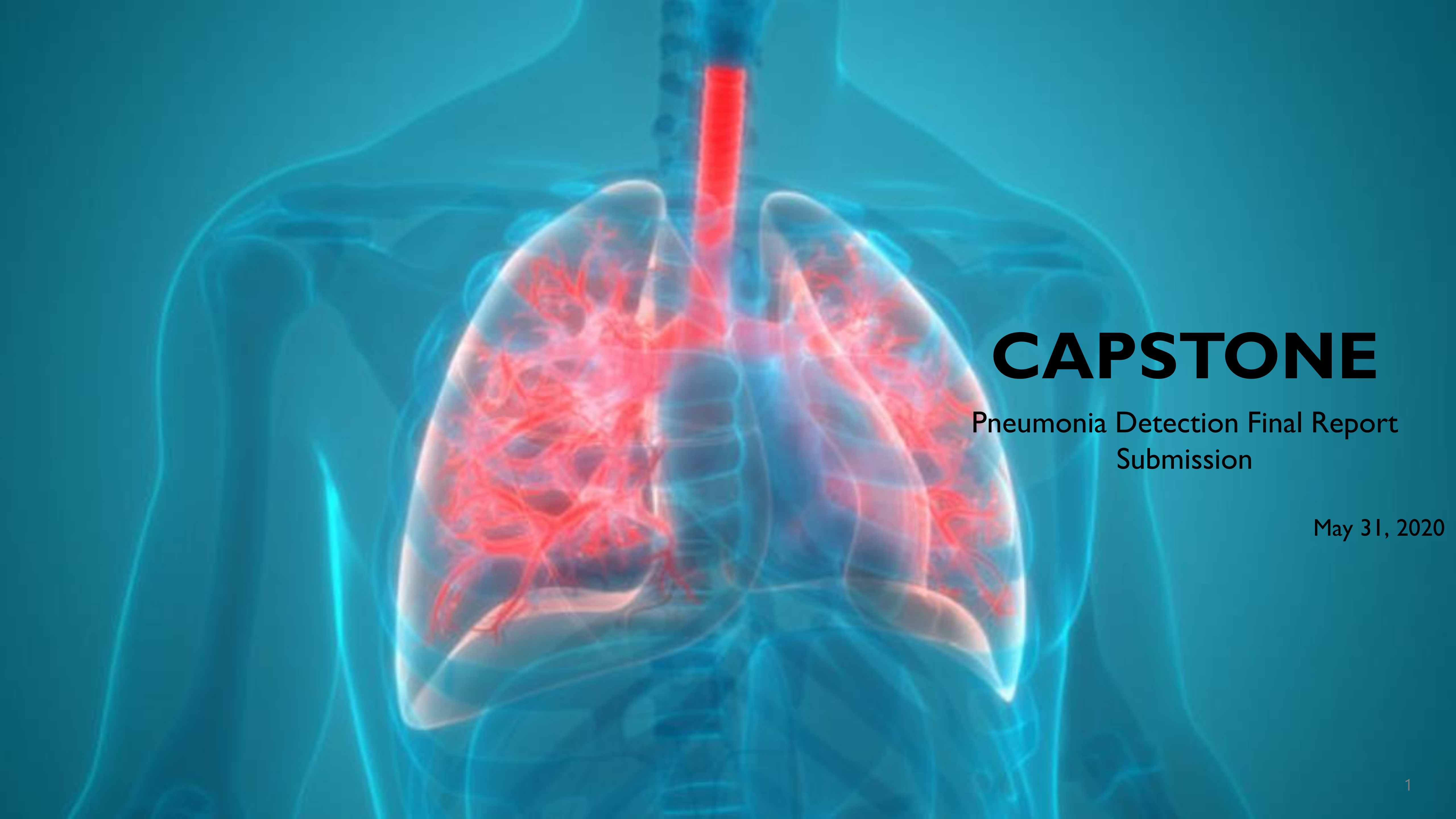


CAPSTONE

Pneumonia Detection Final Report Submission

May 31, 2020

Agenda

- PROBLEM STATEMENT & DATA UNDERSTADNING
- DEVELOPMENT METHODOLOGY
- EDA AND PREPROCESSING
- MODEL DEVELOPMENT
- MODEL PERFORMANCE METRIC
- CONCLUSION

PROBLEM STATEMENT

PROBLEM STATEMENT

DISEASE

Pneumonia is an inflammatory condition of the lung affecting primarily the small air sacs known as alveoli. Pneumonia is usually caused by infection with viruses or bacteria and less commonly by other microorganisms. Symptoms typically include some combination of productive or dry cough, chest pain, fever and difficulty breathing.

IMPACT

- Lower respiratory tract infection and pneumonia are two of the leading causes of death, accounting for more than 4 million fatalities annually
- Pneumonia accounts for over 15% of all deaths of children under 5 years old worldwide
- Older people have higher risk of getting pneumonia

DIAGNOSIS

- Based on symptoms and physical examination
- Chest Radiograph (CXR) by highly trained specialists
- Confirmation through clinical history

CHALLENGE

- Other conditions such as fluid overload, bleeding, volume loss make diagnosis of pneumonia on CXR complicated
- positioning of the patient and depth of inspiration can alter the appearance of the CXR
- Lack of clinical experts to examine high volume of CXR images

SOLUTION OBJECTIVE

- Save lives
- Powered by AI/ML techniques, build a pneumonia detection system to locate the position of inflammation in an image
- Assist physicians to make better clinical decisions or even replace human judgement in certain functional areas of healthcare (e.g., radiology)
- Guided by relevant clinical questions, powerful AI techniques can unlock clinically relevant information hidden in the massive amount of data, assist clinical decision making

INPUT DATASETS – OVERVIEW

#	File	Record Count	Column Count	Understanding
1	stage_2_train_labels.csv	30227	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	30227	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	26684		• Folder containing CXR dicom images for model training • Dicom images contains tag (meta data)
4	stage_2_test_images	3000		• Folder containing CXR dicom images for model testing

patientId	x	y	width	height	Target
ff004b71-fe8a-4c62-acb1-44afc44eb59d					0
ff0090ff-4acb-4dc6-a937-86fa8f5e74da					0
ff00eb58-e9c8-4c4b-8837-88355b3b7d2b					0
ff0200e1-3bac-42d1-880b-17d6a3366d2c	262	511	156	175	1

PatientId, file name

Bounding Box

Classification

0 : Non - Pneumonia
1 : Pneumonia

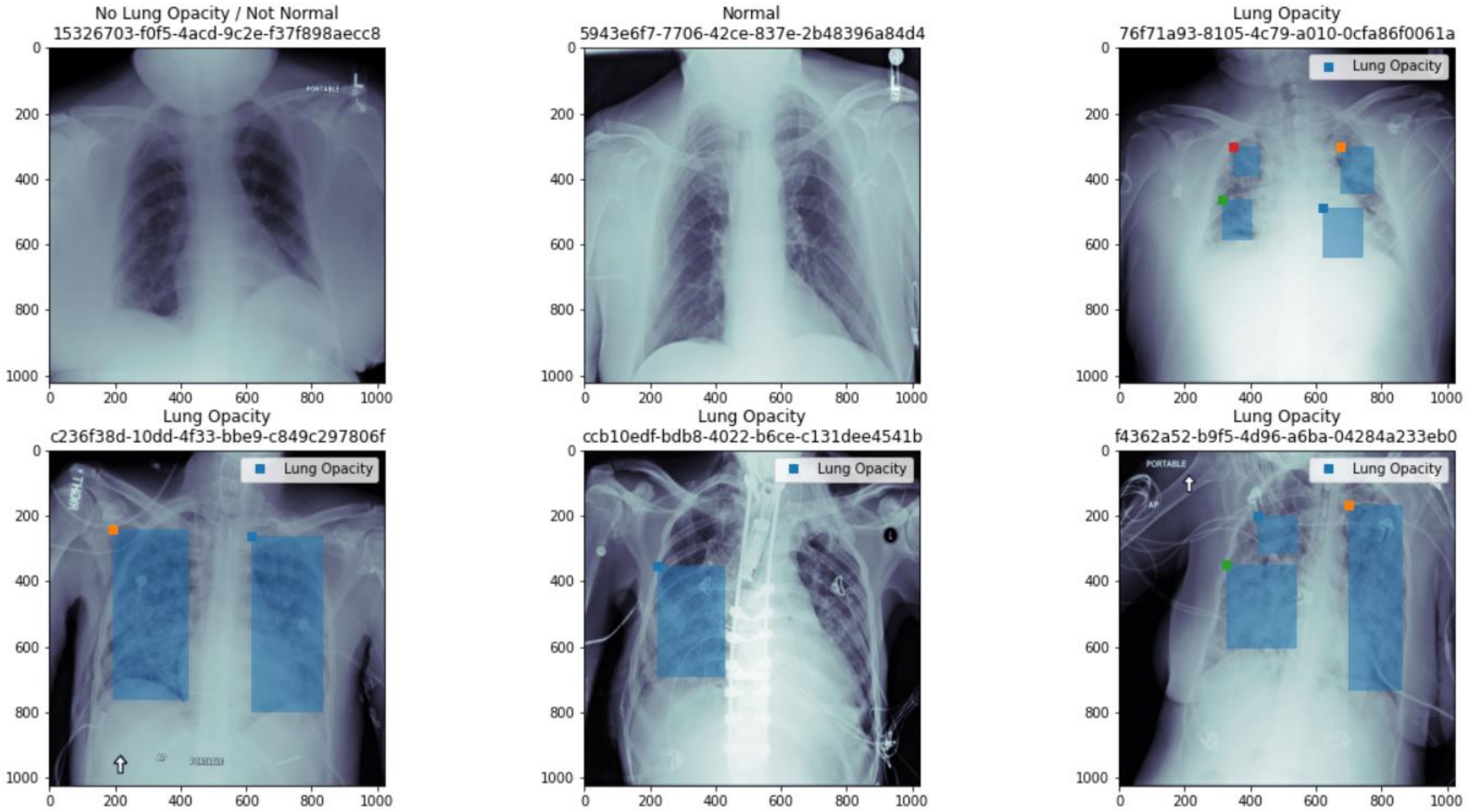
patientId	class
ff004b71-fe8a-4c62-acb1-44afc44eb59d	No Lung Opacity / Not Normal
ff0090ff-4acb-4dc6-a937-86fa8f5e74da	Normal
ff00eb58-e9c8-4c4b-8837-88355b3b7d2b	No Lung Opacity / Not Normal
ff0200e1-3bac-42d1-880b-17d6a3366d2c	Lung Opacity

PatientId, file name

Additional Class Info

Normal
Lung Opacity
No Lung Opacity / Not Normal

DICOM IMAGES – PREVIEW



DEVELOPMENT METHODOLOGY

DEVELOPMENT METHODLOGY (CRISP DM)

ACTIVITIES

Business Requirements Definition

- Gather Background information
  - Define business objective
  - Business success criteria
- Current state assessment
  - Requirements, Assumptions & Constraints
  - Risks & Contingencies
- Determine a quantified data science goal & Success Criteria

PHASE 1: Data Understanding & Preparation

- Collect data from various source systems
- Understand data nature
  - Structured vs Unstructured
  - Granularity
  - # of variables & observations
  - Identify target variables & their thresholds
- Data Pre-Processing
  - Explore data quality
  - Impute missing data with statistical techniques
  - Sampling of Imbalanced data

PHASE 2: Define, Prioritise & Execute EDA Scenarios

- Exploratory Data Analysis (EDA) & Variable Creation
- Univariate & Multivariate analysis
- Variable Selection
- Detect outliers and anomalies
- Uncover underlying structure of data/Variable spread with boxplots, histograms etc.

PHASE 3: Model Selection & Building

- Partition and withhold data.
- Choose a Modeling technique
  - Choose an Algorithm
  - Define modeling goal
- Develop/Build Models on the data
  - Set-up hyper parameters
  - Train the model
  - Describe the results with various plots

PHASE 4: Model Performance Comparison

- Evaluate Model Accuracy
  - Recall, Precision
  - F1 score, accuracy
  - Confusion matrices etc.
- Evaluate model results
  - Explore how well do the models and findings answer business goals
  - Explore additional insights the model results have risen

Model Deployment/Automation

- Automation & Integration using APIs

OUTCOME

- Business Requirements Document
- Project Plan
- Project Report

- Balanced Dataset for Modeling

- Insights/Inferences from EDA
- Finalised Predictor Table

- Baseline Models

- Model Comparison Summary Reports

- Automated tool integrating data, model, visualisation and insights

Iterative

IMPLEMENTATION APPROACH

Initiation

Kick-Off

Functional Requirement Brainstorming/Grooming

Deliverables Identification

Tools and Technologies Identification

Standup Cadence agreement

High Level Model Architecture Agreement

Infra Needs Identification

Execution

AI / ML

Initial Data Exploration → Removal of Clutter → Data Prep for EDA → EDA → ABT → ML Model Development

Cadence

Weekly calls, Team members handshakes, Individual Progress report, Dependencies/Integration points identification/agreements

Tools

TensorFlow

Keras

python

NumPy

pandas

jupyter

amazon web services

Amazon SageMaker

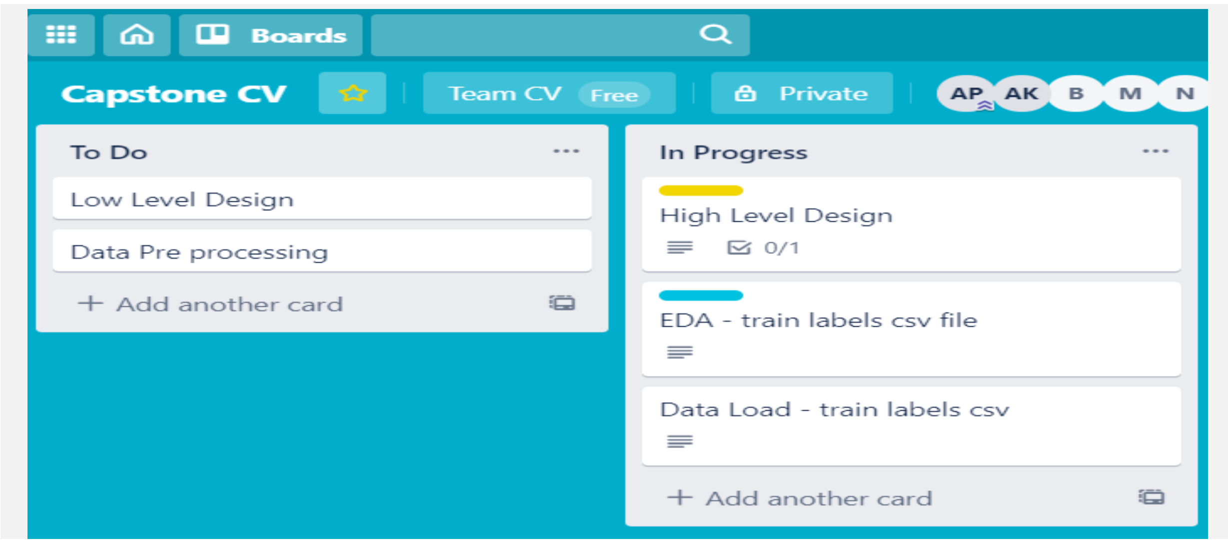
Amazon S3

Trello

GitHub

TEAM CADENCE

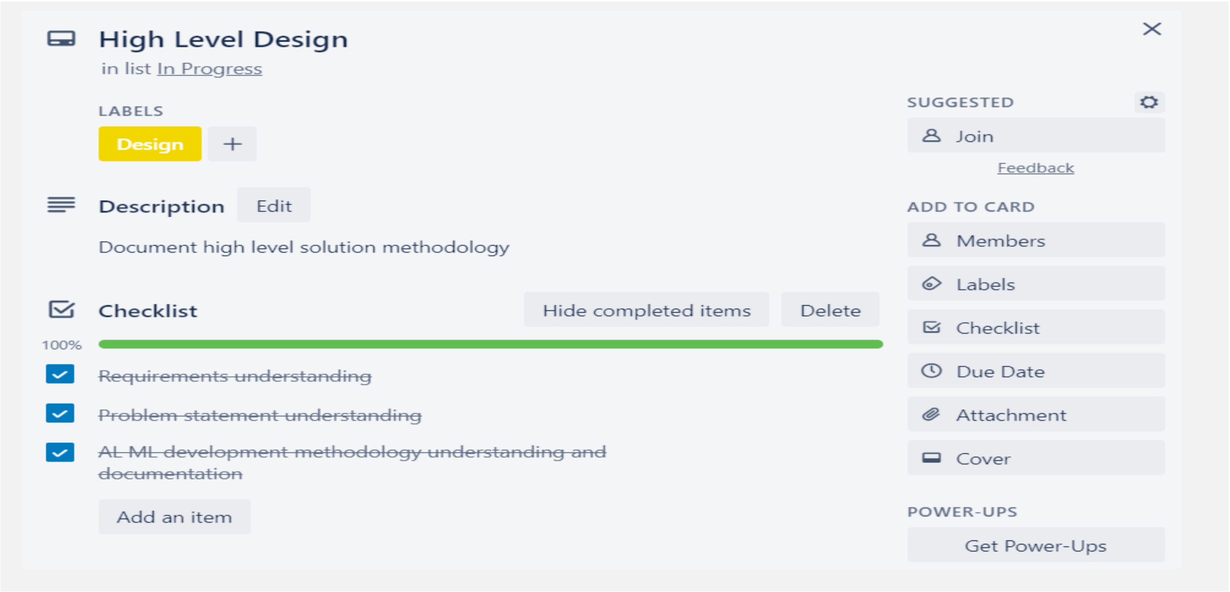
Kanban Backlog



- Team members whatsapp group for frequent coordination
- Group meetings, twice a week
- Trello Kanban board for task planning & tracking
- Github as code repository
- .

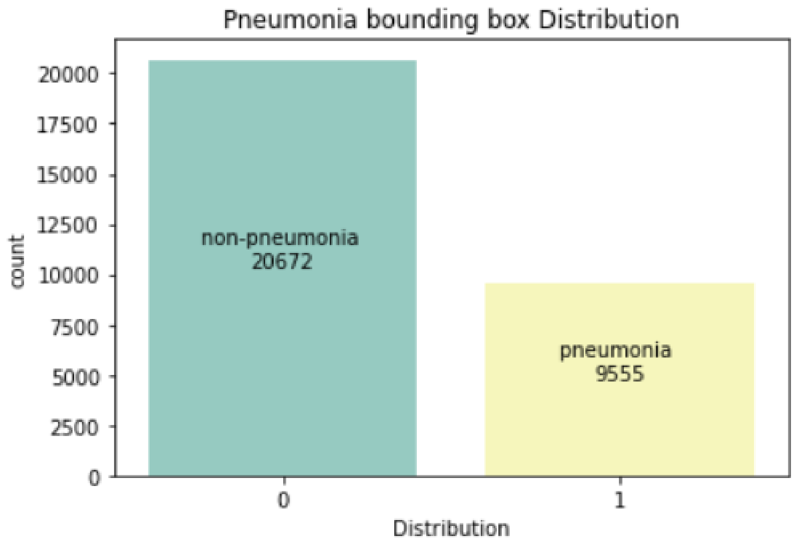
User Story

Story decomposition, Tasks identification, Allocation, Progress tracking..

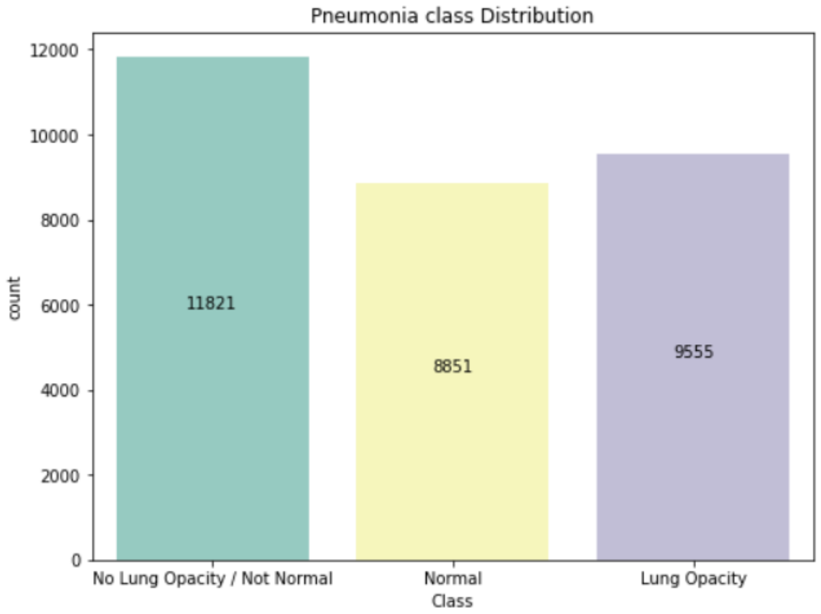


EXPLORATORY DATA ANALYSIS

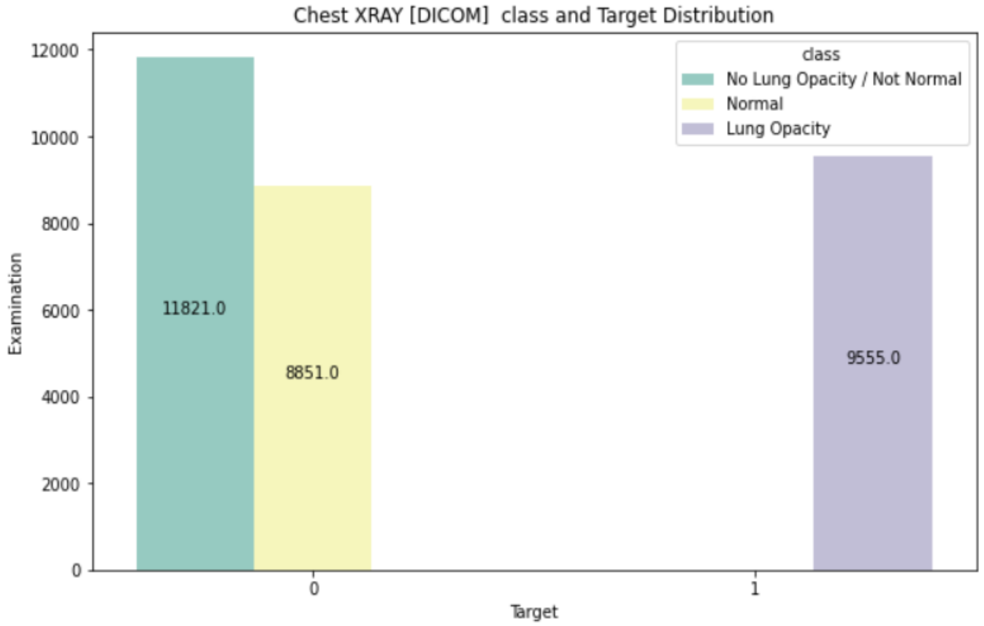
UNIVARIATE ANALYSIS



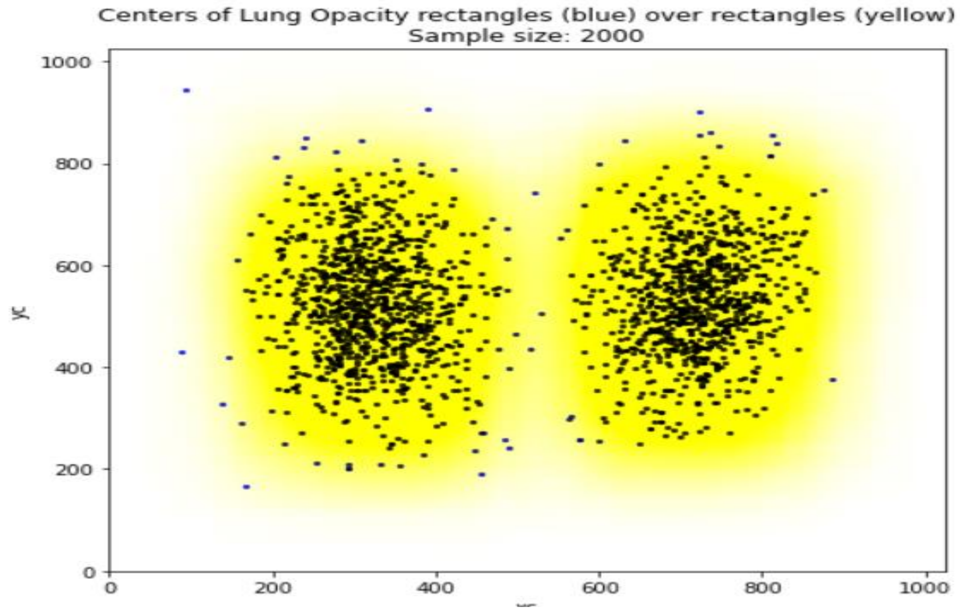
- Total number of labels is 30227, Number of unique patient Id's is 26684
- Total number of training images is 26884
- Patients have more than one bounding box evidences of Pneumonia
- Training labels comprise of 9555 (31.61 %) rows with Target =1 (pneumonia cases)
- There are no missing values for patientId & Target attributes
- Every record with Target =1 has bounding box information.



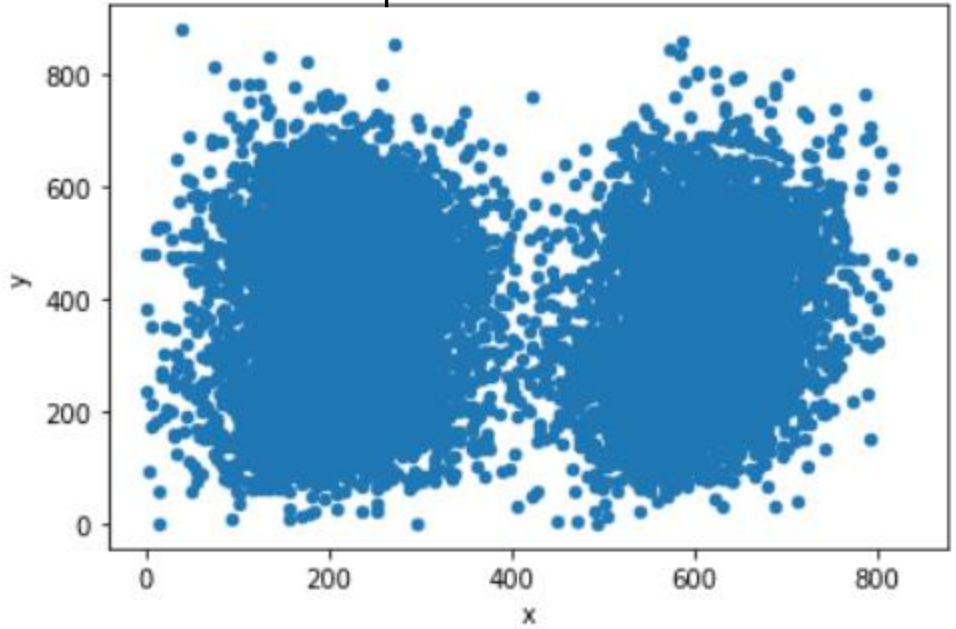
- There are 3 different classes in the dataset
- The distribution of class is consistent across pneumonia (Targ = 1) & non-pneumonia (Target = 0) cases. Target = 1 is identified by 'Lung Opacity' class



UNIVARIATE ANALYSIS



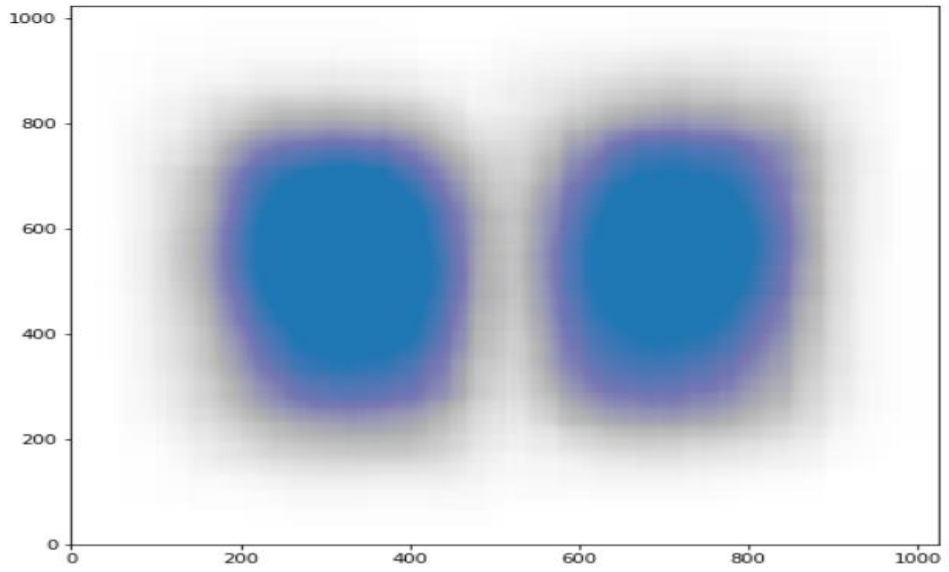
Scatter plot of X & Y Coordinates



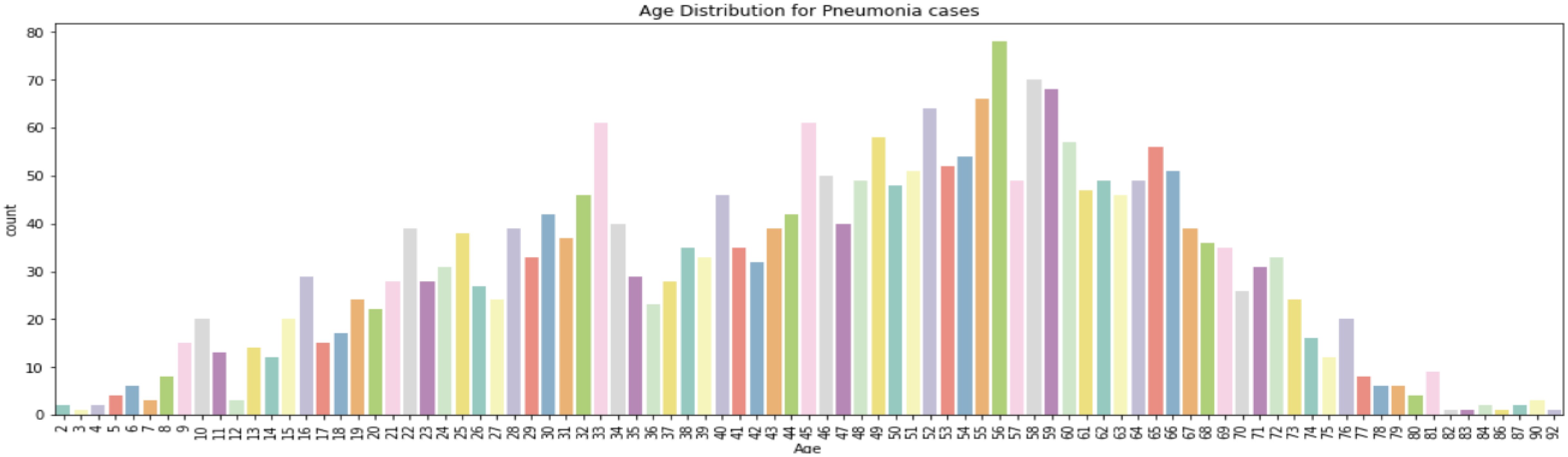
The annotations seem to be spread across both lung region and within the image range

- Few outliers have observed < 1%

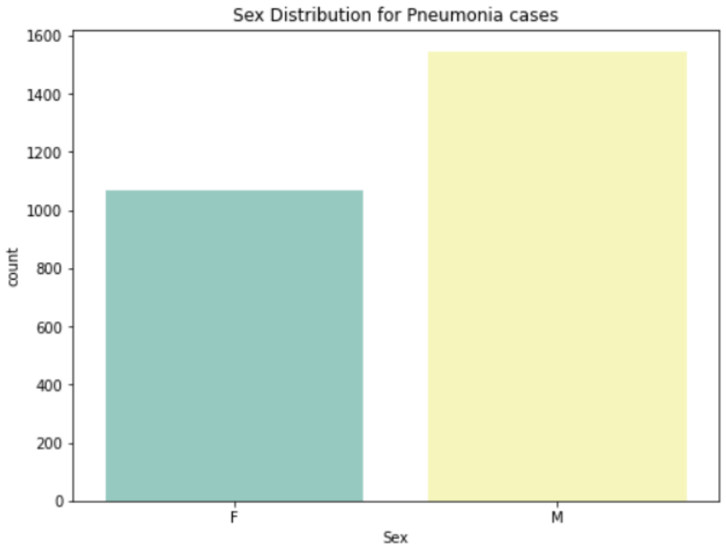
Bounding Box distribution



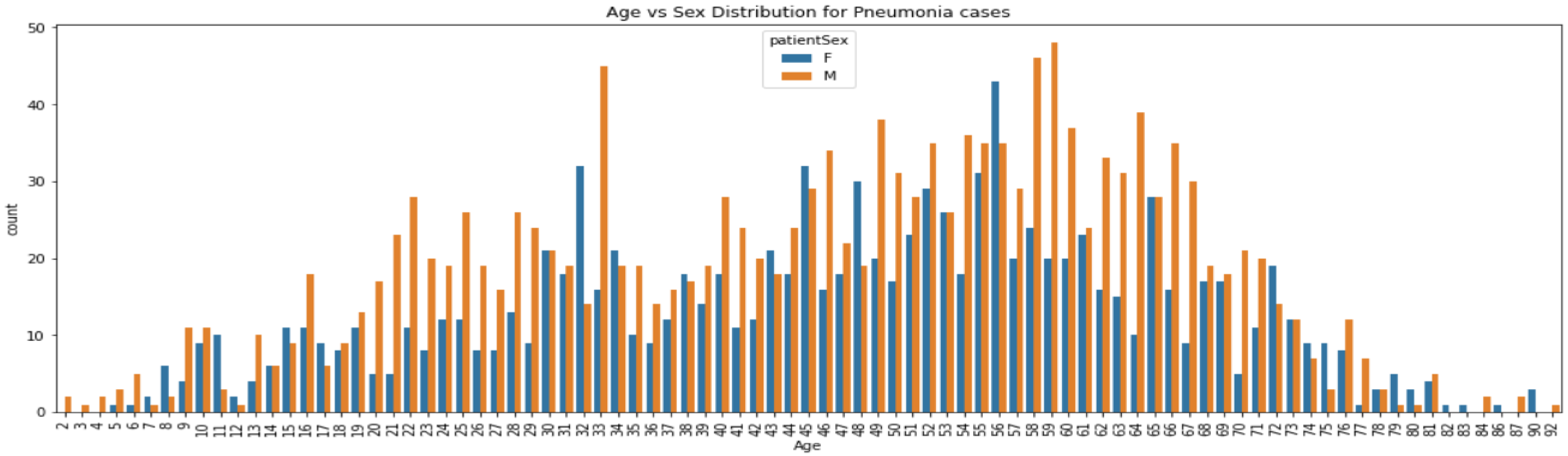
UNIVARIATE ANALYSIS



- Age group of 45- 65 have higher number of patients
- Approx. 50% more cases observed in male patients



BIVARIATE ANALYSIS



- There is uniformity in number of patients by sex (Male, Female) across age groups
- One unusual peak observed at age 33 for male patients

Construction of Null Hypothesis

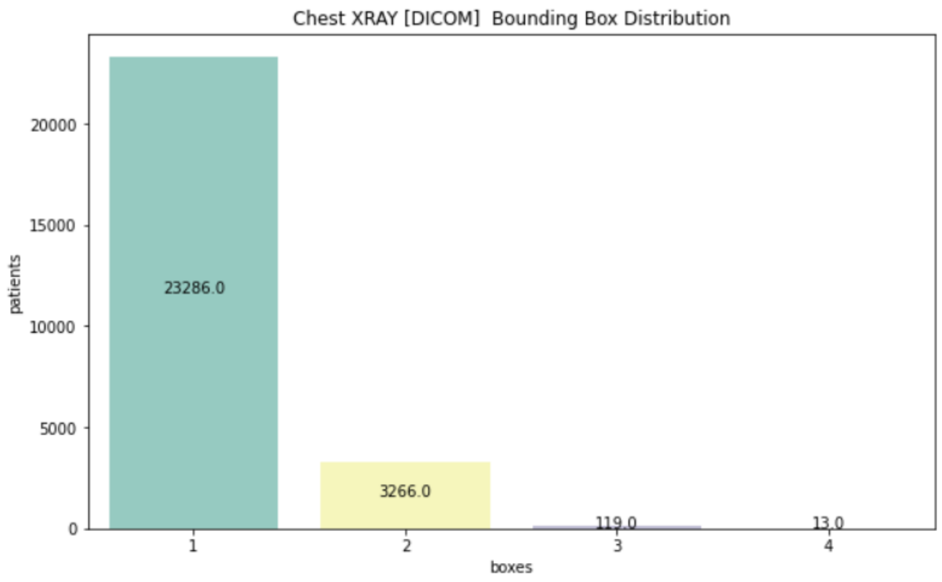
```
# Hypothesis test to check if Gender has an effect on Target class (Pneumonia, Non-Pneumonia)
Ho = "Gender has no effect on Target Class"   # Stating the Null Hypothesis
Ha = "Gender has an effect on Target Class"   # Stating the Alternate Hypothesis

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

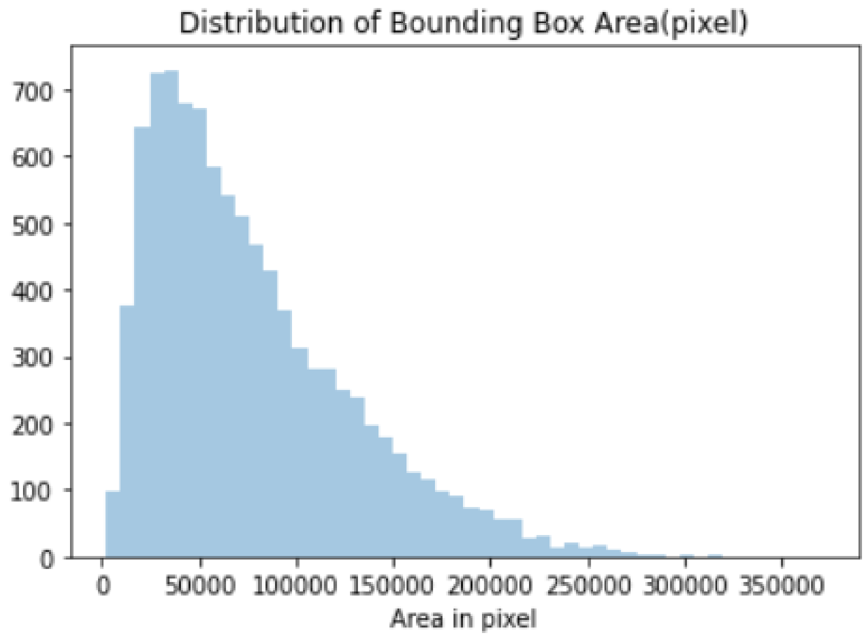
Gender has no effect on Target Class as the p_value (0.57) > 0.05

Hence, Null Hypothesis is Accepted

Feature Engineering



- Maximum of four evidences (bounding box) of pneumonia per patient
- 87.26% of the patients have one evidence (bounding box) of pneumonia



	areaBB
count	9555.000000
mean	77523.448038
std	51807.689206
min	2320.000000
25%	37535.500000
50%	64829.000000
75%	106491.500000
max	371184.000000

- Min size = 2320 , Max size = 371184
- Max size is 16K times larger than min size

MODEL DEVELOPMENT

PROPOSED APPROACH – HIGH LEVEL VIEW

We propose to build Convolutional Neural Network (ConvNet/CNN), a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

Evaluate multiple models using transfer learning and model training mechanism to propose the best model based on performance metrics

Model 1
- MobileNetV2 Transfer Learning
- MobileNetV2 Layer Training

Model 2
- VGG19 Transfer Learning
- VGG19 Layer Training

Model 3
- ResNet50 Transfer Learning
- ResNet50 Layer Training

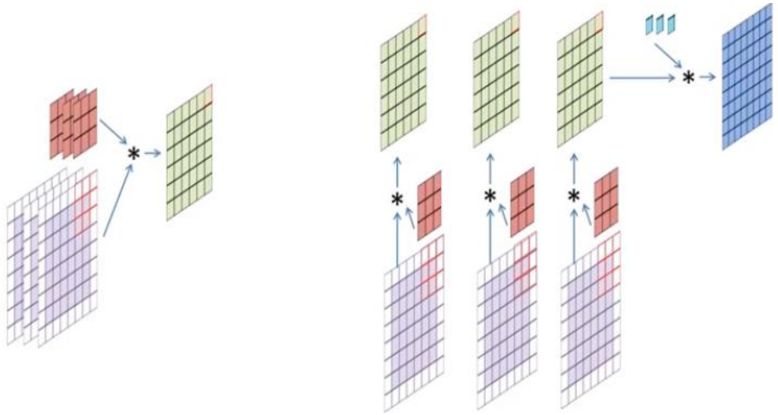
Model 4
- InceptionV3 Transfer Learning
- InceptionV3 Layer Training

Model 5
- Mask R-CNN Resnet50 Backbone, layer training
- Mask R-CNN Resnet101 Backbone, layer training

Proposed Model

MODEL 1 – MobilenetV2

The MobileNet model is based on depthwise separable convolutions which is a procedure of factorized convolutions which factorize a regular convolution into a depthwise convolution and a 1 × 1 convolution named a pointwise convolution. MobileNets depthwise convolution uses a single filter to every input channel. The pointwise convolution then applies a 1 × 1 convolution to merge the outputs the depthwise convolution.



MobilenetV2 CNN Architecture

Actual	0	True Negative	False Positive
	1	False Negative	True Positive
		0	1
		Predicted	

Precision = TruePositives / (TruePositives + FalsePositives)

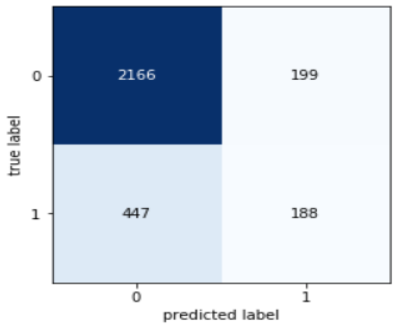
Recall = TruePositives / (TruePositives + FalseNegatives)

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

Transfer Learning : Imagenet weights

Model parameters:
Input Image size : 224 * 224 * 1
Hyper Parameter: alpha = 1.4

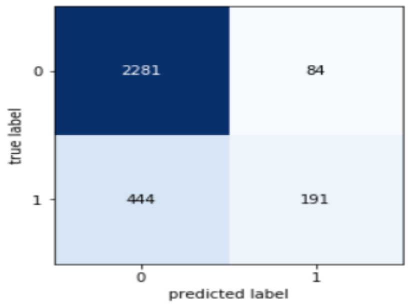
Class	Precision	Recall	F1 Score
Target = 1	0.49	0.30	0.37



Transfer Learning & Model training : Imagenet weights

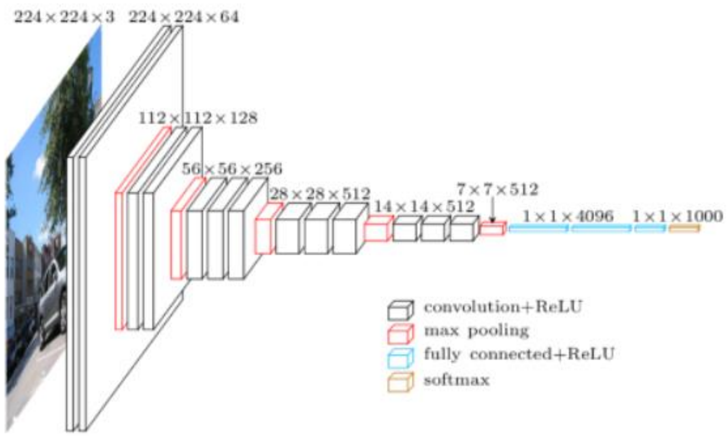
Model parameters:
Input Image size : 224 * 224 * 1
Hyper Parameter: alpha = 1.4
Trainable layers = bottom 3 layers

Class	Precision	Recall	F1 Score
Target = 1	0.69	0.30	0.42



MODEL 2 – VGG19

VGG19 follows a simple architecture, using only *3×3* convolutional layers stacked on top of each other in increasing depth. Reducing volume size is handled by max pooling. Two fully-connected layers, each with 4,096 nodes are then followed by a classifier.



VGG19 CNN Architecture

Actual	0	True Negative	False Positive
	1	False Negative	True Positive
		0	1
		Predicted	

Precision = TruePositives / (TruePositives + FalsePositives)

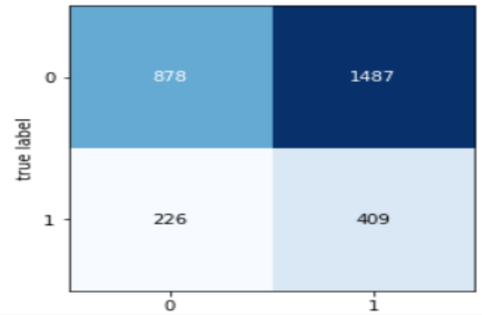
Recall = TruePositives / (TruePositives + FalseNegatives)

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

Transfer Learning : Imagenet weights

Model parameters:
Input Image size : 224 * 224 * 1

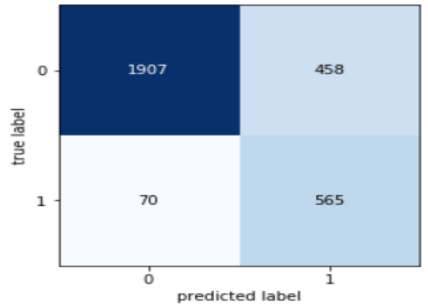
Class	Precision	Recall	F1 Score
Target = 1	0.22	0.64	0.32



Transfer Learning & Model training : Imagenet weights

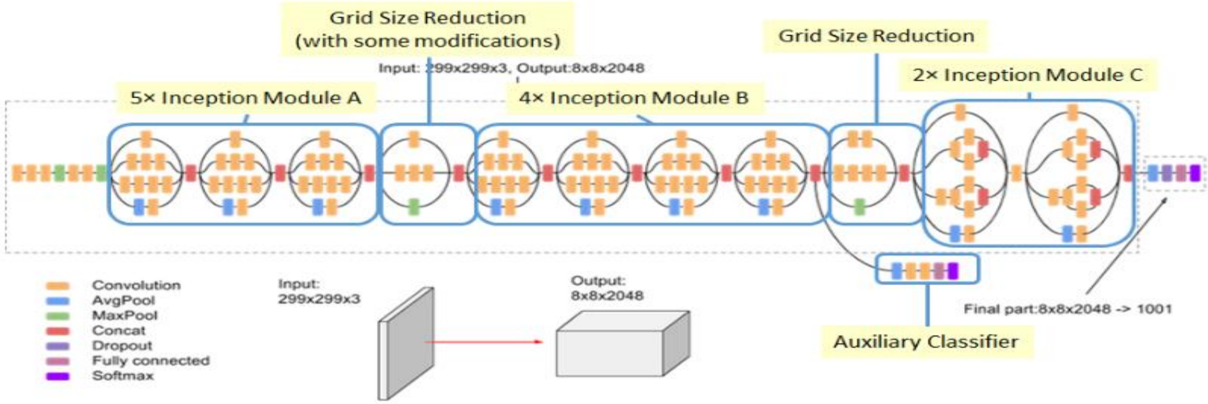
Model parameters:
Input Image size : 224 * 224 * 1
Trainable layers: bottom 2 layers

Class	Precision	Recall	F1 Score
Target =1	0.55	0.89	0.68



MODEL 3 – InceptionV3

The model is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax.



InceptionV3 CNN Architecture

Actual	0	True Negative	False Positive
	1	False Negative	True Positive
		0	1
		Predicted	

Precision = TruePositives / (TruePositives + FalsePositives)

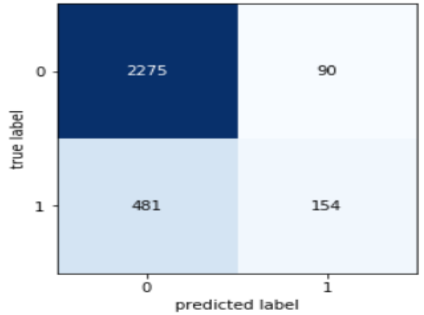
Recall = TruePositives / (TruePositives + FalseNegatives)

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

Transfer Learning : Imagenet weights

Model parameters:
Input Image size : 224 * 224 * 1

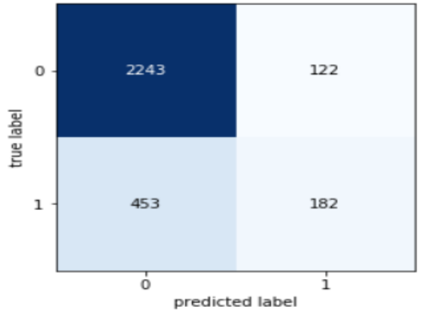
Class	Precision	Recall	F1 Score
Target = 1	0.63	0.24	0.35



Transfer Learning & Model training : Imagenet weights

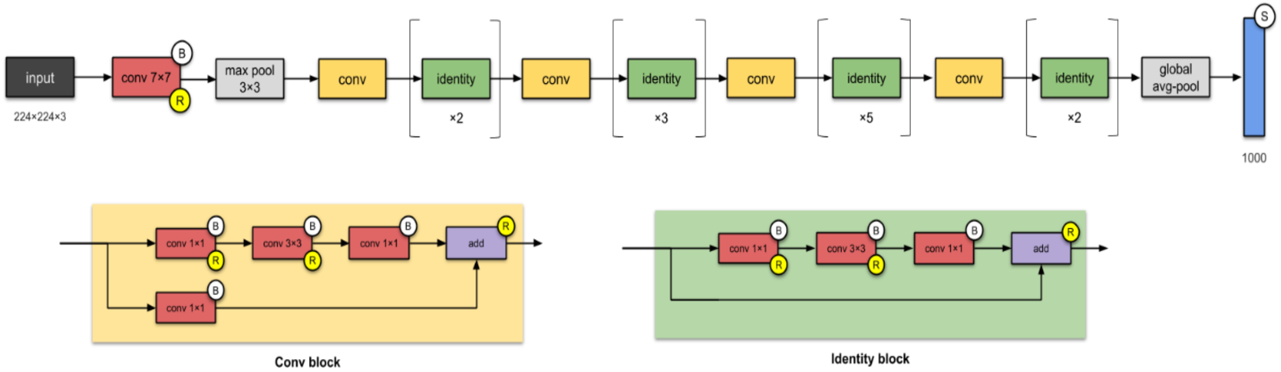
Model parameters:
Input Image size : 224 * 224 * 1
Trainable layers: bottom 5 layers

Class	Precision	Recall	F1 Score
Target = 1	0.60	0.29	0.39



MODEL 4 – ResNet50

The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet architecture introduced skip connections, also known as residual connections to avoid information loss during training of deep network. Skip connection technique enables to train very deep networks and can boost the performance of the model.



ResNet50 CNN Architecture

Actual	0	True Negative	False Positive
	1	False Negative	True Positive
		0	1
		Predicted	

Precision = TruePositives / (TruePositives + FalsePositives)

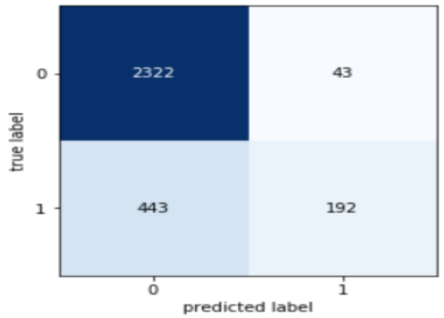
Recall = TruePositives / (TruePositives + FalseNegatives)

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

Transfer Learning : Imagenet weights

Model parameters:
Input Image size : 224 * 224 * 1

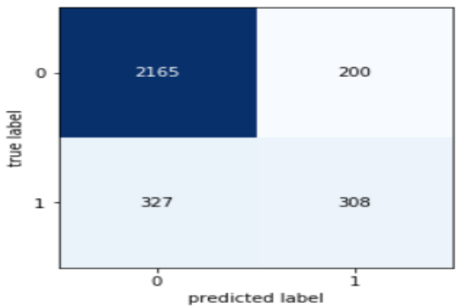
Class	Precision	Recall	F1 Score
Target = 1	0.82	0.30	0.44



Transfer Learning & Model training : Imagenet weights

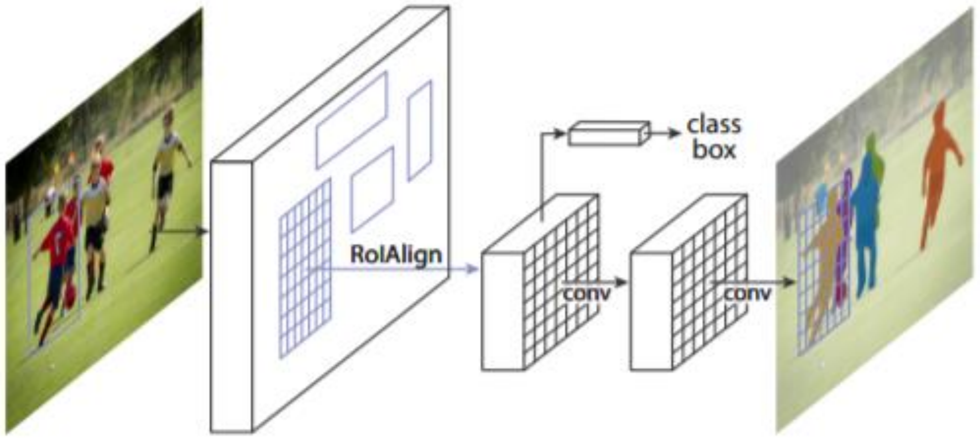
Model parameters:
Input Image size : 224 * 224 * 1
Trainable layers: bottom 5 layers

Class	Precision	Recall	F1 Score
Target =1	0.61	0.49	0.54



MODEL 5 – Mask R-CNN

Mask R-CNN is a deep neural network aimed to solve instance segmentation problem. It adopts two stage procedure, first stage is RPN (Region proposal Network). In the second stage, in parallel to predicting the class and box offset, Mask R-CNN also outputs a binary mask for each RoI.



Mask R-CNN Framework

Actual	0	True Negative	False Positive
	1	False Negative	True Positive
		0	1
		Predicted	

Precision = TruePositives / (TruePositives + FalsePositives)

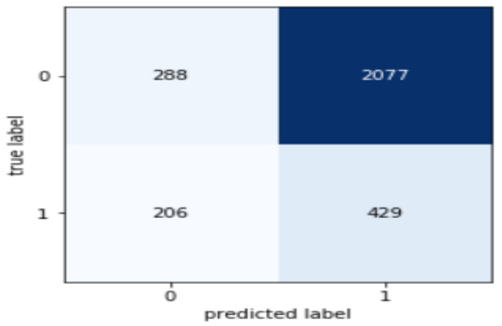
Recall = TruePositives / (TruePositives + FalseNegatives)

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

Model Training with Resnet50 Imagenet weights

Model parameters:
Backbone : Resnet50
Trainable layers : All
Threshold : > 99

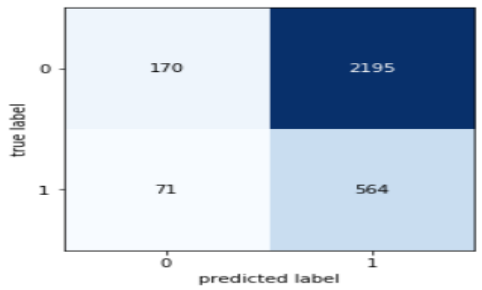
Class	Precision	Recall	F1 Score
Target = 1	0.17	0.68	0.27



Model Training with Resnet50 Imagenet weights

Model parameters:
Backbone : Resnet50
Trainable layers : heads
Threshold : > 0.98

Class	Precision	Recall	F1 Score
Target = 1	0.20	0.89	0.33



MODEL 5 – Mask R-CNN

Actual	0	True Negative	False Positive
	1	False Negative	True Positive
		0	1
		Predicted	

Precision = TruePositives / (TruePositives + FalsePositives)

Recall = TruePositives / (TruePositives + FalseNegatives)

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

Model Training with mask_rcnn_coco.h5 weights

Model parameters:
Backbone : Resnet101
Trainable layers : All
Threshold : > 98

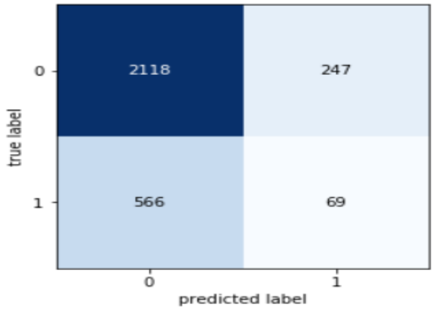
Currently being trained

Class	Precision	Recall	F1 Score
Target = 1	0.22	0.11	0.15

Model Training with mask_rcnn_coco.h5 weights

Model parameters:
Backbone : Resnet101
Trainable layers : heads
Threshold : > 0.98

Class	Precision	Recall	F1 Score
Target = 1	0.22	0.11	0.15



CNN Models – Performance Metrics

Model Input:
# of images : 26684
Training & validation Dataset split : 80 : 20
Batch size : 50
Epoch : 5

Inference
 # Test Images : 3000
Target class=1 threshold set to 0.4 for all models except Mask R-CNN
Mask R-CNN Target class =1 threshold set to >0.98

#	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

CNN Models – Performance Metrics

Model Input:
# of images : 26684
Training & validation Dataset split : 80 : 20
Batch size : 50
Epoch : 5

Inference
 # Test Images : 3000
Target class=1 threshold set to 0.4 for all models except Mask R-CNN
Mask R-CNN Target class =1 threshold set to >0.98

#	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		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		0.22	0.11	0.15	0.19

CONCLUSION

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 five models the best model to detect pneumonia.

The Mask R-CNN model is the best model based on the observations having a model evaluation metric of 0.4.

For future studies, adaptation of other convolutional neural network architectures like yolo, SSD architectures for pneumonia detection must be implemented and the optimization of hyper-parameters should also be considered to improve the accuracy of the model. 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.

Future considerations:

- Explore Resnet50, VGG19, Mask R-CNN with high number of epochs and fine tuning of hyperparameters
- Explore other advanced techniques such as YOLO & SSD

THANK YOU