Pulmonary Disease (Covid, Pneumonia, Viral Lung Infection) Classification based on Chest X-Rays using Neural Network

Mayeesha Humaira (ID: 16.02.04.008)

160204008@aust.edu

Md. Abidur Rahman Khan Jim (ID: 16.02.04.017)

160204017@aust.edu

Ashna Nawar Ahmed (ID: 16.02.04.024)

160204024@aust.edu

1. Introduction

Coronavirus pandemic is the biggest issue that has currently overtaken the world and it has completely halted the development of every possible sector in the world such as economic, industrial, agriculture etc. as people rushed to protect themselves. It all began in 2019, as an outbreak of the novel coronavirus was first reported in December 2019 when cases of viral pneumonia with unknown origin were confirmed in Wuhan, Hubei Province, China. As the disease quickly spread and took on the form of a pandemic, as declared by WHO on March 11, 2020, it became imperative that health care providers be able to distinguish between Covid-19 and other pulmonary diseases. However, due to similarities in symptom presentation between the formally named Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) and typical Pneumonia, it was initially a huge challenge to identify which patients were affected with Covid-19 vs. which ones were affected with pneumonia or lung infection. The lung opacity seen in lung x-rays were incredibly similar in these cases, which caused a lot of covid patients to be misdiagnosed as pneumonia patients or lung infection patients, which in turn caused the spread of the deadly contagious disease even further due to lack of safety precautions taken by the average non-covid patient. This is why we have decided to build a Neural Network model that can distinguish between the x-rays of Covid-19 patients, Pneumonia (bacterial and viral) patients and lung infection (viral) patients. The **uses of this task** could be manifold. Some are mentioned as follows:

- It could aid health care providers to obtain a second opinion about a patient's diagnosis without having to consult colleagues, thereby reducing their workload
- It could cut down on time taken to perform a diagnosis
- It could help to reduce the risk taken by having too many healthcare workers in one place by reducing the load taken on by radiologists

The **motivation** behind this project was to aid in reducing the workload on the overworked healthcare workers by the use of a state-of-the-art Neural Network model that utilises past x-rays of lungs for each of the diseases along with normal, non-diseased lung x-rays to distinguish between the diagnoses and provide a verdict. Solving this problem could have a positive impact on the healthcare sector. This model could help to reduce uncertainty in pulmonary disease diagnosis as well as shrinking the rate of false negatives, which can be incredibly risky as well as potentially deadly for everyone involved. Pulmonary disease diagnosis **applications** could rely on this model to potentially improve their results.

The main **challenge** faced in completing this task was the lack of data. It issued us with a major obstacle while solving this problem, as Neural Networks yield better results when it is provided with more data. We could obtain 23,695 images of chest x-rays in total, which consisted of healthy lungs, lungs with pneumonia and lungs with covid. The other challenge that we faced was getting high quality accuracy without the use of U-Nets and ResNets. As we only used basic Neural Networks instead of pretrained U-Nets and ResNets, our accuracy was unfortunately not as high as it could have been.

2. Related Works

We have found some related works, as COVID is of course the most urgent and important topic that is being researched upon in the current times. We are listing 2 of the related works along with their working approach i.e. how they tackled the issues being addressed as well as their limitations as follows:

Deep neural network to detect COVID-19: one architecture for both CT Scans and Chest X-rays [1] has proposed a Convolutional Neural Network (CNN)-tailored Deep Neural Network (DNN) that can collectively train/test both CT scans and CXRs.

Working Approach:
- The model can take both CXRs and CT scan images as input and classify them as covid or non-covid.
- The model is made of 3 Convolution layers and 3 dense layers. It also consists of Max pooling layers in between the convolutional layers. The third dense layer is responsible for binary classification.
- Collected CT and CXRs images from different sources and could gather a total of 672 images.
- Achieved an overall accuracy of 96.28% (AUC = 0.9808 and false negative rate = 0.0208)

Limitations:
- The total data collected is very small. 672 images are not enough to train a Convolutional Neural Network. Whereas, we collected 23695 images.
- They did binary classification which involves only two classes covid and non-covid. Our model classifies 4 classes.

Artificial Intelligence Applied to Chest X-Ray Images for the Automatic Detection of COVID-19. A Thoughtful Evaluation Approach [2] has proposed to develop an automatic COVID-19 diagnosis tool using chest X-Ray images to differentiate between controls, pneumonia, or COVID-19 groups.

Working Approach:

- The model can take chest X-Ray images as input and provides a score for each of the three classes (i.e. control, pneumonia, or COVID-19) as output.
- The core of the model is a deep CNN based on the COVID-Net2 with some modifications made by tuning the hyperparameters.
- The input image was cropped to a square of 224*224 pixels located in the center of the image.
- A 91.5% classification accuracy is obtained, with an 87.4% average recall.

Limitations:

- The classification was based on only 3 classes. We did classification for 4 classes.
- Cropping the image to the square of 224*224 pixels at the center could eliminate important features in the image. Hence, we resized our images to a size of 400*400.

3. **Project Objective**(s)

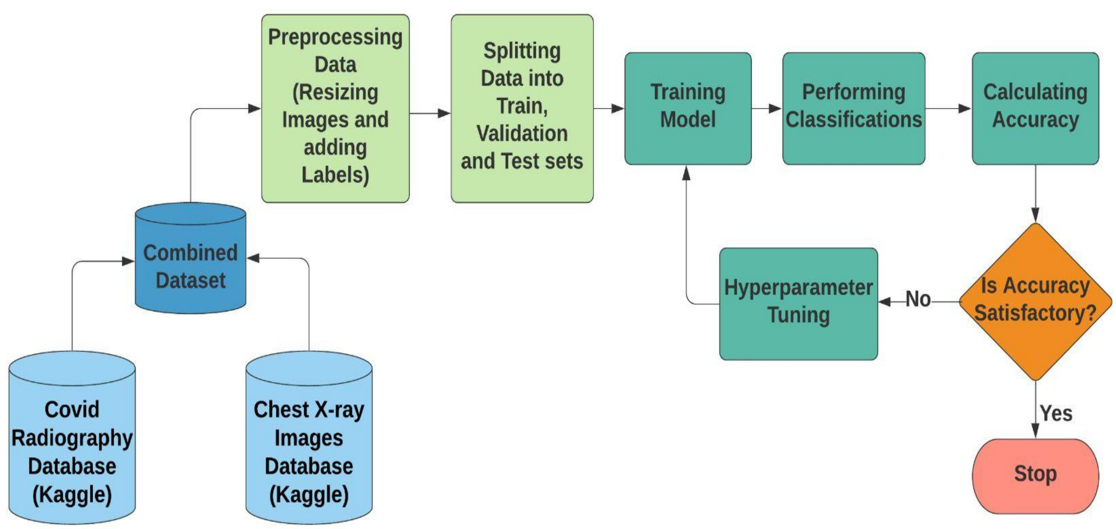
The tasks that will be performed by our proposed system can be divided into the following subtasks:

A. **Compiling X-ray images into datasets:** In the initial step, we obtained data from 2 separate Kaggle databases and compiled them together into a single dataset. The first database (Covid Radiography Database [3]) contained data for x-rays for covid, normal and lung infection affected lungs. The second database (Chest X-ray Images [4]) contained images of normal and pneumonia affected lungs' chest x-rays. The data was compiled from separate CSV files.
B. **Pre-processing of data:** In this step, we processed the data. We resized all the images into 400x400 size for ease of processing. We also added labels such as 0(normal lungs), 1(covid affected lungs), 2(pneumonia affected lungs) and 3(infection affected lungs).
C. **Splitting data into train, validation and test datasets:** In this step, we split our complete dataset into 3 parts: train, validation and test sets. The ratio was 80:10:10 respectively.
D. **Training the model:** In this step, we trained our neural network(NN) model using the training data. Various activation functions were used along with other variations in settings were performed.
E. **Performing Classifications:** In this step, we tested our model, first with the validation dataset, and then with the test dataset. The reason for using separate validation and test datasets is for more accurate tuning of hyperparameters.
F. **Measuring Accuracy of Predictions:** In this step, we calculated the accuracy obtained from our predictions in order to determine how well our model was

performing. Based on the improvement seen, the settings were subsequently adjusted.

G. Tuning Hyperparameters: In this step, the hyperparameters such as batch size, epoch size, number of iterations, learning rate and number of hidden NN layers were tuned in order to obtain better predictions, and subsequently steps D through G were repeated.

A flowchart showing the subtasks is as follows:



Now, some dummy inputs and outputs of the system are shown as follows:

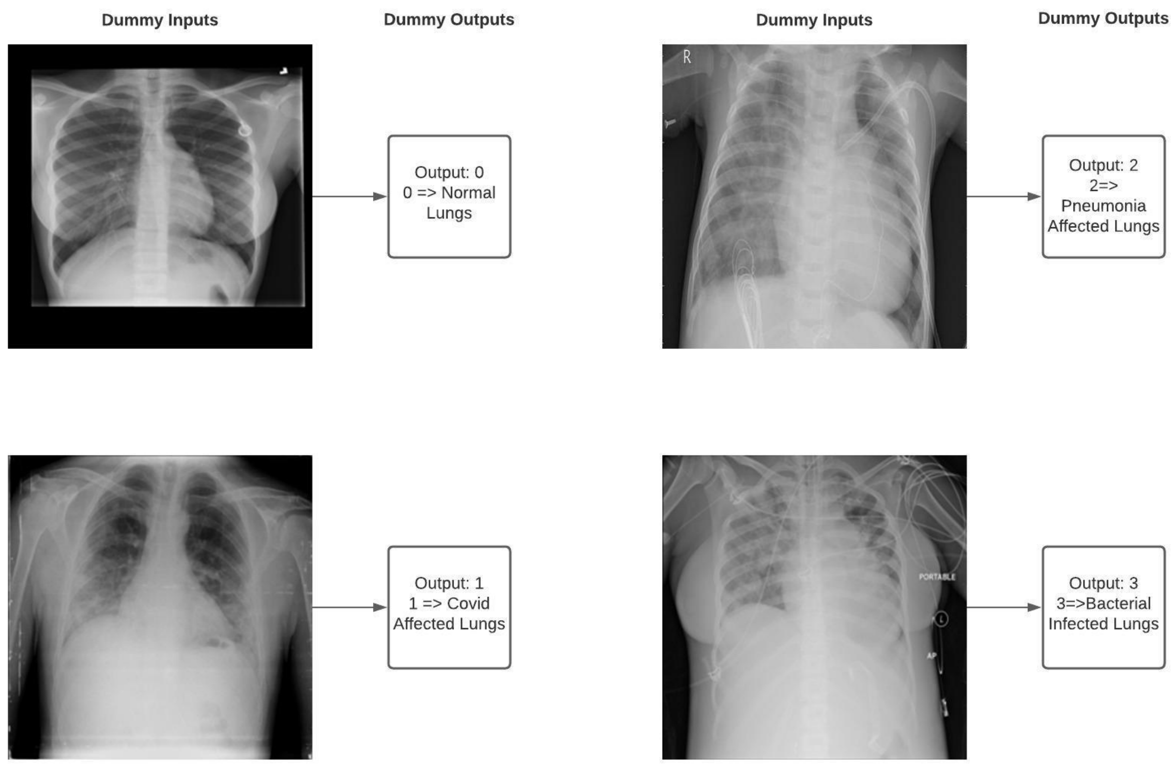


Fig: Dummy Inputs and Outputs

4. **Methodologies / Model**

In this project, we will be using Deep Neural Network as our model. Deep Neural Network is a part of the Machine learning family and is based on Artificial Neural Network (ANN). ANNs were inspired by the biological Neurons that form our brains. We will be using Supervised Learning process as our model will have Gold Standard labeled data as the basis upon which to learn. So our model will be learning how to differentiate between x-rays of 4 different diagnoses through existing, labeled x-rays.

The term "deep" in deep learning refers to the use of multiple layers in the network. Having more than one hidden layer differentiates between Shallow and Deep Neural Networks. We will be using DNNs as they tend to provide better results. DNN uses multiple layers to progressively extract higher-level features from the raw input. So lower layers might identify edges of the images we use, while the higher layers will help differentiate between the various types of x-ray images. In an image recognition application like ours, the original, raw input is a matrix of pixels. So how the model works on these images might be as follows:

- The first layer may encode edges and abstract the pixels
- The second layer may encode the various parts of the chest x-rays and differentiate between the bones and lungs
- The third layer may recognize and identify which x-rays belong to which category of disease

While we might hypothesize the above functions happening in each layer, we cannot truly know for sure, as a DNN learns which features to optimally place on which level on its own, and the process remains hidden from us. All we can do is tune the hyperparameters such as varying the number of hidden layers, layer sizes, which layer uses which Activation functions and so on.

The model we are using will look as follows:

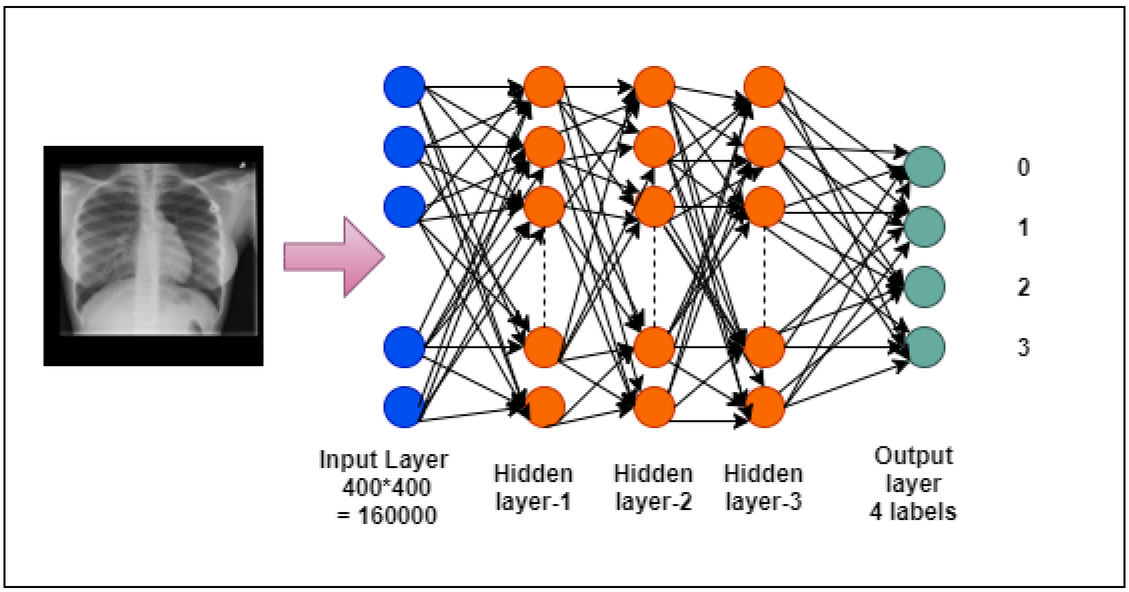


Fig: Deep Neural Network Model

Here, we are using an **input layer** of size 400x400=**160000,** as each image is 400x400 in size. The **output layer** will have size=**4**, as there are 4 different classes (normal, COVID, pneumonia, lung opacity due to infection) we are trying to identify. We have determined through ablation experiments that will be shown later in the report that the number of optimal **hidden layers** for our experiment is **3**.

5. **Experiments**

A. Dataset:

We utilised two datasets from Kaggle for this project. The first dataset (COVID-19 Radiography Database) has the following specifications:

Diagnosis	Number of Samples
Normal Lungs	10192
COVID-affected Lungs	3616
Infected Lungs with Opacity	6012

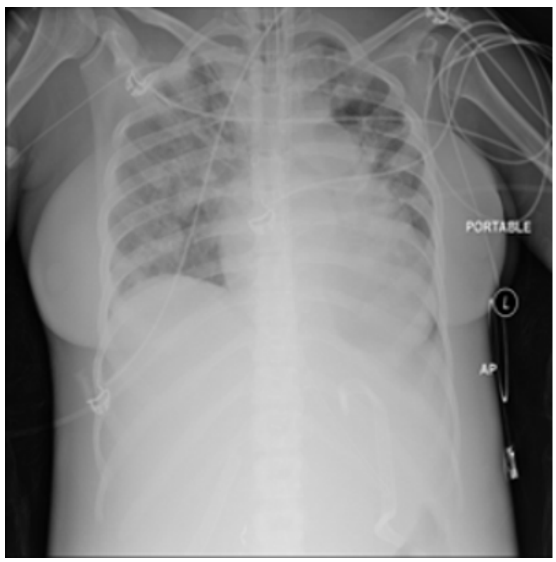
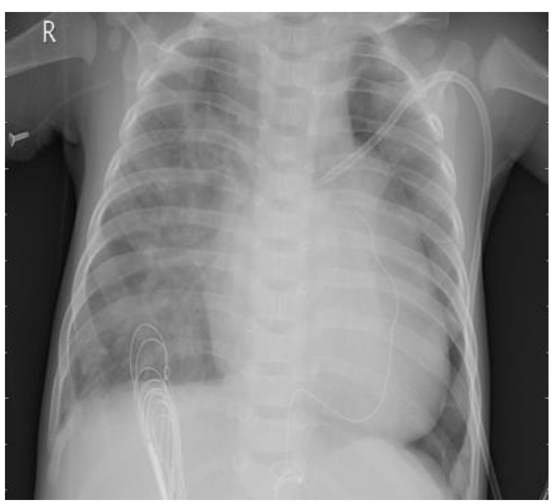
The second dataset (Chest X-ray Images) has the following specifications:

Diagnosis	Number of Samples
Pneumonia-affected Lungs	3875

Thus we had (10192+3616+6012+3875)=23695 data in total.

Sample Data from each set along with their respective classes is shown as follows:

Diagnosis	Sample Data	Class Label
Normal Lungs		**0**

COVID-affected Lungs		1
Infected Lungs with Opacity		2
Pneumonia-affected Lungs		3

We utilised a **80:10:10 ratio split** for the train:validation:test datasets. So the specifications became as follows:

Datasets	Number of Samples
Total Dataset	23695
Training Set (80%)	18957
Validation Set (10%)	2369
Testing Set (10%)	2369

B. Evaluation Metric

Evaluation metrics are used to measure the quality of the statistical or machine learning model. There are many different types of evaluation metrics available to test a model. Among these, we will be using the following 3:

- **Classification Accuracy:** Here we will measure how many predictions were correct out of the total numbers of predictions, and calculate the percentage of accuracy based on that.
- **Loss Function:** Loss function is considered to be the penalty for an incorrect classification of an example. It can also be considered as some function of the difference between estimated and true values for an instance of data. Our goal would be to minimize the Loss Function in order to obtain the most optimum results possible.
- **Confusion Matrix**: It is a matrix that represents True Prediction Rate and False Prediction Rate by showing the relationship between Actual data and Predicted data. We will also be using this representation to evaluate our system.

We will be seeing all three types of evaluation metrics for each of the experiments we perform.

C. Results

We have performed **ablation experiment** based on our hyperparameters (Batch Size, Epoch Size, Number of Hidden Layers, Size of Hidden Layers, Learning Rate and Activation functions used in each layer) in order to incrementally improve our results and determine which setting provides the best result for our experiment without overfitting. The hyperparameters chosen for the ablation experiment are shown as follows:

Settings	Batch Size	Epoch Size	Learning Rate	Size of Hidden Layers	Number of Hidden NN Layers	Activation Functions in Hidden Layers
1	130	30	0.001	50	3	ReLU->ReLU->ReLU
2	100	23	0.001	50	3	ReLU->ReLU->ReLU
3	150	35	0.001	50	3	ReLU->ReLU->ReLU
4	100	23	0.001	80	3	ReLU->ReLU->ReLU
5	100	23	0.001	20	3	ReLU->ReLU->ReLU
6	100	23	0.0001	20	3	ReLU->ReLU->ReLU
7	100	23	0.01	20	3	ReLU->ReLU->ReLU
8	100	23	0.001	20	2	ReLU->ReLU
9	100	23	0.001	20	4	ReLU->ReLU->ReLU->ReLU
10	100	23	0.001	20	3	RReLU->RReLU->RReLU
11	100	23	0.001	20	3	SELU->SELU->SELU
12	100	23	0.001	20	3	RReLU->ReLU->SELU

Table 1: Hyperparameters utilised in each Setting

The experimental settings were varied as follows:

- In **setting 1, 2 and 3**, we varied the Batch Size and Epoch Size hyperparameters. The other hyperparameters remained constant.
- After deciding on the optimal **Batch Size (100)** and **Epoch Size (23)**, we then varied the Size of Hidden Layers in **settings 4 and 5**.
- After deciding on the optimal **Size of Hidden Layers (20)**, we then varied the Learning Rate in **settings 6 and 7**.
- After deciding on the optimal **Learning Rate (0.001)**, we then varied the Number of Hidden Layers in **settings 8 and 9**.
- After deciding on the optimal **Number of Hidden Layers (3)**, we then varied the Activation Functions of Hidden Layers in **settings 10, 11 and 12**.

The results obtained through the **ablation experiment** are shown as follows:

Settings	Training Accuracy	Validation Accuracy
1	93.73	85.353
2	92.073	87.455
3	93.773	83.867
4	92.979	85
5	88.531	85.318
6	90.469	84.727
7	83.068	81.227
8	84.99	81.045
9	90.901	85.318
10	90.641	85.273
11	88.891	84.273
12	90.547	86.591

Table 2: Accuracy for Training Set and Validation Set

Here, we are evaluating loss on the basis of the **difference between Training Accuracy and Validation Accuracy**, and adjusting our hyperparameters according to the loss and accuracy. The results are discussed below:

- In **settings 1, 2 and 3**, after varying Epoch Size (30, **23**, 35) and Batch Size (130, **100**, 150) as we can see, the losses were 8.377%, **4.618%** and 9.906% respectively. As setting 2 had the least amount of loss, moving forward we used the values of **Epoch Size=23** and **Batch Size=100**. The graphs and confusion matrices for each setting are shown as follows:

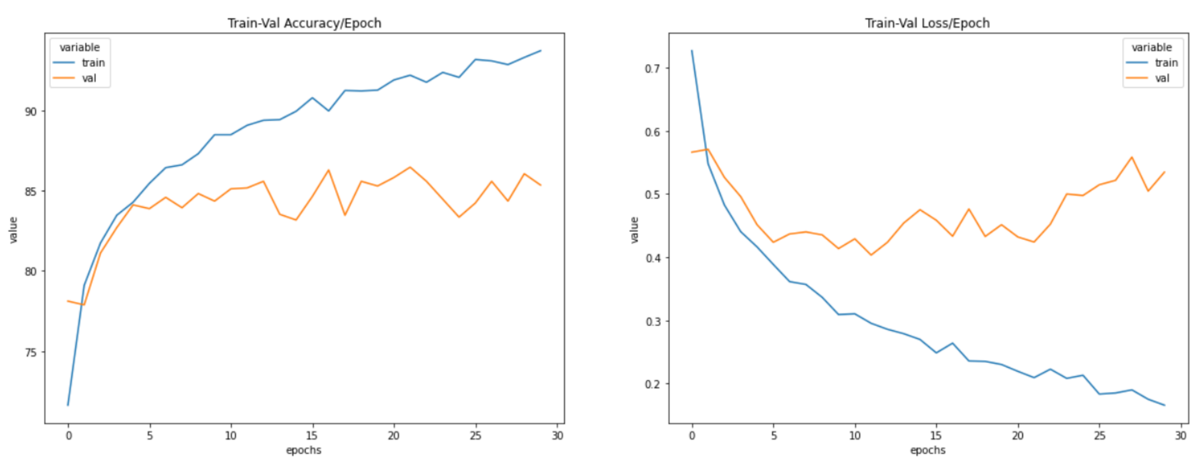


Fig: Loss and Accuracy for Setting 1

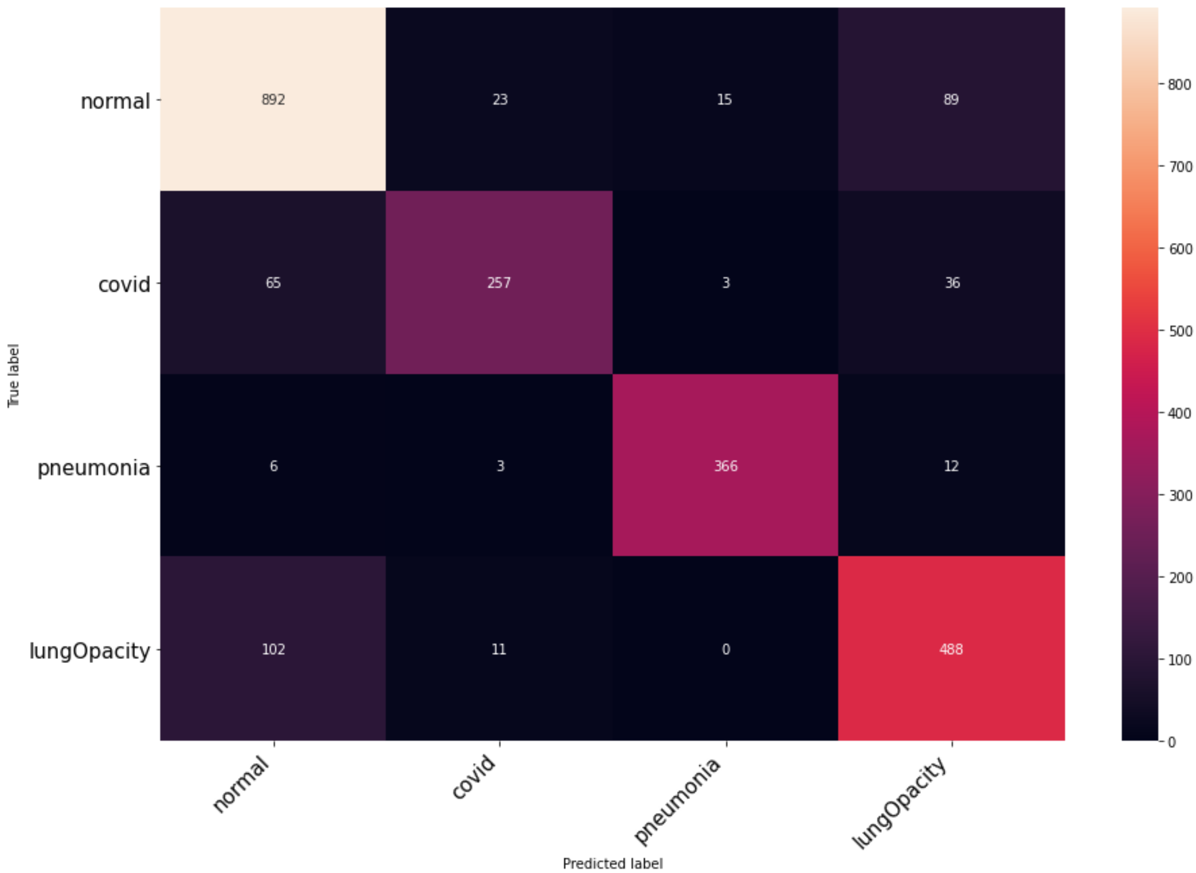


Fig: Confusion Matrix for Setting 1

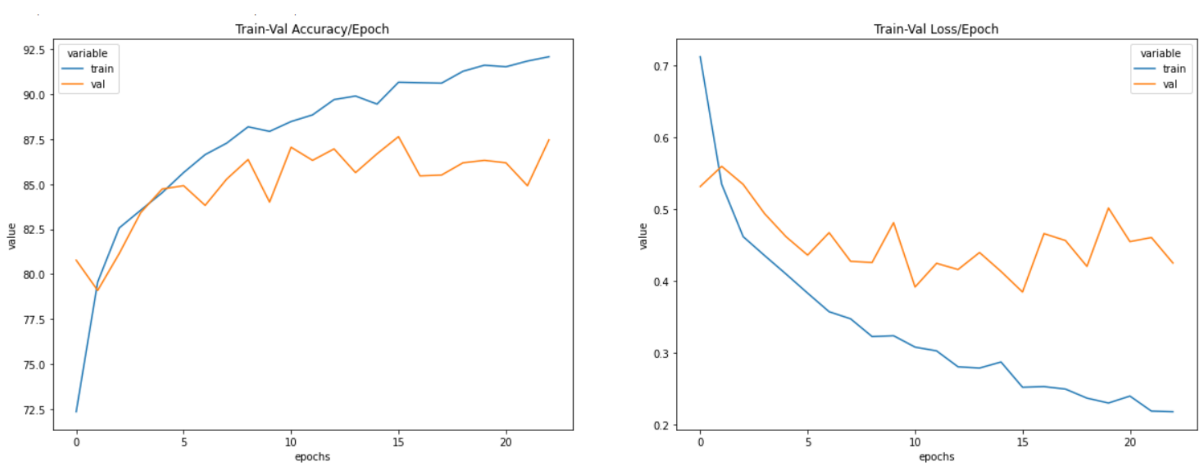


Fig: Loss and Accuracy for Setting 2

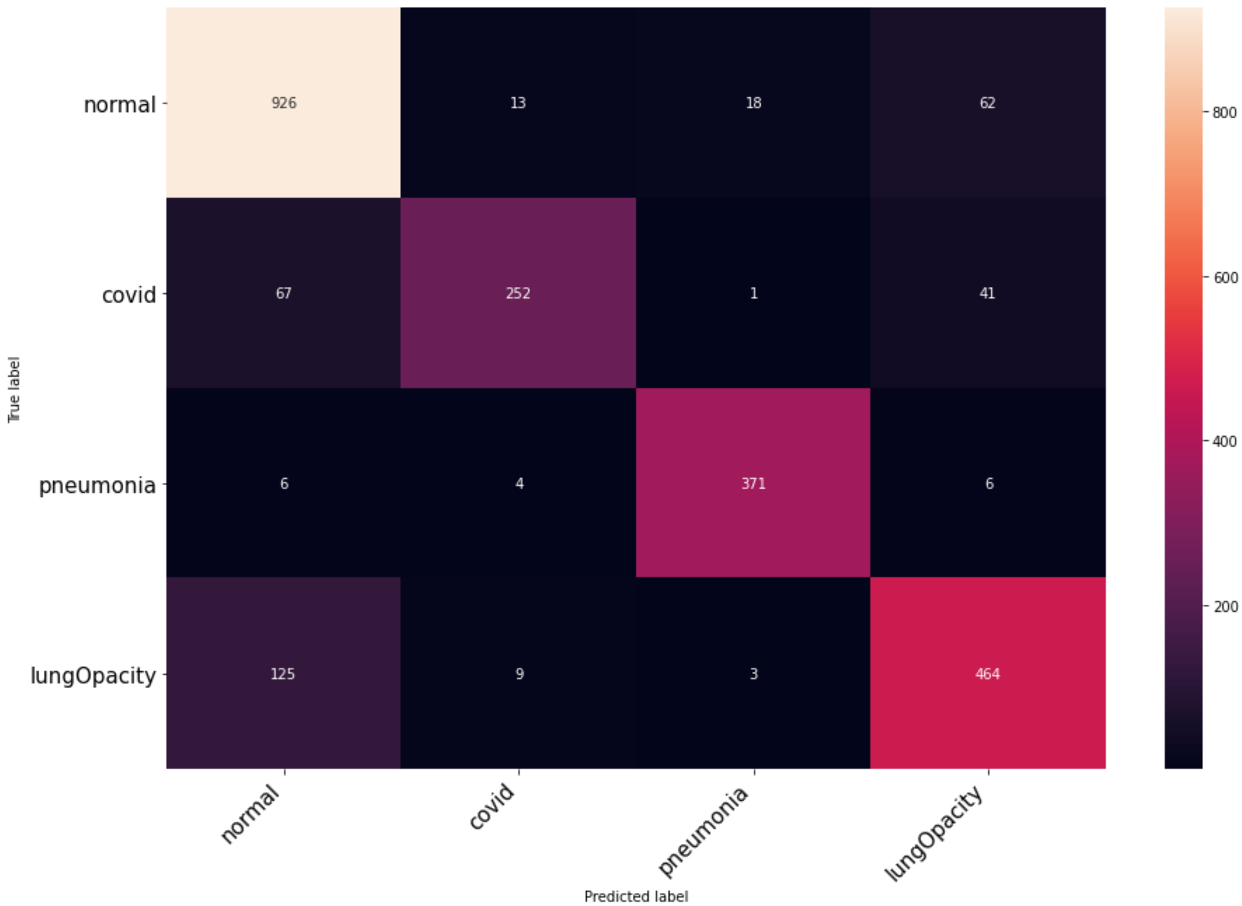


Fig: Confusion Matrix for Setting 2

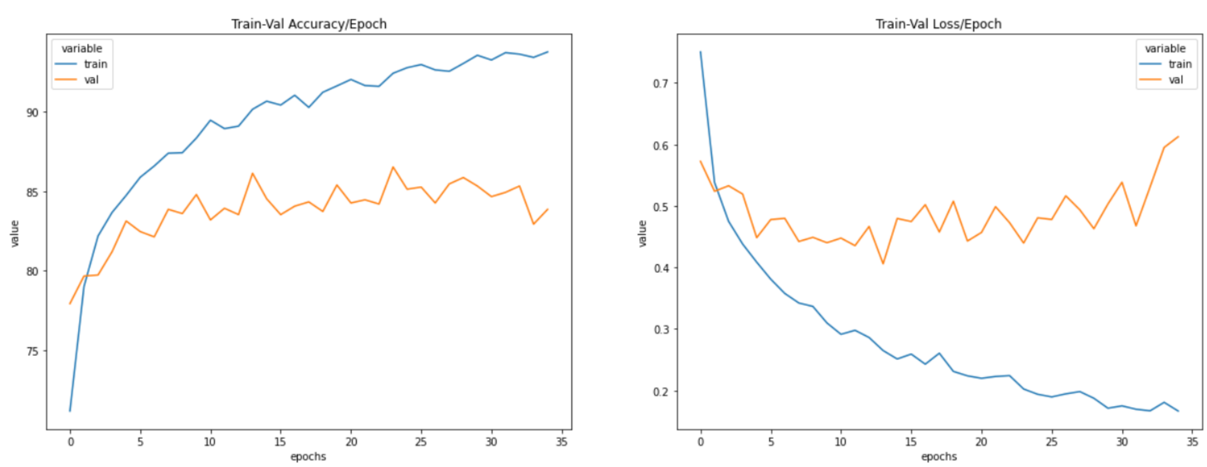


Fig: Loss and Accuracy for Setting 3

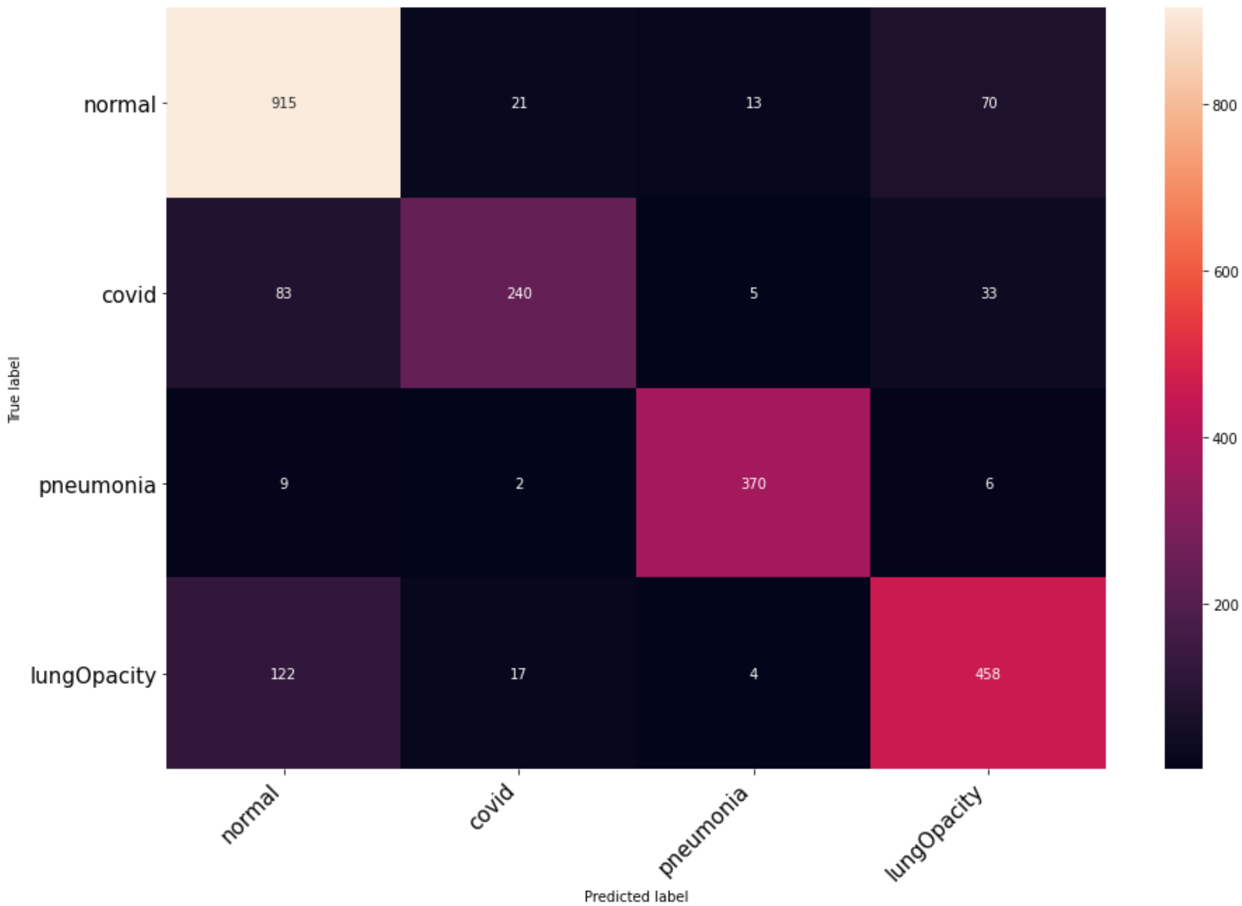


Fig: Confusion Matrix for Setting 3

- In **settings 4 and 5** we varied the Size of Hidden Layers, by changing it in both directions from the baseline (50); to 80 (increasing) and 20 (decreasing) respectively. After performing this, we observed that the losses were 7.979% and **3.213%** for settings 4 and 5 while that for the baseline, setting 2, was 4.618% as seen previously. As the loss for setting 5 was the least, we used **Size of Hidden Layers=20** in the subsequent experiments. The graphs and confusion matrices for each setting are shown as follows:

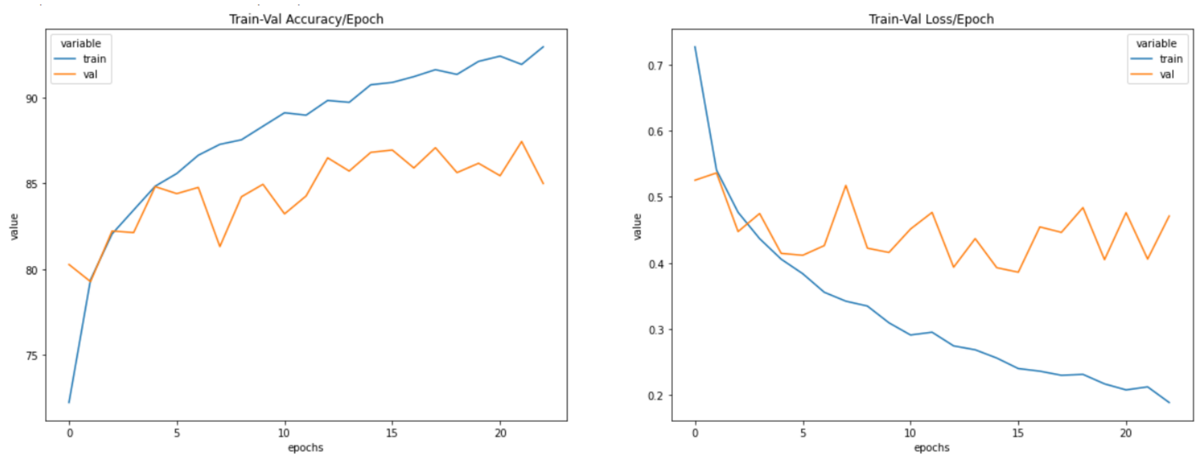


Fig: Loss and Accuracy for Setting 4

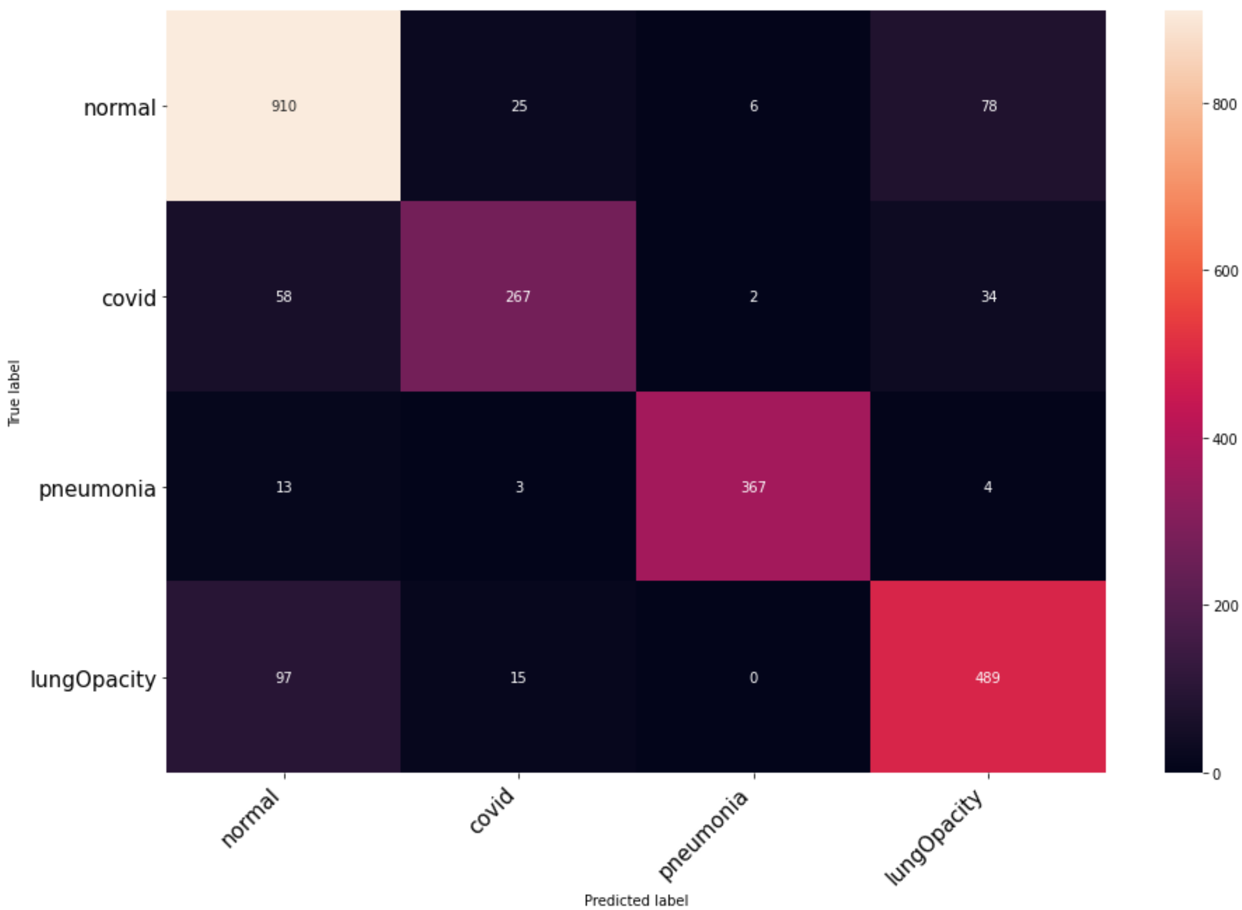


Fig: Confusion Matrix for Setting 4

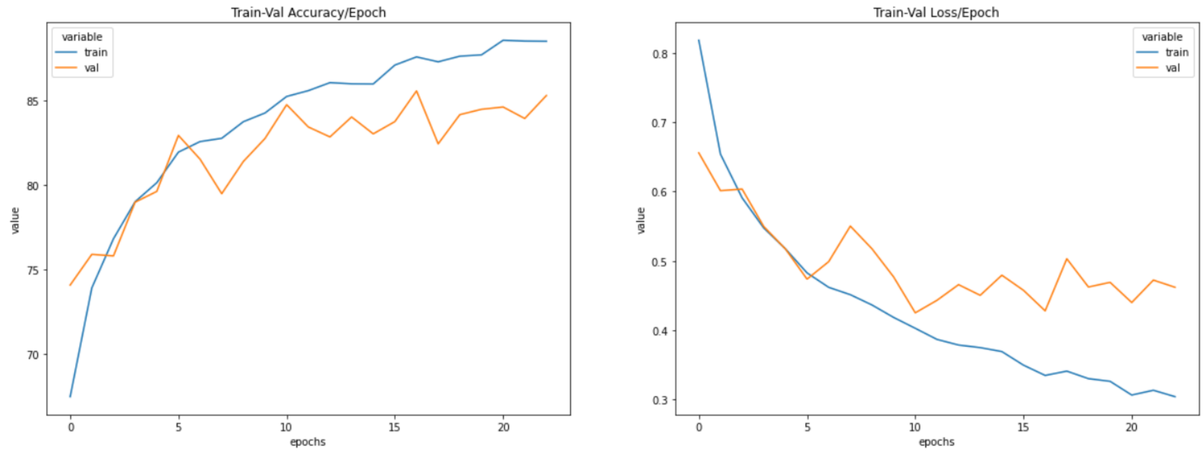


Fig: Loss and Accuracy for Setting 5

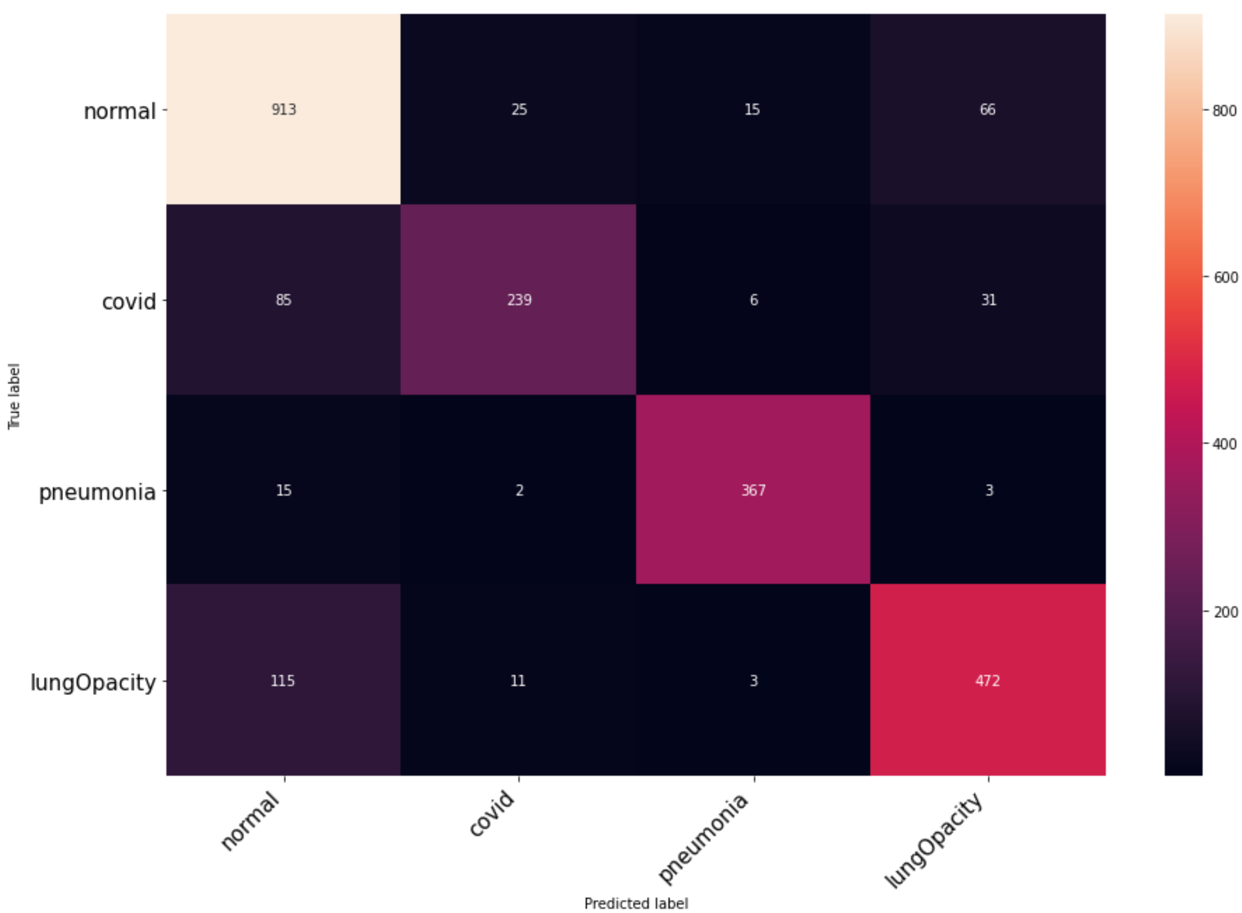


Fig: Confusion Matrix for Setting 5

- In **settings 6 and 7,** we varied Learning Rate (LR), both in upwards and downwards directions. While the baseline for LR was 0.001, we raised it to 0.01 in setting 7 and lowered it to 0.0001 in setting 6. After performing these we observed that the losses were 5.742% and 1.841% respectively. Even though the loss in setting 7 was very low, however, the accuracy was also very low (a mere 83.068% for the Training Dataset). So we decided **LR=0.001** was the optimal setting, which is setting 5. The graphs and confusion matrices for each setting are shown as follows:

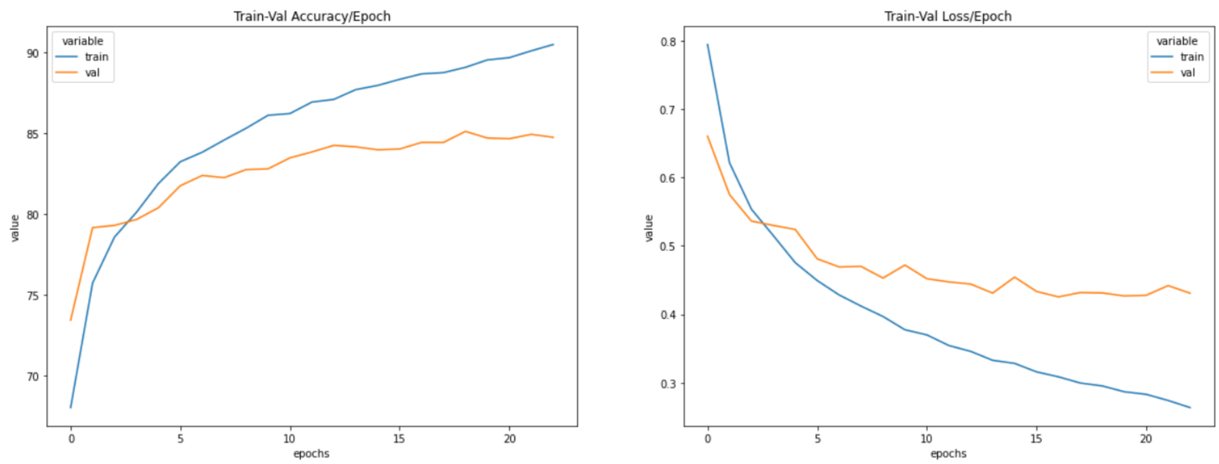


Fig: Loss and Accuracy for Setting 6

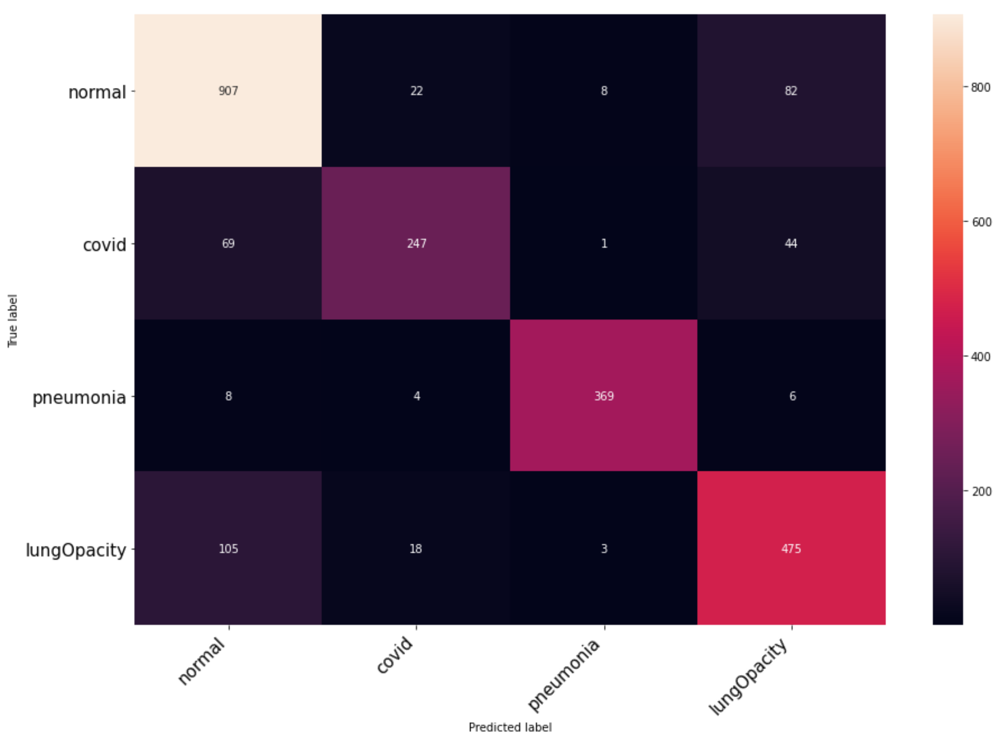


Fig: Confusion Matrix for Setting 6

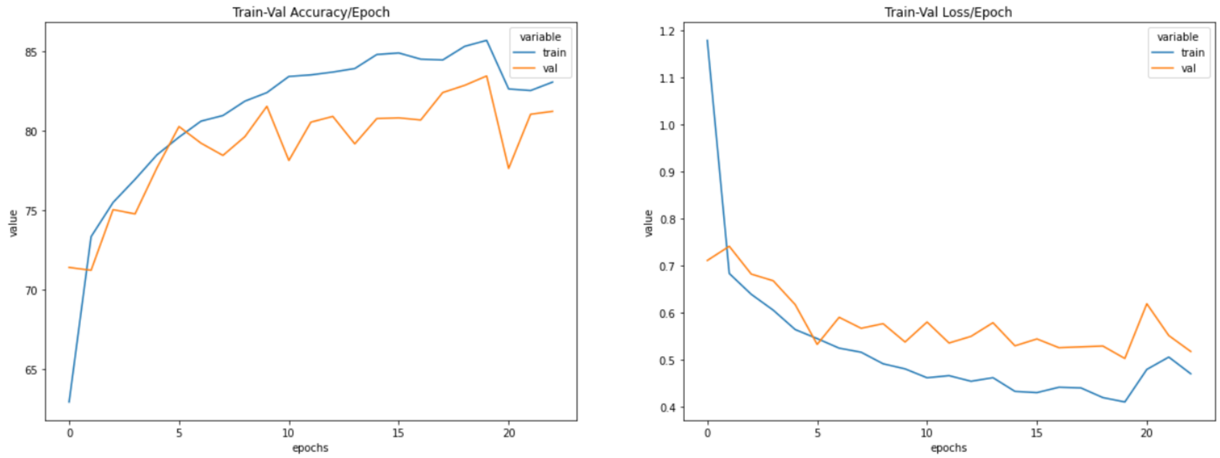


Fig: Loss and Accuracy for Setting 7

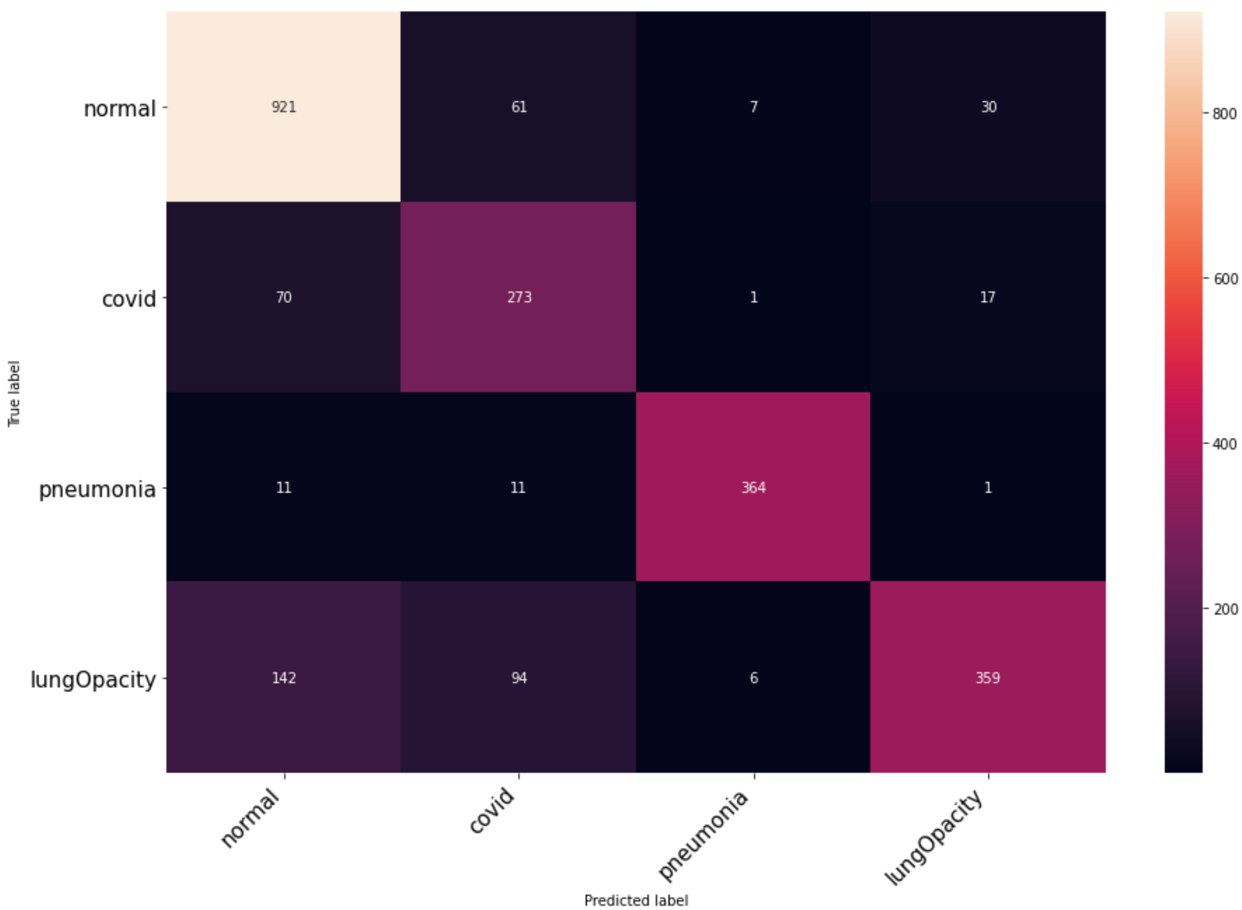


Fig: Confusion Matrix for Setting 7

● In **settings 8 and 9** we varied the Number of Hidden Layers. While previously we used 3 hidden layers, now we decreased it to 2 and 4 hidden layers respectively for these settings. We observed losses were 3.945% and 5.583% respectively. As the baseline of 3 hidden layers (from setting 5) provided **3.213%** loss, which is the lowest value, so we decided that the optimal setting for the **number of hidden layers=3**. The graphs and confusion matrices for each setting are shown as follows:

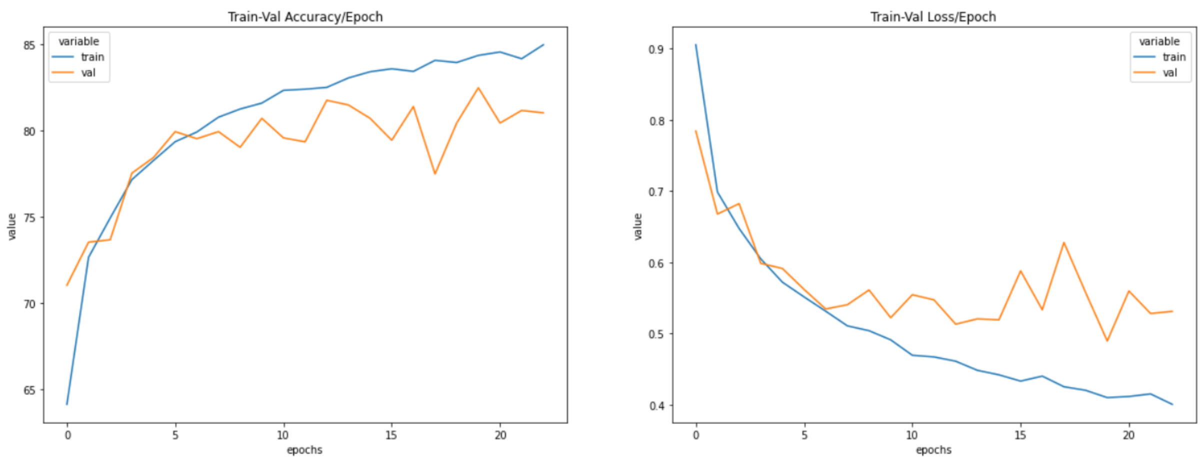


Fig: Loss and Accuracy for Setting 8

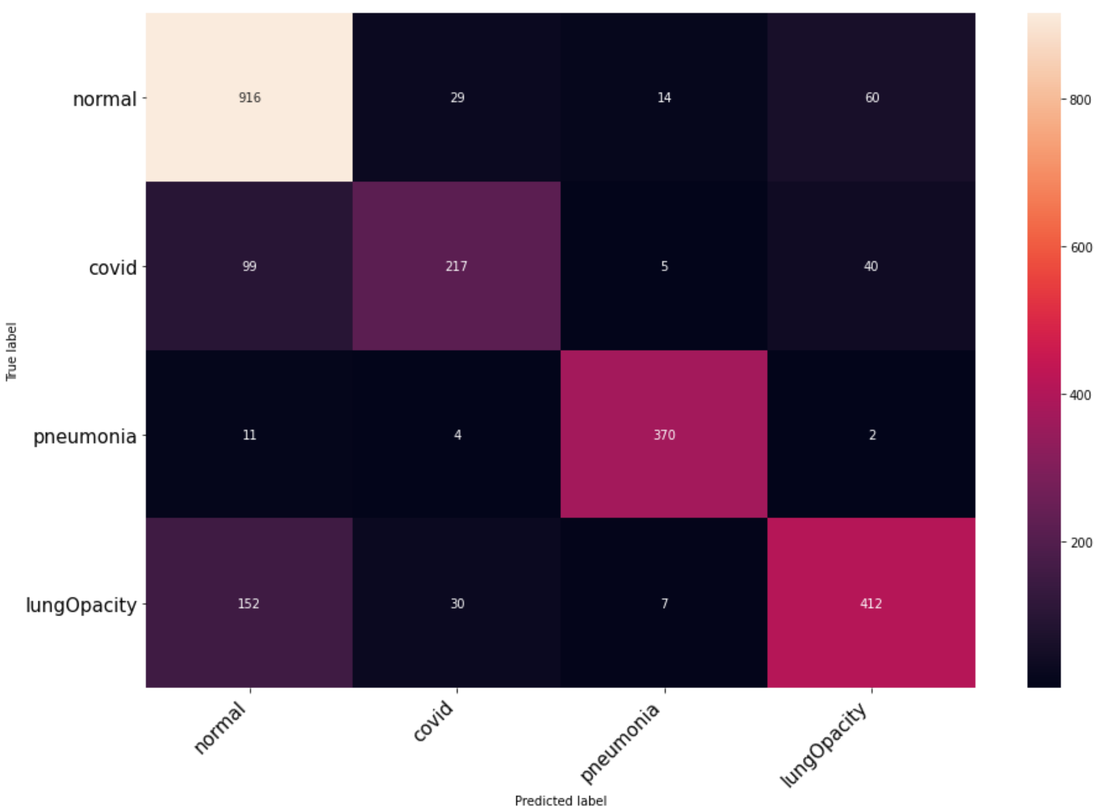


Fig: Confusion Matrix for Setting 8

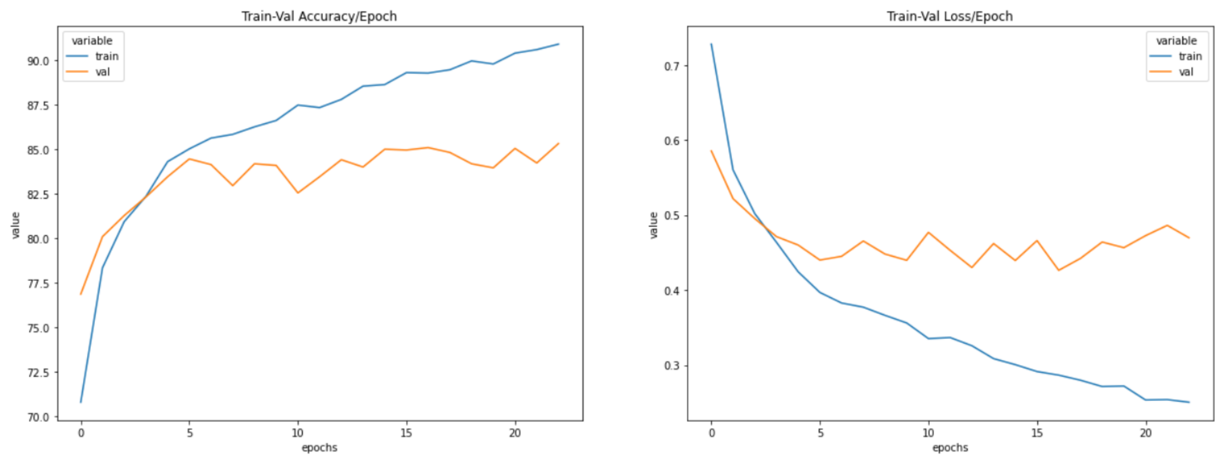


Fig: Loss and Accuracy for Setting 9

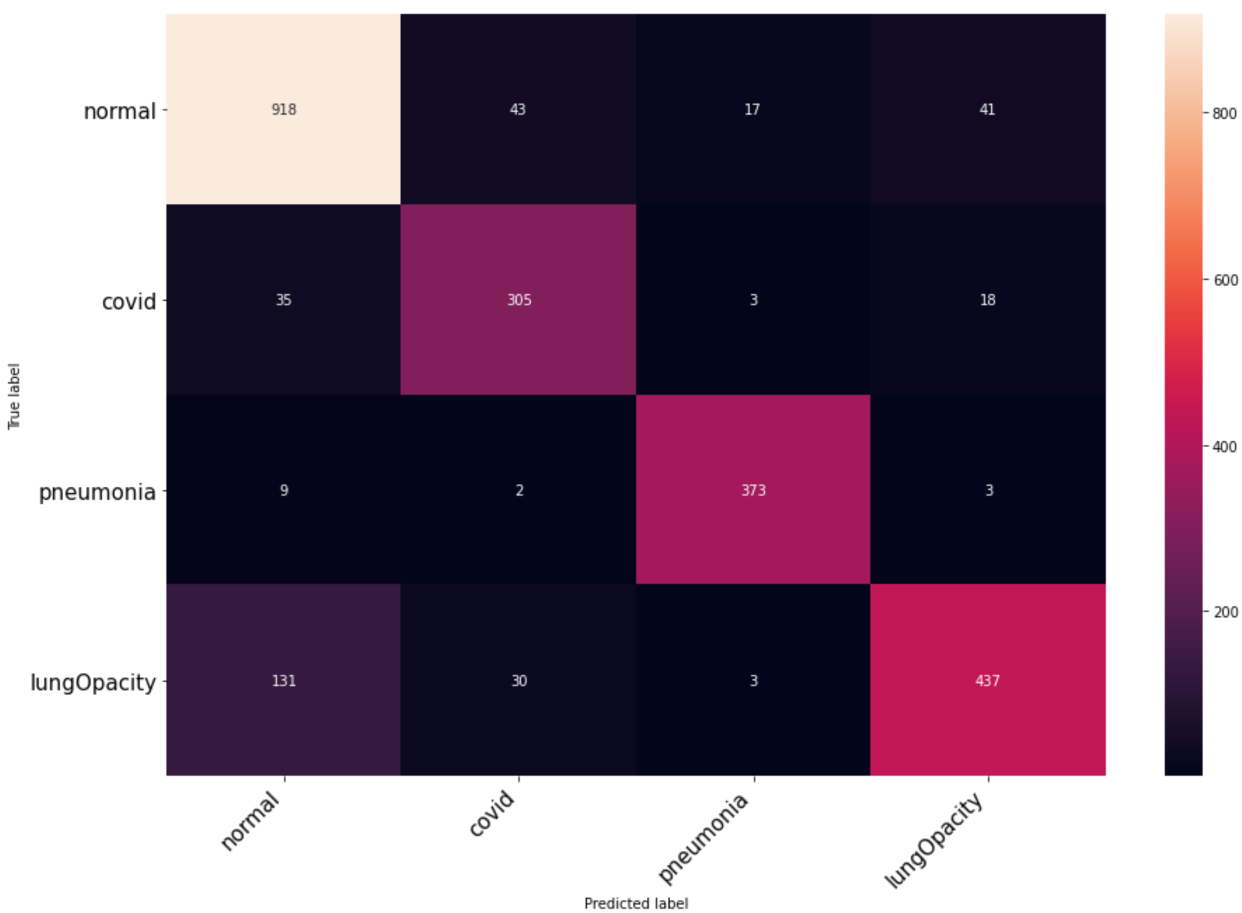


Fig: Confusion Matrix for Setting 9

- Finally, in **settings 10, 11 and 12**, we varied the activation functions used in each layer. In the previous 9 experiments, we had used ReLU activation Function. So that is our baseline. This time we varied the activation function to RReLU (all 3 layers), SELU (all 3 layers) and different activation functions in each layer (RReLU->ReLU->SELU) in the final 3 settings. Here we could observe, the losses were 5.368%, 4.618% and 3.956% respectively. While the loss observed in setting 5 (our baseline) was lower (3.213%) than that in setting 12 (3.956%), the accuracy observed was also lower (only 88.531% for Training Dataset). Thus we decided that **setting 12** was our optimal setting, where the **activation functions were RReLU->ReLU->SELU.** Here, the highest accuracy was observed to be 90.547% for the Training Dataset. The graphs and confusion matrices for each setting are shown as follows:

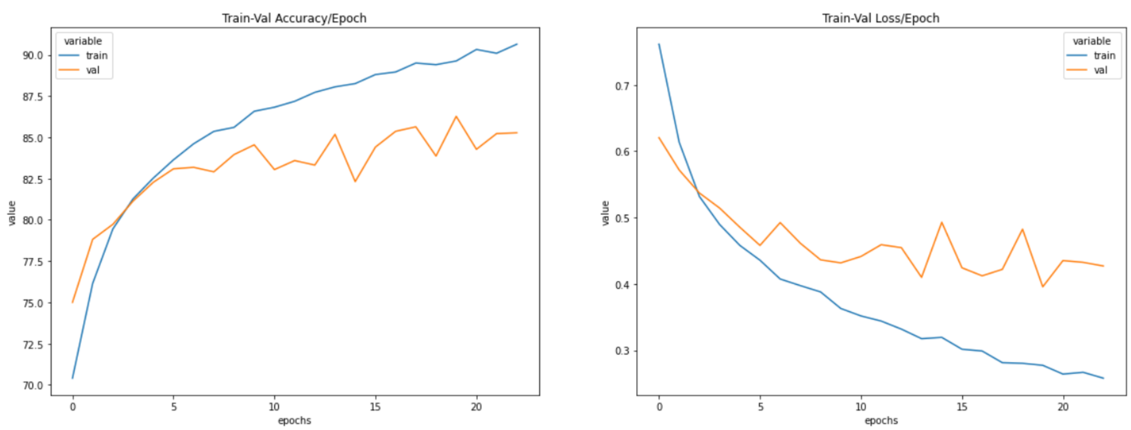


Fig: Loss and Accuracy for Setting 10

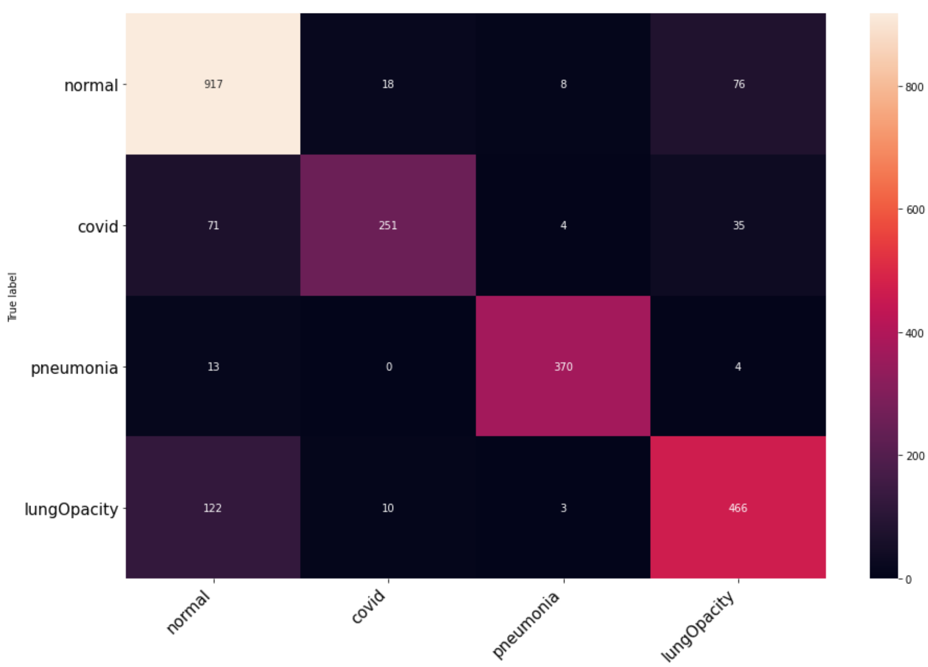


Fig: Confusion Matrix for Setting 10

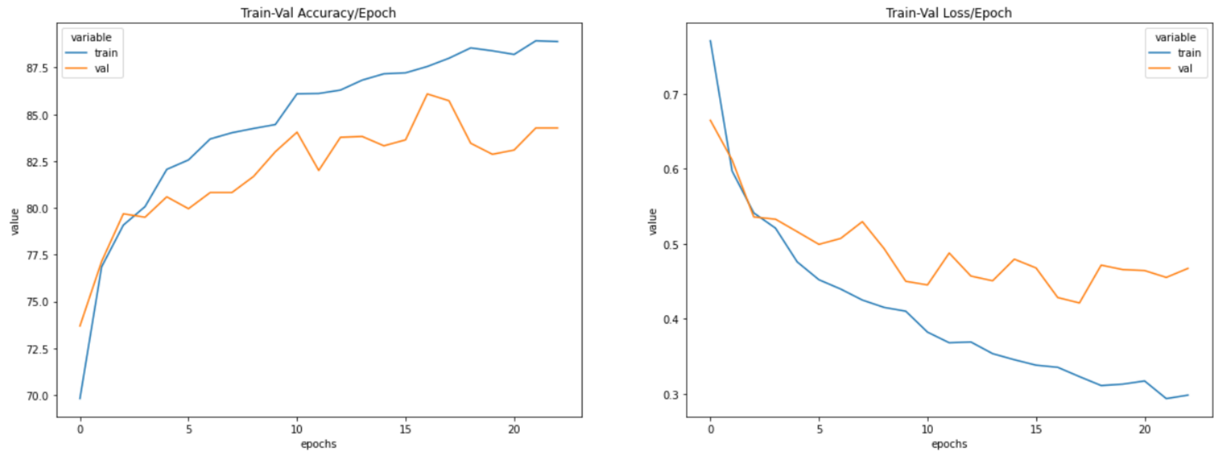


Fig: Loss and Accuracy for Setting 11

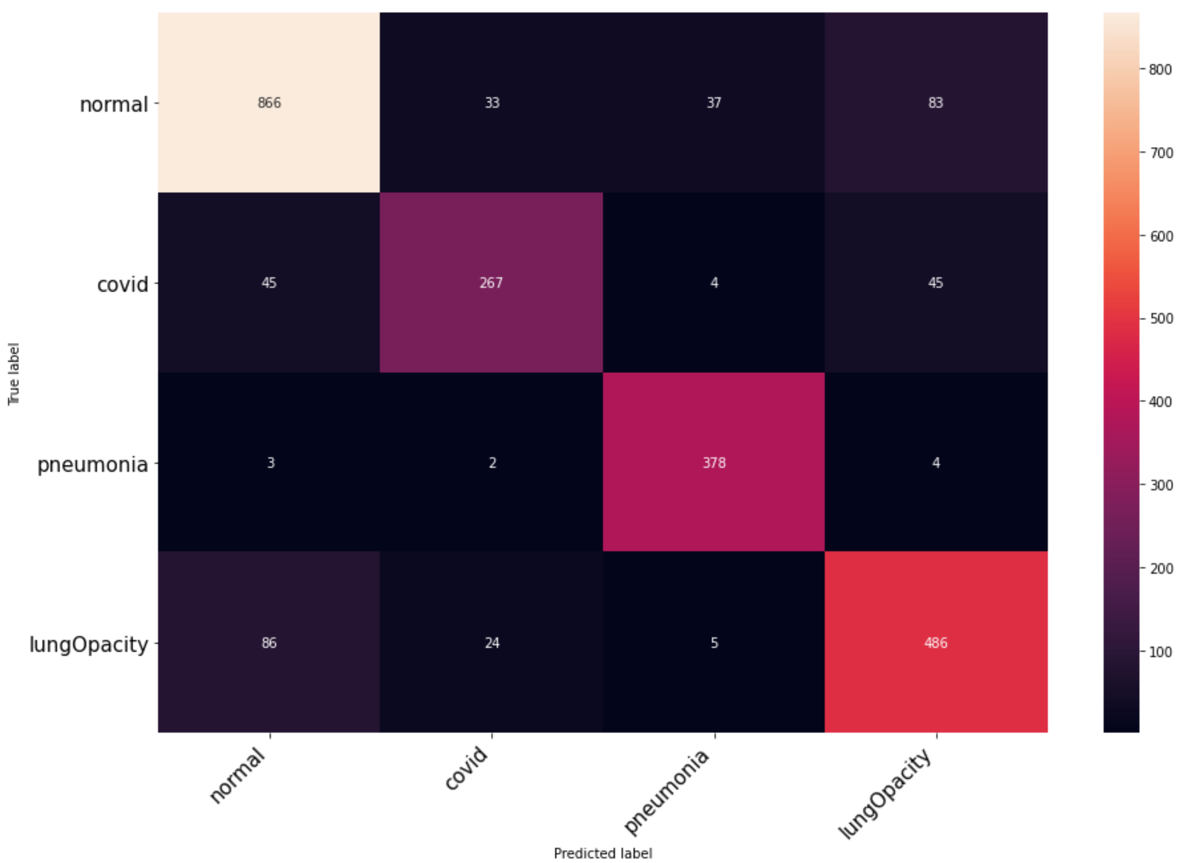


Fig: Confusion Matrix for Setting 11

The graph for loss and accuracy and confusion matrix for the final, optimum setting 12 is shown as follows:

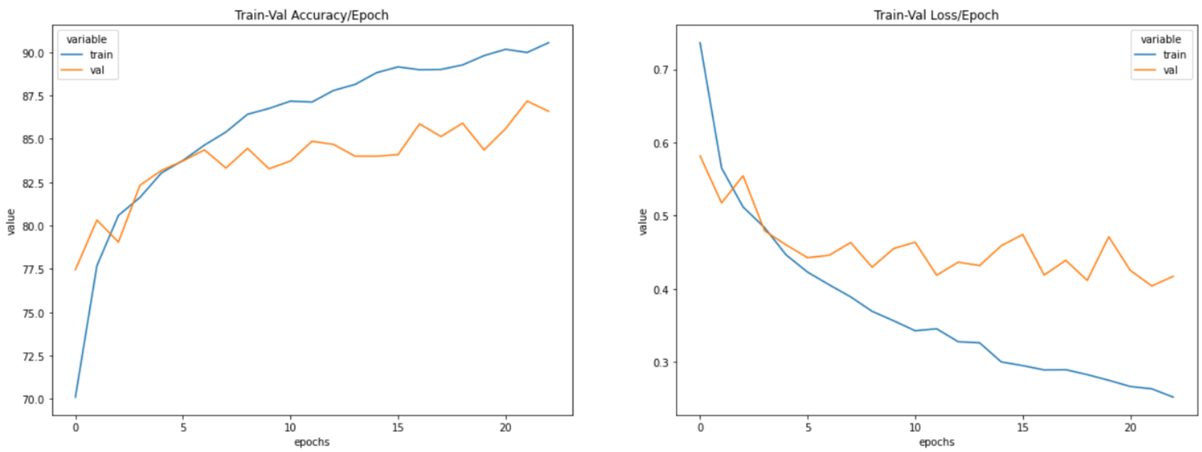


Fig: Loss and Accuracy for optimum Setting 12

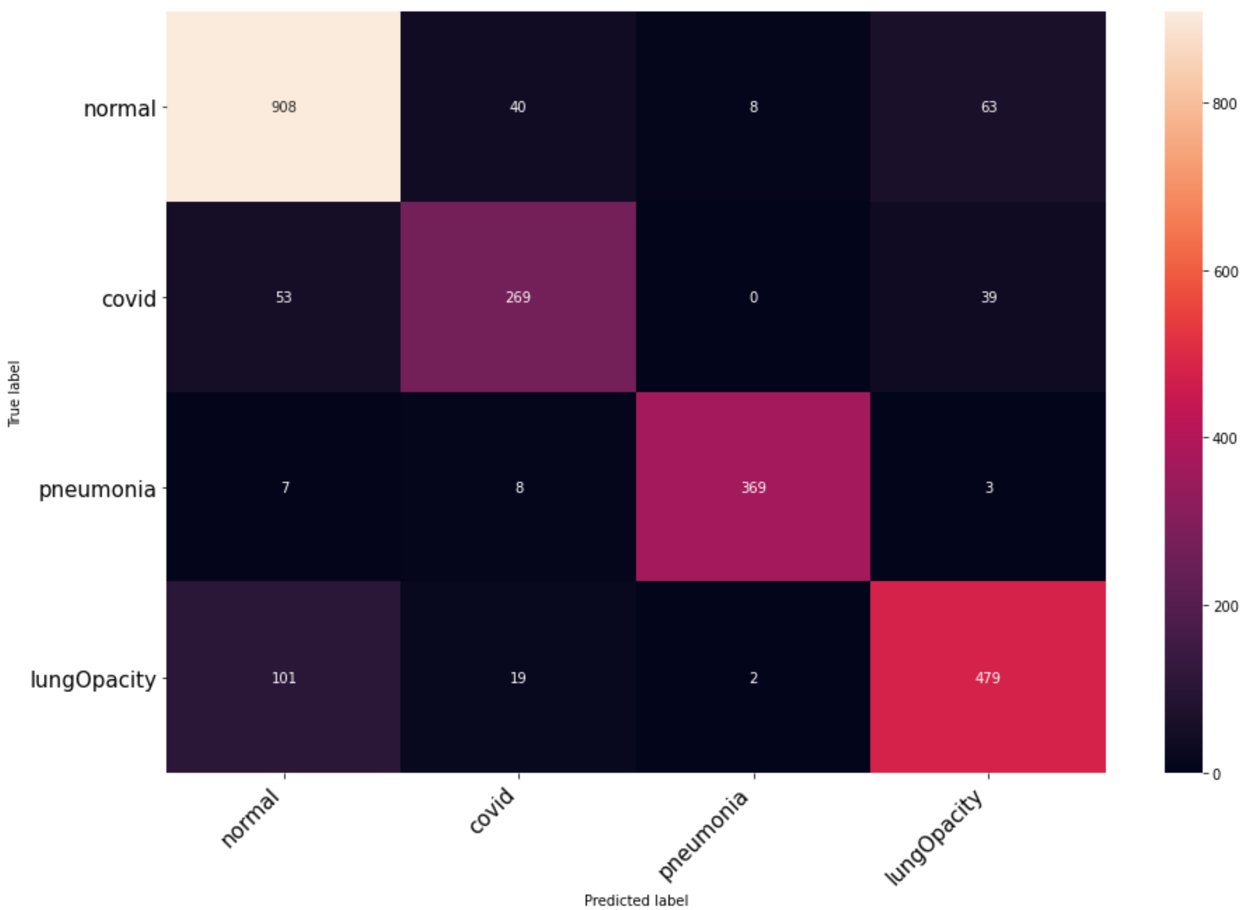


Fig: Confusion Matrix for Setting 12

Thus we can conclude that our optimal settings observed through the ablation experiments were:

- **Epoch Size**: 23
- **Batch Size**: 100.
- **Size of Hidden Layers**: 20
- **Learning Rate**: 0.001
- **Number of Hidden Layers**: 3
- **Activation Functions**: RReLU->ReLU->SELU

In our project, after we determined our hyperparameters through ablation experiments, we also calculated the accuracy that could be obtained from applying our models on the test datasets. We used separate test sets for each of the 4 classes in order to calculate these accuracies. The accuracies can be observed in Table 3:

Settings	Accuracy for Normal Lungs	Accuracy for Covid Lungs	Accuracy for Pneumonia Lungs	Accuracy for Lung Opacity (Infected Lungs)
1	87.536	71.191	94.573	81.198
2	90.873	69.806	95.865	77.204
3	89.793	66.481	95.607	76.206
4	89.303	73.961	94.832	81.364
5	89.597	66.204	94.832	78.535
6	89.008	68.421	95.348	79.034
7	90.382	75.623	94.056	59.733
8	89.892	60.11	95.607	68.552
9	90.088	84.487	96.382	72.712
10	93.915	61.495	86.563	64.226
11	84.985	73.961	97.674	80.865
12	89.106	74.515	95.348	79.7

Table 3: Accuracy for Test Sets for Normal Lungs, COVID-affected Lungs, Pneumonia-affected Lungs and Lungs with Opacity due to Infection

6. Conclusion:

In conclusion, it can be said that this project was chosen due to its high relevance with current world events. This study evaluates a neural network model for classifying 4 classes Normal, Covid, Pneumonia and lung opacity using x-ray images. These four classes were chosen to make it easy for radiologists to detect them from far in this pandemic. Additionally, the proposed model was fine-tuned by captioning different hyperparameters. Finally, the evaluation matrices and accuracies obtained by each class were illustrated. As novices, we have tried our best to solve the problem at hand with the limited tools we have learned so far. Hopefully, we can improve upon this project in the future with more advanced tools and resources so that it may have some positive impact.

References:

[1] Mukherjee H, Ghosh S, Dhar A, Obaidullah SM, Santosh KC, Roy K. Deep neural network to detect COVID-19: one architecture for both CT Scans and Chest X-rays. Applied Intelligence. 2020 Nov 6:1-3.

[2] Arias-Londoño JD, Gomez-Garcia JA, Moro-Velázquez L, Godino-Llorente JI. Artificial Intelligence applied to chest X-Ray images for the automatic detection of COVID-19. A thoughtful evaluation approach. IEEE Access. 2020 Dec 14.

[3] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

[4] https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia