Covid 19 Detection in X-Ray Images

Kellie Halladay, Jake Prichard, and Ruiying Liu

COMP 4449: Data Science Capstone

Abstract

This study examines a handful of X-ray images, some with healthy lungs, some with covid, and some with pneumonia, builds models to predict a person's covid status based on their X-ray images. These images were cropped, divided into multiple small pixels, transcribed into a workable dataset, and augmented, then run through various different models including logistic regression, decision trees, random forest, XGBoost, support vector machines, and neural networks to determine the best way to make an accurate prediction. It was found that the support vector machine was the most favorable model, yielding a recall of 90.48%, precision of 97.44% and f1_score of 93.83% on the test set. And, recall of 93.11%, precision of 62.93% and f1_score of 75.10% on a second test set, and kept the simplicity that we were aiming for in a model.

1. Problem

COVID-19, a highly contagious disease, caused a global pandemic in 2020. Chest X-ray imaging proved to be an effective tool for detecting lung diseases, including COVID-19. However, symptoms (Figure 1) [1] of COVID-19 closely resemble those of viral pneumonia and other types of pneumonia when viewed on X-rays. During the pandemic, biological methods such as Reverse Transcription Polymerase Chain Reaction (RT-PCR) and ELISA antibody kits were widely used for diagnosis, but these methods are costly and time-consuming.

Machine learning (ML) is widely used in image identification. By building ML models, we can assist medical professionals in diagnosing COVID-19-associated pneumonia more quickly and cost-effectively. This study examines X-ray images of healthy lungs, COVID-19-infected lungs, and pneumonia-infected lungs. The images were cropped, divided into small pixels, transcribed into a workable dataset, and augmented. Various models, including logistic regression (LR), decision trees(DT), random forests(RF), XGBoost(XGB), support vector machines (SVM), and neural networks(NN), were used to predict COVID-19 status based on these X-ray images. The primary challenge of this study is to build a model that can distinguish between covid-19 pneumonia and other viral and bacterial pneumonia.

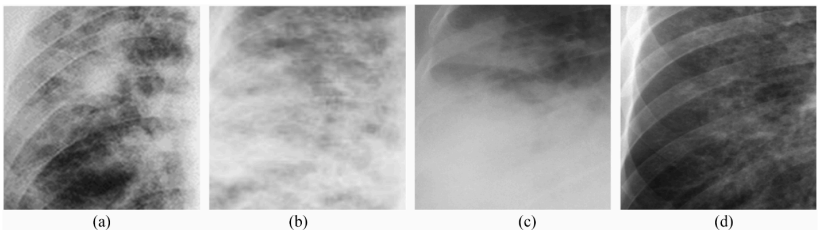

Figure 1 COVID-19 infected Chest X-ray images.
(a) patchy Ground-glass opacities, (b) Reticular opacities,
(c) Pulmonary consolidation, (d) Mild opacities.

2. Data Sources

-We used a collection of 603 X-ray images from data.mendeley.com - 221 images with Covid-19, 148 images with pneumonia, and 234 images without any illness [2]. In order to analyze this set and create models, we started by resizing all the images to be 224 x 224 pixels, and assign each pixel a number based on the shade in the image. We then created a flag to indicate whether or not the picture was specified as Covid-19. Here is an overview of the variables used in the model, along with the type, range and encoding(Table 1):

Table 1 Data information

Input/Output	Name	Description	Type	Categorical/ Continuous	Range	Encoding
Input	feature0 - feature50175	Each feature# indicates the shade for a certain pixel in image	float32	Continuous	0 - 1	Numeric value scaled down to decimal from 0 - 1
Output	binary_label	Covid indicator	float32	Categorical	0 - 1	0 - not Covid, 1 - Covid

3. Exploratory Data Analysis/Data Preprocessing

We originally started with a dataset of 603 chest X-ray images of patients. The data consisted of 234 normal chest X-ray, 220 X-ray of patients who had pneumonia caused by covid-19, and another 148 X-ray of patients who had other viral or bacterial pneumonia that was not caused by covid-19. We created our label which is a binary variable where 1 represented the

presence of covid-19 related pneumonia and 0 represented not having covid-19 related pneumonia. From the images we extracted 50176 features, each of which representing the grayscale intensity for a segment of a 224 x 224 image. The values ranged from 0 to 255, 0 representing complete blackness and 255 representing complete whiteness. We normalized these values to be on the scale 0 to 1.

Once we had translated the images into a workable dataset, we analyzed the data accordingly. First, checked for null values and found that every row and column of the dataset was populated and there were no missing values. Next, we computed the range of values for each feature and found that the minimum range for a given feature was 0.66, which implies that every feature has a decent spread of values. Then, we examined the correlation between the variables and found that the closer the variables were to each other, the higher the correlation. The variables have an average correlation of 0.97 with others within 5 variables of them, 0.84 with others within 25 variables of them, and 0.5 with others within 125 variables of them.

The dataset provided us with many challenges. Firstly, we were dealing with a dataset which had many more features than it did data points. Second, we had a slight imbalance in the class distribution. The dataset contained about 36% covid and 64% non covid patients. Lastly, and perhaps most importantly we were stuck with the task of distinguishing between two things that more or less look exactly the same and in essence are the same thing. Covid-19 pneumonia and other types of pneumonia would only have subtle differences in the X-ray.

4. Model training on original data set

4.1 Initial models

We then began fitting initial models to see how well these values could predict the presence of covid-19 in the images without any additional feature extraction or model tuning. We set aside 20% of the data to use for testing. We stratified the data so that both the training and testing sets would have the same class distribution balance. We chose to fit 4 types of models, a logistic regression, a support vector machine, a neural network, and a random forest. Surprisingly, the models were able to distinguish between the two types of pneumonia in addition to distinguishing between covid-19 and normal X-rays..

When running the initial models, we received the following results(Table 2):

Table 2 Initial models

	Accuracy	Recall	Precision	f1_score
Logistic Regression	95.45%	95.45%	93.33%	94.38%
Support Vector Machine	96.69%	97.72%	93.47%	95.55%
Neural Network	95.04%	93.18%	93.18%	93.18%

4.2 Fine Tuning Models

After running the initial model, we decided to do some tuning to improve the model. The hyperparameters, hyperparameter ranges and optimal values for fine-tuned Logistic LR, DT, RF, XGB and SVM model were shown in Table 3 and the evaluation results were shown in Table 4:

Table 3 Hyperparameters used in fine_tuning LR, DT, RF, XGB, SVM models

Model name	Hyperparameter		Hyperparameter Range	Optimal Value
Logistic Regression	C		0.001, 0.01, 0.1, 1, 10, 100, 1000	1
	solver		lbfgs', 'liblinear ', 'saga'	'liblinear '
Decision Tree	max_depth		10, 20, 30, None	10
	min_samples_split		2, 10, 20	2
	min_samples_leaf		1, 5, 10	1
	max_features		sqrt', 'log2', None	'sqrt'
	criterion		gini', 'entropy'	'gini'
Random Forest	n_estimators		50, 100	100
	max_depth		5,10, 20	20
	max_features		0.2, default	0.2
Xgboost	n_estimators		50, 100	100
	max_depth		5, 10	5
	learning_rate		0.1, 0.2	0.2
	subsample		0.7, 0.8, 0.9	0.8
	colsample_bytree		0.7, 0.8, 0.9	0.8
Supper Vector Machine	rbf'	C	0.1, 1, 10, 100, 1000	10
		gamma	1, 0.1, 0.01, 0.001, 0.0001	0.001
	linear'	C	0.1, 1, 10, 100, 1000	N/A

Table 4 Evaluation on fine_tuning LR, DT, RF, XGB, SVM models

Name	Accuracy	Precision	Recall	F1_score
Logistic Regression	95.03%	95.95%	92.21%	94.04%
Decision Tree	83.98%	88.71%	71.43%	79.14%
Random Forest	93.92%	93.42%	92.21%	92.81%
Xgboost	95.58%	98.59%	90.91%	94.59%
Support vector machine	95.03%	93.59%	94.81%	94.19%

Fine-tuning the NN model was also included. First, we added a 2D convolutional layer, and the data was reshaped and run through this new layer rather than strictly dense layers. Next,

the hyperparameters of the Adam optimizer were tested to determine the best combination for running and improving the model efficiently. We found the best parameters to be a learning rate of 0.005, beta 1 of 0.999, and beta 2 of 0.999. Then, since the model was potentially underfitting the data, we tried adding more layers with varying amounts of neurons, and found that the optimal solution was to add 5 more layers each with 128 neurons. Running this updated model gave us a precision of 0.96, recall of 0.98, and f1 score of 0.97. Here is a visual of the recall over the span of 15 epochs (Figure 2):

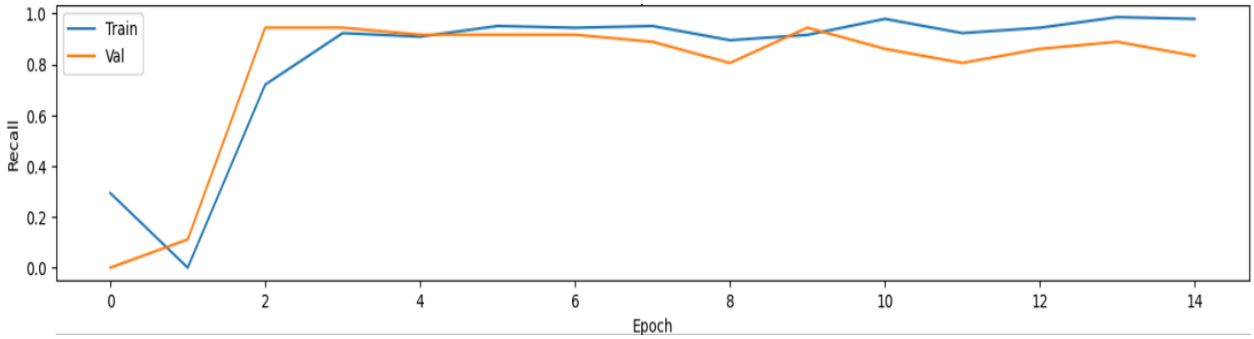

Figure 2 Visualization of the recall over the span of 15 epochs on best Neural Network model

5 Image processing

We were suspicious that our models were performing so well without any preprocessing or tuning. Because of this we decided to run the initial models on other data which was gathered from the COVID-19 Radiography Database [3]. From this we sampled a dataset of images that was relatively balanced between the two classes of interest. The breakdown was roughly 25% normal , 25% non-pneumonia , and 50% covid-19 pneumonia. We tested our models on these images and found that models did not perform nearly as well as they did on our original small test set. Specifically, we found that our initial logistic regression model was producing many false positives. The model performed okay with respect to predicting the positive class, with 89% recall. However, it could only predict the absence of covid-19 at a rate of around 60% (Figure 3). Interestingly, the vast majority of the incorrectly classified images were of normal x-rays. In other words, the model could more easily distinguish between covid-19 pneumonia and non-covid-19 pneumonia, than it could between covid-19 and healthy lungs.

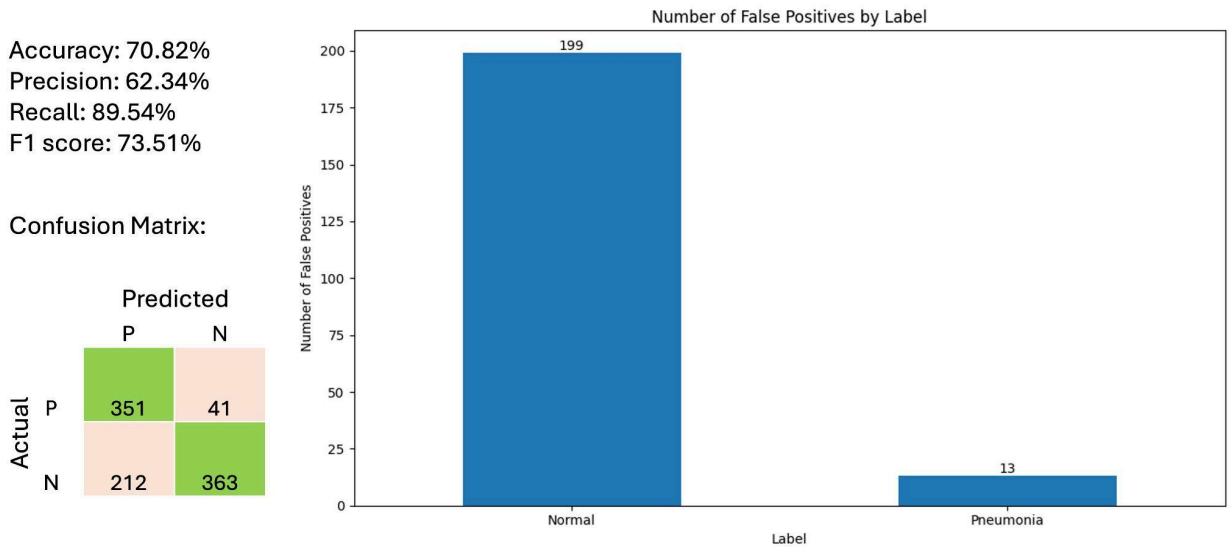

Figure 3 Model Performance on Larger Testing Set

5.1 Outlier detection

Because of the discrepancy between the models performance on the small test set versus the larger one, we decided to look for anomalies in the data that might help explain why the models were performing so well on both our training and test sets, but not generalizing well to outside data. The first thing that was done was to use an isolation forest to find outliers. The isolation forest revealed many things about the images in the data. We found images that were taken from a side angle (Figure 4A). Other images had annotations done by radiologists, these included mostly arrows pointing at the opacities that represented covid-19(Figure 4B). There were also a few images that were from computed tomography (CT) scans(Figure 4C). This seemed like a problem because our models could potentially be interpreting these attributes as important in some way, or they could possibly serve as hidden labels. The problem became more intensified when we realized that each of these types of outliers were present only in the covid-19 cases. There were no x-rays of these types that belonged to the negative class (normal/pneumonia). We decided to remove all of the instances of these three types of images from the dataset.

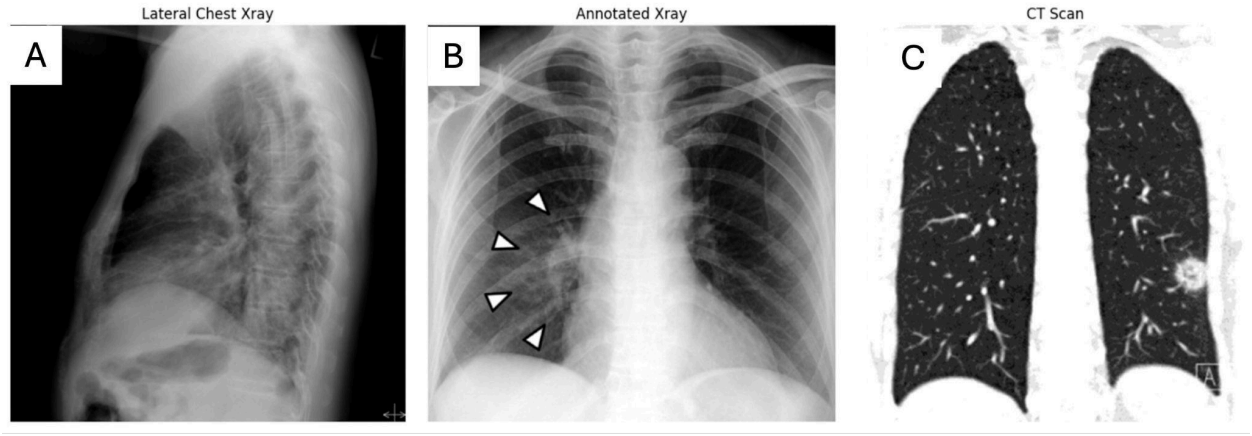

Figure 4 Anomalies in the data set

5.2 Other Imbalances in the Images

We found two other imbalances in the data set. First, we found that most or all of the x-rays of female patients were of the covid class. Secondly, we found that most or all of the x-rays where medical equipment is present are of the covid class. There were too many of these instances to consider removing them.

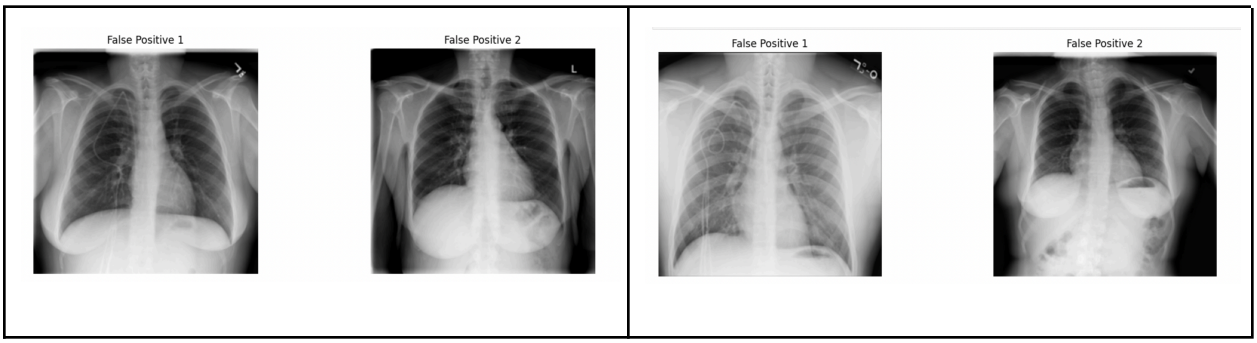

Figure 5 A look at false positives

Visual inspection of the false positives (Figure 5) from the Logistic Regression model reveals that a lot of them are women. Without knowing the gender balance of the dataset, it's hard to say for sure if this is significant. However, in rerunning plots of the false positives, it seemed like there was a high frequency of female x-rays. Since our models don't have any training examples of female patients that are from normal or pneumonia x-rays, it makes sense that they are frequently classifying them as covid-19. Furthermore, it possibly explains the discrepancy between our test set and the outside test set. In our test, all of the female x-rays are actually covid-19 cases, whereas in the outside test set female patients are both covid and

non-covid cases. Therefore, the models predictions are correct within our test set, but incorrect half the time in the outside test set, presuming equal distribution across gender in the outside test set.

5.3 Correlation Analysis

Since linear models such as logistic regression were performing so well on our data, we decided to look at the correlation coefficients between each feature and our target variable. We took the absolute value correlation coefficient between each feature and the target variable. We then plotted the top 7000 correlation coefficients, overlaying the image at the location that the feature represents. The figure (Figure 6) below shows the correlation coefficients for the original 224 x 224 data after outlier removal.

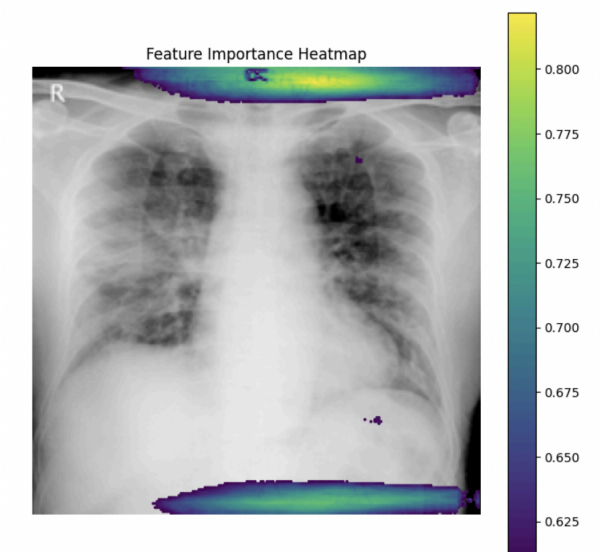

Figure 6 Feature importance heatmap for the original 224 x 224 data after outlier removal

Note that most of the areas that are highly correlated with the target, are areas that are actually irrelevant to covid-19 infection. This suggests that models (especially logistic regression) are probably using irrelevant features to make predictions. This means that further processing was needed.

5.4 Cropping to the Lungs

We decided to crop the images to the lungs to make sure that we were filtering out irrelevant features. This time we read in the images at a resolution of 256 x 256. After cropping the images, the correlation heatmap looked like this (Figure 7):

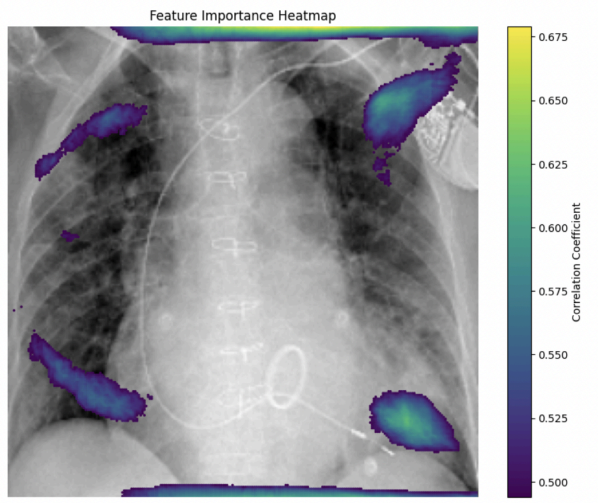

Figure 7 Feature importance heatmap for the 256 x 256 data after cropping

This was a favorable result because now we had data where the most correlated features represent areas of the lungs.

We also decided to plot the same heatmap when non-covid pneumonia was treated as the target variable. The plot below shows the top 7000 absolute value correlation coefficients between the features and non-covid pneumonia (Figure 8).

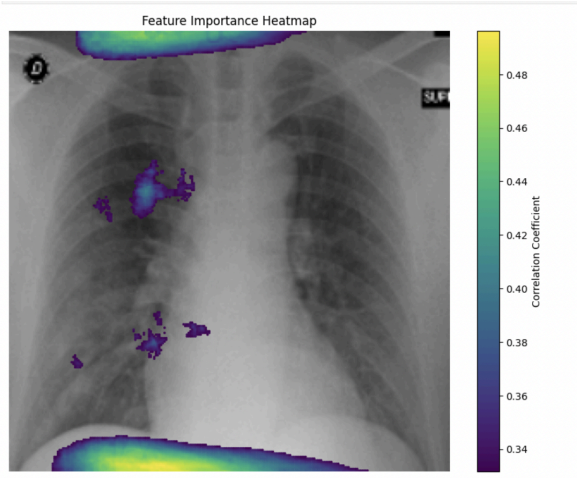

Figure 8 Top 7000 absolute value correlation coefficients between the features and non-covid pneumonia

Like in other heatmaps we still see the existence of high correlation coefficients in irrelevant areas. However, the heatmap also reveals that non-covid pneumonia might not settle in the same place as covid-19 pneumonia. The two graphs show that covid-19 is more correlated

with features that represent the periphery of the chest, whereas non-covid pneumonia is correlated with features that represent the inner part of the chest. Furthermore, the features associated with covid-19 seem to be at the top and bottom of the chest, whereas non-covid pneumonia related features are more central. This is consistent with what we were visually observing. In fact, if you take a look at the figure above, it is actually a covid-19 patient with correlation coefficients for non-covid pneumonia. Notice that the white opacities are not in the areas where the correlation coefficients are high for non-covid pneumonia.

5.5 Removing more Noise

To reduce the effect of irrelevant features, we removed roughly 15,000 features from the data (Figure 9).

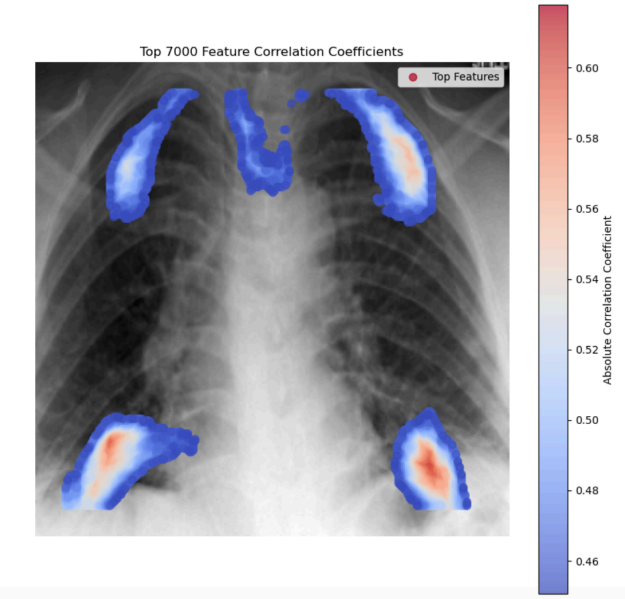

Figure 9 Correlation coefficients after removing features

5.6 Data Augmentation

Next we decided to augment the data by adding transformations of our original images to the training set. Because of the vertical symmetry seen in the correlation (with covid-19) heat map we decided to add a vertically flipped version of each training example to the data set. This would mean, for example, that if a training example is showing covid-19 in the upper periphery of the chest we would also have an example where they are showing covid-19 in the lower periphery of the chest. Because covid-19 is associated with both the top and the bottom of the

chest, this seemed reasonable. This is illustrated in figure 10A and an example of one of our training images along with the augmented image is shown in Figure 10B.

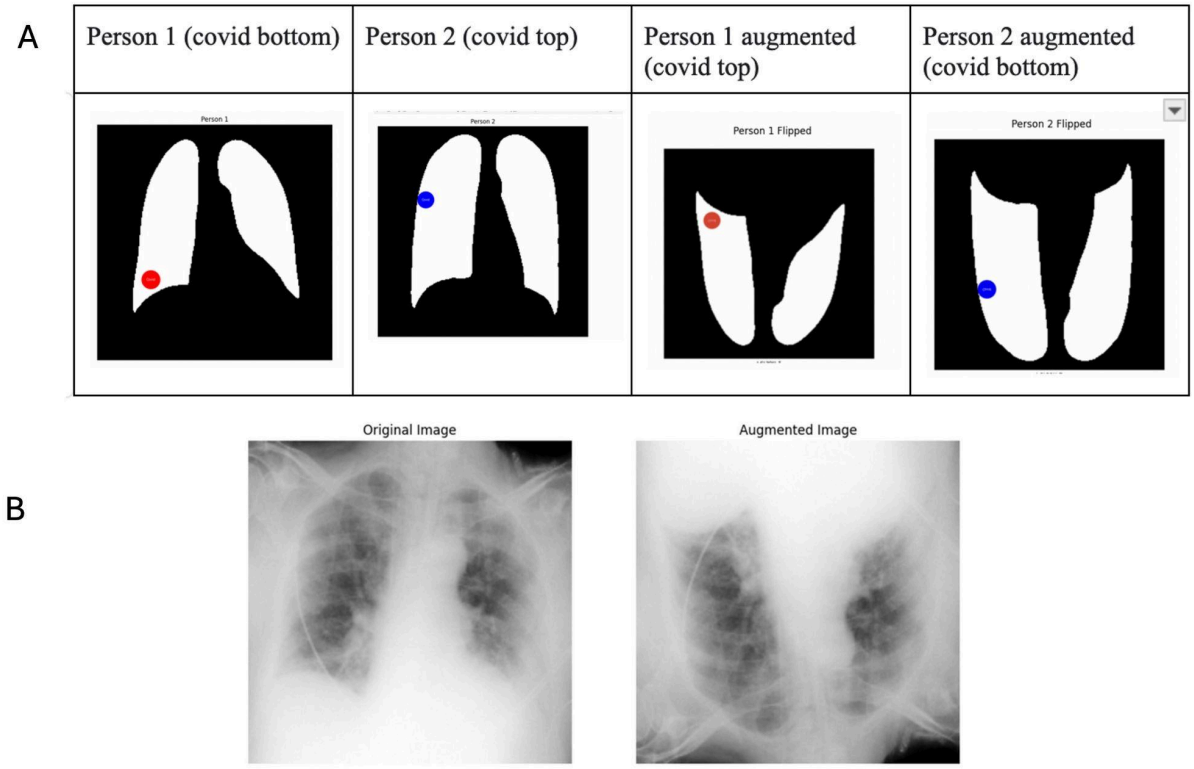

Figure 10 Illustration of image augmentation

6. Model training after image processing

The logistic regression model was used as a baseline before model tuning (Table 5) . However, evaluations on small test sets are finely good while the scores went down on large test sets.

Table 5 Evaluation on baseline model

	Small Test set	Large Test Set
Accuracy	94.69%	70.93%
Precision	94.12%	62.46%
Recall	88.89%	89.54%
F1_score	92.43%	73.58%

To improve scores on the large test while maintaining performance on the small test set, different types of models and grid search in hyperparameters were conducted to find the best model (Tabel 6, Table 7 & Table 8).

Table 6 Hyperparameters used in fine_tuning models

Model name		Hyperparameter	Hyperparameter Range	Optimal Value
Logistic Regression		C	0.001, 0.01, 0.1, 1, 10, 100, 1000	0.001
		solver	lbfgs', 'liblinear ', 'saga'	'lbfgs'
Decision Tree		max_depth	10, 20, 30, None	10
		min_samples_split	2, 10, 20	10
		min_samples_leaf	1, 5, 10	1
		max_features	sqrt', 'log2', None	sqrt'
		criterion	gini', 'entropy'	'entropy'
Random Forest		n_estimators	50, 100	100
		max_depth	5,10, 20	10
		max_features	0.2, 'sqrt'	'sqrt'
Xgboost		n_estimators	50, 100	50
		max_depth	5, 10	5
		learning_rate	0.1, 0.2	0.1
		subsample	0.7, 0.8, 0.9	0.8
		colsample_bytree	0.7, 0.8, 0.9	0.8
Supper Vector Machine	rbf'	C	0.1, 0.5, 1, 10, 100, 1000	0.5
		gamma	1, 0.1, 0.01, 0.001, 0.0001	0.0006
	linear'	C	0.1, 1, 10, 100, 1000	N/A
Neural Network		learning_rate	0.0005, 0.001, 0.005, 0.01	0.005
		beta1	0.9, 0.99, 0.999	0.999
		beta2	0.9, 0.99, 0.999	0.999
		num_layers	1, 2, 5, 10	5
		num_neurons	64, 128	128
		activation	ReLU, leakyReLU	ReLU

Table 7 Evaluation scores on Top 5 models on small test data

Model	Accuracy	Precision	Recall	F1_score
Logistic Regression	92.04%	92.31%	85.71%	88.89%
Random Forest	87.18%	94.44%	80.95%	87.18%
Xgboost	92.04%	94.59%	83.33%	88.61%
Supper Vector Machine	95.58%	97.44%	90.48%	93.83%
Neural Network	91.15%	97.06%	78.57%	86.84%

Table 8 Evaluation scores on Top 5 models on large test data

Model	Accuracy	Precision	Recall	F1_score
Logistic Regression	68.05%	61.08%	80.87%	69.59%
Random Forest	70.36%	62.34%	86.99%	72.63%
Xgboost	67.59%	60.95%	78.83%	68.74%
Supper Vector Machine	72.09%	62.93%	93.11%	75.10%
Neural Network	67.94%	61.59%	77.30%	68.55%

To summarize, an SVM model with parameters as ‘rbf’ kernel, C value equal to 0.5, and gamma equal to 0.0006 works fairly well on the test set and has the highest scores when running on large test data as well.

7. Conclusion

After creating and training many models and tweaking our data in multiple ways, we ultimately decided that the best model was the support vector machine. We received high values when running the model on the test set with a a recall of 90.48%, precision of 97.44% and f1_score of 93.83%, and we also received the highest values when running this model on our outside data that we found from a different source, with an recall of 93.11%, precision of 62.93% and f1_score of 75.10%. Between both testing data sets, the SVM model was able to correctly classify 412 (93.11%) of the 443 possible covid-cases. Of the 477 non-covid cases (both pneumonia and non-infected lungs) the SVM correctly classified 251 (56.15%) of them correctly. The SVM is also a relatively simple model, and we preferred this type of model over something more complex, such as a neural network. With the SVM model, we are able to accurately identify covid within an X-ray image most of the time. However, the defect of this model is in correctly detecting the true negative which gave an overall low scores on precision and F1 score when evaluating on the large test data.

To build a model that we can be fully confident in, additional steps are needed. Firstly, a larger, more balanced dataset is needed. We need images that are from a wider array of patients, including patients of various genders and ages. Secondly, we need better techniques for isolating the lungs and neutralizing the effects of other elements within the x-ray, including the bone, diaphragm, spine, and heart. Lastly, Through correlation analysis and other EDA, we were able to get a good sense of what the important features are. We need to continue down this path and gain a better understanding of these features and their relationships to each other and the target variable.

8. Future Suggestions

The next steps would be to group the images differently so that all covid and pneumonia images are in the positive class, and all normal images are in the negative class, then try creating

a model to distinguish between all sick lung images vs healthy lung images. We did get a chance to look into this a little bit. In this case, we have 522 sick x ray images and 382 healthy x ray images.

9. References

[1] T. Chandra, K. Verma, B. Singh, D Jain, S. Netam, "Coronavirus disease (COVID-19) detection in chest X-ray images using majority voting based classifier ensemble," *Expert Systems with Applications*, Vol. 165, 1, Mar. 2021

[2] M. Shams, O. Elzeki, and M. Abd Elfattah, "Chest X-ray images with three classes: COVID-19, Normal, and Pneumonia," *data.mendeley.com*, vol. 1, Jun. 2020, doi: https://doi.org/10.17632/fvk7h5dg2p.1.

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