**Media Engineering and Technology Faculty**
**German University in Cairo**

# COVID-19 Diagnosis from Lung Ultrasound using Machine Learning Methods

**Bachelor Thesis**

Author: Bassel Amgad Redaa Sharaf

Supervisors: Associate Professor Dr.Seif Eldawlatly

Submission Date: 1 August, 2021

**Media Engineering and Technology Faculty**
**German University in Cairo**

# COVID-19 Diagnosis from Lung Ultrasound using Machine Learning Methods

**Bachelor Thesis**

Author: Bassel Amgad Redaa Sharaf

Supervisors: Associate Professor Dr.Seif Eldawlatly

Submission Date: 1 August, 2021

This is to certify that:

(i) The thesis comprises only my original work toward the Bachelor Degree

(ii) Due acknowledgement has been made in the text to all other material used

---

Bassel Amgad Redaa Sharaf
1 August, 2021

# Acknowledgments

Alhamdulillah, I thank and praise Allah Subhanahu wa ta'ala for giving me the strength, guidance and courage to finish this thesis.

First and foremost, I offer my sincerest gratitude to my supervisor, Assoc. Prof. Dr. Seif El-Dawlatly, who has helped and guided me throughout my thesis with his patience and expertise, while also giving me the space to explore in different directions. Furthermore, special thanks to my family, who have always surrounded me with a cheerful and positive environment, I would have not been able to pursue my studies without their infinite aid and support. Finally, I would like to thank my colleague Moustafa Sherif, for always being there whenever I am stuck, and aiding me with new ideas and concepts that have impacted my thesis.

# Abstract

Due to the rise of the Covid-19 pandemic, the safety of healthcare workers and patients hinges on safe, fast and highly sensitive diagnostic tools. Although, ML has shown success in medical imaging, existing studies focus on Covid-19 diagnostics using Deep Learning (DL) with X-ray and Computed Tomography (CT) scans. In this study we aim to explore different image processing and Machine Learning (ML) methods on Lung Ultra Sound (LUS), to aid doctors with the diagnosis of Covid-19 patients. We chose LUS since it is faster, cheaper and more available in rural areas compared to CT and X-ray. We used the largest publicly available dataset containing LUS images and videos of Covid, pneumonia and healthy patients that has been collected from different resources. We tried out two different frame level approaches, one that extracted five frames per patient video and the other used fifteen frames per patient video. We will use this dataset to experiment with different ML models including Support Vector Machines (SVM), Logistic Regression (LR), Decision Trees (DT) and Random Forest Classifier (RFC). We also created a Convolutional neural network (CNN) model to compare it with our previous models. Furthermore, we will experiment with different data preprocessing techniques that might aid with pattern recognition and increasing the ML models accuracy like histogram equalization, standardization, Principle Component Analysis (PCA) and Synthetic Minority Oversampling Technique (SMOTE). Lastly, we created a simple application that diagnoses LUS videos with our CNN model, and shows the frame results with visual representation of why the model has taken a certain prediction with he aid of Gradient-Weighted Class Activation Maping (Grad-CAM).

# Contents

**Acknowledgments** **V**

**Contents** **X**

**1 Introduction** **1**
1.1 Motivation . . . 1
1.2 Objective . . . 1
1.3 Contributions . . . 1
1.4 Thesis Outline . . . 2

**2 Background** **3**
2.1 Covid . . . 3
2.1.1 What is Covid-19? . . . 3
2.1.2 Covid Testing Methods . . . 3
2.1.3 Lung Ultrasound . . . 4
2.2 Data Preprocessing . . . 4
2.2.1 Histogram Equalization . . . 4
2.2.2 Standardization . . . 5
2.2.3 Principal Component Analysis . . . 6
2.2.4 Oversampling . . . 7
2.3 Machine Learning . . . 7
2.3.1 What is Machine Learning? . . . 7
2.3.2 Support Vector Machines . . . 9
2.3.3 Logistic Regression . . . 9
2.3.4 Decision Trees . . . 10
2.3.5 Random Forest . . . 11
2.4 Deep Learning . . . 12
2.4.1 What is deep learning? . . . 13
2.4.2 Neural Networks . . . 13
2.4.3 Convolutional Neural Network . . . 15
2.4.4 Model Interpretation . . . 17
2.4.5 Grad-CAM . . . 18
2.5 Literature Review . . . 18

**3 Approach and Methodology** **21**
3.1 Approach Overview . . . 21
3.2 Dataset Description . . . 21
3.3 Methodology . . . 21
3.3.1 Data Processing . . . 21
3.3.2 Classifiers Overview . . . 24
3.3.3 GUI . . . 27
3.4 Evaluation Metrics . . . 28

**4 Results** **29**
4.1 Five Frames Per Video . . . 29
4.1.1 Frame Level . . . 29
4.1.2 Patient Level . . . 31
4.2 Fifteen Frames Per Video . . . 33
4.2.1 Frame Level . . . 33
4.2.2 Patient Level . . . 34
4.3 Results discussion . . . 35
4.4 GUI . . . 36

**5 Conclusion** **39**

**6 Future Work** **41**

**Appendix** **42**

**A Lists** **43**
List of Abbreviations . . . 43
List of Figures . . . 46
List of Tables . . . 47

**References** **50**

# Chapter 1

# Introduction

## 1.1 Motivation

4.5 million is the current estimate of the number of deaths caused worldwide from Covid-19 [1]. Some of the main symptoms of Covid-19 include fever, dry cough, and fatigue. However, some patients did not exhibit any respiratory symptoms but had some neurological symptoms such as headache, languidness, unstable walking, and malaise [2]. Although CT scans currently represent the leading technique to identify respiratory infection related to COVID-19, Lung Ultrasound has recently emerged as another method that could be used for the same task. Furthermore, ML have seen a rise of use in medical imaging. However, the use of ML methods with LUS has not been vastly explored compared to methods like CT scans can X-ray. We hope that this study could provide an additional input that will help physicians when diagnosing COVID-19.

## 1.2 Objective

The objective of this project is to use different image processing and machine learning methods to recognize COVID-19 patterns in lung ultrasound images, and see which method would have the highest success rate in identifying Covid-19 patients.

## 1.3 Contributions

Different machine learning models are experimented with like support vector machines, logistic regression, decision trees and random forest classifiers. In addition, each model was trained using different preprocessing methods including histogram equalization, principle component analysis. Oversampling was preformed to balance the different classes. Overall, an accuracy of 92% was achieved with our SVM model on the frame level test set, and 95% of the patients were correctly diagnosed. Moreover, our application with grad-cam was successful in showing the different patterns in the lung ultrasound video that have the largest weights in the model's prediction.

## 1.4 Thesis Outline

The rest of the thesis consists of five main chapters:

(i) **Background:** This chapter's goal is to give the reader the needed background in order to understand the different methods used in this thesis. We go over the history and origin of Covid-19, and explain the different diagnostic tools that are used to identify Covid-19, like CT scans. Moreover, we thoroughly discuss why LUS is starting to see a rise in the Covid-19 diagnosis area, and how psychiatrists use it, to help them with detecting covid symptoms. Furthermore, we thoroughly explain the different machine learning and data prepossessing algorithms used in this thesis like support vector machines and histogram equalization.

(ii) **Methodology:** A detailed overview of the dataset used is given, and what are the different classes the data was divided to. We also explain what are the different preprocessing stages that the data went through. Moreover, we illustrate how each model was built and its different parameters. Furthermore, a simple application was also discussed as a proof of concept for this study. Lastly, we went over the evaluation metrics that will be used to compare our models.

(iii) **Results:** We discuss and analyse the results of each model with different image preprocessing algorithms. We compare the different approaches on frame level and patient level using F1-score, cross-validation and confusion matrix. Moreover, the application results are shown, and how model interpretability was used to show the highlighted areas in the lung ultrasound video results. Finally, we decide on which was the best method overall.

(iv) **Conclusion:** We conclude and discuss the overall view on the experiment, and how it will help psychiatrists in real life clinical environment.

(v) **Future Work:** In the future work section, more approaches and methods that can be experimented with are proposed. In addition, to what parts of the experimented can be enhanced and improved on.

# Chapter 2

# Background

## 2.1 Covid

### 2.1.1 What is Covid-19?

The Coronavirus disease 2019 (COVID-19) is an illness caused by severe acute respiratory syndrome, which had the first case reported in the middle of an unknown illness outbreak at the end of 2019 in Wuhan City, Hubei Province, China. Coronaviruses are a single stranded RNA viruses that generally infect humans and also animals. Coronaviruses have seven subtypes, one of these subtypes are the beta-coronavirueses which cause asymptotic or mildly symptotic infections. SARS-Cov2(Covid-19) belongs to the beta subtype specifically the B-lineage and was a known virus among animals but have successfully made its transition among humans [3].

### 2.1.2 Covid Testing Methods

Since the rise of the Covid-19 pandemic all over the world researchers have been racing to find out the best and most accurate way to test for the virus. Unfortunately, there was not many ways which you can test for the virus. Polymerse Chain Reaction (PCR) testing is used by preforming a nasopharyngeal swab which is then taken to the labs to see if they can detect two specific segments of Covid-19 genome, if both segments are detected then the test is positive. [4]. Although, till now PCR is thought to be the gold-standard for Covid-19 testing, the clinical sensitivity of PCR ranges from 84% to 64% [5] and it needs physical contact. One of the other methods that was also used was CT scans on the lungs, since Covid-19 is a respiratory disease. Although CT-scans might have high accuracy but it is highly irradiating, expensive and can cause cross infections since all patients use the same machine [6]. Chest X-ray scans have been also used as a helping aid for identifying covid although it has low sensitivity and specificity for covid 19 [7].

### 2.1.3 Lung Ultrasound

Ultrasound is cheap, durable and have seen an increase in underdeveloped parts of the world. Not to mention that it is available at the bedside of any patients [8]. LUS is preformed by passing the probe longitudinally, perpendicular and obliquely on the chest ribs. The LUS approach can vary differently according to the settings and clinical situations but they usually follow the principles of "point-of-care ultrasound". By following these principles scans become highly accurate for diagnosing specific pulmonary conditions [9]. Due to further research, it was discovered that certain LUS patterns appear in Covid-19 patients. Such as vertical pneumogenic artifacts which originate from the plural lines and consolidations which are usually visible in the scans as seen in Figure 2.1 [10]. Which have pushed psychiatrists to use LUS to help them with the detection Covid-19 patients.

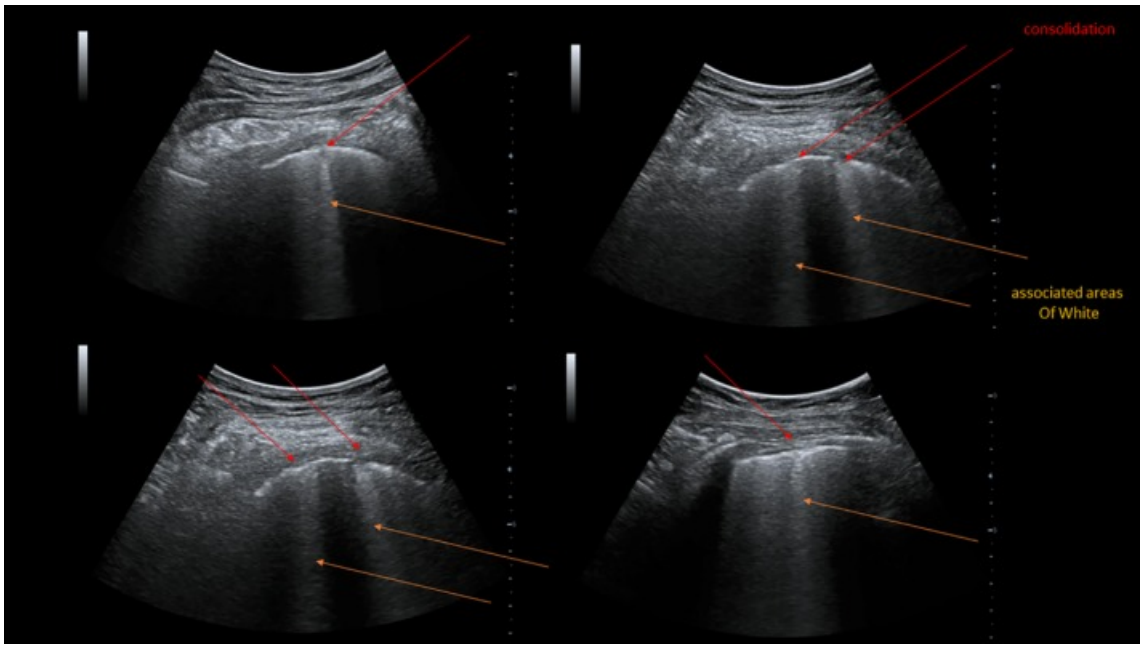


**Figure 2.1:** An example of Covid-19 infected lung ultrasound scan. Pleural lines marked with yellow arrows and consolidations marked with red arrows [10].

## 2.2 Data Preprocessing

### 2.2.1 Histogram Equalization

Adaptive Histogram Equalization (AHE) is a great method for improving image contrast for different types of images such as natural, medical and other initially non-visual images. Its automatic operation and effective presentation in the medical images criteria made it

compete with standard contrast enhancement method. The original form of the method was invented by Ketcham et al. , Hummel and Pizer. Basically the method relies on applying histogram equalization for each single pixel based on the pixels surrounding it in a certain region. Which means each pixel has histogram equalization applied to it based on its intensity rank to its surrounding pixels. However, this method proved to be rather slow and sometimes the image output showed undesirable features. Therefore, AHE addressed those problems [11]. The method that will be used during this project is CV2 Equalize Hist, which uses AHE to stretch out the intensity of range of the pixels. An example of this can be seen in Figure 2.2 where the dog image seems to have all pixels at the same intensity and same grey color. As seen in its intensity distribution graph all the pixels seem to be clustered in the middle region but after applying AHE the pixel intensities are stretched out which results in a picture with clearer details of different objects.

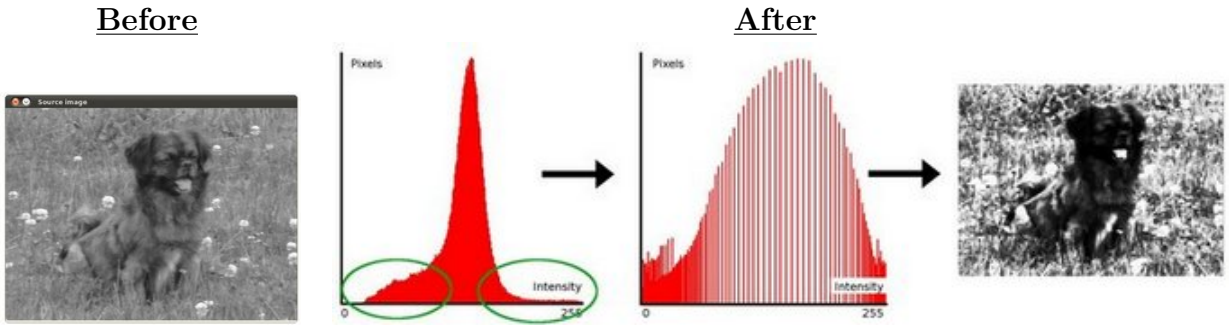


**Figure 2.2:** Figure shows an image of a dog along with its pixel intensity graph before and after applying AHE. source

### 2.2.2 Standardization

There are different types of data preprocessing methods like non-linear transformations, normalization and discretization. However, in this study we will use standardization (also referred as z-score). Data standardization is a widely used method in the data preprocessing world. It makes the attributes of the values fall in a specific range based on the mean and variance of the data itself. It has also been found that standardization improve the performance of ML methods like SVM in classification problems [12]. Data standardization is also beneficial for neural network training but the advantage gained by it can diminish if the network is exceedingly large [13]. We will use sklearn standard scalar method in this study. which standardizes the features by cutting out the mean and scaling the unit variance. For example if we have a sample **x** the standard score is calculated as:

$$\mathbf{z = (x - u) / s}$$

**u** and **s** are the mean and the standard deviation of the training samples respectively.

### 2.2.3 Principal Component Analysis

PCA is a method used for dimensionality reduction for large data sets. It does that by simply transforming the large data set into a smaller one by reducing the number of variables but with keeping the most important information. Multiple steps need to be carried to preform PCA. First, the data set is analyzed to extract the most important information (variables) and these variables are sorted in a hierarchical fashion and are referred to as principal components of the dataset. The first principle components is required to have the largest variance of the dataset, which means it has the largest effect on the data. The second component is then calculated with the restriction of being orthogonal to the first principal component. The rest of the components are calculated the same way. After calculating all the principal components in order of their variance the dataset is then compressed by removing the least important features (components) and keeping the most important ones. Therefore, reducing the dimensionality of the dataset [14]. In Figure 2.3 a 2d dataset plot can be seen after extracting its principal components the next step is selecting which components do we remove in order to reduce the dimensionality of our dataset. It is best to remove the component with the least variance.

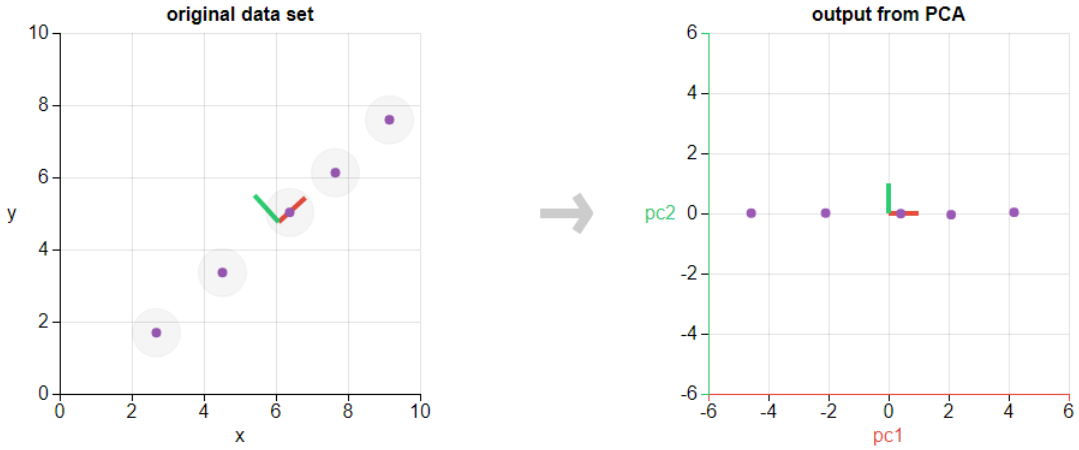


**Figure 2.3:** On the left we have a 2D plot of a dataset showing its x and y variables. On the other side its the same plot of the dataset but after calculating the principal components pc1 and pc2. source


**Figure 2.4:** Shows how the dataset in Figure 2.3 will look after using one of the components and removing the other. source

Figure 2.4 shows how the data set will look like when removing each component and using the other. If this was a real life example we would use pc1 and exclude pc2 since pc1

shows a much larger variance than pc2. The function that will be used for this study to apply pca is the one from sklearn decomposition library which follows the same approach and centers the data without scaling it for each feature.

### 2.2.4 Oversampling

#### Data Imbalance

One of the main problems in the field of feature extraction and classification is data imbalance. When a dataset that contains multiple classes have one or more of its classes have large dominance over the number of samples of the other classes, which means that the dataset is implicitly unbalanced. This imbalanced form can be seen in multiple ratios like 100:1, 1000:1 etc. which is present in many real life datasets [15]. Since the number of samples for each class plays a vital role in training classifiers. Therefore, due to the small occurrences of the important classes the classifiers gets insufficiently trained on those classes. Hence, making the classifier biased towards the majority class [16].

#### Synthetic Minority Oversampling Technique

One of the leading techniques to address the problem of imbalanced data for classifiers is SMOTE. SMOTE was described by Nitesh Chawla, etal. in 2002. What SMOTE does is creating more samples of the minority class till it is the same size as the majority class. It achieves this by selecting samples that are relatively close in feature space and draws line between those samples in the feature space. It then creates different samples at different points along that line [17].
Figure 2.5 shows an example of an imbalanced dataset that would cause classifiers trained on it to be biased toward the class with green dots. In Figure 2.6 it can be seen how SMOTE is applied and the lines are drawn between the minority samples. Along these lines synthetic samples are generated as small red bright dots. Now both classes have a 1:1 regarding the number of their samples.

## 2.3 Machine Learning

### 2.3.1 What is Machine Learning?

ML is a subset of artificial intelligence that gives systems the ability to learn and improve from experience without humans explicitly programming it to improve. The main focus of ML is to create computer programs that can access data and use this data to learn and improve themselves. The learning process starts with the program trying to look for patterns in the data, to help it make better decisions on future examples similar to the ones it has. However, there are categories of ML methods like supervised, unsupervised,

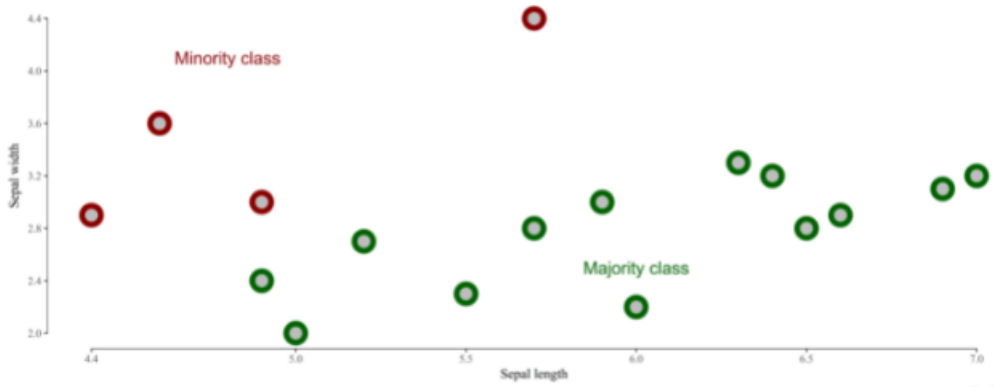


**Figure 2.5:** The 2D Plot shows a imbalanced dataset plot. Where red points are the minority class and green points the majority class. source

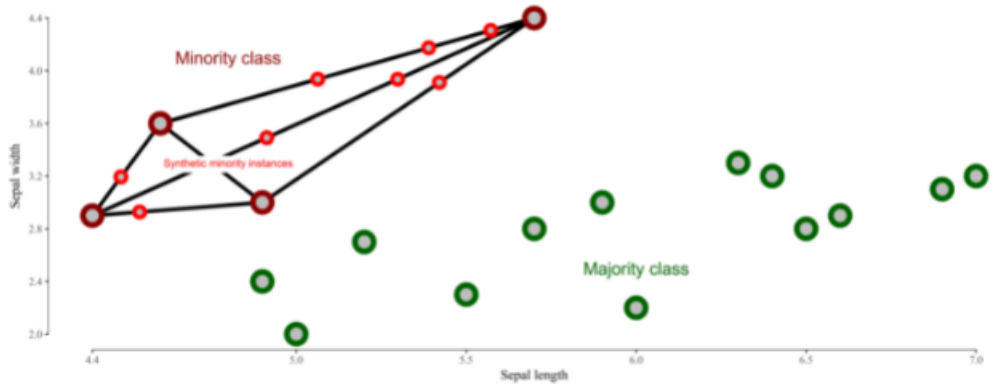


**Figure 2.6:** Same 2D plot seen in Figure 2.5 but after applying SMOTE. Small red dots are synthetic samples produced by SMOTE. source

semi-supervised and reinforcement learning. This study focuses on supervised ML algorithms. In contrast, supervised learning uses labeled datasets to train its algorithms to classify data or predict future outcomes accurately. It does that by adjusting the weights of the model each time a new data is fed into it, which happens as a part of the cross validation process.

### 2.3.2 Support Vector Machines

SVM is an algorithm which has the objective of finding the best hyperplane in an N-dimensional space (N is the number of features) that will be used to distinctly classify data points. Multiple classifiers use the method of finding a separating hyperplane to separate different instances of the data. Like as in Figure 2.7 (a), there is 2-Dimensional plane in 3-dimensional space that separates two classes. These classifiers are called hyperplane-based classifiers. Therefore, it is not only unique for SVM.

However, in many instances their are multiple separating hyperplanes that can be selected. The method SVM uses to select which one would be the best is where it differs from other hyperplane-based classifiers. One of the ways that the SVM can select the best hyperplane is by defining the distance that is between the hyperplane and the nearest expression vector as the margin of the hyperplane. Therefore, the SVM selects the maximum-margin separating hyperplane. However, this method assumes that the data is clearly separated and does not contain outliers like in Figure 2.7 (b). Which would cause the SVM to have many miss classifications. In order for the SVM to be able to deal with those outliers it uses a soft margin like in Figure 2.7 (c) that allows a specified set of examples that are allowed to be miss classified in order for the SVM to work correctly. But in some cases the data is inseparable and there is not a possible way to find a soft margin between them. In that case the kernel function is used. What the kernel function does is adding additional dimensions to the data. The new dimension is added by applying an expression like squaring a 1-dimensional data and making the output the 2nd dimension. In essence the kernel function is a mathematical expression that allows the SVM to preform higher-dimensional classification on lower-dimensional data. One of those kernel functions is the Radial Basis Function which uses curve-shaped boundaries to separate different classes. It mainly has two parameters.First one is gamma which is how spread the decision region will be. The second one is the penalty of miss classifying an instance which is referred to as C [18].

### 2.3.3 Logistic Regression

LR is one of many regression techniques used for analysis and ML like linear,ridge and polynomial regression. Linear regression works by fitting a line through it as seen in Figure 2.8 a. Using that line we can calculate $R^2$ and determine if salary and experience are correlated. Moreover, we can calculate p-value to determine if $R^2$ value is statistically significant.

$$R^2 = \frac{Variance\,explained\,by\,the\,model}{Total\,variance}$$

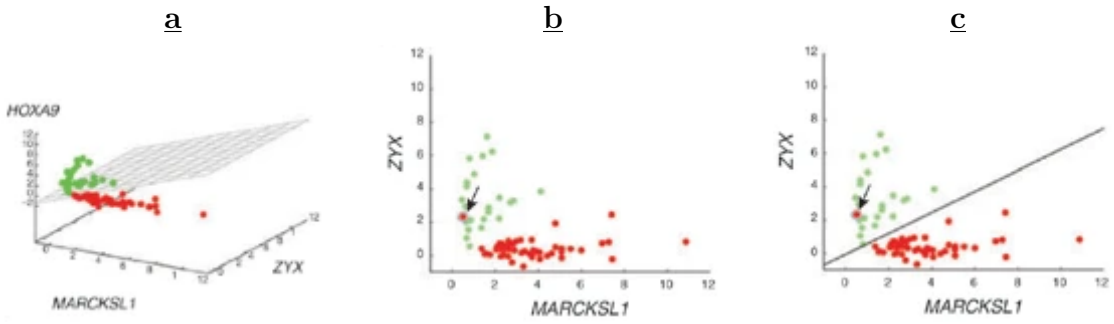


**Figure 2.7:** **a:** Hyperplane seperating two classes from a public dataset. **b:** 2D plot representation of dataset with a red point outlier. **c:** Separation of the dataset by a 1-Dimensional soft-margin [18].

Then we can use the line to predict size using a weight value. Logistic is similar to linear except that it predicts whether something is true of false instead of predicting something continues like linear. The way it does that is by fitting a sigmoid logistic function to the data as seen in Figure 2.8 b. The curve goes between 0 and 1 which means that the curve helps predict the probability that Y would happen given X as seen in Figure 2.8. LR however can not calculate R2 like linear. Instead it calculates something called maximum likelihood estimation. Which is a method that maximizes a likelihood function in order to estimate the parameters of a probability distribution.

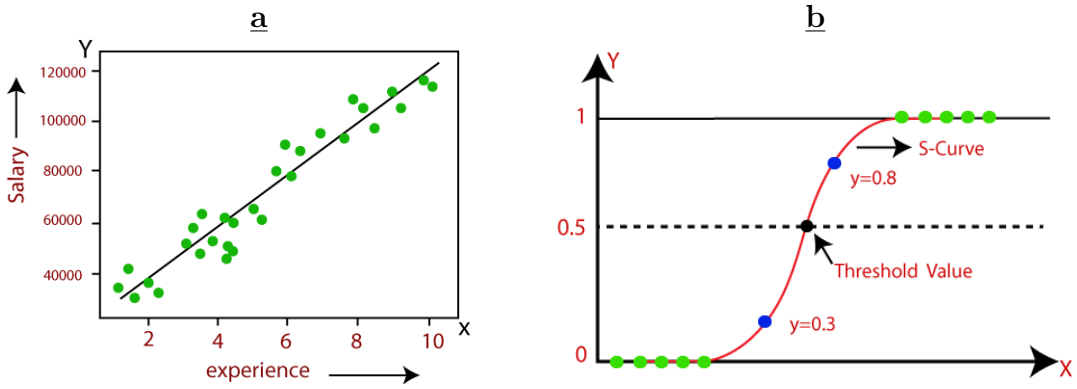


**Figure 2.8:** **a:** shows a graph of a a company dataset where Y is the given salary and X is the years of experience of each employee, the line fitted between them is the linear regression fit. **b:** is a plot showing how the the sigmoid function (S-curve) is plotted the green dots are data points for either 1 or 0 that where used to plot the S-curve. source

### 2.3.4 Decision Trees

DT are a non-parametric supervised learning method used for classification and regression, which follows the divide and conquer approach. DT have the ability to extract

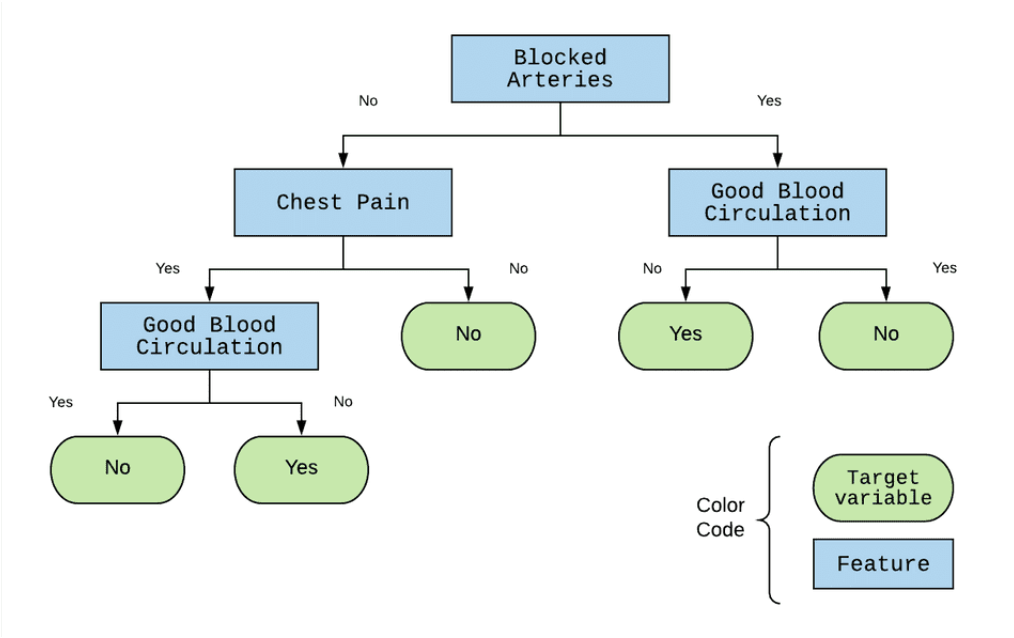


**Figure 2.9:** A simple decision tree that answers the question of whether a given artery is blocked or not by following multiple decisions.

features and find patterns from large datasets that are critical for discrimination and predictive modeling. DT have already made their mark in ML and artificial intelligence literature and are starting to develop in chemical and biochemical sciences [19]. One of the main advantages of decision tree modeling that gives it an edge over other classification techniques, is its interpretability of the constructed model. In Figure 2.9 there is a simple decision tree which can be easily explained because of its interpretation. The tree has multiple questions in order to take a decision of which sub-node it should proceed to. The sub-node will either lead to a final answer or another question.
There are different algorithms for DT to decide if the node will be split into two more sub-nodes or not. Whenever a new sub-node is created it leads to an increase in the homogeneity of the resultant sub-nodes. In other words, the tree splits the nodes of all given variables and proceeds to select the split which results in the most homogeneous sub-nodes. Many algorithms are used in order to find the optimal splits like ID3, C4.5, C5.0 and CART. The decision tree classifier implementation provided by sklearn that will be used in this study uses CART. CART is able to support numerical target variables and will not compute rule sets. It also uses the feature and threshold that has the largest information gain at each node to construct binary trees.

### 2.3.5 Random Forest

Random forest is an ensemble learning method used for classification and regression. It works by constructing various decision trees at training time as seen in Figure 2.10. Then fits them to sub-samples of the dataset. Which means for Figure 2.10 the final class will

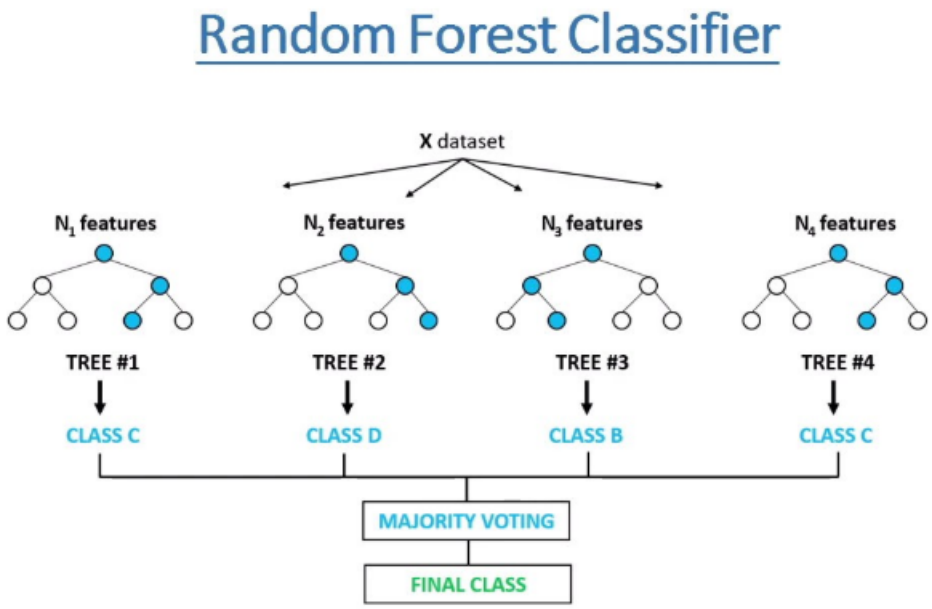


**Figure 2.10:** An overview of an RFC which fits dataset X to four decision trees. It then proceeds to take majority voting to decide the final result.

be class c since its the most voted by the different trees. It is used in classification tasks by selecting the output of most trees constructed inside the forest. RFC usually give better results than DT since DT have the habit of overfitting to their dataset. Moreover it uses averaging methods to improve predictive accuracy.

## 2.4 Deep Learning

DL is a part of the ML family, it is mainly based on neural networks for short. It has many types of neural networks including but not limited to deep neural networks, deep believe networks and recurrent neural networks. The aim of these neural networks is to try and mimic how the human brain works, specifically the part of learning from large amount of data. Although, DL is a subset of ML they do differ. ML mainly works with structured labeled data in order to make its predictions. This does not mean that it can not work with unstructured data. However, if it does the data needs to be pre-processed to organize it for the ML model. Nevertheless, DL gets rid of some of the pre-processing steps that is involved with ML. It can consume and process unstructured data directly, like text and images, and it takes care of the feature extraction.

### 2.4.1 What is deep learning?

### 2.4.2 Neural Networks

Neural networks are a part of ML and are the skeleton of DL. The human brain is the inspiration behind their name because they resemble how the biological neurons signal to each other. Neural networks contain multiple node layers as seen in Figure 2.11, which contain an input layer, a minimum of one hidden layer and an output layer. Each node, or artificial neuron, is connected to another neuron and has a certain weight and threshold associated with it. If any individual node has an outbout above the specified threshold value, the node is activated, sending data to the next layer in the network. If not, no data is moved to the next layer in the network. Neural networks count on being trained on data to learn and increase their accuracy over time. Once these networks are fine-tuned for accuracy, they become a powerful tool in computer science and artificial intelligence.

The way neural network works is as if each individual node is its own linear regression model, with its own input data, weights, bias and output. The formula would look like this:

$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

$$\text{output} = f(x) = \begin{cases} 1 \text{ if } \sum w_1 x_1 + b \geq 0 \\ 0 \text{ if } \sum w_1 x_1 + b < 0 \end{cases}$$

Whenever an input layer is decided the weights are assigned. The weights are what gives each value its importance, which means variable with larger weights have bigger effect on the output. After all the inputs are multiplied with their weights and summed. The output is given to the activation function, which decides the output and whether or not the node will be activated and pass the data to the next layer. This procedure of moving the data through the neural network is called feedforward network. However, in order to use the neural network for more practical cases like image recognition or classification. We will need to evaluate the accuracy of the model using the cost (or loss) function. Which is also known as the mean squared error. The function looks like this:

$$Cost\ Function = MSE = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y} - y)^2$$

In the function i represents the sample index, y-hat is the outcome, y is the actual value and m is the total number of samples. The ultimate goal is to minimize the cost function to assure correctness of fit for any observation. As the model calculates its weights and bias, it tries to reach the point of convergence or local minimum like in Figure 2.12, by

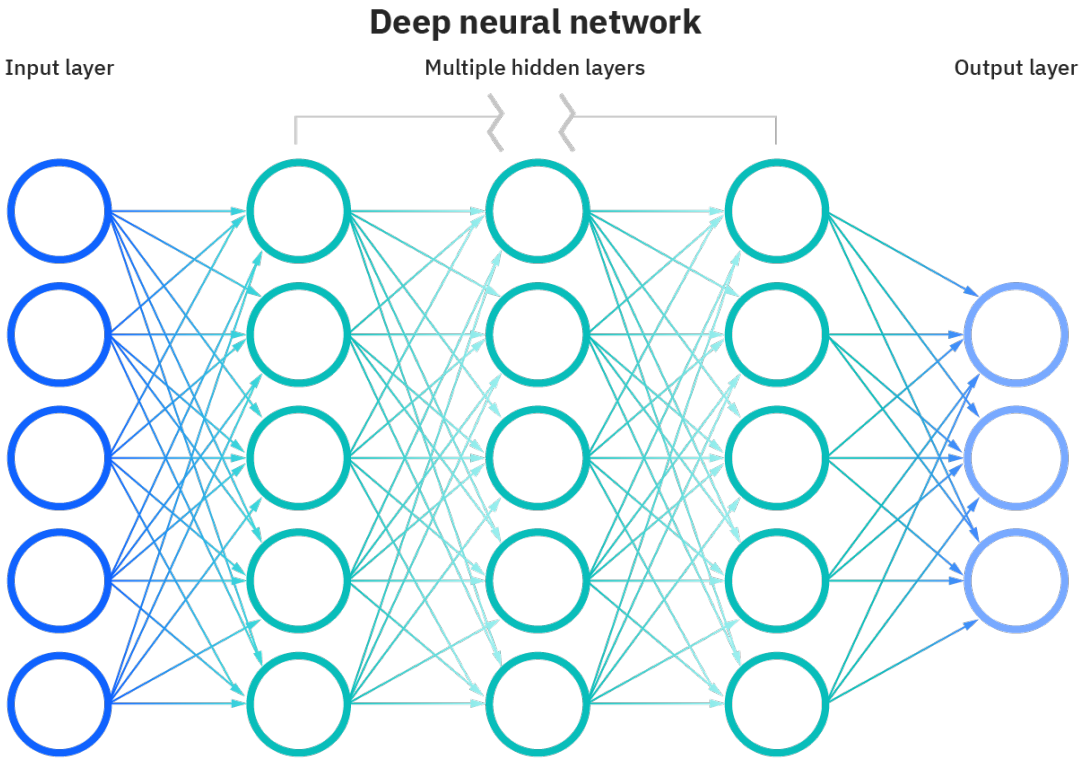


**Figure 2.11:** An overview of how neural network layers are interconnected. The blue nodes are the input layer, green nodes are the hidden layer and last is the output layer. source

using the cost function and reinforcement learning. Gradient descent is how the model keeps adjusting the weights, allowing it to decide the best direction to take for it to reduce error. The parameters of the model change gradually with each training example, to converge at the minimum [20].

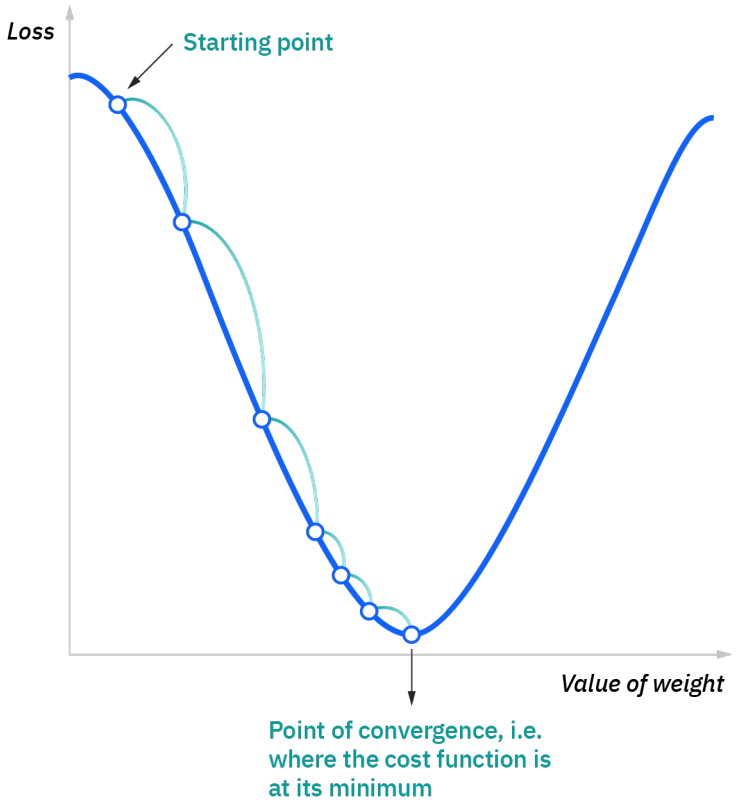


**Figure 2.12:** A plot showing how the model converges to reduce the loss function and reach the local minimum. The blue starting point is where the model loss function is at the start of the training. the green lines shows how the model keeps gradually converging with each training sample, to reach the minimum cost function. source

### 2.4.3 Convolutional Neural Network

CNN is one of many types of neural networks, which usually excel at different data types and tasks. Such as, recurrent neural networks are mostly used for speech recognition and natural language processing. However, we picked CNN for our task because it excels at classification and computer vision tasks. Before CNNs where first introduced, feature extraction methods were manual and time-consuming, and they were the only option for object recognition tasks and image classification. But CNNs leverages many principles from linear algebra specially matrix multiplication, to detect patterns in images. CNNs

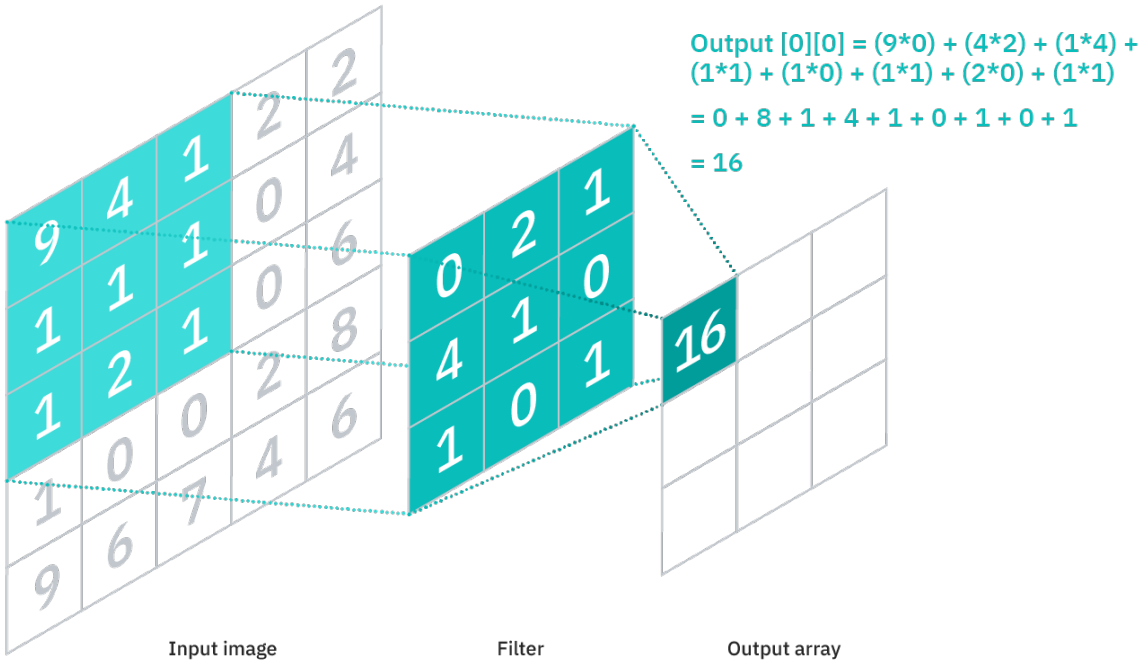


**Figure 2.13:** A simple convolutional layer representation of an input image on the left and the filter in the middle being applied to a part of the image. Then the dot product is calculated and projected to the output array. source

mainly consist of three types of layers: Convolutional layer, pooling layer and fully-connected (FC) layer.

### Convolutional Layer

convolutional layer is the heart of the CNN, and is where most of the work is done. It consists multiple components which are the input data, filter and feature map. If the input data is an RGB image, which means it will be a 3-D matrix . The kernel or filter is normally a two dimensional array of weights that represents the part it is scanning in the image. Kernels might be different in size but for this explanation it will be 3x3. As seen in Figure 2.13 the filter is applied to a part of the image, then the dot product is calculated and projected into an output array. Afterwards, the filter keeps shifting and repeating the same process until it has swept the whole image. The final output of that process is called a feature map or activation map. It is important to note that the weights in the filter remain the same as it moves through the image.

There are some weights that change and adjust during training like weight values, which adjust during back-propagation and gradient descent. However, there are some hyper parameters that need to be set before training the neural network, since they affect the volume size of the output. First, the number of filters since four distinct filters yield four different feature maps, which will create a depth of four. Second is, stride is the

distance (usually represented in the number of pixels) that the kernel will move over the input matrix. A big stride will result in a smaller output. And lastly is padding which is used when the filters do not fit the given image. What it does is that it sets all pixels that are outside the input matrix to zero. These are the main points of the convolutional layer and as mentioned above a CNN can have one or more convolutional layer.

#### Pooling Layer

Pooling layers, are what is used to conduct dimensionality reduction, to reduce the number of parameters of the input. They work in a similar way to CNNs, pooling sweeps across the entire input with a filter. However, the filter does not contain any weights. It instead applies an aggregation function toe the values in its field. Which results in the population of the output array. Pooling mainly consists of two methods, max pooling and average pooling. Max pooling is what is used in this study, and it works by sweeping the filter across the input then selecting the pixel with the maximum value to send it to the output array. Although a lot information gets lost in the pooling layer, it is a necessary step that helps improve efficiency ,reduce complexity and reduce the risk of overfitting.

#### Fully-Connected Layer

Since each pixel value of the input image are not connected to the final output layer during their processing in the partially connected layers. In the fully-connected layer each node in the output layers gets connected to each node in the previous layer. Moreover, this layer is the one responsible for the classification based on the features extracted from the previous layers. Note that, while convolutional and pooling layers usually use ReLu functions, this layer tends to use softmax function. It does that in order to classify inputs, producing probability from zero to one.

### 2.4.4 Model Interpretation

#### Why we need model interpretation?

Usually most of the DL models are referred to as "Black Boxes". They have picked up that term because most of the time people do not know why a machine has arrived to a certain solution. Therefore, we have two options either we just accept the machine solution and trust it or take a deep dive and try to Figure out why the machine has decided on that particular solution. Trusting a model for its prediction for some movie recommendations is okay. However, we can not say the same for an advanced model used to predict which kind of drug should be given to a certain patient.

**ELI5**

ELI5 is a python package that aids us in interpreting ML model predictions. ELI5 is specially popular with sklearn regressors and classifiers, XGBoost, XGBoost, Keras, etc. . In this study we will use it with keras. Given a CNN model and an image, ELI5 will decide which parts of the image had the largest impact on deciding the prediction taken by the CNN. It does that using a method called Grad-CAM .

### 2.4.5 Grad-CAM

Grad-CAM is a technique that produces visual explanations from decisions taken by mostly any CNN model. Grad-CAM uses gradients from any target concept, moving into the final convolutional layer to create a coarse localization map highlighting the important parts in the input image that resulted in predicting the concept. To be more precise, Grad-CAM goes into the last convolutional layer of the cnn and fetches the gradient information flowing into it, to understand the importance of each neuron in the taken prediction. Afterwards, it generates a heat-map that can be projected into the original input image to see which part of the image was the most important like in Figure 2.14 [21].

(a) Original Image (b) Grad-CAM 'Dog'

**Figure 2.14:** (a) is the input image with a cat and dog. (b) The result image from Grad-CAM after using CNN trained to identify dogs [21]

## 2.5 Literature Review

There were three papers in total two of which use the same repo with the same dataset and one which used a different dataset but did not provide it, and when I emailed them asking for it no reply was given.

Born, Jannis, Brändle,etc., (2020) is the closest paper to our goal which is detecting covid-19 with lung ultrasound using ML or DL techniques they only used the convex data probe and processed the videos with a frequency of 3Hz taking maximum of 30 frames from each video which resulted in a total of 1103 images (654 COVID-19, 277 bacterial pneumonia, 172 healthy). Images were resized to 224x224 and any measure bars or text

was cropped from the frames. Then they used the convolutional part of VGG-16 followed by one hidden layer of 64 neurons with ReLU activation, dropout of 0.5 and batch normalization and further by the output layer with SoftMax activation. In the results they reached an accuracy of 89% with their model [22].

Born, Jannis, Wiedemann, Nina,etc., (2021) is from the same institute and mostly same authors of the first paper hence why they used the same repo and dataset. However, they focused on how to accelerate and enhance the detection of lung pathologies using different DL techniques and analyzing the difference between them. It was mainly divided into Frame based experiments and video-based experiments. The frame-based experiments used different models such as (VGG, VGG-CAM, NASNET-Mobile, VGG-Segment, Segment-Enc) VGG had the best performance with an accuracy of 87.8%. Video-based experiments used (VGG, Models Genesis) VGG also had the best performance with an accuracy of 90%. The paper also experimented with class activation maping (CAM) for model explainability to see if the model predictions were based on a visible pathologies patterns. They compared their two best models to each other VGG and VGG-CAM and they found out the latter was better quality by observation [23].

Roy, Subhankar, Menapace, etc.,(2020) similar to the second paper, this one provides different techniques for classification and localization of Covid-19 markers. Which means it does not just detect Covid but it also tries to highlight the area which is the reason the model thinks its Covid. They used a dataset from Italian hospitals which is provided with labels indicating the degree of disease severity at a frame-level, video-level and pixel-level (segmentation masks). Which is not available in the first two papers and we could not get hold of their dataset. They also introduced a novel deep network, derived from Spatial Transformer Networks, which simultaneously predicts the disease severity score associated to an input frame and provides localization of pathological artefacts in a weakly-supervised way, and introduced a new method based on uninorms for effective frame score aggregation at a video-level. The paper techniques were out of the scope of this project, we chose to mainly focus on the previous two with their dataset [24].

# Chapter 3

# Approach and Methodology

## 3.1 Approach Overview

Our approach as illustrated in Figure 3.1 starts by extracting our sample frames from the dataset that will be used in this experiment. Then, passing those frames through the data processing stage to get the frames ready to be fed to the classifiers. We have 2 categories of models CNN and machine learning. These models are trained and tested on the data by our selected evaluation metrics to finally get their results.

## 3.2 Dataset Description

We are using the largest publicly available LUS dataset (https://github.com/jannisborn/covid19_ultrasound), it includes samples of Covid-19 patients, patients with bacterial pneumonia, viral pneumonia and healthy samples. As we can see in Table 3.1 we have a total of 165 samples. We decided to only use Convex probe data due to its availability and we ignored viral Pneumonia as its not the focus of this study. We mainly focused on the 124 (30 Covid-19, 36 Pneumonia, 56 Healthy) Convex videos and 28 Convex images. You can see three sample images from the convex probe in Figure 3.2. Image A is Covid-19 infected lung with irregular A-lines, image B is a pneumonia infected lung and image C is a healthy lung.

## 3.3 Methodology

### 3.3.1 Data Processing

When extracting frames from videos we noticed that lung ultrasound videos have similar frames with low variance. Which means that extracting all frames from each video would

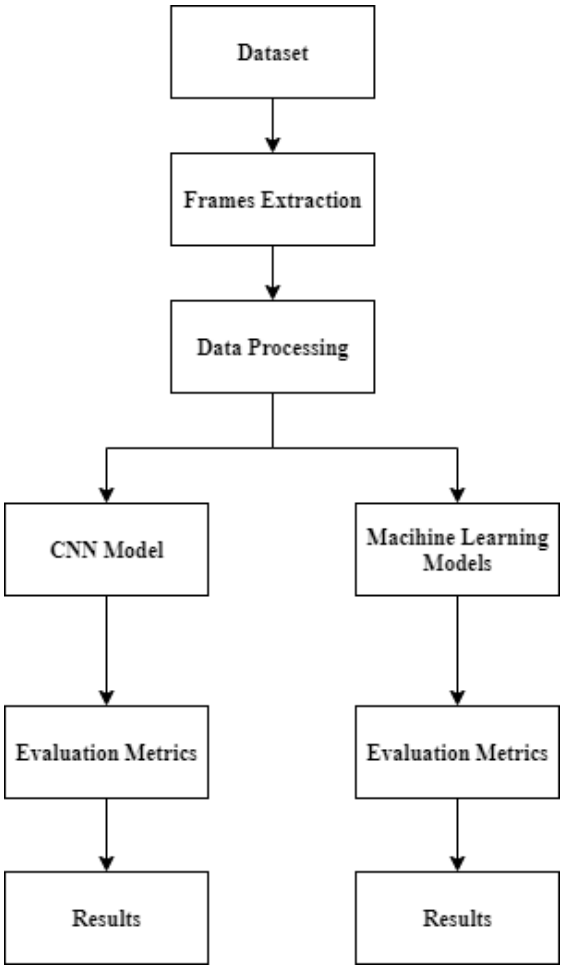


**Figure 3.1:** A diagram showing an overview of our approach

make the classifiers have too much identical frames. Moreover, each patient video had a different frame rate with different total number of frames. Which would cause an imbalance for different cases and biased classifiers. We decided that we will experiment with 5 frames per video with a total of 610 frames(150 Covid-19, 180 pneumonia, 280 healthy) and 15 frames per video with a total of 1830 (450 Covid-19, 525 pneumonia, 840 healthy).

Our frames where extracted from equally separated intervals in the video to make sure no sequentially identical frames were taken. We found that it is best to convert our images to Grey-scale and resize them to 224x224x1. Histogram equalization was preformed on all frames to improve the contrast of the image as we can notice in Figure 3.3. Since, standardized data produce more consistent hyper-planes with SVM and other machine learning models [25]. Standardization was experimented with, using Sklearn standard scalar function, which standardizes the data by removing the mean and scaling the unit variance. We tried applying PCA to reduce the dimensionality of the data. Since high numbers of redundant or highly correlated attributes may lessen our model's classification accuracy [26]. We examined different number of components for our PCA.

| | Convex | | Linear | | Sum |
|---|---|---|---|---|---|
| | **Img.** | **Vid.** | **Img.** | **Vid.** | |
| **Covid-19** | 4 | 30 | 3 | 3 | 40 |
| **B**acterial Pneu. | 14 | 36 | - | 1 | 51 |
| **Viral Pneu.** | - | 2 | - | 3 | 5 |
| **Healthy.** | 10 | 56 | - | 3 | 69 |
| **Sum** | 28 | 124 | 3 | 10 | 165 |

**Table 3.1:** Dataset size. Number of images and videos of every class per probe

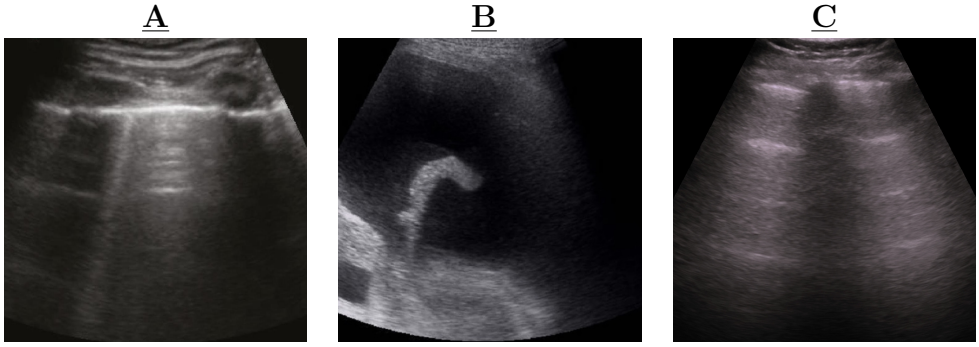


**Figure 3.2: Example lung ultrasound images from the convex probe data.** **A:** Shows a Covid-19 infected Lung. **B:** A pneumonia infected lung,with dynamic air bronchograms surrounded by alveolar consolidation. **C:** Healthy lung with normally aerated horizontal A-lines.

Furthermore, As we can see in Figure 3.4 our dataset classes are highly imbalanced, which will cause our classifiers to be biased towards the majority classes (pneumonia and healthy). Which made us experiment with oversampling to make all classes equal to each other. The oversampling technique we decided to use after trying out different methods was SMOTE which is provided by sklearn library. SMOTE synthesizes new examples for the minority class. All these methods were experimented with before the frames were fed to different classifiers to see which would give the best accuracy. But for our cnn there was an extra step that can not be applied to normal machine learning models, which is applying data augmentation, especially flips and rotations (up to 10 degrees) and translations (up to 10%) to alter the dataset and avoid overfitting. Finally, data was split on a patient-level, hence it was guaranteed that no frames of single video would be present in the train and test folds. We also ensured that the ratio of the classes in the test fold is the same as the train fold to get the most convincing results. 3.5.

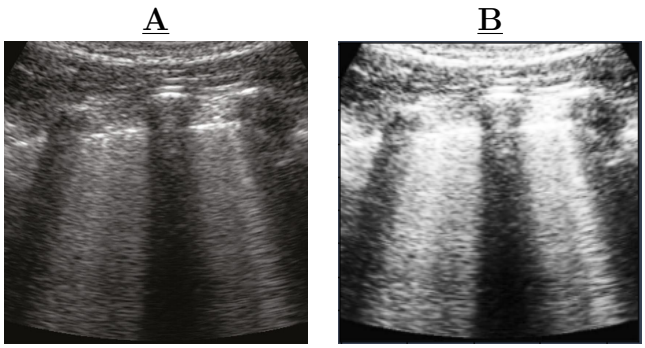


**Figure 3.3:** **A** Shows lung ultrasound sample before applying histogram equalization. **B** Same sample after histogram equalization.

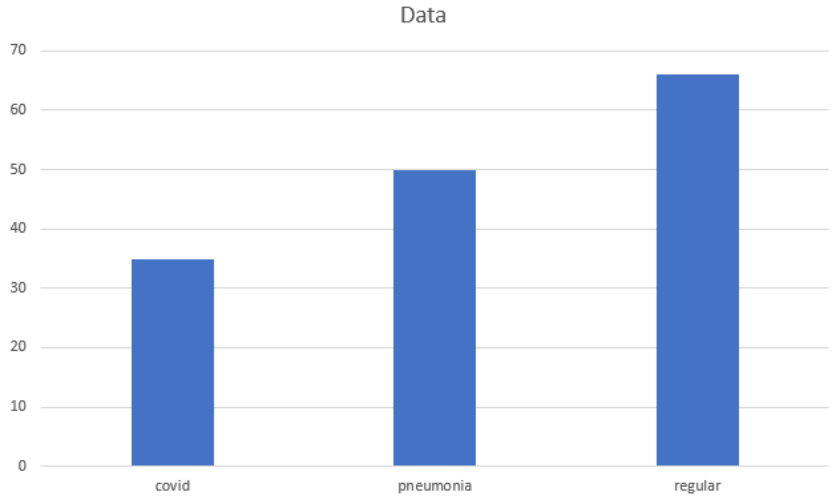


**Figure 3.4:** Shows Frame count imbalance of the classes after extracting 5 frames per video

### 3.3.2 Classifiers Overview

#### Models Introduction

Different models were experimented with to Figure out which had the best results with our dataset. All models had the same random state and were fed the same train and test data to ensure accurate comparison. Note that for our ML approach as seen in Figure 3.5 we tried standardizing the data and applying PCA for our ML models but we did not do that for our CNN approach as seen in Figure 3.6. Because the linear transformation of data preformed by the PCA is also preformed just as well by the layer weights of our CNN so we decided it was not necessary. As for our CNN approach random data augmentations were done as mentioned in the data processing sections which applies transformations randomly during training epochs of our CNN model and that can not be done with our ML models.

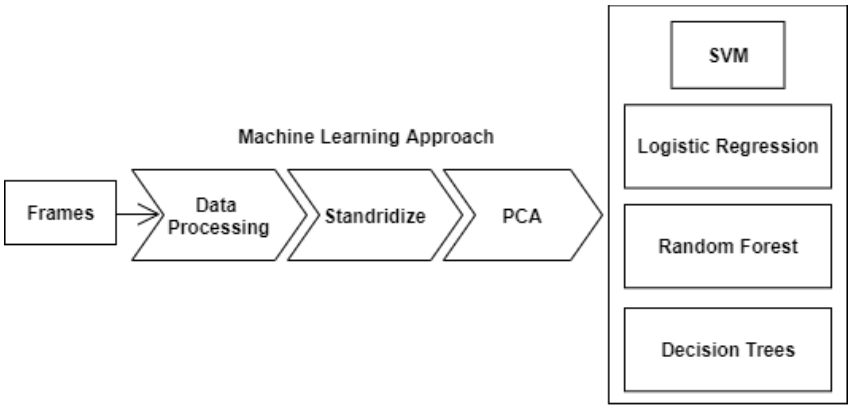


**Figure 3.5:** Diagram showing the process which the frames go through before reaching Pipeline 1 classifiers

### Support Vector Machines

SVMs are a collection of supervised learning algorithms used for classification and regression, they are also highly effective in high dimensional spaces. Support Vector Classification model was used. The regularization parameter was set to 1.0 with an RBF kernel , degree of polynomial kernel function was set to 3 and gamma set to auto.

### Logistic Regression

Logistic Regression is an algorithm that predicts the probability of a categorical dependent variable. Model penalty was set to l2 with dual formality, fit intercept was also enabled and the inverse of regularization strength was set to 1.0.

### Decision Trees Classifier

Decision trees utilize numerous algorithms to choose to part a node into two or more sub-nodes. The creation of sub-nodes increments the homogeneity of produced sub-nodes. The selected criterion for the model was gini with the best splitter strategy and max depth 10000.

### Random Forest Classifier

A Random Forest Classifier creates a set of decision trees and fits them on a number of sub samples then uses averaging to enhance predictive accuracy. The model max depth was set to 10000 and 100 estimators (trees in the forest).

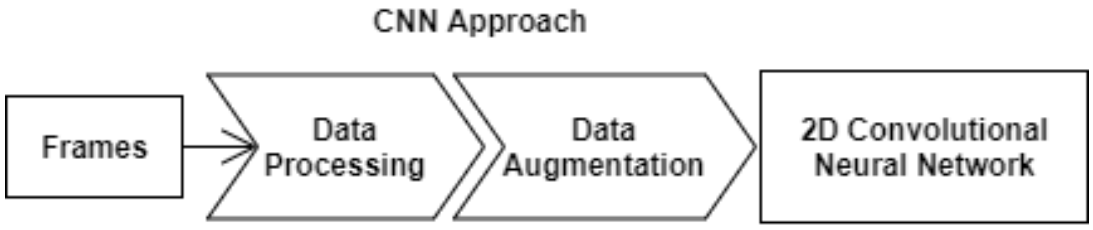


**Figure 3.6:** Diagram showing the process which the frames go through before reaching Pipeline 2 classifiers

### CNN Classifier

We created a 2 Dimensional CNN with 12 layers having an architecture similar to the VGG-16 architecture which can be seen in Figure 3.8. Our CNN architecture in figure **??** contains six convolutional blocks. Each block contains three convolutional layers which decrements by half from 224x224 till 7x7. The layers have kernel size 2x2, ReLU set as the activation function, kernel initializer set as he-uniform and same padding between layers. At the end of our arhcitecture we flattened the results and added two dense layers with 4096 neurons and ReLU activation function. Finally, we added one last dense layer with three neurons and softmax activation to get our classification. To train our network the batch size was set to

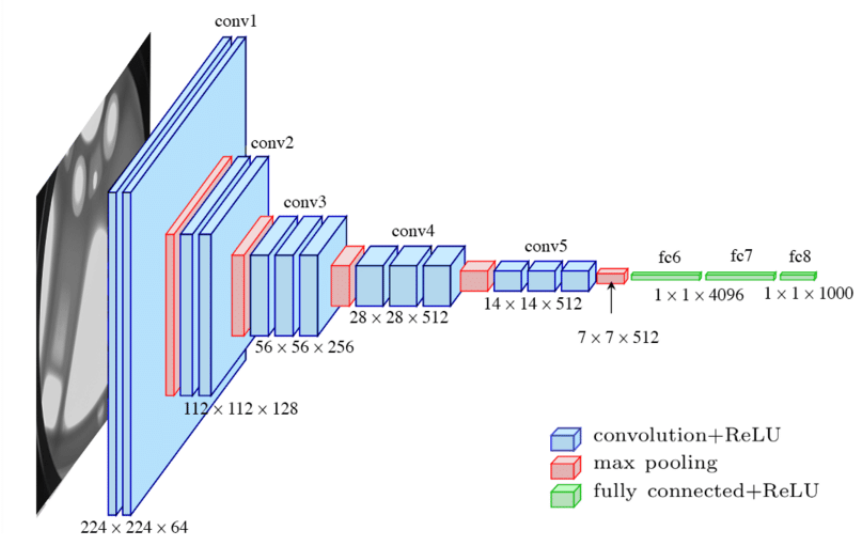


**Figure 3.7:** Image shows the standard VGG-16 network architecture which is constructed from multiple convulotional blocks followed by max-pooling [27].

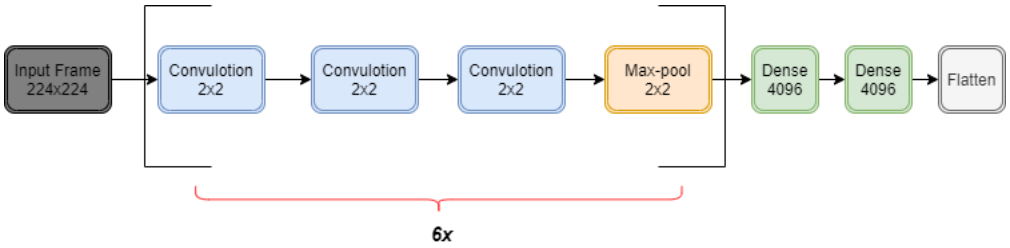


**Figure 3.8:** Image shows an illustration of our CNN model architecture.

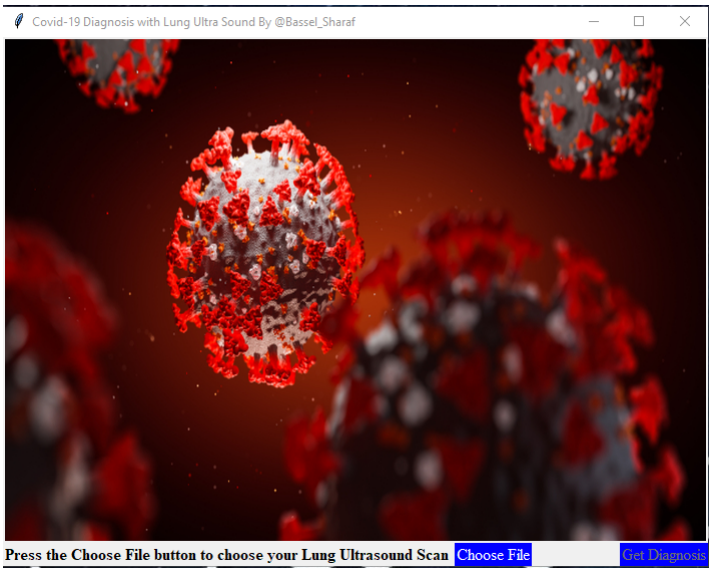


**Figure 3.9:** Image shows the GUI of the python application. It mainly consists of two buttons and middle screen to show the chosen video. The first button is the choose file button, that will allow the user to select a video. The get prediction button will start the application to diagnose the video.

### 3.3.3 GUI

A simple python application was created as a proof of concept for this study with the GUI seen in Figure 3.9. The main aim of the application is to diagnose lung ultrasound video if it is covid, pneumonia or healthy using the CNN model used in this study. First the user clicks the choose file button to select the lung ultrasound video from local files, then proceeds to click the choose diagnosis button. Afterwards, it processes the video frames and applies the necessary pre-processing steps, then feeds it to the CNN model to the get the final diagnosis. Lastly, it shows the visual explanation of why the CNN decided on the given prediction on the selected frames, to be able to understand the reasoning behind the CNN model decision.

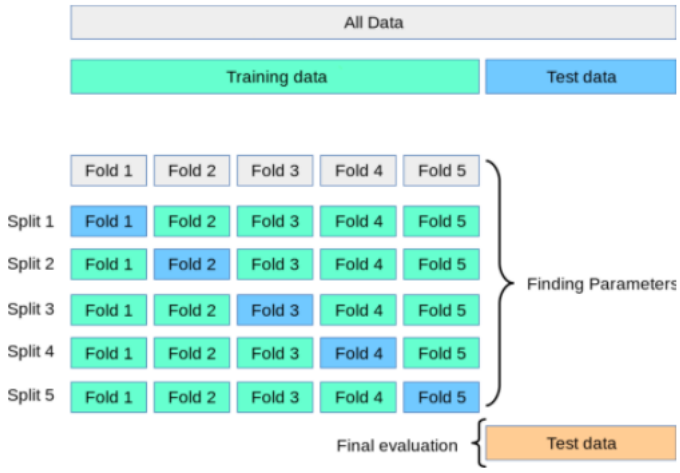


**Figure 3.10:** A diagram illustrating the evaluation method done on our models. Showing 5-fold cross-validation on the test set and the final test on the test set [28].

## 3.4 Evaluation Metrics

To evaluate the performance of our different models we preformed 5-fold cross-validation on the train set, then measure F1-accuracy score on test set as demonstrated in Figure 3.10. Furthermore, we will measure the patient accuracy by aggregating the frames prediction of each patient so that each patient video would have one prediction as illustrated in Figure 3.11.

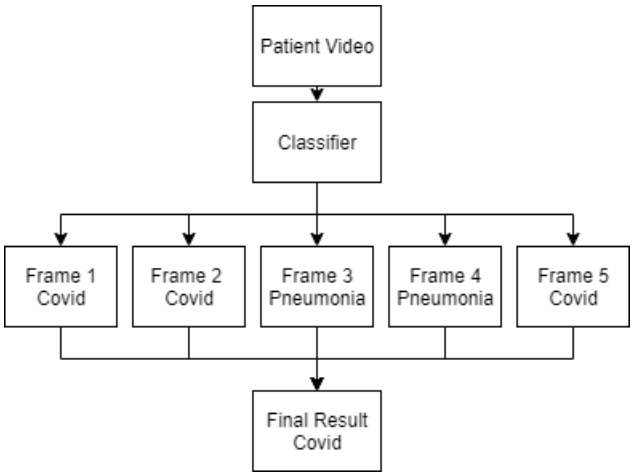


**Figure 3.11:** Figure demonstrates how the classifier will have a prediction for each frame from the patient video then all predictions will result in one final result by the majority vote.

# Chapter 4

# Results

## 4.1 Five Frames Per Video

### 4.1.1 Frame Level

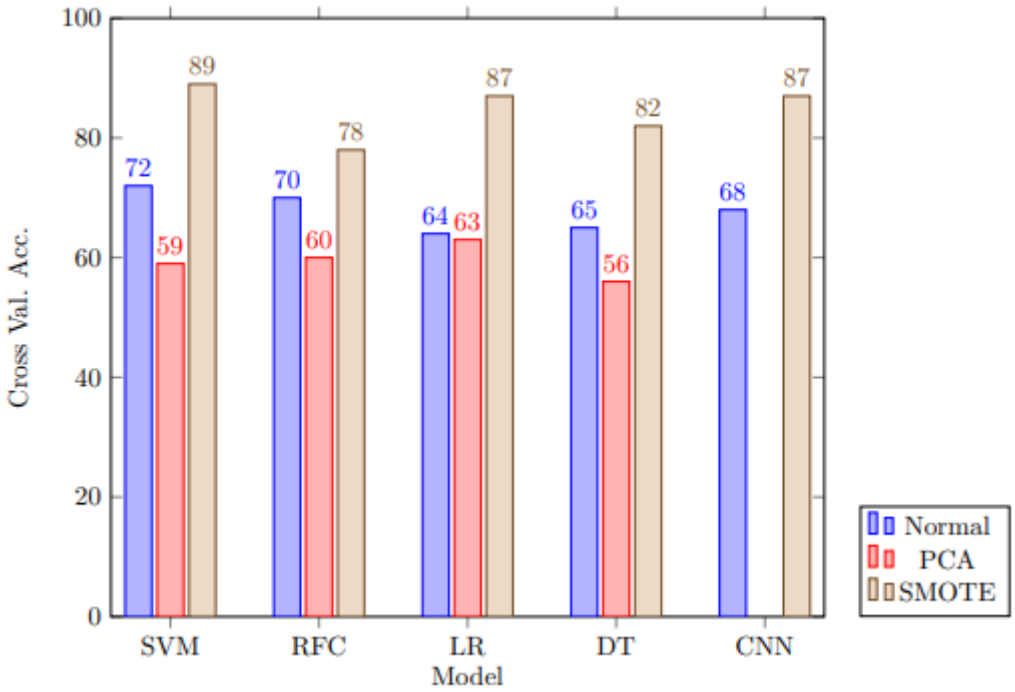


**Figure 4.1:** Bar chart has cross validation accuracy(cross val. acc.) percentage as its y-axis and the model type used for the x-axis. Each colour represents a different method used for each model.

We can see the results of preforming 5-fold cross validation on the training set of each of our models in Figure 4.1. The Figure shows the results for each method used and how

it affected the cross validation accuracy for our models. For our Normal approach SVM had the best cross val. acc. of 72% followed closely by RFC at 70% and CNN at 68%, which makes the SVM have the best accuracy in the Normal approach.
As for the PCA category we can notice that it did not improve the accuracy for any model when it is applied with its best accuracy being with the LR model at 63% with it being 1% lower than the LR normal accuracy. That might be because PCA usually lowers the dimensionality of the data by getting rid of low variance components which in our case might be discriminate information that the model needs to discriminate one class from another.
For our SMOTE category it caused a noticeable increase in accuracy across all models. Which is to be expected since it balances the classes which makes the models less biased. SVM ,LR and CNN had the best cross val. acc. in the SMOTE category at 89% ,87% and 87% respectively. Which makes SVM with SMOTE the best cross val. acc. across all models and methods. However, in order for us to get a better understanding of the results we need to look at the F1 accuracy score on our test data.

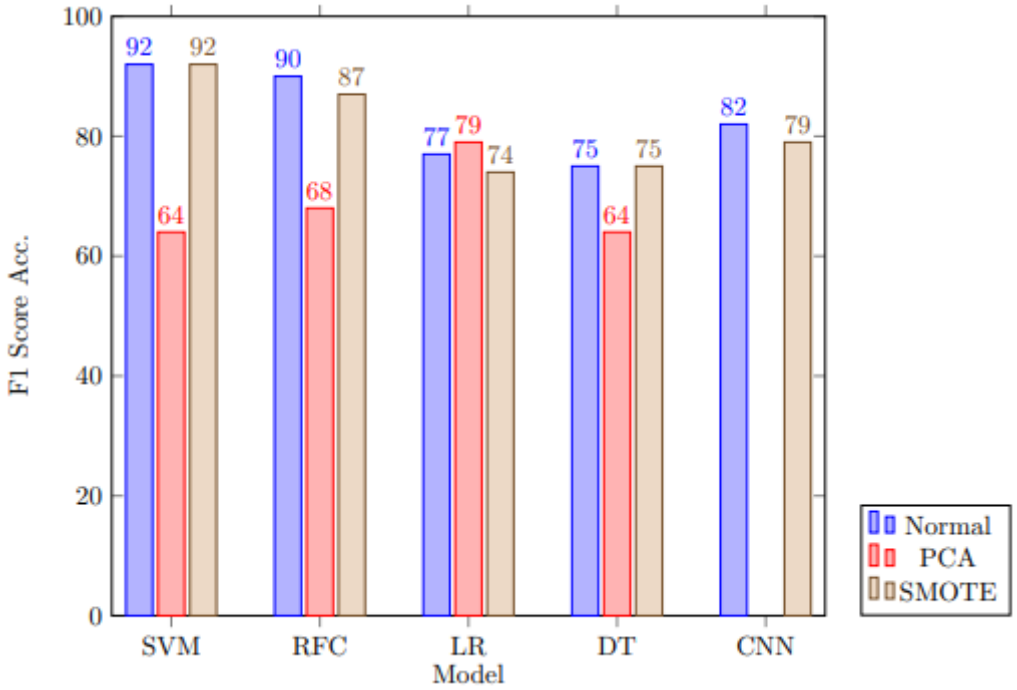


**Figure 4.2:** Bar chart has F1-score accuracy percentage as its y-axis and the model type used for the x-axis. Each colour represents a different method used for each model.

Figure 4.2 shows the accuracy F1-score of all the models with different methods. For the normal category without any PCA or SMOTE, SVM and RFC had the best F1-accuracy at 92% and 90% respectively beating CNN by 10% and 8%.
PCA category results are similar to the cross val. acc. where no improvement was seen for most of the models. However, the LR model had the best PCA accuracy at 79% with a small improvement over the normal method by 2%.

For our SMOTE category we can notice that it has different results across the models. For LR it decreased the accuracy by 3% compared to the normal method. As for the CNN and RFC it decreased by a noticeable 4% and 3% respectively. However, for SVM and DT it had no effect on the accuracy which makes it equal to the normal category across both models. Which makes the normal SVM and SVM with SMOTE both have the best F1-accuracy. but if we compare the models taking into account cross val. acc. and F1-accuracy we notice that SVM SMOTE is the best method followed by RFC SMOTE.

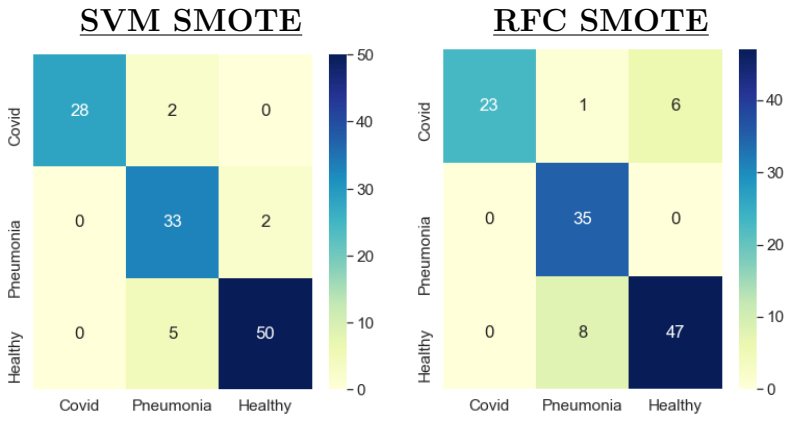


**Figure 4.3:** Figure shows frame level confusion matrix of SVM SMOTE and RFC SMOTE results on our 5 frames per patient test set.

For further comparison we can take a look at the frame level confusion matrix of both models in Figure 4.3 . For the Covid-19 class SVM managed to predict 28 correct frames out of 30 while RFC only predicted 23 out of 30 and miss predicted 6 frames as healthy frames. Which makes the SVM better than RFC at identifying Covid frames. For the pneumonia class RFC predicted all frames correctly while SVM miss predicted 2 frames as healthy. As for the regular class SVM got 50 out of 55 correct and RFC only got 47. Both models are close to each other in the pneumonia and healthy classes but what we care about is the Covid-19 class which SVM has better performance than RFC. Lets further compare all our models again but with patient level identification accuracy to see which one would preform best in clinical environment.

### 4.1.2 Patient Level

The test data used in the patient level is the same as the frame level the only difference is we aggregated the score of the frames of each patient into one prediction by taking the majority vote. So that it would mimic real life diagnosis. However, since our dataset only includes 122 patients and our test data has 24 patients in total (6 Covid, 7 pneumonia, 11 healthy). It will cause the models to have similar percentages since the data is quite low so the variance will also be low.
Figure 4.1 shows the accuracy of correctly predicted patients of each model with different

| Method | Normal | PCA | SMOTE |
|---|---|---|---|
| **Model** | **Patient Acc.** | **Patient Acc.** | **Patient Acc.** |
| **SVM** | **0.95** | 0.70 | **0.95** |
| **RFC** | **0.95** | 0.70 | 0.87 |
| **LR** | 0.79 | 0.79 | 0.79 |
| **DT** | 0.75 | 0.70 | 0.83 |
| **CNN** | 0.83 | - | 0.83 |

**Table 4.1:** Table shows patient level accuracy to see which model had the best preformance at indentifyin patients by taking the vote score for each video.

methods. As expected SVM, SVM SMOTE and RFC had the best results with 95% (23 out of 24) correctly predicted patients.
To know which patients these models have miss-identified we can take a look at their confusion-matrix in Figure 4.4. We can notice that although the models had different accuracy at frame level but at patient level they have the same results. All covid and pneumonia patients have been correctly identified and only 1 healthy patient was miss identified as pneumonia. And since in the medical world false positives are not as important as false negatives. The models look promising when taking the vote of different frames on each patient.

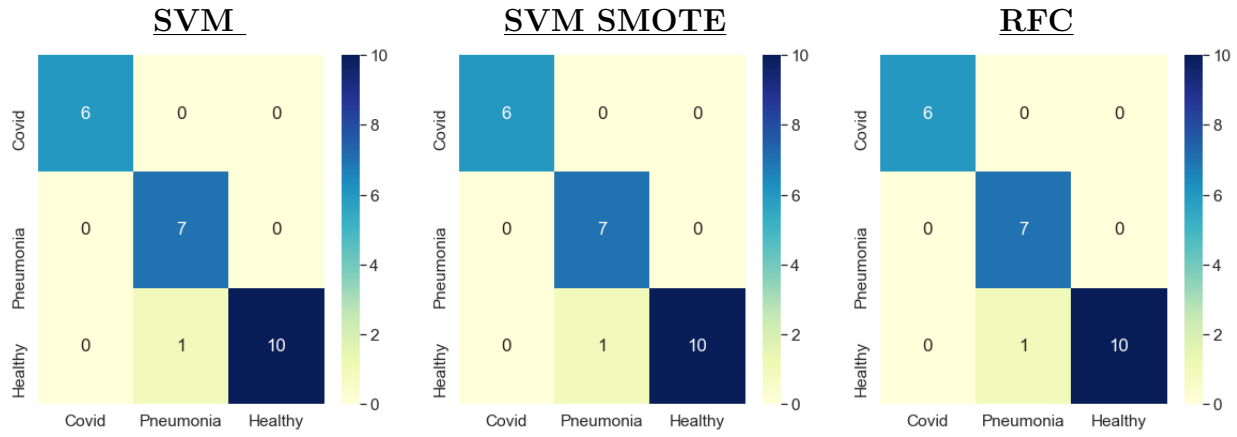


**Figure 4.4:** Figure shows frame level confusion matrix of SVM SMOTE and RFC SMOTE results on our test set.

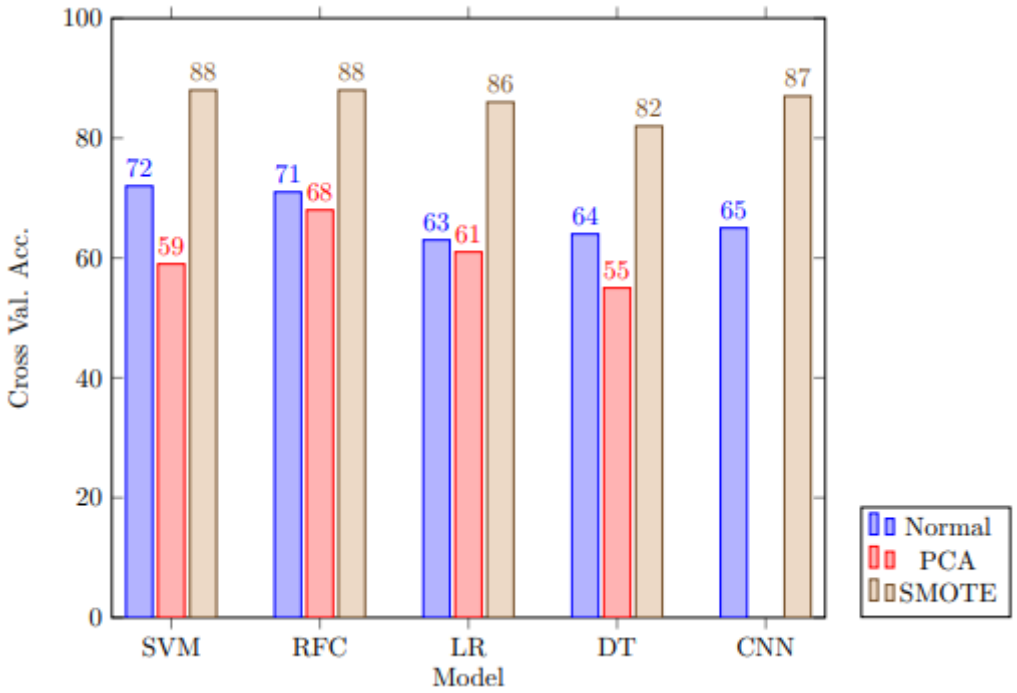


**Figure 4.5:** Figure shows cross validation accuracy when using 15 frames for each patient. Bar chart has cross validation accuracy(cross val. acc.) percentage as its y-axis and the model type used for the x-axis. Each colour represents a different method used for each model.

## 4.2 Fifteen Frames Per Video

### 4.2.1 Frame Level

Figure 4.5 shows the results of preforming 5-fold cross validation on our models train set using the 15 frames per patient dataset. The Figure also shows cross val. acc. for each method used in data processing. For the Normal category, SVM leads the rest with 72% accuracy followed closely by RFC with 71%.
PCA processing method have not caused any improvements with its best accuracy being with RFC at only 68%. As for the SMOTE method it can be clearly noticed that it caused an improvement across all models. The top 2 models are SVM and RFC at 88% followed by CNN at 87%. We also need to compare the results of the F1-score accuracy on the test set.

Figure 4.6 shows F1-score accuracy results for the models with different data processing methods when tested on our 15 frames test set. For the normal dataset the best two models are RFC and SVM at an accracy of 92% and 91% respectively followed by CNN at 86%.

As for using PCA it did not cause any improvements and had worse results than

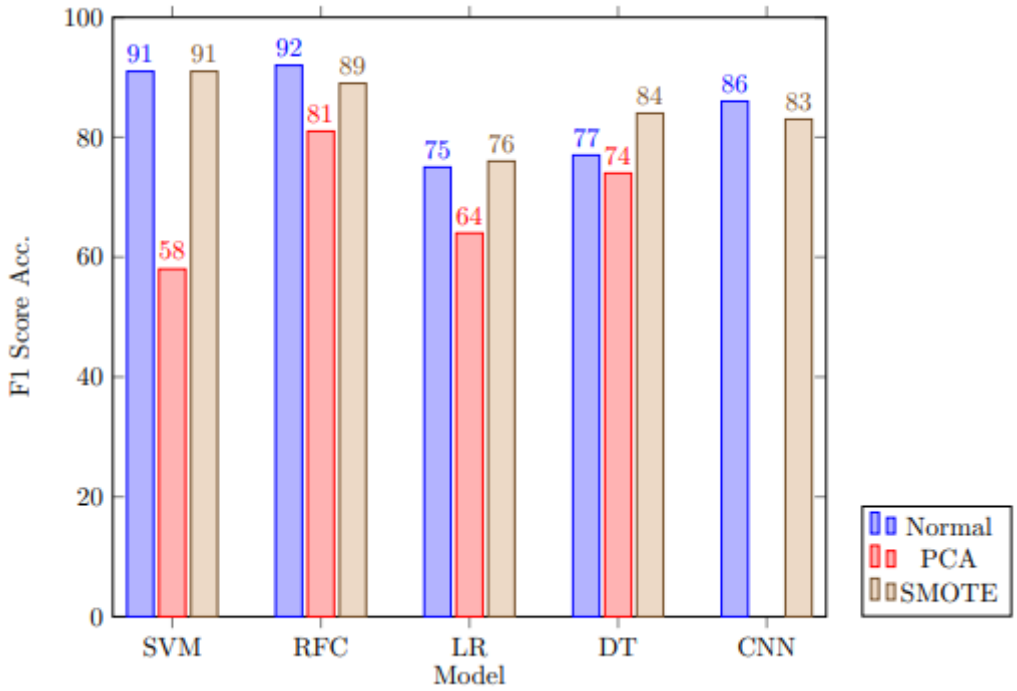


**Figure 4.6:** Figure shows F1-Score accuracy when using 15 frames for each patient. Bar chart has F1-score accuracy percentage as its y-axis and the model type used for the x-axis. Each colour represents a different method used for each model.

normal and SMOTE. The best PCA accuracy was achieved with the RFC model at 81% followed by DT at 74%.

When using SMOTE on our 15 frames dataset there is a clear accuracy improvement across all models except CNN which had a 3% drop in accuracy compared to the normal category. The best SMOTE accuracy was achieved by the SVM model at 91% followed by RFC at 89%.

If we compare or two best models, which are SVM SMOTE and RFC SMOTE using their confusion matrix seen in Figure 4.7 to decide which was better at detecting each class. RFC had 85 frames correctly detected as covid and 5 as healthy compared to only 81 frames correctly detected as covid with SVM which means RFC had better results in the covid class. However, in pneumonia detection SVM detected 99 frames compared to only 95 in RFC. As for the healthy class SVM had only 16 frames falsely detected as pneumonia compared to 23 frames with RFC. Which means that RFC had better detection for covid frames but RFC was better at detecting pneumonia and healthy frames.

### 4.2.2 Patient Level

For the patient level accuracy the three best models are SVM , RFC and SVM SMOTE. They all have same accuracies because as mentioned in the 5 frames section the test set

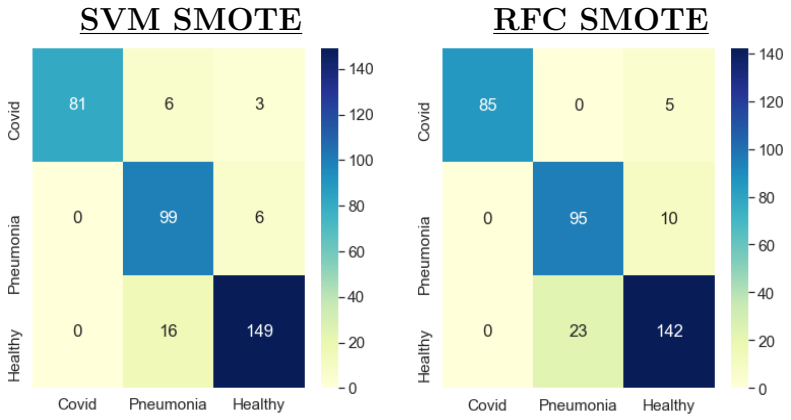


**Figure 4.7:** Figure shows frame level confusion matrix of SVM SMOTE and RFC SMOTE results on our 15 frames per patient test set.

| Method | Normal | PCA | SMOTE |
|---|---|---|---|
| **Model** | **Patient Acc.** | **Patient Acc.** | **Patient Acc.** |
| **SVM** | **0.95** | 0.58 | **0.95** |
| **RFC** | **0.95** | 0.79 | 0.91 |
| **LR** | 0.83 | 0.62 | 0.83 |
| **DT** | 0.79 | 0.83 | 0.87 |
| **CNN** | 0.75 | _ | 0.83 |

**Table 4.2:** Table shows patient level accuracy for 15 frames per patient models to see which model had the best performance at indentifying patients by taking the vote score for each video.

is quite small. and their confusion matrix results are exactly identical to the 5 frames results seen in Figure 4.4 which means that having 15 frames instead of 5 showed no difference when diagnosing patients by their whole frame score.

## 4.3 Results discussion

If we compare the models average F1-score accuracy in the normal category between 5 frames and 15 frames per patient the results are 83.2% for 5 frames and 84.2% for 15 frames. Moreover, if we compare the effect of PCA when used with 5 frames against with 15 frames the average PCA accuracy across all models is 68.75% for 5 frames against 69.25% for 15 frames.The average Smote accuracy of 5 frames is 81% and 84.6% for 15 frames.

The average F1-accuracy of all models across different categories may have increased by a small amount but in order to see if taking 15 frames would cause any difference in real life prediction we need to look at the F1-accuracy and cross val. acc of our best model

from both sides. Which are SVM SMOTE for 5 frames at 89% cross val. acc. and 92% F1-accuracy, and SVM SMOTE for 15 frames at 88% cross val.acc. and 91% F1-score. There have been a 1% drop in both the cross val and F1 accuracy when using 15 frames instead of 1% and these results are at frame level which means that although the models with 5 frames may use less data but it was successful at noticing higher number of frames and since we have seen that using 15 frames instead of 5 per patient showed no difference in patient predictions. We can conclude that SVM SMOTE with 5 frames per patient is the best model to be used.

## 4.4 GUI

The GUI of the application in Figure **??**, shows the result after the user has entered the lung ultrasound video and clicked the get diagnosis button. The GUI shows three tabs. Tab A, shows the input video the user chose to upload to the application. Tab B, shows the final result the diagnosis the CNN model have predicted for the input video. Tab C, contains the five frames selected at different intervals from the patient video, along with their prediction and Grad-CAM visual explanation. It can be seen that all frames were predicted as covid, and the highlighted areas in all frames mark the B-lines in the patients lung which are a sign of covid-19 infection. Which means that our CNN model was successful in noticing the B-lines patterns in different covid-19 patients.

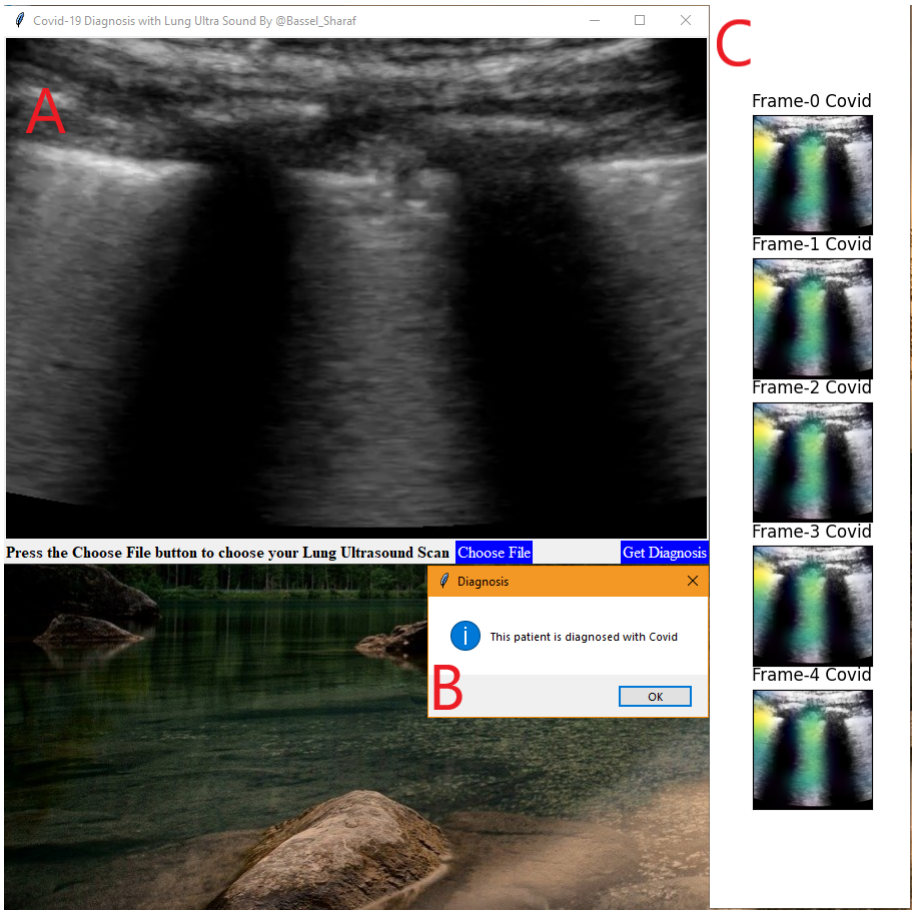


**Figure 4.8:** Image shows the our GUI showing final diagnosis result and the result of each frame prediction. Along with the highlighted areas which had the biggest weight in the prediction the model has taken.

# Chapter 5

# Conclusion

To sum up, SVM and RFC were the best performers with the highest accuracy across different processing methods. Moreover, they were able to correctly identify all covid patients in the test set and most of the covid frames with extremely low false identification rate. SMOTE increased the number of samples for under numbered frames which provided an overall higher accuracy in the cross validation of all models. Although, PCA reduced the number of components which made the training process easier for our models, but it did decrease the accuracy of most models. Which is why it is better not to used it with sensitive datasets in medical imaging. The five and fifteen frames per patient video did not show any clear change in accuracy, since LUS are usually low frame rate and short with repeating frames. So increase the number of frames taken from each video would did not increase the accuracy of the models. Lastly, the simple application which uses the CNN model with Grad-CAM shows a lot of promise. Since it clearly showed correct patterns that identify covid patients, which means the model has learned the correct patterns of the infection. The simple application could clearly help physicians with their diagnosis in terms of speed, accuracy and portability. In conclusion, the study shows that ML with LUS shows a clear promise in providing a quick, easy and low budget method to identify possible Covid-19 and pneumonia patients.

# Chapter 6

# Future Work

As the Covid-19 pandemic is still a recent event and its still new to medical research, there is a clear lack of databases with significant resources. Therefore, a larger dataset in the future will definitely aid in the results and show a clearer illustration of how the discussed methods can preform in a clinical environment. Moreover, from the prospective of ML several improvements are possible, like trying out other models like K-Nearest Neighbors and Naive Bayes. Furthermore, trying different splitting algorithms for the DT and RFC should be considered, to see if it will result in any improvement. Many, enhancements can be done to the CNN architecture and trying different DL methods should also be considered, like Recurrent neural networks. Spacial Transformer Networks should be experimented with, to enhance the geometric invariance of the model. Lastly, the application can have a better interface and should be deployed to the web and mobile environment. So that patients can upload their LUS for getting a diagnosis, or provide us with already diagnosed LUS videos to increase the available dataset and further improve our models.

# Appendix

# Appendix A

# Lists

nowadays

| | |
|---|---|
| **AHE** | Adaptive Histogram Equalization |
| **CNN** | Convolutional neural network |
| **CT** | Computed Tomography |
| **DT** | Decision Trees |
| **DL** | Deep Learning |
| **Grad-CAM** | Gradient-Weighted Class Activation Maping |
| **LR** | Logistic Regression |
| **LUS** | Lung Ultra Sound |
| **ML** | Machine Learning |
| **PCR** | Polymerse Chain Reaction |
| **PCA** | Principle Component Analysis |
| **RFC** | Random Forest Classifier |
| **SVM** | Support Vector Machines |
| **SMOTE** | Synthetic Minority Oversampling Technique |

# List of Figures


2.1 An example of Covid-19 infected lung ultrasound scan. Pleural lines marked with yellow arrows and consolidations marked with red arrows [10]. . . . . 4

2.2 Figure shows an image of a dog along with its pixel intensity graph before and after applying AHE. source . . . . . 5

2.3 On the left we have a 2D plot of a dataset showing its x and y variables. On the other side its the same plot of the dataset but after calculating the principal components pc1 and pc2. source . . . . . 6

2.4 Shows how the dataset in Figure 2.3 will look after using one of the components and removing the other. source . . . . . 6

2.5 The 2D Plot shows a imbalanced dataset plot. Where red points are the minority class and green points the majority class. source . . . . . 8

2.6 Same 2D plot seen in Figure 2.5 but after applying SMOTE. Small red dots are synthetic samples produced by SMOTE. source . . . . . 8

2.7 **a:** Hyperplane seperating two classes from a public dataset. **b:** 2D plot representation of dataset with a red point outlier. **c:** Separation of the dataset by a 1-Dimensional soft-margin [18]. . . . . 10

2.8 **a:** shows a graph of a a company dataset where Y is the given salary and X is the years of experience of each employee, the line fitted between them is the linear regression fit. **b:** is a plot showing how the the sigmoid function (S-curve) is plotted the green dots are data points for either 1 or 0 that where used to plot the S-curve. source . . . . . 10

2.9 A simple decision tree that answers the question of whether a given artery is blocked or not by following multiple decisions. . . . . 11

2.10 An overview of an RFC which fits dataset X to four decision trees. It then proceeds to take majority voting to decide the final result. . . . . 12

2.11 An overview of how neural network layers are interconnected. The blue nodes are the input layer, green nodes are the hidden layer and last is the output layer. source . . . . . 14

2.12 A plot showing how the model converges to reduce the loss function and reach the local minimum. The blue starting point is where the model loss function is at the start of the training. the green lines shows how the model keeps gradually converging with each training sample, to reach the minimum cost function. source . . . . . 15

2.13 A simple convolutional layer representation of an input image on the left and the filter in the middle being applied to a part of the image. Then the dot product is calculated and projected to the output array. source . . . 16

2.14 (a) is the input image with a cat and dog. (b) The result image from Grad-CAM after using CNN trained to identify dogs [21] . . . . . 18

3.1 A diagram showing an overview of our approach . . . . . 22

3.2 **Example lung ultrasound images from the convex probe data. A:** Shows a Covid-19 infected Lung. **B:** A pneumonia infected lung,with dynamic air bronchograms surrounded by alveolar consolidation. **C:** Healthy lung with normally aerated horizontal A-lines. . . . . 23

3.3 **A** Shows lung ultrasound sample before applying histogram equalization. **B** Same sample after histogram equalization. . . . . 24

3.4 Shows Frame count imbalance of the classes after extracting 5 frames per video . . . . . 24

3.5 Diagram showing the process which the frames go through before reaching Pipeline 1 classifiers . . . . . 25

3.6 Diagram showing the process which the frames go through before reaching Pipeline 2 classifiers . . . . . 26

3.7 Image shows the standard VGG-16 network architecture which is constructed from multiple convulotional blocks followed by max-pooling [27]. 26

3.8 Image shows an illustration of our CNN model architecture. . . . . 27

3.9 Image shows the GUI of the python application. It mainly consists of two buttons and middle screen to show the chosen video. The first button is the choose file button, that will allow the user to select a video. The get prediction button will start the application to diagnose the video. . . . . 27

3.10 A diagram illustrating the evaluation method done on our models. Showing 5-fold cross-validation on the test set and the final test on the test set [28]. 28

3.11 Figure demonstrates how the classifier will have a prediction for each frame from the patient video then all predictions will result in one final result by the majority vote. . . . . 28

4.1 Bar chart has cross validation accuracy(cross val. acc.) percentage as its y-axis and the model type used for the x-axis. Each colour represents a different method used for each model. . . . . 29

4.2 Bar chart has F1-score accuracy percentage as its y-axis and the model type used for the x-axis. Each colour represents a different method used for each model. . . . . 30

4.3 Figure shows frame level confusion matrix of SVM SMOTE and RFC SMOTE results on our 5 frames per patient test set. . . . . 31

4.4 Figure shows frame level confusion matrix of SVM SMOTE and RFC SMOTE results on our test set. . . . . 32

4.5 Figure shows cross validation accuracy when using 15 frames for each patient. Bar chart has cross validation accuracy(cross val. acc.) percentage as its y-axis and the model type used for the x-axis. Each colour represents a different method used for each model. . . . . 33

4.6 Figure shows F1-Score accuracy when using 15 frames for each patient. Bar chart has F1-score accuracy percentage as its y-axis and the model type used for the x-axis. Each colour represents a different method used for each model. . . . . 34

4.7 Figure shows frame level confusion matrix of SVM SMOTE and RFC SMOTE results on our 15 frames per patient test set. . . . . 35

4.8 Image shows the our GUI showing final diagnosis result and the result of each frame prediction. Along with the highlighted areas which had the biggest weight in the prediction the model has taken. . . . . 37

# List of Tables

3.1 Dataset size. Number of images and videos of every class per probe . . . 23

4.1 Table shows patient level accuracy to see which model had the best preformance at indentifyin patients by taking the vote score for each video. . 32

4.2 Table shows patient level accuracy for 15 frames per patient models to see which model had the best performance at indentifying patients by taking the vote score for each video. . . . . 35

# Bibliography


[1] Ariel Karlinsky and Dmitry Kobak. The World Mortality Dataset: Tracking excess mortality across countries during the COVID-19 pandemic. *medRxiv*, page 2021.01.27.21250604, June 2021.

[2] Hai-Yang Wang, Xue-Lin Li, Zhong-Rui Yan, Xiao-Pei Sun, Jie Han, and Bing-Wei Zhang. Potential neurological symptoms of covid-19. *Therapeutic Advances in Neurological Disorders*, 13:1756286420917830, 2020. PMID: 32284735.

[3] Thirumalaisamy P. Velavan and Christian G. Meyer. The COVID-19 epidemic. *Tropical Medicine & International Health*, 25(3):278–280, March 2020.

[4] Daniel B. Vinh, Xiao Zhao, Kimberley L. Kiong, Theresa Guo, Yelda Jozaghi, Chris Yao, James M. Kelley, and Ehab Y. Hanna. Overview of COVID-19 testing and implications for otolaryngologists. *Head & Neck*, 42(7), 2020.

[5] Tyler E Miller, Wilfredo F Garcia Beltran, Adam Z Bard, Tasos Gogakos, Melis N Anahtar, Michael Gerino Astudillo, Diane Yang, Julia Thierauf, Adam S Fisch, Grace K Mahowald, et al. Clinical sensitivity and interpretation of pcr and serological covid-19 diagnostics for patients presenting to the hospital. *The FASEB Journal*, 34(10):13877–13884, 2020.

[6] Mahmud Mossa-Basha, Carolyn C. Meltzer, Danny C. Kim, Michael J. Tuite, K. Pallav Kolli, and Bien Soo Tan. Radiology Department Preparedness for COVID-19: Radiology Scientific Expert Review Panel. *Radiology*, 296(2):E106–E112, August 2020.

[7] Michael B Weinstock, ANA Echenique, JW Russell, ARI Leib, J Miller, D Cohen, Stephen Waite, A Frye, and F Illuzzi. Chest x-ray findings in 636 ambulatory patients with covid-19 presenting to an urgent care center: a normal chest x-ray is no guarantee. *J Urgent Care Med*, 14(7):13–18, 2020.

[8] Stephanie Sippel, Krithika Muruganandan, Adam Levine, and Sachita Shah. Use of ultrasound in the developing world. *International journal of emergency medicine*, 4(1):1–11, 2011.

[9] Luna Gargani and Giovanni Volpicelli. How i do it: lung ultrasound. *Cardiovascular ultrasound*, 12(1):1–10, 2014.

[10] Gino Soldati, Andrea Smargiassi, Riccardo Inchingolo, Danilo Buonsenso, Tiziano Perrone, Domenica Federica Briganti, Stefano Perlini, Elena Torri, Alberto Mariani, Elisa Eleonora Mossolani, et al. Is there a role for lung ultrasound during the covid-19 pandemic? *Journal of Ultrasound in Medicine*, 2020.

[11] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[12] Dah-Chin Luor. A comparative assessment of data standardization on support vector machine for classification problems. *Intelligent Data Analysis*, 19(3):529–546, 2015.

[13] Murali Shanker, Michael Y Hu, and Ming S Hung. Effect of data standardization on neural network training. *Omega*, 24(4):385–397, 1996.

[14] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[15] Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine learning*, 42(3):203–231, 2001.

[16] Akila Somasundaram and U Srinivasulu Reddy. Data imbalance: effects and solutions for classification of large and highly imbalanced data. In *International Conference on Research in Engineering, Computers and Technology (ICRECT 2016)*, pages 1–16, 2016.

[17] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[18] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[19] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(6):275–285, 2004.

[20] IBM. What are neural networks?, July 2021.

[21] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[22] Jannis Born, Gabriel Brändle, Manuel Cossio, Marion Disdier, Julie Goulet, Jérémie Roulin, and Nina Wiedemann. Pocovid-net: automatic detection of covid-19 from a new lung ultrasound imaging dataset (pocus). *arXiv preprint arXiv:2004.12084*, 2020.

[23] Jannis Born, Nina Wiedemann, Manuel Cossio, Charlotte Buhre, Gabriel Brändle, Konstantin Leidermann, Avinash Aujayeb, Michael Moor, Bastian Rieck, and Karsten Borgwardt. Accelerating detection of lung pathologies with explainable ultrasound image analysis. *Applied Sciences*, 11(2):672, 2021.

[24] Subhankar Roy, Willi Menapace, Sebastiaan Oei, Ben Luijten, Enrico Fini, Cristiano Saltori, Iris Huijben, Nishith Chennakeshava, Federico Mento, Alessandro Sentelli, et al. Deep learning for classification and localization of covid-19 markers in point-of-care lung ultrasound. *IEEE transactions on medical imaging*, 39(8):2676–2687, 2020.

[25] Dave Sotelo. Effect of Feature Standardization on Linear Support Vector Machines, July 2017.

[26] Tom Howley, Michael G. Madden, Marie-Louise O'Connell, and Alan G. Ryder. The effect of principal component analysis on machine learning accuracy with high dimensional spectral data. In Ann Macintosh, Richard Ellis, and Tony Allen, editors, *Applications and Innovations in Intelligent Systems XIII*, pages 209–222, London, 2006. Springer London.

[27] Max Ferguson, Ronay Ak, Yung-Tsun Tina Lee, and Kincho H Law. Automatic localization of casting defects with convolutional neural networks. In *2017 IEEE international conference on big data (big data)*, pages 1726–1735. IEEE, 2017.

[28] Kumar Ajitesh. K-Fold Cross Validation - Python Example, August 2020.