Aalborg University Copenhagen
A.C. Meyers Vænge 15
2450 København SV

AALBORG UNIVERSITY
STUDENT REPORT

Semester:
ICTE 4

Title:
Pneumonary - A system that can detect potential pneumonia case from an X-ray image.

Theme:
Thesis Project

Project Period:
01.03.2022 - 02.06.2022

Participants:
Diganta Das
Study no. 20191157
Muheb Moshfiq
Study no. 20173803

Supervisors:
Morten Falch

Page Numbers: 79

Date of Completion:
June 1, 2022

Abstract:

This master thesis project is about Pneumonary- A machine learning system that can detect pneumonia from X-ray images. Pneumonia, a disease that is caused by bacteria and virus, is responsible for taking away thousands of lives every year. Especially in developing and overpopulated countries where the doctors are limited. This project tries to assist the healthcare system by creating a machine learning model that can accelerate the diagnosis of pneumonia. To make it possible, expert interviews have been conducted to get a picture of the present situation in different countries and analyze the need of assistance by such system and research on relevant theories, methods and related works have been done. By using ResNet101, it has been possible to get a maximum of 99% accuracy from the model. An interface has also been created for the purpose of showing the results where it's possible to choose an image and determined whether it is a pneumonia or a normal case.

Acknowledgement

This would not have been possible without the support and guidelines from our thesis supervisor Associate Prof. Morten Falch from Aalborg University, Copenhagen from the time of submitting thesis proposal until the final submission. We received support in the forms of regular advises and inputs which lead to the completion of the report successfully in a presentable form.

We would also like to express our gratitude to Associate Prof. Ming Shen who inspired us through his course Machine Learning, to Allan Hammershøj - COO and Co-founder of mediathand who inspired us for the idea.

Finally, we would like to express our gratitude to our friends and family for their support.

Contents

List of Figures

1 Introduction

Pneumonia is an illness of the lungs caused by bacteria, viruses, or fungus that can affect one or both lungs. An infection of the air sacs that causes pus and other fluids to fill the sacs is a dangerous illness and creates difficulty in breathing. Those of any age can get pneumonia, but there are some groups who are more at risk than others, including those over the age of 65, children under the age of 2, people with specific medical problems, and those who smoke [1].

On a global scale, pneumonia is responsible for more than 15 percent of all fatalities among children under the age of five. In 2015, the illness claimed the lives of 920,000 children under the age of five. According to the emergency department of "National Hospital Ambulatory Medical Care", in United States, there were registered more than 500,000 visits of pneumonia cases to the emergency rooms [2] and a report from "National Vital Statistics" shows that in United States more than 50,000 deaths occurred only in 2015 [3], maintaining its position among the country's top ten leading causes of mortality. According to the World Health Organization, pneumonia is the leading infectious cause of mortality in children under the age of five on a global scale. A total of 740 180 children under the age of five died as a result of pneumonia in the year 2019, accounting for 14 percent of all children under the age of five [4]. Despite the fact that pneumonia affects children and their families all over the world, South Asia and sub-Saharan Africa have the greatest rates of death from the disease [4]. It is possible that the situation will deteriorate even further in such countries due to a paucity of medical resources and personnel. For example, there is a shortfall of 2.3 million doctors and nurses across the 57 countries of Africa [5]. The precision and speed with which diagnoses are made are therefore essential determinants in the well-being of these groups. Individuals who are already battling with poverty can benefit greatly from this because it can assure fast access to therapy while also saving time and money.

Despite the fact that pneumonia is widespread, effectively identifying it is a difficult task, as there is need for a doctor to acquire and evaluate the patient's medical history and physical examination to initiate the diagnosis of pneumonia. However, it's not enough to confirm a potential case of pneumonia in a patient, therefore an X-ray image (CXR) scanning is required for the doctors to identify a potential case. The CXR image is being reviewed by highly skilled medical professionals, which usually look for some patterns that indicates a potential case of pneumonia, such as, looking for a white spot in the lungs, which is also called infiltrates [6]. In some cases, even a highly trained professional radiologist may have difficulty analyzing an X-ray image to determine whether a patient has pneumonia, resulting in delays in diagnosis or even misdiagnosing of the disease that increase the severity it and the associated mortality risk. As a result, early detection of illness is critical for medical professionals because it allows them to treat a patient at the earliest possible stage of the disease [7]. A research conducted by "US National Library of Medicine National Institutes of Health", where the medical specialist have examined 280 patients for detecting pneumonia by performing X-ray (CXR). The findings showed that 48 (17.1%) cases were misdiagnosed, where all of them were false negative cases [8].

On the other hand, artificial intelligence/machine learning technologies have the potential to analyze massive datasets and extract valuable insights that may be used to improve outcomes. This capacity is proven to be particularly useful in radiology and pathology, where it is currently being tested. Clinical imaging techniques such as magnetic resonance imaging (MRI), computed tomography (CT), and X-rays, as well as biopsy samples, allow doctors to observe the inner workings of the body. However, because these photos frequently include enormous quantities of complicated data, evaluating them can be challenging and time-consuming for human service providers. Artificial Intelligence (AI) solutions can improve the workflow of radiologists and pathologists by serving as clinical decision support and accelerating the delivery of treatment [9]. Hence, in this project, we will be trying to go through the methods and make a machine learning system that can efficiently detect pneumonia from chest X-ray images and provide an excellent accuracy.

1.1 Research Question

With the comparison of the two aspects- the traditional method of detecting a potential case of pneumonia, which is carried out by professional medical personnel, and the application of a machine learning method carried out by computer systems, we came up with the following problem formulation?

"How to create a machine learning system that can assist the health sectors in detecting pneumonia cases via chest X-ray of a patient?"

The following sub-questions will emphasize the above research question:

- Which machine learning techniques can be used to leverage the highest accuracy for identifying a potential pneumonia case?
- For the training of the machine learning model, what type of data is necessary, and how can it be accessed?
- How much research has been done in this field, and how far has the problem progressed in terms of resolution?

Note: For the sake of clarity, we will refer to the platform/system that we are creating as "Pneumonary" and the medical professionals who will be utilizing it as User(s), from this point forward.

1.2 Motivation

There has been a substantial amount of research into machine learning and artificial intelligence over the previous several decades. As a consequence of improvements in artificial intelligence (AI) and constant technological development, machine learning (ML) and deep learning (DL) algorithms can assess medical data more swiftly. Researchers are taking advantage of this general increase by building stronger machine learning and deep learning models to cope with medical data that is becoming increasingly difficult and voluminous. Our motivation of this thesis project is to develop a system, with the use of Machine Learning, that will aid in the diagnosis of heart diseases like pneumonia, and will be useful to the health-care industry. As can be seen, there have been a significant number of pneumonia instances reported, and the healthcare sector has a large amount of X-ray picture data at their disposal. These data can be used in a variety of useful ways by extracting the relevant information from them. As a result, sufficient data can be found to enable the construction of a Machine learning system for it. Additionally, the Machine Learning course that we took as part of our master's degree has provided us with the motivation to continue working with it and to apply it to our future career endeavours in various fields.

1.3 Delimitation

In the beginning of the project, we have tried to contact several experts in hopes that we would be able to get enough data. We were optimistic to have an opportunity to contact one of the experts in health care sector in Denmark but unfortunately it didn't happen. As a result, we worked on the dataset that was available for us to work on. Since the data we could collect wasn't sufficient to have an excellent accuracy in terms of distinguishing between viral and bacterial pneumonia, we eliminated the idea and decided to focus on combining both types of pneumonia into one class, which led us to create two distinct classes, which we called "Pneumonia" and "Normal". We also have not included research on some other machine learning methods, techniques and features that are not relevant for the development of our machine learning model. We only focused on researching on the relevant theories and methods.

It was abundantly clear from the start of the project that the technical aspect of machine learning and the analysis of relevant theories should be the primary focus of our attention. We will not conduct any research or analysis on how to generate revenue from this solution because it is only a prototype and is not intended to be released to the public. Therefore, creating a business model is not a part of this thesis project.

1.4 Expected Outcome

By the end of this project, we hope to have a system that works, which implies that we will be working towards making a machine learning model that would generate a high accuracy and consistency in terms of predicting pneumonia and also distinguish between a pneumonia and a normal case. After that we would work towards developing an interface so that one can interact with the system. The primary goal behind the development of an interface is to show the results generated by the trained machine learning model. We wish to deploy more than one method to train our model and choose the one that has greater performance and consistency. So finally we would like to have our system that we named "Pneumonary", ready and document all the progress and steps as we proceed further.

2 Methodology

In this section, we will be going through how we have tried to solve the problem formulation by adopting different approaches.

2.1 Creative process

When we started this project, we knew we wanted to work with machine learning and with some research done we focused on doing something with image recognition. We went to work and started researching on relevant subjects and found out that there is a crying need for the automation of the detection of many X-rays related diseases. We found it interesting and we chose to work with pneumonia detection. We also had to figure out which kind of methods would be relevant, efficient and doable given the time. We found out about the scarcity of medical data because of privacy issues and we continued to look for further information. Along the road, conducting expert interviews became a part of our project during discussions with our supervisor. We also believed that this could give us more insights about the health care sector. Also, after a lot of discussion, a need for including an interface that could show the result was generated to be able to make our system more understandable. The goal was to make the interface simple and easy to understand for now.

2.1.1 Process model

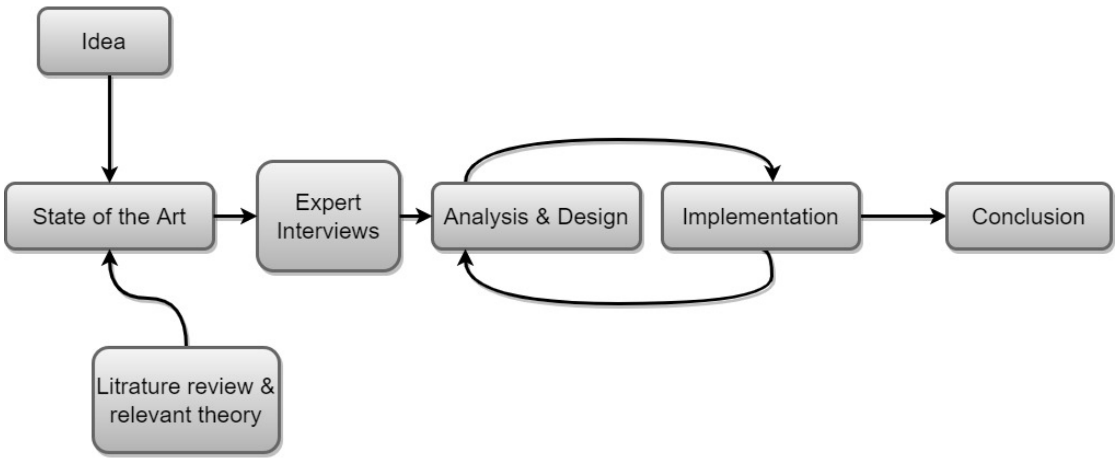


Figure 1: Process Model

Figure 1 shows the flow of the process that will be followed throughout the project. After our idea was formulated, to put the idea into work, we have done literature reviews and research on relevant theories and also conducted interviews. As our project deals with machine learning technologies, methods and processes for image detection. We combine state of the art technologies with a light touch of software development for the creation of an interface. The final design, requirement specification and system architecture were done based on the research that have been done throughout the project period and the data we could collect. What we must needed to solve the problem formulation, was a good understanding about the area we were working on and this is why we had to go through a lot of desktop research, relevant theories, books and other research on the same field, the majority of which has been illustrated in the state of the art and literature review chapter. We also looked into the healthcare situation of different countries and got knowledge about how they tackle such situations.

Interviews are a great way to get detailed information and knowledge from experts which can be prepared in different ways such as structured, unstructured and semi-structured. Our project will be applying semi structured interviews with experts. The people we have contacted and sought after have expertise in the healthcare sector and it was necessary for our project to get an overview from the experts. It also opened an opportunity for getting the inside knowledge of healthcare sector in different countries which we wouldn't be capable of getting to know on our own. The semi structured interview was designed in such a way that would give the interviewer some specific predetermined questions and the flexibility to add additional questions or off topic information to discuss [57].

After that, based on the research, we will carry out an analysis and implementation which have been an iterative process. As things progressed further, we had to go back and forth from analysis to implementation to adjust some of the things. The analysis is carried out on the basis of the research done in the previous sections. The analysis's purpose is to discover and describe how we obtained the requirements, including where we found them, such as, state of the art, literature review, or interview. Following the definition of the requirements, the MoSCoW tool is used to rank the requirements in descending order of importance, beginning with the most important and progressing to the least important. This is done in order to discover and implement the requirements that are necessary for our project to function. This will allow us to save a significant amount of time on requirements that are not critical and without them, the project will still run smoothly. To move this project forward, various types of UML diagrams [58], such as use case diagrams, sequence diagrams, context diagrams, and architecture diagrams, are going to be added that will provide useful information about the overall process of implementing the system.

2.1.2 Machine Learning steps

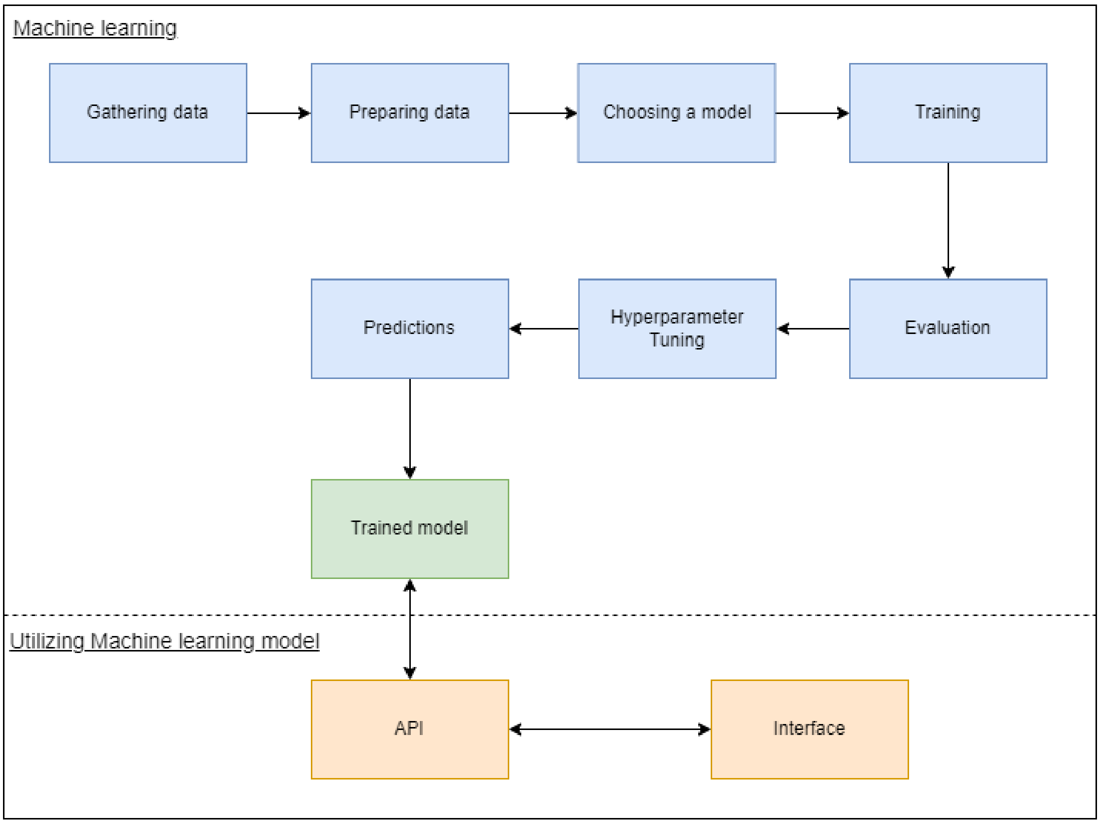


Figure 2: Machine Learning Model

Figure 2 depicts the steps involved in the development of our machine learning model. The machine learning model is created in seven steps: gathering data, preparing data, choosing a model, training, evaluation, hyperparameters tuning, and making predictions. When developing a model for machine learning, it is critical to use a process model because doing so ensures that we develop a model that takes into account all of its aspects. When working on a machine learning project, the following sequence of steps is recommended as best practices and is used by machine learning experts[15][16]. After creating the machine learning model, the model is saved in a file. This file is then consumed by an application programming interface (API), which utilizes the model to show predictions based on the X-ray images provided through the interface by users on demand.

3 State Of The Art

In this section, there is given an insight on the existing solutions that are currently in use and how they might be applied to our project.

3.1 Machine learning technologies

3.1.1 Basic concepts

There are several definitions of machine learning, and one of them can be defined as such, ”*Machine learning teaches computers to do what comes naturally to humans and animals: learn from experience. Machine learning algorithms use computational methods to “learn” information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases*” [10]. The concept of machine learning encompasses a wide range of concepts, but to put it into proper perspective, it can be divided into two main categories. The first section is referred to as Supervised learning, and the second section is referred to as Unsupervised learning. The terms Traditional Machine Learning and Deep Learning are also used to refer to these techniques in some of the literature [11].

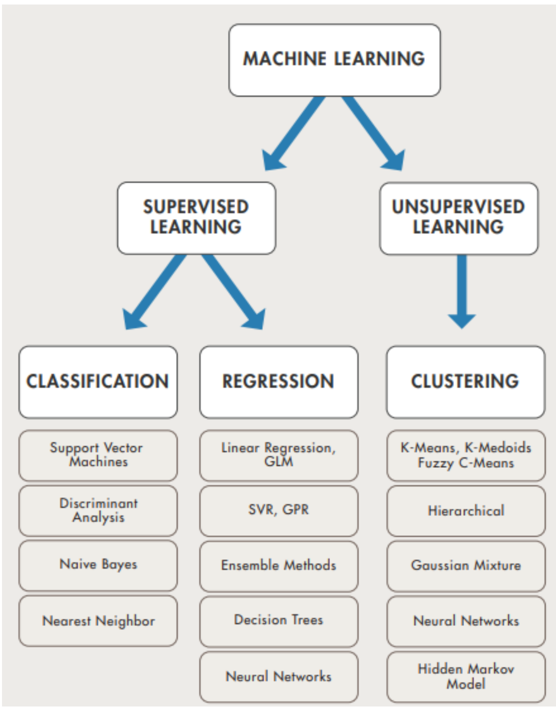


Figure 3: Machine Learning Models [10]

As illustrated in figure 3, there can be found various types of the algorithms that can be used for creating a machine learning model. The most significant distinction between supervised and unsupervised learning is that supervised learning uses labeled data-sets, whereas unsupervised learning does not. These data-sets are intended to train or "supervise" algorithms in order to improve their accuracy in categorizing data or anticipating outcomes. Using labeled inputs and outputs, the model's accuracy may be examined and improved over time. Unsupervised learning, on the other hand, employs machine learning algorithms to examine and cluster unlabeled data-sets. These algorithms can find hidden patterns in data without requiring human participation. Moreover, Supervised learning is separated into two parts: classification and regression. When comparing regression with classification, the most notable difference is that in regression, the quantity can be predicted, but in classification, the labels can be predicted. Making a decision on which algorithm would work best for a project's aim might be a tough one to make. This is also demonstrated in the book, *"There is no best method or one size fits all. Finding the right algorithm is partly just trial and error—even highly experienced data scientists can't tell whether an algorithm will work without trying it out."*, this also explains why it is difficult to determine which algorithm would be the most appropriate until it has been tried and used [10].

A. Supervised learning

Support vector machines: One of the techniques that is widely favored by many is the support vector machine, which produces substantial accuracy while requiring little computational power. SVM is an abbreviation for support vector machine and it may be used for both regression and classification applications. However, it is frequently employed in classification aims. Typically, an SVM model maps the data points into a higher-dimensional space, where it divides the data points into two classes based on the biggest distance between them using a hyperplane [12].

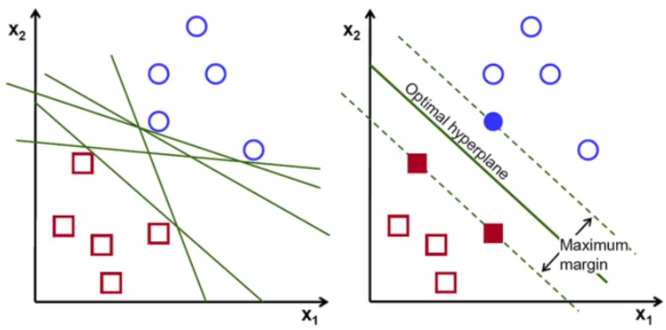


Figure 4: SVM possible hyperplanes [13]

There are several hyperplanes from which to pick in order to divide the two groups of data points. The purpose is to choose a plane with the biggest margin, defined as the maximum distance between data points from both categories. By increasing the margin distance to its maximum value, some reinforcement is provided, allowing following data points to be classified with more accuracy. Transverse hyperplanes are decision boundaries that help in data point categorization. Based on their position on the graph, points on each side of the hyperplane can be allocated to distinct classes. Furthermore, the number of features defines the dimension of the hyperplane. If there are just two input features, the hyperplane is nothing more than a straight line. The hyperplane degenerates into a two-dimensional plane when the number of input characteristics approaches three. They are closer to the hyperplane and have an effect on its placement and orientation in regard to that plane. Using these support vectors, it may enhance the classifier's margin as much as feasible [14].

Decision trees: Decision trees are another machine learning model that employs a supervised learning method and is most commonly used for binary classification [14]. Decision trees are useful for binary classification because they interpolate learnt knowledge from a dataset into a tree guided by if-then rules. The learning variable assesses how effectively each node in the tree can categorize the labeled data using the information gain and entropy of each node in the tree to estimate the accuracy of each node. The best node is chosen as the parent node, while the child nodes retain the possible values of the given input data. The entire procedure is recursively repeated endlessly until there are no more splits. Even if they have no prior knowledge of the dataset, DTs may select the most matched features for it [14].

Random forests Random Forests (RFs), which are a collection of decision trees that vote collectively for the classification aim, are used to classify unknown data points. The categorization job is chosen by which class receives the most 'votes' for the unknown data point. In other words, unlike a single decision tree, the random forest gathers forecasts from each tree and predicts the final output based on the predictions with the most votes from the majority of the trees. To avoid being confined to a single decision tree, the random forest collects forecasts from each of the tree and predicts the final output based on the predictions with the most votes from the voters. The mean of the predictions given by the trees in the random forest is computed. Eq. for a random forest with m trees and individual weights Wj, where m = number of trees and individual weights Wj. Because a tree becomes deeper over time, there is more overfitting in the training process, which implies that for a little change in the input, there is a higher or larger variance. Recurrent networks (RFs) receive vectorized input, with each decision tree attempting to classify a different segment of the vector input. The functionality of decision trees in RFs is merged by passing the input vector through each decision tree in the forest and having each decision tree classify the input vector based on a specific component of the vector that they receive as an input. The RF then chooses the class with the most 'votes' as the classification output for the input vector, or it takes the average of all 'votes' as the classification conclusion. Because RFs contain a large number of decision trees, the variance (overfitting) issue that may come from utilizing a single decision tree is reduced. The voting procedures can be changed if there is a considerable gap between the classes [12].

B. Unsupervised learning

Clustering: Data clustering is the problem of categorizing a population or a set of data points into a number of groups so that data points in the same group are more similar to other data points in the same group and more dissimilar to data points in different groups. In essence, it is a collection of goods that have been categorized together based on their similarities and differences. This method is quite important since it determines the intrinsic grouping that exists among the unlabeled data that is now available. There are no well-established criteria for successful clustering in the scientific community. In order to establish whether or not a need has been satisfied, it is up to the user to specify what criteria will be used to make this determination. Examples include finding representative data objects for homogeneous groups (data reduction), discovering "natural clusters" and describing their unknown properties, discovering useful and appropriate groupings, and discovering unusual data objects. There are several assumptions that must be made in order for this approach to work, and each assumption leads in a different but equally acceptable cluster formation for the points [14].

K-means clustering K-means is the most well-known and commonly utilized of the unsupervised learning algorithms. The purpose of this technique is straightforward: partition the data space in such a way that data points inside the same cluster (intra-class similarity) are as similar as possible, while data points from other clusters (inter-class dissimilarity) are as dissimilar as possible (inter-class similarity). Each cluster is represented in K-means clustering by its center (also known as a "centroid"), which corresponds to the arithmetic mean of the data points assigned to that cluster. Centroids are data points that represent the mean of a cluster and are not always connected with a member of the dataset in which they occur. As a result, the approach iterates until each data point is closer to the centroid of its own cluster than it is to the centroids of other clusters, lowering the intra-cluster distance between each data point at each step. It detects a given number of clusters within an unlabeled dataset using an iterative process, and then provides a final grouping based on the number of clusters chosen by the user (represented by the variable K). It recalculates new centroids repeatedly after starting with randomly selected data points as recommended centroids of the groups, until the data points are grouped into a final clustering using K-means [14].

3.2 Image recognition methods and Platforms

The approaches that may be employed for image recognition methods will be explored in this subsection, as the project's goal is to identify pneumonia cases using X-ray (CXR) pictures.

3.2.1 Deep Learning

Deep Learning is a machine learning technique that could basically be stated as Learning from examples. By filtering inputs through the layers, a computer learns how to do prediction and classification of information. Deep Learning tries to mimic how a human brain uses its neurons to process information. The idea behind a deep learning algorithm is to connect nodes in each layer and train a computer to learn by itself. Modern deep learning architectures are mostly relied upon artificial neural network which is ANN in short and CNN or convolutional neural network, a subset of ANN, is a popular method of deep learning. Once the architecture has been created for a neural network, it can start to learn. After telling a neural network about the input and desired output, the neural network starts learning on its own and for the next input, it can distinguish the output. Many layers of non linear processing units are used for the purpose of better feature extraction and image classification where each output produced by a layer works as an input for the next layer. By transforming inputs from each level to more composite and abstract representation, a learning system is established and a hierarchy of concepts is formed[20].

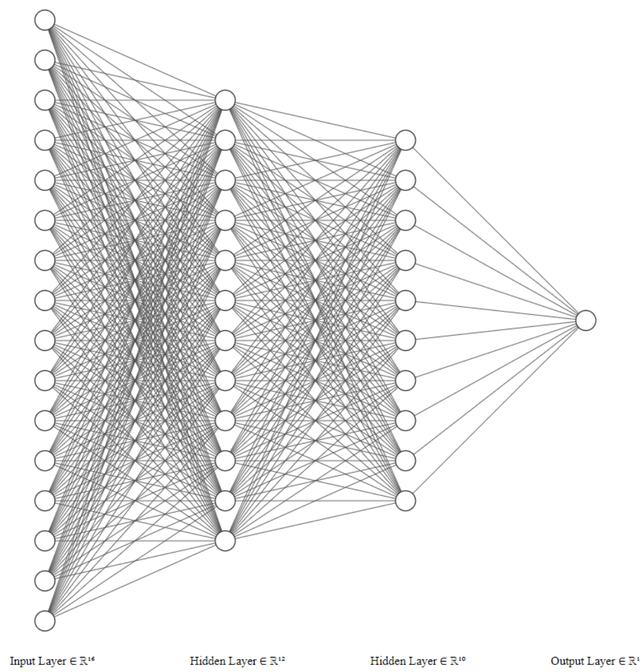


Figure 5: An Abstract Representation of Deep Neural Network

Figure 5 shows an abstract visualization of a deep neural network. Each input layer has an initial value. After multiplying the initial value with a weight, it is added to the hidden layer as an input. Then the values are calculated through linear algebra and an activation function that helps understanding the non linearity of the real world.

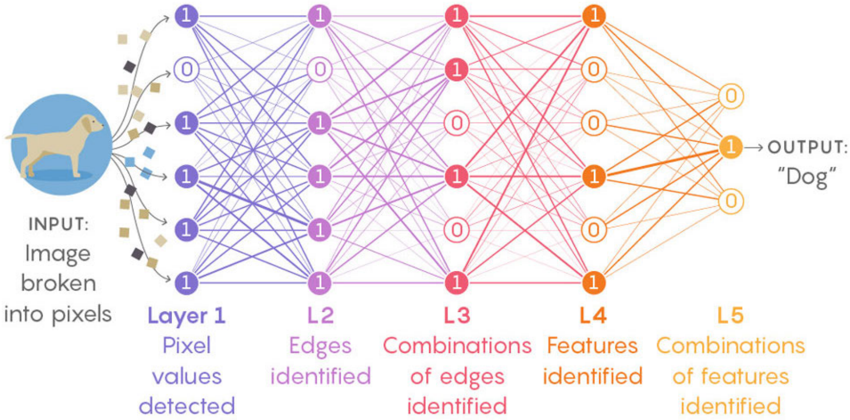


Figure 6: Deep Neural Network Learning and Predicting stages[21]

The basic structure of a neural network consists of Input neurons, output neurons and hidden layers between the input and output neurons. Input neurons are basically the number of features the neural network uses to generate predictions. One input neuron per feature is required by the input vector which can be the number of relevant features in a dataset. The features have to be selected very carefully. Any feature that might characterize patterns beyond the training set and lead to cause an overfitting issue, should be omitted. The output neuron stands for the number of predictions that are needed to be made. The probability of the positive class is represented by one output neuron per positive class for binary classification. To ensure the final result of the probabilities sum to 1, an activation function such as Softmax activation has to be used for multi-class classification where an instance can be a different image than the other. For hidden layers, the number of hidden layers between input layer and output layer depends on the complexity of the problem and the architecture of the neural network. For most problems and simpler training models, upto 5 hidden layers can generate satisfactory results. But to work with image or speech data, there can be dozens to hundreds of layers although all of them might not be fully connected. To get a performance boost, it is better to add more layers than adding more neurons in each layer[21].

Activation Function

An activation function calculates the weights and adds bias and then takes the decision of activating a neuron. The activation of a neuron entirely depends on the activation function. As the real world is hard to explain by linearity, the activation function introduces non linearity.

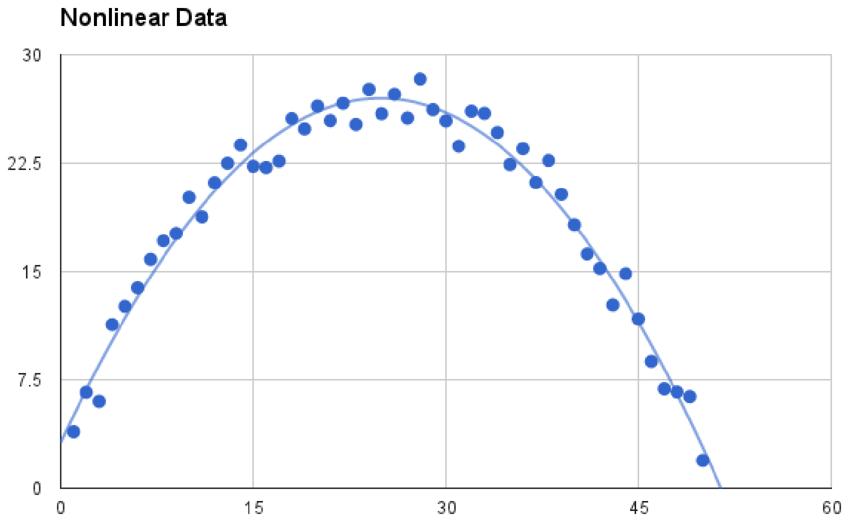


Figure 7: An example of Non-Linear Activation Function[25]

On the basis of the error at the output, the weights and biases of the neurons are updated in a neural network which is known as back-propagation. Performing back propagation to update the weights and biases are made possible by an activation function as it supplies the gradients along with the error. And by introducing non-linearity, it makes the neural network capable of learning and performing complex tasks. Linear function, Softmax function, Tanh function, RELU, Sigmoid function etc are examples of activation functions in deep neural network[23]. One of the most popular activation functions is hyperbolic tangent function which is derived as the following equation:[24]

$$tanh(x) = \frac{e^x - e^- x}{e^x + e^- x} \tag{1}$$

Below in the figure 8, example of two activation that produces non-linearity is shown:

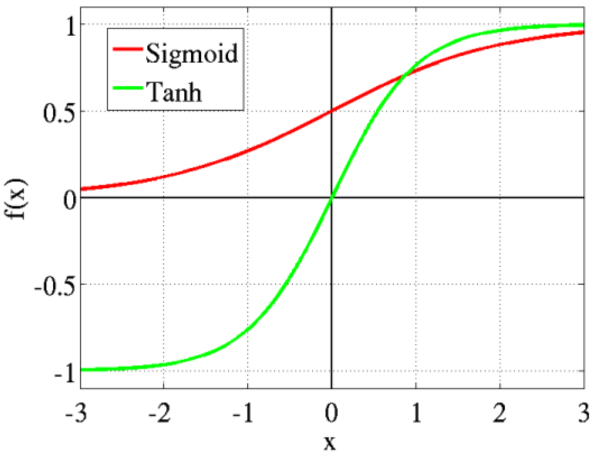


Figure 8: Example of two activation function(Sigmoid and Hyperbolic Tangent Function)[25]

Loss Function

A loss function is a method that demonstrates how well an algorithm models a dataset. The output of a loss function provides a higher number if the prediction is totally off and for a better prediction, the loss function provides a lower number. So, the lower the output gets, the better prediction can be achieved. The loss function measures the absolute difference between the prediction and actual value. It doesn't matter how high or low the prediction is, what matters the most is how much incorrect it is. But all loss functions don't have the same feature, in fact depending on the domain and uniqueness of the problem, the loss functions can vary quite significantly. Mean squared error, likelihood loss, log loss etc. are examples of different loss functions[22].

Batch Size

The number of samples from the dataset that will be passed through the network at a time is known as batch size. During training, if the batch size is larger, each epoch can be completed quickly by a neural network model. This of course, depends on the computational resources which might be able to process more than one sample at a time based on its configurations. But there is a huge drawback for training samples in larger batches. Even if the machine can handle more samples or larger batches at a time, it is more likely that the quality will deteriorate ending in not being capable of generalizing the data as expected. Batch size is a hyperparameter and it needs to be tested and tuned as well based on the performance of the designed model. This is why generally different batch sizes are used for testing for optimal resource utilization. Often batch size and number of epochs are mistaken to be the same which in reality are quite different. It can be shown as: batches in epoch = training set size / batch size[26].

Number of Epochs

When the neural network is trained with all the training data from the dataset for one cycle, it is considered to be completed an epoch. All the samples need to go through a forward pass and a backward pass for an epoch and the passes together are counted as one pass. Depending on the batch size, the iterations of an epoch may vary. Similarly, the number of epochs can be more than one which means the neural network is fed the same samples more than once for satisfactory performance[27].

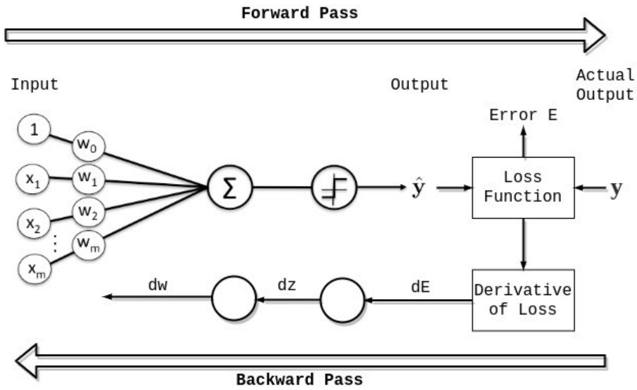


Figure 9: Cycle of an Epoch[27]

Learning Rate

Learning rate is considered to be a crucial hyperparameter that needs to be tuned and tested in order to get excellent results for a neural network. It helps to determine step sizes that need to be tuned at every iteration while training and also it progresses towards an optimum loss function ensuring a better performance. A gradient descent optimization algorithm is used to get the weight parameters and the bias parameters updated and the gradient of the loss function is found by the algorithm. In short, learning rate is the amount that updates the weights and bias parameters. The value of learning rate can be in the range between 0 and 1 and the symbol that represents the learning rate is α. The weights and bias parameters are updated by the following mathematical equation[28]:

$$W = W - \alpha * \frac{\partial L}{\partial W} \tag{2}$$

$$b = b - \alpha * \frac{\partial L}{\partial b} \tag{3}$$

Where L, W, and b represents the Loss function, weight and bias parameters respectively. As the learning rate being a tuning parameter controls the rate of learning of the model, finding the optimal value for a learning rate can be extremely difficult. A low learning rate indicates of a slow and careful learning by the model which can take a long time while a high learning rate gives indication of a fast learning model that might end in overshooting the minimal points[28].

3.2.2 Inception

Although machine learning models at present can gain or even exceed human level in classification or detection of objects, Inception method being developed in 2014, still remains a high performing method. The concept of inception method can be depicted as layers within layers as a neural network needs to be large in order to gain high performance. Humans can identify patterns at different scales because of how their cortex functions creating larger object perceptions and multiscale convolutional neural networks are also designed the same way and they generate the potential to learn further. Besides providing high performance gains there are other benefits of using the inception method such as[35]:

- Makes the neural network capable of extracting features that vary in scales and it can be possible by the proper utilization of varying convolutional filter sizes[35].
- The inception module can be unpacked and understood very easily and this is why researchers with little experience can also use this model to build a machine learning model[35].
- The inception module can help to understand the basics and formulation of modern day deep neural networks[35].

3.2.3 Residual Network (Resnet)

For complex problems such as image classification and image recognition, it has been possible to get outstanding outcomes with the introduction of deep convolutional neural networks resulting in a breakthrough in the field of computer vision. This has created a tendency for the researchers to create more complex and deeper neural networks by addition of more layers to the neural network ameliorating classification and recognition accuracy. But more layers create difficulties in training the whole neural network and ends up introducing saturation and probable degradation in accuracy. To solve this problem Residual Network or ResNet was developed which is a learning framework that doesn't make additional stacked layers fit a predefined model rather they explicitly fit a residual mapping. This basically about making a direct connection skipping some of the layers which as a result changes the output of that layer and it can be realized by feedforward neural network with shortcut connections formulating the equation F(x)+x [32][33].

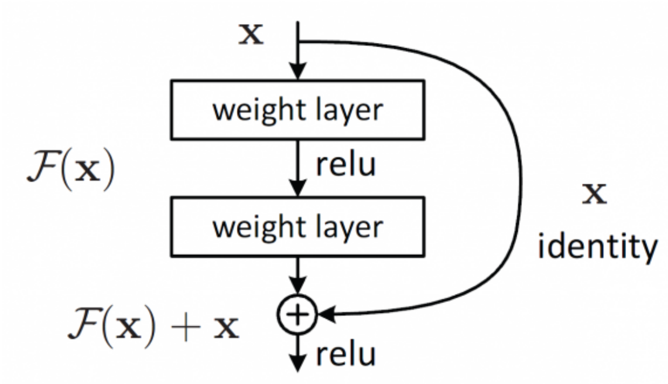


Figure 10: Skipping Layers and creating a direct connection in Residual Network [33]

To solve complex problems, most of the time additional layers are stacked and the intuition behind it is that adding more layers make them progressively learn features that are complex. As a result, these additional layers start to detect edges, textures, objects respectively and may go on. Resnet 50,101,152 respectively construct 50,101 and 152 layer Resnets[32][33]. As shown in Figure 11, error% decreases with the increase of added layers. This model can be ideal for building a system with higher accuracy.

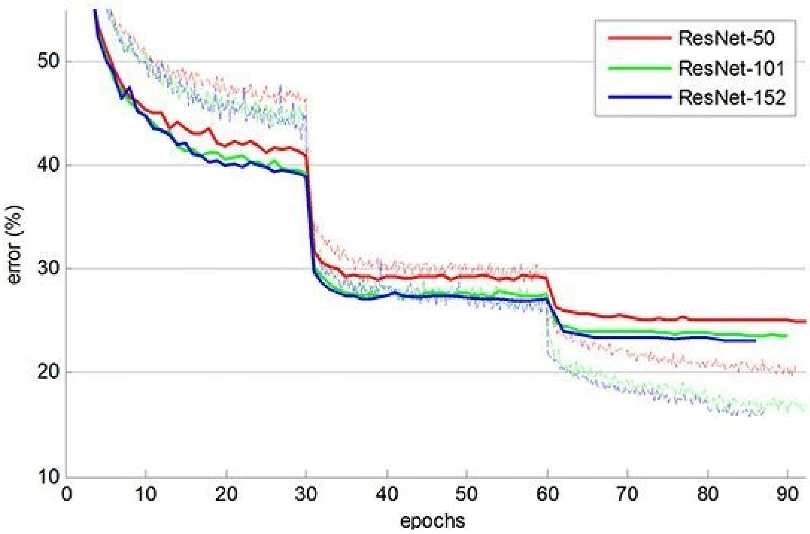


Figure 11: Error% decreases with addition of more layers (Resnet 50,101,152) [33]

3.2.4 Tensorflow

Although machine learning is complex, implementation of machine learning has become far less stressful and difficult because of Googles Tensorflow. It is a machine learning framework that eases the process for numerical computation and machine learning on a large scale. Machine learning in Tensorflow is done via python programming language which acts as a traffic between pieces as the mathematical equations in the library are written in C++ providing high level programming abstraction[34].

A pre-trained model is a network that has previously been saved and trained on a large dataset, typically for the purpose of a large-scale image-classification task. One can either use the pre-trained model in its current state or use transfer learning to tailor this model to a specific task. The concept of transfer learning for image classification is based on the idea that if a model is trained on a large and diverse dataset, this model will effectively serve as a base model. Following that, one will be able to train on top of these pre-trained models as a starting point, which will generate newly learned feature maps without having to start from scratch. This enables the users to modify previously acquired knowledge. The initial convolutional network includes some features that are already useful for image classification in general, makes the transfer learning model to identify relevant patterns and increases the performance [59].

4 Literature Review

For image analysis and classification, artificial intelligence has been playing a significantly prominent role when applied to detect and diagnose chest-related diseases. Albahli et al. discussed about state of the art methods working greatly in some cases but in some specific cases, they fail to provide desired outcomes as there is always a scarcity of available data and also a lack of balance in the available data. However, in their research, they have proposed three convolutional neural network architectures. The model they designed were DenseNet121, InceptionResnetV2 and ResNet 152V2 and they gained a score of 0.80 overall in accuracy. However, throughout their process they have faced significant problems with their model due to insufficient data and this is why to get better accuracy with high effectiveness they had to overcome the problem by considering the generalization of data by data augmentation. Their data was distributed as 80% for training and 20% for testing. From their results they found out that among the 3 CNN models they designed, DenseNet 121 and InceptionResNetV2 were the most efficient. The models were trained with a learning rate of 0.001, batch size- 32, the activation functions were Softmax and Relu and the number of epochs were 40 and 30 for DenseNet 121 and InceptionResNetV2. To get better results they have shown interest working with chest X-ray with lateral view in the future[36].

Rahman et al. have discussed the seriousness of pneumonia disease by stating some facts including the number of people that are exposed to this disease which is 2 billion and 1.4 million children dying from it each year. In their research, they have presented a detailed report on pneumonia detection and they have used four pre-trained deep convolutional neural networks which are AlexNet, ResNet18, DenseNet201 and SqueezeNet. They have used X-ray images that were preprocessed and the team of 7 has managed to show outstanding results such as 95% and 93.3% respectively on their two out of three schemes of classification with a learning rate of 0.0003 and a batch size of 16. They believed to have the highest accuracy during the time and they proposed further studies that could be helpful in the future for the radiologists and also in the airport screening of pneumonia patients[37].

A different approach to making image classification more efficient and accurate by addressing the limitations of the classical convolutional neural network was the primary focus for Liang et al. in this paper. They say that classification results are generated by abstracting the original image hierarchically which is not much sentient of the orientation and the position of the image and to solve that problem they have put an effort to develop a training strategy and neural network framework. What they propose is a deep learning framework with residual thought for the detection of pneumonia in children. Their proposed framework tries to overcome problems like overfitting, degradation, loss of feature etc. However, they also had to work on overcoming the problem of available data that was not sufficient. Finally, they were able to achieve 96.7% accuracy in child pneumonia detection[38].

As stated by Ibrahim et al. , the outbreak of covid 19, which was declared a pandemic by WHO, has broken into more than 200 countries affecting more than 37 million people and led to the death of more than 1 million people all over the world and so, pneumonia detection caused by corona virus was the focus of their research. They had designed a deep learning approach for X-ray detection and acquired 94.43% accuracy. However, as doctors all over the world were struggling with a lot of patients at that time, the data were not sufficient and that was a drawback for their work[39].

In the paper written by Rajpurkar et al., a deep learning algorithm had been developed which they named 'CheXNet' which is a 121-layer convolutional neural network. The focus of their research was to show that their designed convolutional neural network model provides more efficiency and accuracy in performance than the radiologists. The unique part of their research was they had compared their model with the performance of radiologists who had been a part of their project. And they have been successful in showing that their model performed better than the radiologists. The machine learning model reached an astounding 95% score in accuracy which was slightly better than the accuracy of the radiologists. However, this project had its limitation as well. Since a patient's history is sensitive and private, both the radiologists and the model were not allowed to use the patient's history. As a result, they had to rely on what was available to them[40].

Developing an advanced deep learning-based architecture was the primary focus for Shah et al. to detect pneumonia from X-ray images and they did it by the utilization of a neural network of 16 fully connected layers providing an accuracy of a substantial 96.6%. The study was to differentiate between pneumonia and normal chest X-ray. Their model could successfully classify 839 images out of 855 images belonging to pneumonia cases and 293 images out of 317 images belonging to normal cases were correctly labelled by the model. However, their research was done on a total of 1172 chest X-ray images with 855 pneumonia X-rays and 317 normal X-rays which is not a big dataset. The goal of their work was to reduce physician workload and make a better health sector. Their deep learning model provided better accuracy results but a small dataset will always raise the question of the reliability of such a system as the real-world data can vary a lot and research and development of such a machine learning model on a small dataset can be controversial when it comes to dependability on a system[41].

Albahli et al. have put a reflection on the performance gap in chest related diseases due to the increasing demand of doctors which is 15% every five years and they have indicated that to avoid such gap that could lead to far reaching consequences, automation of chest disease detection could be very effective. Their study was on detection of cardiothoracic diseases via chest X-rays. They have developed four deep learning models, three of them were ResNet-152 with and without image augmentation and one of them was Inception model. Without image augmentation with ResNet-152, the highest accuracy they could manage was 67% and with image augmentation they have reached an accuracy of 83% where with inception model they have managed to get 68% accuracy. They realized the importance of a large available data and they also have intention to build optimization algorithms for hyperparameters in the future[43].

The paper of Chandra et al. have presented a method which includes multilayer perceptron, Random forest, Logistic Regression, classification via regression etc. for automatic detection of pneumonia. Their experimental results have been able to outperform the existing methods by achieving an accuracy of 95.63%. However, their model has been evaluated on only 412 images which contained 206 normal cases and 206 pneumonia cases, as a result of which this cannot be generalized[44].

Kuo et al. have shed some light on how anti psychotic drugs are frequently prescribed to the schizophrenic patients which has been known to lead to cases of pneumonia. Their purpose of research was to build a machine learning model that could detect pneumonia on hospital acquired patients. They have succeeded to manage data from a Taiwanese district mental hospital including 185 patients that were diagnosed with pneumonia between year 2013-2018. 11 features or predictors were used and several machine learning algorithms such as classification, regression, decision tree, k-nearest neighbour, random forest, Logistic regression etc were utilized to build the model and among them random forest and decision tree have produced optimal results. Random forest had the highest accuracy score of 0.917 and decision tree had provided an accuracy score of 0.912[45].

The study of Yue et al. included hospital stay patients that had pneumonia and their goal was to develop and test machine learning based models for detection of pneumonia caused by Covid-19. The CT scan images were confirmed cases of SARS-CoV-2 infection and the patients had been classified in two classes which included hospital stay patients for more than 10 days vs hospital stay patients for less than 10 days. The models developed were Logistic Regression and Random forest models and they achieved an accuracy score of 0.97 and 0.92. The number of patients that the data were collected from was 52[46].

The paper by Antin et al. uses supervised learning for the detection of pneumonia and the output includes binary classification classes which are pneumonia or non-pneumonia. Their deep learning model that uses logistic regression has failed to capture the complexities of the dataset efficiently as detection of pneumonia is complex and some areas of interests were blocked by ribs. Another challenge they had to face which they also mentioned was the dataset they used had images of other images as well and a lot of the images looked quite similar to the images of pneumonia. However, they focused more on errors than the accuracy and so as a result they mentioned about comparing their network with ChexNet documents that had a score of 82.8% and their network failed to match that score. They suspected that the number of pneumonia cases being significantly low than the number of non pneumonia cases might be a valid reason behind the incompatible results of their network. They have also struggled with pixels of the images and for their future work, they have shown interest in using different features, making more error analysis and improved logistic regression[47].

Sharma et al. in their study have stated the fact that inflammation of lungs caused by pneumonia can be fatal and the time consuming process by doctors and radiologists to detect pneumonia which is not accurate all the time. Keeping that in mind they have worked towards the detection of pneumonia from X-ray images by machine learning and they have proposed different Convolutional neural network or CNN models. To see the different aspects and effect of their model they have trained their model on the original dataset as well as on an augmented dataset. Their models pulled 90.68%, 89.32%, 79.80% and 74.98% respectively[48].

The paper presented by Stephen et al. focuses on their creation of a convolutional neural network from scratch to detect pneumonia from a limited amount of available X-ray images. Instead of relying on traditional approaches, they have built their own convolutional neural network from scratch that could detect if a patient has pneumonia or not from an X-ray image. As there is scarcity of available data of patients, several data augmentation algorithms have been implemented by them for the purpose of improving the classification and validation and gaining remarkable accuracy. The experiments were conducted 10 times each and for three hours for the proposed approach and to attain an efficient performance by the model, parameter and hyperparameters were scaled heavily and as a result of which by deploying data augmentation and learning rate variation they got an accuracy of 95.31%[49].

Machine learning models such as deep learning models perform well if trained and tested under proper utilization of the methods but some machine learning models while performing excellent on a dataset collected from a hospital system fails to show satisfying results if they are applied to a different hospital system leading to a question of reliability. Solving this problem was the research done by Janizek et al. and they worked to create a model that could provide substantial predictive results when applied on different datasets collected from different hospital systems[50].

Because of recent pandemic and other issues, it is of great importance to be able to detect pneumonia from chest X-rays as quickly as possible when other sophisticated imaging models are hard to get access to. Relying on the chest X-rays only can be risky in terms of pneumonia detection since it has been well known that the viruses can have new forms or bring new symptoms because of their nature of mutation. As a result of which there can be a great dataset shift because of the novel mutated viruses and this reason alone can deteriorate the performance of classification based machine learning models. Taking these factors into account, Zhang et al. have proposed an approach to make a model consisting of shared feature extractors, anomaly detection to detect and classify between viral-pneumonia and non-pneumonia. The purpose was to detect anomaly cases by looking into some factors or scores pulled out by the model. Based on how big or small the score will be, there would be a way to understand an anomaly case. Their dataset contained 5,977 viral pneumonia cases without Covid-19 and 37,393 non- pneumonia or healthy cases. Their model was able to achieve a sore of 83.61%[51].

As Covid 19 has similarities with pneumonia, Tuncer et al. have presented a classification approach for covid-19 detection. They had gathered three classes of datasets which were Covid-19, pneumonia and normal chest X-rays. The machine learning model that they presented was named the exemplar model. The main objective was to differentiate covid 19 patients from pneumonia patients. The data they managed to gather included 1495 covid 19 and 1027 pneumonia patients. Their model achieved an excellent 97.01% accuracy. This approach is considered to be a cognitive approach that takes out the monotonous task of inserting hundreds of parameters like deep learning models. In fact, their proposed research have been able to bag a better performance achievement than 10 percent of state of the art deep neural networks. For the future work, they have stated that they have interest in building an interface or a real time interface that would show the results immediately on screen to the user[53].

The study of Jaiswal et al. shows how to identify and localize pneumonia from chest X-ray images where they have stated that positioning of a patient and other factors can complicate building such models. They have built a model based on convolutional neural network which works on pixel-wise segmentation. By critically modifying the training process their model achieves robustness. However, their model is only focused on chest X-rays with pneumonia[54].

Gabruseva et al. in their work, have used deep neural network for the detection of pneumonia utilizing data augmentation and multi-task learning. Their idea was evaluated on Radiological society of North America pneumonia detection challenge and has been one of the best in terms of performance with excellent accuracy. The dataset they used included frontal chest X-ray images of 26684 patients and they had classified them into three classes that included "normal", "not normal and no Lung opacity" and "Lung opacity". Their ensemble model has reached 87.5% accuracy on average[55].

5 Analysis

In this chapter, an analysis is carried out, such as, conducting an expert interview, the design consideration, and the reliability of the system.

Our project focuses on building a system that can assist the doctors in diagnosing pneumonia and continue to learn and provide better results. To able to build something like that we needed input from experts from the field of healthcare to know where we stand with our idea. So we will start what we gained from the interviews first.

5.1 Interviews

As part of methodology, Interviews have been conducted which are added in the appendix section. These interviews gave us insights about the current situation about pneumonia patients and the treatment. The experts were from different countries which also played an important role giving us an international scenario. We weren't looking for any technical views as we knew they are from healthcare sector. What we were interested in was their opinion about a system that could assist them in the treatment of patients with pneumonia and what do they think about it. In return we not only got positive feedbacks but also, we got to know how important it can be in certain situations in certain countries.

5.1.1 Interview with Dr. Nowshin

On Sunday, 24th April 2022, we interviewed Dr. Nowshin from Bangladesh, a pediatrician who is working in a Swedish company (Terrades Homes) with ailing children which can be found here [Appendix A.1.2]. The interview was initially designed as a semi structured interview that included some already prepared questions. We were looking for some specific answers but they could expand on any questions or topics as they saw fit.

As our contact is from a country which is densely populated, where doctors are very less in numbers compared to the number of patients, our expectation from this interview was to get an overview of the situation and know if our project could bring significantly positive outcomes for doctors in countries like Bangladesh.

5.1.2 Interview with Dr. Sulagna

On Monday, 25th April 2022, we interviewed Dr. Sulagna from Ohio, USA who is a resident Physician in internal medicine at Kettering Hospital, Ohio which can be found here [Appendix A.1.3]. We designed a semi structured interview for Dr. Sulagna with some questions in mind and we also let her expand on some topics as she saw fit.

By this time, we already had some information about pneumonia in a densely populated country like Bangladesh. Now this interview was a scope to find some information about patients in a developed country like the United stated of America. Our goal was to introduce our project idea to Dr. Sulagna and get to know what she thinks about it and how it could be helpful in that country. As the interview continued to progress, we kept learning new things and got a picture which is quite distinct from that of Bangladesh.

5.1.3 Acquired Knowledge

We will discuss the topics that affect our project the most. Based on the interview from Dr. Nowshin we got a scenario of pneumonia patients of Bangladesh, which gave us a lot of knowledge about the state of the art solutions and the lack of proper technology and Doctors.

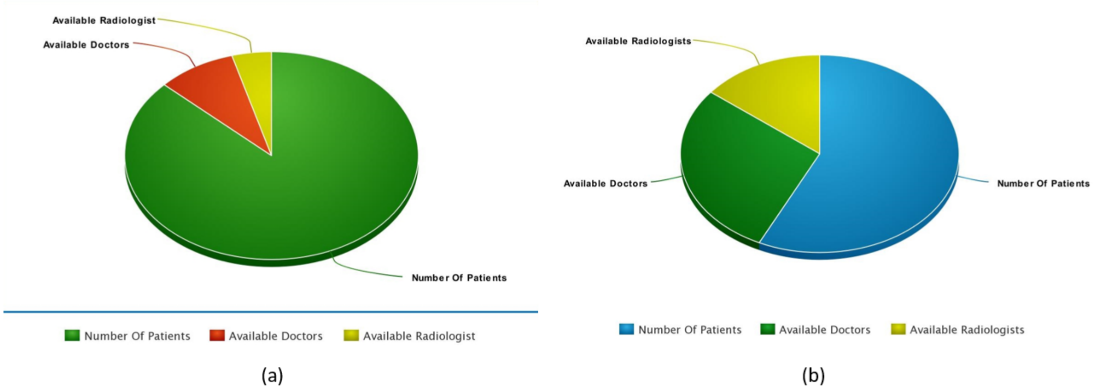


Figure 12: Two scenarios based on the interviews

In Figure 12 (a), the chart tries to depict a random picture of patients, available doctors and radiologists in Bangladesh based on what we learned from the interview with Dr. Nowshin. What we learned from the interview was there could be upto 20 patients per hour and the process of diagnosing pneumonia was quite time consuming in Bangladesh. People have to get an X-ray and hand it over to the radiologist. As there could be as many as 20 patients per hour, the radiologist cannot provide a conclusive result immediately. The radiologist provides a day when the patients could get their final conclusive report on pneumonia. This could be fatal for patients who need immediate attention specially children. Children with pneumonia need the most attention by the doctors and they need to be treated immediately. If there is any delay in diagnosing pneumonia or a certain kind of pneumonia, it can prove to be deadly. So in such emergency states, anything that would save a lot of time for the doctors and the patients, could be very useful for both parties. This information gave us a lot of motivation towards our project as it could help detect pneumonia from an X-ray almost immediately.

Figure 13: Patients waiting in a public medical college indoor in Bangladesh[17]

From the interview, it is quite clear that the amount of pressure the doctors have to handle is immense and our project in a country like Bangladesh could bring unprecedented results and immediate impact. What we also learned is the cost that the doctors are paying in the form of time. With increasing number of patients, the patient to doctor ratio is very alarming. Each doctor has to take care of a large number of patients which can be extremely time consuming and tiring - which also indicates a possibility of wrong diagnosis. Considering these, such kind of solution can be very handy. However, from a doctor's point of view, we got some indication about how our system should look to those whose expertise are not in developing a machine learning environment. Although we were given a very positive outcome by Dr. Nowshin but she also expressed her interest towards an easy to use user interface and a simpler system for a better user experience. They don't need to know the complex network architecture of our system as long as they can use it. At the time of this interview, the implementation was in progress. Our idea was to show that we could create something that could learn from the images by training and then feeding it some random test sets and see how accurately it could guess or detect. That alone would take a lot of time and knowledge as creating such a system needed understanding of complex architecture, which was challenging for us. However, as Dr. Nowshin saw a potential in our idea, we decided to also include a touch of software development by creating an interface that would show the results.

Figure 12 (b), tries to depict a picture of what we learned about the situation from the interview with Dr. Sulagna. The situation is more 'under control' than that of Bangladesh. Doctors are available most of the time as the number of patients in an hour is not high. For 15-20 patients there could be two or more doctors. We also learned how the process is done for a patient with pneumonia. Most of the doctors already have some idea just by looking at or knowing about the symptoms and then they go through the X-ray or send the X-ray to the radiologist for a final result. They call this process 'Imaging'. This process can be done in a short amount of time and sometimes due to unavoidable circumstances it might take a day or two. However, although detecting pneumonia might not be the problem or difficult for them, we learned the challenge is to detect which kind of pneumonia it is as they have to be treated differently. This is where Dr. Sulagna was interested mostly. Because they mostly look at the infiltrates to distinguish between viral and bacterial pneumonia and she mentioned, there is a very thin line between them. We also learned that Covid-19 patients can also be exposed to pneumonia and they have to be treated carefully. Covid-19 patients are sensitive and they need attentive care. She also mentioned about a possibility about false negative or false positive results. At the end, she saw potential in such kind of solution as it will enhance their suspicion towards a certain kind of pneumonia and accelerate the process. Diagnosing a disease via X-ray didn't seem to be costly or difficult in USA, but giving them a head start towards a suspicion is considered to be definitely helpful. She was also aware that such systems may not have hundred percent accuracy but she mentioned that they would already have an idea and such solutions would definitely help not only the doctors but also the patients and save a lot of time. The interview with Dr. Sulagna inspired us to work on detecting more kinds of diseases and more types of pneumonia in the future. She was aware of the technological advancements, such as apple healthcare and she had a lot of inspiring words for us to work with this in the future.

5.2 System Design

It has been determined, following the collection of data from a variety of sources, such as the state of the art and the literature review, that there are two primary designs of subsystems that need to be taken into consideration. This conclusion was arrived at after compiling information from these and other sources. The initial stage of the design process will involve the instruction of a machine learning model that is capable of distinguishing between a normal case and a case of pneumonia. This part will be further elaborated in the next subsection 5.2.1. The next aspect of the design that needs to be taken into consideration is the creation of an interface, which will be utilized by the users to initiate interaction. The interface will make users able to interact with the system and upload an X-ray image. The image will then be analyzed by the machine learning model that was developed in the first component of the design. This will also be elaborated in the subsection 5.2.2.

5.2.1 Machine learning Process

Users can be able to make better decisions on patient's outcome in a variety of situations by employing a complex but well-executed machine learning model. A human can only deal with a limited number of facts and important factors when making a decision, whereas machine learning algorithms can analyze and correlate massive amounts of data in real time. The machine learning models have gained an entirely new perspective as a result of these connections, which would not have been possible if only the standard manual approach had been used. It will be possible to assist the health-care sector in identifying pneumonia in its early stages by using a model that can distinguish between a pneumonia case and a normal case. To develop an image recognition model, it is necessary to investigate various machine learning methods and techniques that can be used. This is due to the project's goal of processing and analyzing patient X-ray images. This will be accomplished by first investigating some of the process models for creating our system. It will provide us with insight into how to make a good machine learning model, which steps to consider, and what challenges must be overcome when developing an image recognition model. The process of developing machine learning models can be modeled in a variety of ways, but the most commonly found process model is the seven-step approach process model, which covers all aspects of developing a model of such type. There are seven steps that must be completed in order to create a machine learning model. Gathering data, prepping data, selecting the model, training the model, evaluating the model, tuning the hyperparameters, and making predictions are the steps involved [15] [16].

Gathering data: The first is to gather the data for the related purpose, this can be images, audio/speech files or their related data. This is also one of the challenges that exist while making a machine learning model, as without enough data, it would be not possible to make a well sophisticated model. From our state of the art, we found that more data would mean more questions and more questions would mean more answers [10]. Therefore, we tried to find as much as datasets for our project in order to a good accuracy. For our data gathering we will be using two different datasets [18][19] that is available online publicly.

Preparing data: The next step would be to prepare data for use in subsequent phases after gathering data for instance on two features. The identification and minimization of any potential biases in our data sets for the two attributes is a major emphasis of this stage. To begin, we would randomize the sequence in which our data for the two features was collected. We do this because we do not want the order to have any influence on the model's decisions in any way. Furthermore, we would check our data sets to see if there was any skewness in favor of a specific attribute. This would once again aid in the identification and correction of a potential bias, since it would indicate that the model would be proficient at properly detecting one feature but would struggle with the other features. Additionally, the data should be separated into two parts: the first half should be used for training, and the second part should be used for testing. Depending on the size of the data-set, the data should be divided into two groups using an 80/20 or 70/30 split ratio. It is suggested that more data be used in the training phase, with the remaining information used in the testing phase.

Choosing a model: The next step is to choose a model. Academics and data scientists have created a myriad of models for a variety of objectives throughout the years. Certain are particularly well suited to image data, while others are particularly well suited to sequences, such as text or music, while yet others are particularly well suited to numerical data, and yet others to text-based data. In this stage, different image recognition models will be examined in order to find the best model for our project.

Training: The data that we have collected and compiled will be utilized to instruct the participants throughout this phase. As a result of their training, the algorithms will begin to produce the accurate results at this time. If the approach employs neural networks, it will run through the iterations that the users choose, which are also known as hidden layers in certain circumstances. It can train until it obtains the required output or a level of confidence or prediction accuracy that the user is happy with. It should be noted, however, that some of the data from the data-set should also be used for testing reasons.

Evaluation: After the training is completed, it is required to evaluate the data. This is the stage at which the model's quality may be determined. This is done by comparing the data that was separated during the "Preparing data" stage with the data that has been trained. Given that the test data has never been presented to this model before, it is a fantastic strategy for testing the real data and determining how well the training model is functioning.

Hyperparameter Tuning: Following the conclusion of the evaluation, the next stage would be to decide if there is room for improvement. If there is overfitting or underfitting, for example, some of the improvements may be made by adjusting/tuning some of the parameters, such as adding more data to the training set, introducing noise to the training set, tweaking the learning rate, and so on.

Predictions: Following the completion of the last step, the machine learning model is ready to generate predictions based on user-submitted queries. At this stage, the machine learning model is self-sufficient and can draw inferences based on the training data.

5.2.2 User Interface

It is necessary to use the machine learning model after it has been created in order to detect pneumonia and normal cases. An application programming interface (API) must first be created before the machine learning model can be used. This API will take an X-ray image from the user as an input, which will then be processed by the machine learning model to determine whether or not there is a sign of pneumonia and the result will be showcased back to the user. However, an API would not make sense without having an interface that is integrated with it. Therefore, it is necessity for creating an interface that prompts users to submit an X-ray image. The image will then be sent to a server containing both the API and the machine learning model. Nonetheless, there is a need for a system that will use this machine learning model in order for it to be consumed by an API capable of detecting pneumonia cases. The user interface should include a button that allows users to upload images. This button will be used by users. When an image is successfully uploaded to the system, both the image's name and a preview of the image should be displayed. The user would benefit from knowing which image they have uploaded in order to ensure that they have chosen the correct image to insert using this method. After uploading the image, the user should be able to see the results by sending a request to the API, which will compare the image to the trained machine learning model. The results could indicate whether or not the image contains pneumonia, as well as a confidence level expressed as a percentage indicating how accurate the predicted value is.

5.2.3 Classic Machine learning vs Deep learning

The findings from our state of the art in the fields of deep learning and traditional/classic machine learning have led to the discovery of various methods used in both of these subfields. Traditional machine learning employs a number of learning strategies, such as supervised and unsupervised learning. The supervised learning process resulted in the development of a number of methodologies, including classification and regression. We will use classification methods to distinguish between pneumonia and normal cases based on X-ray (CXR) images because we are aware that these methods are used to distinguish between distinct classes of labeled data. This is relevant to the work we will be doing for this project. Because regression methods predict a continuous quantity, we cannot use them because the quantities are continuous. Furthermore, because unsupervised learning works best with unlabeled data, it is inapplicable to our situation and should be avoided. We investigated several classification strategies, including Support Vector Machines, Naive Bayes, and Nearest Neighbor. On the other hand, we investigated deep learning, which is a subset of machine learning in its most fundamental form. We investigated a number of Tensorflow image classification models, including Resnet and Inception, among others.

At this point, the question is, "What are the primary key differences between traditional machine learning and deep learning, and which would be best suited for our project?"

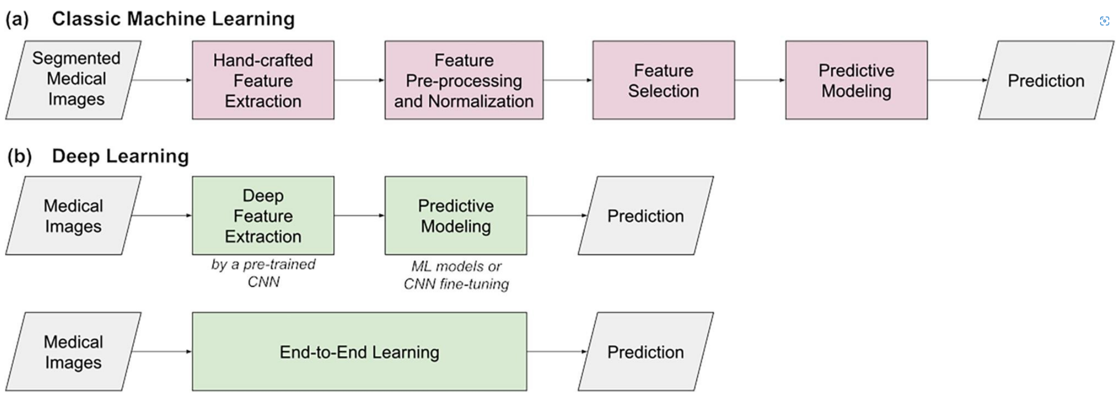


Figure 14: Classic Machine learning vs Deep learning [60]

To describe the key differences of these two terms, it is illustrated in the figure 14. As we can see from the figure, classic machine learning methods require more steps than deep learning. In classic machine learning, we would manually define relevant features of a medical image also called "Hand-crafted feature extraction", such as edges, corners, or other relevant patterns, in order to train the machine learning model. This would mean that there is need for to pin-point the spots in the images for instance, how a pneumonia would look like and if the features are defined incorrectly, then the model would more likely to do errors and give wrong predictions. After defining the features, the model will then analyze and classify new objects within the image using those features to complete the task of identifying new objects in an image. To achieve this goal, pre-processing and normalization techniques are used to highlight the important features contained within the images, thus select the most relevant features. Deep learning, on the other hand, allows us to avoid manually extracting features from images and automatically defines the features for us. This is done by feeding images directly into the deep learning algorithm, which then makes an object prediction and extract features by a pre-trained models called Convolutional Neural Network (CNN). This feature extraction phase is also called "End-to-End Learning". However, deep learning requires a powerful graphics processing unit (GPU) and a large amount of data. This is because the learning typically processes data using pre-trained CNNs, which typically have many hidden layers consuming a lot of energy power in each layer to extract the features required to build a complex model.

Following a thorough examination of the key differences between classic machine learning and deep learning, it makes sense for us to use deep learning because it eliminates the need for us to go through the manual feature extraction process.

5.3 Requirement Definition

Tables 1 and 2, which are presented below, show the functional and non-functional requirements criteria that were identified as a set of specifications based on how the system should work. A Functional Requirement is a description of the service that the software must be able to provide (FR). It describes a software system or a component of a software system. A function's only components are inputs to the software system, system behavior, and outputs. It could be a calculation, data manipulation, a business process, a user interaction, or any other specific functionality; this determines what function a system is likely to perform. Where the non-functional requirements are the constraints or requirements that are imposed on the system. They cover critical aspects of system quality such as scalability, maintainability, performance, portability, security, and reliability [42].

Some of the functional and non-functional requirements are discovered during the course of our project while brainstorming, but a significant number of these requirements are discovered while collecting data in the early stages of this project. The system must meet all of the functional and non-functional requirements listed below in order to serve the project's goal, which will be followed by a web application, and it must be able to generate system results whenever they are required.

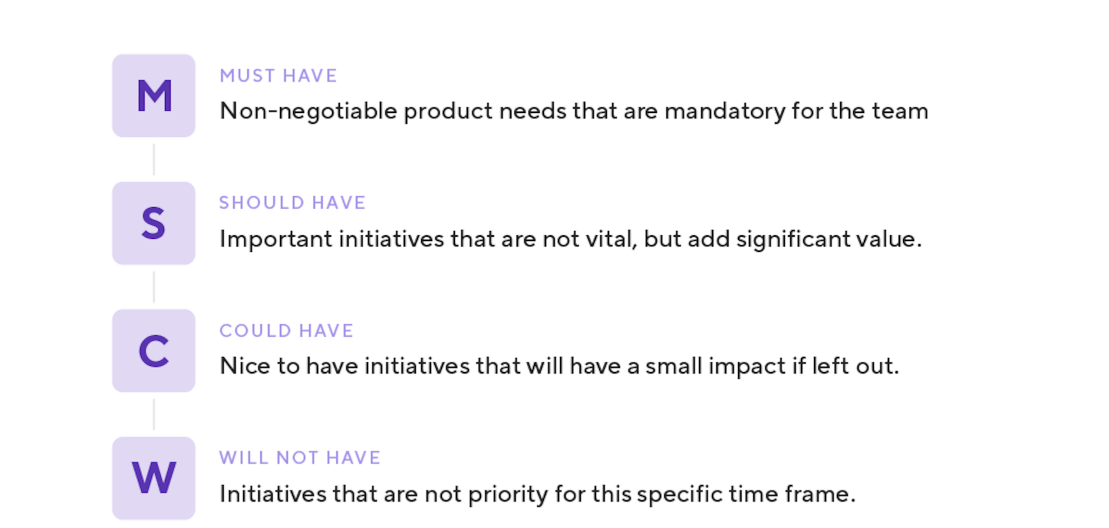


Figure 15: MoSCoW Prioritization[52]

As shown in Figure 15, the requirements are ranked in descending order of importance using the MoSCoW prioritization method and incremental approach. MoSCoW refers to four distinct types of initiatives: those that are required, those that should have been done, those that could have been done, and those that will not be done at this time[52].

M (must-have): These define the Minimum Usable Subset (MUST) of requirements that the project must deliver or the system will not function. The project also guarantees that these requirements will be met.

S (should-have): The initiatives that should be pursued are only one step down from the must-haves. They are not required for the completion of the project; however, their presence is strongly advised. If it is left out, the functionality of the project is unaffected. Nonetheless, the initiatives may add a significant amount of value.

C (could-have): These requirements are nice to have but are considered less important because they have no effect on the project's functionality.

W (won't-have): These requirements will not be implemented, and the delivery will not be made at this time.

NO.	Functional Requirements	Priority
1	The system must allow a user to upload an X-ray image	Must
2	The system must showcase the score on both pneumonia and normal case	Must
3	The system must recognize and differentiate between a pneumonia and normal case	Must
4	The system must be able to showcase the result of a pneumonia and a normal case to the user.	Must
5	The system should showcase a preview of the selected X-ray image to the user	Should
6	The system should display the selected image's name to the user	Should
7	The system could store the results on a storage	Could
8	The system won't detect other heart diseases other than pneumonia	Won't

Table 1: Functional Requirements

NO.	Non-Functional Requirement	Priority
1	The time required for the system to load the machine learning model and generate a response must be less than ten seconds	Must
2	The system must be capable of producing accurate results	Must
3	The system must be accessible and functional to the users all the time	Must
4	The system must be able to send a X-ray image to an API that can detect pneumonia and normal cases	Must
5	The system should provide descriptive information on each text, button, and other component for the sake of user accessibility.	Should
6	The system should provide a salable server with sufficient resources and designed to support a large number of users	Should
7	The system could be supported without accessing it using a web browser	Could

Table 2: Non-Functional Requirements

5.4 Use cases

A use case diagram is a diagram that depicts the dynamic behavior of a system. It incorporates use cases, actors, and their relationships in order to encapsulate the system's functionality. It simulates the activities, services, and functions that a system or subsystem of an application must perform. It not only describes how a user interacts with a system, but it also illustrates the system's high-level functionality [42].

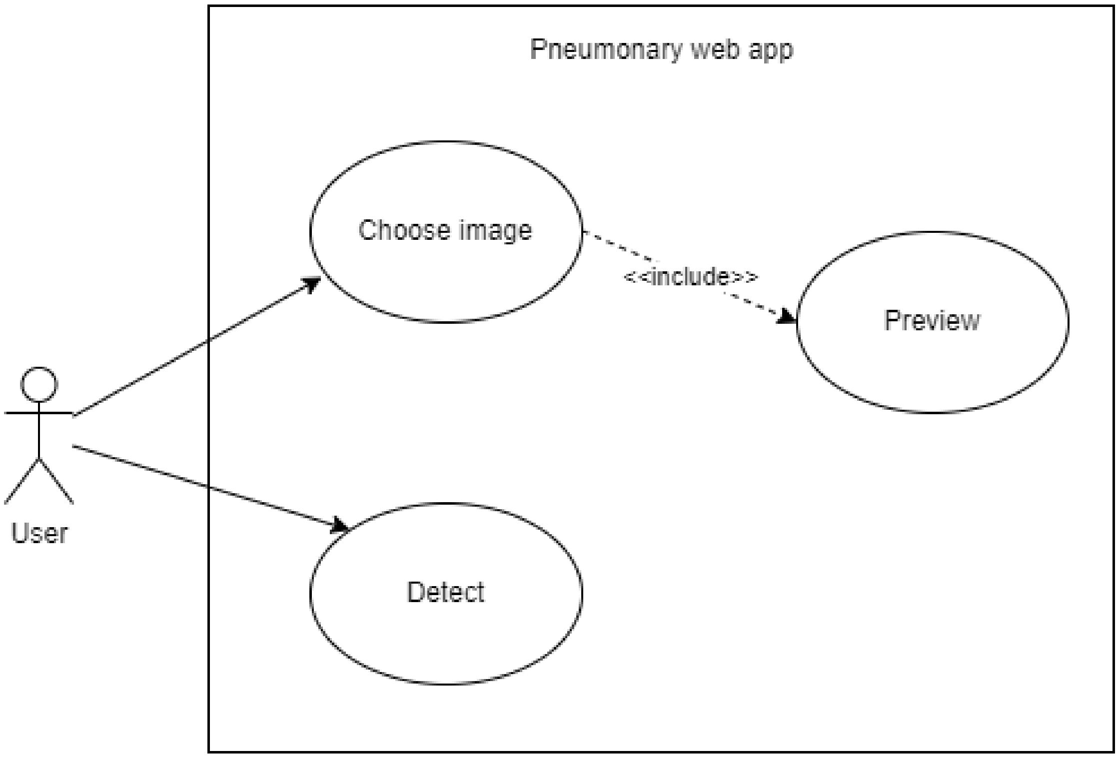


Figure 16: Use case diagram

Figure 16 depicts how the various use cases are linked so that a user can interact to choose/upload an image, preview the image, and detect the results. The use-case diagram is constructed from its constituent parts, which are systems, actors, use cases and relationships. The system is a representation of whatever we are developing; in this case, it is our web application, which is depicted as the large rectangle. An actor is someone or something who uses our system to accomplish a goal, and in our case, the person referred to as a user is someone who works in the medical field that is going to use our system. The use case element, represented by the oval shape, provides a description of what a system is responsible for. This shape represents an activity that is in charge of completing a task within the system. Because they correspond to activities that occur within the web application, these use cases have been placed within the rectangle. The last component of the use case diagram is the relationships between the use cases in our system. These are the interactions that an actor has with each of our system's use cases.

After making the use case diagram the following two use cases (UC1 & UC2) has been made.

Name	UC1 Choose image
Preconditions	None.
Main Flow	The user is prompt to choose and upload an image, so it can be processes against the machine learning model to perform the detection of pneumonia.
Sub flows	After the user has chosen an X-ray image, the system will show a preview of the image to the user.
Alternative Flows	None.

Name	UC2 Detect
Preconditions	The user has chosen a X-ray image.
Main Flow	The flow will detect the image by sending the image to the API, where it will be further processed against a machine learning model to perform the detection of pneumonia.
Sub flows	None.
Alternative Flows	None.

5.5 Sequence diagram

The sequence diagram depicts the details of a system in the order in which operations are performed. They can record the objects' collaboration as well as how they interact with one another. The vertical axis of a sequence diagram represents time and shows what messages are sent and when they are sent. Sequence diagrams are time-oriented and visually depict the sequence of interactions [42].

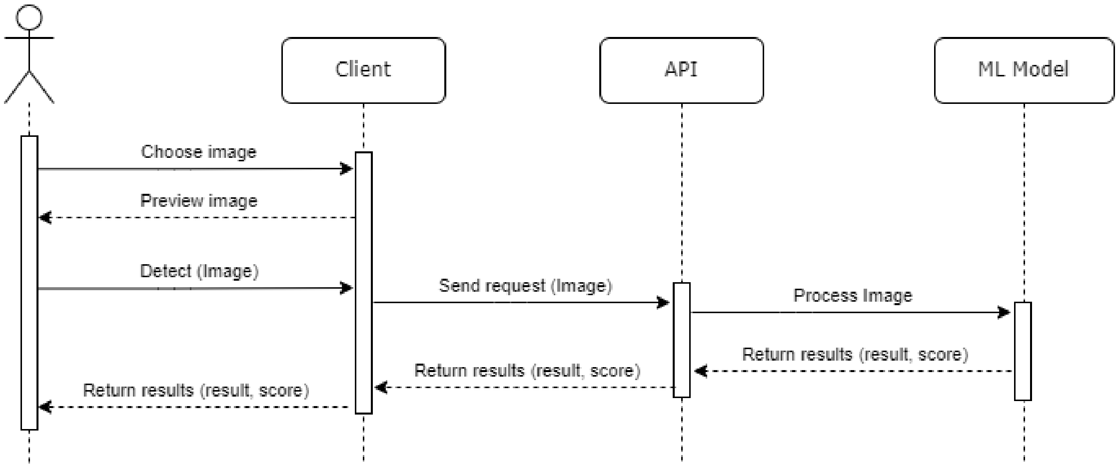


Figure 17: Sequence diagram

Figure 17 depicts the overall system operation process, which is as follows:

The user choose an image in the client (web app), where it showcases a preview of the image back to the user. Then the user perform the detect action on the image, which takes the image and is being transferred from the client to an application programming interface (API), where it is then processed by the machine learning model. After the machine learning model has processed the image, the results will be returned to the client via the application programming interface (API). The sequence diagram is built using various essential components such as the object, activation box, actor, lifeline, and message. The object symbol is represented by a rectangle box in UML and represents a class or object demonstrating how an object will behave in the system, where we have four objects: User, Client, API, and Machine learning model. The lifeline symbol is a vertical line with dots that extends downward from each object in a sequential manner. It represents the passage of time as it moves from top to bottom of the page. The vertical bars placed on the lifeline represent the activation box. These bars appear to indicate when a specific object is active during an interaction. The actor symbol, also known as the stick actor figure, is a representation of a system interaction. In this scenario, the user takes on the role of the actor symbol. The message describes what happens during each stage. This is represented by a one-sided lined arrow, which represents when a sender must wait for a response to a message before continuing to show both the call and the response. In other words, the sender must wait for the recipient to respond before continuing to display both the call and the reply. The message depicted as an arrowhead with dashes represents a response to a request that was initially made.

5.6 Context diagram

A graphical representation that summarizes an entire system. When it comes to context diagrams, there can only be one circle or process that represents the entire system. The goal of this diagram is to depict the system's expected inputs and outputs, both to and from a variety of external entities. Through this display, a system analyst can model what expected data will enter the system. After the system has processed the data, the analyst can model the information that will be returned to the external entities [42].

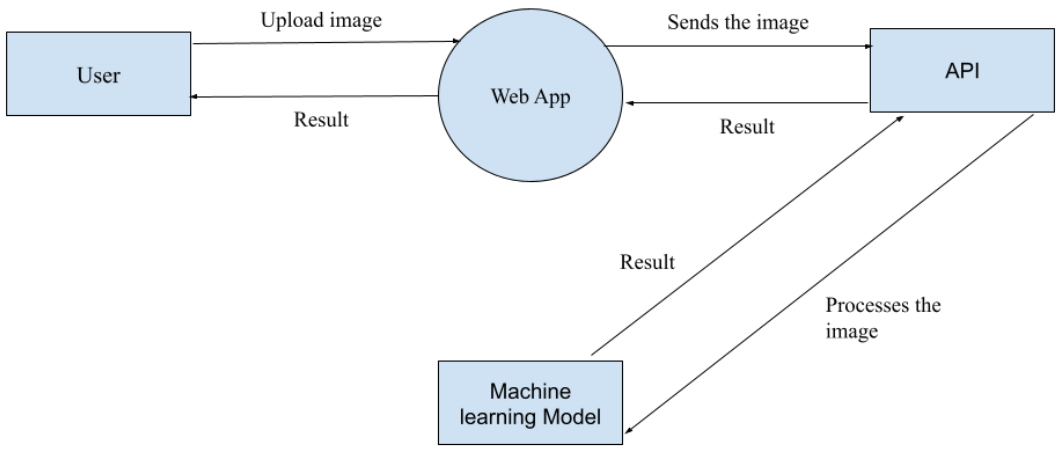


Figure 18: Context diagram

The figure 18 depicts the various components of our system, as well as the interactions that occur between them. The components are the user, web application, application programming interface (API), and machine learning model. A user uploads an X-ray image to the web app, where it is being sent further to an API. The image is then sent to a machine learning model, which is then tested to determine whether or not there is pneumonia. The result sent back to the user via API to the web application.

5.7 Architecture diagram

An architecture diagram is a visual representation of all the components that make up a portion or the entirety of a system. Above all, it helps engineers, designers, stakeholders, and anyone else involved in the project understand the layout of a system or app [42].

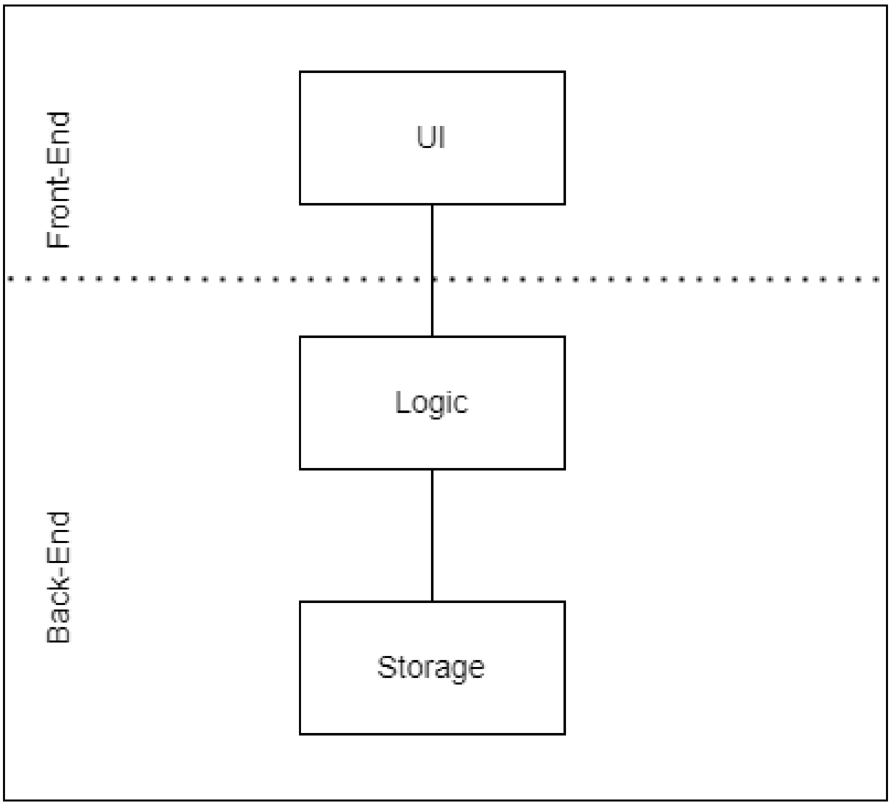


Figure 19: Architecture diagram

Figure 19 depicts our project's architecture diagram, which is made up of two primary components. The back-end of our project is the first component, where storage is a file containing a machine learning model that can be consumed by an application programming interface (API) or logic. The front-end of our project will take the form of a user interface and will be the primary point of interaction between users and the system.

6 Implementation

After doing analysis and design, we will carry out an implementation of the prototype and provide an insight of the approaches used during the implementation of the system. due to the fact that this project requires the development of two distinct components The first step is to train a machine learning model, followed by the creation of an interface with an application programming interface (API) that will use this machine learning model to detect pneumonia from an X-ray image. To complete the first part of this project, we experimented with various image recognition techniques. For the second part, we created a user interface that is integrated with the machine learning model.

6.1 System Information

The specification of the system that our machine learning model was built in has been shown below:

- Operating System: windows 11 pro(64 bit).
- Processor: intel(R) Core(TM) i5-8400 CPU @2.80 GHz.
- System Model: H310 Gaming Codex 3(MS-B913).
- Memory: 16.384 GB.
- GPU: NVIDIA Geforce GTX 1060 (6 GB)
- DirectX Version: DirectX 12.

6.2 Machine learning

The first stage in implementing the project is to create a machine learning model that is capable of identifying between a normal and pneumonia case. The development is carried out in the Visual Studio programming environment, which includes the ML.Net library [30]. We chose ML.Net as our platform of choice because we are not only developing a machine learning model but also building a system for it in which the trained model can be used to predict incidences of pneumonia. Because the system will be created in a web application (ASP.Net), which is also a Microsoft's product, it gives more sense to build the machine learning model under the same eco-system. This allows the system to be compatible with other systems and interoperate with them without any issue. Moreover, ML.Net is also written in C#, which has the similar syntax as JAVA. This gives us an advantage in that we can understand the coding better because the members of our group have had more experience in the programming language throughout our academic education. Python can be used as an alternative to ML.Net, which are totally two different ecosystems, and if we decided to use Python instead of ML.Net, we may have some difficulties in integrating these two eco-systems together. Furthermore, Python have challenges when it comes to production execution, particularly when combined with a project like ASP.NET [29]. Furthermore, Tensorflow transfer learning with the combination of ML.Net is used since it provides a pre-defined model capable of recognizing patterns in images, making it faster to train a model.

6.2.1 Dataset

For the training of the machine learning model, a mixture of two separate datasets is used [18] [19]. As a result of the fact that the objective of this project is to distinguish between a case of pneumonia and a normal case using an X-ray image, a binary image classification method, more specifically supervised learning, is used. This is due to the fact that there are two different classes that have been labeled. The datasets are divided into two folder groups: "Pneumonia" and "Normal".

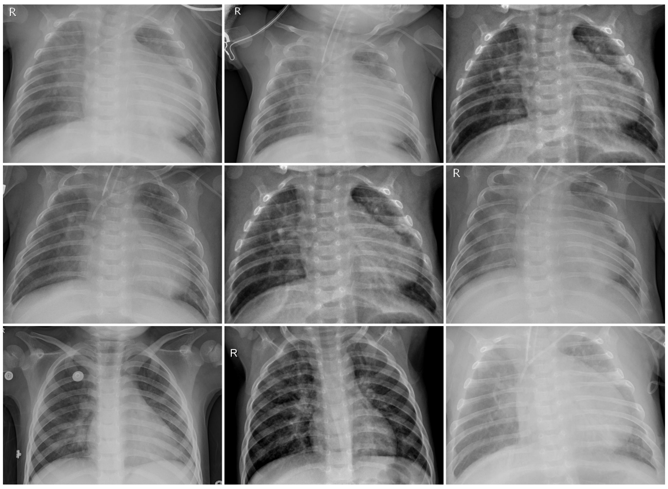


Figure 20: Sample X-ray images from the dataset

Initially, the datasets were skewed, with more data for normal cases and less data for pneumonia cases. To avoid bias training the Machine Learning model, which can lead to overfitting and underfitting, the amount of data for normal cases has been lowered to match the number of data for pneumonia patients. As a result, 8470 images have been prepared for both label classes to be trained on, resulting in 4235 images used for each group. Furthermore, 60 images have been separated that will be used for the test purposes after training the model. This means that these test data will not be introduced to the machine learning model before training. This way, we can assure that these test data are new to the model to make predictions.

6.2.2 Preparing Data

```
class ImageData
{
    1 reference
    public string ImagePath { get; set; }

    1 reference
    public string Label { get; set; }
}

8 references
class ModelInput
{
    1 reference
    public byte[] Image { get; set; }

    0 references
    public UInt32 LabelAsKey { get; set; }

    1 reference
    public string ImagePath { get; set; }

    1 reference
    public string Label { get; set; }
}

9 references
class ModelOutput
{
    1 reference
    public string ImagePath { get; set; }

    1 reference
    public string Label { get; set; }

    2 references
    public string PredictedLabel { get; set; }
}
```

Figure 21: Preparing data classes

As illustrated in figure 21, there are used three different classes for preparing the data for feeding the model. The first class is "ImageData", which includes the properties "ImagePath" and "Label". The ImagePath variable contains the fully qualified path to the location where the image is saved. The "Label" is the image, which indicates to which category of it belongs to. This is the value to be predicted.

ModelInput and ModelOutput are the other two classes, and they are essentially the model classes for the data that is input and output, respectively.

6.2.3 Loading Data

```
public static IEnumerable<ImageData> LoadImagesFromDirectory(string folder, bool useFolderNameAsLabel = true)
{
    var files = Directory.GetFiles(folder, "*",
        searchOption: SearchOption.AllDirectories);

    foreach (var file in files)
    {
        if ((Path.GetExtension(file) != ".jpeg") && (Path.GetExtension(file) != ".png"))
            continue;

        var label = Path.GetFileName(file);

        if (useFolderNameAsLabel)
            label = Directory.GetParent(file).Name;
        else
        {
            for (int index = 0; index < label.Length; index++)
            {
                if (!char.IsLetter(label[index]))
                {
                    label = label.Substring(0, index);
                    break;
                }
            }
        }

        yield return new ImageData()
        {
            ImagePath = file,
            Label = label
        };
    }
}
```

Figure 22: Loading data

Figure 22 illustrates how the data is being loaded. The images are arranged and stored in two distinct subdirectories. To effectively load the data, it must first be arranged into a list of "ImageData" objects, from where there is used the "LoadImagesFromDirectory" function to accomplish it. The foreach statement is then used to do an iteration across each of the files. Within each foreach statement, a check is performed against the file extensions that can be handled. In our case, this includes the jpeg and png file extensions because our data contains both of these. Moreover, the lable from the file is being retrieved and assigned into a variable. The "useFolderNameAsLabel" option is set to true, so the label will be decided by the name of the parent directory where the file is kept. Otherwise, it is expected that the label will be a prefix to or the file name itself. As the final step, a new instance of the "ModelInput" class is being created, which can be used afterwards.

6.2.4 Preprocessing data

```
MLContext mlContext = new MLContext();

IEnumerable<ImageData> images = LoadImagesFromDirectory(folder: assetsRelativePath, useFolderNameAsLabel: true);
System.Threading.Thread.Sleep(5000);

IDataView imageData = mlContext.Data.LoadFromEnumerable(images);

IDataView shuffledData = mlContext.Data.ShuffleRows(imageData);

var preprocessingPipeline = mlContext.Transforms.Conversion.MapValueToKey(
        inputColumnName: "Label",
        outputColumnName: "LabelAsKey")
    .Append(mlContext.Transforms.LoadRawImageBytes(
        outputColumnName: "Image",
        imageFolder: assetsRelativePath,
        inputColumnName: "ImagePath"));

IDataView preProcessedData = preprocessingPipeline
                    .Fit(shuffledData)
                    .Transform(shuffledData);

TrainTestData trainSplit = mlContext.Data.TrainTestSplit(data: preProcessedData, testFraction: 0.3);
TrainTestData validationTestSplit = mlContext.Data.TrainTestSplit(trainSplit.TestSet);

IDataView trainSet = trainSplit.TrainSet;
IDataView validationSet = validationTestSplit.TrainSet;
IDataView testSet = validationTestSplit.TestSet;
```

Figure 23: Preprocessing data

Figure 23 depicts the procedure of preparing the data before to the training session. The "LoadImagesFromDirectory" method, which is a utility method, is called the first time the "mlContext" variable is initialized. This method returns a list of images used for training. The images are then loaded into a "IDataView" using the "LoadFromEnumerable" function. The loaded training data is then stored in memory using this "IDataView". The data is loaded in the order that it was read from the directories; however, a method that shuffles it in order to randomize it is used to ensure that the data is evenly dispersed. When dealing with machine learning models, inputs are typically in the form of numbers. As a result, before the training can begin, the data must be prepared in some way. To accomplish this, the label or value to be predicted is converted into a numerical value, and then the images are being read into "byte[]". Furthermore, the "fit" method is applied to the data saved on the "preprocessingPipelin," which is then followed by the "Transform" method, which returns a "IDataView" populated with the pre-processed data. Before attempting to train a model, you must have both a validation set and a training dataset available. The model is trained using training data and then tested using validation data to see how well it can predict outcomes based on data that is yet to be given. To improve, the model revises its previous knowledge based on the results of that performance and applies what it has learned. The validation set is derived from the process of separating the original dataset. The pre-processed data is then divided into two parts: the first portion, 70 percent of which is used for training, and the second part, 30 percent of which is used for validation. Following that, the thirty percent validation set is further broken into validation and test sets, with the former holding 90 percent of the data and the latter containing 10 percent of the data for the testing purposes. The last part of the code just assigns the partitions to their related values for the training, validation and test data.

6.2.5 Defining the training pipeline

```
var classifierOptions = new ImageClassificationTrainer.Options()
{
    FeatureColumnName = "Image",
    LabelColumnName = "LabelAsKey",
    ValidationSet = validationSet,
    Arch = ImageClassificationTrainer.Architecture.ResnetV2101,
    MetricsCallback = (metrics) => Console.WriteLine(metrics),
    TestOnTrainSet = false,
    ReuseTrainSetBottleneckCachedValues = true,
    ReuseValidationSetBottleneckCachedValues = true
};

var trainingPipeline = mlContext.MulticlassClassification.Trainers.ImageClassification(classifierOptions)
    .Append(mlContext.Transforms.Conversion.MapKeyToValue("PredictedLabel"));

ITransformer trainedModel = trainingPipeline.Fit(trainSet);
mlContext.Model.Save(trainedModel, null, "pnmodel3.zip");

ClassifySingleImage(mlContext, testSet, trainedModel);
```

Figure 24: Defining the TrainingPipeline

The figure 24 demonstrates how the training is being initiated. Some of the metrics are set while training and being retrieved for the training purposes. The "ImageClassificationTrainer" takes several parameters.

The column known as "FeatureColumnName" is used as an input by the model. "LabelColumnName" is the name of the column whose value the expression will forecast. The "ValidationSet" is the IDataView that contains the validation data. Arch indicates which of the pretrained architectures of the models to be used. In this case, the ResNetv2 model with the 101-layer version will be used. "Metrics" A callback connects a function to an argument so that it can track progress during training. In the absence of a validation set, the "TestOnTrainSet" command advises the model to compare its performance to that of the training set. "ReuseTrainSetBottleneckCached" Values tells the model whether to employ the cached values from the bottleneck phase in later iterations. The bottleneck phase is a one-time pass-through computation that is particularly resource-intensive the first time it is performed. If the training data remains constant and if we wish to experiment with a different number of epochs or batch size, using cached values significantly reduces the amount of time required to train a model. Even if we want to train the model with a different number of epochs, this is true. In terms of functionality, "ReuseValidationSetBottleneckCached" Values is similar to "ReuseTrainSetBottleneckCached". It just contains the values required for the validation dataset in this particular case.

A variable called ”trainingPipeline” is assigned, which gets the predicted lable and is being stored in a model called ”pmodel3.zip” through ITransformer object. An application programming interface (API), which is further defined in the following section, can now consume and utilise this model (x).

6.2.6 Training/Validation

The Image Classification API begins the training process by loading a TensorFlow model that has already been trained. The training process is divided into two stages: Bottleneck Production Phase and Training Phase.

The collection of training images is loaded during the bottleneck phase, and the pixel values are used as input, or features, for the frozen layers of the pre-trained model. The pre-trained model is at its most accurate during this period. All of the layers in the neural network up to and including the bottleneck layer in certain circles, are included in the frozen layers. Because these layers will not be used for training, they are referred to be frozen. These layers will instead be used for pass-through activities. These frozen layers are where lower-level patterns that help a model differentiate between distinct classes are computed, hence it's critical to keep these layers frozen. As the number of levels in the structure rises, this stage takes progressively more processing power. The results of this one-time calculation may be cached and used in subsequent iterations of the experiment to test out other parameter values.

After the bottleneck phase's output values have been computed, they are used as input in the process of retraining the model's final layer. This technique is performed a number of times defined by the model's parameters. It is a continuous process. Both the loss and the accuracy are evaluated at each stage of the procedure. The model is then enhanced by making the appropriate changes to fulfill the goal of achieving the highest degree of accuracy while incurring the fewest losses. When the training is finished, the model is output in two formats. The first is the model stored in.pb format, and the second in.zip format. Model version that has been serialized with ML.NET. When working in an environment that supports ML.NET, the zip version of the model can be used. If it is other than ML.NET environment then the pb version can be used. The values of some parameters and hyperparameters as well as the model architecture are shown in table 3.

NO.	Hyperparameter/Parameter Name	Value
1	Batch Size	10
2	Epoch	200
3	Learning Rate	0.01
4	Arch	ResnetV2101
5	ScoreColumnName	”Score”
6	PredictedLabelColumnName	”PredictedLabel”

Table 3: Hyperparameters and other parameters

6.2.7 Code Workflow

To have an overview of the code's workflow of creating our machine learning model, it is illustrated in figure 25.

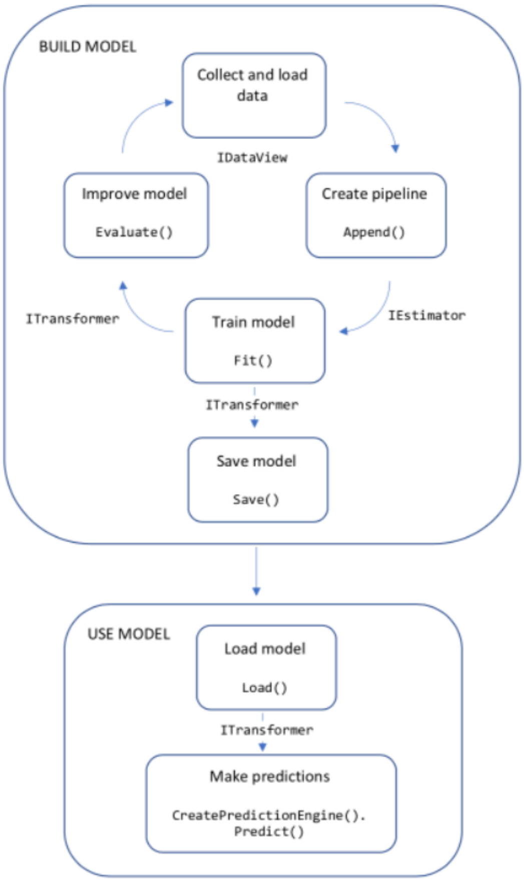


Figure 25: Code Workflow[61]

The above figure depicts the workflow of our code, which is also the workflow that is typically used with the ML.Net project. Figure 25 is comprised of two parts. The first one is building a model, which is what we have done so far in subsection 6.2. The second part is using the created model in other applications, which we have further explained in the subsection 6.3. The terms of the code's workflow are described as below:

Collect and load data: It gets the data and store it in an object called "IDataView".
Create pipeline: Specify a series of operations in a pipeline to extract features and apply a machine learning algorithm.
Train model: Train a model by invoking the pipeline's Fit() function. This is where the training begins.
Improve model: In order to improve the model, evaluate it and then repeat the process.
Save model: Here the model is being stored in a binary format, which can be used in other applications.
Load model: Here we load and use the model by initiating it into an object called "ITrans-

former".

Make predictions: Using the "CreatePredictionEngine.Predict()" method, we can create our own prediction engine by which we can make predictions.

6.2.8 Results

We have tried a variety of pre-defined convolutional neural network methods provided by Tensorflow, and each one has produced its own unique set of outcomes. Among the methods, we have tried the Restnet, VGGNet, MobileNet and Inception model. The ResnetV2 algorithm, which has 101 hidden layers, produced the best results that we were able to achieve.

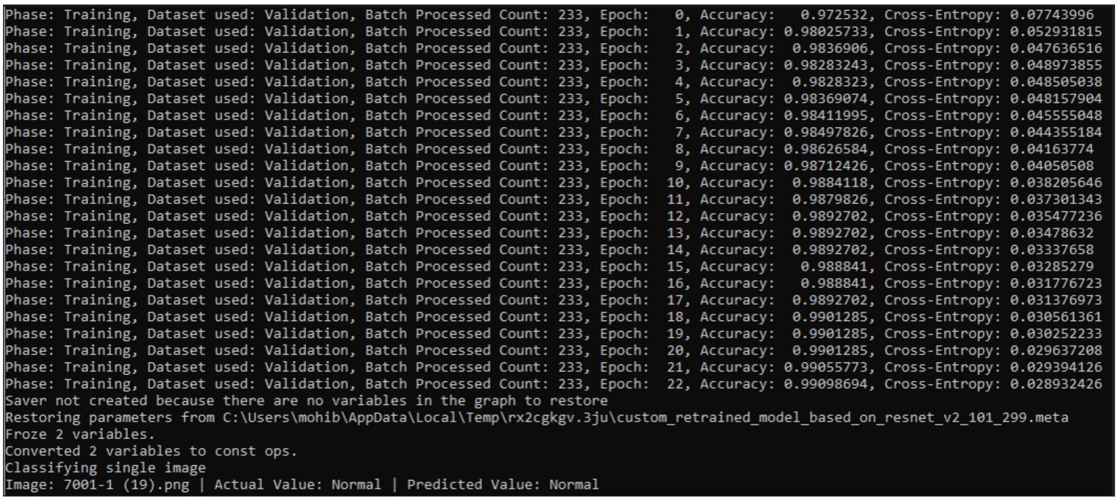
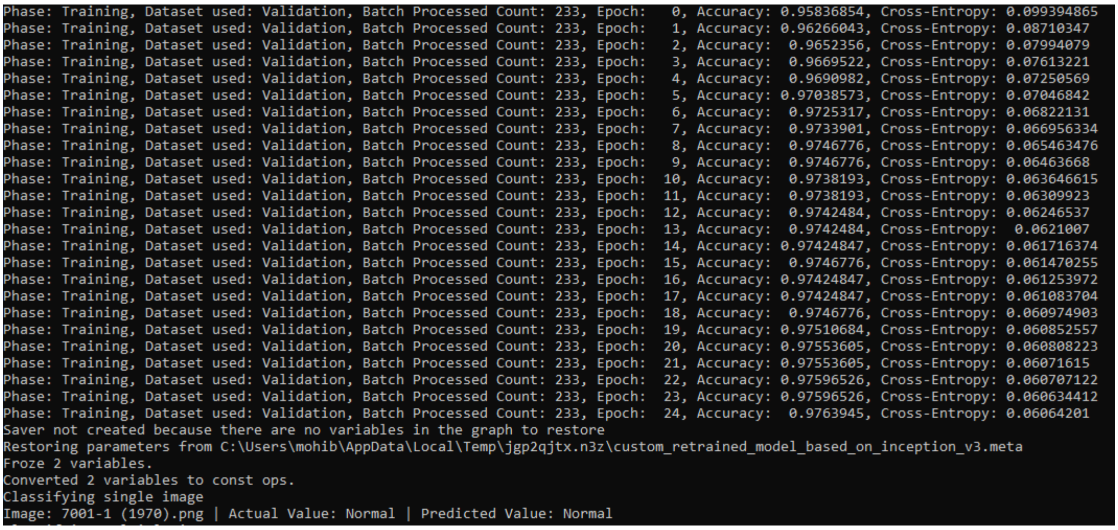

```
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   0, Accuracy:   0.972532, Cross-Entropy: 0.07743996
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   1, Accuracy: 0.98025733, Cross-Entropy: 0.052931815
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   2, Accuracy:  0.9836906, Cross-Entropy: 0.047636516
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   3, Accuracy: 0.98283243, Cross-Entropy: 0.048973855
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   4, Accuracy:  0.9828323, Cross-Entropy: 0.048505038
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   5, Accuracy: 0.98369074, Cross-Entropy: 0.048157904
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   6, Accuracy: 0.98411995, Cross-Entropy: 0.045555048
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   7, Accuracy: 0.98497826, Cross-Entropy: 0.044355184
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   8, Accuracy: 0.98626584, Cross-Entropy: 0.04163774
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   9, Accuracy: 0.98712426, Cross-Entropy: 0.04050508
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  10, Accuracy:  0.9884118, Cross-Entropy: 0.038205646
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  11, Accuracy:  0.9879826, Cross-Entropy: 0.037301343
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  12, Accuracy:  0.9892702, Cross-Entropy: 0.035477236
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  13, Accuracy:  0.9892702, Cross-Entropy: 0.03478632
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  14, Accuracy:  0.9892702, Cross-Entropy: 0.03337658
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  15, Accuracy:   0.988841, Cross-Entropy: 0.03285279
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  16, Accuracy:   0.988841, Cross-Entropy: 0.031776723
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  17, Accuracy:  0.9892702, Cross-Entropy: 0.031376973
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  18, Accuracy:  0.9901285, Cross-Entropy: 0.030561361
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  19, Accuracy:  0.9901285, Cross-Entropy: 0.030252233
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  20, Accuracy:  0.9901285, Cross-Entropy: 0.029637208
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  21, Accuracy: 0.99055773, Cross-Entropy: 0.029394126
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  22, Accuracy: 0.99098694, Cross-Entropy: 0.028932426
Saver not created because there are no variables in the graph to restore
Restoring parameters from C:\Users\mohib\AppData\Local\Temp\rx2cgkgv.3ju\custom_retrained_model_based_on_resnet_v2_101_299.meta
Froze 2 variables.
Converted 2 variables to const ops.
Classifying single image
Image: 7001-1 (19).png | Actual Value: Normal | Predicted Value: Normal
```

Figure 26: Training with ResnetV2_101

Figure 26 illustrates the training results with Resnet, where the maximum of 99% accuracy is achieved.



```
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   0, Accuracy: 0.95836854, Cross-Entropy: 0.099394865
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   1, Accuracy: 0.96266043, Cross-Entropy: 0.08710347
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   2, Accuracy:  0.9652356, Cross-Entropy: 0.07994079
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   3, Accuracy:  0.9669522, Cross-Entropy: 0.07613221
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   4, Accuracy:  0.9690982, Cross-Entropy: 0.07250569
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   5, Accuracy: 0.97038573, Cross-Entropy: 0.07046842
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   6, Accuracy:  0.9725317, Cross-Entropy: 0.06822131
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   7, Accuracy:  0.9733901, Cross-Entropy: 0.066956334
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   8, Accuracy:  0.9746776, Cross-Entropy: 0.065463476
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:   9, Accuracy:  0.9746776, Cross-Entropy: 0.06463668
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  10, Accuracy:  0.9738193, Cross-Entropy: 0.063646615
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  11, Accuracy:  0.9738193, Cross-Entropy: 0.06309923
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  12, Accuracy:  0.9742484, Cross-Entropy: 0.06246537
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  13, Accuracy:  0.9742484, Cross-Entropy:  0.0621007
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  14, Accuracy: 0.97424847, Cross-Entropy: 0.061716374
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  15, Accuracy:  0.9746776, Cross-Entropy: 0.061470255
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  16, Accuracy: 0.97424847, Cross-Entropy: 0.061253972
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  17, Accuracy: 0.97424847, Cross-Entropy: 0.061083704
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  18, Accuracy:  0.9746776, Cross-Entropy: 0.060974903
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  19, Accuracy: 0.97510684, Cross-Entropy: 0.060852557
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  20, Accuracy: 0.97553605, Cross-Entropy: 0.060808223
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  21, Accuracy: 0.97553605, Cross-Entropy: 0.06071615
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  22, Accuracy: 0.97596526, Cross-Entropy: 0.060707122
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  23, Accuracy: 0.97596526, Cross-Entropy: 0.060634412
Phase: Training, Dataset used: Validation, Batch Processed Count: 233, Epoch:  24, Accuracy:  0.9763945, Cross-Entropy: 0.06064201
Saver not created because there are no variables in the graph to restore
Restoring parameters from C:\Users\mohib\AppData\Local\Temp\jgp2qjtx.n3z\custom_retrained_model_based_on_inception_v3.meta
Froze 2 variables.
Converted 2 variables to const ops.
Classifying single image
Image: 7001-1 (1970).png | Actual Value: Normal | Predicted Value: Normal
```

Figure 27: Training with InceptionV3

Figure 27 illustrates the training results with Inception, where the maximum of 97% accuracy is achieved.

```
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:   4, Accuracy:  0.6894527, Cross-Entropy:  1.0065256
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:   5, Accuracy:  0.6922563, Cross-Entropy:  0.9900575
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:   6, Accuracy:  0.6913217, Cross-Entropy: 0.98402214
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:   7, Accuracy:  0.6922563, Cross-Entropy:   0.971753
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:   8, Accuracy:  0.6931909, Cross-Entropy:  0.9732613
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:   9, Accuracy:  0.6922563, Cross-Entropy: 0.95799357
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  10, Accuracy:  0.6875833, Cross-Entropy: 0.96165437
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  11, Accuracy: 0.69412535, Cross-Entropy: 0.94285464
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  12, Accuracy: 0.68664885, Cross-Entropy:  0.9475466
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  13, Accuracy:  0.6931908, Cross-Entropy: 0.92601484
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  14, Accuracy:  0.6913217, Cross-Entropy:  0.9310215
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  15, Accuracy:  0.6941255, Cross-Entropy:  0.9073995
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  16, Accuracy:  0.6931908, Cross-Entropy: 0.91209257
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  17, Accuracy: 0.69599456, Cross-Entropy:  0.8867832
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  18, Accuracy: 0.69692916, Cross-Entropy:  0.8904953
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  19, Accuracy:  0.6978637, Cross-Entropy:  0.8639758
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  20, Accuracy:  0.6987983, Cross-Entropy: 0.86622727
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  21, Accuracy:  0.7016021, Cross-Entropy:  0.8391612
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  22, Accuracy:  0.6997329, Cross-Entropy: 0.83982205
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  23, Accuracy:  0.7044058, Cross-Entropy: 0.81294984
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  24, Accuracy:  0.7053404, Cross-Entropy: 0.81217897
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  25, Accuracy:  0.7090787, Cross-Entropy:  0.7861698
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  26, Accuracy: 0.71094793, Cross-Entropy: 0.78427166
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  27, Accuracy: 0.71094775, Cross-Entropy:  0.7596404
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  28, Accuracy:  0.7118824, Cross-Entropy:  0.7569499
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  29, Accuracy: 0.72029364, Cross-Entropy: 0.73403764
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  30, Accuracy: 0.72029364, Cross-Entropy:  0.7308597
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  31, Accuracy:  0.7249666, Cross-Entropy:  0.7098582
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  32, Accuracy:  0.7259012, Cross-Entropy:  0.7064423
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  33, Accuracy:  0.7324432, Cross-Entropy:  0.6874272
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  34, Accuracy: 0.73377836, Cross-Entropy: 0.68396837
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  35, Accuracy: 0.73845124, Cross-Entropy:  0.6669311
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  36, Accuracy: 0.73845124, Cross-Entropy:  0.6635745
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  37, Accuracy: 0.74125504, Cross-Entropy: 0.64845234
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  38, Accuracy: 0.74218965, Cross-Entropy: 0.64530027
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  39, Accuracy:  0.7444593, Cross-Entropy: 0.63199353
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  40, Accuracy:  0.7453938, Cross-Entropy:  0.6291138
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  41, Accuracy:  0.7463284, Cross-Entropy:  0.6175023
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  42, Accuracy:  0.7463284, Cross-Entropy: 0.61493385
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  43, Accuracy:  0.7481976, Cross-Entropy: 0.60488456
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  44, Accuracy:  0.7500668, Cross-Entropy: 0.60264343
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  45, Accuracy: 0.75140196, Cross-Entropy:  0.5940191
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  46, Accuracy: 0.75233656, Cross-Entropy:  0.5921037
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  47, Accuracy:  0.7495328, Cross-Entropy:  0.5847659
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  48, Accuracy:   0.751402, Cross-Entropy:  0.5831621
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  49, Accuracy: 0.75420576, Cross-Entropy: 0.57697564
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  50, Accuracy: 0.75420576, Cross-Entropy:  0.5756596
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  51, Accuracy:  0.7523367, Cross-Entropy:  0.5704946
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  52, Accuracy:  0.7523367, Cross-Entropy:  0.5694369
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  53, Accuracy:  0.7542059, Cross-Entropy:  0.5651708
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  54, Accuracy:  0.7532713, Cross-Entropy: 0.56433886
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  55, Accuracy: 0.75233674, Cross-Entropy: 0.56085706
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  56, Accuracy: 0.75140214, Cross-Entropy:  0.5602169
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  57, Accuracy: 0.75233674, Cross-Entropy: 0.55741405
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  58, Accuracy: 0.75420594, Cross-Entropy: 0.55693245
Phase: Training, Dataset used: Validation, Batch Processed Count: 107, Epoch:  59, Accuracy: 0.75420594, Cross-Entropy:  0.5547121
Saver not created because there are no variables in the graph to restore
Restoring parameters from C:\Users\mohib\AppData\Local\Temp\5kgh0uc5.nxp\custom_retrained_model_based_on_resnet_v2_101_299.meta
Froze 2 variables.
Converted 2 variables to const ops.
Classifying single image
Image: person264_bacteria_1233.jpeg | Actual Value: Bacterial | Predicted Value: Bacterial
```

Figure 28: Training of bacterial and viral pneumonia

Furthermore, we ran a test to see what accuracy we would have if we had to figure out what form of pneumonia it is, as we have datasets for both bacterial and viral pneumonia. Figure 28 depicts the training results using Resnet, where the accuracy is maximum of 75%, which is significantly lowered due to a lack of data on each kind of pneumonia, hence we have considered both types of pneumonia as one label.

As a result, we have created a model with maximum of 99% of accuracy using Resnet and is being stored in a zip file, which will be consumed by the web application in the next subsection.

6.3 System development

Following the development of our machine learning model, the subsequent stage of our implementation will consist of consuming and utilizing the model in order to generate predictions. We developed a back-end system and a front-end system where a user can interact with. The back-end system is created in the form of a web application ASP.Net using Visual Studio, so that it can generate predictions in response to a request made by the client side. This will function as a server, where our machine learning model will be deployed. The model will make predictions based on the images it receives from the client side. The front-end or the client side is created in Visual Studio Code by using different elements of of various frameworks, including HTML, CSS, JavaScript, JQuery and Bootstrap.

6.3.1 ASP.Net Web Application

We created an application programming interface (API) that may return the outcome of determining whether or not an X-ray image contains pneumonia. Using the machine learning model that we constructed during training, we can make predictions based on the images that a user has provided us with. The system uses this application programming interface (API) to manipulate user data through a range of functions, including CRUD operations (Create, read, update, and delete). This type of API is critical for our system since we need to make predictions based on an image submitted by users, and the results will be returned based on those predictions. The essential components of this form of application programming interface are the controller and the model. Our application programming interface is hosted on the Azure web server. We opted to host the web application on a web server so that we could access our application programming interface (API) from anywhere in the globe.

Model: A model is made up of a collection of classes that collaborate to create the logic for a project. This is performed by mapping our input to the matching model, which allows data access via a number of retrieval and posting activities. To put it other words, mapping our input to the model enables us to retrieve the data [31]. When a client receives a communication request—in our case, a web application with which a user can interact to predict a potential case of pneumonia based on an X-ray image—all of the parameters passed through the request are stored in the corresponding model and then manipulated based on the user's needs. When a client sends a communication request, this occurs. Two separate models are currently being created in order to achieve the project's goals. The first is "ModelInput," which will receive the data that the user is now entering into the system. The second model is "ModelOutput," and it is in charge of storing the data that will be displayed to consumers.

```
8 references
public class ModelInput
{
    1 reference
    public byte[] Image { get; set; }

    0 references
    public UInt32 LabelAsKey { get; set; }

    1 reference
    public string ImagePath { get; set; }

    1 reference
    public string Label { get; set; }
}

7 references
public class ModelOutput
{
    0 references
    public string ImagePath { get; set; }

    0 references
    public string Label { get; set; }

    1 reference
    public string PredictedLabel { get; set; }
    3 references
    public float[] Score { get; set; }
}
```

Figure 29: Web application controller

Figure 29 depicts the input and output model, as well as the characteristics that it possesses.

Controller: The majority of the logic for the methods is contained in the controller, and it is this controller that determines which function or action an endpoint should do. The controller is responsible for establishing contact with the model and selecting the best way to manipulate the data [31]. In other words, it serves as a link between the model and the view. As a result, an endpoint will be created that can have its access revoke upon request. There are several methods available, including post, get, put, and delete.

We created a controller that we named "DetectionController." It takes an image as input, predicts the images using the model we created, and provides the result as a predicted value and score.

```
public ModelOutput Detect()
{
    byte[] filebyte = null;

    var file = Request.Form.Files.FirstOrDefault();
    using (var ms = new MemoryStream())
    {
        file.CopyTo(ms);
        filebyte = ms.ToArray();
    }

    ModelInput input = new ModelInput()
    {
        Image = filebyte,
        ImagePath = "",
        Label = "CD",
    };


    var predictionEngine = PredictionInstance.GetEngine(_memoryCache);
    var modelOutput  = predictionEngine.Predict(input);

    var labelBuffer = new VBuffer<ReadOnlyMemory<char>>();
    predictionEngine.OutputSchema["Score"].Annotations.GetValue("SlotNames", ref labelBuffer);

    var labels = labelBuffer.DenseValues().Select(l => l.ToString()).ToArray();

    var index = Array.IndexOf(labels, modelOutput.PredictedLabel);
    var score = modelOutput.Score[index];
    modelOutput.Score = new float[1];
    modelOutput.Score.SetValue(score, 0);

    return modelOutput;
}
```

Figure 30: Web application models

The above snippet code in the figure 30 demonstrates how the controller collects and returns the input image. Furthermore, to avoid having to load the model each time a request is made, which would result in a delayed response to the user, we load the model only once and then store it in the computer's cache memory. By this way, the response gets much faster as the model is not loading every time a request is made.

6.3.2 Web Interface

After the application programming interface (API) has been developed, a client application with an interface that can be utilized by users is required. This client will begin by establishing a connection, at which point it will be able to accept input from a user in the form of an X-ray image. This image will then be transmitted via this API end-point that we have developed, at which point the results will be displayed. The development of this client made use of numerous components derived from a variety of various frameworks, including HTML, CSS, JavaScript, JQuery, Ajax, and Bootstrap.

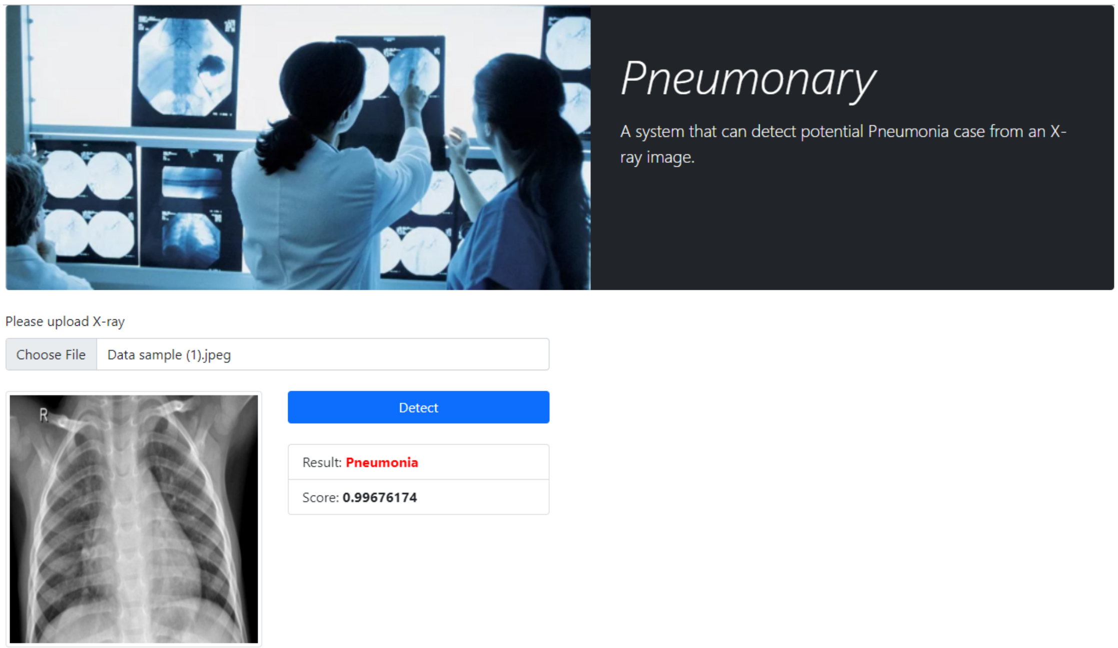


Figure 31: Web interface

The figure 31 is the visualization of the web interface that is being developed. The system works by clicking on the button "Choose File", where they will be prompted with browsing the image that they want to perform prediction. After choosing the image, an image preview will be shown to the user, so they can make sure they have chosen the right image. Finally, the user has to click the "Detect" button, after which the results will be shown to the user, which includes the predictive value that can be either "Normal" or "Pneumonia" and score that describes how sure they system on the predictive value.

7 Discussion

As stated in the introduction, pneumonia has always been a great threat specially for children which needs to be treated as soon as it gets detected. The massive impact it can cause on health is intimidating. We learned from the interviews that countries specially the developing countries have a great need for a solution as they don't have many doctors. Doctors face hundreds of patients with different diseases and symptoms and as a result there is a huge chance for a wrong diagnosis of a disease. Also in developing countries, where the population is quite dense, the pressure a doctor faces on a regular basis is outrageous. As a result, it is also not possible in many cases to give a solution right away. Patients literally wait for days to get their medical report to know what their symptoms are actually about. Cases where the children have pneumonia symptoms but they have to wait until it is confirmed, can prove to be deadly. Other developed countries where the scenario is a bit different than a developing overpopulated country, also has the need to have an accurate diagnosis or detection of serious diseases like pneumonia.

Our Goal was to go for an attempt to create a machine learning model that could have a reliable performance. As a part of this, we also went through similar research that have been done in this field and almost all of them has faced the problem with available data and most of them had produced good accuracy score which were between 75%-90%. Some of them had reached a maximum accuracy of more than 90%, but among those, some had only trained their model on a significantly smaller dataset that cannot be generalized or be reliable. With proper research and analysis we have tried to implement our desired model. Among the different methods that have been applied, RestnetV2101 have outperformed other models and have achieved a maximum of 99% accuracy.

The aim of this, is not to substitute a doctor or a radiologist but rather to assist them, to accelerate their work as their process is quite time consuming. The amount of workload and pressure they have to face, we believe such a system could help them easing the pressure to some degrees. Our prototype also includes an interface to draw a picture such as- how the doctors could interact with such a system easily. We hope this kind of work will not only help them but also will be extremely helpful for patients in emergency. The doctors will not have to stack up the reports to cope up with the pressure of hundreds of patients, they can quickly interact with such a system and let the patients know if they have the disease or not.

Challenges and Limitations: The methods we have been interested at, as discussed in the state of the art section, have been implemented in order to create our pneumonia detection model. The model has shown excellent performance in detecting pneumonia and non-pneumonia or normal cases. However, the unavailability of sufficient data had been a problem. We aimed for classifying between viral and bacterial pneumonia as well but we were limited to a 75.4% accuracy and scarcity of medical images for viral and bacterial pneumonia chest X-rays is a reason behind it. Although anything greater than 70% is considered a good performing model[56], we wanted to achieve an excellent result. This is why we have higher ambition to make our model better once we get more available data or images for viral and bacterial pneumonia in the future. As a result, while our model can distinguish between bacterial and viral pneumonia, the only information displayed on our interface for the time being is whether or not the patient has pneumonia.

8 Conclusion

At the end of this report let us look back to the problem formulation.

How to create a machine learning system that can assist the health sectors in detecting pneumonia cases via chest X-ray of a patient?
To be able to answer the questions, let's start with the sub questions as they will help making the final answer.

Which machine learning techniques can be used to leverage the highest accuracy for identifying a potential pneumonia case?
Deploying deep learning models have always been seen better at being efficient and reliable in terms of performance. From the theoretical research and literature reviews that have been done in this project made it clear that deep learning models are way more popular in image detection and prediction. They are complex and hence better at predicting and detecting complex features.

For the training of the machine learning model, what type of data is necessary, and how can it be accessed?
The data that must be needed to train the machine learning model, are chest X-ray images of pneumonia and it should be as much as possible. There are several ways to access the data. The best option is to collaborate with a hospital that agrees to share some of their patients data but this is too private and consent of the patient is the most important thing in this regard. However, there are other ways to excess data which can be found in various websites that are free to be used for research purposes. These are also X-ray images of patients that have been made available for people who want to train machine learning models for pneumonia detection.

How much research has been done in this field, and how far has the problem progressed in terms of resolution?
A lot of research have been done in past few years in this field. However, machine learning model for auto detection of chest related diseases are not common in healthcare sector to be used by the doctor. This is due to the insufficient availability of data where most of the researchers have struggled to train their models. A lot of models have generated performances from satisfactory to excellent while being trained on the limited available dataset.

Now, let's return to the main question that needs to be answered. To be able to assist the health sector in terms of detecting pneumonia from chest X-rays, a machine learning model needs to be reliable first. And the way to be reliable is to show performance of the highest standard. We tried utilizing the state of the art deep learning methods that are renowned for image detection and prediction. We couldn't have the opportunity to get new hospital data, so we used the dataset that was readily available for us. We went through a brief literature review which helped us to eliminate the worst and utilize the best on our way to designing our machine learning model. We have applied the inception method which has obtained a maximum of 97% accuracy and the ResNet101 has produced the highest accuracy which reached a maximum of 99%.

8.1 Future work

Because of the time limit, data and other challenges we had during the development of this project, there are several things that we won't be able to execute now but are interested to add to our system "Pneumonary" in the future:

- We still hope to get an opportunity for collaborating with a hospital someday where we can work with patients and get more data.
- With more data or by doing data augmentation we might have what it takes to distinguish very efficiently between the types of pneumonia. We would like to add features in our system where it will say which type of pneumonia the patient has.
- In the future, if possible, we would like to add more datasets of different diseases other than pneumonia and try to train our model on those datasets and find out if our machine learning model is able to predict various diseases.

9 References

1. Johns Hopkins. *Pneumonia*[Online]. Available: https://www.hopkinsmedicine.org/health/conditions-and-diseases/pneumonia. Accessed on March 14,2022.

2. National Hospital Ambulatory Medical Care Survey. *National Hospital Ambulatory Medical Care Survey:2015 Emergency Department Summary Tables*, 2015[Online]. Available: https://www.cdc.gov/nchs/data/nhamcs/web_tables/2015_ed_web_tables.pdf.

3. National Vital Statistics Reports. *Deaths: Final Data for 2015*,, Vol. 66, No. 6 [Online], November 27, 2017. Available: https://www.cdc.gov/nchs/data/nvsr/nvsr66/nvsr66_06_tables.pdf.

4. World Health Organization.(2021, Nov. 11). *Pneumonia* [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/pneumonia.

5. S.Naicker, J. Plange-Rhule, R. C Tutt, J. B Eastwood. (2009). *Shortage of healthcare workers in developing countries–Africa*[Online]. Available: https://pubmed.ncbi.nlm.nih.gov/19484878/.

6. National Heart,Lung and Blood Institute. (2022, Mar.24). *What is Pneumonia*[Online]. Available: https://www.nhlbi.nih.gov/health/pneumonia.

7. National Library of Medicine. (2016, Jan.4). *Early identification and treatment of pneumonia: a call to action*[Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5357734/.

8. National Library Of Medicine. (2019,Jan.22). *Comparing Sensitivity of Ultrasonography and Plain Chest Radiography in Detection of Pneumonia; a Diagnostic Value Study*[Online]. Available:
urlhttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC6377225/.

9. National Library Of Medicine. (2019,Oct.4). *The impact of artificial intelligence in medicine on the future role of the physician*[Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6779111/.

10. MathWorks.(2016) *Introducing Machine Learning* [Online]. Available: https://fr.mathworks.com/content/dam/mathworks/tag-team/Objects/i/88174_92991v00_machine_learning_section1_ebook.pdf.

11. P. wang, E.Fan, P. Wang. (2021,Jan). *Comparative analysis of image classification algorithms based on traditional machine learning and deep learning*[Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865520302981.

12. Thyago P.Carvalho, F. A. A. M. N.Soares, R. Vita, R. da P.Francisco, J. P. Basto, S. G. S.Alcalá. (2019,Nov).*A systematic literature review of machine learning methods applied*

to predictive maintenance[Online]. Available: https://doi.org/10.1016/j.cie.2019.106024.

13. R. Gandhi. (2018,Jun. 7). *Support Vector Machine — Introduction to Machine Learning Algorithms.*[Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

14. V K. Ayyadevara. (2018). *Pro Machine Learning Algorithms*[Online]. Available: https://link.springer.com/content/pdf/10.1007/978-1-4842-3564-5.pdf.

15. Live Code Stream. (2020,Jun. 2). *7 Steps of Machine Learning*[Online]. Available: https://livecodestream.dev/post/7-steps-of-machine-learning/.

16. Yufeng G. (2017,Aug. 31). *The 7 Steps of Machine Learning*[Online]. Available: https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e.

17. Aujsmr. *An Extensive Review Of Patient Health Care Service Satisfaction In Bangladesh*[Online]. Available: https://aujmsr.com/an-extensive-review-of-patient-health-care-service-satisfaction-in-bangladesh/.

18. Dataset 1. Available: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia.

19. Dataset 2. Available: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database?resource=download.

20. A. Bonner. (2019,Jan. 19). *The Complete Beginner's Guide to Deep Learning: Artificial Neural Networks*[Online]. Available: https://towardsdatascience.com/simply-deep-learning-an-effortless-introduction-45591a1c4abb.

21. L. Shukla. (2019,Aug. 14). *Fundamentals of Neural Networks*[Online]. Available: https://wandb.ai/site/articles/fundamentals-of-neural-networks.

22. DataRobot. (2018,Apr. 30). *Introduction to Loss Functions*[Online]. Available: https://www.datarobot.com/blog/introduction-to-loss-functions/.

23. GeeksforGeeks. (2020,Oct. 8). *Activation functions in Neural Networks*[Online]. Available: https://www.geeksforgeeks.org/activation-functions-neural-networks/.

24. S. I. Serengil. (2017,Jan. 29). *Hyperbolic Tangent as Neural Network Activation Function*[Online]. Available: https://sefiks.com/2017/01/29/hyperbolic-tangent-as-neural-network-activation-function/.

25. S. Sharma. (2017,Sep. 6). *Activation Functions in Neural Networks*[Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

26. DEEPLIZARD. *Deep Learning Fundamentals - Classic Edition*[Online]. Available: `https://deeplizard.com/learn/video/U4WB9p6ODjM`.

27. Baeldung. (2021,Feb. 27). *Epoch in Neural Networks*[Online]. Available: `https://www.baeldung.com/cs/epoch-neural-networks`.

28. B. Kanani. (2019,Dec. 8). *Neural Network: Introduction to Learning Rate*[Online]. Available: `https://studymachinelearning.com/neural-network-introduction-to-learning-rate/`.

29. S. Karasik. (2021,Apr. 13). *ML.NET: A Robust Framework for Implementing Machine Learning in .NET Environments*[Online]. Available: `https://exadel.com/news/ml-net-a-robust-framework-for-implementing-machine-learning-in-net-environments/#:~:text=Python%20is%20irreplaceable%20for%20Machine,NET%20environment%20more%20quickly`.

30. ML.NET. *ML.NET Documentation*[Online]. Available: `https://docs.microsoft.com/en-us/dotnet/machine-learning/`.

31. MVC Framework - Introduction. *MVC Framework Tutorialspoint*[Online]. Available: `https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm#:~:text=The%20Model%2DView%2DController%20(,development%20aspects%20of%20an%20application`.

32. Great Learning.(2020,Sep. 28). *Introduction to Resnet or Residual Network*[Online]. Available: `https://www.mygreatlearning.com/blog/resnet/`.

33. NeuroHive. (2019,Jan. 23). *ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks*[Online]. Available: `https://neurohive.io/en/popular-networks/resnet/`.

34. S. Yegulalp. (2019,Jun. 18). *What is TensorFlow? The machine learning library explained*[Online]. Available: `https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html`.

35. R. Alake. (2020,Dec. 22). *Deep Learning: Understanding The Inception Module*[Online]. Available: `https://towardsdatascience.com/deep-learning-understand-the-inception-module-56146866e652`.

36. Albahli S, Rauf HT, Algosaibi A, Balas VE. "AI-driven deep CNN approach for multi-label pathology classification using chest X-Rays".*PeerJ Computer Science*,2021,7:e495.

37. Rahman, Tawsifur, Muhammad E.H. Chowdhury, Amith Khandakar, Khandaker R. Islam, Khandaker F. Islam, Zaid B. Mahbub, Muhammad A. Kadir, and Saad Kashem. "Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection Using Chest X-ray". *Applied Sciences*, vol. 10, no. 9, p. 3233, May 2020, doi: 10.3390/app10093233.

38. G. Liang, L. Zheng. (2020,Apr.). *A transfer learning method with deep residual network for pediatric pneumonia diagnosis*[Online]. Available: `https://www.sciencedirect.com/science/article/abs/pii/S0169260719306017?via%3Dihub`.

39. Ibrahim, A.U., Ozsoz, M., Serte, S, F. A. turjaman, P. S. Yakoi. "Pneumonia Classification Using Deep Learning from Chest X-ray Images During COVID-19". *Cogn Comput*, 2021. Available: `https://doi.org/10.1007/s12559-020-09787-5`.

40. P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, R. L. Ball, C. Langlotz, K. Shpanskaya, M. P. Lungren, A. Y. Ng. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*, arXiv:1711.05225v3[Online], Dec 25,2017. Available: `https://arxiv.org/pdf/1711.05225.pdf`.

41. U. Shah, A. ABD-Alrazeq, T. Alam, M. Househ, Z. Shah. "An Efficient Method to Predict Pneumonia from Chest X-Rays Using Deep Learning Approach". *The Importance Of Health Informatics In Public Health During A Pandemic*, 272 pp. 457, 2020.

42. I. Sommerville. *Software Engineering*,9th ed.,[Online],2011. Available: `https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf`

43. S. Albahli, H. T. Rauf, M. Arif , Md. T. Nafis, A. Algosaibi. "Identification of Thoracic Diseases by Exploiting Deep Neural Networks". *Computers, Materials & Continua*, 2021.

44. T. Chandra, K. Verma. "Pneumonia detection on chest X-Ray using machine learning paradigm". Proceedings Of 3rd International Conference On *Computer Vision And Image Processing*, pp. 21-33, 2020.

45. K. M. Kuo, P. C. Talley, C. H. Huang, L. C. Cheng. "Predicting hospital-acquired pneumonia among schizophrenic patients: a machine learning approach". *BMC Medical Informatics and Decision Making*, Article 42, 2019.

46. H. Yue, Q. Yu,C. Liu, Y. Huang, Z. Jiang, C. Shao, et al.. "Machine learning-based CT radiomics method for predicting hospital stay in patients with pneumonia associated with SARS-CoV-2 infection: a multicenter study". *Annals Of Translational Medicine*, 8, 2020, pmid:32793703.

47. B. Antin, J. Kravitz, E. Martayan. " Detecting pneumonia in chest X-Rays with supervised learning", *Semanticscholar. Org.*, 2017.

48. H. Sharma, J. Jain, P. Bansal, S. Gupta. "Feature extraction and classification of chest x-ray images using CNN to detect pneumonia", in *2020 10th International Conference On Cloud Computing, Data Science & Engineering (Confluence)*, pp. 227-231, 2020.

49. O. Stephen, m. Sain, U. Maduh, D. Jeong. "An efficient deep learning approach to pneumonia classification in healthcare". *Journal Of Healthcare Engineering 2019*, 2019.

50. J. Janizek, G. Erion, A. DeGrave, S. Lee. "An adversarial approach for the robust classification of pneumonia from chest radiographs". *Proceedings Of The ACM Conference On Health, Inference, And Learning.* pp. 69-79, 2020.

51. J. Zhang, Y. Xie, G. Pang, Z. Liao, J. Verjans, W. Li, Z. sun, J. he, Y. Li, C. Shen, Y. Xia. " Viral Pneumonia Screening on Chest X-rays Using Confidence-Aware Anomaly Detection". *IEEE Transactions On Medical Imaging*, 2020.

52. ProductPlan. *MoSCoW Prioritization*[Online]. Available: `https://www.productplan.com/glossary/moscow-prioritization/#:~:text=MoSCoW%20prioritization%2C%20also%20known%20as,will%20not%20have%20right%20now`.

53. T. Tuncer, F. Ozyurt, S. Dogan, A. Subasi. "A novel Covid-19 and pneumonia classification method based on F-transform". *Chemometrics And Intelligent Laboratory Systems.* 210 pp, 104256, pmid:33531722, 2021.

54. A. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, J. Rodrigues. "Identifying pneumonia in chest X-rays: A deep learning approach". *Measurement*, 145 pp. 511–518, 2019.

55. T. Gabruseva, D. Poplavskiy, A. Kalinin. "Deep learning for automatic pneumonia detection". *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition Workshops.* pp. 350-351, 2020.

56. K. Barkved. (2022, Mar. 9). *How To Know if Your Machine Learning Model Has Good Performance*[Online]. Available: `https://www.obviously.ai/post/machine-learning-model-performance`.

57. T. Pollock. (2020). *The Difference Between Structured, Unstructured Semi-Structured Interviews*[Online]. Available: `https://www.oliverparks.com/blog-news/the-difference-between-structured-unstructured-amp-semi-structured-interviews`.

58. Creately. (2021, Sep. 10). *UML Diagram Types Guide: Learn About All Types of UML Diagrams with Examples*[Online]. Available: `https://creately.com/blog/diagrams/uml-diagram-types-examples/`.

59. Tensorflow. *Transfer learning and fine-tuning*[Online]. Available: `https://www.tensorflow.org/tutorials/images/transfer_learning`.

60. I. Castiglioni, L. Rundo, M. Codari, GiovanniDi Leo, C.Salvatore, M. Interlenghi, F.Gallivanone, A. Natascha, C. D'Amico, F. Sardanellifi. "AI applications to medical images: From machine learning to deep learning". *Physica Medica*, Vol-83, pp. 9-24, March 2021.

61. Microsoft. (2021,Sep. 15). *What is ML.NET and how does it work?*[Online]. Available: `https://docs.microsoft.com/en-us/dotnet/machine-learning/`

`how-does-mldotnet-work`.

A Appendix

A.1 Interviews

As a part of the project, we needed some expert interviews in order to gather more information about pneumonia and know their opinion about our work. We contacted several doctors from countries like Bangladesh, United states and Denmark in order to get a picture internationally. We made the interviews in a semi structured way, where we wrote down some questions beforehand and some questions were asked as we went on further.

A.1.1 Semi structured Interview Guide

A semi structure interview guide was made for the experts. The guide is as follows:

Introduction

- An introduction about ourselves.
- 'We thank you in advance for agreeing to have an interview with us. The format of the interview will be semi structure. Therefore we have made some questions but with no specific structures so that it remains flexible. In case you don't want to respond to a question (confidentiality), please let us know, so that we can skip it. We would also like to ask you if we could record the interview? Just in case to use it for our project later.'
- A small introduction about our work.

Questions

- What are the latest technologies and trends used for detecting something via X-ray Image?
- How difficult is it to diagnose pneumonia in the X-ray image or what are the Challenges?
- How often can it be misdiagnosed with X-rays?
- How crucial is it to identify the disease as early as possible?
- What is the process of diagnosing pneumonia and how long does it take for a radiologist/ doctor to identify potential cases of pneumonia?
- How costly is it for doctors to diagnose pneumonia through an X-ray image and would a machine learning algorithm help save time and money in assisting to diagnose the disease?
- What do you expect from such a solution?
- Are there any regulations you are aware of that might be a barrier to this solution? Or Are there any regulations against such a solution where the machine publishes or detects the disease?

- What do you expect in the future ? What do you think will be taking over the existing solutions today?

- 'Thank you so much for your valuable time. We are confident that your answers will help us immensely. Have a nice day.'

A.1.2 Interview with Dr. Nowshin

Here is the complete interview with Dr. Nowshin who is a pediatrician and works with ailing children in a swedish company in Chittagong, Bangladesh.

-Diganta: Hi Nowshin. Thank you for taking your time to join us in this Interview. This is my groupmate Muheb. Could you please Introduce yourself?

-Nowshin: Hello, I'm doctor Nowshin, working from Bangladesh. currently I'm working as a pediatrician in a Swedish organization called Terrades Homes and I'm working with ailing children. Nice to meet you.

-Diganta: Nice to meet you too.

-Muheb: Thank you.

-Diganta: As you know I am Diganta, I am studying masters on Innovative communication technology and entrepreneurship. Currently I am working on my master thesis.

-Muheb: I am muheb. I am also studying in the same university as Diganta and we are doing the master thesis together.

-Diganta: Thank you for your accepting our request and agreeing to have a interview with us. I hope it's Okay that we are recording it. Just to make sure that we can write all the stuff down, we need to record it. I hope you are Okay with that.

-Nowshin: Yeah, I am completely okay with that. Go ahead.

-Diganta: So shall we start our questions?

-Muheb: Yeah. So our first question would be what are the latest technologies and trends used for detecting something via X-ray?

-Nowshin: Okay, thank you for your question. Actually, there are no such latest technologies that we depend on for diagnosing a chest X-ray of pneumonia. It's basically based on clinical knowledge that we are taught in our medical schools by our mentors and by gathering that knowledge we are taught some clues and we try to identify them and diagnose them and treat accordingly.

-Diganta: So is it difficult for you to diagnose pneumonia on your own?

-Nowshin: Yeah. Sometimes it's very challenging for us. Especially for our young doctors. who have recently completed their graduation and it's quite a knowledgeable thing together. So much information to diagnose the X-ray because there are lots of signs and symptoms of it. And there's also a subject based on radiology and that is a very huge subject. People do masters on that. So for our junior doctors, we normally can diagnose the common cases, but sometimes we face challenges in diagnosing them when it becomes difficult.

-Diganta: Okay. Is there any cases where you get any kind of false positive or false negative results?

-Nowshin: Yeah, quite often. Because it's based on our knowledge and we can sometimes be wrong. We can often be wrong and most people don't understand the radiology in their MBBS, so sometimes the signs symptoms of the child and the extreme edges are very misleading. So it gets very difficult for us. So I think this is a very tough job for us.

-Diganta: Is it very important to detect pneumonia as early as possible?

-Nowshin: Obviously this is the most important thing because for example, a pneumonic child, as I'm a pediatrician, I will discuss it in my point of view, this is a very critical case and a medical emergency that needs to be tweaked urgently and needs to be treated urgently. And if the diagnosis is made quickly, we can promptly get to the treatment and the child gets improved. So it is very important to diagnose these. As soon as possible.

-Diganta: Okay, great. So, uh, can you tell a little bit about what is the process, how do you do it? How do you detect Peumonia if you know anything about it?

-Nowshin: Okay, if I want to describe it, I would like to talk about something that we learned in our medical school that we are taught. There are some signs like consolidation and some signs like bronchopneumonia. There are patchy opacities like that and most of the OPEC shadows we get in the X-rays in the region of the lung and doctor detects the point. So we suspect that this might be a case of pneumonia, but I'm not a specialist So...... but these are the key factors.

-Diganta: Do you think it's a very costly process?

-Nowshin: I would say the process is not that costly because you just need to do an X-ray and in our country and most of the hospitals provide cheaper X-rays after getting the report. This is a this is a process you can call time consuming but not costly because if you want to get. The report of the chest X-ray, if you have to submit that in the hospital then you have to wait for a radiologist to make a report of it, then you have to wait for the delivery. I have seen many cases like this, so I would say cost is not the main problem. It's very time consuming which is a big problem.

-Diganta: Okay. That was some great information we gathered. So now we would like to talk about what we are doing for our master thesis. Muheb can you introduce our project?

-Muheb: Yeah. We are trying to make a machine learning algorithm which can detect pneumonia by scanning an X-ray image, it will detect whether the X-ray image has pneumonia or not. So my question to you would be would be helpful for you in reducing time consumption? As you said, It's time consuming. So would it save time and money for the doctors or radiologist by assisting them?

-Diganta: Basically what you do is, you give it an X-ray image and it detects if the patient has pneumonia or not within very short time. It can take only one second or less to do so. So if this kind of things are introduced in hospitals or with doctors, do you think it's a good thing to assist doctors? How do you think that the machine can help?

-Nowshin: Yeah, I think this is a very good initiative because I think it will be very helpful for specially the junior doctors who are recently graduated and have little knowledge about the radiology or reading chest X-rays. They are very naive in diagnosing them. I think this can be very helpful although it might not be the ultimate thing because some clinical knowledge will definitely be needed in diagnosing pneumonia, but it will be a very helpful tool and sometimes it will save a lot of time in case of diagnosing immediate case that needs to be treated urgently and immediately. Otherwise a life can be lost. So I think this machine learning process will be very handy in assisting the medical personnel. And sometimes, there are some students who are doing their internships in the medical colleges, and if we get a set up like that, it will be useful for the doctors and as well as for the patients also.

-Diganta: Thank you so much for the answer. We actually got a lot of information from your answer. We want to ask you a few more questions. So now my question would be, what do you expect from this kind of things if you are given something that can assist you, what do you expect from a system like this? For example, do you want it to be very easy to use and do you think it should have some criteria to be able to be accessed in a way that could be very handy to you?

-Nowshin: Yeah. I think as per my country, I should talk about my benefits. So I would like to expect something that it must be cheaper in cost and sometimes most of the people are not so aware about the technologies. So they have to be quite simple, so they can use it very easily. This can be a criteria and other things such as this machine should not be a complicated thing, and I expect one more thing, that the most important clinical criteria of pneumonia should be fulfilled by the machine which means the most common points of pneumonia shouldn't be missed by this system. That will also help the junior doctors by not misleading them.

-Diganta: Right now we are working on two different types of pneumonia, viral and bacterial pneumonia. In the future, we would like to cover more aspects and features of this disease and other diseases if we work on with it further, right Muheb?

-Muheb: Yes exactly.

-Nowshin: Yeah, of course. And I think there are some diseases which need immediate attention and their one and only diagnostic criteria is X-ray like perforation, perforation is an immediate disease that that should be treated urgently and by surgical operation. And this is solely diagnosed by an X-ray and if you in future develop your machine in diagnosing such diseases, this will be very life changing thing for medical sectors, I guess.

-Diganta: That is great to know. Do you have anything to ask Muheb?

-Muheb: Yeah. Our next question would be, are there any regulations that you are aware of that might be a barrier to this solution?

-Nowshin: Actually there is no rule like this. But one thing I would like to say, there can be a negative point using solutions like this. It might limit the source of our knowledge. For example if diagnosing pneumonia becomes so easy, then new students will rely on it much more rather than studying and getting knowledge themselves. As a doctor I don't see any regulations against it but only concern is about limiting knowledge.

-Muheb: What do you expect in the future?

-Diganta: Do you think it will go like the traditional way that the way it is happening now. For example, in Bangladesh people are mostly doing it manually, right? You check everything manually. Do you think in the future machines will affect the Medical Sciences?

-Nowshin: I see a very promising future. This kind of ideas can bring a lot of positive effects in the medical sector. For example, a country like Bangladesh, we get to see 15-20 patients an hour which is very challenging and time consuming. For doctors who are new and just started their career, it's very much difficult for them and these kind of brilliant ideas can make a huge change in that scenario. With that extreme pressure of patients, it's also possible to get misleading results. If something like this is introduced, it will not only save lives but also make doctors life easier.

-Diganta: Thank you for your answer. Muheb do you have any last questions?

-Muheb: Yes, I have one last question. Do you use any kind of machine learning solutions in your country?

-Nowshin: No, we haven't seen anything like that yet.

-Diganta: Well, I guess we are at the end of our interview. It's been a great interview. We have learned a lot and gathered a lot of information. Thank you so much for spending your valuable time with us. We are grateful for that.

-Nowshin: Thanks to you too. I also learned a lot of things about machine learning today which I didn't have much idea about. It's amazing to know what you can do with your idea. Thank you for including me in your thesis and I really think it's a great idea.

-Muheb: Thank you so much. Have a nice day.

-Diganta: Have a nice day.

-Nowshin: You too.

A.1.3 Interview with Dr. Sulagna

Here is the complete interview with Dr. Sulagna who is a physician at kettering health hospital in Ohio, USA.

-Diganta: Hi Sulagna! Can you hear me?

-Sulagna: Yes, I can Hear you.

-Diganta: Okay, let's get started.

-Sulagna: Okay.

-Diganta: Thank you for agreeing to participate in the interview and it means a lot to us and we hope we will try to make it as short as possible.

-Diganta: I'm doing a masters in Innovative communication technology and Entrepreneurship at Aalborg university, Copenhagen and I'm doing a project on Machine learning. We're creating a machine learning algorithm. I think it's better if I introduce you to my project first before I ask you some questions.

-Sulagna: Sure, Go ahead.

-Diganta: We are making a machine learning algorithm that can detect two types of pneumonia. For now one is viral pneumonia and the other one is bacterial pneumonia. So basically you give us an X-ray image and our algorithm detects if that patient has pneumonia or not. That's what this is about and We would like to ask you some questions if you are OK with it. And also we would like to record it if you have no problem with it.

-Sulagna: Thank you Diganta for allowing me to participate in your project. I think it's a very good idea. And then the reason and idea behind it is absolutely brilliant. It will be my pleasure to take part. Yeah, you're free to ask me any question you would like, and then I will try my best to help you as much as I can.

-Sulagna: Would you require me to introduce myself or you know that already?

-Diganta: Yeah, I would like to. Please can you tell me What do you are doing right now?

-Sulagna: So my full name is Sulagna and I am doing Internal medicine residency in Ohio, USA at Kettering Hospital. I'm almost done with my first year. I'm gonna start my second year residency very soon.

-Diganta: Great! Shall I ask you some questions regarding our master thesis?

-Sulagna: Sure! Go ahead.

-Diganta: Yeah. Okay. So can you tell me what are the latest technologies for now to detect something via X-ray? What kinds of stuff do you have in the United States to detect something from X-ray? Do you do it manually or do you use anything?

-Sulagna: Well, that's a very good question. To be honest, it's the radiologist who does the imaging, but as far as I know they use plain X-ray phones to detect pneumonia. It's mostly the infiltrates that we look for when it comes to the type of the viral pneumonia versus bacterial. There is a very thin line between those, so if there has to be like bilateral traits or this unilateral, it depends on what lobes they're affecting. So yeah. And then I think they use very simple plain X-ray detection if it's very much detectable, if it's not complicated or a new one. But if it seems like a complicated case, say if it's due to COVID-19 or some other complicated bacteria then you would require a CT scan as well, so yeah, that's what I know.

-Diganta: Okay. So Do you know how difficult it is to detect something from an X-ray image? I mean, by a radiologist or by a Doctor Who's trying to figure out something from the X-ray?

-Sulagna: Well, if we make the diagnosis already, I mean, especially when it comes to pneumonia, there are certain symptoms and then lab findings. So before we go for the X-ray, we already have something in our mind. So it should not be that difficult when you see the X-ray, if you see the diagnostic infiltrates, or the finding, or the consolidation in the X-ray that goes with your history, it becomes almost definite to diagnose that case.

-Diganta: Okay. So do you know what is the process of doing it? What are the steps to detect pneumonia from an X-ray image?

-Sulagna: Well, that should be very simple. They don't need or actually require any kind of pre imaging prep. So you just go for it and then then you know you cannot have any metals in your body. They do the X-ray in different positions that could be lateral, it could be posterior etc.

-Diganta: So do you think there is a possibility of getting false positive or false negative results?

-Sulagna: There is definitely. There is definitely a possibility of getting a false positive because say, someone who has been lying in the bed for a long time, they're not moving and they're bedridden. So sometimes a lot of them are unable to clear the secretions. So it kind of, it tends to develop a condition we call atelectasis. So that's like a collection of fluid in your lung when you're unable to clear the secretions through coughing or all the reflex mechanisms. So in that case, sometimes it can be mistaken with the pneumonia.

-Diganta: Okay. So is it a costly procedure to detect something from X-ray images?

-Sulagna: Well, it does cost money, but I don't think it's that costly. It should be very noninvasive and simple. I would say it's very affordable, in the US at least.

-Diganta: Okay, thank you for the answer. I would like to say first of all, our project is not about replacing any medical knowledge but rather we want to assist the doctors and say, you have a lot of patients and then you need to go through a lot of extra images. So let's say if we make an algorithm that can detect pneumonia from X-ray images, for example, if you give it to a machine, the machine will tell you the results within less than a second. Do you think it's helpful or do you think there are some positive or negative aspects of it?

-Sulagna: Well, absolutely. I think it's a very good idea. Correct me if I'm wrong. I think it's very similar to what Apple watches produce nowadays. So nowadays the watches can detect your heart rate and then they can detect if you're having a heart attack or sometimes like If you're having an arrhythmia. So they give you kind of a hint and then that makes you aware and then you go to the emergency department and then you can be treated as early as possible. So that has a benefit to it. So from what I understand, your technology actually kind of gives that kind of benefit. So say if someone is having symptoms and then they use your technology, they might be having the idea. 'Okay, maybe I'm having symptoms' and then they will go to the ER and then have an X-ray to confirm it. So absolutely, I think it will be very beneficial not only to the patients the most, but also to the doctors.

-Diganta: Okay. What kind of things do you expect from such technologies? Like if you have a machine that can detect pneumonia in your Chamber, So what do you expect like, for example, do you want it to be easy to use and what kind of features do you expect in a system like that?

-Sulagna: Well, that's again a very good question. Like I said, the doctors most of the time, they have an idea that, Okay, this person is having symptoms very suggestive on pneumonia. But if we have a technology saying, 'Okay, I am sensing something serious' Or that's going more towards our suspicion. So that will be very helpful by giving us hints about where we don't even need to use the imaging. That would be very helpful because a lot of our patients are sometimes critically ill patients, sometimes it's very difficult to move them from the hospital room to the imaging room frequently. And if it can kind of give us a hint which goes with our clinical findings, in that case, it would be very helpful to both the physicians and the patients.

-Diganta: Okay. So are there any rules and regulations that can be against using a machine in your chamber whenever you are checking a patient?

-Sulagna: I'm not sure if physicians have anything to do with it. If it's approved by the government. Yeah, but if it ensures a patient's safety, it should be okay by the physicians.

-Diganta: That's great. Thank you. That's nice. And then do you expect anything in the future for solutions like this? Do you think that introducing machine learning that can do it earlier, can do it faster, would help you in the long run?

-Sulagna: Well, like I said, it doesn't matter if it's one or hundred percent accurate or not, because most of the physicians, they already have an idea. Say someone is coming down with a fever, cough, shortness of breath or, you know, like cough productive situation, they already have a kind of an idea. And then if your app or technology helps us more. You know, it gives us evidence more towards our suspicion that will be, of course very helpful and that's what we expect. And if it's fast enough that sometimes there are certain kinds of pneumonia where you need to start your antibiotic very quickly. So based on the findings that your technology gives us, if we can reach our diagnosis or suspicion early enough, we can start the treatment as early as possible, so it should definitely be very helpful.

-Diganta: Great, so do you have anything to ask or do you have anything to suggest? What can we do to make it better?

-Sulagna: Since I don't know exactly what you're going to do but once it comes with you guys, whatever ideas you're putting into it, once it's more clear then I think I would be able to comfortably give you the answer, but I'm quite impressed with the idea. I think it will be very helpful for everyone.

-Diganta: Okay. Thank you so much. That was so helpful, I think, I'm done with the questions now and thank you for your input. I think this will help us a lot because we now have a lot of information and we can work towards it. Thank you so much for giving your valuable time.

-Sulagna: Okay. Yeah, it was a pleasure. I wish you guys good luck and I'm sure it will be great. And then if you need any help in future, just let us know. It was a pleasure to help.

-Diganta: Thank you. Have a nice day.