Identification and Classification of Viral Pneumonia by Image-Based Deep Learning and Model interpretability

Identification and Classification of Viral Pneumonia by Image-Based Deep Learning and Model interpretability

A project report submitted in partial
fulfillment of the requirements for the degree of
Master of Science

By

Hamza Khokhar

University of Colorado Denver, 2021
Master of Science in Computer Science

Metropolitan State University of Denver, 2019
Bachelor of Science in Computer Science

ABSTRACT

This project aims to identify if a patient has Pneumonia by looking at their chest X-ray. CNN (Convolutional Neural Network) will be used to identify if a given chest X-Ray has Pneumonia or not. Then we will deep dive into the model itself to understand why the model is predicting if a given chest X-Ray has Pneumonia or not. Model interpretability is a fairly new approach in the world of Artificial Intelligence, some time ago it was thought that Deep learning models are like black boxes but with more research in this area, we can have more confidence in our model and their real-world usage such as predicting Pneumonia. This problem is interesting because if we can identify if a person has Pneumonia by looking at their chest x-ray it will be faster than traditional approaches like a human identifying if the person has Pneumonia and we can apply this concept to other pulmonary disorders.

This Project Report is approved for recommendation to the Graduate Committee.

Project Advisor:

Ashis Biswas

_____

Name1, e.g., Jane Doe

MS Project Committee:

_____

Name2

_____

Name3

[This page should be included ONLY in theses that are copyrighted. If you include this page, change the page numbering for the following pages to vi instead of v using the *Insert/Page Numbers* command.]

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Problem

In today's world, Artificial Intelligence is all around us in many ways. It has countless applications. In this paper, we will discuss one of those applications which is to identify if a patient has Pneumonia by looking at their chest X-ray. This will be done with the help of CNN (Convolutional Neural Network). The other main part of this project is model interpretability to see why the model predicts if a person has pneumonia. The main reason why model interpretability is very important because it answers the question "How confident are we in our model and its predictions?". Only by answering these types of questions more and more people will understand the vast benefits of Artificial intelligence and trust this type of technology even more. For example, some people trust self-driving cars, and some don't by answering these types of questions more people will understand its benefits and start trusting it with confidence. The same goes for this approach some people will not trust a computer telling them if they have a pulmonary disease but with the right model interpretability, we can build that trust and confidence in people to appreciate the power of Artificial Intelligence.

1.2 Project Statement

Image-based Deep Learning and Model interpretability for Identification and Classification of Viral Pneumonia

1.3 Approach

The project was started by doing EDA on the dataset which included images of chest X-Rays. After understanding what the dataset looks like the next step was to preprocess the images for training and testing purposes. The step after that was to design a few Convolutional Neural Network to see which one performs the best. After choosing the best model the next step was to evaluate the model on precision and loss on the testing data. After achieving the best results model interpretability research was done as to why the model is predicting if a given person has Pneumonia or not.

1.4 Organization of this Project Report

Chapter 2 will focus on the key concepts to understand the project report such as Deep Learning, Convolutional Neural Networks, Model Interpretability, and will also include some related work. Section 3 will cover the architecture of the model. chapter 4 will include methodologies, evaluation metrics, and results of the project. Chapter 5 will conclude the report with a summary, contributions, and future work.

2. BACKGROUND

2.1 Key Concepts

Before we jump into the project the reader must understand some key concepts about the project. We will start by discussing Deep Learning in general and move onto Convolutional Neural Networks and then finish by understanding what model interpretability is.

2.1.1 Deep Learning

Deep Learning is a subfield of Machine learning it is inspired by the working of the human brain. Deep learning models use neural network architectures. The word deep in deep learning corresponds to the number of hidden layers in the architecture as shown in the figure below [1]. In the Figure below each circle corresponds to a neuron and each neuron is interconnected with all the other neurons. The first layer of the neural network is always the input layer, and the last layer is always the output layer. All the layers in the middle are known as hidden layers.

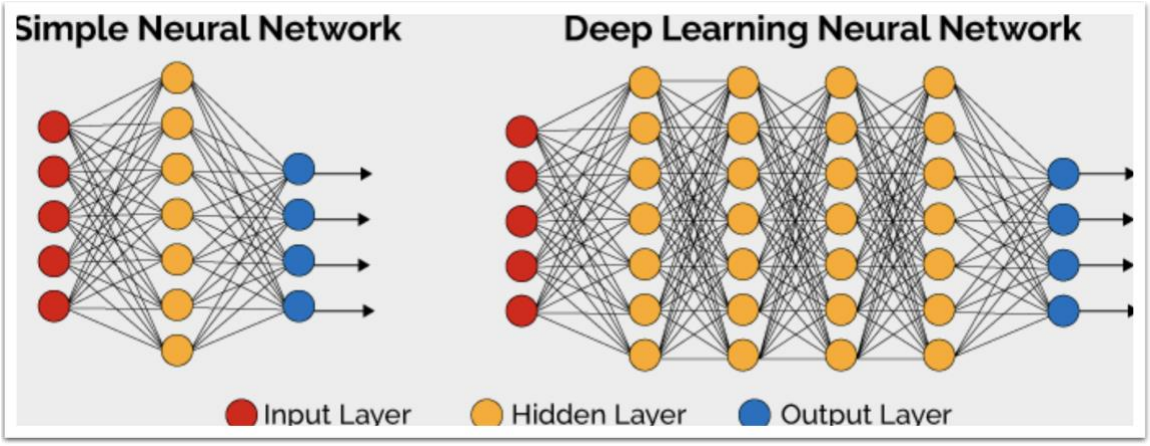

Figure 1: Simple Neural Network Deep Neural Network

2.1.2 Convolutional Neural Network (CNN)

Convolutional Neural Networks are designed to work with computer vision problems. In our case, we have a set of images to classify which are chest X-Rays. The hidden layers in a CNN can extract features from raw data in our case this raw data is the image data. When the image data is passed into the architecture top layers can extract features like cars, faces, and dogs, etc. While the lowers layers can extract edges or shapes in the images [2].

To understand the working of the neural network, we will need to understand three main components of this type of architecture. Which are as follows.

1. Convolutional layers
2. Pooling layers
3. Fully connected layers

2.1.2.1 Convolutional layers

The convolutional layer is usually the first layer after the input layer. The convolutional operation of the input images is computed using kernel filters to remove fundamental features in this layer. For example, if a have input images of size 256x256x3(width, height, depth) and the filter size of 5x5x3 = 75 pixels the filter will only extract those 75 pixels from the image instead of 256x256x3 = 196,608 pixels. The filter

moves over the entire input image step by step, calculating the dot product between the kernel filter weights and the input image value, resulting in an activation map. As a result, CNN will pick up a visual feature which is also known as a feature map. Which is also smaller in size compared to the input image. And the output of this operation is passed through a nonlinear activation function most used is called ReLU [3]. The figure below shows how the computation is done.

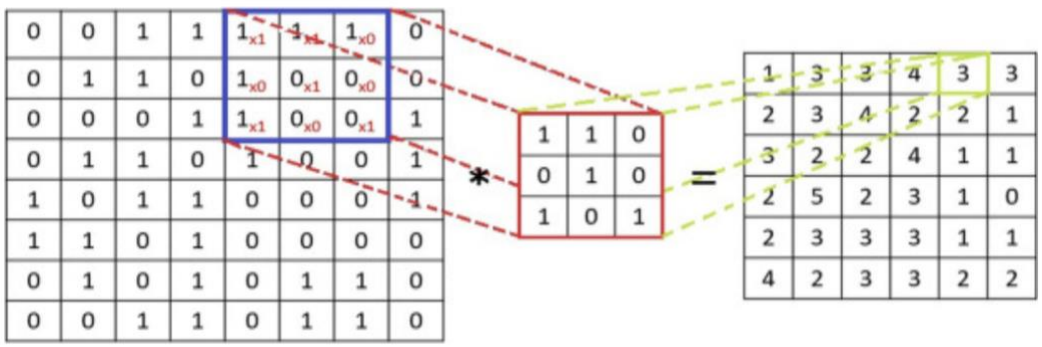

Figure 2: convolutional operation

2.1.2.2 Pooling Layers

There are multiple types of pooling layers, but Max pooling is the most used. The main reason why pooling layers are used is to reduce the size of the feature map generated by the convolutional layer. This is done by extracting a patch from the feature map and taking the highest value and discarding all other values. The most used filter in the max-pooling layer is 2x2. The figure below shows us how the feature map is reduced in size for the sake of computation resources [4].

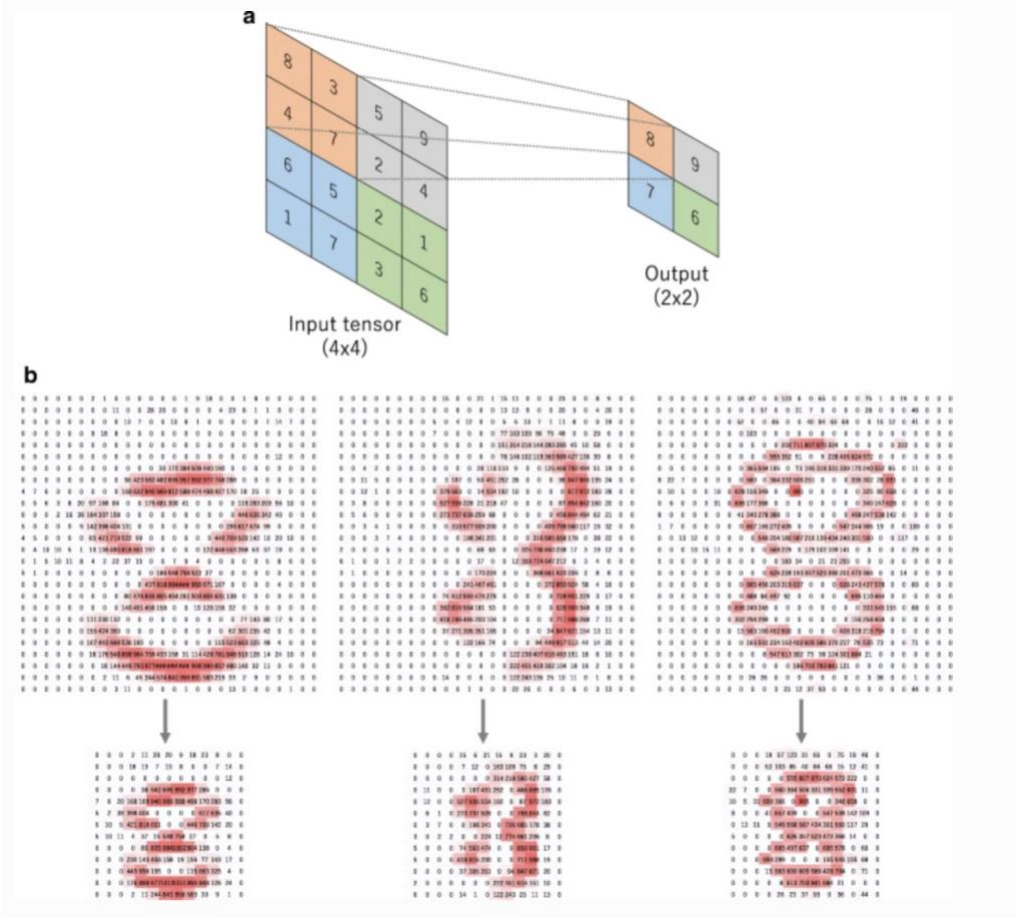

Figure 3: Max pool operation

2.1.2.3 Fully Connected Layers

Convolutional layers and pooling layers help us to find patterns and features in images, but they are not very useful to make the actual prediction. That is exactly why we need fully connected layers at the end of the model to make the actual prediction.

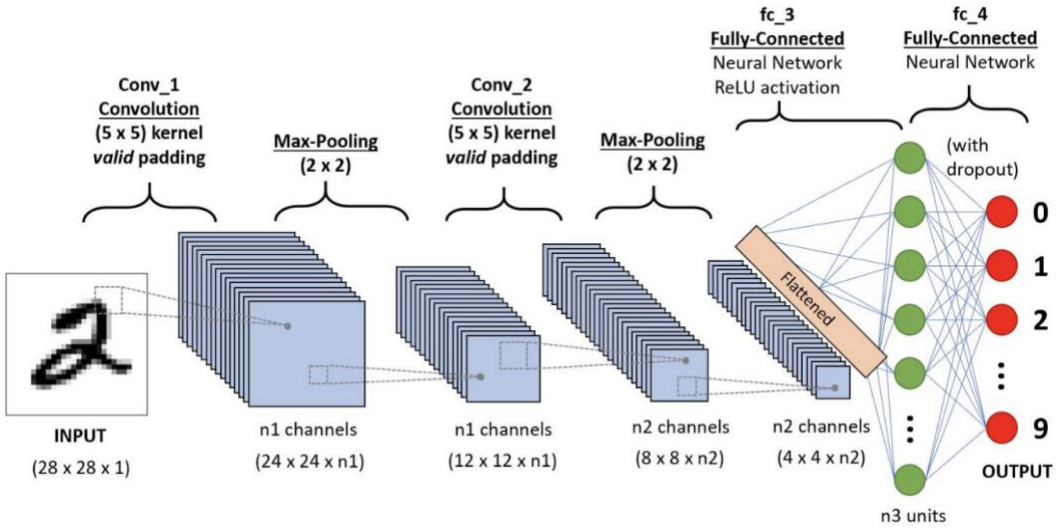

Figure 4: CNN Architecture

In the figure above we can see the whole architecture of a CNN model. If we pay attention to the last pooling layer the output of that layer is flattened and fed into the fully connected layers also known as Dense layers to make the actual prediction [5].

2.1.3 Model Interpretability

In recent years Artificial Intelligence has grown quite a lot with its applications in the medical industry, auto industry, online shopping, and financial industry, etc. The applications of Artificial Intelligence are countless. This technology has changed the world as we know it. Also, to keep in mind that there are more benefits than downsides of this technology. But a few years ago, deep learning models were like black boxes we didn't know what was happening inside them. That is why this new area of research is just getting more prominent to understand the model behavior to a greater extent.

There is no such set definition as to what model interpretability is. The one I like is by Miller “The degree to which an observer can understand the cause of a decision” [7]. There is no standard by which we can measure or conclude the model’s interpretability. But in the book “Interpretable Machine Learning” Molnar explains how we can do it. Molnar has talked about three ways how this can be achieved [7].

1. Application-level evaluation (real task)
2. Human-level evaluation (simple task)
3. Function level evaluation (proxy task)

These three levels of model interpretability are based on how severe the consequences are if the model’s prediction is wrong. For example, if I am working on a side project trying to guess people’s age by looking at their pictures even if I am wrong there are no bad consequences to this. But on the other hand, if I am trying to predict if a given person has pneumonia or not this can have terrible consequences for a wrong prediction by the model. We will discuss this a little more in the results sections of this paper.

2.2 Related Work

2.2.1 Image classification using deep learning

Image classification is something that is very widely used in the world of Artificial intelligence. It has also surpassed humans in many ways. There is a lot of ways how these problems can be solved. The most amount of research for image classification is done on Convolutional Neural Network and it is said to be one of the best image

classification deep learning algorithms [7]. There are a lot of flavors that CNN architecture comes in such as ResNet-50, Xception, Inception-v4, Inception-ResNets, and ResNeX-50, etc. Each new flavor of CNN architecture has surpassed the previous one on the Image-net dataset.

The most closely related research done on this dataset is called “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning”. In this research, they have used the same dataset and got an accuracy of 92.8%. The algorithm used in this study was the Inception V3 model [8]. We will discuss our results in the results section of this report

2.2.2 Model Interpretability

Model Interpretability is a very new area of research. This research asks the question as to why the model is predicting something. This is very important to understand as this looks past the old way of seeing how the model is performing based on just accuracy and some numbers. Human-understandable justifications for the model output is also very important as these models will be used in mission-critical processes such as medical diagnosis as discussed in the cited literature [9].

3. ARCHITECTURE

3.1 High-Level Design

The Deep Learning algorithm used was a Convolutional Neural Network that consisted of a total of 40 layers to construct. The first five blocks of the architecture had convolutional layers and max-pooling layers. The convolutional layers had a filter size of 3x3. The max-pooling layers had a filter size of 2x2. Then followed by the densely connected layers for getting the actual predictions. Please see chapter 2 for more information on different types of layers. The reason the model has a high number of layers is to detect features deep inside the images. A model with few layers didn't perform as well and models with too many layers had a long training time. Each block of layers also consisted of a dropout layer which eliminates some connections in between the neurons to avoid overfitting. Overfitting means when the model learns the training data too well and performs poorly on the testing data. On the other hand, underfitting is caused when the model fails to generalize the training data meaning the model fails to learn the important features in the training data and performs poorly on the testing data [10]. The figure below shows the whole Architecture of the Convolutional Neural Network.

Layer (type)	Output Shape	Param #
conv2d_30 (Conv2D)	(None, 256, 256, 16)	448
conv2d_31 (Conv2D)	(None, 256, 256, 16)	2320
batch_normalization_27 (Batc	(None, 256, 256, 16)	64
max_pooling2d_15 (MaxPooling	(None, 128, 128, 16)	0
dropout_27 (Dropout)	(None, 128, 128, 16)	0
conv2d_32 (Conv2D)	(None, 128, 128, 32)	4640
conv2d_33 (Conv2D)	(None, 128, 128, 32)	9248
batch_normalization_28 (Batc	(None, 128, 128, 32)	128
max_pooling2d_16 (MaxPooling	(None, 64, 64, 32)	0
dropout_28 (Dropout)	(None, 64, 64, 32)	0
conv2d_34 (Conv2D)	(None, 64, 64, 64)	18496
conv2d_35 (Conv2D)	(None, 64, 64, 64)	36928
batch_normalization_29 (Batc	(None, 64, 64, 64)	256
max_pooling2d_17 (MaxPooling	(None, 32, 32, 64)	0
dropout_29 (Dropout)	(None, 32, 32, 64)	0
conv2d_36 (Conv2D)	(None, 32, 32, 128)	73856
conv2d_37 (Conv2D)	(None, 32, 32, 128)	147584
batch_normalization_30 (Batc	(None, 32, 32, 128)	512
max_pooling2d_18 (MaxPooling	(None, 16, 16, 128)	0
dropout_30 (Dropout)	(None, 16, 16, 128)	0
conv2d_38 (Conv2D)	(None, 16, 16, 256)	295168
conv2d_39 (Conv2D)	(None, 16, 16, 256)	590080
batch_normalization_31 (Batc	(None, 16, 16, 256)	1024
max_pooling2d_19 (MaxPooling	(None, 8, 8, 256)	0
dropout_31 (Dropout)	(None, 8, 8, 256)	0
flatten_3 (Flatten)	(None, 16384)	0
dense_19 (Dense)	(None, 1024)	16778240
batch_normalization_32 (Batc	(None, 1024)	4096
dropout_32 (Dropout)	(None, 1024)	0
dense_20 (Dense)	(None, 512)	524800
batch_normalization_33 (Batc	(None, 512)	2048
dropout_33 (Dropout)	(None, 512)	0
dense_21 (Dense)	(None, 256)	131328
batch_normalization_34 (Batc	(None, 256)	1024
dropout_34 (Dropout)	(None, 256)	0
dense_22 (Dense)	(None, 64)	16448
batch_normalization_35 (Batc	(None, 64)	256
dropout_35 (Dropout)	(None, 64)	0
dense_23 (Dense)	(None, 1)	65

Total params: 18,639,057
Trainable params: 18,634,353
Non-trainable params: 4,704

Figure 5: CNN Architecture

3.2 Implementation

The implementation of the project is as follows. Step by step

1. EDA (Exploratory Data Analysis) to gain insights into the dataset.
    1.1. Plotting image sizes.
    1.2. Plotting class distributions.
    1.3. Plotting random image samples for visualization.
2. Preprocessing the dataset for the models.
    2.1. Loading the images.
    2.2. Resizing the images.
    2.3. Normalizing the image data.
    2.4. Generating the labels.
    2.5. Converting image data to NumPy arrays.
    2.6. Shuffling the data.
3. Trying Different CNN Models.
4. Choosing the best CNN model (This report will only discuss the best model)
    4.1. Saving the best model.
5. Making Predictions with the best model.
6. Evaluating the performance of the model.
    6.1. Plotting model Accuracy and Validation Accuracy per Epoch
    6.2. Plotting model loss and Validation loss per Epoch
    6.3. Changing predictions to binary. {1: PNEUMONIA, 2: NORMAL}

6.4. Calculating the Confusion Matrix.

6.5. Calculating Precision, Recall, F1 Score on testing data.

7. Visualizing the model (Model Interpretability)

7.1. Visualizing the model architecture.

7.2. Visualizing each layer of the model by passing in an image.

7.2.1. Generating Feature Maps of each layer

7.2.2. Generating Heat Maps for each layer.

8. Visualizing the predications (Model Interpretability)

8.1. Finding out what features the model is looking at to make the predictions.

8.2. Plotting the images with the most important features and weights of those features.

4. METHODOLOGY AND RESULTS

4.1 Methodology

In this project, we designed a Convolutional Neural Network for the purpose of image classification. After obtaining the results the performance metric used was Recall because accuracy might be a bit misleading because we need to predict a maximum number of positives cases. To better understand this, we can look at another example. If someone was assigned to filter spam email using machine learning the goal is to filter only spam email and not the other ones. It's ok to have a few spam emails in your account but not ok to have some important email in the spam folder. The same concept applies here too. We need our model to predict the maximum number of positive cases because if a person does have pneumonia and we predict he does not have pneumonia is far worse than predicting the person who has no pneumonia as having pneumonia. The figure below shows the report on the performance metric. Class 0 is for normal and class 1 is for pneumonia. The Recall score of our model is 98% which is the ability of our model to predict positive cases.

	precision	recall	f1-score	support
0	0.97	0.82	0.89	242
1	0.90	0.98	0.94	398
accuracy			0.92	640
macro avg	0.93	0.90	0.91	640
weighted avg	0.93	0.92	0.92	640

Figure 6: Classification Report

4.2 Results

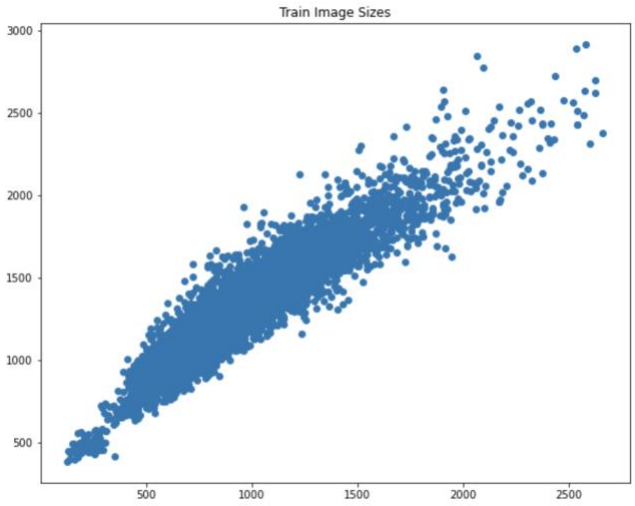

Figure 7: Train Image Sizes

Figure 8: Test Image Sizes

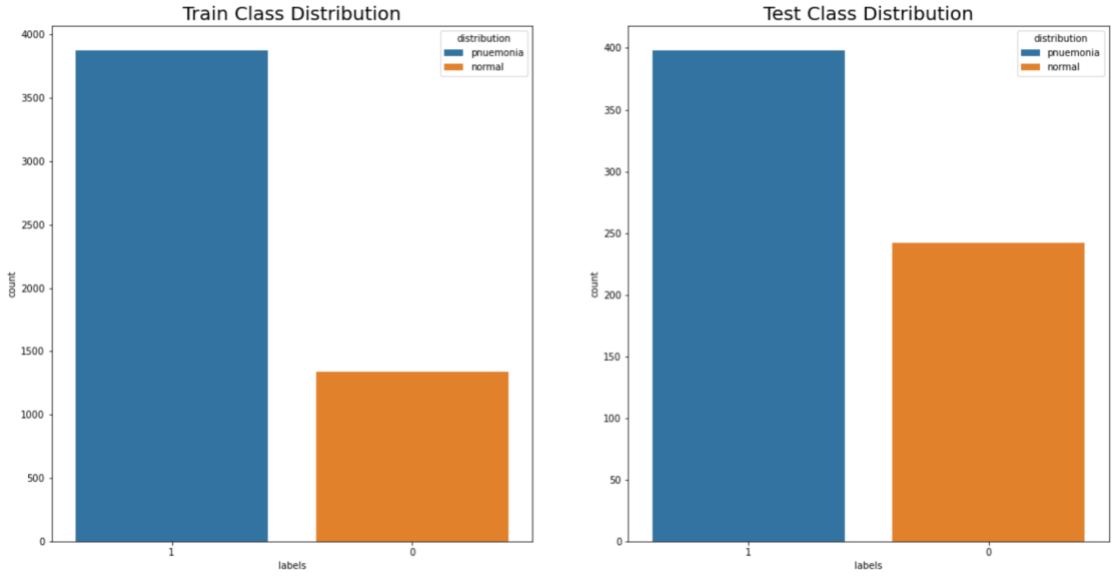

Figure 9: Class Distributions for Labels

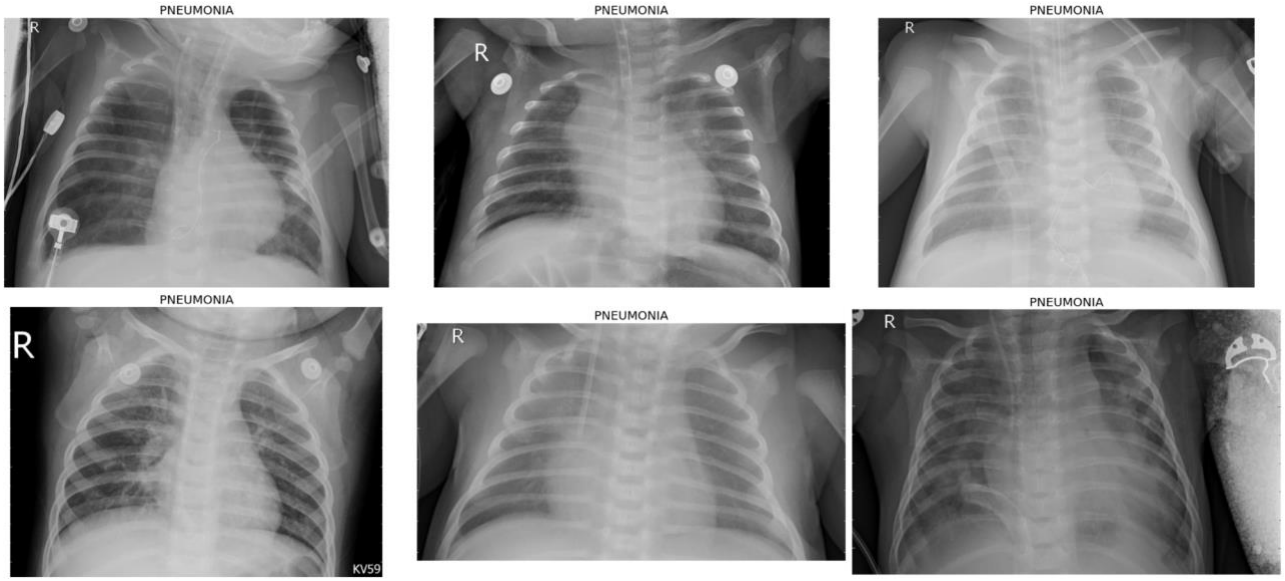

Figure 10: Samples Containing Pneumonia

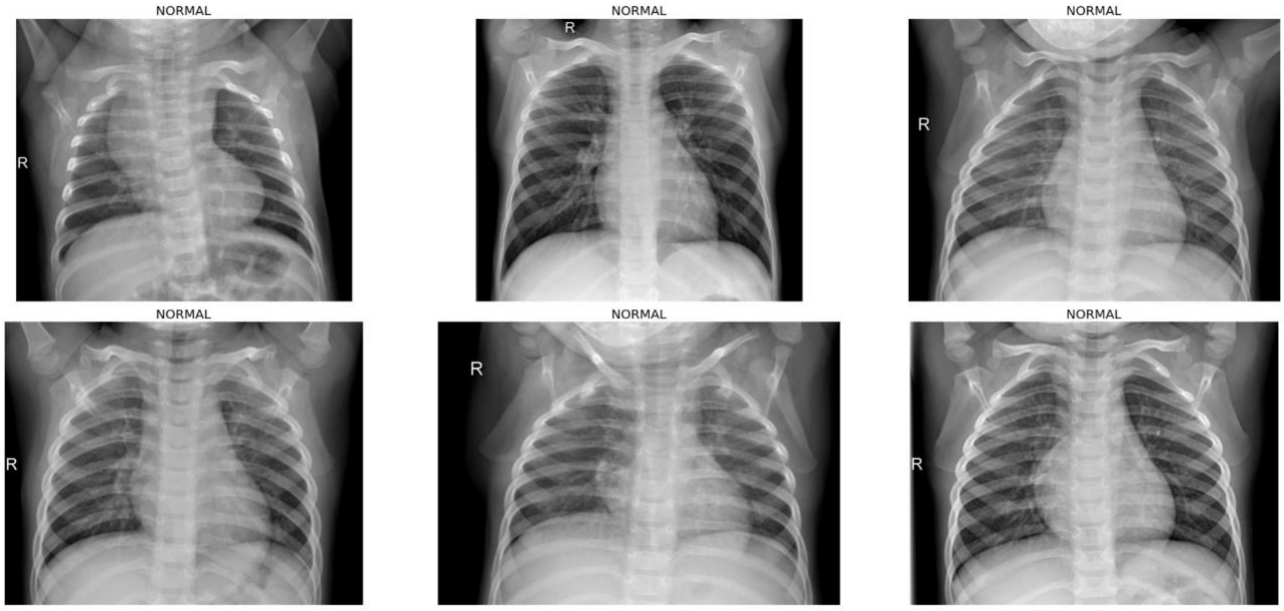

Figure 11: Normal Samples

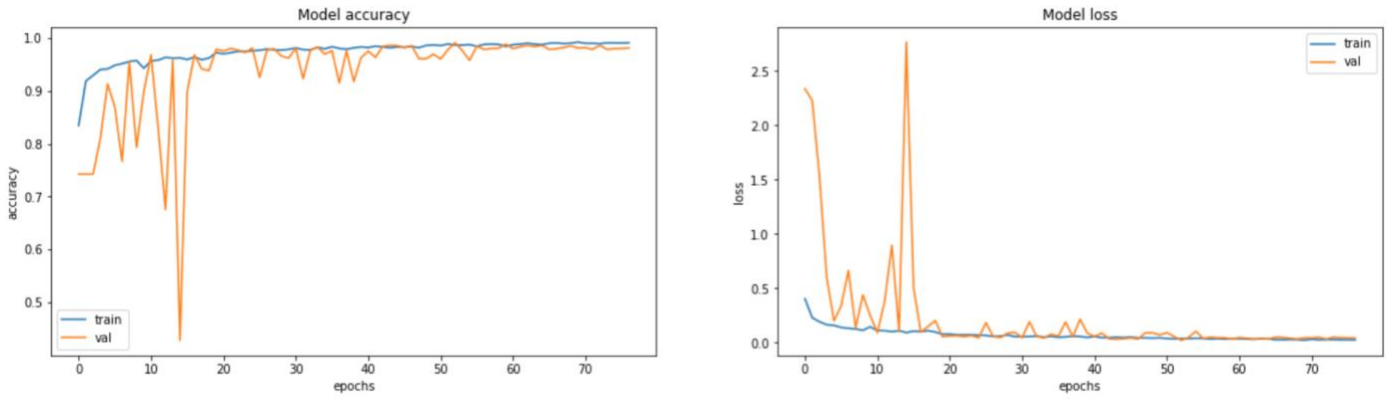

Figure 12: Model Performance

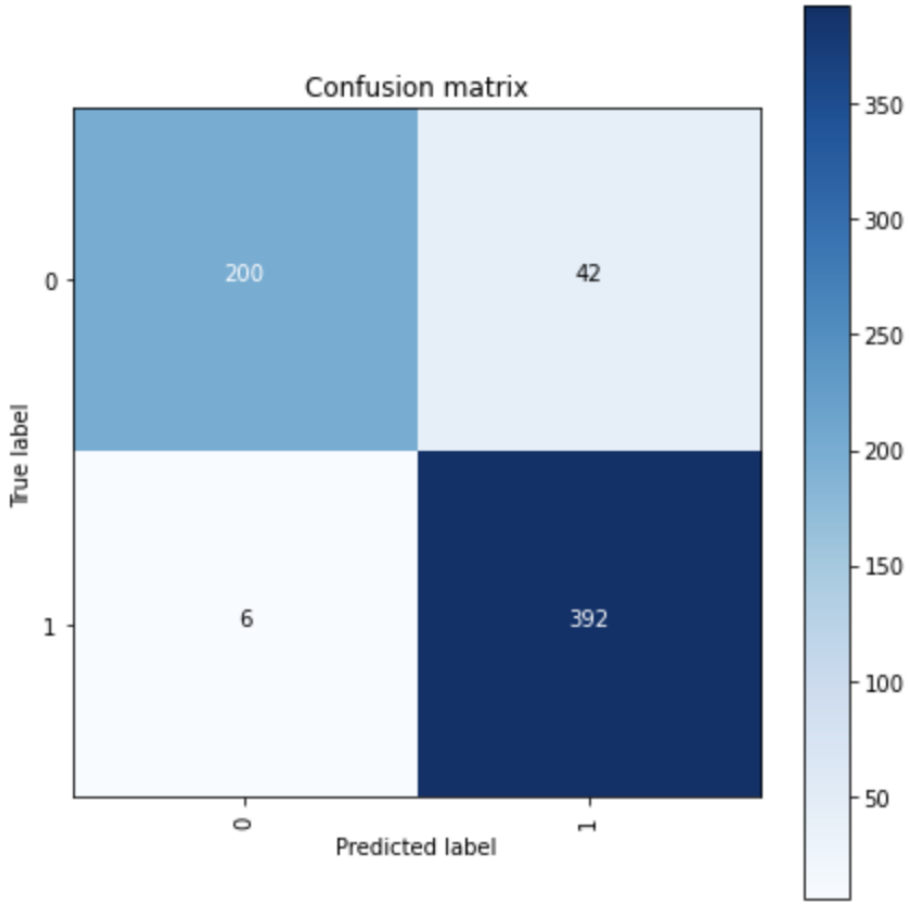

Figure 13: Confusion Matrix

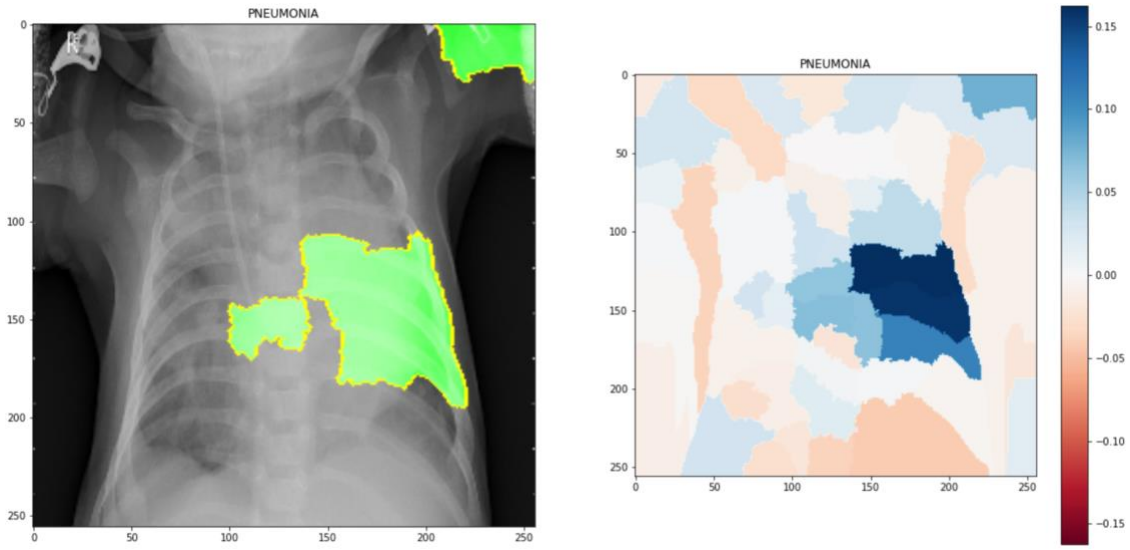

Figure 14: Model interpretability of person having pneumonia

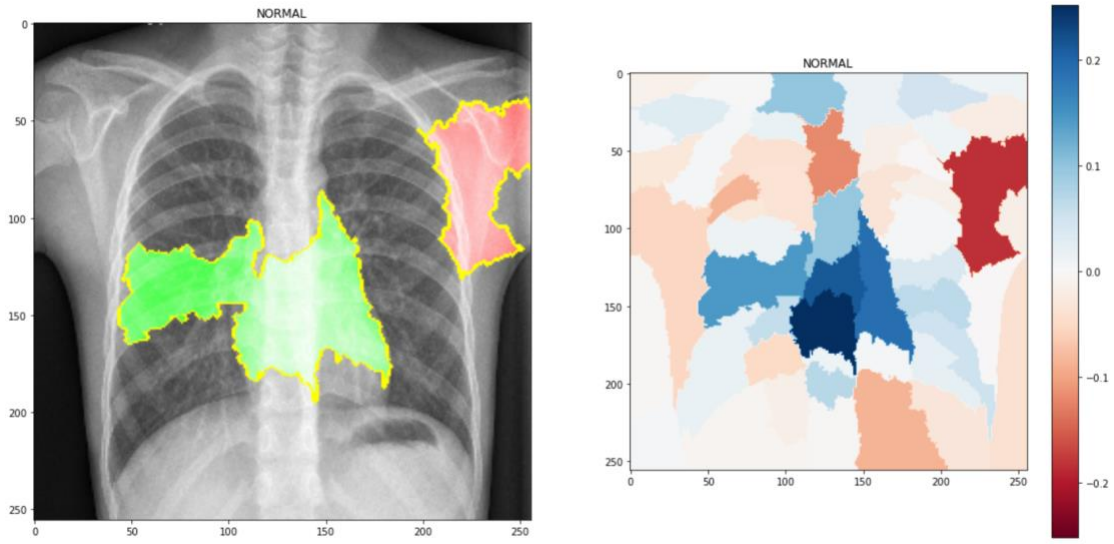

Figure 15: Model interpretability of person having no pneumonia

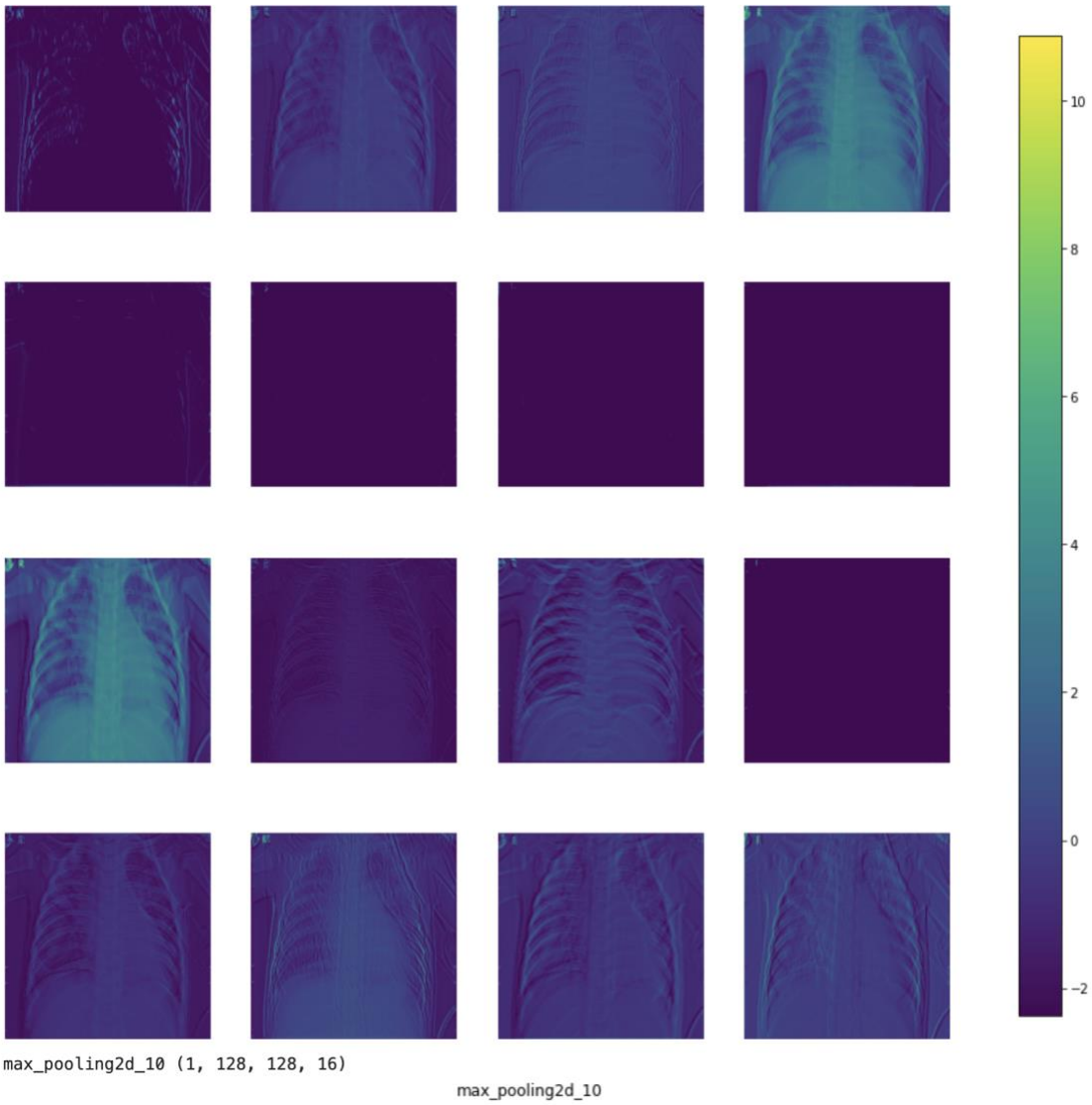

Figure 16: Feature map by a random layer

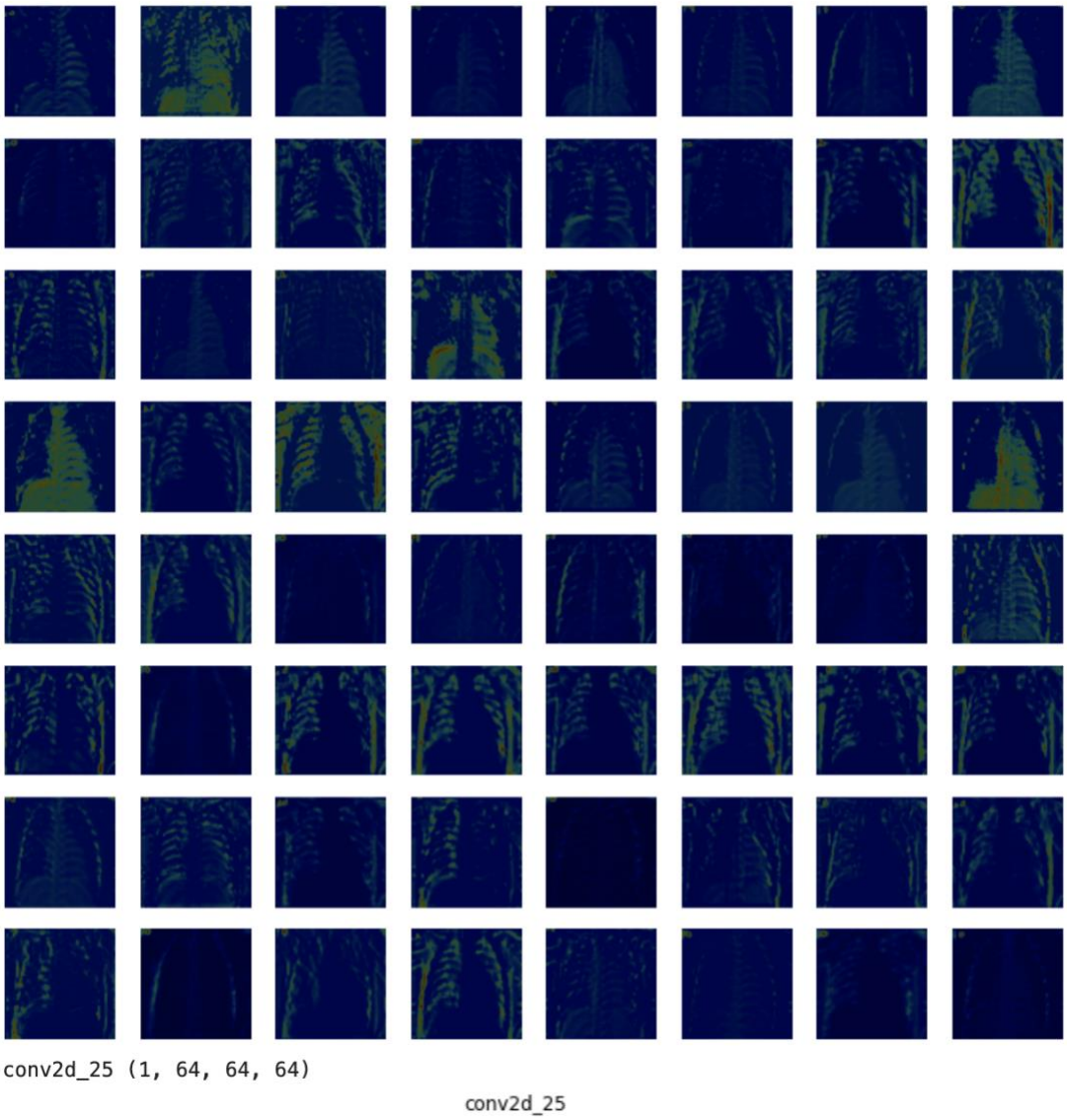

Figure 17: Heat map by a random layer

4.3 **Analysis**

We have already talked about the performance of the model in the methodology section of this report and talked about how Recall is the performance metric we used to evaluate the model's performance. Now, we will discuss model interpretability by looking at figures 14 and 15. By looking at those figures we can see what features the model is

looking at to make a prediction. Each figure has two plots the first plot highlights the area in green and red. Green meaning that it's a pro feature the model is looking at and red means it's a con feature the model is looking at. The next plot is the weights assigned to those features higher the weight the better is the feature for prediction. It does make sense that the model is looking at the lungs of the person in both cases if a person has pneumonia or not. Even if a radiologist is looking at an X-Ray for this purpose, he will be looking at the lung area as that where pneumonia could be present. And our model is doing the same thing which gives us more confidence in our model and its prediction.

5. CONCLUSIONS

5.1 Summary

In this project, we used image-based deep learning for the purpose of classification if a given person has pneumonia or not. We learned about some key concepts such as Deep Learning, Convolutional Neural Network, and Model interpretability. Then we looked at the high-level design of our CNN model and implementation of the project. We also discussed the performance metrics which are important for this project. The CNN model achieved a Recall score of 98% which is the ability of our model to predict positive cases. Then we looked at the different visualizations about the dataset and showed results through different figures. The results also contained a very important part of the report which is the model interpretability figures which tell us what features of the image the model is looking at to predict if a given person has pneumonia or not.

5.2 Contributions Potential Impact

The main contribution of this paper is the Model interpretability this new and evolving research has shown us a different way to elevate our models and this research will allow us to use Artificial Intelligence in more industries such as Medical diagnosis with more confidence and trust. People will also trust these types of technologies more when they see how powerful artificial intelligence is.

5.3 Future Work

The recall score can be improved by using the image classification model such as VGG-16 or Inception, but it was not possible here because training a network of that size on a laptop is not feasible and reliable. But with the evolution of cloud services, this can be achieved easily and further improved by hyperparameter tuning. The next thing will be model interpretability as this is a new research area there is still a long way to go. As there are no standards in this research as of now but with time, we will see more standards taking place. These standards will help us to interpret our models even better.

REFERENCES

[1] F. Q. Lauzon, "An introduction to deep learning," 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), 2012, pp. 1438-1439,.

[2]Li Deng and Dong Yu. 2014. Deep Learning: Methods and Applications. Found. Trends Signal Process. 7, 3–4 (June 2014), 197–387.

[3] Sinam, Ajit & Meitei, Takhellambam & Majumder, Swanirbhar. (2020). Short PCG classification based on deep learning.

[4] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018).

[5] S.H. Shabbeer Basha, Shiv Ram Dubey, Viswanath Pulabaigari, & Snehasis Mukherjee (2020). Impact of fully connected layers on the performance of convolutional neural networks for image classification. Neurocomputing, 378, 112-119.

[6] Authors are required! (2019). Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 267, 1-38.

[7] Rinat Mukhometzianov and Juan Carrillo (2018). CapsNet comparative performance evaluation for image classification. CoRR, abs/1805.11195

[8] Kermany DS, Goldbaum M, Cai W, et al. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. Cell. 2018;172(5):1122-1131.

[9] S. Chakraborty et al., "Interpretability of deep learning models: A survey of results," 2017 IEEE SmartWorld, 2017, pp. 1-6.

[10] H. Zhang, L. Zhang and Y. Jiang, "Overfitting and Underfitting Analysis for Deep Learning-Based End-to-end Communication Systems," *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2019, pp. 1-6.