UNIVERSITY OF NAIROBI
COLLEGE OF BIOLOGICAL AND PHYSICAL SCIENCES
SCHOOL OF COMPUTING AND INFORMATICS

Ground-Glass Opacities Identification Using Neural Networks for Monitoring COVID-19 Progression

BY:
ASTONE ODIWUOR OWINO
P52/31754/2019

Supervisor:
Dr. Miriti Evans A.K.

AUGUST 2021

A research project report submitted in partial fulfillment of the requirements of the degree of Masters of Science in Computational Intelligence of the University of Nairobi, Kenya

Declaration

I declare that this work is my original effort and has not been previously submitted for the degree by the University of Nairobi or any other University in Kenya and the world. To the best of my knowledge, the work used secondary and primary sources of data that have been accredited accordingly.

Signature: …………………………………

Date: ……………………………… 19-08-2021

Name: Astone Odiwuor Owino

Supervisor

This project report has been submitted in partial fulfillment of the requirements for the Master of Science in Computational Intelligence of the University of Nairobi with my approval as the Supervisor.

Signature: …………………………………

Date: ……………………………… 19-08-2021

Dr. Miriti Evans A.K.

Abstract

COVID-19 is a worldwide pandemic since the beginning of 2020. It records a high death rate across many countries in the world. Efforts have been put in place to control the spread and the associated deaths. Vaccination, isolation, mass testing, and artificial intelligence models have been used to control the disease. Due to the high number of cases per day, manual monitoring of progression has been difficult and associated with false negatives. Ground-Glass opacities (GGOs) identification has been used in the detection and classification of COVID-19 positive and negative cases. The localization of the GGOs and the volumes in the lungs can be used to identify and monitor the COVID-19 progression in the lungs. This research developed an adoptive computational model to help in COVID-19 monitoring. It identifies the localization of GGOs in the lungs responsible for COVID-19. The research also used automated feature extraction convolutional neural networks (CNN) models to enhance speed and accuracy. Feature extraction and modeling were done with standard CNN and CNN with transfer learning with augmentation models. CNN with the transfer learning model was chosen for the implementation because of the high accuracy of 97.36%. The model was used to identify GGOs given new examples to classify COVID-19 positive and negative cases accurately.

Keywords: *Ground-glass opacities, COVID-19, convolutional neural networks, progression*

Acknowledgments

I want to thank Dr. Miriti Evans A.K, my supervisor, for the immeasurable support guidance during the development of this project. I also thank the entire Faculty of Information Technology staff of the University of Nairobi for the support and the environment you created for the study; it was encouraging. To my wife Lydia and son Larry, thank you for your support during this period. To my classmates Godfrey, Paul, Elijah, Stephen, Sam, and Fridah, you were so supportive throughout the project. Above all, I thank God for the strength and ability to complete this project.

Dedication

I dedicate this work to all my family members; my wife, sons, mother, father, brothers, and sisters. Your physical and emotional support during the period of study was befitting. This effort was possible because of you.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

GGO: Ground Glass Opacities

CT: Computed Tomography

SARS-CoV-2: Acute respiratory syndrome coronavirus 2

CNN: Convolutional Neural Network

COVID-19: Coronavirus Disease 2019

CAD: Computer Aided Diagnosis

SVM: Support Vector Machine

CRISP-DM: Cross Industry Standard Process for Data Mining

CHAPTER ONE: INTRODUCTION

1.1. Background

Coronavirus disease 2019 (COVID-19) is caused by acute respiratory syndrome coronavirus 2 (SARS-CoV-2) emerged in December 2019 in China and has rapidly developed into a global outbreak. As at March 25th 2021, there were 125,454,721 cases, 2,757,158 deaths, and 101,312,279 recoveries across the world (worldometers.info, 2021). In Kenya alone, as at March 25th 2021, there were 124,707 cases, 2066 deaths, and 90770 recoveries (worldometers.info, 2021). COVID-19 is a global threat and is treated as an urgent threat. Countries across the world have placed efforts and resources in managing the virus through diagnosis, prognosis, and treatment. Nucleic acid detection is currently used as the standard procedure to confirm the virus in the lungs; the strategy is associated with a high false-negative rate. Kenya is fighting the spread of COVID-19 by mass testing, reinforcing compulsory wearing of facemasks and curfews within certain hours to reduce social interactions.

Computed Tomography (CT) images are used in the identification of Ground Glass Opacities (GGOs) that can help in the classification of COVID-19 cases. The Ground Glass Opacities are abnormalities in the lungs that are seen as lighter-colored or gray patches on CT scan, which shows that the lung is sick. Ground-Glass Opacities identification for detection and classification of COVID-19 has been achieved in different ways. Ouyang et al. (2020) used CT scan images to classify COVID-19 and non-COVID-19 cases. However, the research registered false negatives in most cases. Additionally, the study by Wang et al. (2020) used 3D images without feature reduction that faced complexity, low processing speed, and delayed results. Dong et al. (2020) also researched CT imaging in the detection of COVID-19. X-rays have been used in GGOs identification; the research lacked feature identification and could not monitor the progression (Makris, Kontopoulos & Tserpes, 2020).

1.2. Problem Statement

Ground-Glass Opacities come in different shapes, locations, sizes, and quantities and indicate various pathologies such as viral infections, fibrosis, cancers, and chronic lung diseases. Identification of COVID-19 GGOs is difficult due to the numerous pathologies. According to Caruso et al., (2020), the distribution of GGOs for COVID-19 patients is mainly in the lower lobes and periphery; they also appear bilaterally and multifocal, they have round shapes that seem unusual. The numerous characteristics and pathologies associated with the GGOs make it difficult to identify and classify COVID-19 cases accurately. Moreover, various features are considered, such as differences in distribution and their capacities in the lungs; this makes it quite challenging to differentiate COVID-19 cases from other pathogens.

On the other hand, computational problems need improvement to identify GGOs responsible for COVID-19 accurately. The current methods are highly complex and require more computational resources, especially those that consider 3D in identifying the features. It also becomes difficult to extract features that can be used to classify in the lungs that interfere with the results' accuracy. GGOs less than 15% are categorized as cancer, and those more than 15% is classified as mild COVID-19 case and the severity increase with the percentage (Ng et al., 2020). Therefore, it is problematic to accurately identify the percentages of the GGOs to categorize the progression of COVID-19 in the lungs through physical methods.

1.3. Objectives

1.3.1. Overall Objective

To develop a model that classifies Ground-Glass Opacities in the lungs from the CT images samples to monitor COVID-19 progression.

1.3.2. Specific Objectives

i. To extract specific feature characteristics of GGOs representing COVID-19 infection in the lungs.

ii. To classify GGOs representing COVID-19 in the lungs.

iii. To evaluate GGOs representing COVID-19 infection in the lungs for monitoring progression.

1.3.3. Research Questions

i. What are features that can be extracted and used to identify characteristics of GGOs representing COVID-19 in the lungs?

ii. What are the ways available to classify GGOs representing COVID-19 for monitoring progression?

iii. What are the most effective ways available to identify affected lungs by GGOs representing COVID-19 for monitoring progression?

1.4. Justification

Currently, the interpretation of GGOs in CT images is difficult and intensive, given various characteristics and possible pathologies. It is the work of the radiologists and healthcare practitioners to analyze the images for the possible pathologies physically. Since the progression of COVID-19 is rapid, it cost resources in terms of time to investigate several cases. Additionally, a considerable percentage of error may arise from human analysis, making it necessary to automate the entire process. On the other hand, the current models under tests experience complexities that interfere with processing speed, as most false-negative cases have been reported. A long time is taken to give room for severe cases before they could be treated, thus leading to more deaths. The research proposes a faster and less costly, adaptive, accurate, efficient, and learning-based model which can identify COVID-19 Ground-Glass Opacities in a CT scan images samples. The model automates the result analysis process and thus, limiting human expertise in the analysis. Most importantly, the model identified the GGOs specific stages to determine the severity of the case and propose monitoring approaches.

1.5. Scope

COVID-19 monitoring and progression were based on Ground-Glass Opacity characteristics. The images were obtained from CT scans from different COVID-19 and non-COVID-19 patients with respiratory complications admitted to various hospitals worldwide. The work only focused on those patients admitted in particular treatment options for COVID-19, cancer, pneumonia, and those experiencing COVID-19 like symptoms.

CHAPTER TWO: LITERATURE REVIEW

2.1. Introduction

This chapter covers methods that are currently employed in the identification of Ground-Glass Opacities to monitor COVID-19 progression. It further investigates the use of machine learning models in GGOs identification in COVID-19 progression monitoring. The model process for this research work is also provided in the chapter.

2.2. Overview of CAD systems for Lung Infections

The use of images have been applied in the diagnosis of most lung diseases. However, some of the specific diagnostic disorders are not restricted to pneumonia, lung cancer, and tuberculosis. According to Makris, Kontopoulos & Tserpes (2020) X-rays and CT scans have been used in the diagnosis process. At the same time, X-rays provide images that are flattened, CT scan results into a cross-sectional image that can represent the 3D model of the lung. The advancement in artificial intelligence (AI) such as in computer vision and imaging has made it possible to obtain accurate results in the diagnosis, identification, and progression monitoring for various infections. According to Chhikara et al. (2020), AI-based methods used to identify GGOs are of three types: those that can take X-rays, CT scans, and both as inputs. Chhikara et al. (2020) also mentioned that the medical experts base most of the existing methods on manual crafting features, while only a few techniques use deep learning to extract GGOs features automatically.

2.3. AI-Based CAD systems for Identification of GGOs

GGOs for COVID-19 are usually identified in 3D images from CT scans. According to an article by Shi et al. (2020), COVID-19 GGOs have specific characteristics that differentiate it with other infections. These characteristics can only be identified correctly using 3D images rather than 2D images from X-rays. On another study conducted by Makris, Kontopoulos & Tserpes (2020), X-rays are considered the fastest in imaging; however, it produces 2D images that cannot identify the volume of COVID-19 in the lungs. Over the past few months, researchers have developed AI-based CAD systems that can be used in the diagnose COVID-19. According to Kermany et al. (2018), most of the designs are based on agency. They are not focused on proposing principled machine learning methods that meet the level of recommended accuracy. Another study by Chhikara et al., (2020), stated that these models are based on a system that can work rather without taking an interest in the notable features that improved the accuracy. Thus, a gap in these studies needs future considerations.

Identifying and classifying GGOs for COVID-19 based on CT scan images is essential in diagnosing and monitoring the disease. Front-line clinicians have played a role in the segmentation and classification of the infection volumes in the patient lungs. According to Kermany et al. (2020), most cases that have involved clinicians in the classification of volumes has been faced with false-negative instances. The GGOs volumes of more than 50% may mean that the patient is in critical conditions and needs urgent help, the percentage can increase to more than 90% that render the situation worst. The segmentation shows lower percentages of accuracy due to lack of feature extraction and that the authors use 3D images that require high computational power.

2.4. Convolutional Neural Network

CAD has been used in the classification of COVID-19 patients. Butt, Gill, Chun & Babu (2020) used ResNet and CNN models for the classification of COVID-19 cases. In this study, different images for 110 COVID-19 patients, those with influenza, and healthy people were trained and tested. The classification accuracy was 86.7%. Similarly, Song et al., (2020) used details relation extraction neural network (DRE-Net) model knows as the DeepPneumonia to classify different CT images for various infections (Wang, 2020). The classification accuracy was 86%. Another study by Wang (2020) on classification with accuracy of 89.5%, the authors first reduced the complexity of the problem on identification by extracting the region of interest based on GGOs localization using the CT scan images, they modify the network and uses fully connected ones. Based on this study, there is an improvement in the accuracy, and thus, there is the need to extract specific features and reduce complexity for more accurate results.

Figure 1 represents the process model. The labelled 3D image are loaded and normalized. The resultant images are used to extract features and then combines in the training model. The images are split into training set and test sets at 80% and 20% respectively. The features are then transformed to achieve the trained model. Test images are used in the trained model for prediction and classification to obtain the GGOs status.

The conceptual architecture used was CNN model that consists of several layers, such as the input layer, convolutional layers, pooling layers, full-connected layers, and output layers. CNN can be trained end-to-end to allow for data processing and prediction and classification (Albawi, Mohammed & Al-Zawi, 2017).

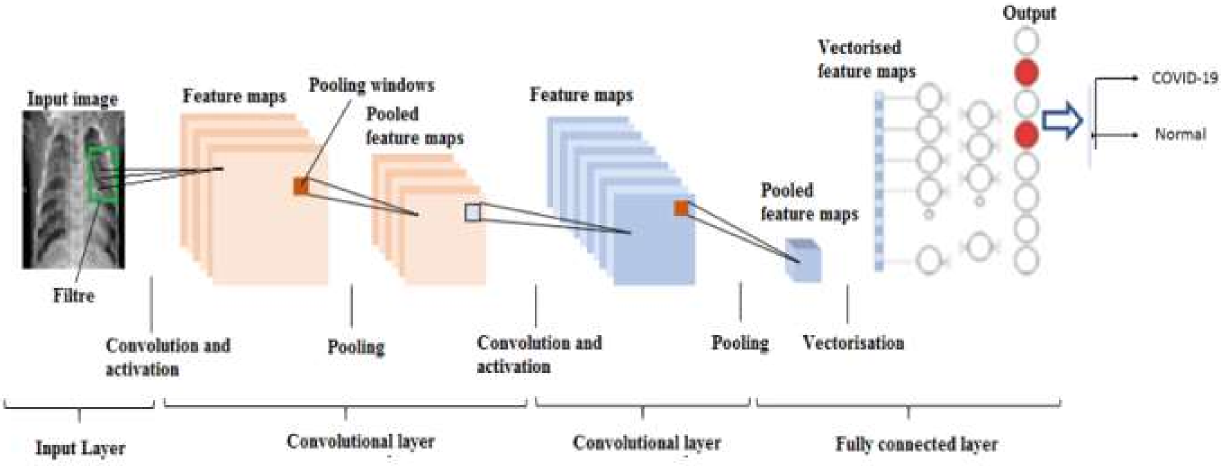


Figure 1: CNN Architecture. Source (Albawi, Mohammed & Al-Zawi, 2017)

2.4.1. Input Layer

The training images comprised of the chest CT scan images. The training images were loaded into X variables while target labels were loaded into Y matrix. The data was loaded in batches during training. The parameters were defined in the image dimension (244*244).

2.4.2. Convolution Layers

Convolution layers is considered as a linear operation that multiplies weights with the input. It is designed for a 2D input and the multiplication performed in the 2D arrays of weights and array of input data (Albawi, Mohammed & Al-Zawi, 2017. In the CNN architecture proposed, we have 3 layers that has 3*3 weights or filters and zero paddings.

2.4.3. Pooling Layers

The pooling layer is responsible for down sampling features by reducing the presence of the feature map. There are two types of pooling methods in the architecture; average pooling and max pooling (Albawi, Mohammed & Al-Zawi, 2017.

2.4.4. ReLU Layer

The convolutional layer used 4 ReLU layers. These are activation layers

2.4.5. Fully Connected Layers

These layers treat input data as a simple vector and output data as a single vector. These layers are connected during the training.

2.5. Segmentation of COVID-GGOs

Ground-Glass Opacities have played a critical role in identifying the level of severity in COVID-19 patients. A few studies have been done to automate segmentation COVID-19 GGOs based on various characteristics as representation. Chen, Yao & Zhang (2020) incorporated U-

Net to segment COVID-GGOs from other diseases. Similarly, Saeedizadeh et al. (2020) used U-Net to measure opacity percentages in the lungs to assess the condition. Fan et al. (2020) also used UNet++ in the detection and segmentation of GGOs. The study could segment multiple features in the lungs as well as the infection regions from the images gotten from COVID-19 patients. Besides, the study incorporated human-in-the loop strategy for efficient annotation. The problem with this CAD automation is that does segmentation without consideration on noise in their annotations.

Segmentation of different parts of the lungs based on GGO characteristics can provide critical information to monitor the progression of the disease in the lungs. Various algorithms have been used in the segmentation process that gives a certain level of accuracy. Hassanien et al. (2020), uses a support vector machine (SVM) classifier to detect GGOs from lungs CT scan images. Similarly, Shen, Bui, Cong & Hsu (2015) employed an automated lung segmentation that uses bidirectional chain codes to improve performance. However, the limitation of these studies is that they used a similar visual appearance of background and nodules makes it challenging to extract the GGOs regions.

Unsupervised learning has been used in the segmentation of GGOs in the lungs. A study conducted by Hassanien et al. (2020), used unsupervised anomaly detection to segment the anomaly regions in the lungs. The weakness of this study is that there was no feature extraction based on the images and that the system could not be able to identify if the GGOs are for COVID-19 or not. Besides, unsupervised learning could not identify the volumes of the GGOs, and thus, the study had limitations that could not correctly classify GGOs for COVID-19. Therefore, there is a need to propose a model that is fully supervised with the capability to identify GGOs responsible for COVID-19 correctly.

Artificial Intelligence technologies have been used in several applications in the war against COVID-19 pandemic. According to Wang et al. (2020), these applications can be categorized as patient scale, imaging and diagnosis, and protein structure prediction. The interest is in imaging in AI against the pandemic. Also, Wang et al. (2020) mentioned that CT images had been used in the identification of GGOs. The authors proposed a modified inception neural networks to classify GGOs responsible for COVID-19. The authors trained the network only on the region of interest identified by the radiologists as opposed to training the complete image. However, the study could not be able to monitor progression.

2.6. Research Gaps

Most of the studies identified have some flaws that may make them less effective. Most of the existing methods are based on manual crafting features that make the extract GGOs features difficult. COVID-19 GGOs has its features in the lungs differentiated from other infections, and these features need to be identified and extracted for the automation process. These characteristics can only be identified using 3D images rather than 2D images from X-rays. There is a need to extract more features and reduce complexity for more accurate results. On the other hand, the use of 2D images has limited the monitoring of the progression of the GGOs features which can only be observed in 3D images from CT scans, all the studies that use 2D images cannot monitor GGOs for COVID-19. Moreover, the use of 3D images at once, as implemented by most reviews is faced by complexity and low computation power that affects the reliability of the model.

Also, front-line clinicians have been used to segmentation and classify the infection volumes in the patient lungs from the previous study. Due to large volumes of current COVID-19 cases, there is a need to automate the whole process without human intervention that may compromise the observation quality. The human intervention should only be needed when identifying features of the GGOs responsible for COVID-19, and the effort comes from the clinicians and radiologists. Human interventions are always associated with errors and slow processing speed and thus, may be less effective. For example, the GGOs volumes of more than 50% show that a patient may be in the worst condition, yet the clinicians may not be able to monitor the progression.

Moreover, previous studies showed that segmentation results to lower accuracy since there is less feature extraction. Most of the current automation systems use 2D images that may not identify GGOs features. While 3D showed more features, the studies that use such models experiences low processing speed. Besides, unsupervised learning is incapable of identifying the GGOs, and thus, such studies may not help in monitoring progressions GGOs for COVID-19.

2.7. Proposed Model

The model proposed is Convolutional Neural Network (CNN). CNN is a type of machine learning methodology that has been used in the identification and monitoring of GGOs for COVID-19 progression. Neural Network with several layers stuck on each other with many hidden layers. The approach learns the input features automatically and decides on how well the model is trained. According to Hinton, Vinyals & Dean (2015), deep learning has helped in improving the accuracy of object recognition and detection based on the minor features that cannot be seen physically. CNN helps to solve small-object problems with minimal complexity. The model was able to classify COVID-19 GGOs from 3D CT scan images and differentiate them from other GGOs for various pathologies based on the features and characteristics for target GGOs. The CNN was able to learn at a low-level representative features over a large space. The CNN deep learning model was able to extract small features of the GGOs identify them based on volumes, and match them with other images to enhance the identification and monitoring process.

Deep learning is a function of Artificial intelligence, a subset of machine learning that has network capability of learning without being supervised from unstructured and unlabeled data. Deep learning mimics the working of the human brain in processing data inform of objects, recognizing speech, and translating language to make decisions. The CNN is a class of artificial neural networks that that can be used in computer vision tasks. The concept attracts interest across various domain such as radiology. CNN is significant in the study because it learns and adapt automatically. CNN represents an adaptive method for image processing and links between the general feed-forward neural networks and adaptive filters. CNNs have been a breakthrough in image recognition since they have been used to analyze visual imagery and in image classification.

2.8. Process Model

Based on the literature, the proposed model follows the process in figure 2. The images are obtained from the lung CT scan collected from different COVID and none-COVID patients. The images are then used to extract desirable features that represent COVID-19 GGOs. Based on the features, a model is trained to be able to identify if the image is having GGOs for COVID or not. The process iterated for new instances of images to make the classification.

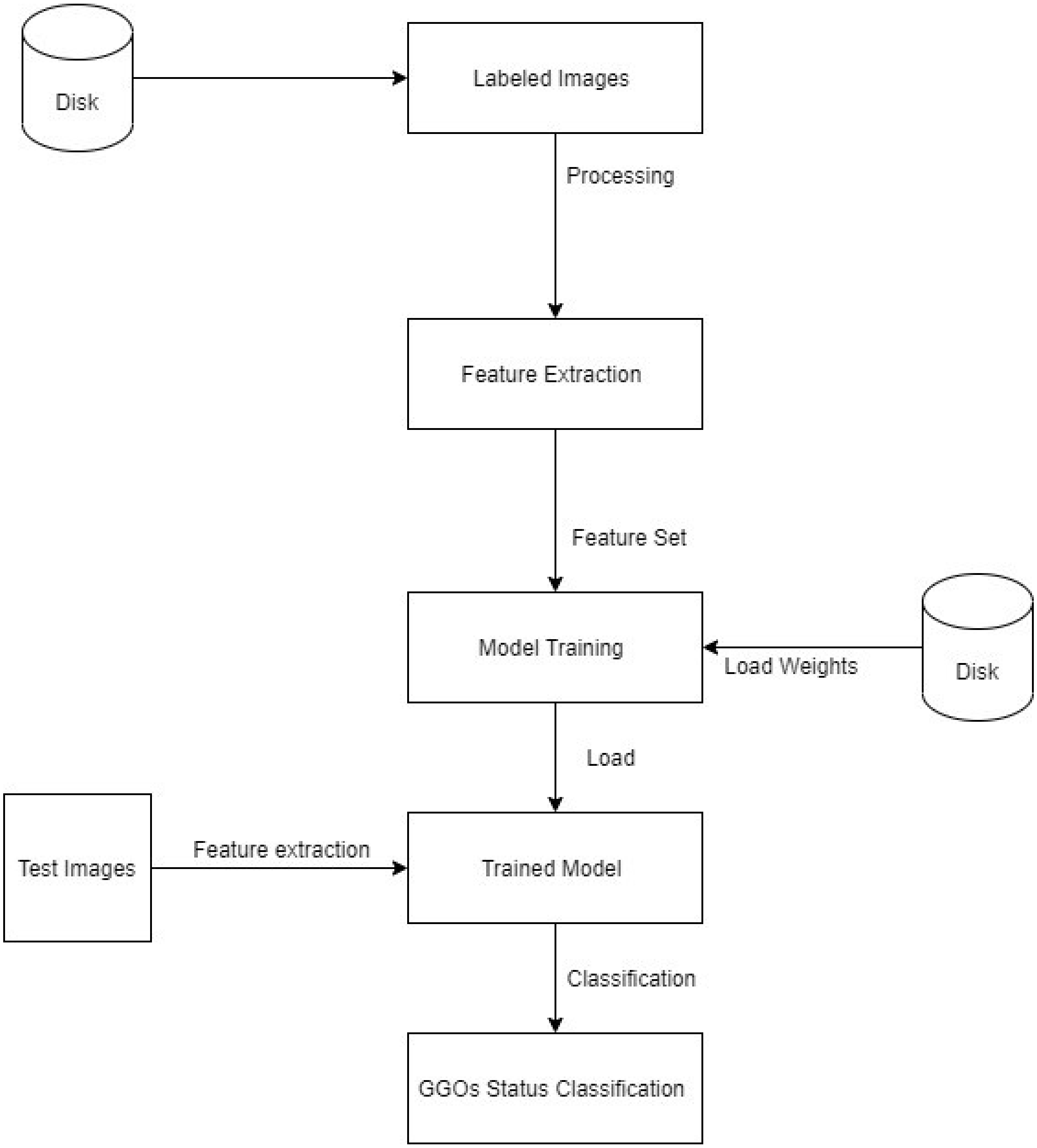


Figure 2: Process Model

CHAPTER THREE: METHODOLOGY

3.1. Introduction

The section described the research design used to achieve the project objectives. It also describes data collection procedures and data analysis approaches that were used to meet the goals. It includes research quality and validation.

3.2. Methodology Overview

The methodology used in the project is the Cross-Industry Standard Process for Data Mining (CRISP-DM). The methodology details a six phase model that describes the data science life cycle. CRISP-DM help in planning, organizing, and implementing machine learning projects.

The first phase of the methodology is business understanding. This phase focuses on understanding the project's objectives based on the requirement. At this stage, the project's objectives are determined, and the business success criteria are defined. Also, the feasibility study is done to determine the available resources in terms of data availability, risks, and contingencies and to conduct a cost-benefit analysis (Wang, 2011). Data mining goals are also defined as well as the selection of technologies and tools for each phase.

The second phase is data understanding, which drives the focus to identify, collect, and analysis of the data sets that helps in accomplishing the project objectives. The data needs to be collected and loaded in the analysis tool. The description of the data is also added, including the properties among format, number of records, and number of fields. The relationship within data is established at this phase (Shafique & Qaiser, 2014). Lastly, data is verified for quality issues

The third phase is data preparation, where the data is prepared for final analysis. The first step of data preparation is data collection of the intended data set that answers the research questions. The data is then cleaned through correction, impute, and remove unnecessary data. Next, the data is constructed by deriving new attributes and characteristics that helped in the analysis. Data is integrated where data is obtained from multiple sources and combined in one repository for use (Ayele, 2020). The data is then reformatted, ready to be passed to the intended model.

The fourth phase is modeling, where the selection of modeling techniques such as the algorithm used to analyze the data is selected. The data is split into training and test set and validation test. Building the model is also completed at this phase. The model is then assessed and interpreted based on the previously set procedures.

The fifth phase is the evaluation of the model that follows model design. The model is then evaluated if it meets the business success criteria set at phase 1 of the methodology. The review process is conducted at this stage, where the system is evaluated if it can execute properly (Martínez-Plumed et al., 2019). Once everything is confirmed to work properly, the process proceeds to deployment, if there are issues, the process iterates.

The last phase is deployment in which a document is developed on the functionality of the product. A monitoring and maintenance plan is developed to minimize issues during operation phase. Finally, the final product is released and the whole project is reviewed. The overview of the methodology is presented in figure 3.

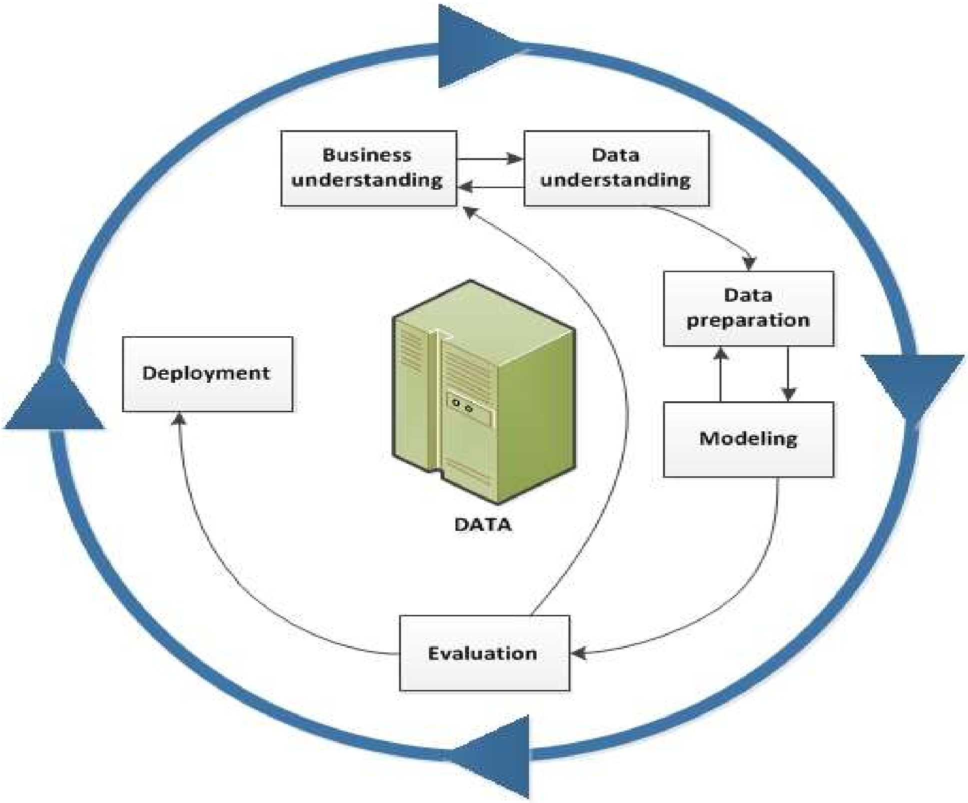


Figure 3: CRISP-DM Methodology. Source: (Huber et al., 2019)

3.2.1. Business Understanding

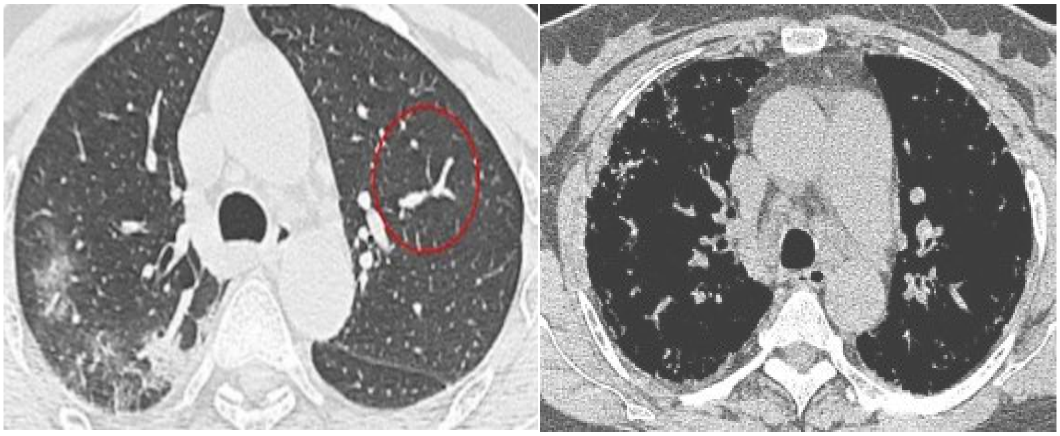
The understanding of the project need involved consultation with the medical experts, specifically radiologists, about COVID-19. Contact to a radiologist at Kenyatta National Hospital radiologist showed that lung CT scan images can be used in the segmentation of COVID and NoCOVID cases, however, the problem was in the extraction of the GGOs that were done manually. Together with the online research, the understanding led to GGOs segmentation in the lungs for COVID and NoCOVID cases. An automated data exploration strategy was used in the study. The exploration was done in Google Colab in which both the positive and negative cases of COVID-19 were loaded in CSV file and the trend visualized. A visualization of images was also done to see what type of images are positive GGOs for COVID and those that are negative.

3.2.2. Data Understanding

The dataset contains chest CT scan images from different patients with COVID-19 disease, pneumonia, and those with no chest infections. The dataset was obtained from a GitHub repository. The repository contains X-ray and CT images of patients that were taken during the treatment for COVID-19 patients. This data is publicly available for researchers who want to study the classification, detection, and other aspects relating to COVID-19. 757 images were collected for the study. The dataset consisted of 349 images of positive COVID-19 patients obtained from different hospitals and 408 images of negative COVID-19 patients. The images were obtained from other patients of different ages and gender.

3.2.3. Data Preparation

Out of the total images collected, 80% were used for training algorithms while 20% was used to test the model. The sample positive and negative images are shown in figure 4.



Positive GGO

Negative GGO

Figure 4: Positive and negative images for GGOs

The images were converted to grayscale with a single channel. A python function was used to convert the images to grayscale is provided as shown in code listing 1 in the appendix. A sample of the grayscale image obtained is shown in figure 5.

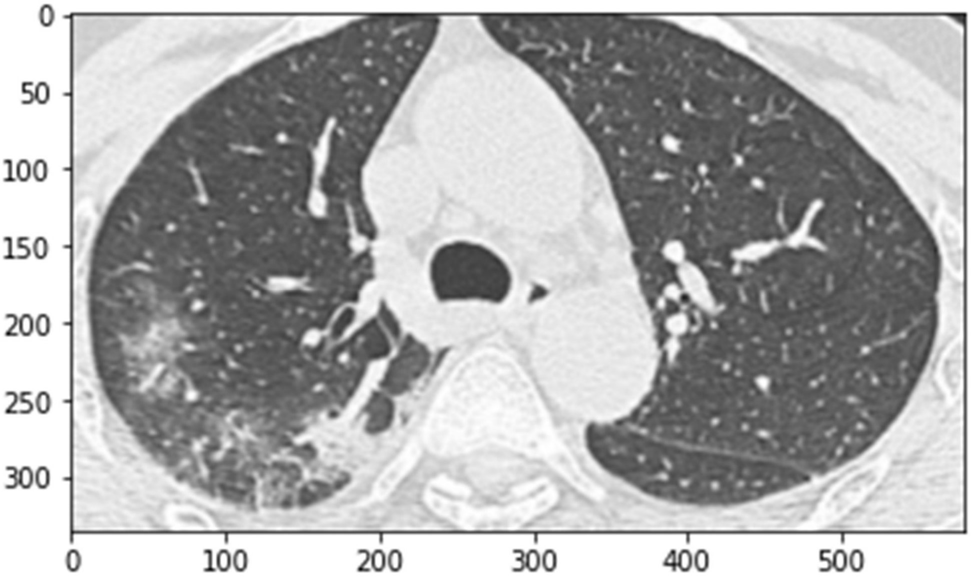


Figure 5: sample of grayscale image

After converting the image to grayscale, the process was followed by thresholding. In image processing, thresholding creates a black and white image out of greyscale image involving setting exactly the pixels of the images to white whose values are above given threshold while setting the other pixels black (Albawi, Mohammed & Al-Zawi, 2017). In this case, binary thresholding was used in which the pixels were converted to either black or white. The pixel intensity values that were less than 240 to 0 and 255 otherwise were converted. Thresholding allowed the precious location of the GGOs in the lungs. The code that implements thresholding is shown in code listing 2 in the appendix.

Thresholding is done using the same function and x,y coordinates. The white pixels, 255 are located and mapped to the corresponding grayscale image. Thresholding is followed by cropping out of positive 40 by 40 and a negative 40 by 40. The negative 40 by 40 is randomly generated in x,y coordinates and cropping it out. The samples are stacked on separate vertical matrices. Cropping is repeated for all images. The positive and negative resultant images have 3 dimensions; batch size, length, and width. While length and width are sample image sizes based on pixels, depth is stack images in the data structure. Cropping based on positive and negative samples and stacking can be generated based on the algorithm in code listing 3 in the appendix.

The corresponding label and targets were also generated with label 1 assigned positive set and 0 assigned negative. The labeling was done to retain the meaning of COVID-19 and negative COVID-19 from the images. After labeling the positive and negative labels, they were concatenated and scaled by dividing the feature matrix by value 255. The reason for dividing the pixel is to standardize the value. In addition, since the values are squashed between 0 and 1, the approach standardizes the images. Finally, the images were scaled to reduce noise. The resultant data was then shuffled, split into 80:20 training and testing, respectively. The code for the splitting approach is given in code listing 4 in the appendix.

The available data was augmented to increase the number of available samples due to the lack of larger sample size images. In this case, data augmentation with preprocessing technique was performed on the available data. The transformation that was employed included a random rotation of the image at a maximum of 15 degrees.

3.2.4. Modeling

The modeling was done using Convolutional Neural Network (ConvNet/CNN). The ConvNet requires low preprocessing as compared to other classification models. ConvNet is a variant of the CNN and was created using Keras API. In the ConvNet model created, the first convolutional layer, a 40 by 40 input image was fed in the network with a kernel of size 5*5. It was then followed with a 2*2 max-pooling layer. In the next convolutional layer, rectified linear units (RELU's) activated a kernel of 5*5. The two fully connected layers then succeeded, the first layer was activated by RELU and the second layer activated by softmax functions. The softmax function generated a probability distribution. The model was trained for 200 epochs. The code used for modeling is provided in listing 5 in appendix.

The second modeling was done using transfer learning with augmentation. Transfer learning adapts the concept that learning a new task relies on the previously learned tasks. In this concept, learning process is faster, more accurate, and needs less training data. Our transfer learning approach is based on the previously trained CNN. The general outline for transfer learning used is: the pre-trained CNN model was loaded, freeze weights in the lower convolutional layer. We then added a custom classifier with several layers of trained parameters. Lastly, the hyperparameters were unfreeze. We used MobileNetV2 architecture for the transfer learning. The code to implement the algorithm is provided in code listing 6 in the appendix.

Scrum was used as a system development methodology. The choice was based on the ability to iterate and the incremental possibility to allow future updates in the model. The approach facilitates constant communication and improvement (Srivastava, Bhardwaj & Saraswat, 2017). In addition, scrum approach made it easier to build a prototype within a static development plan. The scrum process is shown in figure 6.

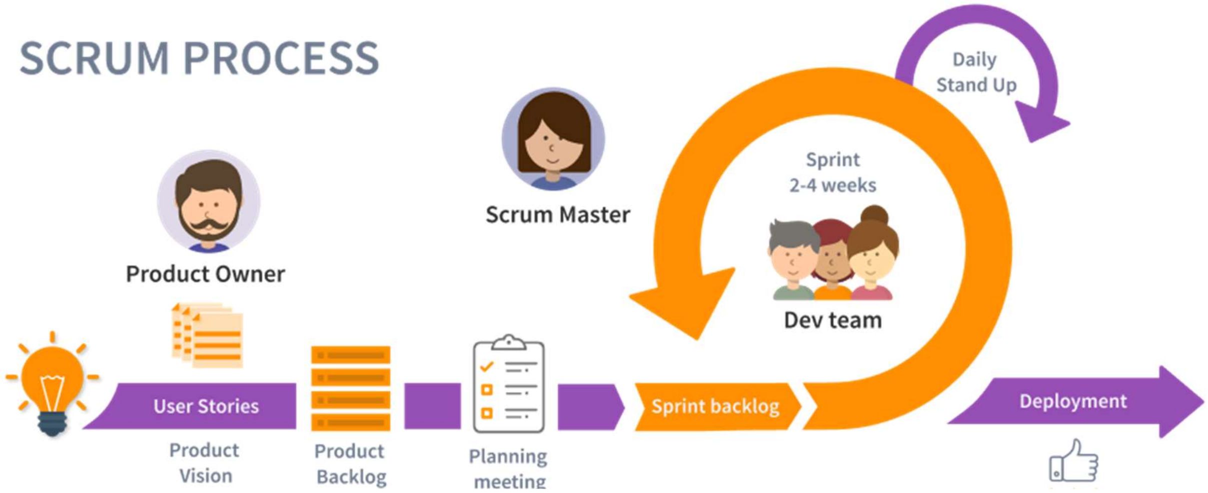


Figure 6: Scrum development methodology. Source (Srivastava, Bhardwaj & Saraswat, 2017)

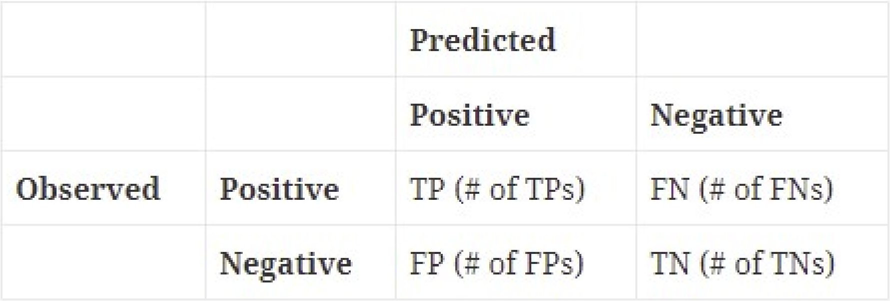
3.2.5. Evaluation

The performance metrics of the classification model was based on accuracy. The function created accepts the input and computes the metrics and gives results, whether the image COVID or NoCOVID. The code listing that implements the function is provided in code listing 7 in appendix.

The project's reliability and quality was done by developing more than one model and comparing them. The first model was a standard CNN while the second model was CNN with augmentation and transfer learning from the standard model. The approach was resourceful in reducing the complexity in the model. The CNN with transfer learning model was benchmarked against the standard CNN model. A confusion matrix was used to derive measures that were used to evaluate the system. The first measure derived from the confusion matrix is accuracy. It is calculated based on the number of correct predictions divided by the total number of the dataset (Visa et al., 2011). This was the main measure that was used in the evaluation of the model. Also, precision measure was used in the assessment. Precision is calculated by dividing the number of optimistic predictions with the positive values. Other considerable measures are specificity and recall. The equations are:

A confusion matrix is show in table 2.

Table 1: Confusion Matrix



		Predicted	
		Positive	Negative
Observed	Positive	TP (# of TPs)	FN (# of FNs)
	Negative	FP (# of FPs)	TN (# of TNs)

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Where; TP - True Positive, FN - False Negative, TN - True Negative, FP - False Positive

3.2.6. Deployment

A deployment document is developed on the functionality of the product. Monitoring and maintenance plan is developed to minimize issues during operation phase. A model was developed and deployment was done on cloud server as functional.

The experiments were done in Google cloud platform. The Google Colaboratory cloud platform was used because it provides free GPU that can help manage high computing resources needed in image processing. Some of the benefits of using this cloud platform are zero configuration required, free access to GPUs, and easy sharing. Python was used as a development language because it is rich in machine learning vision and imaging libraries such as Scikit-learn, Keras, Numpy, and Tensorflow, which were used in the analysis. Table 1 summarizes hardware and software specifications.

Table 2: Hardware and Software Requirements

Platform	Application	Specification
Software (Python 3.8)	Keras Tensorflow Numpy OpenCV	2.4.0 2.2 1.20.0 4.5.2
Hardware Google Cloud	RAM CPU	8GB 7

CHAPTER FOUR: RESULTS AND DISCUSSION

4.1. Introduction

The chapter covers results and discussion based on the analysis. The analysis was done for the standard CNN and CNN with transfer learning and data augmentation.

4.2. Results

The standard CNN model was trained for 200 epochs. Based on the training, the accuracy was 77.91%. Some of the important metric measure such as loss, accuracy, and validation accuracy are presented in the table 3 below. The standard CNN had a precision of 76.86% and Recall of 50.19%.

Table 3: Standard CNN

Metric	Value (%)
Accuracy	77.91
Precision	76.86
Recall	71.19

The ConvNet accuracy against epoch for the train and test is plotted in the figure below. Based on the visualization, it is evidenced that accuracy for both the train and test samples increases with an increase in number of epoch. As accuracy increases, the model is portrayed as performing effectively based on the given data. More of the images are being classified correctly as being positive and negative for COVID-19. When accuracy is plotted against epoch, there is a progressive increase in the performance of the model. Accuracy against epoch for both trained and test data is shown in figure 7 below.

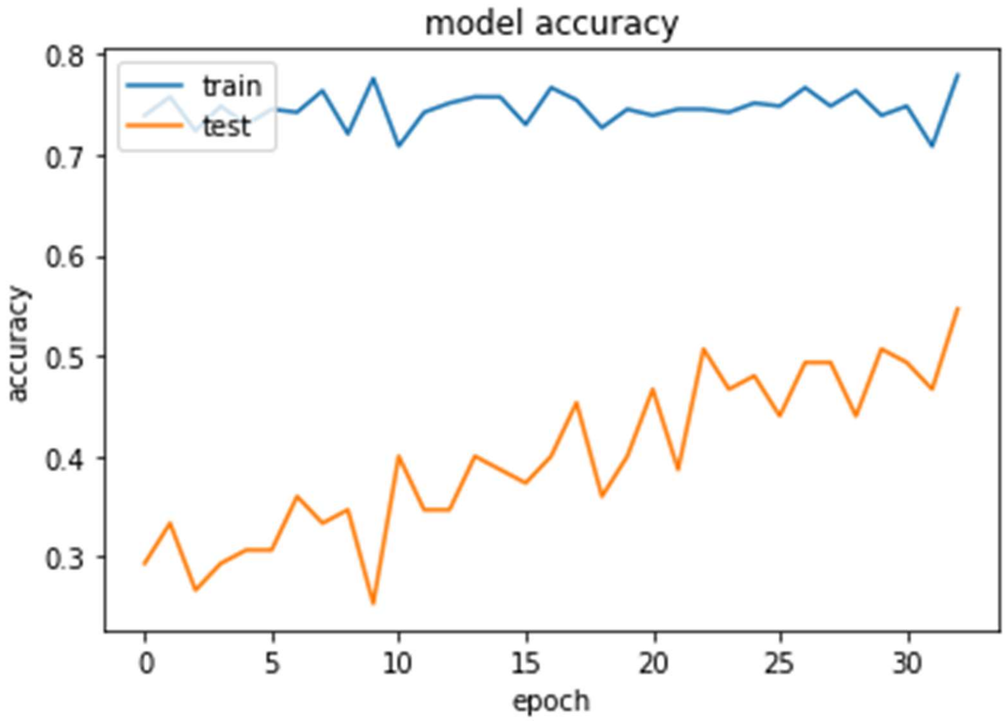


Figure 7: Model Accuracy

When loss is plotted against epoch, there is a progressive decrease in loss with an increase in the number of epochs for both training and test data. Loss, in this case, occur as a result of a bad prediction. It indicates how bad the model's prediction is based on the single example given (Albawi, Mohammed & Al-Zawi, 2017. The model is considered perfect if the loss is zero or near zero, otherwise, the loss is greater (Albawi, Mohammed & Al-Zawi, 2017. Training a model ensures that the loss function is reduced as low as possible. Higher loss indicates bad prediction for a model. Based on our training, the loss reduces from 7 to an average of 2 which means that as the model continues to train, a high level of accuracy is evidenced because of the lower loss value. The ConvNet loss against epoch for the train and test data is plotted in figure 8 below.

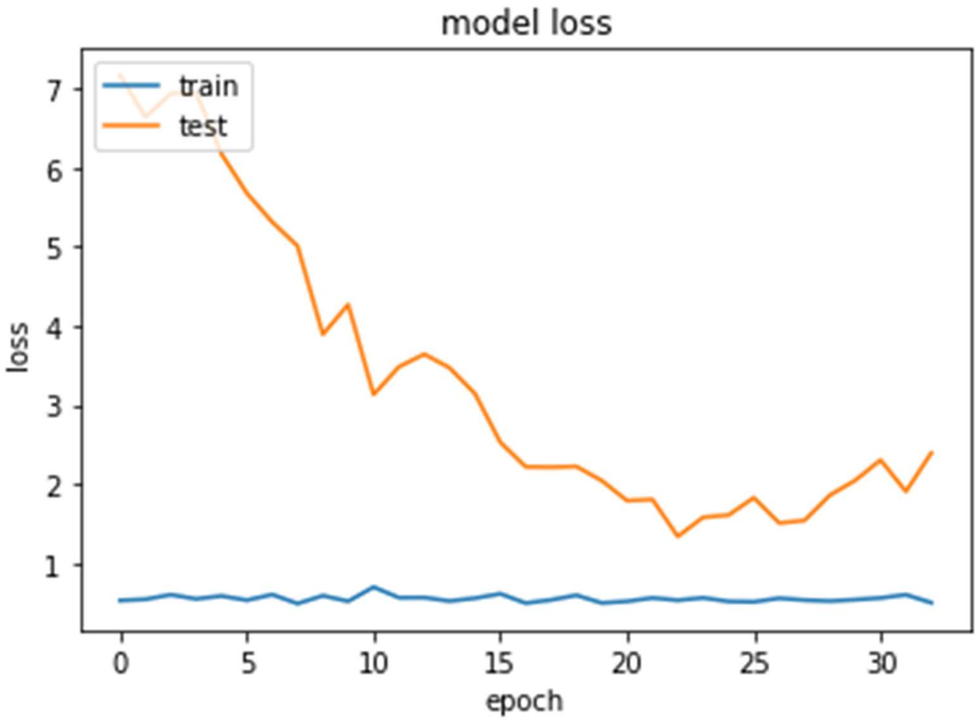


Figure 8: Model Loss

Based on the modeling with transfer learning and data augmentation, improved performance in accuracy and loss value was improved. Transfer learning was able to extract more features that effectively classified the GGOs as either COVID-19 or NoCOVID-19. The model comprises a total of 155 layers, and trainable layers were 56. The model was exposed to 70 epochs. The result shows that the accuracy rate was 97.36 on average and loss function 0.09. The precision was 96.19% and Recall was 8.33%. Table 4 below shows the result.

Table 4: CNN with transfer learning and data augmentation result

Metric	Value (%)
Accuracy	97.36
Precision	96.19
Recall	88.33

The ConvNet accuracy against epoch for the train and text is plotted in the figure below for the trained and test data sample that employs transfer learning and data augmentation. The graphs show that accuracy increases as the number of epochs increases. In addition, it is demonstrated that more features have been extracted that are adequate to classify the images. The graph of the accuracy of the model for the train and test data is shown in figure 9 below.

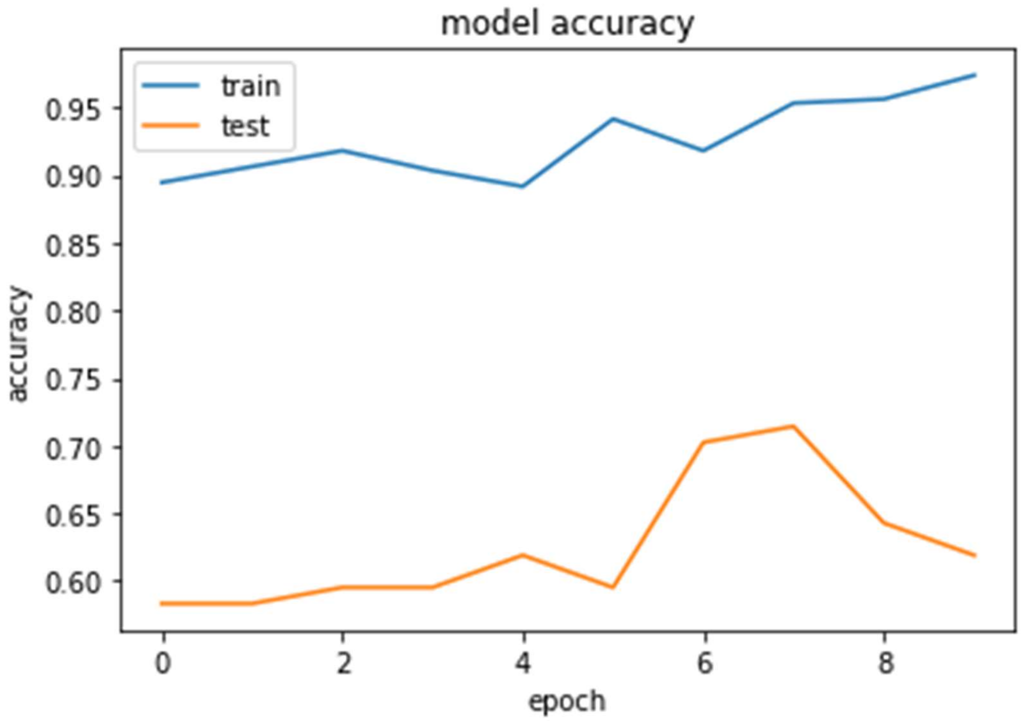


Figure 9: Model Accuracy

Compared to the standard CNN, the CNN with transfer learning and data augmentation produces higher accuracy and reduces loss. Furthermore, it shows that the model performs better as more features have been extracted and classified accordingly. The ConvNet loss against epoch for the train and text is plotted in figure 10 below.

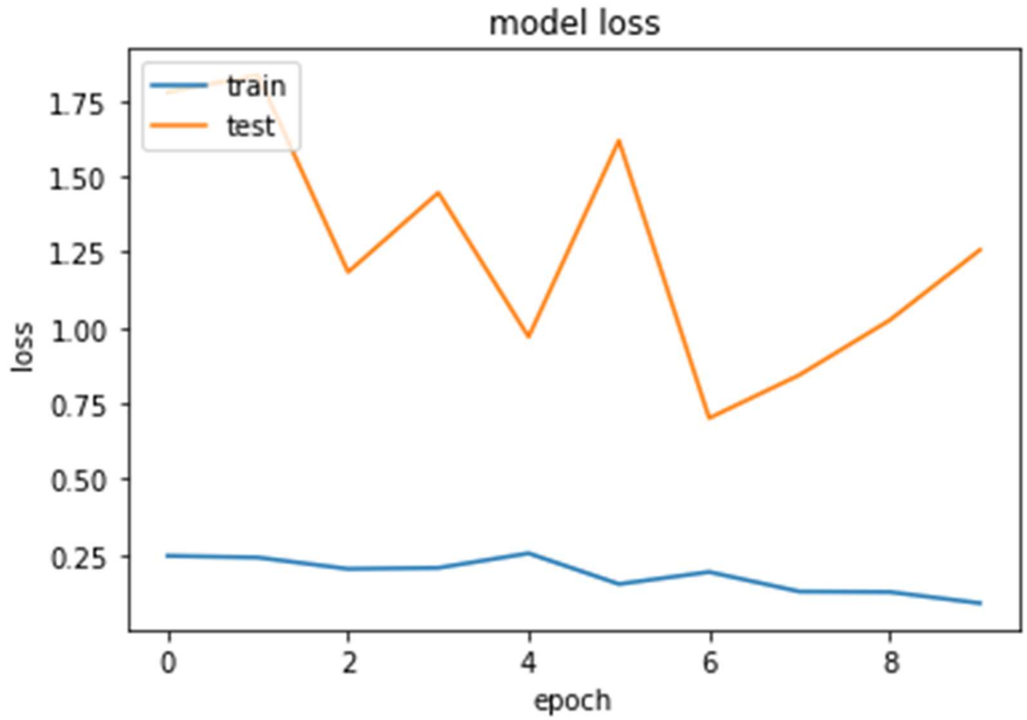


Figure 10: Model Loss

4.3. Discussion

The study's main objective was to develop a Machine Learning Model for identifying Ground Glass Opacities in the lungs CT scan images. A CNN classification model was used for the experiment. The CNN model was modified and used for transfer learning with data augmentation. A CNN model with transfer learning and data augmentation was selected for implementation. The CNN network architecture was modified based on activation functions and loss functions to produce a continuous value.

As opposed to the standard CNN model, the CNN with transfer learning and data augmentation was the ideal deep learning methodology. MobileNet was used in transfer learning. The deep learning methodology learned the best informative features automatically during the training process without human intervention. While manual feature extraction suffer from high dimensionality, learning these features automatically was ideal in the research. CNN with transfer learning was selected as the best because of its high accuracy as compared to the standard model. The model was able to learn the best set of representatives in a high dimension space for discrimination of predefined classes. The weights of the model was updated through backpropagation and minimization of cross-entropy error, this was done over several epochs.

The CNN model with transfer learning also demonstrates an increasing accuracy with a reducing loss. It shows that as the number of epoch increases during training as loss reduces considerably.

Other models have been developed to classify GGOs in the lungs CT scan images for various pathologies. A model by Wang et al., (2020) on weakly-supervised framework for COVID-19 classification had an accuracy of 84% for positive prediction and negative prediction accuracy of 98.2%. Also, according to Han et al. (2020) on the attention-based deep 3D model for COVID-19 screening, the algorithm obtained an accuracy of 95.7%. Another model developed by Ouyang et al. (2020) on dual-sampling attention for diagnosis of COVID-19 from pneumonia recorded an accuracy of 87.5%. Makris, Kontopoulos & Tserpes (2020) also proposed a model for detecting COVID-19 from chest X-Ray images that uses deep learning and CNN with a classification performance accuracy reaching 95%. Our proposed model is much higher in accuracy than some of these models. With improvements to our model and adjustments that can extract more features of the GGOs, it will perform better in the classification of GGOs in monitoring COVID-19 progression.

4.4. Achievements

The overall objective was to develop a model that would classify Ground-Glass Opacities in the lungs from the CT images samples to monitor COVID-19 progression. The research has generated a working model that is able to organize COVID and NoCOVID images from CT scan using CNN with transfer learning and data augmentation with a higher performance accuracy of 97.36%. As compared to other classification algorithms for the classification of GGOs, the model performs better than most of them. With little improvement, the model developed will increase its performance while reducing the error rate as low as possible.

The feature characteristics of GGOs representing COVID-19 infection in the lungs were extracted. The feature extraction technique used in the project was an automated approach. Some of these features were occurrence in the lower lobes and periphery, appear bilaterally and multifocal, and they have round shapes. CNN was able to learn automatically and extract feature characteristics during learning. Transfer learning using pre-trained model was used to extract more feature characteristics of GGOs from the images. During training, filters are updated accordingly, optimized, and iterated over multiple epochs to minimize the loss function. Through the process of feature extraction, the GGOs responsible for COVID-19 infection were identified and classified accordingly. Two approaches used to validate the model were standard CNN and

CNN with transfer learning. The CNN with transfer learning model was chosen for implementation due to its high classification accuracy of 97.36%.

Additionally, the presence or absence of GGOs in the lungs was used to evaluate COVID-19 disease progression and determine the effectiveness of the current treatment option. It can also help in planning for a better treatment approach or course of action that will change the patient's current condition.

4.5. Limitations

One of the limitations of the study was getting the right amount of data. A convolutional neural network is data-hungry and requires a large amount of data. We got a total of 757 images for the dataset. The dataset consisted of 349 images of positive COVID-19 and 408 images of negative COVID-19. With an accuracy of 97.36% that was accompanied by false positives and negatives, having a large dataset for training could have resulted in higher accuracy.

Another limitation is on the identification of the GGOs characteristics that represent COVID-19. Most of these characteristics are manually identified, labeled, and training the model based on them. Since COVID-19 and imaging detection models are at an early stage of development and GGOs represent various pathologies, it was challenging to train the model for such specific features.

4.6. Further Work

There is a need to collect more data for effective classification. More feature extraction techniques should also be employed to improve the accuracy and reduce loss in the model. The model needs to fit mobile detection to help transmit data in remote servers to improve the detection rate. This will allow patients to screen themselves and relay results remotely to the hospital over the cloud. Many people will gain access to the service, which will help in faster diagnosis. Since the data will be large and can deprive computing resources, the model needs to be hosted in a cloud platform to facilitate faster processing and improve data accuracy.

References

1. Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1-6). IEEE.
2. Ayele, W. (2020). Adapting CRISP-DM for Idea Mining. *Int. J. Adv. Comput. Sci. Appl*, *11*(6), 20-32.
3. Butt, C., Gill, J., Chun, D., & Babu, B. A. (2020). Deep learning system to screen coronavirus disease 2019 pneumonia. *Applied Intelligence*, 1.
4. Caruso, S., Marrone, G., Liotta, R., Martino, L., & Vitulo, P. (2020). An Uncommon Presentation of Sarcoidosis with Ground-Glass Opacities. *American Journal of Respiratory and Critical Care Medicine*, *201*(8), e54-e55.
5. Chhikara, P., Singh, P., Gupta, P., & Bhatia, T. (2020). Deep Convolutional Neural Network with Transfer Learning for Detecting Pneumonia on Chest X-Rays. In *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals* (pp. 155-168). Springer, Singapore.
6. Chen, X., Yao, L., & Zhang, Y. (2020). Residual Attention U-Net for Automated Multi-Class Segmentation of COVID-19 Chest CT Images. *arXiv preprint arXiv:2004.05645*.
7. Dong, D., Tang, Z., Wang, S., Hui, H., Gong, L., Lu, Y., ... & Jin, R. (2020). The role of imaging in the detection and management of COVID-19: a review. *IEEE reviews in biomedical engineering*
8. Fan, D. P., Zhou, T., Ji, G. P., Zhou, Y., Chen, G., Fu, H., ... & Shao, L. (2020). Inf-Net: Automatic COVID-19 Lung Infection Segmentation from CT Images. *IEEE Transactions on Medical Imaging*.
9. Hassanien, A. E., Mahdy, L. N., Ezzat, K. A., Elmousalami, H. H., & Ella, H. A. (2020). Automatic x-ray covid-19 lung image classification system based on multi-level thresholding and support vector machine. *medRxiv*.
10. Han, Z., Wei, B., Hong, Y., Li, T., Cong, J., Zhu, X., ... & Zhang, W. (2020). Accurate screening of COVID-19 using attention-based deep 3D multiple instance learning. *IEEE transactions on medical imaging*, *39*(8), 2584-2594.
11. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

12. Huber, S., Wiemer, H., Schneider, D., & Ihlenfeldt, S. (2019). DMME: Data mining methodology for engineering applications–a holistic extension to the CRISP-DM model. *Procedia Cirp*, *79*, 403-408.
13. Kang, H., Xia, L., Yan, F., Wan, Z., Shi, F., Yuan, H., & Shen, D. (2020). Diagnosis of coronavirus disease 2019 (covid-19) with structured latent multi-view representation learning. *IEEE transactions on medical imaging*.
14. Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., ... & Dong, J. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, *172*(5), 1122-1131.
15. Kong, W., & Agarwal, P. P. (2020). Chest imaging appearance of COVID-19 infection. *Radiology: Cardiothoracic Imaging*, *2*(1), e200028.
16. Makris, A., Kontopoulos, I., & Tserpes, K. (2020). COVID-19 detection from chest X-Ray images using Deep Learning and Convolutional Neural Networks. *medRxiv*.
17. Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Orallo, J. H., Kull, M., Lachiche, N., ... & Flach, P. A. (2019). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*.
18. Ng, M. Y., Lee, E. Y., Yang, J., Yang, F., Li, X., Wang, H., ... & Hui, C. K. M. (2020). Imaging profile of the COVID-19 infection: radiologic findings and literature review. *Radiology: Cardiothoracic Imaging*, *2*(1), e200034.
19. Ouyang, X., Huo, J., Xia, L., Shan, F., Liu, J., Mo, Z., ... & Shi, F. (2020). Dual-Sampling Attention Network for Diagnosis of COVID-19 from Community Acquired Pneumonia. *IEEE Transactions on Medical Imaging*.
20. Shi, H., Han, X., Jiang, N., Cao, Y., Alwalid, O., Gu, J., ... & Zheng, C. (2020). Radiological findings from 81 patients with COVID-19 pneumonia in Wuhan, China: a descriptive study. *The Lancet Infectious Diseases*
21. Song, Y., Zheng, S., Li, L., Zhang, X., Zhang, X., Huang, Z., ... & Chong, Y. (2020). Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. *medRxiv*.
22. Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017, May). SCRUM model for agile methodology. In *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 864-869). IEEE.

23. Saeedizadeh, N., Minaee, S., Kafieh, R., Yazdani, S., & Sonka, M. (2020). COVID TV-UNet: Segmenting COVID-19 Chest CT Images Using Connectivity Imposed U-Net. *arXiv preprint arXiv:2007.12303*.
24. Shafique, U., & Qaiser, H. (2014). A comparative study of data mining process models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, *12*(1), 217-222.
25. Shen, S., Bui, A. A., Cong, J., & Hsu, W. (2015). An automated lung segmentation approach using bidirectional chain codes to improve nodule detection accuracy. *Computers in biology and medicine*, *57*, 139-149.
26. Visa, S., Ramsay, B., Ralescu, A. L., & Van Der Knaap, E. (2011). Confusion Matrix-based Feature Selection. *MAICS*, *710*, 120-127.
27. Wang, X., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., ... & Zheng, C. (2020). A Weakly-supervised Framework for COVID-19 Classification and Lesion Localization from Chest CT. *IEEE Transactions on Medical Imaging*.
28. Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., ... & Xu, B. (2020). A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID19). *MedRxiv*.
29. Wang, J. (2011). The Study on Cross-Industry Standard Process for Data Mining in E-Marketing. In *Applied Mechanics and Materials* (Vol. 66, pp. 2298-2303). Trans Tech Publications Ltd.
30. World Health Organization. (2020). Coronavirus disease 2019 (COVID-19): situation report, 46.

Appendices

Appendix 1: Code Listings

Code Listing 1: Converting image to grayscale

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
def rgb2gray(rgb):
    return np.dot(rgb[...,:3], [0.2989, 0.5870, 0.1140])
img = mpimg.imread('image.png')
gray = rgb2gray(img)
plt.imshow(gray, cmap=plt.get_cmap('gray'), vmin=0, vmax=1)
plt.show()
```

Code Listing 2: Thresholding

```
def read_image(name=None):
  image = cv2.imread('image.png')
  grey_scaled = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
  _, threshold_image = cv2.threshold(grey_scaled,240,255,cv2.THRESH_BINAR)
  return threshold_image
```

Code Listing 3: cropping image

```
#image cropping
def crop_coordinates():
  pos = np.empty((0,40,40),dtype=float)
  neg = np.empty((0,40,40),dtype=float)
  store_xy = defaultdict(list)
  all_coordinates = get_all_coordinates()
  for img,xy in all_coordinates.items():
    for loc in list(xy[-1]):
      ggo_pos,ggo_neg = generate_negpos_sample(int(loc[1]),
      int(loc[0]),
      read_image_bw(xy[0]))
  if ggo_pos.shape ==(40,40) and ggo_neg.shape ==(40,40):
    afeature_pos = np.expand_dims(ggo_pos,axis=0)
    afeature_neg = np.expand_dims(ggo_neg,axis=0)
    pos = np.append(pos,afeature_pos,axis=0)
    neg = np.append(neg,afeature_neg,axis=0)
  return pos,neg
```

Code listing 4: Splitting train set and test set

```
positive, negative = crop_coordinates()
```

```
train_set = np.append(positive,negative,axis=0)/255
labels = np_utils.to_categorical(np.array([1]*
           positive.shape[0]
           +[0]*negative.shape[0]))
X_train, X_test, y_train, y_test = train_test_split(train_set,
           labels,test_size=0.20,
           random_state=42)
outfile = '/content/gdrive/MyDrive/Msc Project/COVID-CT-master/COVID-CT-
master'
np.savez(outfile,
           X_train=X_train,
           X_test=X_test,
           y_train=y_train,
           y_test=y_test)
```

Code Listing 5: Standard CNN model

```
model = Sequential()

model.add(Conv2D(32,(5,5), activation = 'relu', input_shape = (Image_width
, Image_width, Image_channels)))

model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(64,(5,5), activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512, activation = 'relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(2, activation = 'softmax'))

#compiles the model
model.compile(loss = 'categorical_crossentropy', optimizer='rmsprop', metr
ics=['accuracy'])
```

Code Listing 6: MobileNetV2 for transfer learning

```
mobilenet_conv_base = MobileNetV2(input_shape=IMAGE_SHAPE,include_top=Fals
e,weights='imagenet')
mobilenet_conv_base.trainable = True
```

```
# How many layers are in the mobilenetv2 base model?
print("Number of layers in the base model: ", len(mobilenet_conv_base.layers))
mobilenet_conv_base.summary()

# Fine tune from this layer onwards and freeze all the layers before the `fine_tune_at` layer
fine_tune_at = 100
for layer in mobilenet_conv_base.layers[:fine_tune_at]:
    layer.trainable = False

#Optimizing the model
# Added transfer learning with top layer off
# Changed global average pooling to max pooling

Regularizer = l2(0.001)

model = Sequential([
    mobilenet_conv_base,
    layers.Dropout(0.2),
    layers.Conv2D(32, 3, activation='relu'),  #,activity_regularizer=Regularizer, kernel_regularizer=Regularizer
    layers.Dropout(0.2),
    layers.GlobalAveragePooling2D(),
    layers.Dense(2, activation='softmax')
])
print('This is the number of trainable weights ''after freezing the conv base:', len(model.trainable_weights))
model.compile(optimizer=optimizers.Adam(lr=BASE_LEARNING_RATE),loss = 'categorical_crossentropy',metrics=['accuracy'])
model.summary()
```

Code Listing 7: Testing image

```
img = load_img(os.path.join(drive_path, 'test','test997.png'), target_size=(IMAGE_SIZE, IMAGE_SIZE))
img_array = img_to_array(img)
img_array = tf.expand_dims(img_array, 0)  # Create batch axis
print(img_array.shape)
predictions = model.predict(img_array)
predicted_probabilities = predictions.max(1)
print("predicted prob", predicted_probabilities)
print(predictions)
score = predictions[0]
print(score)
```