REPUBLIC OF TURKEY

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

CONCEPTUAL UNDERSTANDINGOF DEEP LEARNING WIRELESS NEURAL NETWORK

Ibtisam Jassim Mohammed MOHAMMED

Master`s Thesis

Supervisor

Asst. Prof. Dr. Ayca Kurnaz TÜRKBEN

Istanbul, 2022

CONCEPTUAL UNDERSTANDINGOF DEEP LEARNING WIRELESS NEURAL NETWORK

Ibtisam Jassim Mohammed MOHAMMED

Information Technologies

Master`s Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled CONCEPTUAL UNDERSTANDINGOF DEEP LEARNING WIRELESS NEURAL NETWORK prepared by IBTİSAM JASSİM MOHAMMED and submitted on 11/08/2022 has been **accepted unanimously** for the degree of Master of Science in Information Technologies

Asst. Prof. Dr. Ayca Kurnaz TÜRKBEN

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Ayça Kurnaz TÜRKBEN	Faculty of Engineering and Architecture, Altinbas University	_____
Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Faculty of Engineering and Architecture, Altinbas University	_____
Asst. Prof. Dr. Serdar KARGIN	Faculty of Engineering and Architecture, Beykent University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master`s Thesis.

Submission date of the thesis to the Graduate Education Institute: ____/____/____

I hereby declare that all information presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Ibtisam Jassim Mohammed MOHAMMED

Signature

DEDICATION

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears.

PREFACE

I might want to thank my administrator: Asst. Prof. Dr. Ayca KURNAZ Please let me express my profound feeling of appreciation and gratefulness to both of you for the information: direction and unrestricted help you have given me. I want you to enjoy all that life has to offer and further achievements and accomplishments throughout your life.

ABSTRACT

CONCEPTUAL UNDERSTANDINGOF DEEP LEARNING WIRELESS NEURAL NETWORK

Mohammed, Ibtisam

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst. Prof. Dr. Ayca KURNAZ

Date: 08/2022

Pages: 117

New methods for identifying COVID-19 in light of the principles of artificial intelligence (AI) have been offered, specifically substantial learning (DL) and AI. Colleague devices in the clinical consideration area are intended to ensure appropriate assurance with faster and more thorough outcomes (ML). Traditional ML and DL approaches have been developed by a number of analysts to aid specialists in reaching the proper conclusion. Such technologies can assist in categorizing the results of the chest's X-shaft or CT scan into two categories: tainted and common. Past investigation around here and our goal in this paper is to uncover knowledge into the assessment. This paper reviews 17 conveyed research papers that have been inspected and situated considering unequivocal standards to get an extensive point of view on the usage of AI models to Existing public COVID-19 datasets.

Keywords: Artificial Neural Network, CNN Algorithm, COVID-19, Images Detected, Python, Layers, Label

TABLE OF CONTENTS

Pages

ABSTRACT vii

LIST OF TABLES x

LIST OF FIGURES xi

ABBREVIATIONS xvi

1 INTRODUCTION 1

1.1 A PROTOCOL FOR CHOOSING STUDIES RELATED TO COVID-19: 3

1.2 FOR COVID-19, DEEP AND MACHINE LEARNING 5

1.3 TECHNIQUES FOR MACHINE LEARNING 5

1.3.1 Methods of Supervised Learning 8

1.3.2 Methods of Unsupervised Learning 12

1.4 DEEP LEARNING APPROACHES FOR COVID-19 15

1.4.1 Convolutional Neural Network 19

1.4.2 Neural Network in Depth 23

1.4.3 Network of Recurrent Neurons 25

1.4.4 Networks of Generative Adversaries 26

1.5 FOR COVID-19, A HYBRID TECHNIQUE COMBINING MACHINE _LEARNING AND DEEP LEARNING 27

1.6 COVID-19 PUBLIC DATASETS 28

1.7 MEASURES FOR COVID-19_EVALUATION USING DEEP_LEARNING AND MACHINE LEARNING 29

1.8 AIMS AND OBJECTIVES 33

1.9 CONTRIBUTION 34

1.10 PLANNING OF CHAPATERS 34

2 LITERATURE REVIEW 35

2.1 COVID-19 35
2.2 DEEP LEARNING TO DETECT COVID-19: 36
2.3 APPLICATION DEEP LEARNING TO DETECT COVID-19: 37
2.3.1 ML Regressors for Forecasting COVID-19: 37
2.3.2 ML Classifiers for COVID-19 39
2.3.3 Application of DL for Diagong of COVID-19: 40
2.4 CHALLENGE OF DL TO PREDICT TO MANAGE 42
3 RESEARCH METHODOLOGY 43
3.1 DATA SET DESCRIPTION 44
4 RESULT 95
5 FUTURE WORK AND RECOMMENDATIONS 99
5.1 FUTURE WORK 99
5.2 RECOMMENDATIONS: 101
REFERENCES 104

LIST OF TABLES

Pages

Table 1.1: COVID-19 datasets that are currently available to the public.....31

Table 1.2: COVID-19 datasets that are currently available to the public “tables continued”...32

Table 1.2: COVID-19 datasets that are currently available to the public “tables continued”...33

Table 2.1: Some pandemics in human history.....36

Table 2.2: ML for forecasting and prediction COVID-19 cases. 38

Table 2.3: Some existing ML methods/classifier with its results.....39

Table 2.4: Distribution of studies for various medical imaging modalities. 40

Table 2.5: Summary of features, applications, and limitations of different modalities for COVID-19 diagnosis. 41

Table 3.1: Compare Experiments in terms of reaches required Val accuracy. 93

Table 3.2: Compare Experiments in terms of number of epochs. 94

Table 3.3: Layers and dropout for each experiment. 94

Table 4.1: pc resource.....95

Table 4.2: Experiment Number Effectiveness and resource consumption.....96

Table 5.1: Comp between different algorithms. 99

LIST OF FIGURES

Pages

Figure 1.1: Covid person to person spread. 3

Figure 1.2: Covid publication. 4

Figure 1.3: Flowchart of selected studies involving the queryand inclusion criteria. 6

Figure1.4: Categorizing of related works presents the organized AI systems (ML and DL) picked in this study paper 7

Figure 1.5: Machine learning for covid19 & number of Ml & mapping of ML publication As shown by 12

Figure 1.6: Machine learning for covid19 & number of Ml & mapping of ML publication As shown by 13

Figure 1.7: Supervised_learning_approches for_covid19 15

Figure 1.8: Application_of_agglomerative&devised_five_object 16

Figure 1.9: Deep_learning_steps for covid-19 18

Figure1.10: Deep learning publication in different counties for covid-19 18

Figure 2.1: Framework for COVID-19 diagnosis. 41

Figure 3.1: Chest_xray_Corona_Metadata. 45

Figure 3.2: Chest_xray_Corona_dataset_Summary. 46

Figure 3.3: Loading csv files. 46

Figure 3.4: divide data by column dataset_type. 47

Figure 3.5: the first 2 ‘rows of Chest_xray_Corona_Metadata.csv file. 48

Figure 3.6: The first five lines of Chest_xray_Corona_dataset_Summary.csv file.....49

Figure 3.7: Chest_xray_Corona_Metadata.csv file describe method.....49

Figure 3.8: Chest_xray_Corona_dataset_Summary.csv file describe method. 50

Figure 3.9: Shape of the csv files. 50

Figure 3.10: Data type of Chest_xray_Corona_Metadata.csv file. 51

Figure 3.11: Data type of Chest_xray_Corona_dataset_Summary.csv file.....51

Figure 3.12: Unique values in label_2_Virus_category column. 52

Figure 3.13: Unique values in label_1_Virus_category. 52

Figure 3.14: The first five lines of training data.....53

Figure 3.15: Null values in training data. 53

Figure 3.16: Unique values in label_1_Virus_category in training data.....54

Figure 3.17: Bar chart for label in training data. 54

Figure 3.18: Bar chart for label_1_Virus_Category in training data.....55

Figure 3.19: Bar chart for label_2_Virus_Category in training data.....55

Figure 3.20: Shape and null values in test data. 56

Figure 3.21: Unique values in label_1_Virus_categor in test data.....56

Figure 3.22: Bar chart for label_1_Viirus_category in test data. 57

Figure 3.23: Bar chart for label in first csv file. 57

Figure 3.24: Bar chart for label_1_Viirus_category in first csv file. 58

Figure 3.25: Replace empty values and fill column label in training data. 59

Figure 3.26: Replace empty values and fill column label in test data. 59

Figure 3.27: First two lines of test data in which the Label column is equal to Normal/NA. 60

Figure 3.28: First two lines of test data in which the Label column is equal to Pnemonia/NA. 60

Figure 3.29: Test data in which the Label column is equal to Pneumonia/ COVID-19. 61

Figure 3.30: Split training data into train and test data. 61

Figure 3.31: First five lines of the test data will be shown after the previous division process. 62

Figure 3.32: Image augmentation. 64

Figure 3.33: Augment training images directly. 66

Figure 3.34: Augment test images directly. 67

Figure 3.35: Call back function. 68

Figure 3.36: The experiment of using learning rate equal to " 2. 68

Figure 3.37: Val_accuracy with learning rate equals "2 69

Figure 3.38: The experiment of using learning rate equal to "3. 70

Figure 3.39: Val accuracy with learning rate equals "3 71

Figure 3.40: The experiment of using learning rate equal to "4. 71

Figure 3.41: Val accuracy with learning rate equals "4 72

Figure 3.42: The CNN model used to determine the optimal learning rate for the rest of the experiments. 73

Figure 3.43: First experiment 7 layers and dropout ".4. 74
Figure 3.44: Trainable params in first experiment. 75
Figure 3.45: The results of the first experiment. 75
Figure 3.46: Layers of the second experiment. 76
Figure 3.47: Trainable params in second experiment 77
Figure 3.48: The results of the second experiment. 77
Figure 3.49: Trainable params in third experiment. 78
Figure 3.50: The result of the third experiment. 79
Figure 3.51: The layers in the fourth experiment. 80
Figure 3.52: Trainable params of the fourth experiment. 80
Figure 3.53: Result of fourth experiments. 81
Figure 3.54: Layers of fifth experiment. 82
Figure 3.55: Trainable params of the fifth experiment. 83
Figure 3.56: Result of fifth experiment. 83
Figure 3.57: Sixth experiment parameters. 84
Figure 3.58: Trainable params of the sixth experiment. 85
Figure 3.59: Result of sixth experiment. 85
Figure 3.60: Model accuracy and model loss. 86
Figure 3.61: Layers of Seventh experiment. 87
Figure 3.62: Trainable params of the Seventh experiment. 88

Figure 3.63: Result of Seventh experiment. 88

Figure 3.64: Model accuracy and model loss. 89

Figure 3.65: Layers of eighth experiment. 90

Figure 3.66: Trainable params of the eighth experiment. 91

Figure 3.67: Result of eighth experiment. 91

Figure 3.68: Model accuracy and model loss in eighth experiment. 92

Figure 3.69: Compare time taken by each experiment. 93

Figure 4.1: Cnn_archeticture. 97

Figure 4.2: Attr_histogram. 98

Figure 5.1: Percentage of sick people. 100

Figure 5.2: Breath_problem. 101

ABBREVIATIONS

AI : Artificial Intelligence

DL : Deep learning

BDT : Binary decision tree

ML : Machine Learning

CNN : Convolutional Neural Networks

1. INTRODUCTION

The condition or infection known as coronavirus disease (COVID-19), also recognized as the middle east respiratory syndrome coronavirus 2 (SARS-CoV-2), is brought on by a new kind of coronavirus termed 2 " "19-nCoV. (Chung&et al&; Gralinski &Menachery, & ;&Rothe et

al.&) [6]. It was first found in the Chinese city of Wuhan in Hubei. The World Health Organization (WHO) [11] was notified of the very first COVID-19 case on Dec 31, 2 " "19. On January 3 ", ", the COVID-19 epidemic was recognized as both a global outbreak. (Sohrabi and colleagues, "“) [32. H1N1 has been recognized by the WHO as a worldwide pandemic disease since 2 ' "9; COVID-19 was indeed the second to be identified (Cucinotta & Vanelli_ " "; Zarocostas-2 ' "9) [12]. Although experts are looking into possible disease transmission mechanisms, the sickness spread mostly via intimate contact with patients who already had the condition. Fever, a wet cough, and respiratory problems are the main signs of COVID-19 infection. According to Tsatsakis et al. [22], up to 1 percent of individuals have gastrointestinal problems including diarrhoea, while others may feel tired or have muscle aches. At first, it was believed that direct contact between people might spread the virus. As a result, social isolation can lower the risk of contracting the condition. Up to 6 meters can separate a person from a disease. Consequently, one of the main ways the disease spreads is by the inhalation of droplets produced when an infected person talks or sneezes. Additionally, some people might not always show signs of COVID-19 Figure 1 depicts a person to another without any social isolation. Numerous techniques, even those based on nucleotides and the polymerase (PCR), can be used to diagnose COVID-19 (Abdulkareem, Mohammed, et al., Because these procedures are used to identify organ changes, patients may be compelled to have these pathology investigations. The two most common pathological tests are CXR and CT scanning (Kumar, Nagpal, et al.; Kumari et al., "; Mohammed et al) [32]. There are several benefits to using chest CT & X-ray imaging for COVID-19 diagnosis.

These technical advancements are notable. For instance, compared to other technologies, the CT scan is relatively rapid and gives more information about the patient's condition. The

results of an X-ray chest can be obtained. The possible influence of adjacent bone on brains CT scans and the limitations of X-ray chest scanning to offer 3D information are two constraints that may restrict the efficacy and applications of both technologies. Sadly, there is no effective way to diagnose COVID-19 at this time. TypicallyCOVID-19 diagnosis uses medical pictures (such as X-rays and CT scans). These photos are examined by an expert, who uses their diagnostic abilities to assess the data. Generally speaking, lengthy workdays can cause weariness, which can lead to doctors misdiagnosing patients. Similar to COVID19, various lung conditions might cause aberrant CT or CXR readings in those who have the disease. Furthermore, a normal Ct or CXR result does not necessarily indicate a negative COVID-19 case. As a result, the healthcare industry needs support technologies to guarantee accurate diagnosis. New methods for identifying COVID-19 have been developed utilizing the principles of ai technology (AI), notably transfer learning (DL) & machine learning, enabling quicker and more accurate findings (ML) (Alimadadi et al.) Classical ML and DL were created by a number of academics. strategies that help doctors make accurate diagnoses [8]. These methods can aid in identifying good and unhealthy tissue in heart X-rays or CT scans. Following a number of steps, such as detecting an X-ray image, third generation (3, and extracting differentiating characteristics from input images, images are fed into ML or DL model for a final prediction conclusion. The analysis of COVID-19 data to foresee projected red zones and the amount of infected patients are two more uses for comparable systems. Numerous Ai systems (DL and ML) were applied to COVID-19 patients based on X-ray / CT scans: 1. Guided instruction 2. Cui and associates' research on self-directed learning. Discover coronavirus, predict viral infection, and identify viruses using DL methods. The most often used DL techniques are: A convolution is the first. Alazab et al., and Apostolopoulos et al.) [7] Generative adversarial networks (GANs), sometimes referred to as deep neural networks (DNNs) (et al.,) [11] A deliberate choice was made based on the importance and quality of the literature. The primary source for the ML and DL research papers utilized in COVID-19 is depicted in Figure 1. Wiley are now only a lot of small digital archives that were searched in order to find the chosen works, each of which is labeled in the picture. The dispersion of

COVID-19 research is shown in Figure 2 by the kind of publishing, such as public access and close access

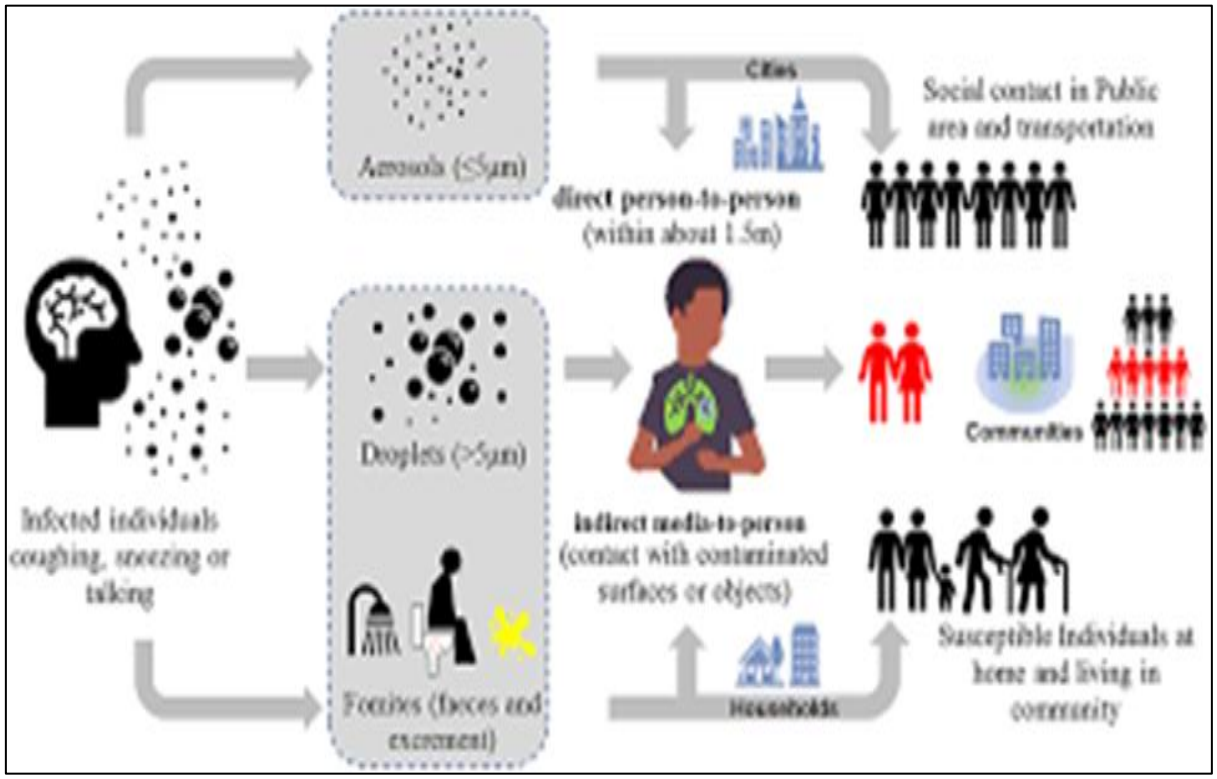


Figure 1.1: Covid person to person spread.

1.1 A PROTOCOL FOR CHOOSING STUDIES RELATED TO COVID-19:

The most pertinent keywords, "COVID-19," "machine learning," and "deep learning," were used to choose the studies. Digital databases such Elsevier, IEEE_Xplore_ MDPI, Springer, IOP Press, and Wiley were used, The COVID-19 episode was proclaimed as a worldwide flare-up on 30 January 2020 [14]., and only studies that were written in English were gathered. and up to 10% make GI-related side impacts, for instance, the runs [3]. Our protocol has five steps in its process. Finding papers on the COVID-19 epidemic was the first step. Studies on the examination of ML and DL many benefits [2]. methods for the COVID-19 pandemic were

contained in 255 publications that were retrieved in total. In this stage, three were combined in different ways and joined using the "&" operator the results of COVID-19 sometimes are not conspicuous [21].

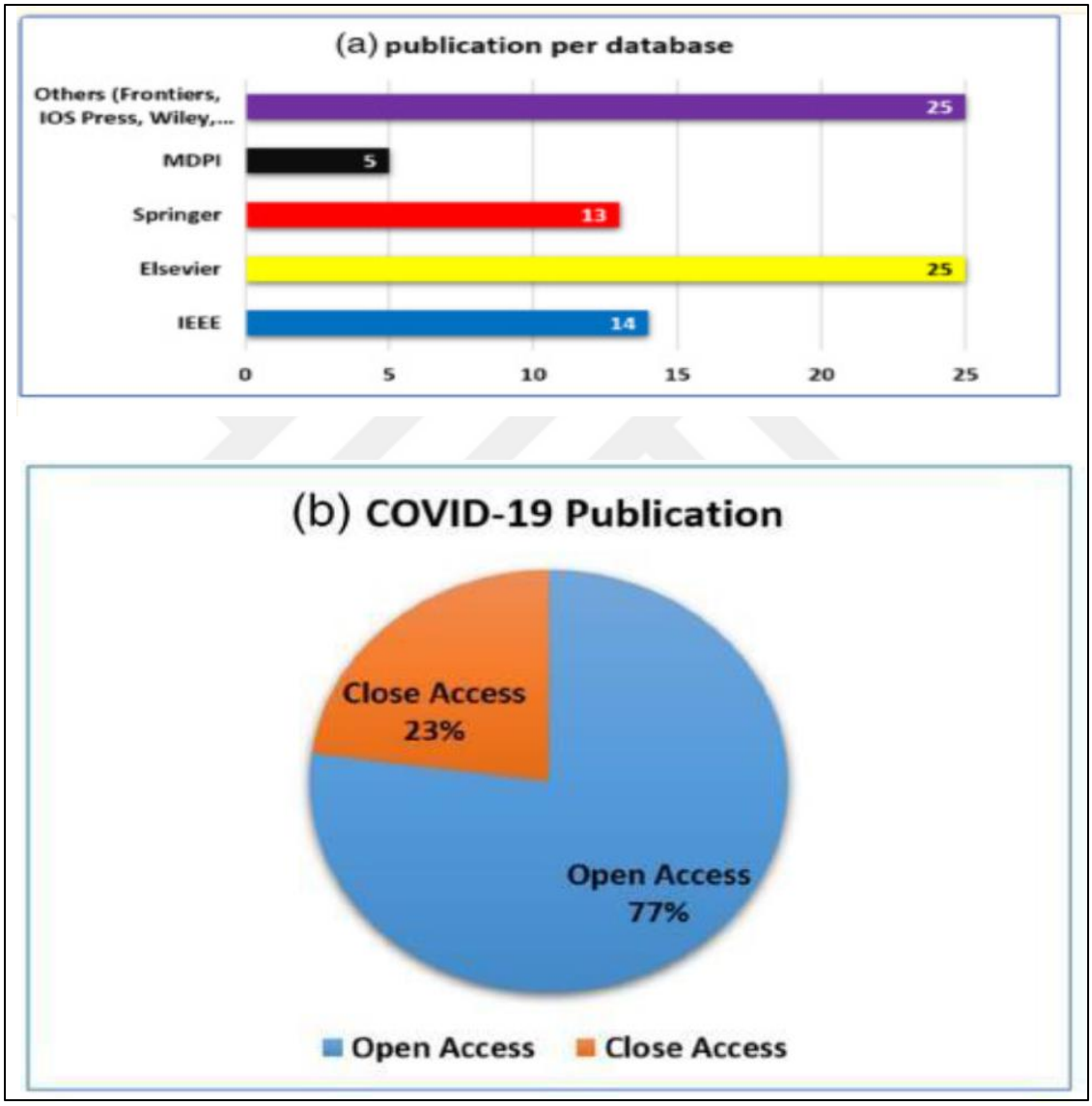


Figure 1.2: Covid publication.

1.2 FOR COVID-19, DEEP AND MACHINE LEARNING

Due to the widespread coronavirus infection, the application of ML and DL has improved the efficacy of traditional methods prediction. ML methods, such as, will be discussed in Section 3.1, were used to discover COVID-19. A variety of DL techniques have also been used for COVID-19 detection; Section 3.2 will examine these techniques. Hybrid approaches that combine ML and DL are employed to identify COVID-19 infection. Figure 4 displays the categorized AI techniques (ML and DL) applied for this survey piece.

1.3 TECHNIQUES FOR MACHINE LEARNING

By offering instructions, ML teaches computers to perform tasks on their own. It is a method of data analysis that entails building and fitting models to give computers the ability to "learn" via repetition and predict the future such methods are foreseeing a flare-up partition by investigating COVID-19 information and anticipating red zones and number of tainted cases with AI [15] [16].. ML is used for COVID-19 identification by looking at the input image distinguishing characteristics from it. MDPI, Springer, IOP Press and Wiley [11]. The prediction of the input picture is either supplied depending on these features the conjecture of the data picture will be given as a regular case or as a defiled case [24].

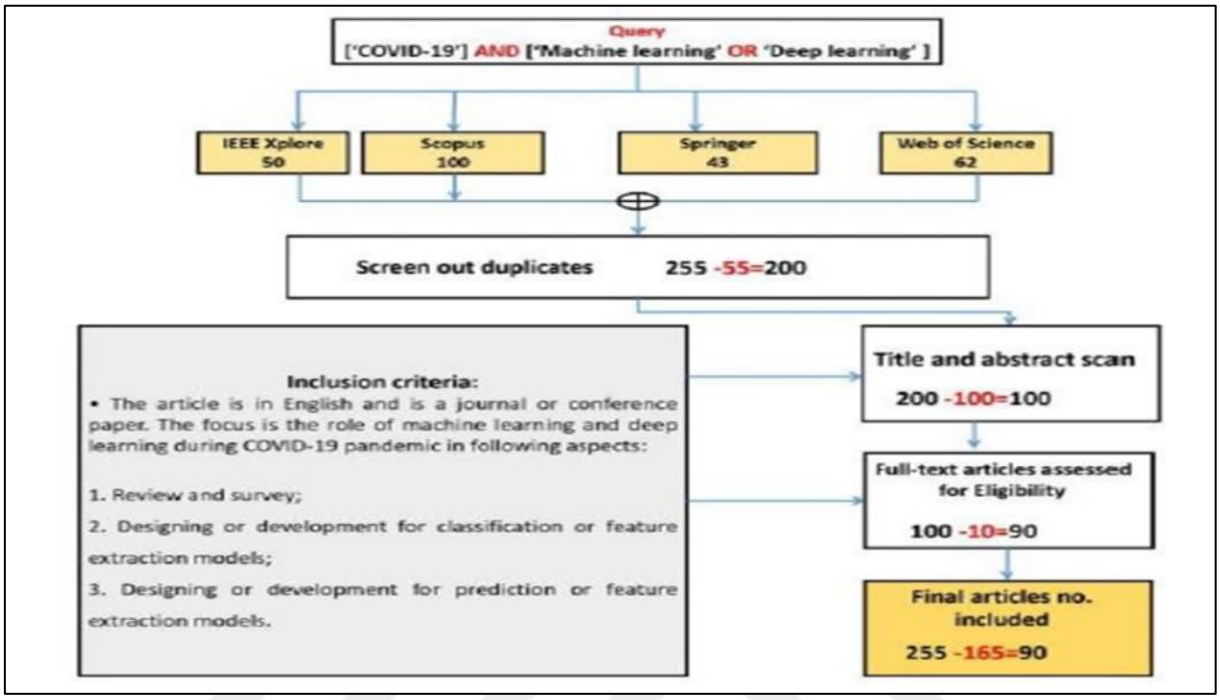


Figure 1.3: Flowchart of selected studies involving the queryand inclusion criteria.

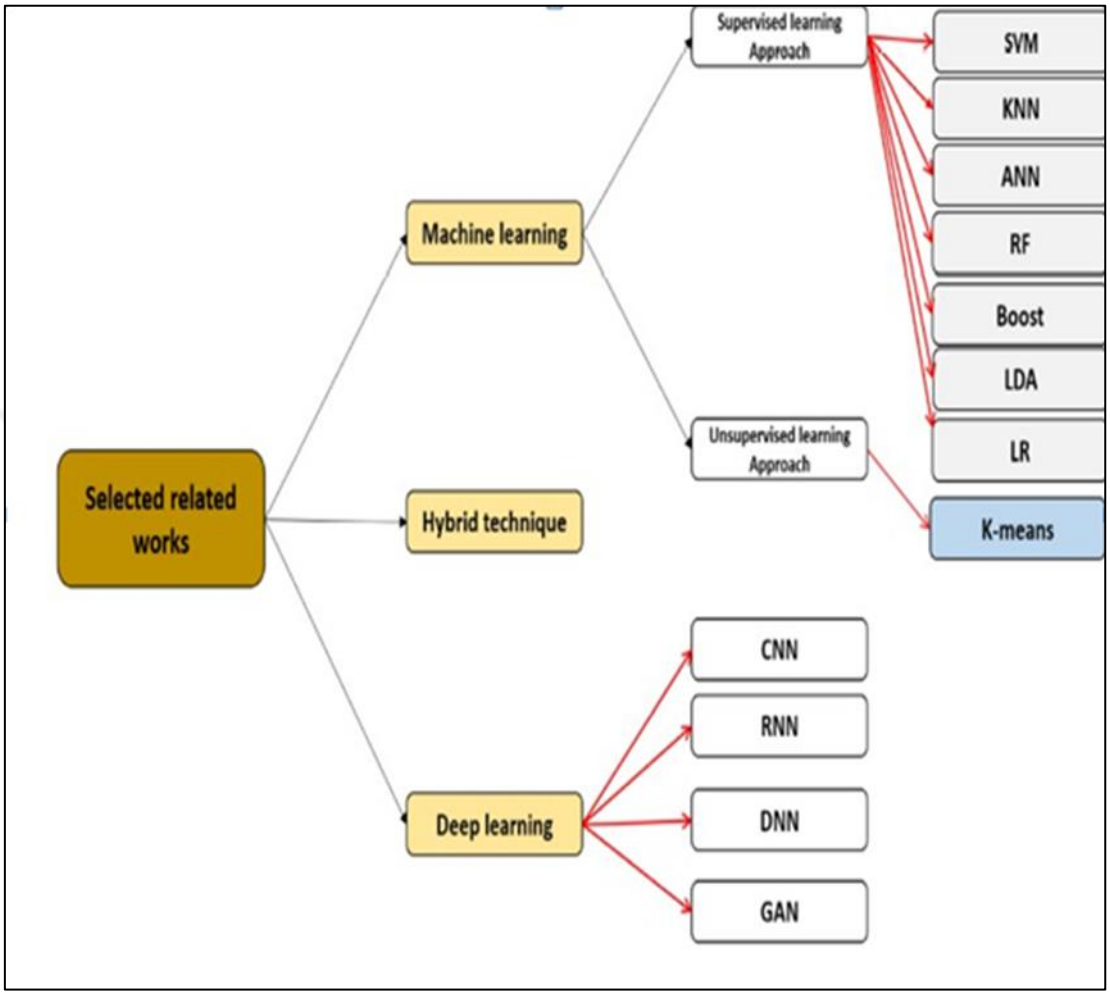


Figure 1.4: Categorizing of related works presents the organized AI systems (ML and DL) picked in this study paper [24].

Learning approaches described in 3.1.2 are generally the two groups into which ML algorithms employed for COVID-19 diagnosis_may_be grouped. The ML research for COVID-19 is displayed in Figure 5.

1.3.1 Methods of Supervised Learning

The identification of COVID-19 has utilised many regression, classification, and feature extraction techniques (Peng & Nagata,) [3]. The following objectives are achieved using these techniques: predicting the spread of the coronavirus throughout areas, analysing the pace of growth and available treatment options across various nations, analysing the association of the

Weather conditions' impact on the coronavirus, as well as (v) an examination of the virus's rate of transmission that was (M. Yadav et al., “) [12]. Recent studies have focused on the use of logistic forecasting models good when we use (Wang, Zheng, et al., “) [15], neural (Melin et al.,"a)[21], prediction of COVID 19 (Kavadi et al., "), and (Hasan et al.,"). In", Siddiqui et al. looked at the relationship between patient We compared the temperature with thewhen we talk aboutCOVID-19 case status using the k-means approach (suspected, confirmed, and dead). Clustering, data collection, and database design make up the three parts of the procedure. The dataset used in the first study is made up of situation reports on the coronavirus illness (COVID-2 " "19) [12] which have been gathered from the WHO. This dataset includes the infection that rates must in several regions of China. The second part describes the dataset, which consists of seven features The features of the lowest and highest temperatures are now included in the dataset. The rationale is that patient temperature is one of the crucial factors in identifying the state of a COVID-19 case. During the last phase, fresh K-means-based clustering algorithms are used to find patterns. The patterns demonstrated how temperature affected each location in the three COVID-19 states. Ong, Wong, et al. [5] proposed a computer model in this search based on ML prospective vaccine candidate proteins after analyzing the present COVID-19 vaccine candidates. RV's main job is to look into the bioinformatics of pathogen genomes. As a consequence, promising vaccine candidates are discovered. The dataset utilized in this study was sourced from the UniProt and NCBI datasets we talk about (Bairoch et al., 2 '" "5). It is composed of all the SARS-CoV-2 sequences and proteins that have been isolated from known human coronavirus strains. the creators of The Vaxign-ML model, which is based on RV and ML, was enhanced by applying XGBoost, KNN, and logistic regression (LR) approaches to predict the protein levels of all SARS-CoV-2

proteins (Wang, et al., Randhawa et al. '[8] used intrinsic COVID-19 genetic markers to build a decision tree. ML alignment-free approach to estimating the gene sequence of the COVID-19 virus. By analyzing just raw DNA sequence data work on logarithmic misfortune, the exactness of identification and testing time [19]., the alignment-free method quickly generates the taxonomy classifications of novel illnesses. The proposed method was assessed on a big dataset encompassing more than 5 'unique viral genomic sequences. These data are from the rang of well-known Virus-Host DB database. The method offers precise real-time unknown sequences and is a distinct method for analyzing pathogen genome sequences. Pinter et al. looked into the possibility of using a hybridization model of network-based fuzzy inference system and multi-layered perceptron-imperialist competitive algorithm to predict the outbreak of COVID-19 based on time series data compiled from the Hungary statistical reports of this study in this cap infected cases and death rates. Three measures were used to evaluate the efficacy of the proposed prediction model: absolute mean percentage The recommended prediction model fared well in terms of error, root mean, the overall fatality estimate, and the forecast for the COVID-19 epidemic. Using CXR images, Elaziz et al. '[12] created a visual diagnostic method for differentiating COVID-19 patients from healthy individuals. The initial step was to extract features from CXR images using fractional multichannel exponent moment. The calculation of images and data processing was sped up using a fully parallel multicore framework since it is a costly and time-consuming process. A unique feature selection method was proposed in the current study, and a refined manta-ray foraging optimal solution based on color development (MRFODE) was used to extract pertinent subset characteristics from the entire collection of features. A KNN approach is used to evaluate a potential set of attributes that are created repeatedly using MRFODE. In the first dataset, images from 43 distinct periodicals as well as works by Joseph Francis Cohenand, Paul Douglas, and Lan Dao were combined from two sources. These images belonged to two categories: viral pneumonia and common pneumonia. 1675 COVID-19 negative photographs and 216 COVID-19 positive shots were included in this collection. a research team composed of people from several the second dataset was gathered by Pakistan and Malaysia. The research team provided fresh images from COVID-19 database. These images were put

together by the Italian Association of Medical Radiology and Interventional Radiology. In this sample, there had been 219 COVID-19 good shots and 1341 COVID-19 bad photos. Classification recall, precision, and accuracy were used to rate the proposed strategy (MRFODE). According to the results, MRFODE was able to successfully classify COVID-19 patient samples at a promising level of accuracy. For Jackson's normal and COVID-19 patients, Fayyoumi et al. '[11] developed an online survey. Using information from the questionnaire, it was established if symptoms and indicators were present in both groups. The researchers assembled a COVID-19 dataset using data from different patientssymptoms, signs, and symptoms. The researchers then entered this data into a range of computer vision (ML) methods (SVM, cross recurrent neural network [Multi - layer perceptron]), and also statistical methodology (i.e., LR), in ability to predict potential COVID-19 patients. The classification accuracy demonstrated that MLP functioned at its highest level (91.62 percent). SVM delivered the most accurate results (91.67 percent). A alternative approach was proposed by Kavadi et colleagues. to end the COVID-19 pandemic in India. The COVID-19 database from India was used by the researchers. The proposed approach combines two well-known approaches that Nonlinear machine learning and partial derivative regression (NML). PDL was used to normalize the dataset, while NML was used for prediction. Experimental results show that this strategy outperforms past attempts in terms of process precision and forecast time. Wang, Zhang, and collaborators " "have used logistic modeling technique or the FbProphet ML algorithms to forecast the COVID-19 trend. The dataset utilized in this study has been the latest COVID-19 epidemical collection for data sets at the national level. This study covers a wide range of countries, such as Bolivia, China, India, Peru, and Jakarta. In reality, the cap value for the pandemic trend was determined using logistic modeling. The learning model for the FbProphet model was then created utilizing the results to foresee anticipated COVID-19 pandemic trend. The results of the experiment show that the proposed technique will enable judgement in a specific nation to act swiftly during a COVID-19 pandemic. For COVID-19 patients, Pollack et al. proposed a ventilation model which is based on the Classification classifier. The XGBoost classifier employs a decision trees, and the outcomes are combined in a single score. Data on people who joined with but were transferred

to these 5 facilities were used for this inquiry. Five US healthcare systems' hospitals from 24 April 'и 4 May 2 " "18 were compiled. Each patient's diastolic pressure (DBP), temperature, and creatinine levels (BUN) are among the twelve variables tracked (BUN). The proposed method creates a diagnostic ratio for predicting ventilation in relation to the Modified Ekg (MEWS). The recommended model also produced an excellent balance between accuracy and precision, outperforming MEWS in both (p “. "5) and having a greater sensitivity (“.9 " "). Using an MLbibliometric method, De Forte and Polimeni " "examined and evaluated research patterns in COVID-19. Information was collected for research articles on countries, outputs, publications, institutions, financing, keywords, and citation countsretrieved from Scopus. The results revealed the clinical manifestations of COVID-19 patients, named COVID-19 as perhaps the most common problem, and shown a marked surge in the number number articles published over the research period. Daniel et al " "examine tweets that reference the coronavirus to determine how the public feels about the pandemic. The relationship between the emergence of concern emotion and the time when COVID-19 levels peaked in the US was also discussed by the authors. These discoveries were

exemplified by using analytics that analyze the text through data visualization. In the perspective of textual analytics, two important machine learning (ML) algorithms are discussed, and their effectiveness in classifying tweets regarding the coronavirus over a range of timescales is compared. The two ML methods worked effectively, as seen by the high classification accuracy of the two algorithms for short Tweets. The accuracy attained by the two algorithms does not seem promising for longer Tweets. It was demonstrated in Abdur rahman et al. that Models can forecast COVID-19- Affected individuals ‘. Four widely used models were examined in order to predict the COVID-19 threat variables: least square shrink and selecting operator (LASSO), moving average (ES), regression methods (LR), and support vector machines (SVM). Over the course of the next 1 'days, each model predicted three distinct types of data, including that of the number of available infected people, the number or fatalities, and indeed the number of recoveries. The current scenario from the COVID-19 is used to evaluate these methods. The results show the ES model's outstanding performance as it constantly beats other comparable methods in terms of projecting new verified occurrences.

Two data augmentation techniques were proposed by Sedik and colleagues "to enhance the human brain (CNN) LSTM learning procedures, etc. enhancing, highlights were added to the dataset, in particular, most minimal temperature and most elevated temperature [20]. COVID-19 detection precision in the process. The results demonstrate that the offered methods improve testing time, detection precision, and logarithmic loss. Considering the availability of SIP within those counties, Cobb and Seale " "examined the COVID-19 growth patterns in US counties. To calculate the rates of compound rise, they utilized the total number of confirmed COVID-19 cases over the course of two months and ten days from 21 February 'to 31 March ". The tree ML model used the algorithm's single value to depict the virus's propagation rate from across time period under analysis. The results showed that SIP orders, due to high populations, dramatically reduced the frequency of COVID-19cases.

1.3.2 Methods of Unsupervised Learning

Unsupervised learning uses unlabeled data in contrast to supervise learning (i.e., clustering). Without using any extra information, the goal is to group the data into clusters that are comparable. Hierarchical clustering and partition clustering are the two broad categories into which the primary clustering techniques may be divided. There are two types of hierarchical methods: divisive (top-down approach) was used in studying the proposed technique. The request model showed merciless results [2].

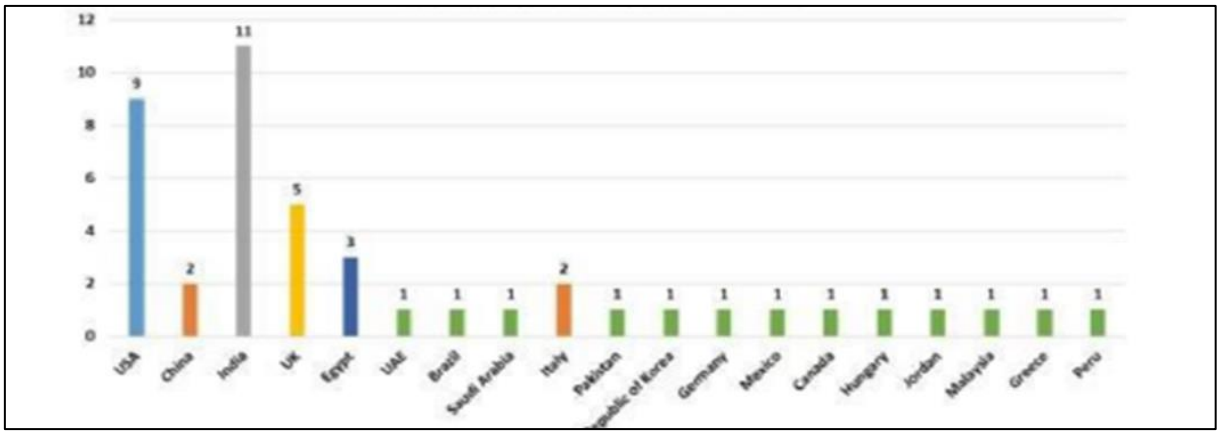


Figure 1.5: Machine learning for covid19 & number of Ml & mapping of ML publication As shown by [13].

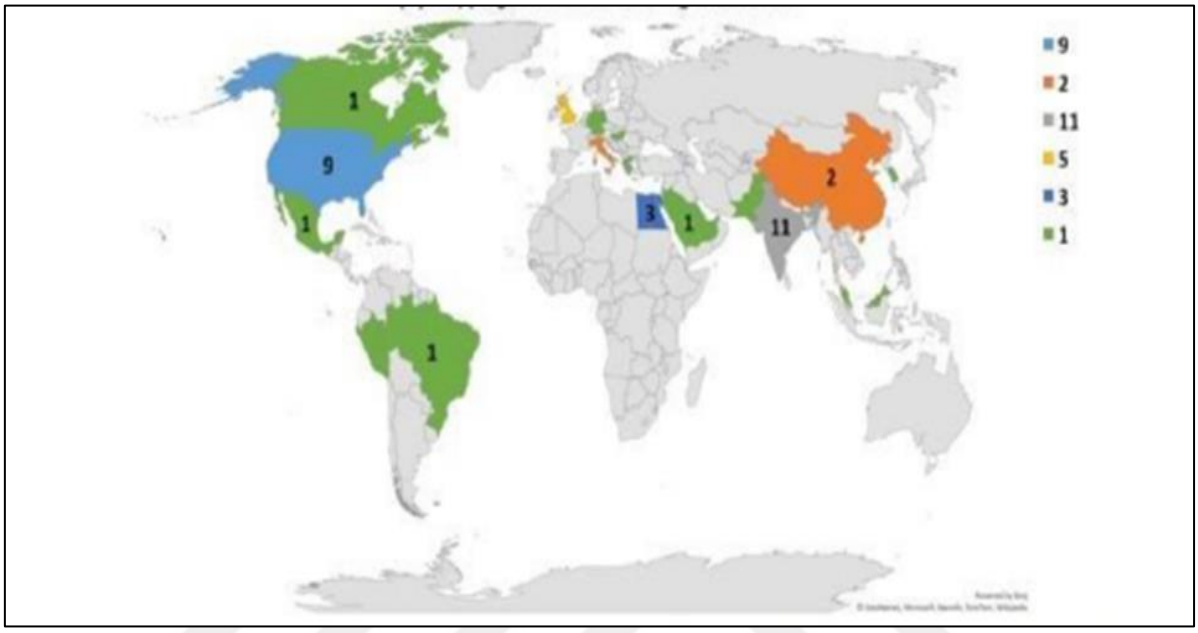


Figure 1.6: Machine learning for covid19 & number of Ml & mapping of ML publication as shown by [13].

agglomerative (bottom-up approach). The divisive clustering starts with all the elements in one cluster and seeks to divide them into smaller groupings until a halting condition is satisfied. In contrast, the agglomerative clustering treats each object as a separate cluster at first before combining them into larger clusters until a stopping condition is satisfied (Abasi, Khader, Al-Betar, Naim,"a). Separating text texts into different groups is the aim of partition algorithms. These methods often employ each cluster-centroid to generate similar information. The ultimate objective of these methods is to efficiently distribute a huge amount of data across several a variety of clusters with uniform data in each (Abasi et al., 1). Figure 7a presents a dataset comprising five clusters (clusters a, b, c, d, and e) as an illustration of agglomerative with divisive clustering algorithm, while Figure 7b presents a dataset having fourteen clusters as a frame for partition clustering. Due to its speed compared to hierarchical clustering methods, researchers prefer using the partition clustering strategy for categorizing data (Abasi et al., 2 " "19b) [12]. Additionally, evaluation findings show that, when utilizing clustering, partition clustering produces the best performance results using a variety of multi-

scale text collections A clustering algorithm is presented by Cui et al "to identify latent groups in COVID-19 patients. Mount Sinai System of The study made use of a cohort of far more nearly 6, ‘elderly patients who had failed a drug test for SARS-CoV-2 infection in New York, USA [9]. Applying the Kmeans algorithm and figuring out how many clusters are best was done using the elbow approach, which links the 18 physical structures and chronicity to patient diagnoses. Four clusters were found. Any patients with positive COVID-19 scans reported respiratory problems. The majority of COVID-19 patients have several chronic diseases and substantial comorbidities. Although comorbidity & chronicity have a strong association, age also has a major impact. Additionally, those who have a history of compromised immune systems are more vulnerable to major problems or medical illnesses in the systemic circulation when they are infected. problems, genitourinary, metabolic, or other illnesses. The identification of these four clusters has implications for the diagnosis of the illness, patient management, and eventually enhanced disease prevention. In order to create data-driven nation clusters for predicting the COVID-19 impact, Carrillo-Larco and CastilloCara use k-means [5]. The socioeconomic status, measures, and health care system coverage all had an impact.

between anticipated sickness frequency and air pollution. The proportion of fatalities, the frequency of confirmed COVID-19 cases, the case chance of dying, and the order in which the The researchers compare the clusters after analyzing all the national reported discoveries. The methodology was developed and used to categorize 155 nations into clusters. The best results are obtained by clustering comparable groupings of countries across 5 or 6 clusters using the model's three PCA (singular value decomposition (svd) parameters)[22]. The findings imply that using a system of five to seven clusters, countries may be grouped according to the official stats of COVID-19 cases. The program, however, was unable to classify nations according to the quantity of fatalities. Figure 8 provides an example of deep learning techniques for COVID-19.

1.4 DEEP LEARNING APPROACHES FOR COVID-19

The main deep learning (DL) algorithms used for COVID-19 include CNN, DNN, RNN, Phan, and a hybrid approach that combines ML and DL; this approach is covered in Section 3.3. Section 3.2.1, 3.2.2, 3.2.3, & 3.2.4, respectively, detail these algorithms. The COVID-19 DL publishing throughout various countries are shown in Figure 9.

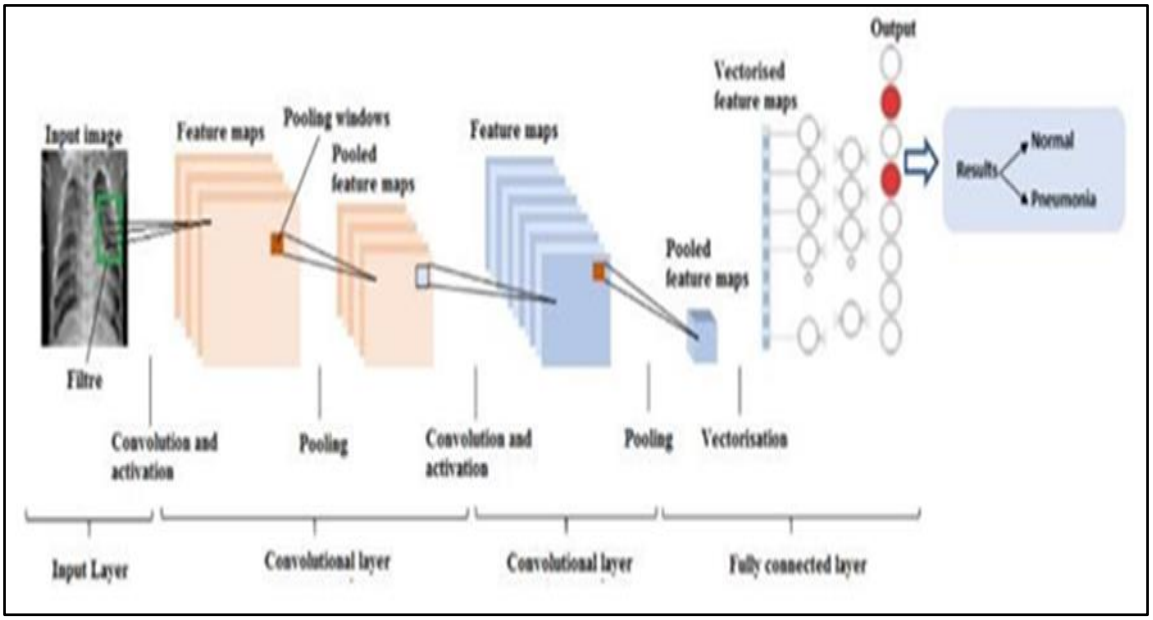


Figure 1.7: Supervised_learning_approches for_covid19.

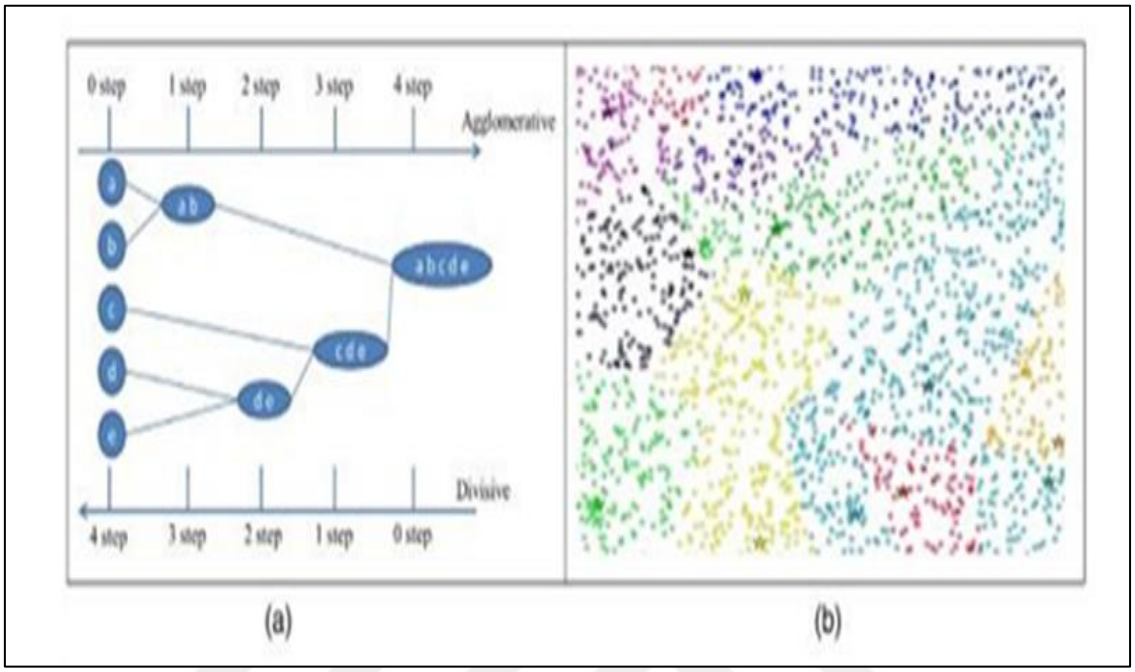


Figure 1.8: Application_of_agglomerative&devised_five_object

The DL algorithms are frequently used to treat COVID-19 illness from various angles. According to Rahman et al. ', edge computing is integrated into the 5G RAN. For COVID-19 edges, the local DL mechanism was spread, and the technique employed a three-phase reconciliation along with a global DL framework to govern the cloud environment. Their suggested DL model assisted experts in the COVID-19 area in enhancing important decision-making with semantics. Additionally, COVID-19 patients are categorised using a DL method known as DenseNet2 "1 (Jaiswal et al., ").[4] In order to diagnose coronavirus infection, a DL system is pretrained using this classification approach, which is based on the results of chest CT scans. The suggested technique was evaluated using the ImageNet database. Results from the categorization model were competitive. In a further investigation, CT scans were utilised to classify the Covid-19 patients using the entropy and DL mechanisms (Hasan et al., " ") [14]. CT images were initially segmented to reduce intensity variations. After that, a histogram thresholding approach was used to separate the CT images' backgrounds. Then, from each CT, characteristics were retrieved.

lung scan using a DL mechanism and a Q-deformed entropy technique. After that, a neural was used to categorise CT images using the retrieved characteristics. Infection with COVID-19 was predicted using DL methods. Which patient has COVID-19 infection was determined using laboratory data and DL. The suggested technique has an accuracy rate of 86.66 percent in identifying patients with COVID-19. Coronavirus, pneumonia, and normal classes can be employed for COVID-19 pictures (Togaçar, Ergen, & Cömert, ““).[22] Data is reorganised using the fuzzy colour approach, and structured data is layered. Then, the feature selection technique known as Social Mimic optimisation is used with the DL approaches (i.e., MobileNetV2 and SqueezeNet).[37] Finally, efficient features are classified using an SVM. The accuracy percentage of their suggested method was 99.27 percent. With patch-based CNN architectures, COVID-19 is diagnosed using CXR images. This method offers comprehensible saliency maps for COVID-19 diagnosis and patient triage. The decision tree classifier with DL was employed in another work to identify COVID-19 infection using CXR pictures. Three binary decision tree (BDT) classifiers trained using a CNN model were part of the proposed picture was classified as normal or abnormal using the first BDT. Using a second BDT with unusual photos, TB symptoms were identified. Using the third BDT of aberrant photos, COVID-19 was found. A 95 percent accuracy rate was attained. The model in Wang, Liu, et al " "as well as three other types of data: normal, viral pneumonia, and COVID-19. 96.1 percent accuracy was attained by the model. CNNs are employed to assess the degree of lung damage caused by COVID-19 infection. This is accomplished by using a radiologist's sickness severity assessment Their CNN can forecast and predict the response to treatment, as well as assess the severity of stage long illness. A type of neural network called a spatial transformer network is tailored for pictures of various COVID-19 patients at various stages Based on LUS pictures, this predicts the illness severity score and determines the COVID-19 infection's diagnosis. The COVID-19 infection and recovery phase are monitored using the sequence prediction of the DL technique (Heni, 'This will assess the effects of Bacille Calmette-Guerin on TB infection rates in certain groups. Hurt et al. give the DL procedures utilised for the COVID19 epidemic ‘. radiography of the chest taken from five

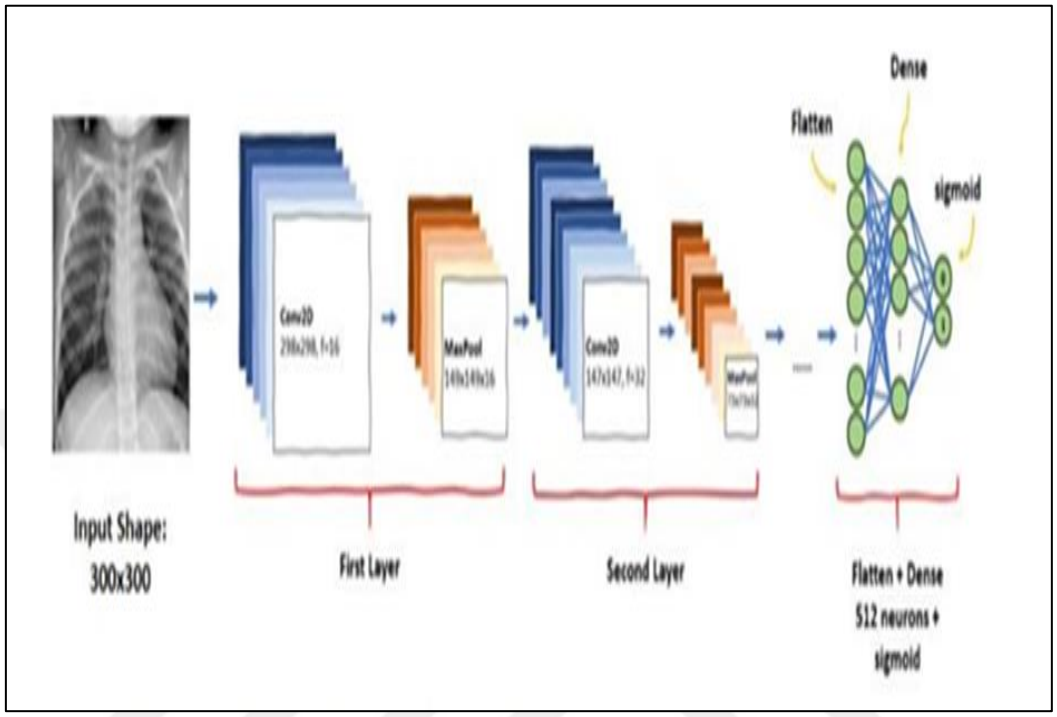


Figure 1.9: Deep_learning_steps for covid-19.

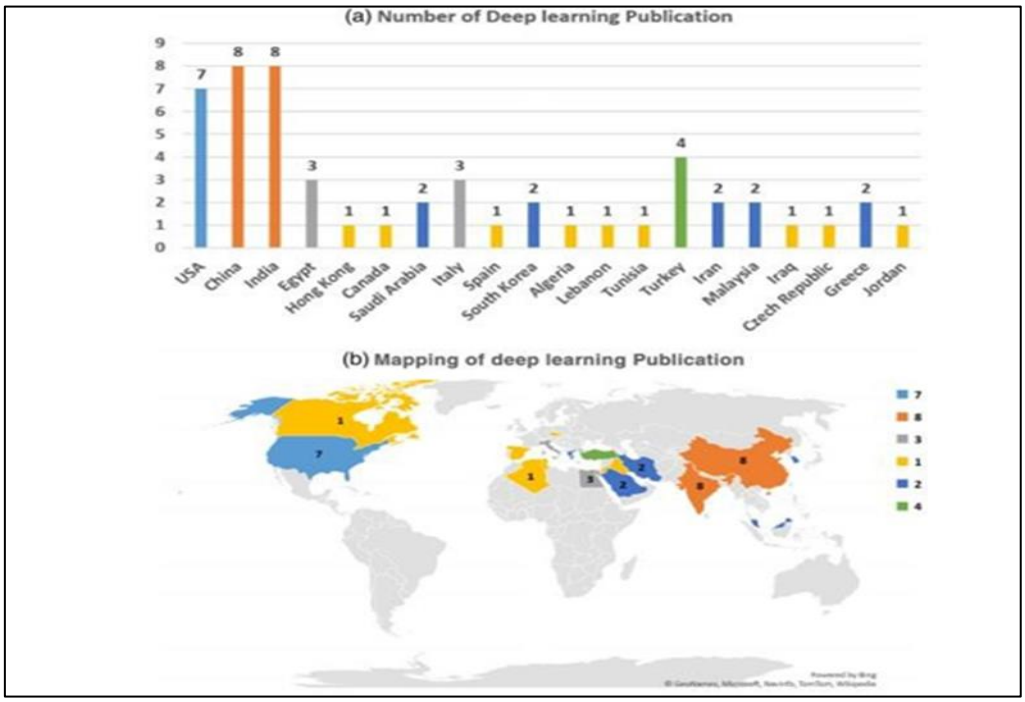


Figure 1.10: Deep learning publication in different counties for covid-19

Used were COVID-19 patients who had had treatment in the_US_and China. transfer learning makes use of previously learned model information to complete a new, related assignment.

EDs such intelligent medical equipment, webcams, drones, IoT, robots, etc. require the minimal retraining. Sufian et al " "give an overview of these methods to highlight how such infrastructure might aid in automation during an outbreak. These methods classify COVID-19 using on CT scans, x-ray of imaging, and viral genome sequences. Some studies utilizing transfer learning to predict COVID-19, reusing a model trained for one job for another after some adjustment to the new task. By analyzing CXR pictures, use transfer learning to identify namely DenseNet-121, ResNet5 [22], ResNet18, and SqueezeNet, were trained by the authors. According to the assessment findings, the majority of models are able to attain sensitivity rates of approximately 98 percent and specificity rates of about 9 ‘percent. Similar to this, employ CNNs together with a transfer learning strategy to improve the findings' accuracy when identifying COVID-19 from cxr of images. The evaluation's accuracy and loss rates are 96.3 percent (" ".151 binary cross-entropy). [15]et al " "present a recent assessment that outlines the state-of-the-art techniques based on DL and pulmonary medical imaging. They said that CT scans were recommended over CXRs for better prediction. A less expensive and secure real-time imaging method that can be utilised to diagnose COVID-19 is lung ultrasound (LUS) imaging. Using LUS pictures, Roy et al " "create a fully annotated dataset. To predict the illness's asperity score, the suggested dataset is combined using spatial transformer networks. They also present a brand-new uninorms-based method for aggregating video-level scores. For the purpose of forecasting COVID-19 imaging biomarker pixel-level segmentation, the suggested dataset is benchmarked against several DL models.

1.4.1 Convolutional Neural Network

According to Albawi et al., multi-layered AI neural networks are referred to as "deep learning" networks (ANN). Since it can manage a huge amount of data, it has emerged as one of the most useful tools in recent years and is widely used in literature. Recent performance of

conventional approaches was surpassed by interest in deeper secret layers in several domains, notably in pattern recognition. The most prevalent DNNs are CNNs. This linear mathematical operation between matrices is known as convolution. Convolutionary, nonlinear, pooling, and totally connected levels are just a few of the numerous levels that make up a CNN. Convolutionary

and completely linked layers need for the input of parameters. A CNN excels in machine learning tasks, particularly those requiring picture data, such the biggest image recognition information gathering, object recognition, and natural language processing. In this article, all of the elements, the main CNN problems, and how they work were outlined and covered. The factors that affect CNN's efficacy were also identified. Readers of this study were presumed to be well-versed in computers including artificial neural networks. Alazab et al "[2] created an AI method that employed a deep CNN to screen patients with COVID-19 illnesses using two real-world datasets acquired from Australia and Jordan. To test their procedure, 1 'X-ray images of real patients were used. Their technique has a 95–99 percent accuracy rate in studies for identifying COVID-19 patients. Using two forecasting approaches, namely the autoregressive model and LSTM, their method was also used to know the number of COVID-19 individuals, cases of recoveries, and fatality over the course of the next seven days. Statistics from Israel and Australia were utilized for training and testing. In Australia and Qatar, the average accuracy for the number of identified COVID-19, recovered, and deaths cases was 94.8 '% (Australia) and 88.43 per (Jordan) (Jordan) Picture biomarkers from COVID-19 X-ray pictures may be automatically extracted using deep CNN. The efficiency of extracted features in detecting COVID-19 biomarkers from X-rays was investigated by Apostolopoulos et al. '. ray pictures through a mobile network, according to CNN. The findings showed that 87.66 percent of the seven COVID-19 classes could be correctly classified, 99.18 percent could accurately distinguish COVID-19 from nonCOVID-19, or 97.36 and type is also known percent, respectively, could be said for sensitivity and specificity. Additionally, utilizing the VGG16 architecture, created a DL model to recognize COVID-19 from X-ray respiratory images The result showed a high specificity of over [1] ' 'percent and a high sensitivity. A DL method for COVID-19 detection from X-rays was

covered by Brunese et company. in " ". Three steps made up the procedure. A CXR was used first to check for pneumonia symptoms. Second, COVID-19 and pneumonia were separated. The location of COVID-19 in the X- Ray might be found. The precision value was 97, and the average detection time was 2.5 seconds, which made the collected data encouraging. Oh, et al " "introduced a gem CNN that can operate by altering a few parameters in order to get over the problem of only having a limited number of CXR images available for COVID-19. In order to identify the patch placements and conclusions from inference, this approach employed several majority votes. The method's accuracy level was 91.9 cent, which is close to the accuracy given by COVID-(92.4 Net) [41] percent. Sedik et al " "suggested two data-augmentation models to improve the process of learning for both CNN and the multilayer LSTM in the classification of medical images. (X-ray and CT). The objective was to improve the accuracy of COVID-19 detection. There were gains in terms of logistic loss, testing length, and detection accuracy when comparing the result obtained with that of Ml algorithms that do not employ data augmentation. Islam et al. developed an automated disease detection technique in. The technology helped clinicians recognize COVID 19 and delivered quick, precise data that showed a decrease with in risk of mortality. This study set out to provide a complete learning method that made use of CNN + LSTM models to automatically identify COVID-19 from CXR images. CNN was used to recover deep functions, while LSTM was used to identify the elements that were retrieved. On the apparatus were put 4575 X-ray pictures in total, including 1525 COVID-19 images. The results of the testing revealed that the proposed method obtained 99.4% accurate, 99.9% selectivity, 99.2percent sensibility, 99.3%, and then a 98.9% Fpr. An invasive and attentive online research model using CNN real-time data and flexible algorithms was described by Farooq and Bazaz [22]. Another difficulty in continually improving training data is modeling and recreating certain situations by changing model parameters over time. The key advantage of the method is the absence of the requirement to redesign the modeling each time a new collection is gathered, unlike conventional DL approaches in a dynamic training environment. The authors used the concept of validation to examine the effects of various disease response techniques. Without a doubt, Divide the population into poor (LR) and greater (HR) compartments to replicate a way for controlled natural immunization

utilizing risk dependent population split (PC), which separates based on risk factors). Sharing content et al " "employed lung X-rays and CT scans to find COVID-19. The researchers proposed a rapid, alternative technique of finding COVID-19 by examining patient X-rays using a Dg nCOVnet cnn[12] architecture. After applying many CNN layers and unbiased pieces of data, the recommended model had a high detection rate of 8 ‘per cent. Fast diagnostic methods can be helpful for treating people in labor-intensive situations and making it easier to track and stop the spread of global illnesses like COVID-19, claim Ardakani et al.

1 " "2 ‘CT images of 1 " "7 COVID-19 patients with viral and typical sinusitis and 86 non-COVID-19 people were obtained for this investigation. The distinction between COVID-19 virus and non-COVID-19 classes was made using ten well-known dcnns, including GoogleNet, AlexNet Fully convolutional, SqueezeNet, ResNet-18, Convolution neural network, ResNet-1 " "1, Xception, and ResNet-5 " "[41]. The two networks ResNet-1 " "1 and Xception delivered the best outcomes. ResNet-1 "1 differentiated between COVID-19 patients and non-COVID-19 cases with an AUC of ““.994, with an ofempirical percent specificity and a 1 ‘" "% sensitivity. A sensitivity of 98. "4 % and just a selectivity of 1 ‘‘percent were attained by the Xception model. However, each AUC's validation result was in line with the standards for sensitivity, specificity, and accuracyconclusions made by the radiologist. They arrived at the conclusion that Denitrification may be used as a replacement tool in the physician's office and as a very sensitive model for the forecasting of COVID-19 infections. A deep belief network and a recurrent max pooling layer which had been trained from scratch using a data sets were both incorporated into parallel architecture (COVID-DeepNet) outlined by Al-Waisy et al. (1). The system effectively identified patients with COVID-19 with a recognition average accuracy of 99.93%, susceptibility of 99.9 %, universality of 1 ‘"21 "%, correctness of 1 ‘"98 "%, andore of 1 ‘" "%. 99.93 percent with a Sem of percentage and mean square error of “\92.16 percent. Paluru et al. introduced Anam-Net, a compact CNN based on asymmetrical depth embedding (1) to identify anomalies in chest CT scans using COVID-19. The Anam- Account that was being suggested contains 7.8 times however many attributes as the modern Classifier (or its derivatives). Anam-Net is lightweight, which suggests mobile or commodity (point-of-care) systems. Anam-applicability Nets for point-of-

care platforms were demonstrated by using them in integrated systems like the Raspberry Pi 4, Cuda Jetpack Xavier, a mobile-based Android application (CovSeg). Monshi et al. (1) proposed based on Related to iot to maximize data augmentation and CNN hyperparameters for recognizing COVID-19 on CXRs in terms of training and validation. CovidXrayNet achieves 95.82 percent accuracy with only 3 ‘practice repetitions. Additionally, this adjustment raises the Visual Geometry precision of well-known along with Residual Neural System (ResNet-5 " “) [42]. Ismael and Sengür (1) suggested deep-learning-based approaches, involving deep feature extraction, okay of pretrained Classifiers, and end-to-end activation of a constructed CNN model, to categorize COVID-19 or normal (healthy) CXR images. Obtaining deep features,

The pretrained cnn Based models ResNet18, ResNet5 " ", Rnns, VGG16,[11] and Louis were applied. The dataset used consisted of 18 ‘COVID-19 images and 2 ‘ordinary (healthy) CXR images. The deep features gathered from the ResNet5 ‘framework and a Svms with only a logistic kernel function produced the results with the highest accuracy score. The achievement of the improved ResNet5 ‘system was 92.6 percent, whereas the side trained of the established Convolution layer produced a performance of 91.6 percent.

1.4.2 Neural Network in Depth

Since 2026 (Liu et al., 2 017) [45], DL techniques have been put out as a rapid learning algorithm for deep beliefs. due to their innate capacity to address the drawbacks of conventional on the neural networks. Deea lear techniques were also shown to be appropriate for natural language recommendation systems, pattern recognition, audio recognition, reading, and complete computer vision application testing. In Liu et al., certain DL designs and real-world setups were covered. The Boltzmann machine, neural networks, auto encoders, and deep belief networks are the four DL architectures that were presented. There was an obvious justification offered the option to choose from a range of potential issues in certain situations, such speech processing, pattern recognition, and computer vision. Six to nine hours are needed to screen for contamination using RT-PCR kits, according to Das et al. '. Because of RT-reduced PCR's sensitivity, it frequently produces inaccurate findings. This conundrum is

addressed by COVID-19's marker and diagnosis, which emphasise radiology using computed tomography (CT) and X-rays. Deep ML is used to create an automatic recognition method. For the purpose of optimising algorithm performance, Cxr are directly analysed utilising DL approaches. The techniques Using big datasets to teach network weights and small datasets to maximise weight using trained networks. In a study by 5372 patients having CT scans from seven cities or provinces were retrospectively examined. First, 41 " "6 CT scans of patients were used to pretrain the DL technique. The pictures were helpful in examining lung characteristics. 1266 patients from six towns or provinces—342 with pneumonia and 924 with were recruited according on the length of qualified and externally verified success of the DL framework. In the four prior rounds of validation, the DL technique was successful in distinguishing COVID-19[51] from other pneumonia) and viral pneumonia (patients. The DL approach was used to stratify the patients, and a significant

The length of time spent in the hospital varied. The DL gadget autonomously and without human input positioned itself in the middle of the questionable regions, displaying traits that are comparable to known radiologic results. By identifying potential high-risk patients and providing a quick and easy means to screen them for COVID-19, DL helps to maximise care services and avert catastrophic symptoms. According to Hu et althe commencement of a major sickness can be linked to passing away due to significant alveolar injury and gradual respiratory failure. The gold standard for clinical assessment is RT-PCR, but can cause erroneous negative consequences. In a pandemic crisis, the unavailability of RT-PCR services for testing may also delay the next clinical decision and care. For patients with COVID-19 DNN calculation for perceiving and classifying COVID-19 contamination from CT pictures was created in [8]., chest visualisation with CT is a potent testing and predicting technique. This study presented a weakly controlled deep learning approach for the detection and classification of COVID-19 infections using CT images. The method could be useful for reducing manual marks in CT scans and telling COVID-19 incidents apart from others. The authors asserted that their work can forecast the extensive use of tried-and-true technology in clinical trials based on the encouraging results. with a precision of 99.01% and AUC worth of 0.9972. [15] To adapt to size changes and the authors proposed a multi-scale learning system

in place of lesions. Function charts and classification layers were input into intermediate CNN representations (Conv3, Conv4, and Conv5), which are employed with 1 convolution. The outcomes from the suggested approach were contrasted with those from

For the gene selection of COVID-19, the optimization algorithm salp swarm algorithm was combined with Strategies that lower mortality rates have garnered a lot of interest, Altan and Karasu suggested a strategy that included a chaotic salp swarm algorithm. Using X-ray datasets, (CSSA) and CNN's technique may detect COVID-19 pneumonia infection. The major goal of employing CSSA is to treat coronavirus pneumonia as effectively as possible. The effectiveness of the suggested hybrid method has demonstrated a high rate of accuracy in detecting COVID-19 illness from X-ray pictures. An iteratively pruned DL model was presented in [22].

1.4.3 Network of Recurrent Neurons

Planning for newly contaminating and recovering COVID-19 cases is necessary while adjusting curfew limits and allocating resources for a developing disease.

progression. Zeroual et al " "examined five DL techniques to predict the frequency of brand-new COVID-19 infections and rehabilitate COVID-19 patients inside 17 days. Bidirectional LSTM, variational autoencoder (VAE), basic RNN, convolution units, and In this study, LSTM (BiL-STM) algorithms were contrasted. Information from Spain, Italy, exposed to separating and screening cycles to choose just the COVID-19 judgments dataset and barred other datasets like cover identification [18]. China, Italy, Australia, and the The Us was utilised in the study. The results showed that VAE was greater to the other approaches in terms of Reduction, MAE, Lr, Predicted values, EV, and RMSLE. Therapeutic survival analysis is used to calculate the probability of a clinical outcome. forecast the risk that COVID-19 patients will acquire a major condition by analyzing clinical data gathered at entrance. The proposed model was evaluated using information from three off really from COVID-19 datasets, which have various qualities, sizes and information types, for example, CT sweep or X-beam pictures [1]. China's Guangdong, Anhui, and Wuhan. An online

computer platform at the hospital used a developed model to prioritize patients upon admission COVID19-Triage, available at https://ai_@healthcare.tencent.com/en.html. Patients at a danger of developing a serious illness were provided access to the required care as promptly as was practical. In data to model the COVID-19 serologically in India, Arora en al " "employed DL models. LSTM variants were used in the RNN's method. Deep, deep, and bidirectional LSTM models were used to distinguish COVID-19 positive occurrences. Experimentally, the convolutional LSTM generated the poorest results for prediction errors, while the bi-directional LSTM provided the best. With four percent little as for each week forecasts, the bi-directional LSTM showed outstanding short-term prediction performance.

1.4.4 Networks of Generative Adversaries

Deep transfer learning and GAN were presented by the authors in Loey, Smarandache et al " "for the purpose of recognising COVID-19 instances from CXR pictures. A dataset containing 3 " "7 photos was used to test this technique. These pictures were divided into four categories: normal, COVID-19, bacterial pneumonia, and pneumonia virus. The Alexnet, Googlenet, and Restnet18 deep transfer models were employed. The identification of COVID-19 situations was proposed using three experimental settings. In the first scenario, there were four kinds of photographs, but in the second, there were only three. The third scenario includes two groups of photos. Each experimental scenario COVID-19 datasets, which have various qualities, sizes and information types, for example, CT sweep or X-beam pictures [1]. employed a different dataset that includes COVID-19 photos. In an experiment, the Googlenet served as a deep

The transfer model's accuracy scores were highest for the first experimental situation. A deep transfer model named Alexnet showed the greatest accuracy in the second treatment case, while Googlenet achieved the best accuracy in the last experimental instance. Multiple types et al " "employed DL techniques such GANs, intense learned machines, and LSTM to detect COVID-19 patients. They proposed an applied bioinformatics method that would incorporate different knowledge elements from multiple structured and unstructured sources to user-friendly platforms for medical experts and scientists. The primary benefit of these Artificial intelligence systems is the quickening of the COVID-19 diagnostic and care stages.

1.5 FOR COVID-19, A HYBRID TECHNIQUE COMBINING MACHINE _LEARNING AND DEEP LEARNING

An iteratively clipped DL model was proposed by Rajaraman, Siegelman, and others ‘. On X-ray imaging, the COVID-19- bronchial symptoms could be detected. In this method, an ImageNet-trained model and a customized CNN were used to learn the precise representations of COVID-19. Using the newly learned information, the patients were subsequently categorized into COVID-19-viral irregularities, normal, and instances of bacterial meningitis. With an effectiveness of 99. " "1 percentage and then an Accuracy value of “.9972, the proposed model performs remarkably well in trials. Using data with subpar labels, Rajaraman et Antani " "increased the quantity of training data used to locate COVID-19. The cause is that COVID-19 has characteristics in common with lung viral infections. The microorganisms in pathogenic microbial pneumonia with weak X-ray tags were included to the training set of data. The chosen photographs were used to train a CNN algorithm and compare the outcomes to a model that did not employ augmented input. The assessment process included six datasets. In terms of experimentally recognising COVID-19 symptoms as viral pneumonia, the poorly labelled data augmentation performed better than the baseline nonaugmentation. Similar to this, Zhu et al. "s authors used a deep-learning CNN to assess the severity of lung disease in COVID-19 patients. A real-world dataset of 131 CXRs from 84 patients with COVID-19 in US hospitals was used to evaluate their method. Divided was the dataset. 2 " "% of the data were utilised for testing, while the remaining 8 " "% were used for training. The proposed technique is evaluated using the correlation analysis and mean square error analysis. The outcomes were pleasing. However, the authors noted that a bigger dataset should be used to evaluate their method. The authors also suggested that their method may be applied in

evaluating the progression of illness and response to treatment in individuals with COVID-19, as well as the severity of lung illnesses. A DL-based AI system was created by Z. Li et al " "to identify COVID-19-infected lung areas and assess the severity and progression of the sickness using thick-section chest CT images. Their approach was evaluated using a dataset of 531 CT images from 2 " "4 COVID19 patients. The patient diagnosis reports and important

characteristics from the radiography result produced utilising the receiver were compared with the simulation results of their system. Cohen's kappa and the operational characteristic curve. Results demonstrated that their approach can segment lung infection areas and use CT images to treatment and diagnosis of COVID-19 patients. The following system disadvantages were listed by the authors: Their method is tested solely utilizing Diagnostic tests of COVID-19 patients, and it is appraised based on the differences in radiological biomarkers at the whole lung level. 1) Organ movement brought on by breathing and heartbeat can lead to erroneous diagnoses. A novel DNN algorithm for recognizing and categorizing COVID-19 infection in CT images was developed by Hu et al ‘. People with and without crowd pneumonia were categorized as COVID-19 patients using the indicated method. The severity of the illness and the location of lesions or infections were identified using this method, which was useful for therapy and triage. Using 6 ‘reported cases from the TCIA dataset, the approach is evaluated. Different sample counts were used. the algorithm's accuracy, precision, and area under the receiver operating characteristic curve all displayed encouraging results. DL-based software was used by Zhang et al " "to identify, pinpoint, and quantify COVID-19 pneumonia. The suggested AI algorithm made use of V-Net bottleneck structures and a 3D CNN. Using 246 ‘photos gathered from the Huoshenshan Hospital in Wuhan, China, their methodology was evaluated. Their approach produced outstanding results in experiments, which were helpful in the diagnosis and development of therapy programmes. In order to gather raw data, Rahman, Hossain, and colleagues (1) looked at nine distinct iterations of a DL algorithm designed to identify COVID-19 occurrences using medical IoT devices. The construction of adversarial cases for each kind and identification of the accountability of these algorithms were evaluated for the DL algorithms. The authors emphasised that DL algorithms take defensive models against adversarial perturbations into consideration, although these models are still susceptible to adversarial instances.

1.6 COVID-19 PUBLIC DATASETS

Finding appropriate datasets for the training, testing, and assessment of suggested methodologies is one of the most difficult jobs in the implementation of ML and DL systems.

Based on this, this part includes the most popular COVID-19 datasets, which include various data formats including CT scan and X-ray pictures and diverse sizes and features. These datasets are fully illustrated in Table 1 along with information on how to acquire them. These datasets were carefully chosen from the literature via a well-thought-out query that gathered relevant articles. In order to find the most pertinent studies, ML, DL, and COVID-19 diagnoses were combined as search terms. These publications were compiled using databases from reputable digital publishers (such as Elsevier, IEEE Xplore, MDPI, Springer, IOP Press, including Wiley). The articles that were returned by the search are put through filtering and screening procedures to reject other datasets, including mask detection, and to choose just the COVID-19 diagnoses dataset. Table 1 shows that writers have accomplished a great deal by offering various volumes and kinds of datasets that can greatly aid COVID-19 diagnosis. However, the vast majority of readily accessible datasets, particularly when it comes to medical imaging, are intended for diagnostic reasons. Given that data collecting based on medical pictures might be less difficult and time-consuming than the procedure employing COVID-19 RT-PCR samples, this scenario is comprehensible. The number of patients included determines the sample size, which differs among the datasets that are available. However, the majority of datasets are created by many examination trials, and only a small number of datasets are compiled from other existing datasets known as secondary datasets (i.e., after processing the original dataset). However, the diagnosis of COVID-19 has benefited considerably from secondary datasets. For example, large-scale COVID-19 CX-R image datasets have been used using the CLAHE algorithm as an improvement method for the resolution of medical pictures in the three main datasets (Al-Waisy et al., " "). Compared to the raw picture dataset, the classification rate in the proceed dataset rose by 8%.

1.7 MEASURES FOR COVID-19_EVALUATION USING DEEP_LEARNING AND MACHINE LEARNING

The effectiveness of DL and ML approaches used to identify COVID-19 is often assessed using a variety of measures. The most often used approach is

classification precision other measures are applied and completely detailed with mathematical formulations in this section. The assessment metrics used to assess the COVID-19 classification outcomes of the DL and ML algorithms are shown in Table 2. Each dataset employs one or more assessment measures, as stated in Table 1. The metrics employed to assess the COVID-19 diagnostic models are displayed in Figure 1 " ". Modern methods employ more than 18 assessment metrics. Clearly, sensitivity is the second most common assessment criterion utilised by researchers, after accuracy. Table 11 demonstrates the classifiers used in the COVID-19 ML and DL technique evaluation. The SVM classifier is the primary methodology employed by the majority of researchers in the COVID-19 area, as illustrated in Figure 11. According to the prevalence of these classifiers in research articles, boost, K-means, and logistic regression are in second place.

Table 1.1: COVID-19 datasets that are currently available to the public.

COVID-19 datasets that are currently available to the public						
Dataset	**Data type**	**Classification output**	**Type of dataset**	**Characteristics**	**Techniques**	**Achievement**
WDR	**outbreak**	**Statistcal report**	**a situation report for Covid-19**	**chinees**	**Bandy_opadhyay & Dutta_ " ";**	**Accuracy scan@ario 1 8 " ".6% 2 scenario2 8 5.2% scenario 3 99.9**
database for	**Laboratory finding**	**Diagnosis**	**annotated proteins**	**397 bacterial protective antigens (PAgs) and 178 _viral **	**In " ", Ong, Wong, et al.; Ong, Wang, et al.**	
Database (DNA sequence)	**Laboratory finding**	**Diagnosis**	**OVID-19 virus sequence**	**5 ' ' 'different genomic viral sequences (61.8 million bp)**	**Fila data set**	**72.7 percent for method 1, 68.7 percent**
Data from Hungary	**Outbreak**	**Statistical report**	**Time series data**	**statistical evaluations of**	**Guasti**	

Table 1.2: COVID-19 datasets that are currently available to the public "tables continued"

Pneumonia database in the first dataset,	**Helthy img**	**digam**	**x-ray img**	**Initial data set: 216 1,675 typical cases and COVID 19.**	**Luján-Garca, Yáez-Márquez, et al., " ";**	**ccuracy Dataset1: 96.**
genuine, original COVID-19)	Laboratory findings	Diagnosis	Signs and symptoms of	41 positive and	(Loey, Bandyopadhyay &	**Accuracy: 91.67%**
COVID-19	Outbreak	Statistical	Time series data	Reports on the incidence of COVID-19 in India	(R. S. Yadav, " "; Prakash et al., " "; Sujatha et al., " "; Shastri et al., " ") Case	**Fols about accurancy 98.9**
data from Facebook-Covid-19 (Sear et al., (" "	**Outbreak**	**Statistical**	**COVID-19 related comments**	**Data on fb**	**Raamkumar et al.,**	**Sensitivity 98.5**
Ultra of cxb data	**Medical image**	**Diagnosis**	**Image_ x-ray**	**Public cxr img**	**(Cohen et al._ " ")**	**Accuracy is 91.9 % ‘**
NIH	**Medical image**	**Diagnosis**	**Image_ x-ray**	**5.6 " "6 images and labels altogether are culled from the NIH Chest X-ray collection.**	**)Barati et al., Filice et al., Tang**	**Accuracy 9 " ".13%**

Table 1.2: COVID-19 datasets that are currently available to the public "tables continued"

Italian Covid-19 Lung Ultrasound Project, or ICLUS (Wang, Peng, et al., 2 " "17b(	**Medical video**	**Diagnosis**	**Lung ultrasound video**	**a database of ultrasound images with the potential to be utilized for classifying patient phases**	**Roy et al. ; Che et**	**Accuracy convex: 84%, linear:94%**
Zaho " ")(	**Medicalphoto**	**Diagnosis**	**CT images**	**Tongji Hospital, Wuhan, China COVID-19 data patients between January and April. Total of 349 CT images that included clinical**	**(Yang et al., " "; Afshar et al., " "; Al-Karawi et al., " "; Roberts et al., " "; X. He et al., " ")**	**Accuracy 95.37%, Sensitivity 95.99%, Specificity 94.76%**

1.8 AIMS AND OBJECTIVES

a. To study the overview, advantages and application of deep learning for the smart wireless neural network.

b. Study and analysis of cove d disease & their preventive measures.

c. The AL calculation for the purpose of identifying chest x-rays for those suffering from respiratory diseases has proven to be positive for Coved from other x-rays. Our goal is to verify this progress.

d. Enhancing the detection of X-ray examples by computing the interpretability of artificial intelligence.

1.9 CONTRIBUTION

We presented an in-depth study of deep learning theories and

their use in addition to revealing the state through X-ray images, by presenting a set of previous research studies to see what researchers have found in this field. This research presented a practical study of corona detection using convolutional neural networks through a large set of experiments for the purpose of determining the dest learning are and the optimal number of layers for corona detection through x-ray images.

1.10 PLANNING OF CHAPATERS

Chapter 1: In this chapter, a high-level overview of deep learning, future wireless networks, and networking applications built using deep learning was studied in detailed, which helps to define the scope and contributions of this project.

Chapter 2: Since deep learning techniques are relatively new in the wireless neural networking community, we provide a basic deep learning background in this chapter. The literature regarding the deep learning in neural network was reviewed with details.

Chapter 3: This chapter includes the methodology of mechanical learning used in the coved disease analysis.

Chapter 4: The result of the mechanical learning used in the coved disease analysis using chest X-ray images.

Chapter 5: will present the final chapter presents a summary and conclusion of the thesis and in the end there will be References.

2. LITERATURE REVIEW

Using data from the literature, we discuss the most recent uses of AI-based techniques for diagnosing, categorizing, and forecasting the spread of COVID-19 in this chapter. The epidemiological studies feature of COVID-19 are not the main focus of this review. Additionally, the current evaluation does not focus on COVID-19 therapy or vaccine development; it is left for future studies.

2.1 COVID-19

Coronavirus syndrome (COVID-19) is a virus infection brought on by coronavirus 2 that causes severe acute respiratory illness (SARs-CoV-2). It made its debut in Dec. in Wuhan, China, before spreading over the world and killing a large number of people. Health systems all around the world have been overburdened since the Corona pandemic began in 2 " "19 till now, and the need for critical care units and gas cylinders is growing due to the vast amount of overall cases in Covid-19, hour by hour. Therefore, it is essential to foresee this sickness in order to restore global health and safety. There are a few difficulties in predicting this illness, though. The primary difficulties in treating Covid-19 patients include tracking down infected individuals, the lack of a cure, the ease with which the disease can be spread, and the lengthy incubation period. Therefore, it is essential to anticipate COVID-19 accurately.

Table 1 compares the COVID-19 pandemic to others that have been documented throughout human history.

Table 2.1: Some pandemics in human history.

Pandemics	Number of Deaths (in Millions)
Rough Flu	1
Spain Flu	4 " "-5 " "
u 4 " "-5 'A	1.1
Third that Plague useful	12
Acquired immune of deficiency syndrome _(AIDS)	25-35
SwiFlu	" ".2
COVID-	2.78

2.2 DEEP LEARNING TO DETECT COVID-19:

Finding the right solutions to manage COVID-19 has been a major concern recently, especially with the volume of data linked to COVID-19 expanding and changing on a regular basis.

Ai technology (AI) can assist in controlling COVID-19 in these circumstances. Artificial intelligence (AI) techniques such as computer science (ML) and pattern recognition (DL) can improve motivation and reduce the need for human resources, logistics, and precise timing. It's also important to note that ML and DL are employed in several medical systems to find patterns in data sets. Doctors can draw patterns from massive data volumes using ML and DL, and they can swiftly advance as fresh data becomes available. Add to AI can ensure the accuracy of COVID-19 data-driven diagnosis, patient identification, and prediction of viral transmission.

In general, AI assists policymakers in managing a crisis, planning for treatment tactics, and optimizing problem-solving methods in addition to huge data. Therefore, robotics play a role in the initial symptoms and patient treatment in an AI-based robotic COVID-19 detection system without involving humans, including medical imaging but instead image processing, which helps reduce the rate of illness onset and prevent the virus from spreading from patients to radiologists and medical staff. In addition, AI-based solutions can be helpful. monitor and track the COVID-19 patients, which reduce the virus spread.

2.3 APPLICATION DEEP LEARNING TO DETECT COVID-19:

2.3.1 ML Regressors for Forecasting COVID-19:

Several studies concentrated on applying ML methods to various COVID-19 features. The authors conducted template analysis on the COVID-19 instances using a variety of mathematical frameworks.

For instance, the logistic, following approaches, and Gompertz models were used in (Pham, et al., "). Although Jia et alcoefficient.'s of determination analysis from ' or Rsquared values R2(C) and R2(N) represented the fitting goodness of the cumulative confirmed and the new confirmed cases. Moreover, parameter was used to indicate the fitting goodness of the cumulative number of death cases. However, those three models could predict the epidemic more accurately in the later stages. In Wuhan city of China, the logistic model showed better performance than the other two in

Three tables, one for each of cases reported, deceased cases, and recovered cases, were utilized in the dataset for Johns Hopkins University research studies. Over forecasting the instances for the following 1 'days, this study also uses a number of ML and DL techniques, including Logistic Regression (SVR), Polynomial Reconstruction (PR), Learning Algorithm (DNN), and Long/Short Term. Additionally, the authors of the (Benvenuto, " ") study used the autoregressive moving averaging (ARIMA) model to calculate the number of verified COVID-19 cases.

Work on machine learning for forecasting and forecasting COVID-19 instances is shown in Table 2.

Table 2.2: ML for forecasting and prediction COVID-19 cases.

Refs.	Adopted Technique	Prediction Result	Objectives	Dataset
[15]	Logistic model	In Mainland China, R^2(C): 0.9993, R^2(N): 0.9183; In Wuhan, R^2(C): 0.9991, R^2(N): 0.8124.	Prediction of epidemic results of COVID-19.	-
	Gompertz model	In Mainland China, R^2(C): 0.9934, R^2(N): 0.3372; In Wuhan, R^2(C): 0.999, R^2(N): 0.813.		-
	Bertalanffy model	In Mainland China, R^2(C): 0.9993, R^2(N): 0.895; In Wuhan, R^2(C): 0.9989, R^2(N): 0.8105.		-
	Logistic model	In Mainland China, R^2(DC): 0.9995; In Wuhan, R^2(DC): 0.9993	Predicting the COVID-19 death toll.	-
	Gompertz model	In Mainland China, R^2(DC): 0.9997; In Wuhan, R^2(DC): 0.9996		-
	Bertalanffy model	In Mainland China, R^2(DC): 0.9991; In Wuhan, R^2(DC): 0.9995		-
[17]	SVR	RMSE for confirmed cases: 27456.47, RMSE for death cases: 1360.47, RMSE for recovered cases: 16762.15	Prediction of future reachability (next 10 days) of the COVID-2019 across the nations.	[16]
[17]	PR	RMSE for confirmed cases: 455.92, RMSE for death cases: 117.94, RMSE for recovered cases: 809.71		[16]
[18]	ARIMA model of order (1,0,3)	Forecast value of COVID-19 incidence: at 11 February, 2020: 2070.66, at 12 February, 2020: 2418.47.	Evaluate the incidence of new confirmed cases of COVID-2019 in the next 2 days.	[16]
[19]	ARIMA model of order (0,1,0)	RMSE (Prediction) for USA: 3963.44, RMSE (Prediction) for Italy: 1258.69, RMSE (Prediction) for China: 59.65	-	[16]
[20]	ARIMA model of order (1,1,0) for confirmed and death cases	R^2(Confirmed Case- Italy): 0.761, R^2(Death Case- Italy): 0.927	Forecasting number of daily confirmed cases and deaths.	[21]
	ARIMA model of order (2,1,0) for confirmed cases and (0,1,0) for death cases	R^2(Confirmed Case- Turkey): 0.817, R^2(Death Case- Turkey): 0.714		
	ARIMA model of order (2,1,1) for confirmed cases and (1,1,2) for death cases	R^2(Confirmed Case- Spain): 0.950, R^2(Death Case- Spain): 0.958		
[22]	ARIMA model of order (2,1,2) for Italy, (2,1,0) for China, (1,0,0) for South Korea, (2,3,0) for Iran.	Forecast for 17 days (from 5 March until 21th of March) with 95% confidence interval.	Forecast of confirmed cases of COVID-19 for 17 days.	[16]
[23]	ARIMA(1,2,0) for confirmed cases, and ARIMA(3,2,0) for recovered cases	For confirmed cases, ME=17.84, RMSE=514.74, MAE=324.46, MPE=4.26, MAPE=6.25. For recovered cases, ME=80.80, RMSE=186.85, MAE=112.27, MPE=10.57, MAPE=15.60.	-	[28]
[24]	Logistic Model	Error degrees of freedom: 52 Root Mean Squared Error: 1.9e+03 R-Squared: 0.997, Adjusted R-Squared 0.997 F-statistic vs. zero model: 1.61e+04, p-value = 4.02e-77	Estimated logistic model parameters for China (data up to 11.Mar 2020).	[29]
[27]	Hybrid Wavelet ARIMA Model	MAE($\times 10^3$): 0.464 (Italy), 0.136 (Spain), 1.341 (USA). RMSE($\times 10^3$): 0.630 (Italy), 0.170 (Spain), 1.974 (USA). R-Squared: 0.9985 (Italy), 0.9996 (Spain), 0.9888 (USA).	Prediction of death cases in the next one-month one-month beyond the data sample end date.	[30]
	ARIMA Model	MAE($\times 10^3$): 1.243 (Italy), 0.693 (Spain), 2.822 (USA). RMSE($\times 10^3$): 1.601 (Italy), 0.884 (Spain), 4.103 (USA). R-Squared: 0.9944 (Italy), 0.9089 (Spain), 0.9806 (USA).		

2.3.2 ML Classifiers for COVID-19

Researchers utilized a number of machine learning (ML) forecasting model on a dataset of 5644 probable COVID-19 patients in research that took place at the Israelita Albert Einstein Hospital in Brazil. The classification accuracy of Fully - connected layers (MLP), Svm classifiers, and logistic regression for patients with COVID-19 was found to be 91%.

Additionally, ML classifiers were used on 253 samples in a study conducted in Lanzhou Chest Hospital and Gansu National Hospital in China. Researchers took samples from 169 suspected patients; 1 " "5 of them were positive for COVID-19 and were regarded as positive samples. Additional samples were taken for TB, common pneumonia, and lung cancer. and considered as negative samples. Random forest algorithm achieved an accuracy of 97.95%.

Table 3 shows the list of research work and the associated ML classifiers for COVID-19.

Table 2.3: Some existing ML methods/classifier with its results.

Ref.	Method	Validation Method	Data Types	Sample Size	No. of Patients	Results
[31]	Classification (Multilayer perceptron)	Holdout	Clinical	5644 samples with 111 attributes. The processed dataset has 1091 records and 61 attributes.	5644	Accuracy: 93.13%, Recall: 93%, Precision: 93%
[32]	Random forest	Cross-validation	Demographics, Clinical	Total 253 samples from 169 patients. Clinical blood test of 49 patients where 24 Covid-19 patients.	253, 169, 49, 24	Accuracy: 95.95% Specificity: 96.95%
[33]	Support Vector Machine	Holdout	Clinical, Demographics, laboratory features	336 COVID-19 patients where 26 critical cases.	336, 220	Accuracy: 77.5%, AUC: 99%, Specificity: 78.4%
[34]	XGBoost classifier	Cross-validation	Blood samples of 75 features, Clinical	485 samples.	485	Accuracy: 90%

2.3.3 Application of DL for Diagong of COVID-19:

Medical imaging are the primary technique for the detection of COVID-19. Researchers are examining the use of DL techniques on X-ray and CT scan pictures to diagnose COVID-19 patients, though.

X-ray radiation may go through several things. So, it is possible to employ X-rays to capture photographs of the inside architecture of the human. While ultrasound pictures are produced as real-time movies utilizing sound waves, CT images are more informative than their counterpart X-ray images. In addition to all of these, several studies use other imaging modalities to identify COVID-19 patients. The quantity of studies on various modalities for the diagnosis - COVID-19 is displayed in Table 4 below. Additionally, table 5 displays many research projects on the use of DL on various photos.

Additionally, we present a framework supporting COVID-19 diagnostic procedures in Figure 2. First, the incoming database is examined, and then doctors choose an imaging approach. The use of X-ray, CT, or ultrasound imaging follows. Based processing and optimization are then carried out after that. Results are approximated and published at the end.

Table 2.4: Distribution of studies for various medical imaging modalities.

Image Types	References Studies	No. of Studies
X-ray	[35, 38-53]	17
CT	[36, 37, 54-69]	18
Ultrasound	[70, 71]	2
Multimodalities	[72, 73]	2

Table 2.5: Summary of features, applications, and limitations of different modalities for COVID-19 diagnosis.

Imaging Modalities	Important Features and Applications	Limitations	Refs.
X-ray	Data augmentation is used	Not appropriate for multiclass problems.	[50]
	DL (neural networks) is considered	Only a limited number of X-ray images are taken into consideration.	[43, 45]
	SqueezeNet is used	Not validated for images except for X-ray images.	[35]
	Using data augmentation with auxiliary classifier GAN	1. Small datasets are considered, and image quality is not good.	[48]
CT	DL is considered	The method is not validated on a clinical study.	[68]
	Using a CAD-based scheme	This is only suitable as an adjuvant method for CT scan images.	[59]
	MADE algorithm is used for optimising the attributes of DBM algorithm	MADE-DBM is validated on chest CT datasets.	[36]
	A residual learning strategy is used to screen CT images	Not appropriate for detecting patients with early COVID-19 infection.	[62]
	Applying Inf-Net to chest CT images.	Accuracy is slightly lower when non-infected regions of the images are taken into consideration. An additional classifier to classify infected and non-infected regions is used to improve the overall accuracy.	[37]
Ultrasound	Using DL for analysing LUS images	The dataset has some limitations, for example, not being very large, collected from only a few hospitals in Italy, and the ultrasound was applied to only critically ill patients. There was also poor image quality in some cases.	[70]
Multimodalities	COVID-19 diagnosis using both X-ray and CT Images	The combined use of X-ray and CT scan images may not be efficient in real-time practical applications.	[72]
	Combination of ML, DL and data augmentation for COVID-19 detection	Feasible only for limited datasets.	[73]

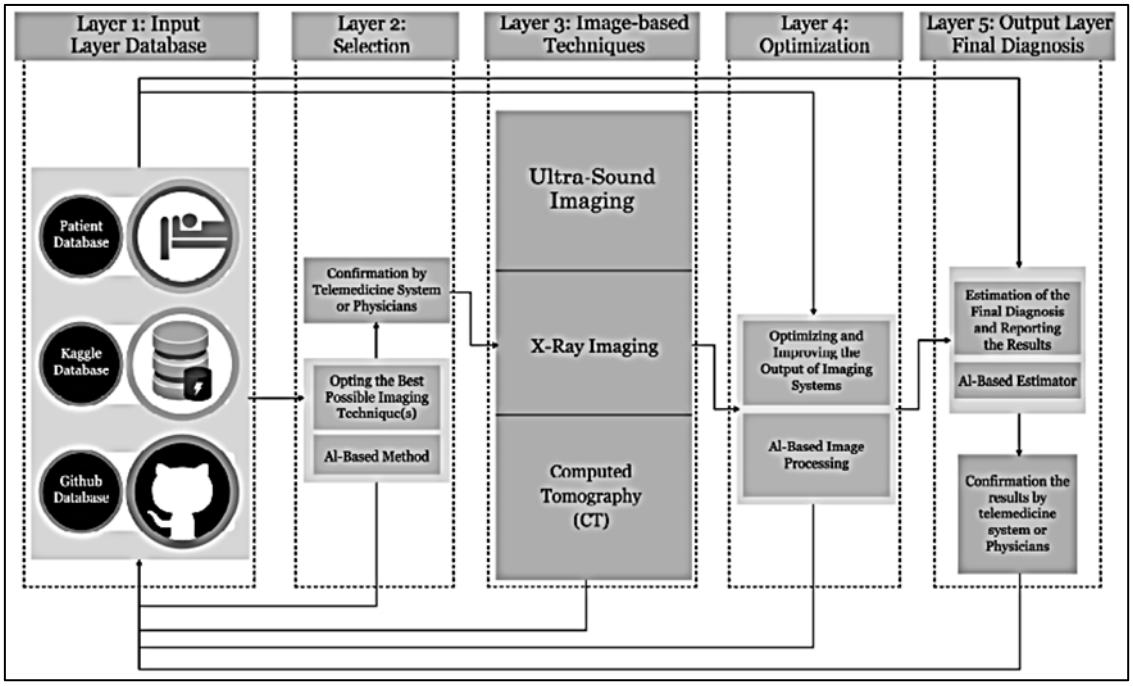


Figure 2.1: Framework for COVID-19 diagnosis.

2.4 CHALLENGE OF DL TO PREDICT TO MANAGE

Covid-19 As during COVID-19 pandemic, nations took a number of precautions to stop the virus's spread, including lockdowns, social exclusion, and the creation of legislation and standards for residents, company owners, medical professionals, scientists, and others.

The management of COVID-19 by ML - DL approaches need large-scale, accurate data, especially data for model training. The illness prediction process is complicated by certain datasets' inaccurate and insufficient labeling. On the other side, having too much data might make it challenging to extract the important information. Real-world medical photos frequently contain numerous forms of aberrations, necessitating diverse image processing techniques. The use of AI approaches is hampered by the existence of this incorrect information. Furthermore, there is no standard dataset for comparing the effectiveness of various methods.

It is crucial to establish solid relationships with information and communication technology (ICT) or computer science professionals or virology of medical science experts in order to maximize AI's efficiency in combating COVID-19. However, creating a strong platform for collaboration among specialists from many fields is a difficult challenge. Another difficulty is the problem of data protection, which necessitates the protection of individual freedoms and privacy. Patients may experience discomfort if artificial intelligence is used to analyze patient data. Additionally, getting access to this information from clinics and hospitals can be challenging.

3. RESEARCH METHODOLOGY

Tools and Libraries Used in This Research:

A. Python

Python is a high-level, dynamic, general-purpose, and interpreted programming language. It encourages the creation of applications using an object-oriented programming methodology. It offers several high-level data structures and is straightforward and simple to learn.

B. Jupyter Notebook App

The client-server Jupyter Notebook program enables editing and execution of notebook papers over a web browser. As detailed in this page, the Jupyter Notebook program may be used locally on a desktop without Internet connection or it can be deployed on a cloud host and accessed online. The Jupyter Notebook software has a "Dashboard" (Notebook Dashboard), a "control panel" that shows local files and enables opening and closing of notebook papers, in addition to reading, editing, and playing notebook documents

C. Pandas

Added on top of the Scripting language, Pandas is an open free data analysis / processing tool that is quick, strong, versatile, and simple to use.

D. Tensor Flow

A complete open-source computer vision platform is called Tensor Flow. It has an extensive and adaptable toolsets, libraries, and assistance programs that enable developers to quickly create and deploy network learning-powered applications and researchers to advance the most recent machine learning technology

E. k eras

K eras is just an API made for people, not computers. By offering consistent, straightforward APIs, minimizing the amount of user steps necessary for typical use cases, and supplying clear, actionable error signals, Keras adheres to best practices for reducing cognitive load. Additionally, it offers developers substantial documentation and instructions.

F. Matplotlib Visualization

Python's Matplotlib toolkit provides a complete tool for building static, animated, and active visualizations. The difficult tasks are made feasible with Matplotlib, as well as the simple things.

3.1 DATA SET DESCRIPTION

Chest X-ray images of healthy vs. patients with pneumonia (Corona) infection are included in a data set from Kaggle, along with a few other categories like Salmonella (acute respiratory syndrome (sars), Streptococcus, and Autoimmune disease (acute respiratory distress syndrome). The data set has a size of more than one gigabyte and is divided into two volumes, the first of which contains 53 " "9 training examples and the second of which contains 6.

This dataset also contains two CSV files:

A. The first one, Chest_xray_Corona_Metadata.csv, contains 6 columns:

a. X_ray_image_name Image name of the X ray, few image name contains Virus or Bacteria tag.

b. Labe It indicates X Ray - Normal or Healthy and Person affected with Pneumonia.

c. Dataset_type X Ray Image belongs to train or test set, it helpful for Machine learning researchers to train machine learning.

d. Label_2_Virus_category Label holds the information about the pneumonia is due to Virus, Bacteria or ARDS

e. Label_1_Virus_category Label holds the information about the pneumonia (Virus, Bacteria or ARDS) & detailed classification

Figure 1 shows screenshot of the first file:

,X_ray_image_name,Label,Dataset_type,Label_2_Virus_category,Label_1_Virus_category
0,IM-0128-0001.jpeg,Normal,TRAIN,,
1,IM-0127-0001.jpeg,Normal,TRAIN,,
2,IM-0125-0001.jpeg,Normal,TRAIN,,
3,IM-0122-0001.jpeg,Normal,TRAIN,,
4,IM-0119-0001.jpeg,Normal,TRAIN,,
5,IM-0117-0001.jpeg,Normal,TRAIN,,
6,IM-0115-0001.jpeg,Normal,TRAIN,,
7,IM-0189-0001.jpeg,Normal,TRAIN,,
8,IM-0187-0001.jpeg,Normal,TRAIN,,
9,IM-0185-0001.jpeg,Normal,TRAIN,,
10,IM-0183-0001.jpeg,Normal,TRAIN,,
11,IM-0182-0001.jpeg,Normal,TRAIN,,
12,IM-0180-0001.jpeg,Normal,TRAIN,,
13,IM-0178-0001.jpeg,Normal,TRAIN,,
14,IM-0177-0001.jpeg,Normal,TRAIN,,
15,IM-0176-0001.jpeg,Normal,TRAIN,,
16,IM-0172-0001.jpeg,Normal,TRAIN,,
17,IM-0170-0001.jpeg,Normal,TRAIN,,
18,IM-0168-0001.jpeg,Normal,TRAIN,,
19,IM-0166-0001.jpeg,Normal,TRAIN,,
20,IM-0164-0001.jpeg,Normal,TRAIN,,
21,IM-0162-0001.jpeg,Normal,TRAIN,,
22,IM-0160-0001.jpeg,Normal,TRAIN,,
23,IM-0158-0001.jpeg,Normal,TRAIN,,
24,IM-0156-0001.jpeg,Normal,TRAIN,,
25,IM-0154-0001.jpeg,Normal,TRAIN,,
26,IM-0152-0001.jpeg,Normal,TRAIN,,

Figure 3.1: Chest_xray_Corona_Metadata.

B. Second file Chest_xray_Corona_dataset_Summary.csv Which contains the summary information about the Labels (Level 1, Level 2 and Level 3) and number of images, below is an explanation of the columns:

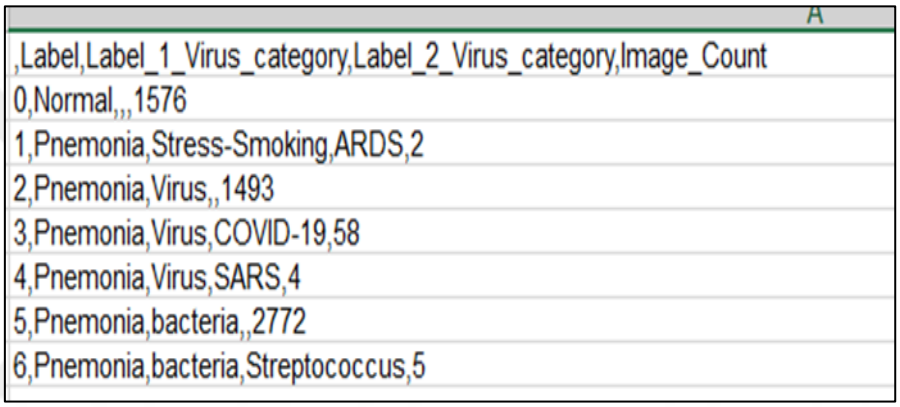
a. Label Normal or Healthy Individual or Pneumonia affected person.

b. Label_1_Virus_category Label_1 holds category as Virus, Bacteria and ARTD.

c. Label_2_Virus_category Label_2 holds Virus, Bacteria detailed category COVID19, SARS, etc.

d. Image_Count Chest X Ray Image count.

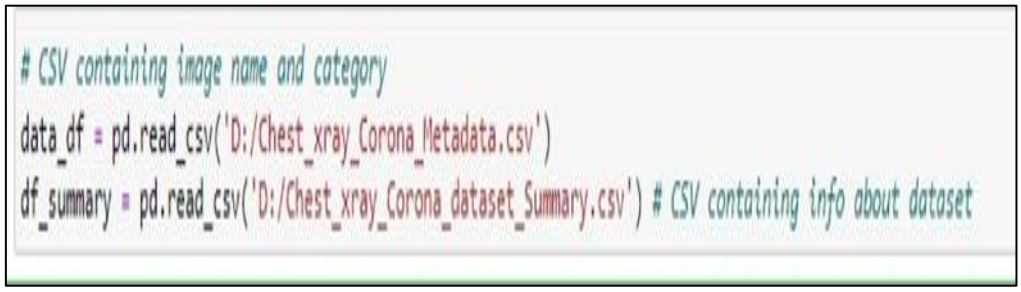
Figure 2 shows screen shot of the second csv file.



	A
,Label,Label_1_Virus_category,Label_2_Virus_category,Image_Count	
0,Normal,,,1576	
1,Pnemonia,Stress-Smoking,ARDS,2	
2,Pnemonia,Virus,,1493	
3,Pnemonia,Virus,COVID-19,58	
4,Pnemonia,Virus,SARS,4	
5,Pnemonia,bacteria,,2772	
6,Pnemonia,bacteria,Streptococcus,5	

Figure 3.2: Chest_xray_Corona_dataset_Summary.

First, the CSV files are loaded as shown in the figure.



```
# CSV containing image name and category
data_df = pd.read_csv('D:/Chest_xray_Corona_Metadata.csv')
df_summary = pd.read_csv('D:/Chest_xray_Corona_dataset_Summary.csv') # CSV containing info about dataset
```

Figure 3.3: Loading csv files.

Then the data is divided for training and testing according to the Dataset_type column as shown in figure 4.

Figure 3.4: divide data by column dataset_type.

Then the data exploration phase begins through a set of operations described below:

Figure 5 shows the review of the first twenty lines of the Chest_xray_Corona_Metadata.csv file.

```
In [6]: data_df.head(20)
```

Out[6]:

	Unnamed: 0	X_ray_image_name	Label	Dataset_type	Label_2_Virus_category	Label_1_Virus_category
0	0	IM-0128-0001.jpeg	Normal	TRAIN	NaN	NaN
1	1	IM-0127-0001.jpeg	Normal	TRAIN	NaN	NaN
2	2	IM-0125-0001.jpeg	Normal	TRAIN	NaN	NaN
3	3	IM-0122-0001.jpeg	Normal	TRAIN	NaN	NaN
4	4	IM-0119-0001.jpeg	Normal	TRAIN	NaN	NaN
5	5	IM-0117-0001.jpeg	Normal	TRAIN	NaN	NaN
6	6	IM-0115-0001.jpeg	Normal	TRAIN	NaN	NaN
7	7	IM-0189-0001.jpeg	Normal	TRAIN	NaN	NaN
8	8	IM-0187-0001.jpeg	Normal	TRAIN	NaN	NaN
9	9	IM-0185-0001.jpeg	Normal	TRAIN	NaN	NaN
10	10	IM-0183-0001.jpeg	Normal	TRAIN	NaN	NaN
11	11	IM-0182-0001.jpeg	Normal	TRAIN	NaN	NaN
12	12	IM-0180-0001.jpeg	Normal	TRAIN	NaN	NaN
13	13	IM-0178-0001.jpeg	Normal	TRAIN	NaN	NaN
14	14	IM-0177-0001.jpeg	Normal	TRAIN	NaN	NaN
15	15	IM-0176-0001.jpeg	Normal	TRAIN	NaN	NaN
16	16	IM-0172-0001.jpeg	Normal	TRAIN	NaN	NaN
17	17	IM-0170-0001.jpeg	Normal	TRAIN	NaN	NaN
18	18	IM-0168-0001.jpeg	Normal	TRAIN	NaN	NaN
19	19	IM-0166-0001.jpeg	Normal	TRAIN	NaN	NaN

Figure 3.5: the first 2 ‘rows of Chest_xray_Corona_Metadata.csv file.

Figure 5 shows the review of the first five lines of the Chest_xray_Corona_dataset_Summary.csv file.

```
In [9]: df_summary.head()
```

Out[9]:

	Unnamed: 0	Label	Label_1_Virus_category	Label_2_Virus_category	Image_Count
0	0	Normal	NaN	NaN	1576
1	1	Pnemonia	Stress-Smoking	ARDS	2
2	2	Pnemonia	Virus	NaN	1493
3	3	Pnemonia	Virus	COVID-19	58
4	4	Pnemonia	Virus	SARS	4

Figure 3.6: The first five lines of Chest_xray_Corona_dataset_Summary.csv file.

Then, a statistical description of the columns of a numerical nature is displayed, where the arithmetic mean, standard deviation, the highest value and the lowest value are displayed in order to identify the data.

Figure 7 shows the statistical description of Chest_xray_Corona_Metadata.csv file.

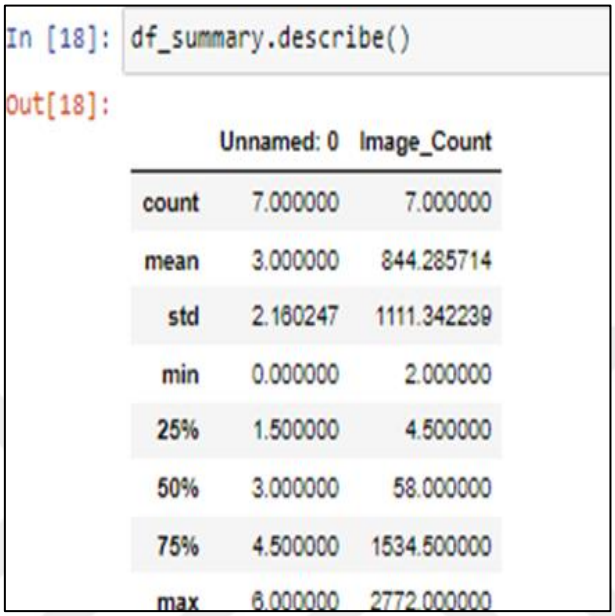
```
In [17]: data_df.describe()
```

Out[17]:

	Unnamed: 0
count	5910.000000
mean	2957.075635
std	1710.186149
min	0.000000
25%	1477.250000
50%	2954.500000
75%	4431.750000
max	5932.000000

Figure 3.7: Chest_xray_Corona_Metadata.csv file describe method.

Figure 8 shows the statistical description of Chest xray Corona dataset Summary file.



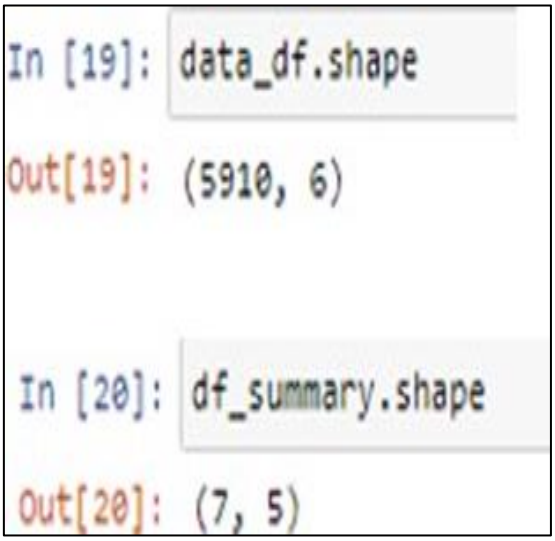
```
In [18]: df_summary.describe()
```

Out[18]:

	Unnamed: 0	Image_Count
count	7.000000	7.000000
mean	3.000000	844.285714
std	2.160247	1111.342239
min	0.000000	2.000000
25%	1.500000	4.500000
50%	3.000000	58.000000
75%	4.500000	1534.500000
max	6.000000	2772.000000

Figure 3.8: Chest_xray_Corona_dataset_Summary.csv file describe method.

Figure 9 shows the shape of two CSV files, where the first file contains 6 columns and 591 'lines, while the second file contains 5 columns and 7 lines.



```
In [19]: data_df.shape

Out[19]: (5910, 6)


In [20]: df_summary.shape

Out[20]: (7, 5)
```

Figure 3.9: Shape of the csv files.

Figure 2 ‘shows the data types in the first CSV file, which contains a column of type INT and the rest of type Object.

```
In [21]: data_df.dtypes

Out[21]: Unnamed: 0                int64
         X_ray_image_name         object
         Label                    object
         Dataset_type             object
         Label_2_Virus_category   object
         Label_1_Virus_category   object
         dtype: object
```

Figure 3.10: Data type of Chest_xray_Corona_Metadata.csv file.

Figure 11 shows the data types in the second CSV file, which contains two columns of type INT and the rest of type Object.

```
In [22]: df_summary.dtypes

Out[22]: Unnamed: 0                int64
         Label                    object
         Label_1_Virus_category   object
         Label_2_Virus_category   object
         Image_Count               int64
         dtype: object
```

Figure 3.11: Data type of Chest_xray_Corona_dataset_Summary.csv file.

Then the unique values in label_2_Virus_category column are returned in Chest_xray_Corona_Metadata.csv file arranged in descending order with the first element being the most frequent element as shown in figure 12 Where there are 58 cases of COVID-19, 5 cases of Streptococcus, 4 cases of SARS, and two cases of ARDS.

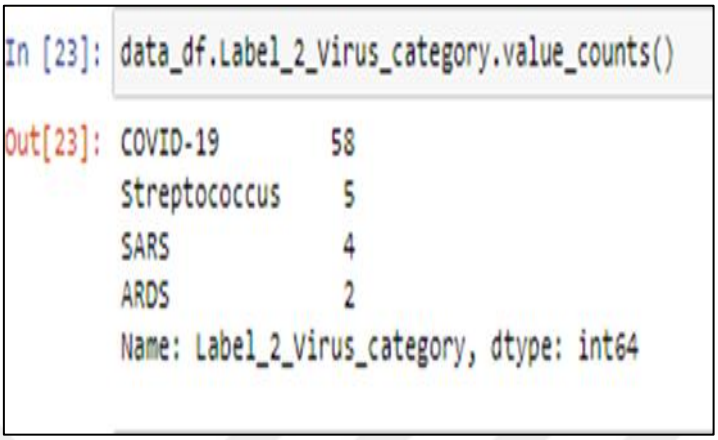


Figure 3.12: Unique values in label_2_Virus_category column.

Then the unique values in label_1_Virus_category column are returned in Chest_xray_Corona_Metadata.csv file arranged in descending order with the first element being the most frequent element as shown in figure 13 Where there are 2777 cases of bacteria, 1555 cases of virus, and two cases of Stress- Smoking.

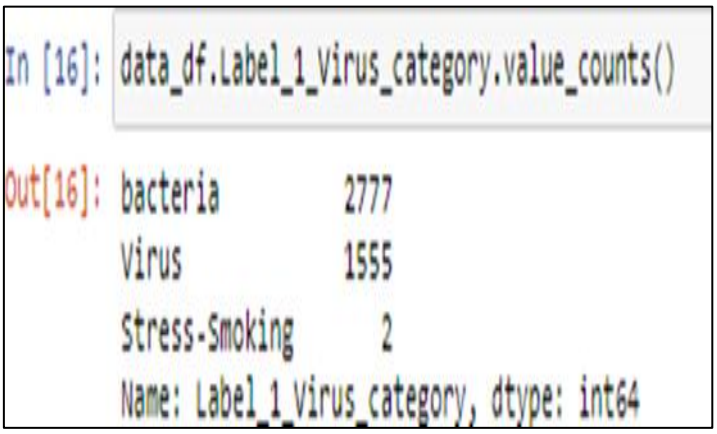


Figure 3.13: Unique values in label_1_Virus_category.

Then the first five lines of training data are displayed as shown in figure 14.

```
train_data.head()
```

	Unnamed: 0	X_ray_image_name	Label	Dataset_type	Label_2_Virus_category	Label_1_Virus_category
0	0	IM-0128-0001.jpeg	Normal	TRAIN	NaN	NaN
1	1	IM-0127-0001.jpeg	Normal	TRAIN	NaN	NaN
2	2	IM-0125-0001.jpeg	Normal	TRAIN	NaN	NaN
3	3	IM-0122-0001.jpeg	Normal	TRAIN	NaN	NaN
4	4	IM-0119-0001.jpeg	Normal	TRAIN	NaN	NaN

Figure 3.14: The first five lines of training data.

Then the nulls are calculated in the training data as shown in the figure 15.

```
train_data.isna().sum()
```

```
Unnamed: 0                 0
X_ray_image_name           0
Label                      0
Dataset_type               0
Label_2_Virus_category    5217
Label_1_Virus_category    1342
dtype: int64
```

Figure 3.15: Null values in training data.

Then the unique values in label_1_Virus_category column are returned in training data arranged in descending order with the first element being the most frequent element as shown in figure 16 Where there are 2535 cases of bacteria, 14 " "7 cases of virus, and two cases of Stress-Smoking.

```
train_data['Label_1_Virus_category'].value_counts()

bacteria          2535
Virus             1407
Stress-Smoking       2
Name: Label_1_Virus_category, dtype: int64
```

Figure 3.16: Unique values in label_1_Virus_category in training data.

After that, graphs are drawn to get a better understanding of the entire data for the following columns label, label_2_Virus_Category, and lable_1_Virus_Category in the training data as shown in figures 17,18,19.

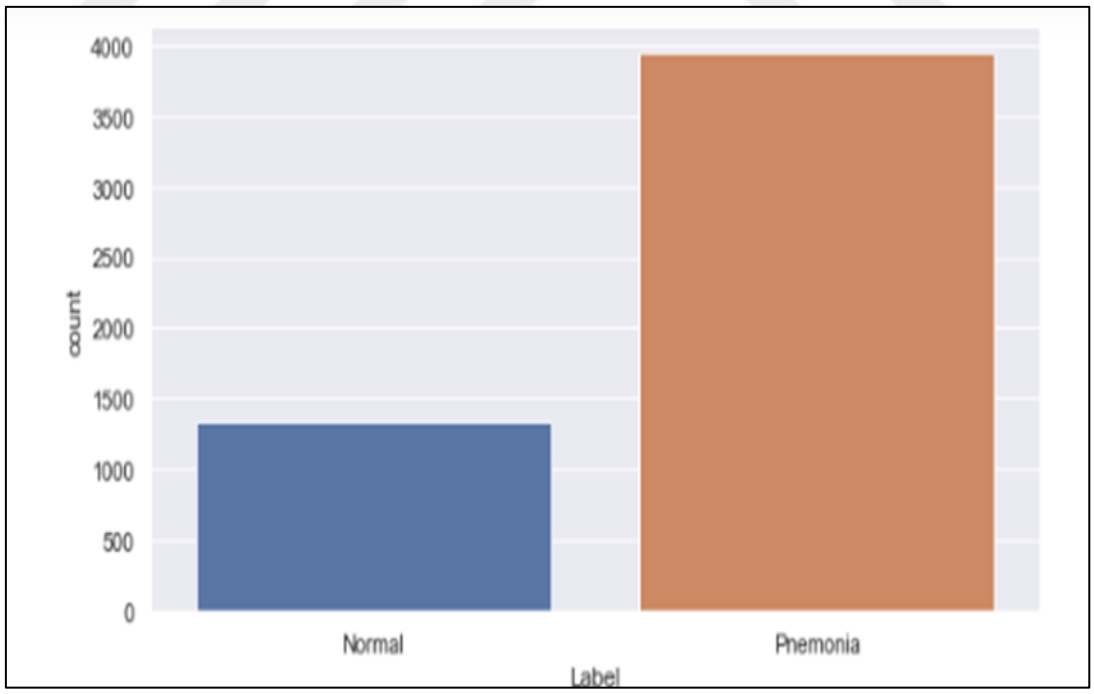


Figure 3.17: Bar chart for label in training data.

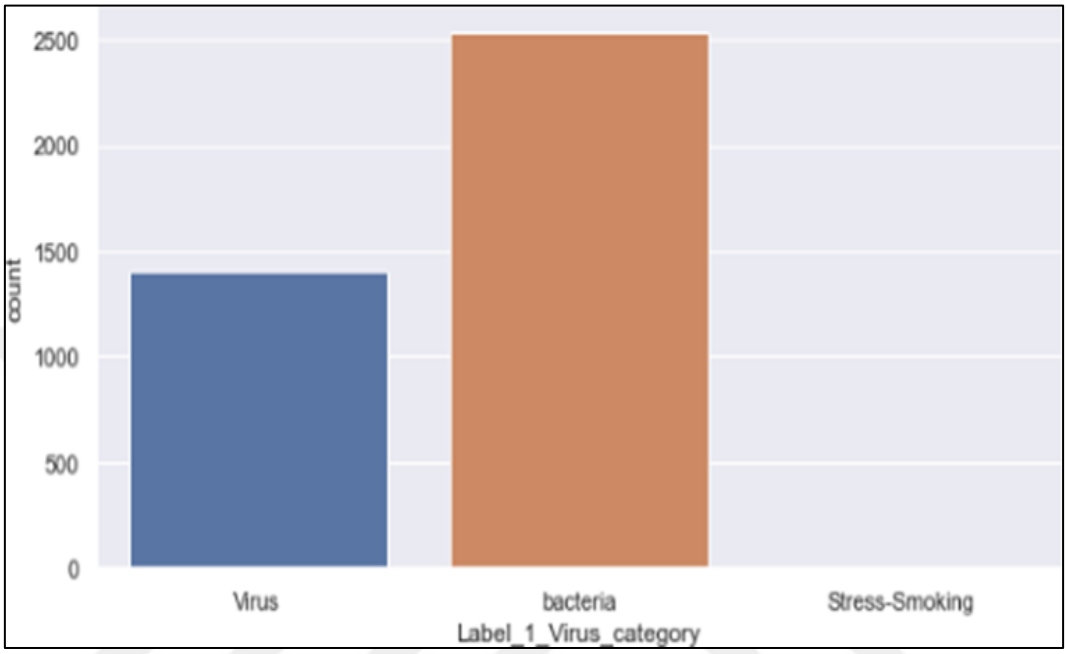


Figure 3.18: Bar chart for label_1_Virus_Category in training data.

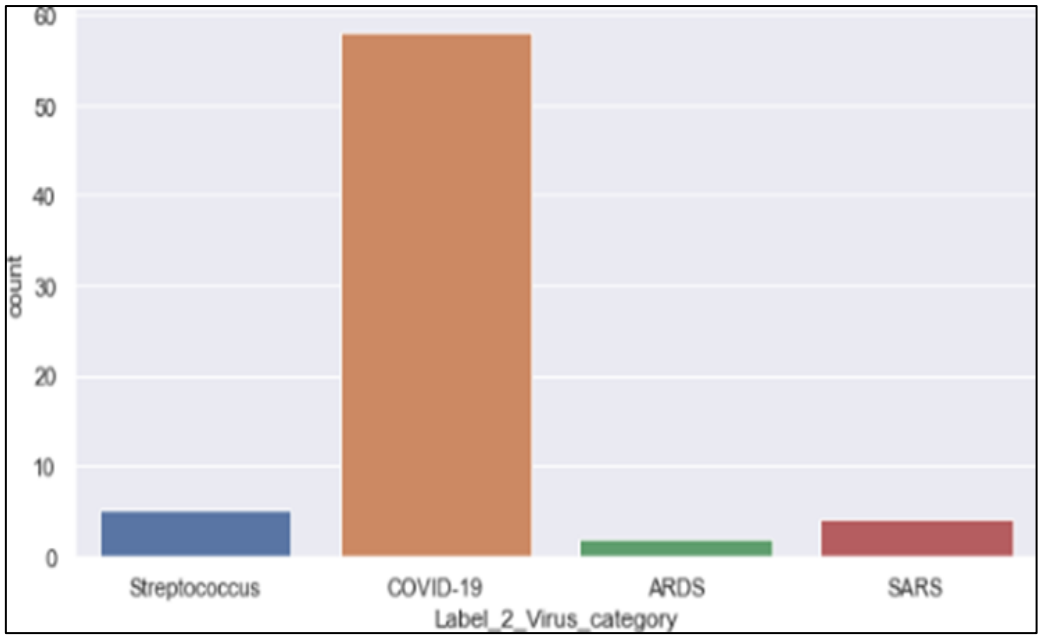


Figure 3.19: Bar chart for label_2_Virus_Category in training data.

Then the test data form is identified, and the null values are known in it as shown in the figure 2.

```
In [30]: test_data=data_df[data_df['Dataset_type']=='TEST']

In [31]: test_data.shape
Out[31]: (624, 6)

In [32]: test_data.isna().sum()
Out[32]: Unnamed: 0                  0
         X_ray_image_name            0
         Label                       0
         Dataset_type                0
         Label_2_Virus_category    624
         Label_1_Virus_category    234
         dtype: int64
```

Figure 3.20: Shape and null values in test data.

Then the unique values in label_1_Virus_category column are returned in test data arranged in descending order with the first element being the most frequent element as shown in figure 21 Where there are 242 cases of bacteria, and 148 cases of virus.

```
test_data['Label_1_Virus_category'].value_counts()

bacteria    242
Virus       148
Name: Label_1_Virus_category, dtype: int64
```

Figure 3.21: Unique values in label_1_Virus_categor in test data.

After that, graphs are drawn for the label_2_Virus_Category, and lable_1_Virus_Category in the training data as shown in figures 22.

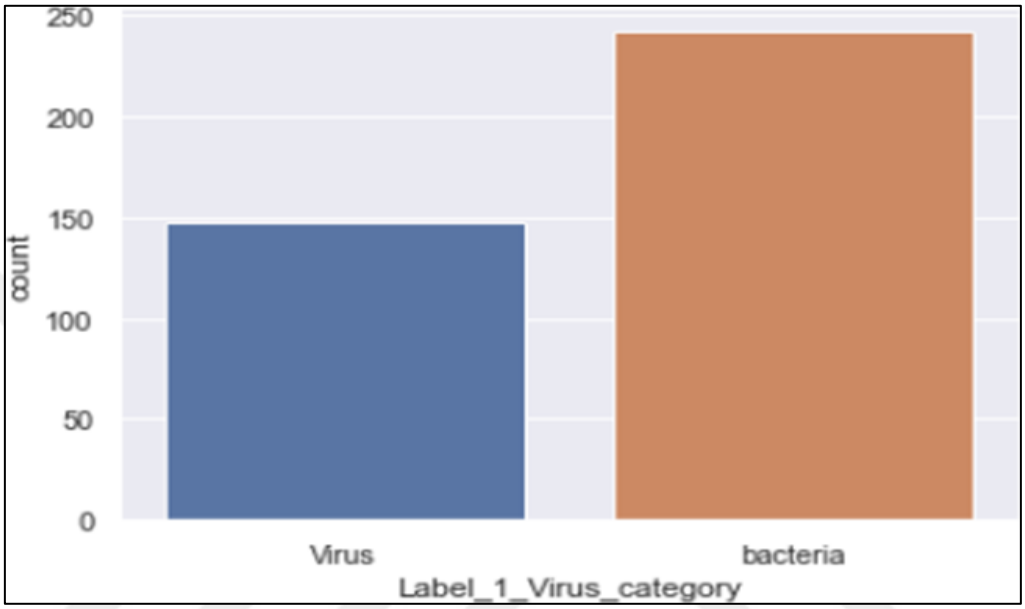


Figure 3.22: Bar chart for label_1_Viirus_category in test data.

After that, graphs are drawn for the label, and lable_1_Virus_Category in the training data as shown in figures 23,24.

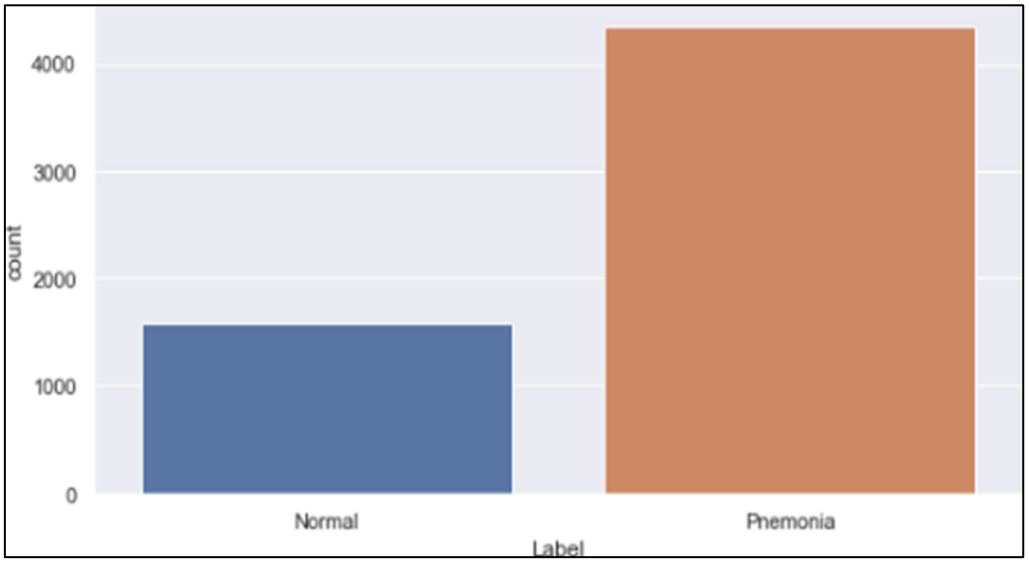


Figure 3.23: Bar chart for label in first csv file.

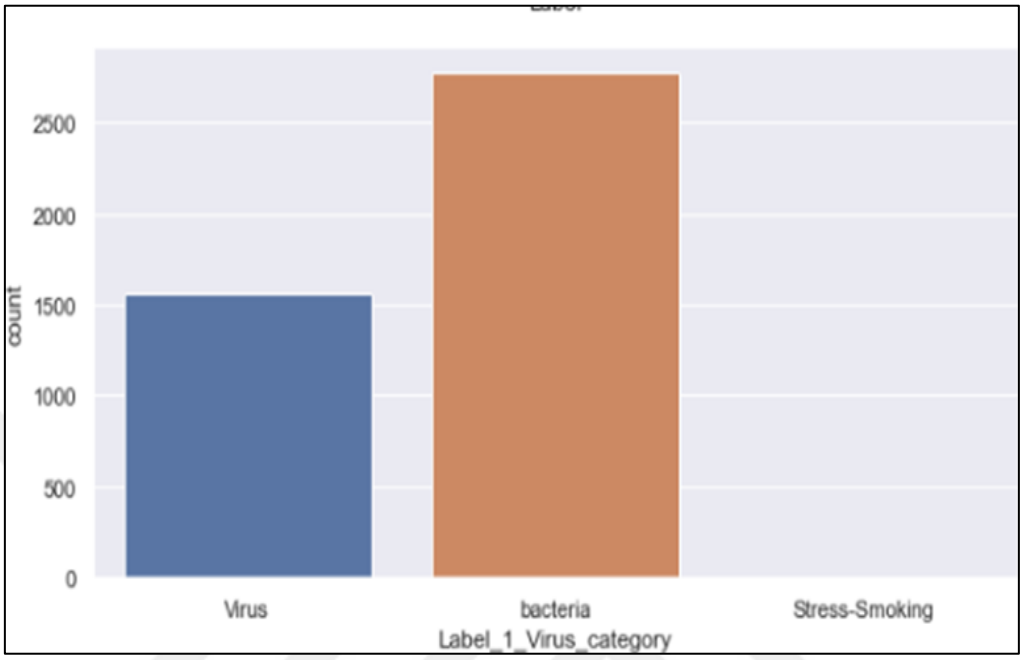


Figure 3.24: Bar chart for label_1_Viirus_category in first csv file.

After the stage of data exploration was completed through the previous statistical operations that were mentioned, the stage of building a deep learning model begins now to analyze images, identify disease, and then predict disease through images that will be passed to the model later.

In order to achieve this, the column named Label will be combined with Label_2_Virus_category because the second column is the one that determines exactly if the infection is caused by the virus. Before that, the empty values in the dataset will be filled with that "NA", these steps will be performed on both the training and testing data as shown in the figures 25, 26.

```
train_data.fillna('NA', inplace = True)
train_data['Label'] = train_data['Label']+"/"+train_data['Label_2_Virus_category']

train_dff = train_data[(train_data['Label'] =='Pnemonia/COVID-19') | (train_data['Label'] == 'Normal/NA') \
                    | (train_data['Label'] == 'Pnemonia/NA')]
train_dff = train_dff.sample(frac = 1)
print(len(train_dff))

5275
```

Figure 3.25: Replace empty values and fill column label in training data.

```
test_data.fillna('NA', inplace = True)
test_data['Label'] = test_data['Label']+"/"+test_data['Label_2_Virus_category']
test_df = test_data[(test_data['Label'] =='Pnemonia/COVID-19') | (test_data['Label'] == 'Normal/NA') \
                  | (test_data['Label'] == 'Pnemonia/NA')]

test_df = test_df.sample(frac = 1)
print(len(test_df))

624
```

Figure 3.26: Replace empty values and fill column label in test data.

Figure 27 shows the first two lines of test data in which the Label column is equal to Normal/NA.

```
test_df[test_df['Label']=='Normal/NA'].head(2)
```

	Unnamed: 0	X_ray_image_name	Label	Dataset_type	Label_2_Virus_category	Label_1_Virus_category
5445	5468	NORMAL2-IM-0219-0001.jpeg	Normal/NA	TEST	NA	NA
5313	5336	NORMAL2-IM-0033-0001.jpeg	Normal/NA	TEST	NA	NA

Figure 3.27: First two lines of test data in which the Label column is equal to Normal/NA.

Figure 28 shows the first two lines of test data in which the Label column is equal to Pnemonia/NA.

```
test_df[test_df['Label']=='Pnemonia/NA'].head(2)
```

	Unnamed: 0	X_ray_image_name	Label	Dataset_type	Label_2_Virus_category	Label_1_Virus_category
5619	5642	person132_bacteria_632.jpeg	Pnemonia/NA	TEST	NA	bacteria
5858	5881	person173_bacteria_831.jpeg	Pnemonia/NA	TEST	NA	bacteria

Figure 3.28: First two lines of test data in which the Label column is equal to Pnemonia/NA.

But when the values are shown in the test data that contain Pnemonia/COVID- 19, which are the confirmed cases of the image of viral origin, specifically Corona, nothing appears as shown in the figure 29.

```
test_df[test_df['Label']=='Pnemonia/COVID-19']
```

	Unnamed: 0	X_ray_image_name	Label	Dataset_type	Label_2_Virus_category	Label_1_Virus_category

Figure 3.29: Test data in which the Label column is equal to Pneumonia/ COVID-19.

Thus, to avoid the previous problem, the training data will be divided into training data and test data, where the test data will be the last 6 ' 'samples and the rest will be training data as shown in figure 3 " ".

```
test_df_covid = train_dff[-600:]
train_dff = train_dff[:-600]
print(len(test_df_covid))
print(len(train_dff))
```

```
600
4675
```

Figure 3.30: Split training data into train and test data.

And then to make sure that the previous problem has been solved, the first five lines of the test data will be shown after the previous division process as shown in figure 31.

```
test_df_covid[test_df_covid['Label']=='Pnemonia/COVID-19']
```

	Unnamed: 0	X_ray_image_name	Label	Dataset_type	Label_2_Virus_category	Label_1_Virus_category
5269	5268	2C10A413-AABE-4807-8CCE-6A2025594067.jpeg	Pnemonia/COVID-19	TRAIN	COVID-19	Virus
5238	5248	F83AB8CE-1958-4154-A70F-913AF154F53D.jpeg	Pnemonia/COVID-19	TRAIN	COVID-19	Virus
5228	5237	gr1_lrg-b.jpg	Pnemonia/COVID-19	TRAIN	COVID-19	Virus
5248	5260	8FDE8DBA-CFBD-4B4C-B1A4-6F36A93B7E87.jpeg	Pnemonia/COVID-19	TRAIN	COVID-19	Virus
5270	5287	23E99E2E-447C-46E5-8EB2-D35D12473C39.png	Pnemonia/COVID-19	TRAIN	COVID-19	Virus

Figure 3.31: First five lines of the test data will be shown after the previous division process.

After that, it will use image augmentation, which is described as a technique for adding various alterations to the original photos to produce many altered copies with same image. However, depending mostly on augmentation techniques you use, such as shifting, flipping, flipping, etc., single version differs from that in certain ways.

These little adjustments to the final picture just offer a different angle for photographing the object as it would actually appear in real life, without altering the target category. Therefore, we frequently employ it to create deep learning models.

This is performed by using Keras' Image Data Generator.

A. Figure 32 provides instructions for using ImageDataGenerator only with following sites Rotations

One of the often utilized zoom methods involves rotating the picture, which enables the model to lock onto the object's orientation.

B. Abrupt Shifts

The image's center may not always contain the item. We may solve this issue by providing a certain specific value to every pixels in order to move the parts of the image alternatively horizontally and vertically.

The height shift range and width shift range arguments for the image's vertical and horizontal shifts, respectively, are both contained in the ImageDataGenerator class. If the number is a floating-point number, it reflects the proportion of the image's width or height to The height shift range parameter for the file's vertical move and the width shift range argument for the file's horizontal shift are both contained in the ImageDataGenerator class. The % of the image's width or height that will be moved if the amount is a floating number. If not, both width or thickness will automatically be off by that many pixel values if the value is an integer.

C. Arbitrary Flips

It makes sense to employ image flipping as a zooming technique with a variety of different items.

The horizontal flip and vertical flip arguments in the ImageDataGenerator class allow for inversion along either the top or bottom axis. This method, meanwhile, has to work with the image's subject. For instance, it wouldn't make too much sense to flip an automobile vertically as opposed to this in something else, like a soccer ball. Having stated that, I'll flip my photograph in both sides to demonstrate how the increase will manifest itself.

D. Variable brightness

randomly varies the image's brightness. Because our item won't always be under ideal lighting, it is also a highly helpful augmentation strategy. It is therefore vital to train your model on photographs taken in various lighting scenarios.

The Brightness range option there in Image Data Genrator class allows brightness to be adjusted. It will select a brightness compensating value from a list of 2 floats and from that

range. Images get darker when the value is less than 1. " ", and brighter when the value is larger than 1.

E. Arbitrary zoom

Randomly, the picture gets scaled up or down.

The zoom range option of the ImageDataGenerator class accepts a floating number of the zoom. The lowest and maximum values can be specified as a list of two figures. If a floating number is used instead, its zoom will be done between [1-zoom range, 1 + zoom range].

Any number lower than 1 will make the picture larger. While the picture will also be scaled down if any value is more than 1

```
image_gen = tf.keras.preprocessing.image.ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.15,
    height_shift_range=0.15,
    brightness_range=None,
    zoom_range=0.10,
    channel_shift_range=0.0,
    fill_mode="nearest",
    cval=0.0,
    horizontal_flip=True,
    rescale=1./255,
    preprocessing_function=None,
    validation_split=0.2,
    dtype=None,
)
```

Figure 3.32: Image augmentation.

Use Flow from Dataframe instead, as it allows you to directly enhance photos by getting their name and specific value from either a dataframe. Flow from Dataframe is another fantastic method in the and then class. Because all the photographs are kept in the same folder, we will utilize it there.

There are few other arguments to this function that should be quickly explained:

picture names and desired outputs are included in the Pandas dataframe.

Directory: The address of the folder holding each and every photograph.

x col: The title of the DataFrame column that houses the picture names.

y col: The title of the DataFrame column that houses the target data.class_mode: setting to binary is for 1-D binary labels while categorical is for 2- D labels with one hot encoder.

target_size: the size of the entered images. batch_size: the size of the data sets.

Seeds: Adjust to reproduce the result.

Figure 33 shows augment training images directly.

```
train_datagen = image_gen.flow_from_dataframe(
    dataframe=train_dff,
    directory=train_dir,
    x_col="X_ray_image_name",
    y_col="Label",
    target_size=(256, 256),
    color_mode="rgb",
    class_mode="categorical",
    batch_size=32,
    seed=25,
    shuffle=True,
    subset='training'
)

valid_datagen = image_gen.flow_from_dataframe(
    dataframe=train_dff,
    directory=train_dir,
    x_col="X_ray_image_name",
    y_col="Label",
    target_size=(256, 256),
    color_mode="rgb",
    class_mode="categorical",
    batch_size=32,
    seed=25,
    shuffle=True,
    subset='validation'
)
```

```
Found 3740 validated image filenames belonging to 3 classes.
Found 935 validated image filenames belonging to 3 classes.
```

Figure 3.33: Augment training images directly.

Figure 34 shows the process of augment test images directly for the test data that took place before the process of splitting the training data and after the splitting process. It contains 624 images.

```
test_datagen_covid = image_gen.flow_from_dataframe(
    dataframe=test_df_covid,
    directory=train_dir,
    x_col="X_ray_image_name",
    y_col="Label",
    classes = ['Normal/NA','Pnemonia/COVID-19','Pnemonia/NA'],
    target_size=(256, 256),
    color_mode="rgb",
    class_mode="categorical",
    batch_size=32,
    seed=25,
    shuffle=True
)
```

```
Found 600 validated image filenames belonging to 3 classes.
```

```
test_datagen = image_gen.flow_from_dataframe(
    dataframe=test_df,
    directory=test_dir,
    x_col="X_ray_image_name",
    y_col="Label",
    classes = ['Normal/NA','Pnemonia/COVID-19','Pnemonia/NA'],
    target_size=(256, 256),
    color_mode="rgb",
    class_mode="categorical",
    batch_size=32,
    seed=25,
    shuffle=True
)
```

```
Found 624 validated image filenames belonging to 3 classes.
```

Figure 3.34: Augment test images directly.

The model will then be built using a Convolutional Neural Network (CNN). From here, many experiments will be carried out in order to first determine LR and then determine the optimal number of layers of the CNN model by getting the best accuracy in the best time.

Initially, experiments will be conducted to determine the optimal learning rate to use Where the callback function will be created to stop the training process training when the val_accuracy reaches below ".2 to prevent overfitting as shown in figure 35.

```
# Creating a callback which stops the training when the val_accuracy reaches below 0.2 to prevent overfitting
class myCallback(tf.keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs={}):
    if(logs.get('val_loss') < 0.2):
      print("\n Reached val_loss < 0.2")
      self.model.stop_training = True
```

Figure 3.35: Call back function.

The learning rate will initially be tried equal to “ ". ' '" "2.

```
Callback =  myCallback()


opt = tf.keras.optimizers.Adam(learning_rate=0.0002)
model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])

history2 = model.fit(train_datagen, steps_per_epoch = 3740/32, epochs =20,validation_data = valid_datagen, validation_steps = 93
)
```

Figure 3.36: The experiment of using learning rate equal to “ 2.

val_accuracy reaches below “ .2 at epoch 19 as shown in figure 37.

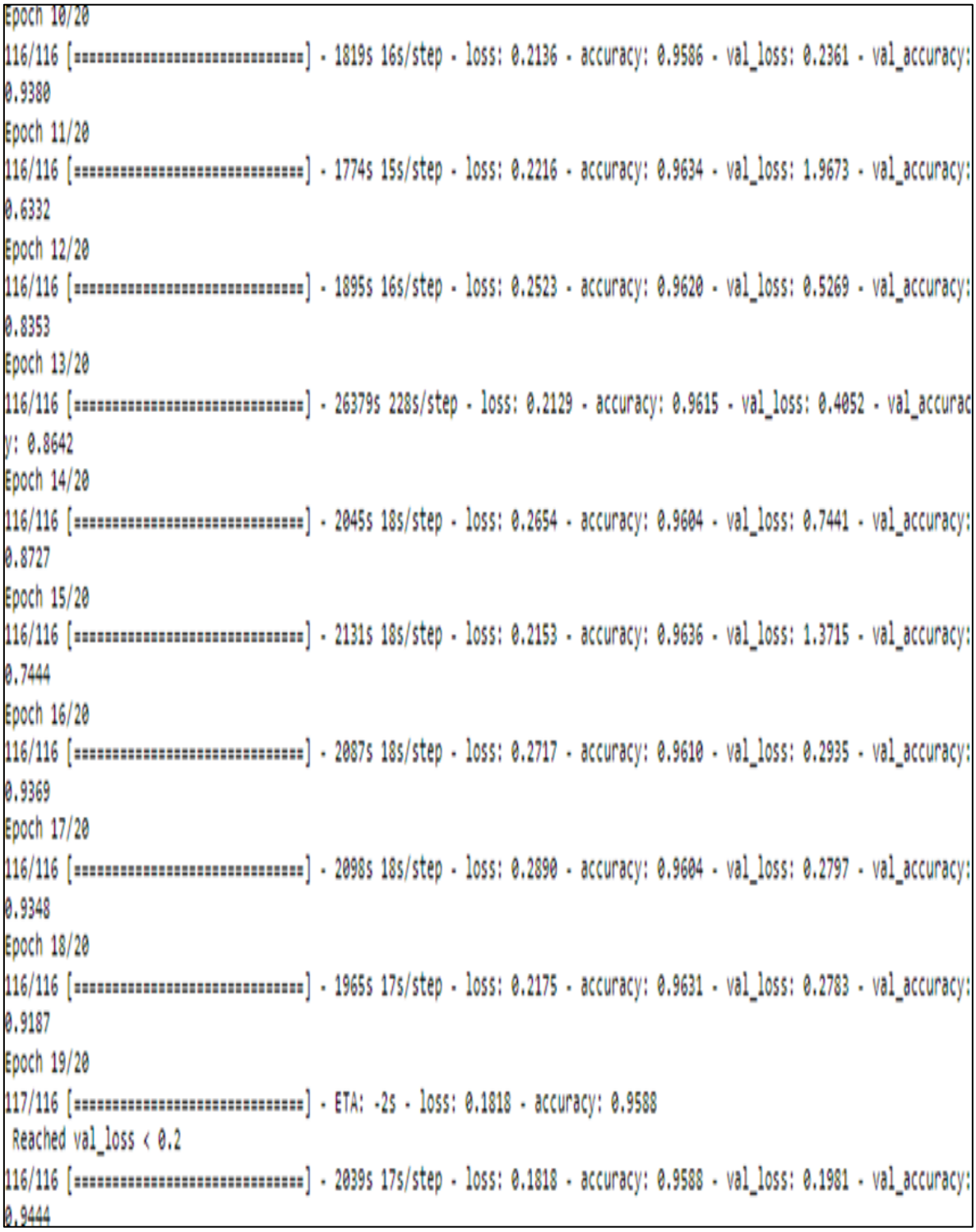


Figure 3.37: Val_accuracy with learning rate equals "2.

Then learning rate will be tried equal to " ". ' '" "3.

```
Callback =  myCallback()


opt = tf.keras.optimizers.Adam(learning_rate=0.0003)
model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])

history2 = model.fit(train_datagen, steps_per_epoch = 3740/32, epochs =20,validation_data = valid_datagen, validation_steps = 93
)
```

Figure 3.38: The experiment of using learning rate equal to "3.

val_accuracy reaches below ".2 at epoch 14 as shown in figure 39.

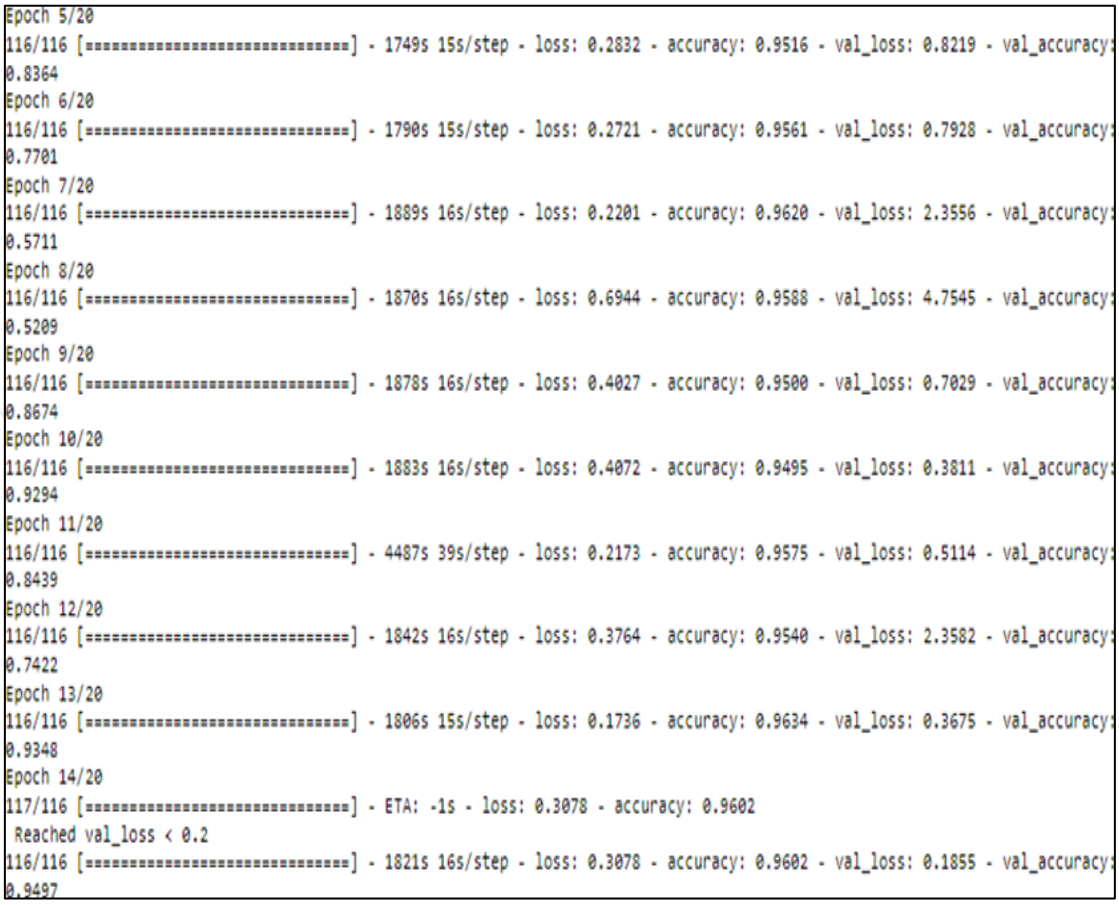


Figure 3.39: Val accuracy with learning rate equals "3.

Then learning rate will be tried equal to “. "4 as shown in figure 4 ".

Figure 3.40: The experiment of using learning rate equal to "4.

As shown in the figure 41, using the learning rate equal to “."4, all epochs were passed and no val_accuracy less than “.2 was reached.

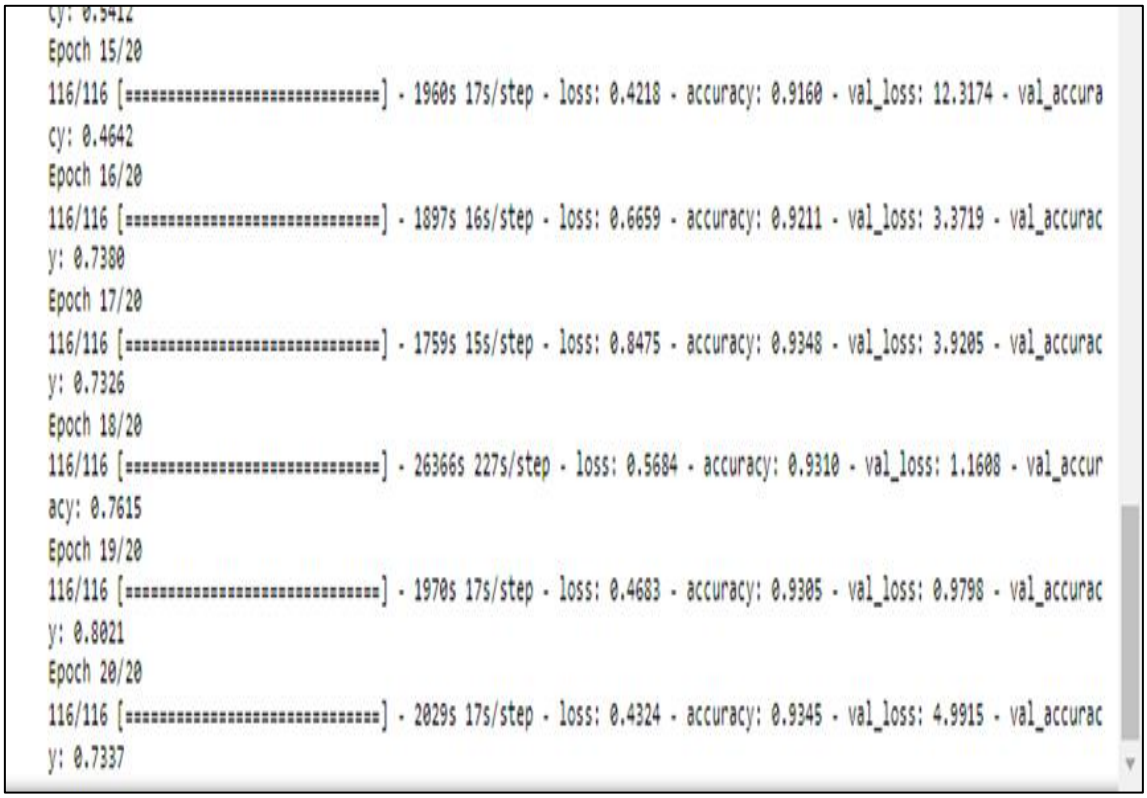


Figure 3.41: Val accuracy with learning rate equals "4.

And based on the previous three experiments to determine the appropriate learning rate, the learning rate will be used equal to “. "3, as it reached the best ratio of val_accuracy with the least number of epochs this experiment will be called the zero experiment.

It is also worth noting that previous experiments to determine the learning rate values were done on a seven-layer CNN model as shown in figure 42.

```
weight_decay = 1e-4
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay), input_shape=[256,256,3]))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Flatten())
model.add(Dense(3, activation='softmax'))

model.summary()
```

Figure 3.42: The CNN model used to determine the optimal learning rate for the rest of the experiments.

And after it has been verified that the learning rate is equal to “. "3 is the best, it will be used permanently in the rest of the experiments to build CNN models.

Experiments:

A. Experiment with increasing the number of layers by one layer to become 7 layers while maintaining dropout ".4

Figure 43 shows the process of building the layers and the parameters used in this experiment

```
weight_decay = 1e-4
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay), input_shape=[256,256,3]))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))


model.add(Flatten())
model.add(Dense(3, activation='softmax'))

model.summary()
```

Figure 3.43: First experiment 7 layers and dropout ".4.

As shown in figure 44 there is 534 " 51 trainable params.

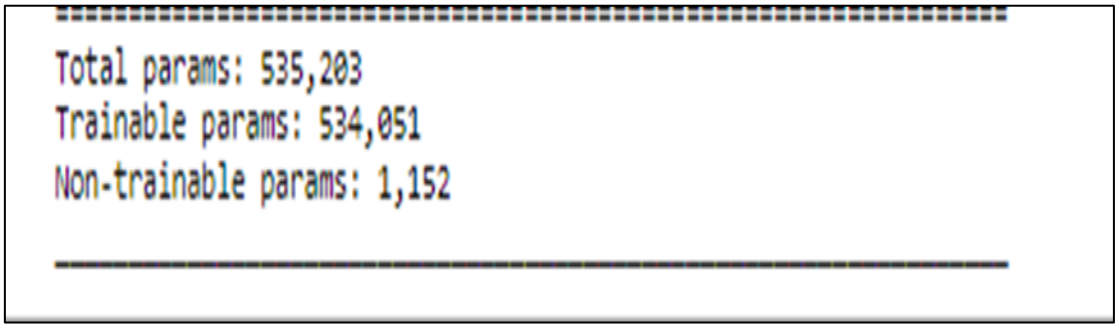


Figure 3.44: Trainable params in first experiment.

As previously explained, the learning rate will be used as “. ' "3, and the learning rate will be the criterion for accuracy and keeping the epochs equal to 2 ".

We went through all the epochs, and we didn't get to val_accuracy less than ".2 as shown in figure 45.

It is worth noting that the execution time was equal to 1867 " "8.

```
Epoch 13/20
116/116 [==============================] - 2010s 17s/step - loss: 0.3202 - accuracy: 0.9291 - val_loss: 3.8541 - val_accuracy:
0.5219
Epoch 14/20
116/116 [==============================] - 2057s 18s/step - loss: 0.2782 - accuracy: 0.9372 - val_loss: 0.2916 - val_accuracy:
0.9412
Epoch 15/20
116/116 [==============================] - 2065s 18s/step - loss: 0.3156 - accuracy: 0.9273 - val_loss: 0.3027 - val_accuracy:
0.9198
Epoch 16/20
116/116 [==============================] - 1973s 17s/step - loss: 0.2851 - accuracy: 0.9364 - val_loss: 0.6784 - val_accuracy:
0.8374
Epoch 17/20
116/116 [==============================] - 1969s 17s/step - loss: 0.2388 - accuracy: 0.9457 - val_loss: 0.8116 - val_accuracy:
0.7840
Epoch 18/20
116/116 [==============================] - 1979s 17s/step - loss: 0.2852 - accuracy: 0.9398 - val_loss: 0.9030 - val_accuracy:
0.6963
Epoch 19/20
116/116 [==============================] - 1914s 16s/step - loss: 0.2568 - accuracy: 0.9372 - val_loss: 2.9170 - val_accuracy:
0.6032
Epoch 20/20
116/116 [==============================] - 1900s 16s/step - loss: 0.2452 - accuracy: 0.9444 - val_loss: 5.4126 - val_accuracy:
0.4909
```

Figure 3.45: The results of the first experiment.

B. The second experiment will be with the same parameters as the previous experiment, but it will be the dropout in the last layer equal to

" ".5 instead of ".4.

Figure 46 shows the layers of the second experiment:

```
weight_decay = 1e-4
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay), input_shape=[256,256,3]))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5
model.add(Dropout(0.5))


model.add(Flatten())
model.add(Dense(3, activation='softmax'))

model.summary()
```

Figure 3.46: Layers of the second experiment.

Figure 47 shows the trainable params of the second experiment where there are 534 " "51 trainable params.

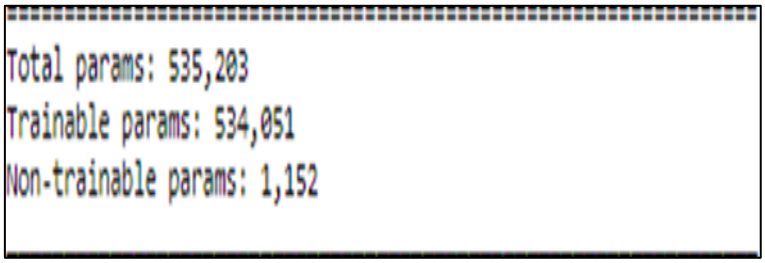


Figure 3.47: Trainable params in second experiment.

We went through all the epochs and we didn't get to val_accuracy less than ".2 as shown in figure 48.

It is worth noting that the execution time was equal to 77882 second.

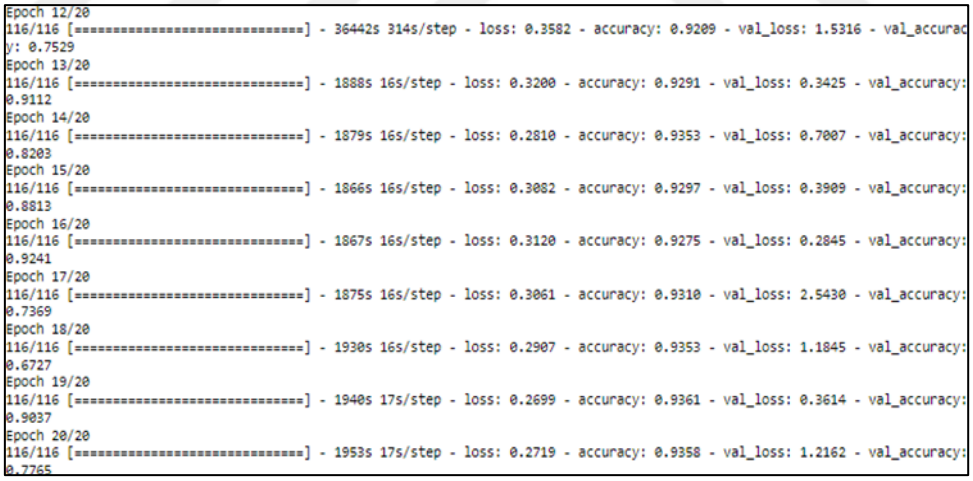


Figure 3.48: The results of the second experiment.

C. In the third experiment, the same layers of the second experiment will be kept with the same percentage of deletion, but here we will add an additional layer with Dropout equal to ".6.

Figure 49 shows the trainable params in third experiment where there are 6 " "8163 trainable params.

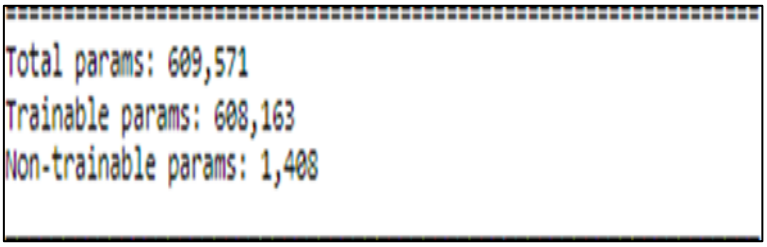


Figure 3.49: Trainable params in third experiment.

We went through all the epochs and we didn't get to val_accuracy less than ".2 as shown in figure 5 ".

The time required for implementation was also equal to 16124 " ".

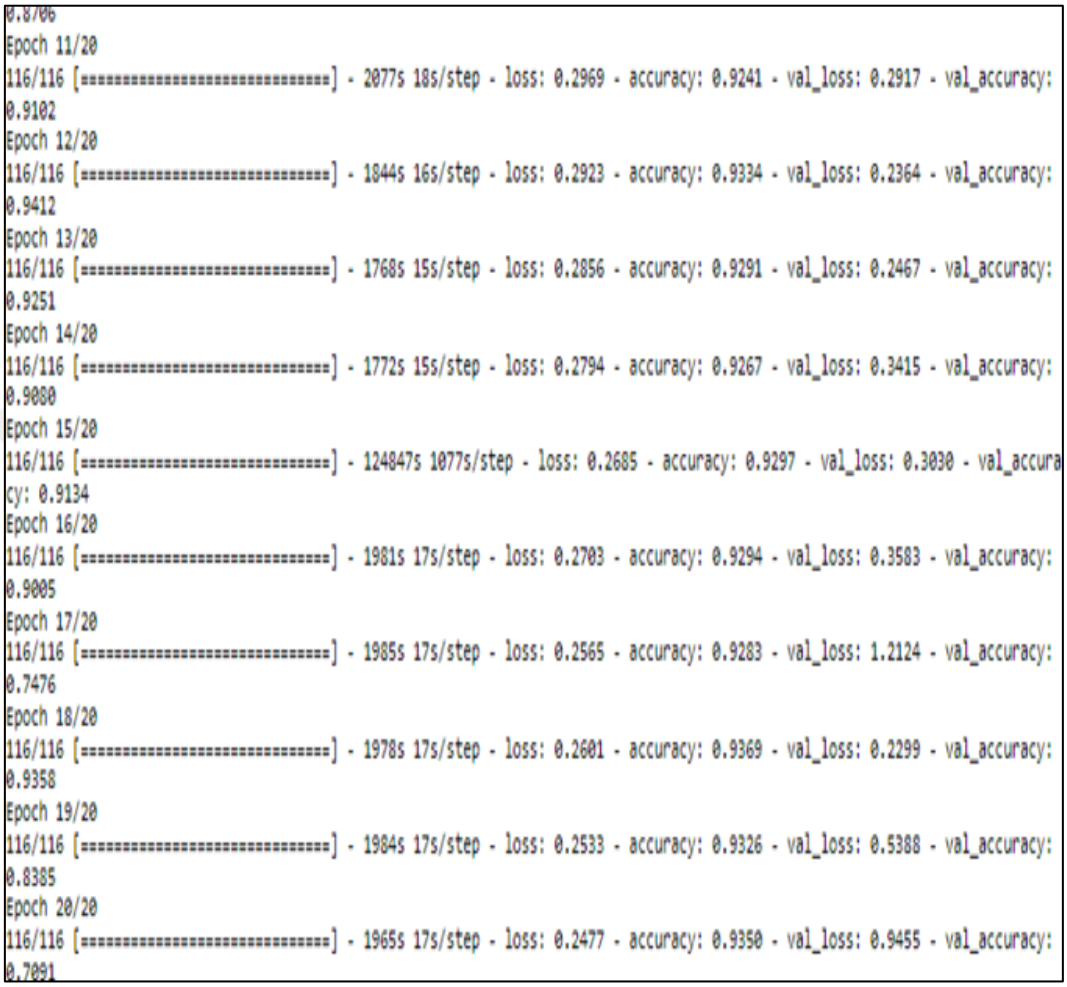


Figure 3.50: The result of the third experiment.

D. In the fourth experiment, five layers will be used according to the following parameters shown in the figure 51.

```
weight_decay = 1e-4
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay), input_shape=[256,256,3]))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))


model.add(Flatten())
model.add(Dense(3, activation='softmax'))

model.summary()
```

Figure 3.51: The layers in the fourth experiment.

Figure 52 shows the trainable params of the fourth experiment.

```
Total params: 852,771
Trainable params: 852,387
Non-trainable params: 384
```

Figure 3.52: Trainable params of the fourth experiment.

We went through all the epochs, and we didn't get to val_accuracy less than ".2 as shown in figure 53.

The time required for implementation was also equal to 63762.

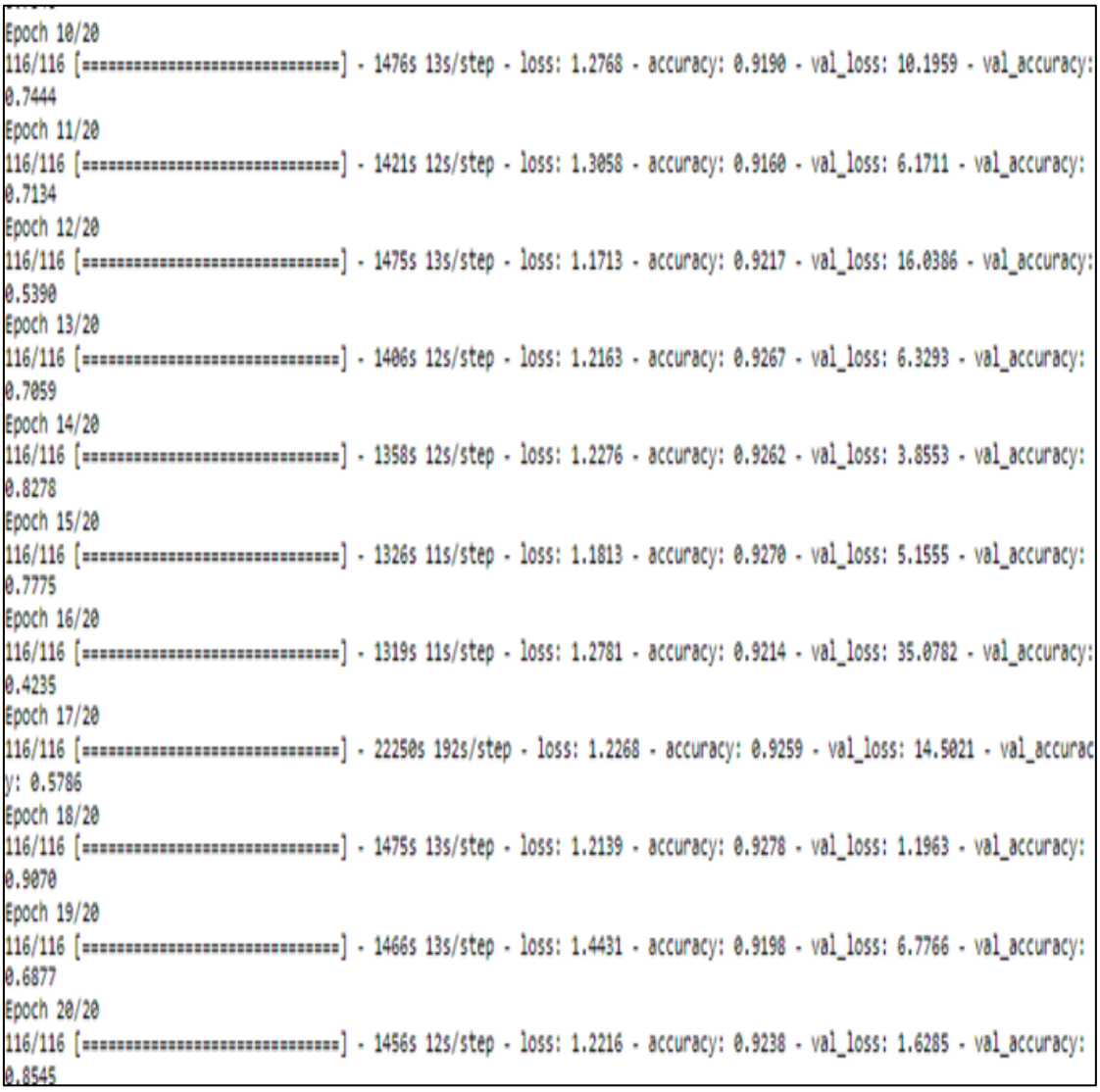


Figure 3.53: Result of fourth experiments.

E. In the fifth experiment, 1 ‘layers will be used according to the following parameters shown in the figure 54.

```
weight_decay = 1e-4
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay), input_shape=[256,256,3]))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5+  2layer
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5
model.add(Dropout(0.5))


model.add(Flatten())
model.add(Dense(3, activation='softmax'))

model.summary()
```

Figure 3.54: Layers of fifth experiment.

Figure 55 shows the trainable params of the fifth experiment.

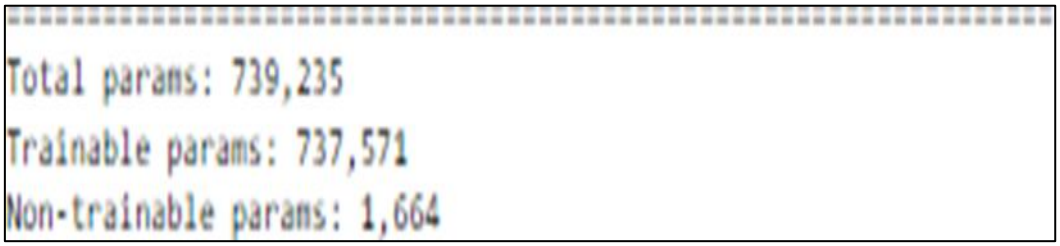


Figure 3.55: Trainable params of the fifth experiment.

We went through all the epochs and we didn't get to val_accuracy less than ".2 as shown in figure 53.

The time required for implementation was also equal to 94192.

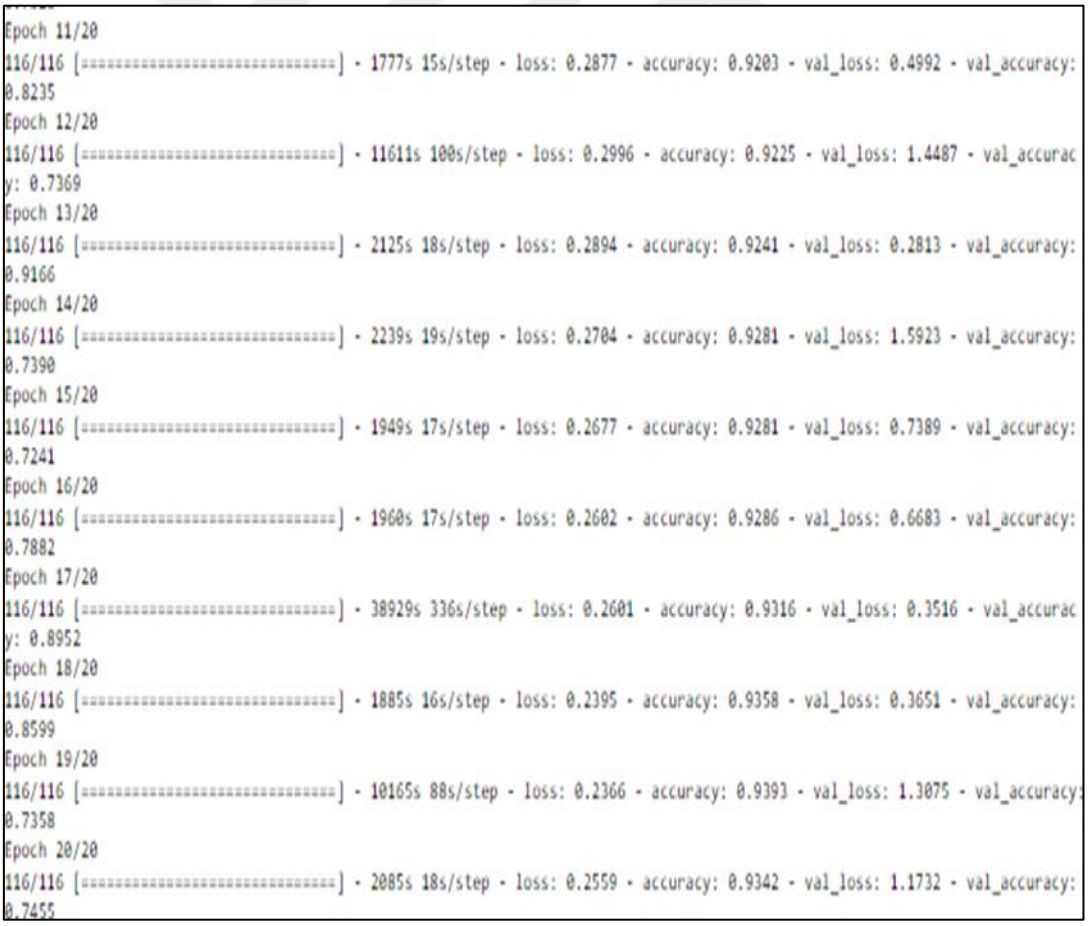


Figure 3.56: Result of fifth experiment.

F. In the Sixth experiment, 11 layers will be used according to the following parameters shown in the figure 57.

```
weight_decay = 1e-4
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay), input_shape=[256,256,3]))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
####0.5
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
####0.5#  4layer
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
####0.5
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
####0.5
model.add(Dropout(0.5))




model.add(Flatten())
model.add(Dense(3, activation='softmax'))
```

Figure 3.57: Sixth experiment parameters.

Figure 58 shows the trainable params of the sixth experiment.

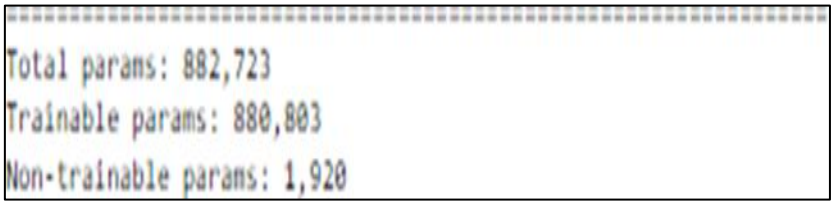


Figure 3.58: Trainable params of the sixth experiment.

We went through all the epochs, and we didn't get to val_accuracy less than ".2 as shown in figure 59. The time required for implementation was also equal to 4 " "4 " "7.

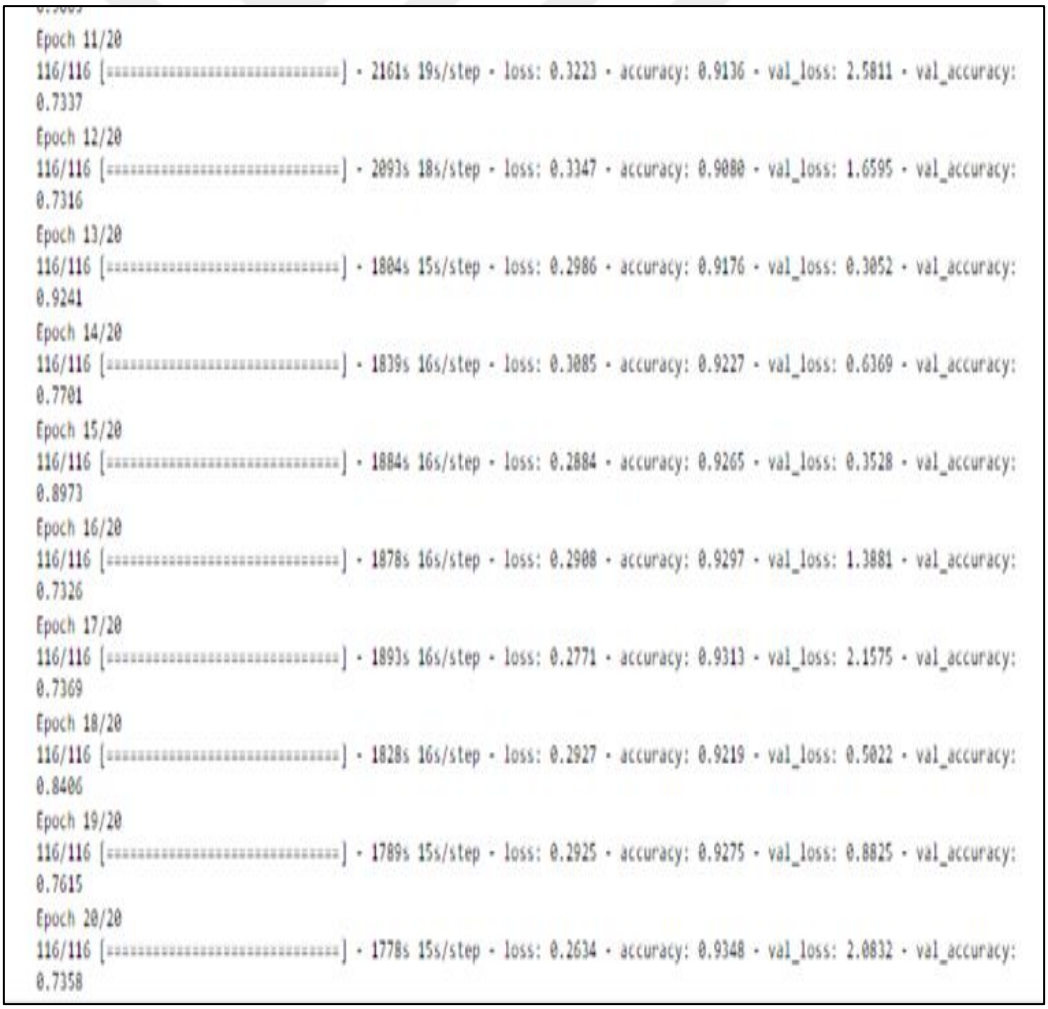


Figure 3.59: Result of sixth experiment.

Figure 6 'shows model accuracy and val_accuracy plot and the model loss through epochs.

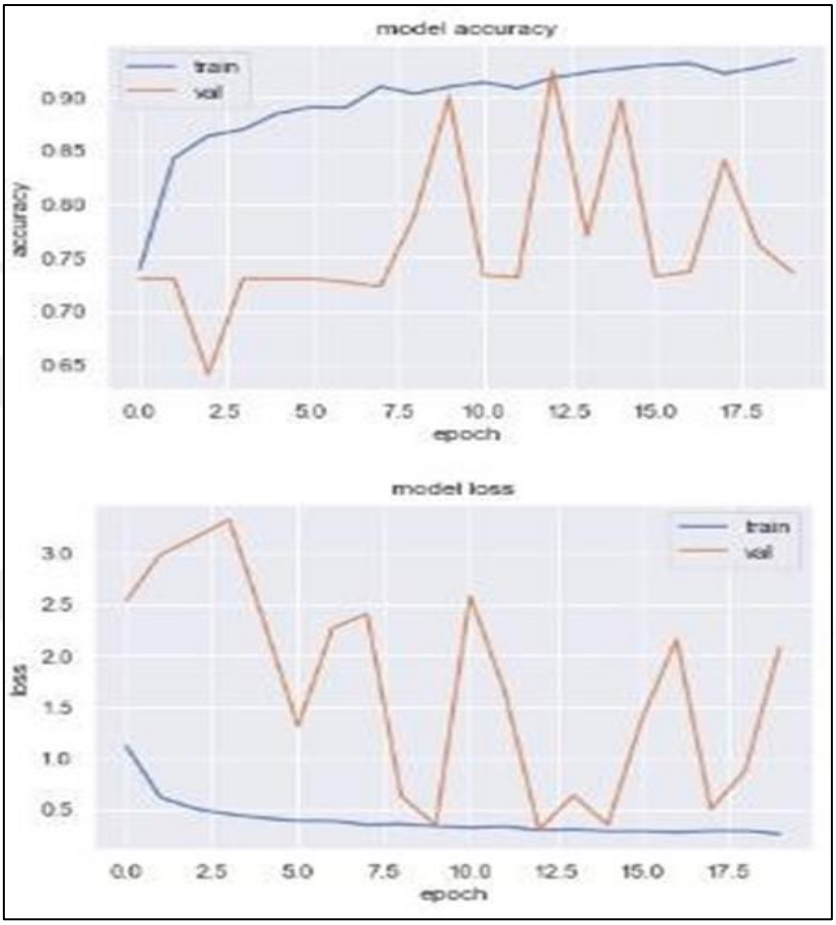


Figure 3.60: Model accuracy and model loss.

G. In the Seventh experiment, 12 layers will be used according to the following parameters shown in the figure 61.

```
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay), input_shape=[256,256,3]))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))
model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5+  5Layer
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5
model.add(Dropout(0.5))

model.add(Flatten())
model.add(Dense(3, activation='softmax'))

model.summary()
```

Figure 3.61: Layers of Seventh experiment.

Figure 62 shows the trainable params of the Seventh experiment.

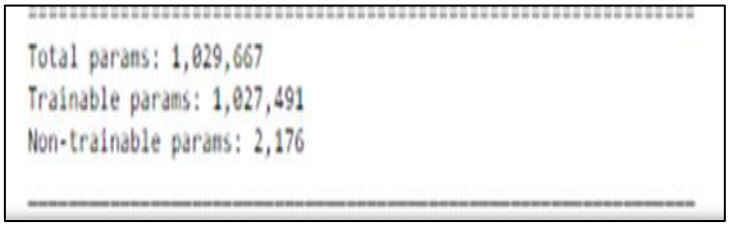


Figure 3.62: Trainable params of the Seventh experiment.

We went through all the epochs and we didn't get to val_accuracy less than ".2 as shown in figure 63. The time required for implementation was also equal to 66713.

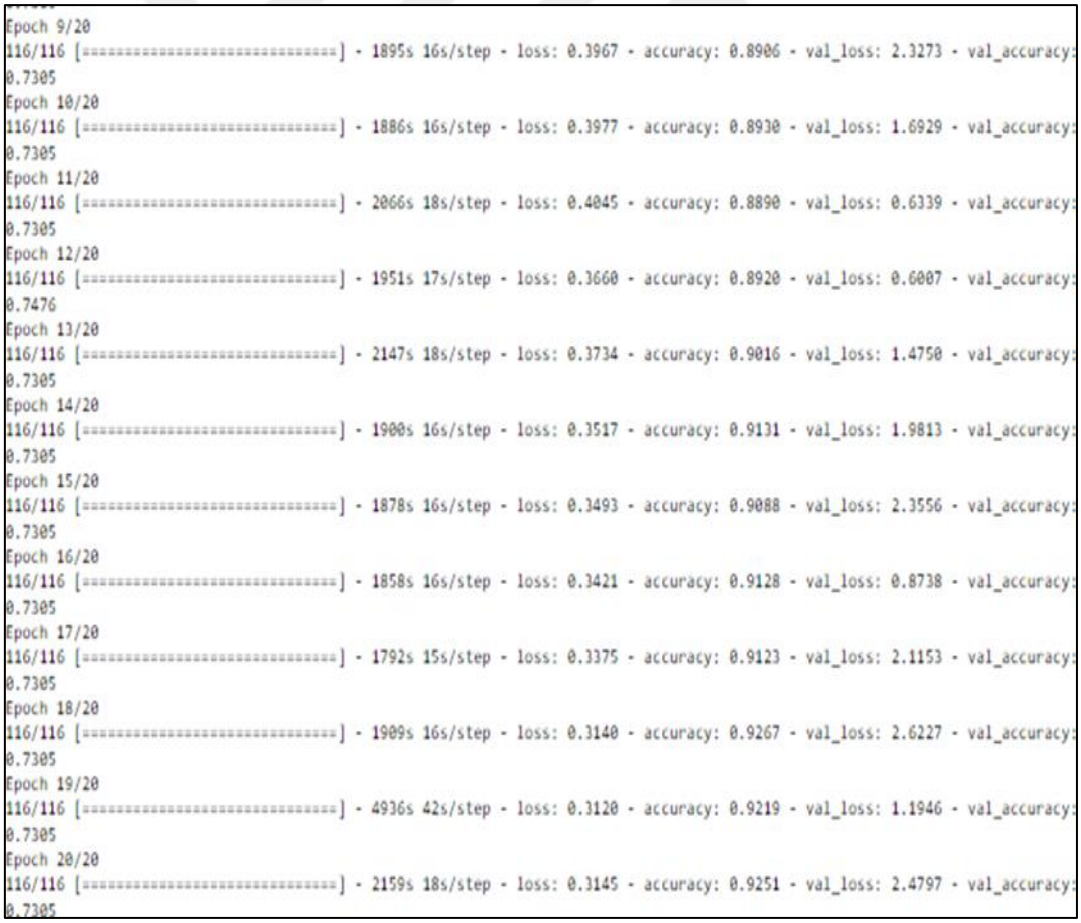


Figure 3.63: Result of Seventh experiment.

Figure 64 shows model accuracy and val_accuracy plot and the model loss through epochs in Seventh experiment.

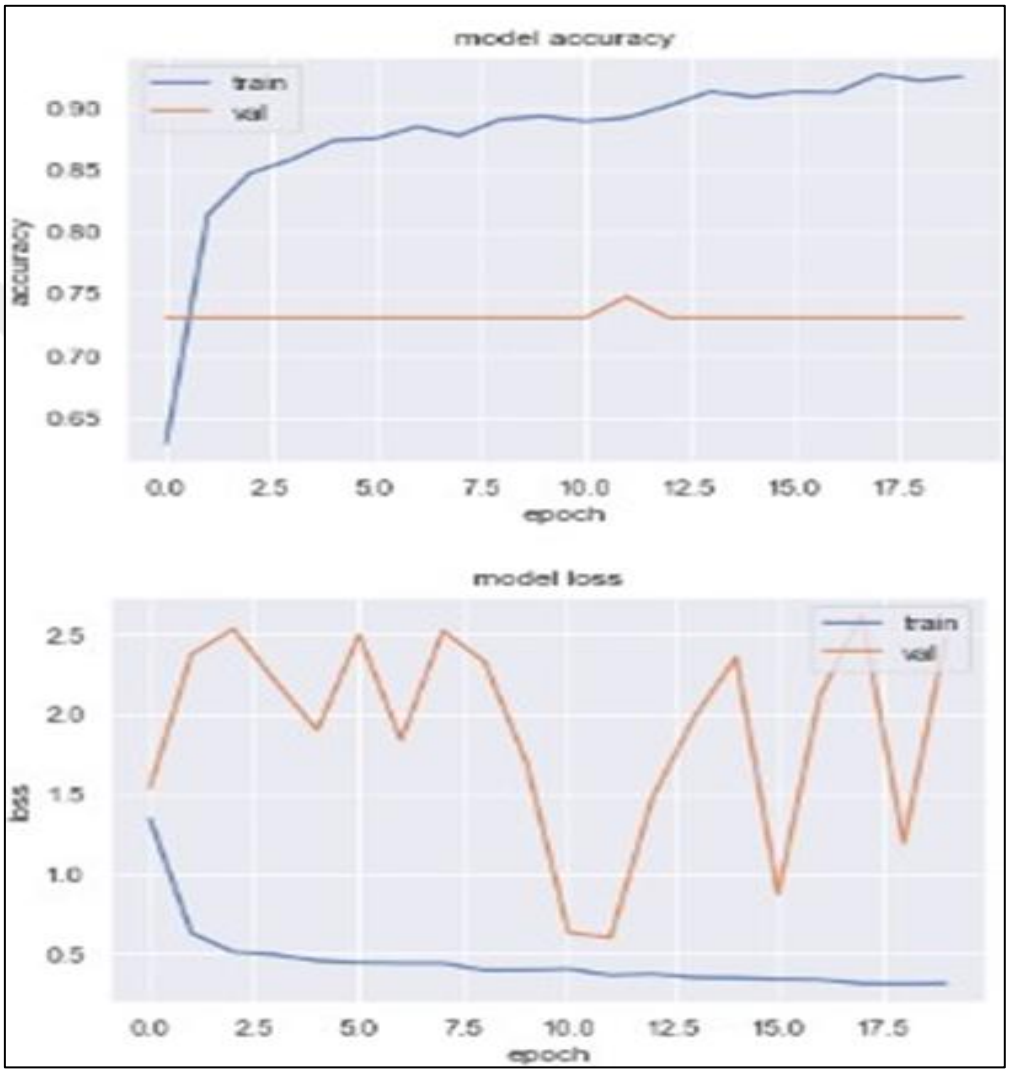


Figure 3.64: Model accuracy and model loss.

H. In the eighth experiment, 11 layers will be used according to the following parameters and with ".6 drop out in last layer as shown in the figure 65.

```
weight_decay = 1e-4
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay), input_shape=[256,256,3]))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.5
model.add(Dropout(0.5))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.6#########
model.add(Dropout(0.6))

model.add(Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
###0.6 ######
model.add(Dropout(0.6))

model.add(Flatten())
model.add(Dense(3, activation='softmax'))

model.summary()
```

Figure 3.65: Layers of eighth experiment.

Figure 66 shows the trainable params of the eighth experiment.

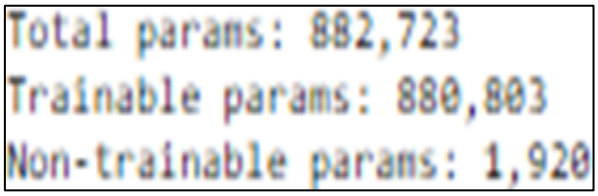


Figure 3.66: Trainable params of the eighth experiment.

We went through all the epochs, and we didn't get to val_accuracy less than ".2 as shown in figure 63. The time required for implementation was also equal to 8745 ".

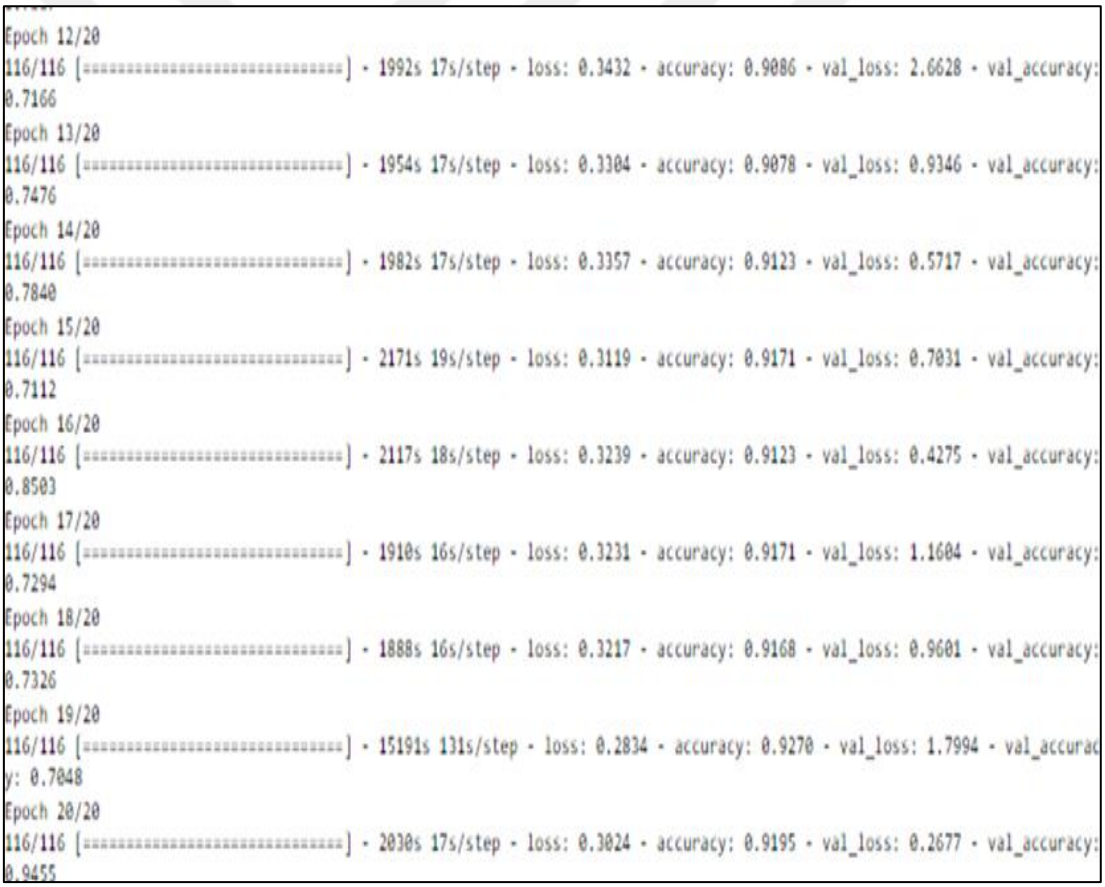


Figure 3.67: Result of eighth experiment.

Figure 68 shows model accuracy and val_accuracy plot and the model loss through epochs in eighth experiment.

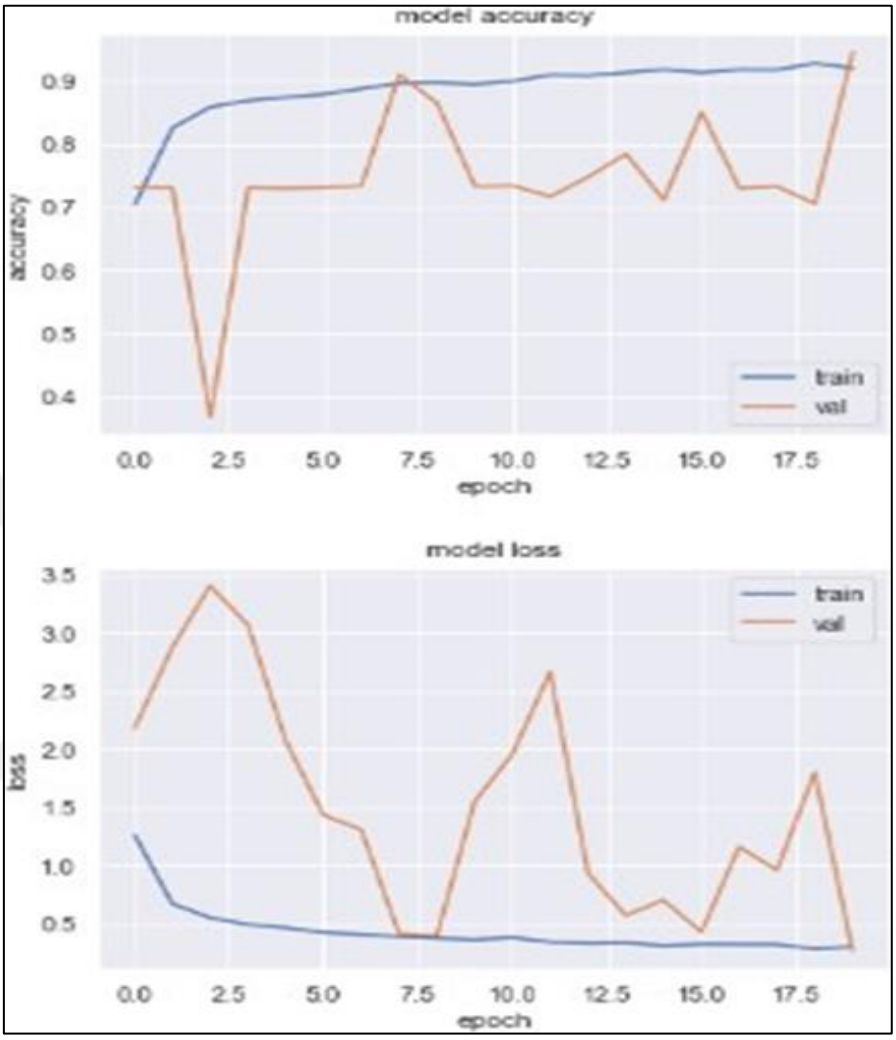


Figure 3.68: Model accuracy and model loss in eighth experiment.

Compare all experiments:

Figure 79 shows the time taken by each experiment

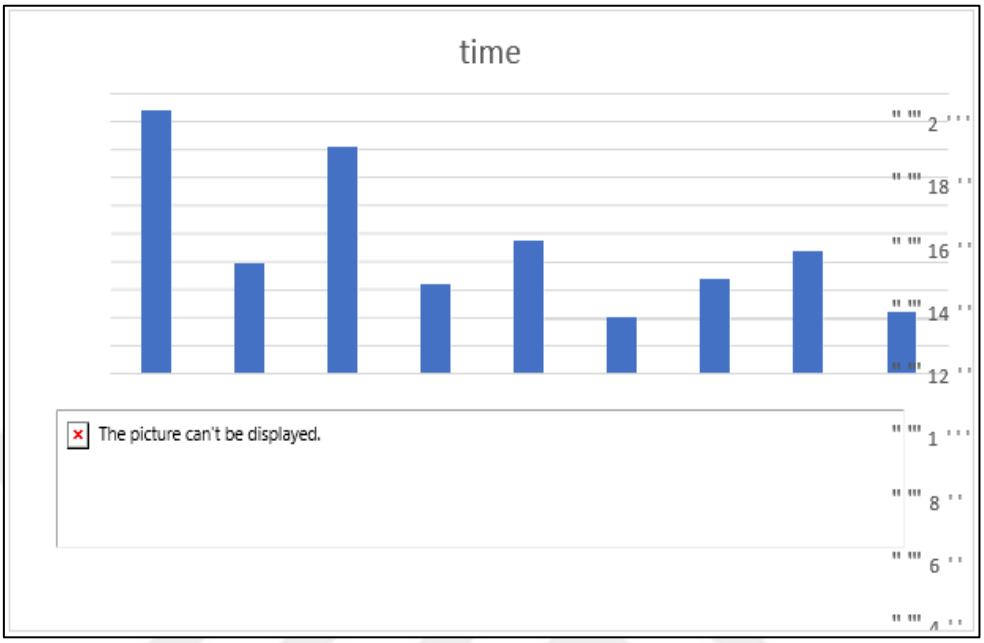


Figure 3.69: Compare time taken by each experiment.

Obviously from figure 69, the sixth experiment took the least time, so it's better than the rest as it only took 4 " "4 " "7 seconds but it did not get to a val_accuracy less than " ".2, In contrast to the zero experiment, which was the second least time, but it reached the required value from Val.

Table 3.1: Compare Experiments in terms of reaches required Val accuracy.

Experiment Number	Reached val_accuracy less than " ".2
" "	Yes
1	No
2	No
3	No
4	No
5	No
6	No
7	No
8	No

Table 3.2: Compare Experiments in terms of number of epochs.

Experiment Number	Number of epochs to reach val_accuracy less than " ".2
" "	14
1	2 " "
2	2 " "
3	2 " "
4	2 " "
5	2 " "
6	2 " "
7	2 " "
8	2 " "

The third table shows a summary of the parameters that were selected in each experiment. The table contains the experiment number, the number of layers, and the maximum percentage of dropout.

Table 3.3: Layers and dropout for each experiment.

Experiment Number	Number of layers	Max droupout
" "	7	" ".4
1	8	" ".4
2	7	" ".5
3	8	" ".6
4	5	" ".3
5	1 " "	" ".5
6	11	" ".5
7	12	" ".5
8	11	" ".6

4. RESULT

In this research, we conduct a practical and programmatic study of a data set concerned with the medical sector in order to show the accuracy of artificial intelligence in determining the nature of the health condition, which in our research deals with issues of chest diseases and the spread of Corona virus in the past period, in addition to working to improve the performance of algorithms in order to reach the desired goal As soon as possible and with the best performance, and therefore we worked on building many experiments to extract the effect of the parameters of the studied algorithm on the performance in terms of loss and the amount of error in prediction. We will list a list of our findings at the end of this research.

Our programming work in this research focused on the use of the Python programming roll, which is the language concerned with issues of artificial intelligence and deep learning with the help of the office group that was mentioned at the beginning of the research. Its specifications will be mentioned in the given table:

Table 4.1: pc resource.

OS	**Windows 1 '64bit**
RAM	8 gb
Processor	Intel core i5 87 ‘" "K
GPU	GeForce GTX 218 '

i. Building code with maximum simplicity and uncomplicatedness.

ii. Limits have been set to stop work in order to reduce the burden and conserve computer resources when the goal assigned to the research is reached.

iii. A program with a high degree of accuracy with a very low error rate has been reached based on a set of scenarios and experiments that simulate several options that can be adopted for deep learning algorithms.

iv. The CNN algorithm is one of the best algorithms designed for artificial intelligence issues related to image analysis due to its flexibility, ease of application, and accuracy resulting from its operation on the dataset passed to it

v. Work has been done on an appropriate and accurate learning rate value that gives the best performance and least loss (“ ". ' '" "3).

vi. This table contain a different between our experiment that tell us about perform, rate of loss and time

Table 4.2: Experiment Number Effectiveness and resource consumption.

Experiment Number	Effectiveness	resource consumption
" "	8 " "%	Normal
1	6 " "%	Normal
2	7 " "%	Low
3	55%	High
4	7 " "%	Normal
5	6 " "%	Low
6	6 " "%	Hight
7	55 %	Low
8	45 %	normal

i. Execution time can be affected by dropout value directly, as it has been observed that increasing this value with increasing layers can lead to an increase in execution time.

ii. The dropout value must be chosen carefully with each layer used.

iii. Based on the previous comparisons (Table 5, Table 6 and Figure 79), the zero experiment, which consists of seven layers, as shown in Figure 42, is better than the rest of the experiments. Therefore, we recommend using this number of layers and

parameters used in the experiment zero, and therefore in order to detect and identify corona from the images.

iv. In pattern recognition and image processing, CNN is a popular and effective recognition technique. It offers several attributes, including flexibility, a straightforward structure, and fewer training requirements. It has grown in popularity in the fields of speech analysis and picture identification.

v. Convolutional Layer (ConvNet/CNN) is a Big Learning method that can take in an output picture, gives meaning (trainable weight vector) to various characteristics and objects in the image, and be able to distinguish between them.

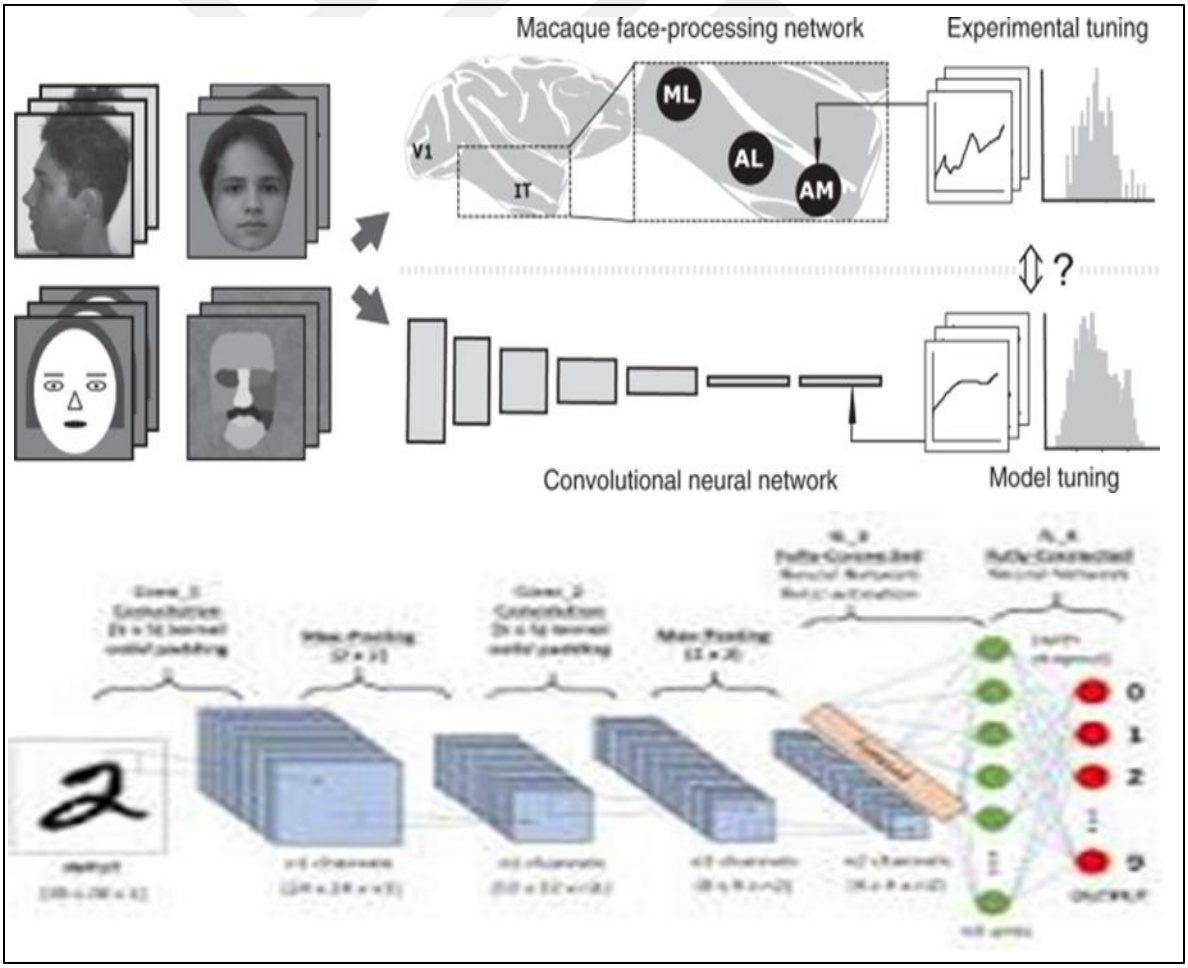


Figure 4.1: Cnn_archeticture.

i. T • In terms of computational complexity, the proposed method is 71.37 percent, 61.82 percent, and 5 " ".78 percent less difficult than even the SR (VDSR) algorithm in the CPU, Pla, and GPU, respectively. This one has a peak transmit ratio loss of " ".49 dB.

ii. • The histogram below illustrates a set of intermediaries that can be used in study to identify disorders.

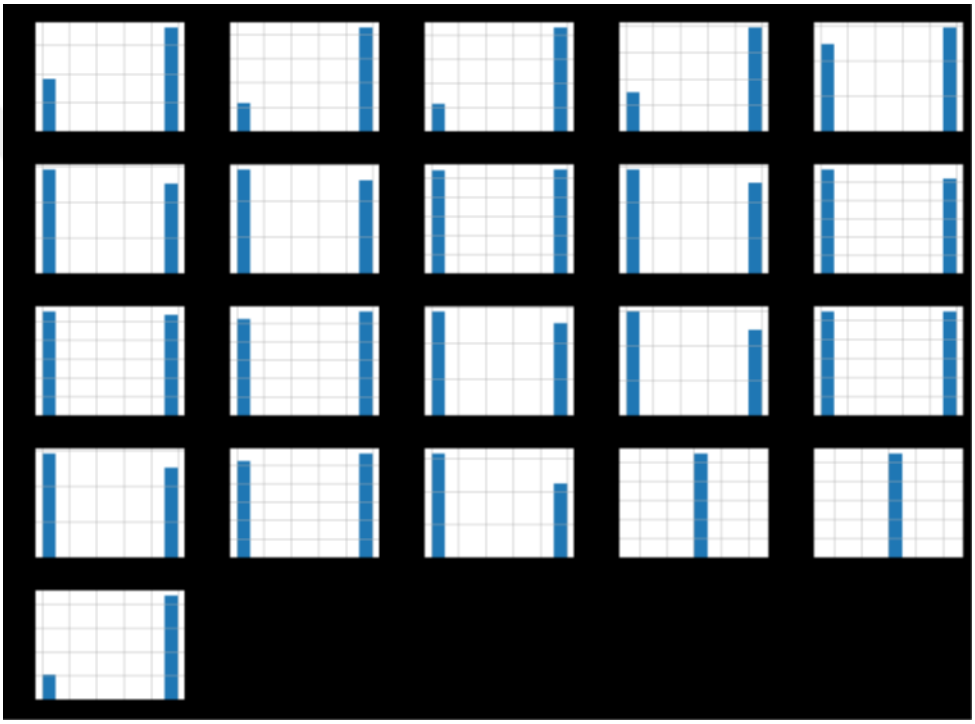


Figure 4.2: Attr_histogram.

5. FUTURE WORK AND RECOMMENDATIONS

1.11 FUTURE WORK

It is possible to work in the Independent on an additional development of the algorithm used in this research so that we improve its work so that it becomes optimal in terms of losses and incorrect expectations, whereby the future endeavors towards making the scale infinitesimally small and the error rate is closer to zero, and this is available if there are many studies working It is not enough for them to be correct, but accurate and appropriate, so that they are neither aware of nor inferior to what is required.

In the future, many studies can be conducted through which experiments are built to compare the performance of algorithms concerned with deep learning approach towards the use of images.

It is possible to work on finding the best version of each algorithm, which leads to the measurement of artificial intelligence using the optimal position of the algorithms, i.e. in their best possible performance.

Here are some of the diagrams and shapes that can be used in the future in preparing extensive studies that include many algorithms. The work is not limited to improving the performance of only one algorithm

Table 5.1: Comp between different algorithms.

	Accuracy	mse	R2 score	Roc score	Running time
KNN	98.37%	2.57	83.1	98.58	**24.252**
Logistic gression	97. " "3%	3. " "36	8 " ". " "89	93.23	**“. " "38**
Random forest	98.39%	2.2 " "7	85.51	97.41	**213.331**

Comparison of metrics for KNN, logistic regression and random forest

It is possible to conduct research based on the detection of corona using vital indicators away from chest images, and here are some schemes that simulate the percentage of sick people in view of the respiratory rate and others

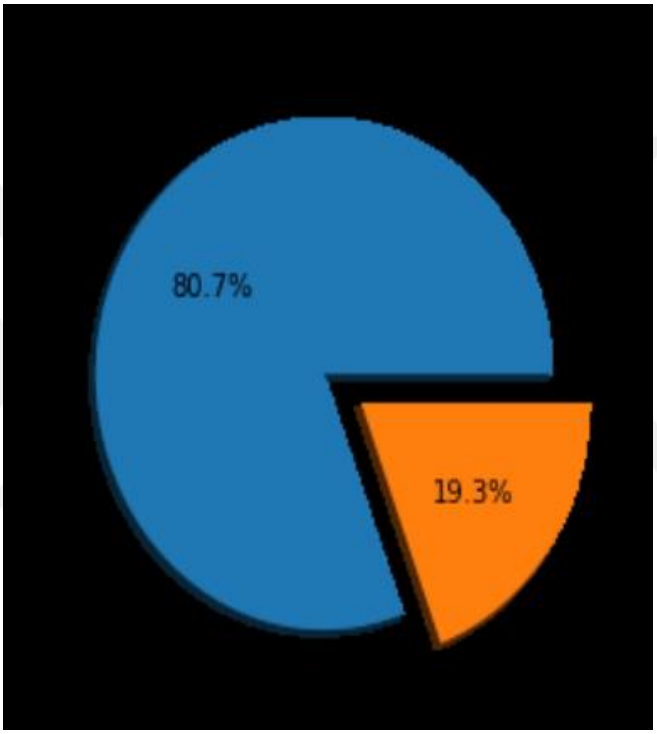


Figure 5.1: Percentage of sick people.

Figure 5.2: Breath_problem.

1.12 RECOMMENDATIONS:

Topics related to artificial intelligence related to physical health remain the subject of researchers' interest, and hundreds of researches are spread for it, and this research is one of them, but it is only a simple contribution to improving the work of an algorithm in order not to waste time and resources in all health sectors.

Basically, this research is directed to artificial intelligence engineers and programmers working in the Python language and interested in deep learning issues, as it contains information based on many experiments, we conducted to reach the best solution to deal with the CNN algorithm, which leads to saving effort, resources and time for researchers in different parts of the earth

This research can be used as a supporting tool for decisionmaking by doctors, with the established model assisting in recognising COVID-19 presence in a person based on them symptoms. Individuals who are suffering COVID-19-related symptoms can also use it to assess if they would be tested positive or negative for COVID-19.

Individuals can quickly determine whether they are at risk of transmitting COVID-19 based on their symptoms.

A. Medical practitioners can employ this test as a primary health assessment for COVID detection.

B. Assisting businesses in limiting physical interaction with clients who may be infected with COVID-19.

Extra information or diagnoses from hospital records, persons who contracted the virus, COVID-19 survivors, patients under assessment, or management can all be included for

future research. A software which can predict the severity of

COVID-19 can indeed be deployed to provide further information

about the steps that must be taken and the interventions

that should be considered.

What is the main advantage of CNN?

Because it employs automation to identify important components without human interaction, CNN has a fundamental edge over its forerunners. It can determine the precise traits of each class for itself using multiple pictures of animals and dogs as that of an example. CNN is successful in calculations.

We may utilize the Adam optimizer since it improved CNN's classification and segmentation capabilities with the greatest accuracy of 99.2%.

An image processing, classification, segmentation, and other auto correlated data are the principal uses for convolutional neural networks (CNNs), which are neural networks with one or more convolutional layers. In essence, a convolution is a filter that is dragged over the input.

REFERENCES

[1] Wu, Fan, et al. "A new coronavirus associated with human respiratory disease in China." Nature 579.7798 (2020): 265-269.

[2] Gallegos, A. "WHO declares public health emergency for novel coronavirus." Medscape Medical News 30.1 (2020).

[3] de Moraes Batista, Andre Filipe, et al. "COVID-19 diagnosis prediction in emergency care patients: a machine learning approach." MedRxiv (2020).

[4] Mondal, M. Rubaiyat Hossain, et al. "Data analytics for novel coronavirus disease." informatics in medicine unlocked 20 (2020): 100374.

[5] Goodman-Meza, David, et al. "A machine learning algorithm to increase COVID-19 inpatient diagnostic capacity." Plos one 15.9 (2020): e0239474.

[6] Schwab, Patrick, et al. "Clinical predictive models for COVID-19: systematic study." Journal of medical Internet research 22.10 (2020): e21439.

[7] Sun, Yinxiaohe, et al. "Epidemiological and clinical predictors of COVID-19." Clinical Infectious Diseases 71.15 (2020): 786-792.

[8] Meng, Zirui, et al. "Development and utilization of an intelligent application for aiding COVID-19 diagnosis." MedRxiv (2020).

[9] Aishwarya, T., and V. Ravi Kumar. "Machine learning and deep learning approaches to analyze and detect COVID-19: a review." SN computer science 2.3 (2021): 1-9.

[10] Al-Awwal, Nasruddeen, et al. "A Review of SARS-CoV-2 Disease (COVID-19): Pandemic in Our Time." Pathogens 11.3 (2022): 368.

[11] Alyasseri, Zaid Abdi Alkareem, et al. "Review on COVID-19 diagnosis models based on machine learning and deep learning approaches." Expert systems 39.3 (2022): e12759.

[12] Ahmad, Bilal, et al. "Brain Tumor Classification Using a Combination of Variational Autoencoders and Generative Adversarial Networks." Biomedicines 10.2 (2022): 223.

[13] Mantas, J. "Unsupervised machine learning for the discovery of latent clusters in COVID-19 patients using electronic health records." The Importance of Health Informatics in Public Health during a Pandemic 272 (2020): 1.

[14] Farooq, Junaid, and Mohammad Abid Bazaz. "A novel adaptive deep learning model of Covid-19 with focus on mortality reduction strategies." Chaos, Solitons & Fractals 138 (2020): 110148.

[15] Yuki, Koichi, Miho Fujiogi, and Sophia Koutsogiannaki. "COVID-19 pathophysiology: A review." Clinical immunology 215 (2020): 108427.

[16] Khan, Irfan Ullah, and Nida Aslam. "A deep-learning-based framework for automated diagnosis of COVID-19 using X-ray images." Information 11.9 (2020): 419.

[17] Loey, Mohamed, Florentin Smarandache, and Nour Eldeen M. Khalifa. "Within the lack of chest COVID-19 X-ray dataset: a novel detection model based on GAN and deep transfer learning." Symmetry 12.4 (2020): 651.

[18] Ma, Qingguo, et al. "A novel recurrent neural network to classify EEG signals for customers' decision-making behavior prediction in brand extension scenario." Frontiers in Human Neuroscience 15 (2021): 610890.

[19] Sahin, M. Emin. "Deep learning-based approach for detecting COVID-19 in chest X-rays." Biomedical Signal Processing and Control 78 (2022): 103977.

[20] Chen, Nanshan, et al. "Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study." The lancet 395.10223 (2020): 507-513.

[21] Zhou, Peng, et al. "A pneumonia outbreak associated with a new coronavirus of probable bat origin." nature 579.7798 (2020): 270-273.

[22] Pan, Feng, et al. "Time course of lung changes on chest CT during recovery from 2019 novel coronavirus (COVID-19) pneumonia." Radiology (2020).

[23] Raja, Ramalingam Karthik, et al. "SARS-CoV-2 and its new variants: a comprehensive review on nanotechnological application insights into potential approaches." Applied Nanoscience (2021): 1-29.

[24] Panahi, Latif, Marzieh Amiri, and Somaye Pouy. "Risks of novel coronavirus disease (COVID-19) in pregnancy; a narrative review." Archives of academic emergency medicine 8.1 (2020).

[25] Wu, Zunyou, and Jennifer M. McGoogan. "Characteristics of and important lessons from the coronavirus disease 2019 (COVID-19) outbreak in China: summary of a report of 72 314 cases from the Chinese Center for Disease Control and Prevention." jama 323.13 (2020): 1239-1242.

[26] Ai, Tao, et al. "Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases." Radiology (2020).

[27] Narin, Ali, Ceren Kaya, and Ziynet Pamuk. "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks." Pattern Analysis and Applications 24.3 (2021): 1207-1220.

[28] Maghdid, Halgurd S., et al. "Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms." Multimodal image exploitation and learning 2021. Vol. 11734. SPIE, 2021.

[29] Bukhari, Syed Usama Khalid, et al. "The diagnostic evaluation of Convolutional Neural Network (CNN) for the assessment of chest X-ray of patients infected with COVID-19." MedRxiv (2020).

[30] Shi, Heshui, et al. "Radiological findings from 81 patients with COVID-19 pneumonia in Wuhan, China: a descriptive study." The Lancet infectious diseases 20.4 (2020): 425-434.

[31] Ogunleye, Adeola, and Qing-Guo Wang. "XGBoost model for chronic kidney disease diagnosis." IEEE/ACM transactions on computational biology and bioinformatics 17.6 (2019): 2131-2140.

[32] Feng, Chiyu, et al. "Deep learning framework for Alzheimer's disease diagnosis via 3D-CNN and FSBi-LSTM." IEEE Access 7 (2019): 63605-63618.

[33] Yin, Hongxu, et al. "DiabDeep: Pervasive diabetes diagnosis based on wearable medical sensors and efficient neural networks." IEEE Transactions on Emerging Topics in Computing 9.3 (2019): 1139-1150.

[34] Santosh, K. C. "AI-driven tools for coronavirus outbreak: need of active learning and cross-population train/test models on multitudinal/multimodal data." Journal of medical systems 44.5 (2020): 1-5.

[35] Wynants, Laure, et al. "Systematic review and critical appraisal of prediction models for diagnosis and prognosis of COVID-19 infection." MedRxiv (2020).

[36] Alazab, Ammar, et al. "Using feature selection for intrusion detection system." 2012 international symposium on communications and information technologies (ISCIT). IEEE, 2012.

[37] Alazab, Mamoun, et al. "Cybercrime: the case of obfuscated malware." Global security, safety and sustainability & e-Democracy. Springer, Berlin, Heidelberg, 2011. 204-211.

[38] Alazab, Mamoun, et al. "Information security governance: the art of detecting hidden malware." IT security governance innovations: theory and research. IGI Global, 2013. 293-315.

[39] Alazab, Ammar, et al. "Web application protection against SQL injection attack." Proceedings of the 7th International Conference on Information Technology and Applications. 2011.

[40] Alazab, Moutaz, and Lynn M. Batten. "Survey in smartphone malware analysis techniques." New threats and countermeasures in digital crime and cyber terrorism (2015): 105-130.

[41] Alazab, Moutaz, Ammar Alazab, and Lynn Batten. "Smartphone malware based on synchronisation vulnerabilities." ICITA 2011: Proceedings of the 7th International Conference on Information Technology and Applications. ICITA, 2011.

[42] Moonsamy, Veelasha, Moutaz Alazab, and Lynn Batten. "Towards an understanding of the impact of advertising on data leaks." International journal of security and networks 7.3 (2012): 181-193.

[43] Batten, Lynn M., Veelasha Moonsamy, and Moutaz Alazab. "Smartphone applications, malware and data theft." Computational intelligence, cyber security and computational models. Springer, Singapore, 2016. 15-24.

[44] Alazab, Moutaz, et al. "Analysis of malicious and benign android applications." 2012 32nd International Conference on Distributed Computing Systems Workshops. IEEE, 2012.

[45] Xu, Yuan, et al. "Medical breast ultrasound image segmentation by machine learning." Ultrasonics 91 (2019): 1-9.

[47] Mesleh, Abdelwadood, et al. "Heart rate extraction from vowel speech signals." Journal of computer science and technology 27.6 (2012): 1243-1251.

[48] Mesleh, Abdelwadood. "Support vector machines based Arabic language text classification system: feature selection comparative study." Advances in Computer and Information Sciences and Engineering. Springer, Dordrecht, 2008. 11-16.