HYBRID DEEP LEARNING-BASED MODEL FOR COVID-19 PREDICTION AND INTERPRETATION USING MULTIPLE DATA MODALITIES

DOKUN OYEWOLE, BTech, M.Sc.

20194197148

A DISSERTATION SUBMITTED TO
POST GRADUATE SCHOOL
FEDERAL UNIVERSITY OF TECHNOLOGY, OWERRI

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DOCTOR OF PHILOPHY (Ph.D) DEGREE IN INFORMATION MANAGEMENT TECHNOLOGY

MAY, 2024

CERTIFICATION

This is to certify that this thesis titled **"Hybrid Deep Learning-Based Model for COVID-19 Prediction and Interpretation Using Multiple Data Modalities"** was carried out by **Dokun Oyewole** with the Registration number **20194197148** and submitted to the Department of Information Technology, School of Information and Communication Technology, Federal University of Technology, Owerri in partial fulfilment of the requirement for the award of Doctor of Philosophy (Ph.D) in Information Management Technology.

Prof. (Mrs.) U. F. Eze (Lead Supervisor)	30/07/24 **Date**
Dr. C. Ikerionwu (Co-supervisor)	30/07/24 **Date**
Dr. A. M. John-Otumu (Co-supervisor)	30/7/24 **Date**
Dr. A. I. Otuonye (Head of Department)	**Date**
Prof. (Mrs.) U. F. Eze (Dean, SICT)	30/07/24 **Date**
Prof. (Mrs.) J. N. Nwosu (Dean, Post Graduate School)	**Date**
Prof. Awodele Oludele External Examiner	22/05/2024 **Date**

iii

INFINIX
AI CAMERA

DEDICATION

This research is dedicated to the Divine source of my strength, guidance, and inspiration throughout my academic journey. I am deeply grateful for the blessings that enabled me to pursue this work. It is also dedicated to all those affected by the COVID-19 pandemic, in the hope that this research contributes to knowledge and solutions for this global crisis.

ACKNOWLEDGEMENTS

This research is a culmination of support and contributions from many individuals. My deepest gratitude goes to my exceptional supervisor, Prof. (Mrs) U. F. Eze, whose guidance, mentorship, and insightful feedback shaped the entire research process. I also extend my thanks to my co-supervisors, Dr. Charles Ikerionwu and Dr. Adetokunbo MacGregor John-Otumu, for their invaluable critiques and contributions that improved the research methodology and enriched the findings.

Furthermore, I am grateful to Dr. A. I. Otuonye, (Head of Department), Dr. C. Etus (PG Coordinator), and the department's lecturers, including Prof. B. Asiegbu, Engr. Dr. O. C. Nwokonkwo, Engr. Dr. E. C. Amadi, Dr. N. Ekedebe, Engr. Dr. E. M. Nwanga, and others, for their teachings and expertise that broadened my knowledge. The administrative staff also deserves my appreciation for their continuous support.

Special thanks go to Mr. Tunde Dokun and Mr. Emmanuel Akogun for their meticulous work in formatting and typesetting the manuscript.

Finally, and most importantly, I express my profound gratitude to God for His unwavering blessings, guidance, and inspiration throughout this journey. This research would not have been possible without the collective support of these individuals, and for that, I am truly grateful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This research addresses the critical need for accurate and timely COVID-19 diagnosis and prognosis by developing a hybrid deep learning model that integrates multiple data modalities, including chest X-rays, Computed Tomography (CT) scans, blood smears, and clinical data. The model employs specialized architectures such as Residual Network with 50 Layers (ResNet50) for Chest X-ray, InceptionV3 for CT scans, Convolutional Neural Network (CNN) for blood smears, and a Random Forest classifier for clinical data analysis. The results demonstrate high accuracy rates: 96.7% for ResNet50, 97.58% for InceptionV3, 96.12% for CNN, and 98.30% for the Random Forest classifier. Grad-CAM enhances transparency by visualizing critical regions in the images, aiding healthcare professionals in understanding the model's decisions. This hybrid model offers improved accuracy and reliability for COVID-19 diagnosis and prognosis, making it a valuable tool for clinical settings and resource allocation. The research underscores the potential of multi-modal data integration in medical AI and suggests further exploration and refinement of such models for broader healthcare applications.

Keywords: Deep Learning, ML, Explainable AI, Classification, COVID-19, GRAD-CAM

CHAPTER ONE

INTRODUCTION

1.1 Background Information

The COVID-19 pandemic has been one of the most significant global events in recent memory, with millions of casualties worldwide. The SARS-CoV-2 virus, which causes the disease, was initially identified in December 2019 in Wuhan, China (Awwalu et al., 2020). Since then, it has quickly spread around the globe, bringing about massive illness, fatalities, and societal upheaval (Muhammad et al., 2022). According to Sait et al. (2021), the Centers for Disease Control and Prevention states that contact with infected surfaces and respiratory droplets is the main way in which the virus is transmitted.

Mild to severe COVID-19 symptoms can include fever, coughing, and breathing difficulties. In elderly adults and those with underlying medical issues, the disease can be more severe (Harmon et al., 2020). Governments and public health organizations all around the world have put in place a variety of measures to halt the virus's spread in reaction to the pandemic. The World Health Organization (WHO) included travel bans, social separation rules, lockdowns, and vaccination drives (WHO, 2021).

Since the start of the pandemic, COVID-19 has been the subject of continuous research. In order to understand the biology of the virus, manufacture safe and effective vaccinations, and develop viable therapies, scientists have been working (WHO, 2020b). This procedure has been fraught with difficulties, such as the requirement for quick development and the appearance of new virus types (Pola, 2021).

Numerous firms were being forced to close or shut-down and unemployment rates rose up at the pick of the COVID-19 pandemic which had a substantial effect on the world economy (Rani et al., 2022).

Additionally, it has brought attention to the disparities in access to healthcare and education and has disproportionately affected underserved populations (Shikang et al., 2021).

Globally, the COVID-19 pandemic has resulted in extraordinary illness and mortality rates, turning it into a serious public health emergency. The World Health Organization (WHO) estimates that as of May 2023, there were more than 400 million confirmed cases of COVID-19 worldwide, resulting in more than 6 million fatalities (Shahanaz et al., 2022). It has become much more difficult to contain the pandemic due to the constant threat of new variations and mutations (Chauha & Modi, 2022). As a result, there is an immediate need for reliable COVID-19 prediction models that can help with rapid disease diagnosis, treatment, and control (Shikang et al., 2021; Chauha & Modi, 2022; Afreen & Reddy, 2022; Shahanaz et al., 2022).

According to Guo et al. (2020), the most common COVID-19 severe symptoms are fever, dry cough, tiredness, headache, sore throat, sneezing, vomiting, dyspnea, myalgia, nasal congestion, and rhinorrhea. It has been recorded that people with severe COVID-19 infection suffer from acute kidney impairment, cardiac damage, pulmonary edema, septic shock, and other devastating effects (Guo et al., 2020).

The nasopharyngeal swab test, the Rapid Antigen Test (RAnT), and Reverse Transcription-Polymerase Chain Reaction (RT-PCR) are among the laboratory testing procedures for the COVID-19 virus that are listed in the literature study (Sait et al., 2021; Soumyajit, Somnath, & Arijit, 2021).

An RT-PCR test, a popular variation of the Nucleic Acid Amplification Test (NAAT), is the current Gold Standard for detecting COVID-19 (Sait et al., 2021). According to Sait et al. (2021), a complicated chain of biochemical reactions directly detects the viral genome by converting the RNA in the collected nasopharyngeal swab sample to DNA by reverse transcription and expanding the complementary DNA strands of the virus.

In order to determine COVID-19 infection, this test looks for the distinctive genetic makeup of the virus. Although this test has the highest accuracy and precision, it is expensive, time-consuming, and difficult to get (Pola, 2021). This sample testing technique also requires expensive infrastructure and skilled lab staff to be carried out successfully (Sait et al., 2021; Afreen & Reddy, 2022; Ghomi et al., 2020). These limitations prevent it from being employed as a widely available confirmatory test, which would allow other rapid assays to detect COVID-19 (Sait et al., 2021).

Building more clinical facilities is essential since, according to Subhalakshmi et al. (2022), it is challenging to diagnose the symptoms of these infections with the available medical services. Social isolation, mask wearing, regular hand washing, and self-quarantining are still advised as better therapies to avoid COVID-19, which is also done by many other countries, despite the extensive attempts made by developers to provide a correct diagnosis.

Artificial Intelligence (AI) and Machine Learning (ML) techniques have showed great potential for developing effective COVID-19 prediction models (Afreen & Reddy, 2022; Shahanaz et al., 2022). Deep learning algorithms have demonstrated exceptional accuracy and robustness, particularly in predicting the spread of COVID-19 and identifying patients at high risk of serious illness (Bashar et al., 2021; Vedika et al., 2022).

Deep learning algorithms have been investigated for COVID-19 prediction in a number of recent publications (Bashar et al., 2021; Vedika et al., 2022; Afreen & Reddy, 2022; Shahanaz et al., 2022; Chauha & Modi, 2022). For instance, based on chest CT images, a study by Abbasi-Kesbi et al. (2022) suggested a deep neural network model for COVID-19 prediction.

The study's ability to distinguish COVID-19 cases from healthy controls had an accuracy rate of 89.5%. A deep learning-based model for COVID-19 prediction utilizing chest CT scans and clinical data was proposed in a subsequent work by Hu et al. (2022). The study found that diagnosing COVID-19 cases had a high accuracy rate of 93.9%.

These studies, however, only used one type of data, which would have limited their generalizability and accuracy rates. For COVID-19 prediction, however, a number of recent researches have looked into the use of multi-modal data. For instance, Gao et al. (2021) suggested a COVID-19 prediction model that incorporates chest CT pictures, clinical data, and demographic characteristics.

The study found that detecting COVID-19 patients had a 95.7% accuracy rate. Another work by Liu et al. (2021) developed a deep learning model for COVID-19 prediction that combines chest CT scans, clinical information, and laboratory results. The study found that diagnosing COVID-19 cases had a high accuracy rate of 93.6%. These results imply that the accuracy and resilience of COVID-19 prediction models may be enhanced by the integration of data from other modalities.

Hence, by creating an integrated intelligent deep learning model that can precisely predict COVID-19 utilizing a variety of data modalities, the proposed research attempts to build on these research works. The model will be tested on a separate testing set after being validated on a sizable dataset of COVID-19 cases and healthy controls. Additionally, the research will look into how different data modalities affect the model's performance and look into ways to make the model easier to understand.

Generally, the proposed research has the potential to help create COVID-19 prediction models that are more reliable and accurate, which can help with early disease detection, treatment, and management with interpretability features that will explain why and what led the proposed model’s to perform its classification and final prediction.

1.2 Problem Statement

The COVID-19 pandemic has become a severe global health crisis, with unprecedented rates of illness and death. The emergence of new variants and mutations has further complicated the accurate prediction of the disease, despite ongoing efforts to control it.

Although current deep learning-based models for COVID-19 prediction have shown high accuracy (Gao et al., 2021; Hu et al., 2022), their reliability is challenged by disease mutations (Nandi & Mulimani, 2021; Al-Shehri et al., 2022; Chokri et al., 2022) and the types and quantities of samples used for testing (Hu et al., 2022; Gao et al., 2021).

Most existing models focus on single data types, such as clinical data or chest X-rays (Sait et al., 2021; Afreen & Reddy, 2022; Ghomi et al., 2020), which may limit their accuracy and generalizability. Additionally, while a few researchers have explored using multiple data modalities (Hu et al., 2022; Gao et al., 2021), these models often lack interpretability.

This research aims to develop an integrated deep learning model that uses multiple data modalities and includes features for interpretability, which will support more accurate and explainable predictions for COVID-19, aiding in prompt diagnosis, treatment, and control.

1.3 Research Objectives

The main objective of this study is to develop a hybrid deep learning based model for COVID-19 prediction and interpretation using multiple data modalities, such as chest X-rays, CT scan images, blood smear images and clinical information.

The specific objectives of the study are to:

1. Design a deep learning model to classify and interpret COVID-19 using preprocessed chest X-ray images, CT scans, blood smears, and clinical data from both COVID-19 positive and

healthy cases, incorporating various convolutional neural networks and explainable AI methods like Grad-CAM.

2. Transform the architecture into an implementable platform for the classification and interpretation of COVID-19 using python programming language.
3. Train and test the developed model using the multiple data modalities gathered
4. Evaluate the performance of the proposed deep learning model developed using standard measuring metrics such as accuracy, recall, precision, f1-score and ROC/AUC curve.

1.4 Research Questions

The proposed research aims to address the following research questions:

1. To what extent can a deep learning model accurately classify and interpret COVID-19 using preprocessed chest X-ray images, CT scans, blood smears, and clinical data?
2. To what extent do various convolutional neural network variants and explainable AI methods like Grad-CAM improve the interpretability of COVID-19 predictions?
3. To what extent can the deep learning architecture be successfully implemented as a functional platform for COVID-19 classification and interpretation using Python?
4. To what extent does the performance of the developed model, evaluated with metrics such as accuracy, recall, precision, F1-score, and ROC/AUC curve, exceed that of existing models?

1.5 Justification of the Study

The need for precise and transparent COVID-19 predictions serves as a justification or reason for the study on an explainable deep learning model for COVID-19 prediction and interpretation employing different data modalities (Hu et al., 2022; Gao et al., 2021; Bashar et al., 2021; Vedika et al., 2022; Shahanaz et al., 2022; Chauha & Modi, 2022; Abbasi-Kesbi et al., 2022).

Traditional machine learning and deep learning models frequently have difficulty being interpreted, which makes it difficult for healthcare personnel to comprehend and believe the predictions (Afreen

& Reddy, 2022; Shahanaz et al., 2022). Healthcare practitioners can obtain understanding of the model's decision-making process and trust in the prediction based on the development of an explainable deep learning model.

Furthermore, using a variety of data modalities, including chest X-rays, CT scan images, clinical data, and lab results, enables a more thorough and precise prediction of COVID-19 (Hu et al., 2022; Gao et al., 2021).

By utilizing the advantages of each modality (Hu et al., 2022; Gao et al., 2021), this proposed study intends to improves prediction accuracy and confidence rates and also makes it easier to understand the model's predictions.

The study tackles the urgent need for accurate and understandable COVID-19 prediction models, which will improve patient care and decision-making in the context of the continuing epidemic.

1.6 Scope of the Study

An extensive dataset of COVID-19 cases and healthy controls, comprising chest x-rays, CT scan images, blood smear and clinical data were gathered and preprocessed as part of this study. In order to effectively diagnose COVID-19 utilizing multiple data modalities, the study developed an integrated intelligent deep learning model that incorporated more than two deep learning techniques, such as Convolutional Neural Networks (CNNs), GRAD-CAM, etc.

By contrasting the integrated intelligent deep learning model with current COVID-19 prediction models based on single and multiple modalities of data, the study will also assess the performance of the model.

The study looked into how various data modalities affect the reliability and accuracy of the COVID-19 prediction model and determine which modalities are most useful for precise COVID-19 prediction. The study also improved the interpretability of the COVID-19 prediction model by building explainable AI (XAI) model using GRAD-CAM technique, which can aid doctors and healthcare professionals in understanding the model's predictions and decision-making,

Data from a variety of sources, including hospitals, clinics, and research organizations were used to perform the study. It is noteworthy to emphasize that the study did not entail direct patient contact or physical examination, and was restricted to data found in electronic health records. Last but not least, the study was time-bound and only created and evaluated an integrated intelligent deep learning model for COVID-19 prediction employing a variety of data modalities.

CHAPTER TWO

LITERATURE REVIEW

2.1 Conceptual Framework

This section presents the fundamental ideas, factors, and connections pertinent to the research study. It serves as a structure and guiding framework for the research, by facilitating the organization and analysis of data, identifying areas of knowledge gaps, formulation of research questions and enhances the reliability and validity of the various research findings.

2.1.1 Historical Development of COVID-19

Up until 2021, the following significant dates and occasions have occurred in relation to COVID-19's history (WHO, 2021):

(a) Wuhan, Hubei Province, China, reports the first cases of a strange respiratory ailment in December 2019.

(b) The World Health Organization (WHO) learns about the outbreak in January 2020 and starts keeping an eye on the situation. The condition, officially known as COVID-19, was determined to be caused by the new coronavirus.

(c) Due to COVID-19's quick global spread, the WHO declared it a global pandemic on March 11, 2020.

(d) In reaction to COVID-19, the United States declared a national emergency on March 13, 2020.

(e) In April 2020, there were more than a million verified COVID-19 cases worldwide.

(f) In July 2020, pharmaceutical companies and academic institutions start creating and running COVID-19 vaccination clinical trials.

(g) COVID-19 immunization efforts begin in a number of nations in December 2020, beginning with high-risk groups and healthcare professionals.

(h) In January 2021, other COVID-19 variations, including those from the UK (B.1.1.7) and South Africa (B.1.351), were discovered, raising worries about their higher transmissibility.

(i) May 2021: The World Health Assembly backed initiatives for a thorough and impartial study of the COVID-19 origins.

(j) September 2021: Worldwide COVID-19 vaccination campaigns continue, and attempts to resolve access and distribution disparities for vaccines were ongoing.

2.1.2 COVID-19 Variants

The following are some significant COVID-19 variants discovered (WHO, 2021)

B.1.1.7 (UK Variant): In September 2020, researchers in the United Kingdom discovered the variant for the first time. It has mutations in the virus's spike protein and is linked to greater transmissibility. The picture is displayed in Figure 2.1

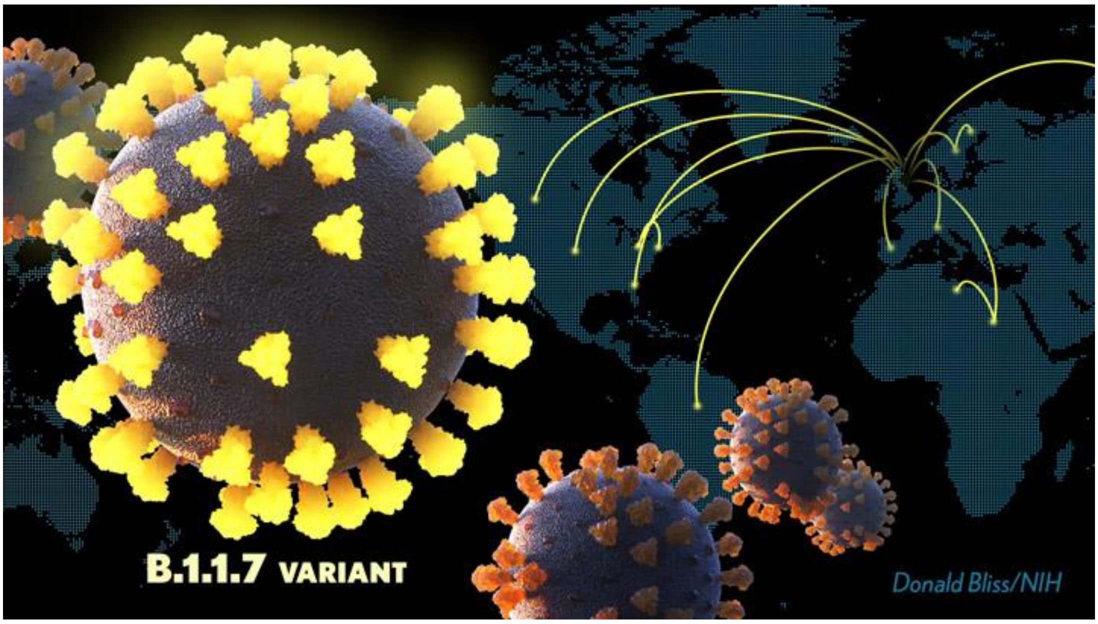

Figure 2.1: B.1.1.7 (UK Variant) (WHO, 2021)

B.1.351 (South African Variant): In October 2020, South Africa reported the discovery of this variant. It also has higher transmissibility and shares certain alterations with the B.1.1.7 variation (See Figure 2.2).

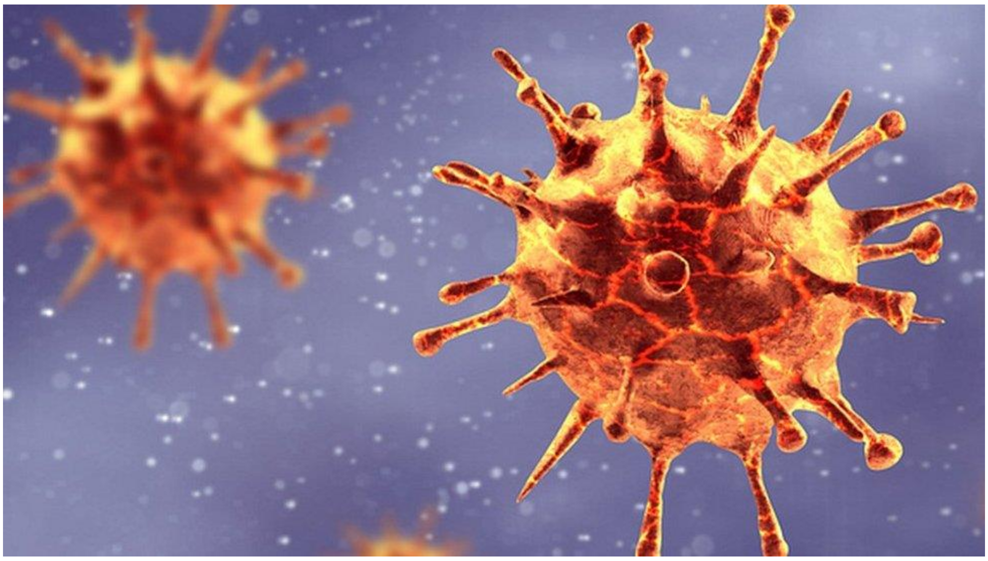

Figure 2.2: B.1.351 (South African Variant) (WHO, 2021)

P.1 (Brazilian variety): This variety was first discovered in Brazilian travelers, and it carries a number of alterations that may affect its transmissibility and immunity-evading capabilities (See Figure 2.3).

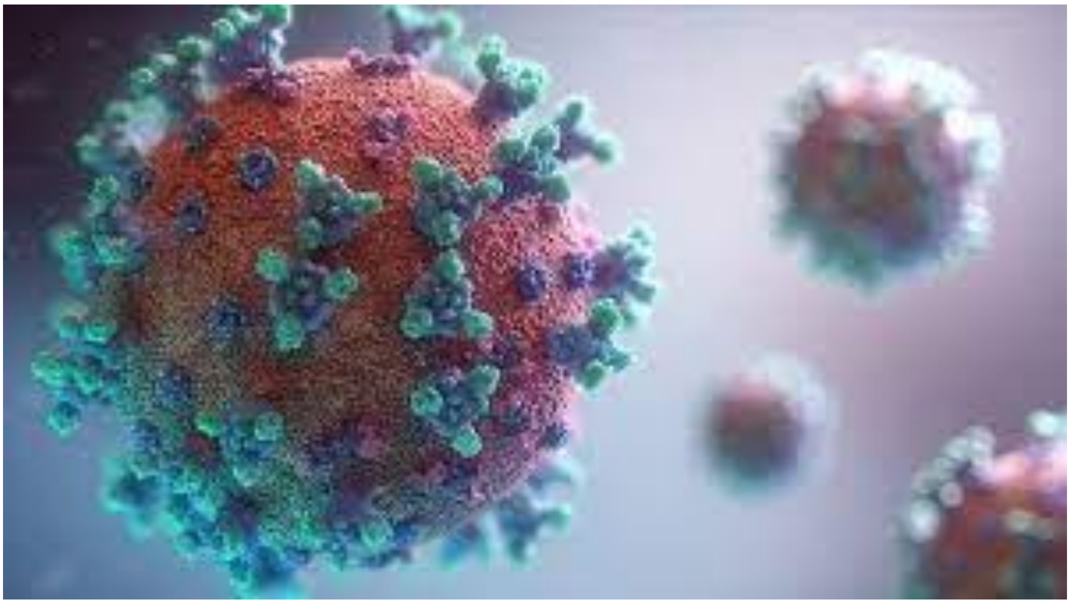

Figure 2.3: P.1 (Brazilian variety) (WHO, 2021)

Indian Variants of B.1.617: There are several sub-lineages that make up the B.1.617 variation, including B.1.617.1, B.1.617.2, and B.1.617.3. These variations, which were initially found in India, are of concern because of the possibility of increased transmissibility and potential effects on vaccination efficacy (See Figure 2.4).

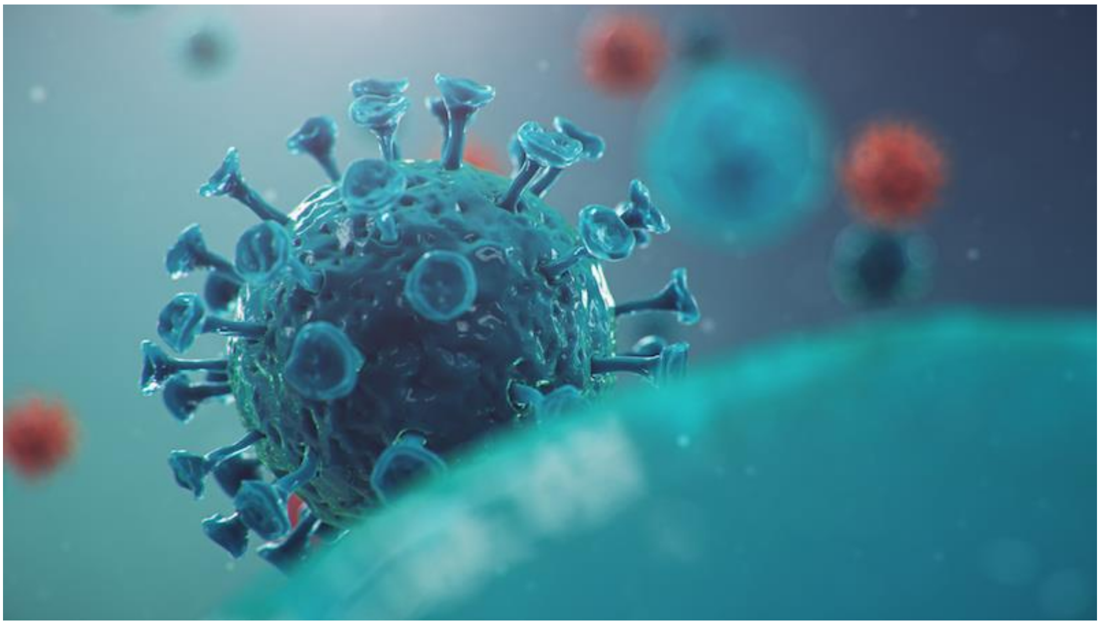

Figure 2.4: Indian Variants of B.1.617 (WHO, 2021)

It is important to keep in mind that knowledge about these variants is always evolving and will aid in selecting the appropriate datasets to be used during the collection, preprocessing, training, and testing phases of the proposed model.

2.1.3 Conventional Approaches for Diagnosing COVID-19

Clinical assessment, laboratory testing, and medical imaging are frequently combined in conventional methods for diagnosing COVID-19 (Sait et al., 2021). The following are a few of the conventional COVID-19 diagnostic methods:

Polymerase Chain Reaction (PCR) Testing: The most popular and accurate way to diagnose COVID-19 is by PCR testing. It looks for genetic material (RNA) from the SARS-CoV-2 virus in respiratory samples, which are normally taken from the nose or throat. PCR tests are extremely sensitive and can detect infections that are active with accuracy. The PCR machine is shown in Figure 2.5

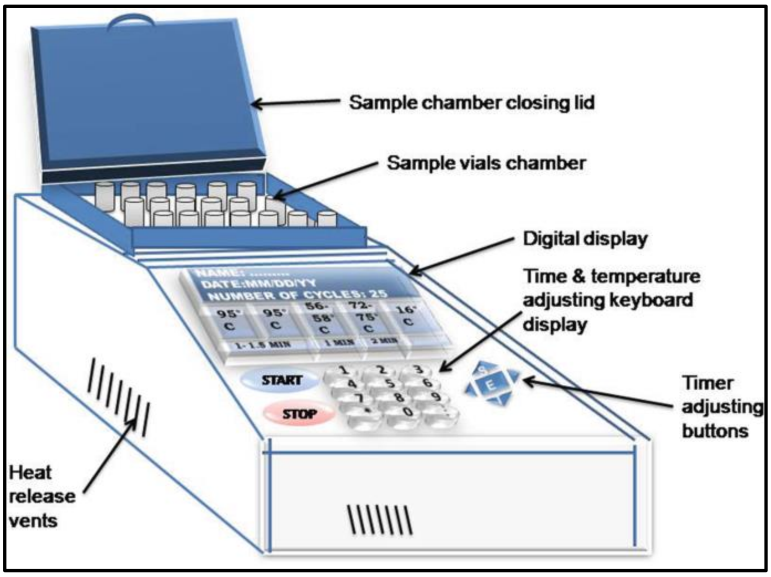

Figure 2.5: PCR Machine (WHO, 2021)

Antigen Testing: SARS-CoV-2 specific viral proteins can be found using fast diagnostic antigen assays. While they deliver quicker answers than PCR testing, they are typically less sensitive. In some situations, antigen tests are helpful for promptly identifying infected people for screening purposes (See Figure 2.6).

Figure 2.6: Antigen machine (WHO, 2021)

Serological testing for antibodies: Serological testing for antibodies identifies antibodies produced by the immune system in response to a SARS-CoV-2 infection (See Figure 2.7). These examinations on blood samples can reveal whether a person has already been exposed to the virus. While not used for early diagnosis, antibody tests can be useful in determining how common previous infections are in a group.

Figure 2.7: Serological testing machine (WHO, 2021)

Chest X-rays: These imaging procedures, which are frequently used on COVID-19 patients, can detect lung abnormalities such as pneumonia or lung inflammation. When paired with clinical symptoms and the findings of laboratory tests, X-rays can support the diagnosis by analyzing the severity and development of the condition (See Figure 2.8).

Figure 2.8: Chest X-ray machine (WHO, 2021)

Computed tomography (CT) scans offer fine-grained cross-sectional images of the lungs and are capable of spotting lung anomalies related to COVID-19, such as ground-glass opacities and lung consolidation (See Figure 2.9). Due to its increased cost, radiation exposure, and risk for false positive results, CT scans are not advised as the primary diagnostic technique even if they are sensitive in detecting lung abnormalities.

Figure 2.9: Computed Tomography (CT) Scan (WHO, 2021)

It is crucial to keep in mind that these conventional diagnostic methods have their limitations, and the precision of the diagnosis depends on elements like testing time, sample quality, and individual variances. To accurately diagnose COVID-19, medical practitioners and health authorities often use a mix of clinical assessment, laboratory tests, and imaging.

2.1.4 Application of Deep Learning frameworks for Diagnosis of COVID-19

In COVID-19 diagnosis, deep learning frameworks are being utilized to evaluate medical data and find patterns connected to the condition. These frameworks use deep neural networks that have been trained on vast datasets to detect COVID-19 situations with accuracy and efficiency. They can combine different data modalities and offer a thorough method of diagnosis. The use of deep learning in COVID-19 diagnosis has potential for enhancing detection and enabling prompt patient interventions.

Multiple steps are involved in the deep learning framework for COVID-19 diagnosis. First, pertinent medical information, such as chest X-rays, CT scans, clinical records, and laboratory findings, are gathered. To ensure consistency and compatibility with deep learning models, the obtained data is subsequently preprocessed. For COVID-19 diagnosis, a suitable deep learning framework is chosen, and unique model architecture is created. The preprocessed data is used to train the model, and performance measures are used to assess it. The model can be used in clinical settings after validation. Updates and improvements are made continuously to improve the performance of the model.

In general, using deep learning frameworks enhances the accuracy and interpretability of COVID-19 diagnostics compared to traditional Artificial Neural Networks (ANNs). Deep learning models, such as convolutional neural networks (CNNs), excel at processing complex data types, including chest X-ray and CT scan images. These models can automatically extract relevant features, identifying patterns that might be subtle or overlooked by human experts, which is crucial for detecting the varied and nuanced signs of COVID-19.

Traditional ANNs, while capable of performing classification tasks, lack the sophisticated architecture needed to efficiently handle large-scale image data. They typically require manual feature extraction, which can be both time-consuming and incomplete, potentially missing critical information. Additionally, ANNs may struggle with the extensive datasets necessary for accurate COVID-19 prediction, resulting in lower performance compared to deep learning models.

Furthermore, deep learning frameworks can incorporate explainable AI techniques like Grad-CAM, which highlight the areas of an image that most influenced the model's decision. This feature provides medical professionals with valuable insights into the model's reasoning, enhancing trust and interpretability qualities that traditional ANNs generally lack. While ANNs can still be employed for COVID-19 prediction, they do not offer the same level of detailed analysis or transparency as advanced deep learning models. Thus, deep learning frameworks present a more robust and effective approach for COVID-19 diagnostics, enabling more accurate and interpretable predictions.

2.1.5 Types of COVID-19 Datasets

These are some illustrations of the several COVID-19 dataset types that are offered. The particular datasets that are used for research and analysis rely on the goals of the study and the data sources that are available.

(a) Epidemiological Datasets: These datasets offer details on COVID-19 transmission and dispersal at local, planetary, or global scales. They frequently contain statistics on the quantity of verified cases, deaths, recoveries, testing rates, hospitalizations, and demographics.

Table 2.1: Sample Epidemiological datasets for COVID-19

S/N	Confirmed	Deaths	Recovered	day_cont	province_country
1:	263292	19413	0	91	New York, US
2:	208389	21717	85915	91	Spain
3:	155860	21340	40657	91	France
4:	150648	5279	99400	91	Germany
5:	133495	18100	0	91	UK
6:	98674	2376	16477	91	Turkey
7:	95914	5150	0	91	New Jersey, US
8:	85996	5391	63113	91	Iran
9:	68128	4512	63519	91	Hubei, Mainland China
10:	66236	12213	19526	88	Lombardia, Italy
11:	57999	513	4420	91	Russia
12:	45757	2906	25318	91	Brazil
13:	42944	2182	0	91	Massachusetts, US
14:	41889	6262	9433	91	Belgium
15:	37344	1421	0	91	California, US
16:	36082	1673	0	91	Pennsylvania, US
17:	35107	1565	0	91	Illinois, US
18:	34842	4054	0	91	Netherlands
19:	33966	2813	0	91	Michigan, US
20:	28309	893	0	91	Florida, US

(Source: **https://kaggle.com)**

(b) Clinical Datasets: Clinical datasets provide comprehensive medical records for COVID-19 patients, including symptoms, results of diagnostic tests, radiological images, treatments used, and results. The clinical characteristics and illness development can be studied using these datasets.

(c) Imaging Datasets: COVID-19 patients' chest X-rays, Computed Tomography (CT) scans, or lung ultrasound pictures make up imaging datasets. For the automated detection and diagnosis of COVID-19, deep learning models are developed and evaluated using these datasets (See Figure 2.10).

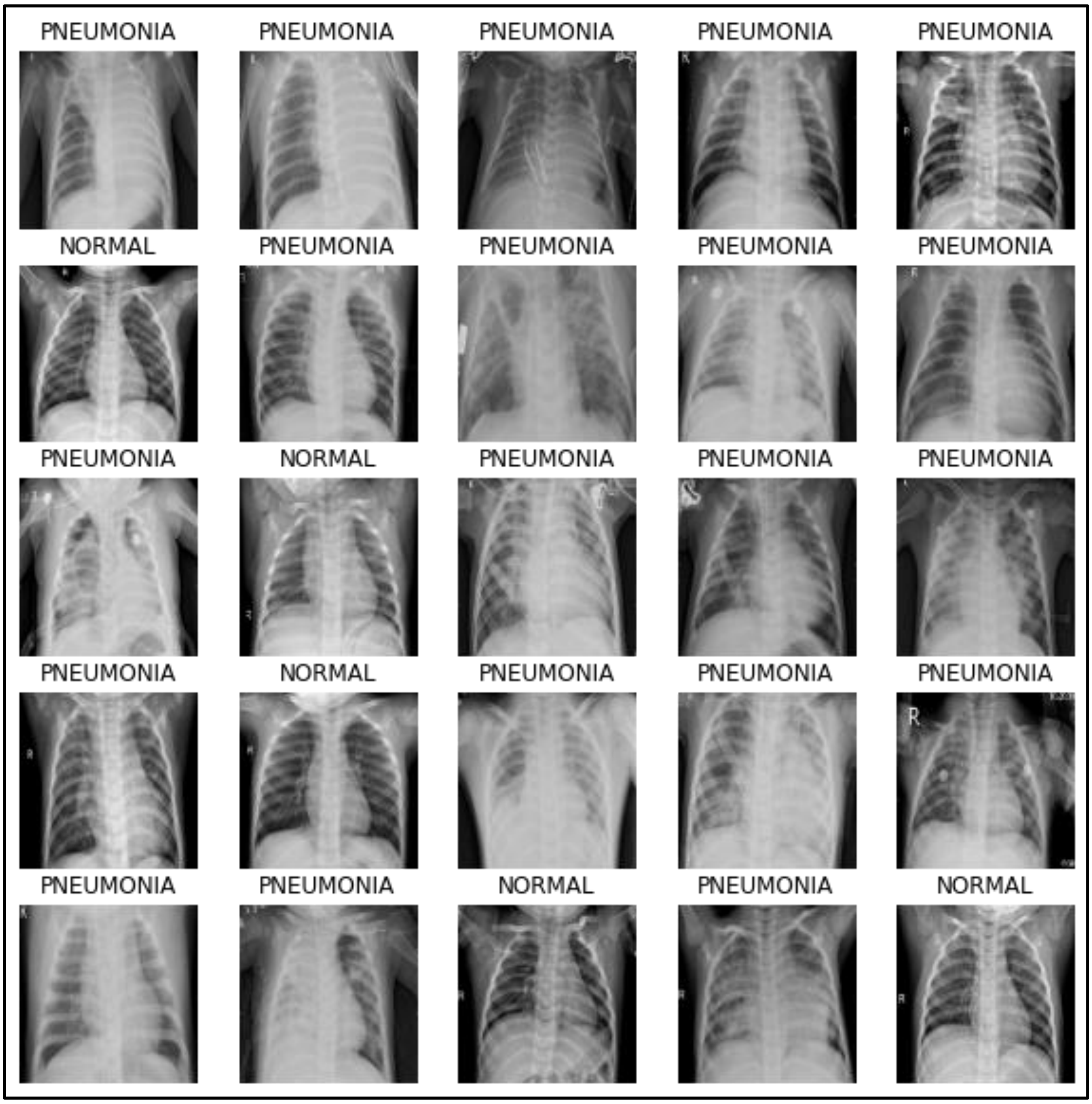

Figure 2.10: A sample Dataset of COVID-19 Chest X-Ray images (Https://Kaggle.com)

(d) Genomic Datasets: The genetic makeup of the COVID-19-causing SARS-CoV-2 virus is the main topic of genomic datasets. They consist of the virus's genome sequences as well as variants and mutations. Understanding the genetic diversity and evolution of the virus is aided by these datasets.

(e) Mobility Datasets: During the epidemic, movement patterns and behaviors were captured using mobility datasets. They contain information obtained from sources like Global Positioning System (GPS) tracking, mobile programs, or transportation systems. These datasets aid in the modeling of disease transmission dynamics and shed light on how mobility affects virus dissemination.

(f) **Social Media Datasets:** Social media datasets include information from sites like Twitter, Facebook, or Instagram that feature posts, tweets, or debates about COVID-19. Understanding public attitude, the spread of false information, and the effect of social media on public health metrics is made possible by these datasets.

(g) **Vaccination Datasets:** With the release of COVID-19 vaccinations, vaccination datasets monitor the administration and distribution of vaccines. They offer details on vaccination coverage, dosage rates, vaccine kinds, and demographics of those who have received vaccinations.

(h) **Economic Datasets:** Economic datasets concentrate on the socioeconomic effects of the epidemic. Data on GDP, consumer expenditure, stock market performance, and other economic metrics impacted by the pandemic are among those included.

2.2 Theoretical Framework

Theoretical framework has an impact on this study's methodology by helping to clarify this research subject field using established theories.

2.2.1 Diagnostic Theory

De Kleer and Williams first proposed the idea of diagnosis in 1993, emphasizing the discovery of model-artifact incompatibilities through the detection of assumptions violations. In order to prevent wasting time on useless therapies, reliable medical data interpretation might be difficult due to human error. The basis for proving or disproving symptoms and illnesses is provided by diagnostic theory, which also directs physicians in developing treatment regimens based on diagnostic testing. Through the provision of a theoretical framework for merging device models, heuristics, and deep learning models, Artificial Intelligence (AI) has improved diagnostic procedures.

Deep Neural Network (DNN) models provide up new views on comprehending cognitive processes like human decision-making due to its huge capacity and data-driven design. To investigate and identify predictable patterns in human behavior, they can be used in conjunction with theory-driven models. Generally, by utilizing deep learning techniques and offering a thorough framework for knowledge representation, the fusion of AI with diagnostic theory has increased the accuracy and efficacy of diagnoses.

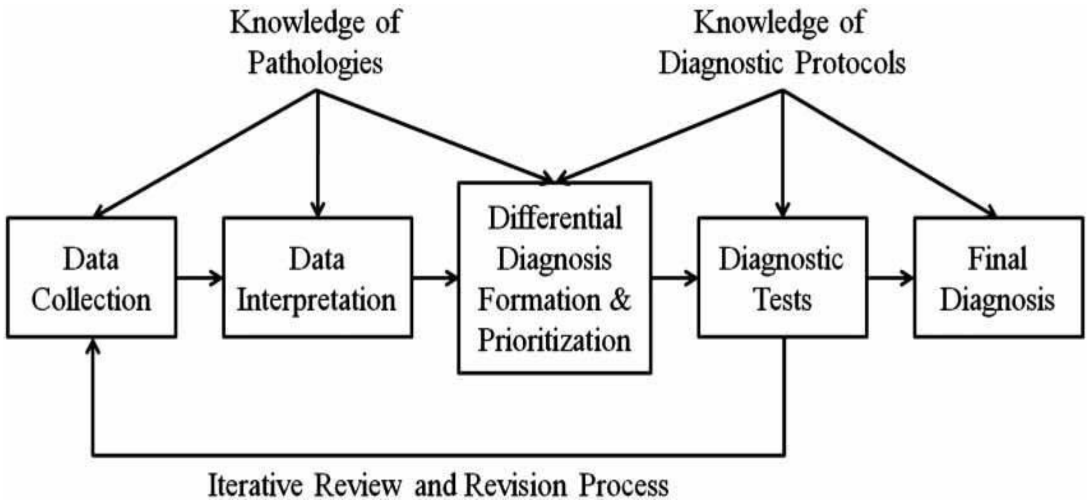

Figure 2.11: Flowchart of the Diagnostic Process (McFillen *et al.*, 2013)

2.2.2 Prediction Theory

The mathematician Jean-Baptiste Joseph Fourier developed the Fourier transform, which enables functions to be examined using their frequency components. In several fields, including Fourier techniques and time series analysis, predictions or forecasts are essential. Deep learning uses the Fourier transform as a mathematical tool to compute layer operations and approximation of functions. Convolutional layers in deep learning networks can be understood as frequency domain multiplication by using the Fourier transform, thus establishing a link between the two ideas. Understanding how the convolutional layer works and how it is akin to polynomial multiplication is made easier by this relationship.

2.2.3 Matrix Theory

Matrix theory, a branch of linear algebra that concentrates on the study of matrices, was first proposed in 1850 by Arthur Cayley. Matrix theory has evolved over time to encompass additional disciplines including graph theory, algebra, combinatorics, and statistics. Numerous fields, including machine learning, statistics, electrical circuits, cryptography, probability, and operations research, use matrices and linear algebra. Convolutional Neural Networks (CNNs) are a significant application where matrices play a key role. The pooling layer aids in reducing the size of feature maps, and the convolutional layer in a CNN uses matrices to conduct convolution operations on input pictures. This results in effective network processing and parameter learning. Figure 2.12 illustrates the CNN pooling action.

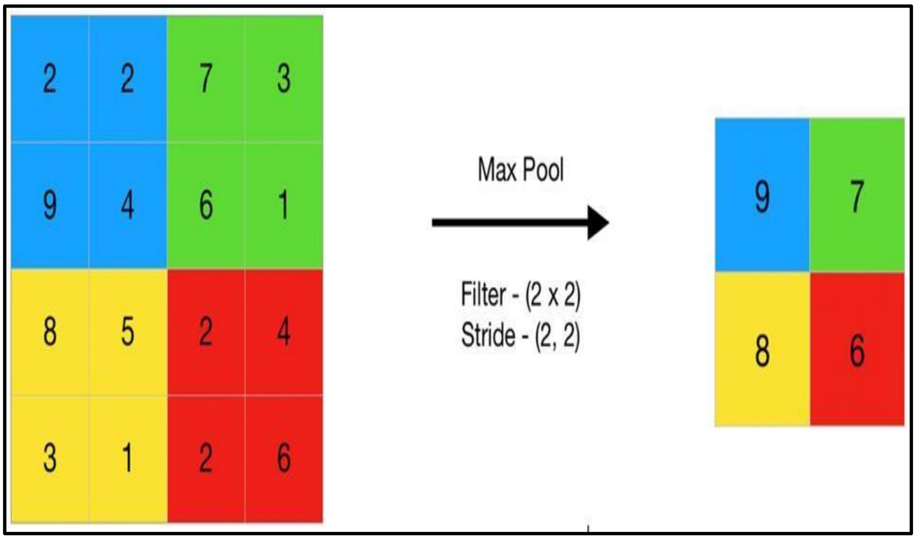

Figure 2.12: Actions of the Pooling Layer (Patel, 2019)

2.2.4 Theory of Reasoning

The term "thinking" was first used in 1991 by Ruth M. J. Byrne and Philip Johnson-Laird, who emphasized that reasoning is more often used for group decision-making than for individual thought. To improve our perspectives and decision-making skills, reasoning entails critically assessing a situation. Recurrent deep neural networks can be used to integrate reasoning theory and deep learning in conventional reasoning systems. By connecting various network components, these networks enable the system to reason by allowing the feedback of output into the input layer. In conclusion, reasoning aids in group decision-making, and the marriage of reasoning theory and deep learning can strengthen reasoning through recurrent deep neural networks.

Deep learning, in contrast to conventional symbolic logic, enables a more adaptable and data-driven knowledge representation structure to be generated by the learning process. Due to their extensive computing power and data-centric architecture, deep neural network models have the potential to reveal novel insights into cognitive processes like human decision-making. They can be used to better understand and predict human behavior when combined with explicit, theory-based models.

2.2.5 Graph Theory

The discovery of graph theory, a field of mathematics that examines the connections between objects represented as nodes and edges, was made possible by Leonard Euler's solution to the Seven Bridges of Königsberg puzzle in 1735. Road networks in mapping applications are one example of a system that is frequently modelled using graphs. Although deep learning models in particular have helped to overcome some of these restrictions, graph theory has historically had limited uses. Deep learning models are particularly suited for tasks involving graphs since they are excellent at processing unidirectional data within a limited amount of memory. Graphs provide a discrete and connected model for complicated processes, and combining graph theory and machine learning has shown promise in furthering both disciplines. This concept is depicted in Figure 2.13.

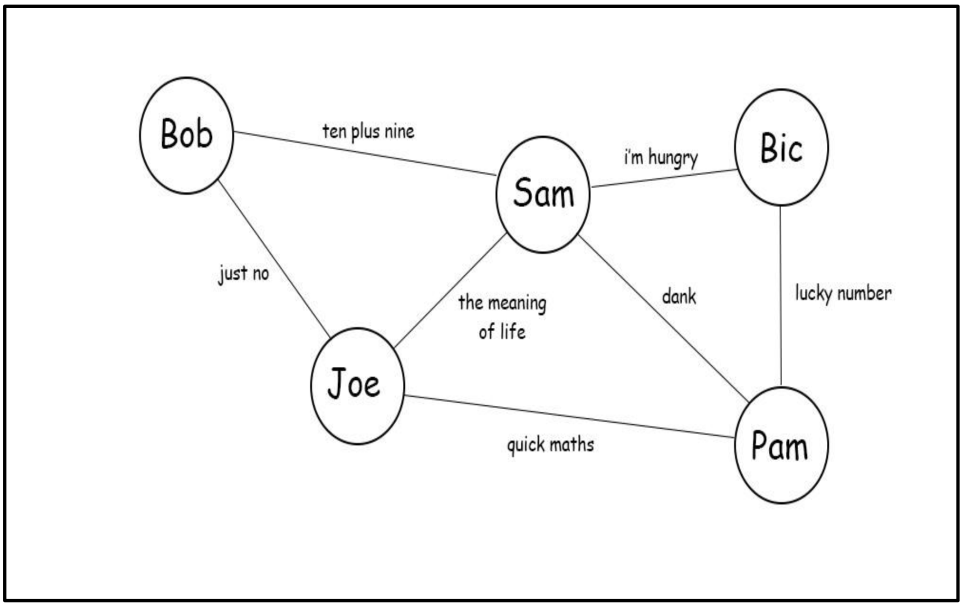

Figure 2.13: Nodes' Connection (Flawnson Tong, 2019))

2.3 Empirical Framework

This section discussed recent research papers that have explored the use of machine learning and deep learning methods for classifying and predicting COVID-19.

Ngabo *et al.* (2021) highlighted the pivotal role of Artificial Intelligence (AI) in transforming healthcare systems, particularly in managing pandemics within smart cities. Their study focuses on harnessing AI algorithms to predict the probability of survival among suspected COVID-19 patients, considering variables like immune system strength, exercise habits, and age distribution. They evaluated the performance of four different algorithms such as Naïve Bayes, Logistic Regression, Decision Tree, and K-Nearest Neighbor (KNN) by analyzing their True Positive (TP) and False Positive (FP) rates on COVID-19 patient data. The results shows that KNN and Decision Tree achieve the highest TP rates at 99.30%, while Naïve Bayes and Logistic Regression score slightly lower at 91.70% and 99.20%, respectively, for negative patients. Notably, Naïve Bayes demonstrates superior performance for positive COVID-19 cases, with a TP rate of 10.90%. Regarding FP rates for negative patients, Naïve Bayes records the lowest score at 89.10%, whereas Logistic Regression, KNN, and Decision Tree attain scores of 93.90%, 93.90%, and 94.50%, respectively.

Kaushik, Chadha, and Sharma (2023) discussed the pressing requirement for reliable COVID-19 detection methods during the worldwide pandemic. They suggest a hybrid deep transfer learning method designed specifically for identifying COVID-19 from chest X-ray images. In contrast to earlier investigations that had a scarcity of COVID-19 X-ray images, this study employs a well-balanced dataset containing 28,384 X-ray images, equally distributed between COVID-19 and normal categories. Through assessments conducted on this dataset, the hybrid technique exhibits better performance compared to conventional transfer learning and deep learning approaches.

Purohit *et al.* (2020) addressed the challenges presented by COVID-19, emphasizing the necessity for accurate diagnostic methods to complement traditional RT-PCR testing. They suggested utilizing chest X-ray and CT scan images for COVID-19 screening to enhance diagnostic accuracy while minimizing false positives and false negatives. The researchers introduced a Convolutional Neural Network (CNN) based multi-image augmentation technique tailored for detecting COVID-19 in these images. This method leverages discontinuity information from filtered images to augment the dataset, thereby enhancing the effectiveness of CNN model training. They reported that the proposed model achieved high classification accuracy, with 95.38% for CT scan images and 98.97% for X-ray images. Furthermore, CT scan images with multi-image augmentation demonstrated a sensitivity of 94.78% and specificity of 95.98%, while X-ray images with multi-image augmentation showed a sensitivity of 99.07% and specificity of 98.88%. The evaluation was conducted using publicly available databases, and the experimental results were compared with ResNet-50 and VGG-16 models.

Bukhari *et al.* (2020) emphasized the importance of chest X-rays in diagnosing COVID-19-induced pneumonia, highlighting the need for computer programs to aid healthcare professionals in this regard. They conducted a study using 278 chest X-ray images, sourced from public repositories, and applied the ResNet-50 convolutional neural network architecture. The images were divided into three groups: normal, pneumonia caused by other pathogens, and COVID-19. Training and testing were performed on 80% and 20% of the dataset, respectively. Results showed that the computer vision-based program achieved a diagnostic accuracy of 98.18% and an F1-score of 98.19, demonstrating excellent performance in distinguishing COVID-19-induced pulmonary changes from other types of pneumonia. The study concludes that the convolutional neural network model could serve as a valuable tool for healthcare professionals in diagnosing COVID-19-related lung abnormalities on chest X-ray images.

Fu *et al.* (2020) developed an Artificial Intelligence (AI)-based diagnostic tool for classifying CT images to identify COVID-19 and other lung infectious diseases due to the time-consuming and subjective nature of human interpretation. Using a convolutional neural network, CT scan images of the lung were retrospectively collected and prospectively analyzed. The dataset included five conditions: COVID-19 pneumonia, non-COVID-19 viral pneumonia, bacterial pneumonia, pulmonary tuberculosis, and normal lung. Training and validation sets were obtained from Wuhan Jin Yin-Tan Hospital, while the test set was sourced from Zhongshan Hospital Xiamen University and the fifth Hospital of Wuhan. The AI framework achieved high accuracies for recognizing normal lung and various pneumonia types, with accuracies ranging from 98.3% to 99.4%. Specifically for COVID-19, the framework demonstrated an accuracy of 98.8%, sensitivity of 98.2%, specificity of 98.9%, Positive Predictive Value (PPV) of 94.5%, and Negative Predictive Value (NPV) of 99.7% on the test dataset, showcasing its excellent performance and balanced sensitivity and specificity.

Wang, Lin, and Wong (2020) highlighted the urgent need for effective screening methods amidst the COVID-19 pandemic, particularly through radiology examination using chest radiography. They introduce COVID-Net, a specialized deep convolutional neural network designed for detecting COVID-19 cases from Chest X-Ray (CXR) images, which is open source and accessible to the public. Additionally, they present COVIDx, an open access benchmark dataset comprising 13,975 CXR images from 13,870 patient cases, featuring the largest number of publicly available COVID-

19 positive cases to date. The authors also explore how COVID-Net makes predictions using an explainability method, aiming to provide deeper insights into critical factors associated with COVID cases and validate the network's decision-making process transparently. While COVID-Net is not intended as a production-ready solution, it is hoped that its open access nature, coupled with the description of constructing the COVIDx dataset, will facilitate further research and development efforts to accelerate the creation of accurate deep learning solutions for COVID-19 detection and treatment.

Qiao *et al.* (2020) conducted a study to explore the feasibility of distinguishing chest X-ray images of COVID-19 patients from those with other forms of pneumonia and healthy individuals using deep neural networks. They compiled a dataset of 5,508 chest X-ray images from 2,874 patients across four categories: normal, bacterial pneumonia, non-COVID-19 viral pneumonia, and COVID-19. Their proposed approach, termed Focal Loss Based Neural Ensemble Network (FLANNEL), employs a flexible module that combines multiple Convolutional Neural Network (CNN) models and integrates focal loss to address class imbalance. FLANNEL consistently outperformed baseline models across all metrics for COVID-19 identification, demonstrating a higher macro-F1 score with a relative increase of 6%. The Precision, Recall, and F1 score achieved by FLANNEL were reported as 0.7833±0.07, 0.8609±0.03, and 0.8168±0.03, respectively. The authors discussed the effectiveness of ensemble learning in enhancing robustness and accuracy, introducing a Neural Weighing Module to assign importance weights to each base model and combining them through weighted ensemble for final classification. Moreover, they adapted Focal loss to address class imbalance challenges in the multiple classification task. In conclusion, FLANNEL effectively integrates state-of-the-art CNN classification models and mitigates class imbalance using Focal loss, resulting in improved performance for COVID-19 detection from X-rays.

Loey, Smarandache, & Khalifa (2020) addressed the urgent need for efficient detection of COVID-19 amidst the global pandemic, emphasizing the potential of Artificial Intelligence (AI) in alleviating strain on healthcare systems. They propose a Generative Adversarial Network (GAN) with deep transfer learning for detecting coronavirus in chest X-ray images, motivated by the scarcity of COVID-19 datasets, particularly in chest X-rays. Their approach involves gathering existing COVID-19 images and using GAN to generate additional images to enhance detection accuracy. The dataset comprises 307 images categorized into four classes: COVID-19, normal, bacterial pneumonia, and viral pneumonia. The study evaluates three deep transfer models—AlexNet, GoogLeNet, and ResNet18 chosen for their architectural simplicity, aiming to reduce model complexity, memory consumption, and execution time. Three case scenarios are explored, varying the number of classes considered. Results indicate GoogLeNet achieves 80.6% testing accuracy in the first scenario, AlexNet achieves 85.2% in the second scenario, and GoogLeNet achieves 100% testing accuracy and 99.9% validation accuracy in the third scenario, where only COVID-19 and normal classes are considered. These findings validate the effectiveness of the proposed approach and highlight its potential for COVID-19 detection in chest X-ray images.

Khan *et al.* (2020) addressed the pressing need for accurate COVID-19 patient screening due to the virus's rapid spread and high false detection rates in conventional methods. They propose a 3D deep learning-based approach utilizing volumetric CT image data for automatic coronavirus screening. Through extensive experiments on CC-19 and COVID-CT datasets, employing various state-of-the-art 3D deep learning methods, including 3D ResNets, C3D, 3D DenseNets, I3D, and LRCN, the study demonstrates the competitive effectiveness of their proposed system.

Anwar & Zakir (2020) also addressed the urgent need for effective COVID-19 diagnosis due to its widespread impact on global health. With PCR kits facing shortages, the study explores the use of radiographic techniques like chest CT scans for diagnosis, leveraging deep learning technology. They employ the EfficientNet architecture to achieve timely and accurate coronavirus detection, reporting an accuracy of 0.897, F1 score of 0.896, and AUC of 0.895. The study investigates three learning rate strategies, including reducing the learning rate on plateau, cyclic learning rate, and constant learning rate. Notably, the reduce on plateau strategy yields the highest F1-score of 0.9, while cyclic learning rate and constant learning rate strategies result in F1-scores of 0.86 and 0.82, respectively. The implementation code is available on GitHub for further exploration and utilization.

Chandra *et al.* (2020) tackled the urgent challenge presented by the novel coronavirus disease (nCOVID-19), which stems from the Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-COV-2) and carries significant global health risks. Their study emphasizes the unique radiographic patterns observed in infected individuals, coupled with typical symptoms such as fever, dry cough, and dyspnea. Despite the valuable role of Chest X-Ray (CXR) in diagnosis, the limited availability of skilled radiologists and the subtle nature of disease-related radiographic features hinder manual interpretation. To address this, the researchers propose an automatic COVID screening (ACoS) system that utilizes radiomic texture descriptors extracted from CXR images. This system employs a two-phase classification strategy, utilizing a majority vote-based ensemble of five supervised classification algorithms to differentiate between normal, suspected, and nCOVID-19 cases. Through comprehensive testing and validation on datasets comprising thousands of CXR images, the ACoS system demonstrates promising performance, with high accuracy and area under the curve (AUC) values. The study also highlights the statistical significance of the ACoS system's performance and compares it with existing methods.

Ismael & Şengür (2020) discussed the challenges posed by COVID-19, emphasizing its impact on the respiratory system and the global escalation of cases and fatalities. They explore the utility of chest X-ray images in monitoring COVID-19 progression. The study employs deep-learning techniques, including deep feature extraction, fine-tuning of pretrained Convolutional Neural

Networks (CNN), and end-to-end training of a CNN model, to classify COVID-19 and normal chest X-ray images. Pretrained CNN models such as ResNet18, ResNet50, ResNet101, VGG16, and VGG19 are utilized for feature extraction, followed by classification using Support Vector Machines (SVM) with various kernel functions. The study also introduces a novel CNN model trained end-to-end. Using a dataset of 180 COVID-19 and 200 normal chest X-ray images, the experiments measure classification accuracy. Results indicate promising potential for deep learning in COVID-19 detection, with the ResNet50 model and SVM classifier achieving the highest accuracy of 94.7%. The study compares deep learning approaches with local texture descriptors and SVM classifications, highlighting the efficiency of deep learning in COVID-19 detection from chest X-ray images.

Karhan & Akal (2020) highlighted the swift global impact of the Covid-19 virus, emphasizing the importance of early detection to curb its spread. They underscore the significance of chest radiological images alongside RT-PCR tests in the diagnostic process. Utilizing the ResNet50 model, a convolutional neural network architecture, they classify Covid-19 cases using chest X-ray images. The study demonstrates the potential of artificial intelligence for rapid chest X-ray analysis and identification of infected individuals, offering promising results for computer-aided pathology. This approach proves valuable in scenarios where traditional testing methods are limited.

Jain *et al.* (2020) addressed the significant impact of the rapidly spreading Covid-19 virus on both humans and animals, affecting daily life, health, and economies worldwide. Despite global efforts, no vaccine has been developed yet. Clinical studies indicate lung infections as a common manifestation of Covid-19, emphasizing the importance of chest x-ray and CT scans for diagnosis, with chest x-ray being a cost-effective option. Leveraging deep learning, specifically CNN models, the study focuses on analyzing PA view chest x-ray scans of Covid-19 patients and healthy individuals. Three CNN models (Inception V3, Xception, and ResNeXt) are compared for accuracy using 6432 chest x-ray samples, with Xception demonstrating the highest accuracy at 97.97%. The study aims to classify Covid-19 patients based on imaging data and does not make medical claims.

Foysal & Aowlad (2021) proposed an ensemble deep convolutional neural network (CNN) method for diagnosing COVID-19 cases from chest computed tomography (CT) images, aiming to accurately identify and isolate infected patients. They design, train, and validate three deep CNN models with CT images, combining their prediction values using ensemble hard voting rules to improve accuracy. Experimental results show an overall accuracy of 96% and sensitivity of 97%, highlighting the model's potential for rapid and precise prediction of COVID-19 cases from CT images, offering a quicker alternative to traditional RTPCR techniques.

Majid *et al.* (2021) introduced an automated system for detecting COVID-19 from CT images, aiming to overcome the limitations of RT-PCR testing. They utilize deep learning models like ResNet50 and DarkNet53 for feature extraction, fine-tuning them via transfer learning. Their approach involves curating a dataset of CT images categorized into Pneumonia, COVID-19, and Healthy classes. Features are optimized using a hybrid contrast stretching method and a combination of optimization algorithms. The final fused features are classified using extreme machine learning. Experimental results showcase an impressive overall accuracy of 95.6%, underscoring the effectiveness of their framework.

Khalil *et al*. (2022) addressed the rapid global spread of the COVID-19 pandemic, originating in Wuhan, China in December 2019, by proposing a deep learning approach utilizing the EfficientnetB4 model for COVID-19 classification. Leveraging transfer learning, which utilizes pre-trained models from the ImageNet database, the study aims to enhance generalization for COVID-19 detection. Through in-depth training, the model extracts visual features specific to COVID-19, offering a preliminary medical assessment before formal testing. Evaluated on a publicly available X-ray dataset, the proposed framework achieves an impressive 97% accuracy. The experimental results underscore its effectiveness as a precise and rapid decision support tool for COVID-19 identification, suggesting its potential utility for healthcare organizations.

Redie *et al*. (2022) delved into the pressing issue of COVID-19, caused by SARS-CoV-2, emphasizing the need for accurate diagnosis to curb its global impact. While RT-PCR remains the primary detection method, there is growing interest in leveraging deep learning for medical diagnostics to complement conventional approaches. Their study aims to develop a model using chest X-ray images for automated COVID-19 identification. Through a modified DarkCovidNet model based on Convolutional Neural Networks (CNN), the researchers conducted experiments for binary and multi-class classification scenarios. Trained on a dataset comprising over 10,000 X-ray images, the model demonstrated remarkable accuracies of 99.53% and 94.18% for binary and multi-class classification, respectively. These findings underscore the efficacy of utilizing X-ray images for COVID-19 detection, offering a potential solution for situations where RT-PCR tests are unavailable.

Idupulapaati & Sreedevi (2022) introduced an innovative CNN architecture inspired by AlexNet, designed to process three-channel image inputs swiftly, addressing the rapid escalation of COVID-19 globally. Detecting positive cases early is crucial for curtailing the disease's spread and ensuring timely isolation. While RT-PCR remains accurate, its time-consuming nature underscores the need for more efficient detection methods. The proposed deep learning approach, integrating radiography, shows promise for swift COVID-19 diagnosis, offering a timely and effective solution amidst the ongoing healthcare challenges.

Udayaraju *et al*. (2023) addressed the pressing need for efficient COVID-19 detection methods due to the virus's rapid spread and severe health implications if not promptly identified. While existing testing methods have limitations like time consumption and radiation exposure, the authors introduce GW-CNNDC, a model fine-tuned from pre-trained deep-learning algorithms for precise COVID-19 detection using CXR images. Leveraging an Enhanced CNN model with RESNET-50 Architecture and a Gradient Weighted model, their approach achieves accurate classification and precise localization of affected areas. The framework demonstrates high accuracy, precision, recall, F1-score, and minimal loss values, enabling rapid processing of extensive datasets.

Mohammadi *et al*. (2023) conducted a cross-sectional study aiming to investigate the use of blood tests and a multilayer perceptron neural network in diagnosing and understanding factors influencing Covid-19. The study involved 200 patients confirmed with Covid-19 through computerized tomography-scan analysis at Sina Hospital, Tehran, Iran, between March 2, 2020, and April 5, 2020. Blood samples were collected to analyze C-reactive protein (CRP), magnesium (Mg), lymphocyte percentage, and vitamin D levels in both healthy and infected individuals. The data were divided into 70% for training and 30% for testing the multilayer perceptron network. Results revealed significant associations between elevated CRP levels, decreased lymphocyte levels, and increased Mg levels in Covid-19 patients ($p < 0.01$). Additionally, gender-specific differences were observed, with higher CRP and Mg levels in women and increased lymphocytes and vitamin D in men among Covid-19 patients ($p < 0.01$). The study highlights the potential of multilayer perceptron neural networks to expedite diagnosis and treatment processes.

Huyut (2022) investigated how Routine Blood Values (RBV) and demographic data impact COVID-19 prognosis, aiming to distinguish between severe and mild cases upon admission using supervised Machine Learning (ML). The study, spanning March to September 2021, examines data from 192 severely and 4010 mildly infected patients. RBV and age-gender characteristics are analyzed, identifying 28 RBV parameters and age as the feature dataset. ML models are compared, with Local Weighted Learning (LWL) achieving the highest overall accuracy (97.86%) and LWL also exhibiting the highest Area Under Curve (AUC) value (0.95%). These findings emphasize the importance of early COVID-19 severity detection upon admission, offering valuable insights for healthcare practitioners.

Abayomi-Alli *et al*. (2022) addressed the challenges in real-time COVID-19 detection using RT-PCR test data and propose an ensemble learning approach utilizing routine laboratory blood test results. Their study introduces a machine learning-based COVID-19 detection system, employing custom CNN models as a first-stage classifier and 15 supervised ML algorithms as a second-stage classifier. Results demonstrate that the ensemble model, particularly based on DNN and ExtraTrees, achieves high accuracy (99.28%) and AUC (99.4%), with AdaBoost also performing well with a mean accuracy of 99.28% and AUC of 98.8% on the San Raffaele Hospital dataset. Comparative analysis with existing methods using the same dataset highlights the superior performance of the proposed approach in COVID-19 detection.

Rehman *et al*. (2021) emphasized the continued importance of rapid COVID-19 diagnosis amidst the global pandemic. They propose a machine learning-based approach to enhance the effectiveness of diagnosing COVID-19-infected individuals, considering various symptoms like flu symptoms, throat pain, immunity status, and more. By modeling these symptoms as ML features, their method predicts the likelihood of COVID-19 contamination. Evaluation using metrics such as accuracy, precision, recall, and F1-score reveals the method's efficacy, achieving over 97% accuracy in predicting COVID-19 presence.

Alves *et al*. (2021) addressed the pressing need for efficient COVID-19 screening tools by employing Machine Learning (ML) techniques applied to routine blood tests. Their study evaluates various ML classifiers using a dataset from Hospital Albert Einstein, São Paulo, Brazil, consisting of 608 patients, including 84 confirmed COVID-19 cases by RT-PCR. The Random Forest (RF) classifier demonstrated superior performance, achieving an accuracy of 0.88, F1-score of 0.76, sensitivity of 0.66, specificity of 0.91, and AUROC of 0.86. Additionally, the authors introduce a local Decision Tree Explainer (DTX) and Criteria Graph to offer interpretable insights into model decisions, aiding clinicians in understanding blood parameter interconnections both globally and on a case-by-case basis. These results are consistent with existing literature, suggesting potential integration of the proposed methodology into electronic health record systems.

Gozes *et al.* (2020) addressed the global impact of the COVID-19 pandemic by developing a deep learning algorithm to assist radiologists in detecting, localizing, and assessing the severity of COVID-19 manifestations on chest CT scans. The algorithm comprises various image processing stages, including lung segmentation, 2D slice classification, and fine-grained localization. Additionally, unsupervised clustering is employed to gain further insights into the disease's manifestations. The study presents results from a dataset of 110 confirmed COVID-19 patients from Zhejiang province, China, demonstrating the algorithm's potential to support healthcare professionals during this challenging period.

Mei *et al.* (2020) highlighted the pressing need for alternative methods to the time-consuming and sometimes inconclusive Reverse Transcriptase Polymerase Chain Reaction (RT–PCR) test for diagnosing COVID-19. They explore the integration of Artificial Intelligence (AI) algorithms with chest Computed Tomography (CT) findings, clinical symptoms, exposure history, and laboratory testing to swiftly identify COVID-19-positive patients. In their study involving 905 patients, the AI system achieved an impressive area under the curve of 0.92 in a test set of 279 patients, demonstrating equal sensitivity compared to a senior thoracic radiologist. Notably, the AI system improved detection, particularly in cases where patients presented with normal CT scans but were positive for COVID-19 via RT–PCR. This approach holds promise for enhancing the rapid diagnosis of COVID-19 patients when CT scans and relevant clinical data are available.

Ferrari *et al.* (2020) addressed the urgent need for alternative and more accessible diagnostic methods for COVID-19, given the limitations of the current gold standard real-time Reverse Transcriptase Polymerase Chain Reaction (rRT-PCR) test. They analyze the plasma levels of various hematological parameters in 207 patients presenting COVID-19 symptoms, comparing results with RT-PCR tests.

Significant differences were found in parameters like white blood cells (WBC), C-Reactive Protein (CRP), Aspartate Amino Transferase (AST), Alanine aminoTransferase (ALT), and Lactate DeHydrogenase (LDH). Empirical Thresholds for AST and LDH were identified, enabling the identification of 70% of COVID-19-positive or -negative patients based on routine blood test results. This suggests that blood test analysis could serve as an alternative to RT-PCR, particularly in regions facing shortages of testing resources or specialized laboratories.

Cheng *et al.* (2020) underscored the critical role of diagnostic testing in controlling the global COVID-19 pandemic caused by Severe Acute Respiratory Syndrome-related Coronavirus 2 (SARS-CoV-2). While some countries have effectively used widespread testing to contain the virus, limited testing capacity has hindered efforts in others, such as the United States. Real-time Reverse Transcriptase Polymerase Chain reaction (rRT-PCR) assays are the standard for diagnosis, but point-of-care technologies and serologic immunoassays are rapidly emerging. Despite progress, challenges persist in screening asymptomatic individuals and accurately determining viral shedding, particularly in convalescent patients. Affluent nations face obstacles in test delivery and specimen collection, and these challenges are likely greater in low-resource settings. Recognizing the urgent need, there's a global push to expand testing capacity, necessitating a review of current tests, identification of diagnostic gaps, and exploration of potential solutions.

Rehman *et al.* (2021) proposed a Computer Aided Diagnosis (CAD) system to provide real-time support for physicians by remotely receiving and promptly analyzing X-ray images. Their model highlights the urgent need worldwide to combat COVID-19 and emphasizes the crucial role of rapid and accurate diagnosis in mitigating its impact on healthcare systems. They advocate for the adoption of CAD systems, leveraging Convolutional Neural Network (CNN) algorithms like ResNet50, to examine chest X-ray images and detect COVID-19 with an impressive 98% accuracy. Additionally, the system incorporates advanced features such as load balancers and resilience mechanisms to ensure fault tolerance and minimize delays, thereby enhancing its effectiveness in identifying infected cases during the ongoing pandemic.

Wang *et al.* (2020) utilized the AlexNet network in deep learning to discern whether lung CT images are afflicted with COVID-19. Initially, in the data preprocessing phase, the original CT image data undergo scaling and normalization to minimize noise interference. Employing batch operations on the training and test sets can enhance training speed. Subsequently, an eight-layer AlexNet network model is constructed, with carefully selected hyperparameters for each layer and the definition of the loss function and optimizer. The model is then trained using the processed data to adjust the weight parameters of each layer. Evaluation metrics such as accuracy, precision, and recall are employed to assess the classification model's effectiveness, with different training durations compared to identify the optimal model. Additionally, a user-friendly auxiliary detection interface is developed using Pyqt5 to facilitate test result display and model selection.

Mporas & Naronglerdrit (2020) explored the role of Artificial Intelligence and Data Science in combating COVID-19 by evaluating various pretrained deep CNN models for detecting the virus from chest X-ray images. They tested different setups using two separate datasets and found DenseNet, ResNet, and Xception models to be the most effective. Their study suggests the feasibility of identifying COVID-19 positive cases through chest X-ray images.

Horry *et al.* (2020) investigated the early detection of COVID-19 using deep learning models trained on X-Ray, Ultrasound, and CT scan images. They aimed to assist overwhelmed medical professionals by providing intelligent image classification models. Through a comparative study, they selected the VGG19 Convolutional Neural Network (CNN) model and optimized it for the different image modalities. However, they faced challenges due to limited and low-quality COVID-19 datasets, affecting the trainability of complex models. They proposed an image pre-processing stage to enhance dataset quality. Their findings revealed that Ultrasound images outperformed X-Ray and CT scans in detection accuracy. Despite data limitations, the tuned VGG19 model achieved notable precision levels: 86% for X-Ray, 100% for Ultrasound, and 84% for CT scans in distinguishing COVID-19 from pneumonia or normal cases.

Salih *et al.* (2020) addressed the urgent need for precise COVID-19 diagnosis due to its global impact, leveraging medical imaging techniques and deep learning algorithms. They employ Convolutional Neural Network (CNN) based on a modified AlexNet architecture to detect and diagnose COVID-19 using Chest X-ray images. By gathering data from two COVID-19 X-ray image datasets and a dataset containing normal and pneumonia X-ray images, they develop models that yield comparable or superior results to the original AlexNet, even with fewer neurons in the second fully connected layer. Their approach demonstrates the effectiveness of deep learning techniques in accurately diagnosing COVID-19 from medical images.

Punia *et al.* (2020) addressed the rapid spread of COVID-19 worldwide, with 128,000 fatalities and nearly 2 million reported cases as of April 14, 2020. Due to the scarcity of medical testing kits, utilizing chest X-rays, which effectively reveal respiratory system abnormalities, emerges as a viable alternative to thermal screening. Their study aimed at devising a methodology utilizing radiology, specifically X-rays, for detecting the novel coronavirus. Additionally, they provided a dataset sourced from various medical research hospitals treating COVID-19 patients, facilitating further research and development in the field.

Singh & Singh (2021) addressed the global pandemic of COVID-19 caused by the SARS-CoV-2 virus, emphasizing the urgent need for swift diagnosis to control its spread effectively. While laboratory tests face limitations due to kit availability and time constraints, radiological methods like CT scans offer alternative diagnostic avenues. The authors proposed an automated approach utilizing chest X-ray images for COVID-19 diagnosis, employing an enhanced depth-wise convolution neural network with wavelet decomposition for multi-resolution analysis integration. The network categorizes images into normal, viral pneumonia, and COVID-19 classes, utilizing predicted outputs combined with Grad-CAM visualization for diagnosis. Comparative analysis with existing methods demonstrates superior performance metrics including accuracy, sensitivity, and F1-measure, underscoring the proposed method's effectiveness in disease diagnosis.

Calderon-Ramirez *et al.* (2021) introduced a COVID-19 detection system based on chest X-ray images, emphasizing the importance of uncertainty estimation for safe medical diagnosis tools. Their work focuses on enhancing uncertainty estimates using unlabelled data through the MixMatch semi-supervised framework. Various uncertainty estimation methods are tested, including Softmax scores, Monte-Carlo dropout, and deterministic uncertainty quantification. They proposed using the Jensen-Shannon distance to compare the reliability of uncertainty estimates, showing significant improvement with unlabelled data, particularly with the Monte Carlo dropout method.

Oyelade, Ezugwu, & Chiroma (2021) proposed a computational intelligence solution to improve COVID-19 detection during the global pandemic. They emphasized the effectiveness of deep learning models and introduced the CovFrameNet framework, which combines deep learning and image pre-processing techniques for identifying and characterizing novel coronavirus infections. Their study introduced a novel CNN architecture with enhanced image pre-processing capabilities and evaluated its performance using the NIH Chest X-Ray dataset and COVID-19 Radiography database. Results indicate that the proposed model achieves high accuracy, recall/precision, F-measure, and specificity, suggesting its potential utility in pre-screening suspected COVID-19 cases and confirming RT-PCR-based detections.

Apostolopoulos & Bessiana (2020) proposed the utilization of a dataset comprising X-ray images from patients diagnosed with common pneumonia, COVID-19, and normal conditions for the automatic detection of the coronavirus. Their study aimed to assess the efficacy of state-of-the-art

Convolutional Neural Network (CNN) architectures, developed in recent years for medical image classification, through the application of transfer learning. By employing transfer learning, which enables leveraging pre-trained models on large datasets, the detection of various abnormalities in small medical image datasets becomes feasible, often yielding remarkable results. The dataset used in the experiment consisted of 1427 X-ray images, including 224 images with confirmed COVID-19, 700 images with confirmed common pneumonia, and 504 images depicting normal conditions, sourced from public medical repositories. Through transfer learning, the proposed approach achieved an impressive overall accuracy of 97.82% in detecting COVID-19.

Arias-Londoño *et al.* (2020) proposed an evaluation of various methods based on deep neural networks to develop an automatic COVID-19 diagnosis tool using chest X-ray images. Their study aimed to differentiate between controls, pneumonia, or COVID-19 groups, providing new evidence for diagnosis. The Convolutional Neural Network was trained with a dataset comprising over 79,500 X-ray images, including more than 8,500 COVID-19 examples. Three experiments with different preprocessing schemes were conducted to assess the impact on results and improve explainability. The methodology yielded 91.5% classification accuracy, with an average recall of 87.4% for the most explainable experiment, albeit requiring automatic segmentation of the lung region beforehand.

Mukri (2023) developed a Machine Learning model for COVID-19 Prediction, recognizing the urgency for innovative virus-control measures amidst the pandemic. The study focuses on leveraging Machine Learning (ML) to predict COVID-19 cases using X-Ray scanning images. Two datasets, comprising COVID-positive X-ray scans and a collection of conventional X-ray images, were utilized for model training and validation. Grayscale conversion and resizing to 100x100 pixels were performed to standardize the image data. The CNN algorithm employed achieved impressive training accuracy of 100% and validation accuracy of 95.24%, with a test accuracy of 96.66% in identifying COVID-19 patients. Additionally, the model predicted the distribution of COVID-19 cases, using

Monte Carlo simulations and Gaussian Mixture Models (GMM), showcasing promising results for pandemic spread forecasting. Ongoing research aims to further enhance the model's performance and predictive capabilities.

Ghafouri-Fard *et al.* (2021) stressed on the global impact of COVID-19 and the critical necessity to accurately predict its spread. They advocate for the use of artificial intelligence (AI) methods due to the disorder's nonlinear and complex nature, which presents challenges for traditional epidemic models. The study examines different machine and deep learning approaches employed for forecasting COVID-19 cases, including adaptive neuro-fuzzy inference systems, long short-term memory networks, recurrent neural networks, and multilayer perceptrons. Performance evaluation based on metrics like RMSE, MAE, R2, and MAPE indicates varied accuracies among the models, with R2 values spanning from 0.64 to 1. These AI models empower precise policy-making by pinpointing factors that influence COVID-19 transmission and aiding in the formulation of effective interventions and policies.

Gonsalves & Nitnaware (2021) emphasized on the urgency of controlling the global COVID-19 pandemic and improving the overall state of health, finance, and livelihood worldwide. Their research aims to contribute to this goal by providing Local Bodies/Municipalities with advanced technology to predict COVID-19 trends in their respective areas. Five machine learning algorithms, including ARIMA, LSTM, FB Prophet, Linear Regression (with FB Prophet), and Random Forest (with FB Prophet), are employed for this purpose. By visualizing trends on graphs and testing predictions against actual data, the most suitable algorithm is identified to forecast future COVID-19 cases. The study also discusses the computation of municipality-level forecasts based on reverse chronological sequence and extrapolation from country-level data. ARIMA and Random Forest models are identified as the most effective for predicting active cases in India over the next 20 days, potentially assisting authorities in implementing preventive measures to curb the spread of COVID-19.

Subhan *et al.* (2021) proposed an AI-powered model for predicting the health status of COVID-19 patients, employing the Artificial Intelligence Boosted Random Forest Algorithm particularly for addressing large-scale pandemics such as the 2019 Coronavirus outbreak. With a global death toll of 175,694 and 254,4792 active cases, and 658,132 positive cases, 603,512 recoveries, and 16,208 deaths recorded in Pakistan by March 2021. they utilized patient health data like geographical location, gender, and marital status, the model predicts the severity and potential outcomes of COVID-19 cases, whether recovery or death. With a 90% accuracy rate and a 0.76 F1 Score, the model reveals a positive correlation between patient gender and mortality, with most patients falling between the ages of twenty and seventy years.

Pal *et al*. (2019) proposed a neural network-based approach for country-wise risk prediction of COVID-19, employing various experiments to evaluate the effectiveness of their method. They utilized a dataset comprising COVID-19 statistics and weather data from January 22, 2020, to March 10, 2020. Feature selection revealed that active cases were universally chosen, with ozone emerging as a significant factor. Network optimization involved Bayesian optimization to generate country-specific networks, with most networks having few layers and hidden units using ReLu activation. Training was conducted using a maximum of 5000 epochs, and prediction accuracy was assessed based on death rate, case rate, and recovery rate. Comparisons with baseline algorithms and advanced neural networks highlighted the efficacy of their approach, particularly in handling the limitations of a small dataset. The results demonstrate the potential of their method for accurate COVID-19 risk prediction.

Dokeroglu (2023) introduced the Harris’ Hawk Optimization (HHO) as a novel metaheuristic inspired by hawks' collective hunting behaviors, aiming to address complex classification problems by leveraging the flight patterns of hawks to generate optimal or near-optimal solutions. Their study presents a novel parallel multi-objective HHO algorithm specifically tailored for predicting COVID-

19 patient mortality risk based on symptoms. The optimization objectives are two-fold: reducing the number of features while simultaneously enhancing prediction accuracy. Extensive experiments conducted on a real-world COVID-19 dataset from Kaggle, along with an augmented version of the dataset, demonstrate significant improvements compared to existing meta-heuristic wrapper algorithms. The results indicate superior classification outcomes with feature selection compared to using the entire feature set, achieving a prediction accuracy of 98.15% with a 45% reduction in the number of features. This study contributes new and improved solutions for COVID-19 risk prediction tasks.

Car *et al.* (2020) addressed on the global significance of Coronavirus (COVID-19) and its impact on public health. They emphasize the importance of modeling such diseases for predicting their effects accurately. While traditional statistical models can be adequate, they may overlook the complexities inherent in the data. The authors utilize a publicly available dataset spanning 406 locations over 51 days, transforming it from a time-series to a regression dataset. They employ a Multi Layer Perceptron (MLP), Artificial Neural Network (ANN) to train a global model predicting the maximum number of patients across all locations per time unit. Hyperparameters of the MLP are optimized using a grid search algorithm, resulting in 48384 trained ANNs. The best models feature 4 hidden layers with 4 neurons each, employing a ReLU activation function, achieving high R2 scores of 0.986 for deceased patients, 0.98599 for confirmed cases, and 0.99429 for deceased patients. Cross-validation further validates the robustness of the models, with slight reductions in R2 scores.

Zhang *et al.* (2020) proposed a real-time system for predicting sentiment on Twitter regarding the coronavirus pandemic, leveraging the platform's significant role as a data source for analysis and prediction. The system comprises offline and online components. Historical tweets from January 23, 2020, to June 1, 2020, filtered by COVID-19 hashtags, form the offline dataset, from which essential features are extracted using n-gram and TF-ID methods. Five machine learning algorithms are evaluated, including decision tree, logistic regression, k-nearest neighbors, random forest, and support vector machine, with random forest performing the best. The online prediction pipeline utilizes Twitter Streaming API, Apache Kafka, and Apache Spark. The experimental results highlight the effectiveness of the random forest model with unigram feature extraction for sentiment prediction on real-time Twitter data.

Marappan *et al.* (2022) addressed the pressing concern of accurately predicting the trends and mortality rates associated with the Coronavirus Disease 2019 (COVID-19) caused by the SARS-CoV-2 virus, which has become a global health crisis. While statistical and epidemiological models are widely used for this purpose, they often suffer from drawbacks such as insufficient information and high uncertainty levels, leading to lower prediction accuracy. To overcome these limitations, the researchers propose a novel model leveraging machine learning techniques to forecast infection trends and mortality rates more accurately. The study focuses on employing various soft computing methods to design and evaluate the performance of the model in predicting and analyzing COVID-19 infection and mortality rates. The aim is to develop a model that outperforms existing methods in accurately predicting COVID-19 case trends and mortality rates.

Agarwal, Sharma, & Kunwar (2022) highlighted the significant impact of the COVID-19 pandemic and the critical need for accurate prediction of case numbers to ensure adequate hospital bed availability. Utilizing machine learning techniques, they aim to forecast future COVID-19 cases based on past data. What distinguishes their work is achieving a 90% prediction accuracy, providing valuable insights through various graphical representations. These visualizations aid users in comprehending COVID-19 case information effectively and enable comparisons between different trends in case numbers.

Awwalu *et al.* (2020) created a decision support system employing multinomial Naive Bayes for early COVID-19 detection based on observed indications and symptoms. The study didn't go into detail about the detection accuracy percentage, but the testing findings showed that the disease could be detected with a high degree of precision.

Muhammad *et al*. (2022) developed a deep transfer learning method based on a CNN-based model for the purpose of detecting COVID-19 in chest X-ray images. They used a variety of pre-trained networks, including VGG19, Xception, ResNet152, ResNet152v2, ResNet101, ResNet101v2, DenseNet201, DenseNet169, and DenseNet121, for evaluation. In order to improve the quality of the images, these pre-trained models were changed by include extra CNN layers, such as average pooling, dense layers, and ReLU with softmax activation function. Chest X-ray images of COVID-19 patients that were made accessible to the public were subjected to the experimental findings to determine whether they were positive or negative. The model performed admirably, demonstrating its usefulness in identifying COVID-19 instances with 95% accuracy, 94% precision, 88% recall value, and 92% F1-scores.

Shamik and Anurag (2021) suggested a Convolutional Capsule Network model called VGG-CapsNet for diagnosing and identifying COVID-19 using chest X-ray images. The VGG-CapsNet method distinguished COVID-19 patients from non-COVID-19 cases with a high accuracy of 97% and classified COVID-19, normal, and viral pneumonia cases with a high accuracy of 92%. The outcomes show that the VGG-CapsNet model is capable of correctly detecting COVID-19 cases and differentiating them from those with other respiratory diseases.

Sait *et al.* (2021) proposed a deep learning-based multimodal strategy for diagnosing COVID-19, utilizing chest X-ray pictures and breathing sounds. To lower false-negative rates, they employed the Inception-v3 model in conjunction with Multi-Layered Perceptron (MLP) for transfer learning. Using a curated CXR picture dataset, the suggested model achieved a high accuracy of 99.66% for COVID-19 detection. Additionally, the study of respiration sounds yielded an initial accuracy of 80%. The approach was operationalized as a smartphone application to enhance accessibility and alleviate the strain on healthcare institutions.

Harmon *et al*. (2020) created an artificial intelligence-based model for identifying COVID-19 pneumonia based on datasets from chest CT scans. The model showed up to 90.8% accuracy, 84% sensitivity, and 93% specificity. It successfully separated non-COVID-related pneumonias with high specificity across various patient demographics and detected CT scans connected to COVID-19 pneumonia.

Pola (2021) developed a model for the early diagnosis of COVID-19 utilizing CT scan images using the Xception architecture, a deep transfer learning technique based on CNN. A dataset of 2,482 correctly classified CT scans was used to evaluate the model, and it performed with an accuracy of 98.89%.

Rani *et al.* (2022) proposed a deep learning-based multimodal approach for the efficient identification of coronavirus by analyzing the genetic similarity among severe acute respiratory disorders using chest radiographs and genomic sequences. The datasets were gathered from repositories such as the National Centre for Biotechnology Information, GitHub, and Kaggle. The suggested model proved to be a useful tool for quickly classifying infected and non-infected genomes with an accuracy of 99.27% and a sensitivity of 95.47%.

Shikang *et al.* (2021) created a hybrid model that incorporated data from public health centers, chest X-rays, and CT scans to predict and identify COVID-19 individuals at high risk. The approach beat the performance of a traditional linear regression model by adding deep learning and big data techniques, yielding an amazing accuracy rate of 95%.

Shahanaz *et al.* (2022) developed a deep learning approach for the early classification and identification of COVID-19 utilizing chest X-ray images by applying deep learning multi-layered networks, their algorithm successfully identified X-ray pictures as positive or negative for COVID-19. People who had been infected with the coronavirus and had multilobar lesions, as determined by skilled radiologists, made up the dataset used in the study.

Subhalakshmi *et al.* (2022) created a fusion model based on deep learning for the purpose of diagnosing and categorizing COVID-19 using CT images. Their model included feature extraction, classification using VGG16 and Inception v4 models, and pre-processing using Weiner Filtering. Combining deep features with a Gaussian Naive Bayes classifier made it easier to recognize and divide test CT images into various categories. The model performed with 96.53% sensitivity, 95.81% specificity, 96.81% accuracy, and 96.73% F1-score in experiments utilizing an open-source COVID-CT dataset of 760 CT digital images.

Chauha and Modi (2022) created a multi-modal severity prediction system for COVID-19 using machine learning methods. They used datasets that comprised X-ray pictures and laboratory test results while they investigated Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and ANN classifiers. The model had a 96% accuracy rate.

Afreen & Reddy (2022) suggested a novel transfer learning-based deep learning algorithm for the accurate categorization and detection of COVID-19 utilizing lung X-rays. For feature extraction, they used a pre-trained CNN model, such as Distributed Deep Convolutional VGGNet. The severity levels (mild, moderate, and severe) might be determined by the model. The model performed with an 88% F1-score according to experimental findings.

Ghomi *et al*. (2020) created a deep learning method to precisely diagnose and track COVID-19 pneumonia lesions using lung CT images. A dataset of 2,469 CT scan pictures from 55 COVID-19 patients and 41 healthy persons was used in their investigation. These photos were used to train a Deep Convolutional Neural Network (DCNN) model based on Detectron2, which demonstrated remarkable performance with 95.4% accuracy, 92.8% sensitivity, and 96.1% specificity. When applying a threshold of 0.4% for lung connection percentage, the model showed strong capacity to detect COVID-19 pneumonia patients, with 98.2% sensitivity and 88.5% specificity.

Soumyajit, Somnath, and Arijit (2021) proposed an effective approach for categorizing and predicting COVID-19 using Mask R-CNN on chest X-ray images. With their suggested model, they collected a dataset of 668 chest X-ray pictures and attained high accuracy of 96.98%, specificity of 97.36%, and precision of 96.60%.

Bashar *et al.* (2021) created a model for COVID-19 pneumonia detection based on improved deep learning methods and X-ray image data. They acquired a dataset of 21,165 X-ray images from the online community Kaggle, which includes categories for normal, COVID-19, lung opacity, and viral pneumonia. To expand dataset size and prevent overfitting, data augmentation techniques were used, leading to a categorization accuracy of 95.63%.

Sekeroglu and Ozsahin (2020) created a model for COVID-19 detection from chest X-ray images through the use of convolutional neural networks. Their data set included 225 confirmed COVID-19 cases, 4,292 pneumonia patients, and 1,583 healthy cases with X-ray scans. The model performed admirably, with average scores for the receiver operating characteristic-area under the curve of 96.51%, sensitivity of 93.84%, specificity of 99.18%, and accuracy of 98.50%.

Bouchareb *et al.* (2021) created an artificial intelligence-driven assessment algorithm for COVID-19 using radiological images such as computed tomography (CT) and chest X-ray (CXR). Their findings imply that the suggested AI-based evaluation model may be a useful tool for COVID-19 diagnosis, monitoring, and prognosis.

Kapal *et al.* (2021) suggested a Hierarchical Convolutional Networks (HCN) model in order to diagnose COVID-19 using chest X-ray (CXR) images. To extract pertinent information for the CXR modality, the HCN model used convolutional layers from trained networks, such as the CovidNet. The investigations showed that the performance of the HCN design was superior to that of earlier studies, enabling precise triage of possible COVID-19 patients using CXR images.

Himadri *et al.* (2020) created a deep neural network model that integrated various data modalities, such as Computed Tomography (CT) scans and chest X-rays (CXRs) for the purpose of predicting, screening, and diagnosing COVID-19 positive cases. Their model's excellent accuracy rating of 96.28% was attained.

Vedika *et al*. (2022) created a better deep learning model utilizing chest X-ray images for early categorization and prediction of COVID-19. 2,905 pictures made up their collection, including 1,341 healthy cases, 1,345 viral pneumonia cases, and 219 COVID-19 patients. According to the findings, their AlexNet-based model had an accuracy of 97.6%, a precision of 98%, a recall of 97%, and an F1 score of 98%.

Ju *et al.* (2022) created a unique deep learning algorithm to help radiologists identify and categorize COVID-19 pneumonia from CT images. The technique addresses the RT-PCR's sensitivity limits for diagnosing COVID-19. They used a dataset of 37,049 instances, which included cases of COVID-19, cases of Community-Acquired Pneumonia (CAP), and normal cases. A test set sensitivity of 98.03%, 89.28%, and 92.15%, respectively, were reached by the suggested model, which is based on U-Net and ResNet-50, in the diagnosis of normal cases, CAP, and COVID-19 patients. The overall diagnostic accuracy of the deep learning model was 93.84%, exceeding that of untrained physicians and coming very close to that of specialists. Furthermore, compared to radiologists, the proposed deep learning model showed notable time economy.

Nandi & Mulimani (2021) created a hybrid deep learning model for COVID-19 identification from chest X-rays by fusing the ResNet50 and MobileNet architectures,. On two publicly accessible

COVID-19 chest X-ray datasets, the hybrid model's accuracy was 84.35% and 94.43%, proving its efficacy in COVID-19 identification.

Al-Shehri *et al*. (2022) created a deep learning-based method for the purpose of detecting COVID-19 from CT scans and X-ray pictures. In terms of COVID-19 detection accuracy, their model fared better than those already in use. They hope that the results of their study will help doctors make well-informed choices about the health of their patients.

Chokri *et al.* (2022) suggested a hybrid AI-based model that combined a Deep Convolutional Neural Network (D-CNN) built on the MobileNetV2 architecture with the Advanced Encryption Standard (AES) for secure data transfer to notify medical workers in order to detect and prevent COVID-19. The experimental outcomes demonstrated the effectiveness of the proposed model, obtaining a 99.7% accuracy rate in mask detection with a runtime of just 1.54 seconds

Nagendra *et al.* (2022) discussed about the global increase in COVID-19 testing and the strain placed on medical resources, prompting the development of a machine learning model aimed at predicting COVID-19 likelihood based on symptoms. They mentioned that using a dataset compiled from online health records and Kaggle, the model was trained with four classification algorithms. Among these algorithms, KNN was identified as the most effective for COVID-19 prediction. They emphasized the simplicity of KNN compared to other complex algorithms, noting its utility in straightforward tasks such as weather and health predictions. The authors also highlighted the challenge of managing prediction outputs with other algorithms due to their complexity and fluctuating weights with data changes.

Thangamma *et al.* (2021) highlighted the global impact of the COVID-19 pandemic, which they noted has resulted in widespread deaths, economic downturns, and social upheaval since its emergence in Wuhan, China. They pointed out that the pandemic has affected nearly every country, endangering lives and livelihoods worldwide. According to them, efforts to combat the virus include the urgent need for an effective vaccine and the application of Artificial Intelligence (AI) and Machine Learning (ML) technologies to understand and mitigate its spread. The paper reportedly emphasized the role of ML frameworks in analyzing disease transmission patterns and predicting future trends, offering insights into effective strategies for combating infectious diseases like COVID-19.

Prakash *et al.* (2020) reported that COVID-19, known as Corona Virus Disease-2019, belongs to the genus of Coronaviridae. They stated that the virus, which lacks a vaccine, has caused unpredictable havoc in human lives and financial and economic systems worldwide. They mentioned that their analysis of COVID-19 datasets aimed to understand which age group is mostly affected by the virus. They also mentioned that they built different prediction models using machine learning algorithms and evaluated their performances, finding that Random Forest Regressor and Random Forest Classifier outperformed other machine learning models such as SVM, KNN+NCA, Decision Tree Classifier, Gaussian Naïve Bayesian Classifier, Multilinear Regression, Logistic Regression, and XGBoost Classifier.

Reddy *et al.* (2023) addressed the significant impact of the COVID-19 pandemic on the global economy and public health. They mentioned that precise forecasts of the virus's progression are crucial for policymakers and public health experts to allocate resources effectively. They introduced a methodology utilizing Support Vector Machines (SVM), Multiple Linear Regression (MLR), and Bayesian Ridge Regression (BRR) algorithms to forecast the number of COVID-19 cases. They claimed that their approach provided a more accurate forecast by considering various socioeconomic and environmental variables. By employing feature selection techniques, they identified key variables for predicting COVID-19 cases. Their model, trained with SVM, MLR, and BRR algorithms, was evaluated using metrics such as Mean Absolute Error (MAE), coefficient of determination (R^2), and Root Mean Squared Error (RMSE). Their results indicated a high level of accuracy in forecasting COVID-19 cases using their proposed approach, providing insights into influential variables for decision-making by health professionals and policymakers.

Sujath, Chatterjee, & Hassanien (2020), proposed a machine learning forecasting model for the COVID-19 pandemic in India caused by the Coronavirus disease (COVID-19). They described COVID-19 as a respiratory illness with symptoms similar to influenza, including cold, cough, fever, and, in severe cases, difficulty breathing. Recognizing COVID-19 as a global pandemic, they highlighted the ongoing research using various mathematical models to predict its progression, which may be subject to bias. Their model aimed to predict the spread of COVID-19, utilizing linear regression, Multilayer perceptron, and Vector autoregression methods on Kaggle data. By analyzing data on confirmed, death, and recovered cases across India, they anticipated the epidemiological pattern and rate of COVID-19 cases. They emphasized the importance of maintaining continuous case definition and data integration for further assessment and future perspectives.

Basha *et al.* (2021) introduced a COVID-19 Prediction model using Machine Learning to address the challenging issue posed by the structure of Lung Nodule, where cells often overlap. They noted the emergence of the COVID-19 pandemic since late December 2019 in Wuhan, China, leading to serious illness and potential death due to alveolar damage and respiratory failure. While laboratory testing such as RT-PCR is the standard for diagnosis, it can yield false negatives and face resource shortages during the pandemic. Thus, they proposed a weakly supervised deep learning approach to detect and classify COVID-19 infections from CT images, aiming to reduce manual labeling requirements while ensuring accurate infection detection and differentiation from non-COVID cases.

Muhammad *et al.* (2020) proposed predictive data mining models for Novel Coronavirus (COVID-19) Infected Patients Recovery due to the absence of clinically proven vaccines or drugs for the pandemic. They aimed to address the urgent need for efficient solutions to contain and manage COVID-19 spread using non-clinical approaches like data mining and artificial intelligence.

Developing models based on epidemiological data from South Korea, they utilized various algorithms including decision tree, support vector machine, naive Bayes, logistic regression, random forest, and K-nearest neighbor to predict recovery time, high-risk age groups, and likelihood of recovery. The decision tree model exhibited superior efficiency with a remarkable accuracy of 99.85%, surpassing other algorithms tested.

Becerra-Sánchez *et al.* (2022) discussed the strain on medical care quality worldwide due to the COVID-19 pandemic, prompting the need for efficient prediction of infections and their outcomes. They propose an algorithmic analysis for predicting the health status of COVID-19 patients in Mexico, utilizing various models including KNN, logistic regression, random forests, ANN, and majority vote. These models incorporate risk factors to predict patient mortality. The ANN-based model demonstrates the highest success, achieving 90% accuracy and an F1 score of 89.64%. Analysis indicates that pneumonia, advanced age, and intubation requirement are the most influential risk factors for COVID-19 mortality in Mexico.

Majumder *et al.* (2021) introduced an Intelligent System for COVID-19 Case Prediction utilizing the Logistic Regression framework within a critical global context. With COVID-19 causing significant impact worldwide, including 3.23 million active cases and 59,449 deaths in India alone, efforts in medicine and vaccine development intensify. Concurrently, research in machine learning and AI aims to predict disease spread and detect virus presence in patients. The research proposed a Logistic Regression-based method to assess COVID-19 risk by analyzing multiple symptoms, offering potential advancements in medical science.

Ahmed (2020) conducted an analysis and forecasting study on the COVID-19 outbreak in Ethiopia using Machine Learning techniques. Recognizing the significant impact of coronavirus outbreaks on global health, the study emphasizes the crucial role of Machine Learning models in disease prediction and forecasting. Utilizing Support Vector Machine (SVM) and Polynomial Regression (PR) models, the study predicts the aggressive risk of Covid-19 based on confirmed, recovered, and death cases. Results indicate that SVM outperforms PR in predicting the pandemic's progression in Ethiopia. The study suggests an impending increase in hospital shortages and quarantine facilities in Ethiopia due to the escalating pandemic, particularly evident in the mid of July 2020.

Brinati *et al.* (2020) conducted a feasibility study on using machine learning to detect COVID-19 infection from routine blood exams. They highlighted the challenges posed by the COVID-19 pandemic, emphasizing the limitations of the current gold standard test real-time Reverse Transcription Polymerase Chain Reaction (rRT-PCR). To address the need for alternative, faster, and more accessible tests, they developed two machine learning classification models using hematochemical values from routine blood exams. These models demonstrated accuracies ranging from 82% to 86% and sensitivities between 92% and 95%, comparable to the gold standard. Additionally, they created an interpretable Decision Tree model to aid clinicians in interpreting blood tests for COVID-19 suspect cases. The study showed the feasibility and clinical effectiveness of using blood tests analysis and machine learning as an alternative to rRT-PCR for identifying COVID-19 positive patients, particularly beneficial in countries facing shortages of rRT-PCR reagents and specialized laboratories. They also provided a Web-based tool for clinical reference and evaluation.

Das, Ayus, & Gupta (2023) discussed the emergence of the COVID-19 pandemic originating in Wuhan, China, and its global threat. They emphasized the integration of artificial intelligence, particularly machine learning and deep learning models, for efficient COVID-19 detection. The study reviewed existing ML/DL models utilized for COVID-19 detection, aiming to provide a concise overview to research experts for future exploration. The researchers utilized various ML/DL models to extract significant features and classify health conditions in COVID-19 patients using image modalities like CT-Scan and X-Ray. From over 200 collected research papers, 50 were selected, showcasing ML/DL models' high accuracy of 99% and above in COVID-19 classification. The study highlighted the clinical applications and the importance of ML/DL models in medical diagnosis and research, while also acknowledging the need to address limitations such as overfitting to enhance their effectiveness and efficiency in the future.

Meraihi *et al.* (2022) discussed the global impact of the COVID-19 pandemic in 2020, highlighting the absence of treatment and the consequent surge in research efforts across various fields. In the realm of Computer Science, methods for COVID-19 diagnosis, detection, and prediction have been developed, with a predominant focus on data science and Machine Learning (ML) techniques. The paper provides an overview of over 160 ML-based approaches sourced from platforms like Elsevier, Springer, ArXiv, MedRxiv, and IEEE Xplore. These approaches are categorized into Supervised Learning-based and Deep Learning-based methods, specifying the ML algorithms used and parameters employed. The study discusses the distribution of approaches and reveals that Deep Learning, particularly Convolutional Neural Networks (CNN), is predominant, constituting 79% of cases, while Supervised Learning methods are less common, utilized in only 16% of the reviewed approaches, employing algorithms such as Random Forest, Support Vector Machine (SVM), and Regression algorithms.

Sher, Rehman, & Kim (2023) addressed the significant impact of COVID-19 and its variants on global health and economy, emphasizing the importance of early diagnosis to prevent further spread. They highlighted the role of technology, particularly the Internet of Things (IoT) and social IoT (SIoT), in leveraging data from smart devices to combat the disease. Utilizing data mining techniques, the study employed five supervised machine learning (ML) strategies to create a model for analyzing and forecasting COVID-19 presence using the Kaggle dataset COVID-19 Symptoms and Presence. They utilized algorithms such as Decision Tree, Random Forest, K-Nearest Neighbors, Naive Bayes, and Integrated Decision Tree, with Decision Tree yielding the highest accuracy of

98.42% and a root mean square error of 0.11. The developed model is envisioned to be valuable for early disease prediction and diagnosis with diverse datasets.

Sultana *et al.* (2022) highlighted the significant impact of COVID-19, emphasizing its varied symptoms and global spread, prompting researchers to employ statistical methodologies to model the pandemic's progression and its impact on different regions. Utilizing data from the COVID-19 repository at Johns Hopkins University, the study focused on Asian countries and employed Machine Learning classifiers such as Linear Regression, Multi-Layer Perception, and Vector Auto Regression to predict COVID-19 outbreaks. Their analysis revealed varying prediction accuracies, with Linear Regression and Multi-layer Perception predicting 75% for confirmed cases, 62% for death cases, and 20% for recovered cases in Asia. The research underscores the importance of deep analysis and novel methodologies to assist affected individuals in the region.

Butt *et al.* (2020) conducted a technical review comparing the performance of multiple convolutional neural network (CNN) models in classifying CT samples with COVID-19, Influenza viral pneumonia, or no-infection. Their study achieved an AUC of 0.996 (95%CI: 0.989–1.00) for distinguishing Coronavirus vs Non-coronavirus cases per thoracic CT studies. They reported a sensitivity of 98.2% and a specificity of 92.2%, highlighting the superior sensitivity and specificity of radiographic patterns on CT chest scans compared to RT-PCR detection of COVID-19, particularly in the early stages.

Apostolopoulos & Mpesiana (2020) utilized a dataset comprising X-ray images from patients with common bacterial pneumonia, confirmed Covid-19 disease, and normal incidents to automatically detect Coronavirus disease. Their study aimed to assess the performance of state-of-the-art convolutional neural network architectures for medical image classification, employing Transfer Learning. By leveraging transfer learning, the detection of various abnormalities in small medical image datasets became achievable, yielding remarkable results. Two datasets were used, one with

1427 X-ray images and another with 1442 images, both sourced from public medical repositories. Deep Learning with X-ray imaging demonstrated potential in extracting significant biomarkers related to Covid-19, achieving an accuracy, sensitivity, and specificity of 96.78%, 98.66%, and 96.46%, respectively. These findings suggest the potential incorporation of X-rays into Covid-19 diagnosis, pending further evaluation by the medical community.

Luján-García et al. (2020) introduced a transfer learning method designed for the classification and visualization of pneumonia, a major cause of mortality in children under five years old. Leveraging Chest X-ray images, the study utilized Convolutional Neural Networks (CNNs), specifically the Xception Network with pre-trained weights on ImageNet for initialization. The model automatically classified 3883 chest X-ray images depicting pneumonia and 1349 labeled as normal. The proposed method demonstrated competitiveness with state-of-the-art approaches, achieving precision (0.84), recall (0.99), F1-score (0.91), and area under the ROC curve (0.97). These promising outcomes position the proposal as a viable alternative, particularly beneficial in regions lacking equipment and specialized radiologists.

Fanelli & Piazza (2020) conducted an analysis of the temporal dynamics of the COVID-19 outbreak in China, Italy, and France from January 22nd to March 15th, 2020. Their study suggests universality in epidemic spreading across these countries, allowing for meaningful application of simple mean-field models to quantify the epidemic's progression, including the peak number of confirmed infected individuals and its timing. The analysis within a susceptible-infected-recovered-deaths model indicates consistent recovery rates across countries, while infection and death rates vary. In Italy, the peak of infections is estimated around March 21st, 2020, with approximately 260,000 infected individuals and 18,000 deaths. Drastic containment measures can reduce the infection rate and quench the epidemic peak, but require concerted efforts from the population.

Panwar *et al.* (2020) introduced nCOVnet, a deep learning-based method for fast COVID-19 detection using X-rays. With a training accuracy ranging from 93% to 97%, nCOVnet demonstrates promising results for classifying COVID-19 patients with 97.97% confidence and negative cases with 98.68% confidence. Its prediction accuracy of 97.62% suggests its potential as a substitute for RT-PCR, providing results in under 5 seconds compared to RT-PCR's 4-10 hours. The model addresses limitations of RT-PCR, such as cost and availability, by utilizing existing X-ray machines in hospitals. By rapidly detecting COVID-19 cases, nCOVnet aids in timely isolation and control of community transmission, crucial for managing the pandemic effectively. The model's unbiased results, achieved through rigorous training, offer valuable support to healthcare professionals in managing COVID-19 patients efficiently.

Ozturk *et al.* (2020) developed an automated detection system for COVID-19 cases using deep neural networks with X-ray images. The COVID-19 pandemic, originating in Wuhan, China, has spurred a global health crisis, necessitating swift and accurate diagnosis to curb transmission. Radiological imaging has emerged as a valuable tool for detecting COVID-19-related abnormalities. Leveraging advanced artificial intelligence techniques, their model achieves high accuracy rates of 98.08% for binary classification (COVID vs. No-Findings) and 87.02% for multi-class classification (COVID vs. No-Findings vs. Pneumonia) using raw chest X-ray images. Employing the DarkNet model for classification, their system offers potential assistance to radiologists in preliminary screening, with the model readily deployable via cloud services for immediate patient evaluation.

2.3.1 Summary of Related Works Reviewed

The empirical studies on COVID-19 categorization and prediction that were carried out by various researchers are summarized in Table 2.2. The table contains information about the author's name and year of publication, the study's field, the methodologies utilized, the data source, the kind and size of the dataset, the evaluation metrics employed, and the weaknesses.

Table 2.2: Summary of related work reviewed (2019 – 2024)

S/N	Author(s)/Year	Algorithm(s)	Type of Dataset	Accuracy (%)	Weaknesses	
					XAI	Concurrent/ Parallel Execution
1	Ngabo et al. (2021)	Naïve Bayes, Logistic Regression, Decision Tree, KNN	Clinical data	NB=94.50% KNN=99.30% DT =99.30% LR =91.70%	No	No
2	Kaushik et al. (2023)	Hybrid Deep Transfer Learning	Chest X-ray images	Not specified	No	No
3	Purohit et al. (2020)	CNN with multi-image augmentation	Chest X-ray & CT scan images	CT: accuracy 95.38% X-ray: accuracy 98.97%	No	No
4	Bukhari et al. (2020)	ResNet-50 CNN	Chest X-ray	98.18% accuracy, 98.19 F1-score	No	No
5	Fu et al. (2020)	Convolutional Neural Network	CT scan images	Acc. 98.8%, Sens. 98.2%, Spec. 98.9%, PPV 94.5%, NPV 99.7%	No	No
6	Wang et al. (2020)	COVID-Net CNN (open-source)	Chest X-ray images	Not specified	Yes	No
7	Qiao et al. (2020)	FLANNEL (ensemble CNN + focal loss)	Chest X-ray	81.68% (F1 score)	No	No
8	Loey et al. (2020)	GAN + Deep Transfer Learning	Chest X-ray (COVID-19,	85.2% accuracy	No	No
9	Khan et al. (2020)	3D Deep Learning (3D ResNets,	CT scan (CC-19 & COVID-CT)	Not specified	No	No
10	Anwar & Zakir (2020)	EfficientNet	Chest CT scan	89.7% accuracy	No	No
11	Chandra et al. (2020)	ACoS (radiomic texture + ensemble classification)	Chest X-ray (normal, suspected, nCOVID-19)	Not specified	No	No
12	Ismael & Şengür (2020)	Deep Feature Extraction + Fine-tuning CNNs (ResNet, VGG) + End-to-End CNN	Chest X-ray (COVID-19, normal)	94.7% accuracy	No	No
13	Karhan & Akal (2020)	ResNet50 CNN	Chest X-ray (COVID-19, normal)	Not specified	No	No
14	Jain et al. (2020)	CNNs (Inception V3, Xception, ResNeXt)	Chest X-ray (COVID-19, normal)	97.97% accuracy	No	No

Table 2.2: Summary of related work reviewed (Cont'd)

S/N	Author(s)/Year	Algorithm(s)	Type of Dataset	Accuracy (%)	Weaknesses	
					XAI	Concurrent/ Parallel Execution
15	Foysal & Aowlad (2021)	Ensemble Deep CNN	CT scan	96% accuracy	No	No
16	Majid et al. (2021)	Transfer Learning (ResNet50, DarkNet53) + Extreme Machine Learning	CT scan (Pneumonia, COVID-19, Healthy)	95.6% accuracy	No	No
17	Khalil et al. (2022)	EfficientnetB4	Chest X-ray	97% accuracy	No	No
18	Redie et al. (2022)	Modified DarkCovidNet (CNN)	Chest X-ray	99.53% sensitivity	No	No
19	Idupulapaati & Sreedevi (2022)	CNN inspired by AlexNet	Chest X-ray	Not specified	No	No
20	Udayaraju et al. (2023)	GW-CNNDC (fine-tuned deep learning)	Chest X-ray	Not specified	No	No
21	Mohammadi et al. (2023)	Multilayer perceptron neural network	Blood tests	Not specified	No	No
22	Huyut (2022)	Supervised ML (LWL, etc.)	Blood tests & demographics	97.86% accuracy	No	No
23	Abayomi-Alli et al. (2022)	Ensemble learning (CNN, DNN, ExtraTrees)	Blood tests	99.28% accuracy	No	No

Table 2.2: Summary of related work reviewed (Cont'd)

S/N	Author(s)/Year	Algorithm(s)	Type of Dataset	Accuracy (%)	Weaknesses	
					XAI	Concurrent/ Parallel Execution
24	Rehman et al. (2021)	Machine learning	Symptoms	97% accuracy	No	No
25	Alves et al. (2021)	Machine learning (Random Forest)	Blood tests	88% accuracy	Yes (DTX & Criteria Graph)	No
26	Gozes et al. (2020)	Deep learning algorithm	Chest CT scans	Not specified	No	No
27	Mei et al. (2020)	AI with CT, clinical data, etc.	Chest CT scans, clinical data, etc.	Not specified	No	No
28	Ferrari et al. (2020)	Blood test analysis	Blood tests	70% accuracy	No	No
29	Cheng et al. (2020)	Review of diagnostic tests	Various	Not applicable	No	No
30	Rehman et al. (2021)	CNN (ResNet50) based CAD system	Chest X-ray	98% accuracy	No	No
31	Wang et al. (2020)	AlexNet deep learning	Chest CT scans	Not specified	No	No
32	Mporas & Naronglerdrit (2020)	Pre-trained deep CNN models (DenseNet, ResNet, Xception)	Chest X-ray	Not specified	No	No
33	Horry et al. (2020)	Optimized VGG19 CNN	X-ray, Ultrasound, CT scans	86% accuracy	No	No
34	Salih et al. (2020)	Modified AlexNet CNN	Chest X-ray	Not specified	No	No
35	Mukri (2023)	CNN	X-Ray scans (2 types)	96.66% accuracy	No	No
36	Ghafouri-Fard et al. (2021)	Various (MLP, ANFIS, LSTM, RNN)	Not specified	Not specified	No	No
37	Gonsalves & Nitnaware (2021)	ARIMA, LSTM, FB Prophet, Linear Regression, Random Forest	Not specified	Not specified	No	No

Table 2.2: Summary of related work reviewed (Cont'd)

S/N	Author(s)/Year	Algorithm(s)	Type of Dataset	Accuracy (%)	Weaknesses	
					XAI	Concurrent/ Parallel Execution
38	Subhan et al. (2021)	AI Boosted Random Forest	Patient health data	90% accuracy	No	No
39	Pal et al. (2019)	Neural Network	COVID-19 statistics & weather data	Not specified	No	No
40	Dokeroglu (2023)	Harris' Hawk Optimization (HHO)	Real-world & augmented COVID-19 datasets	98.15% Sensitivity	No	No
41	Car et al. (2020)	Multilayer Perceptron (MLP)	Public dataset - time-series to regression	Not specified	No	No
42	Zhang et al. (2020)	Random Forest	Twitter data	Not specified	No	No
43	Marappan et al. (2022)	Soft Computing Methods	COVID-19 Dataset	Not specified	No	No
44	Agarwal, Sharma & Kunwar (2022)	Machine Learning Techniques	COVID-19 Dataset	90% accuracy	No	No
45	Awwalu et al. (2020)	Multinomial Naive Bayes	Not specified	Not specified	No	No
46	Muhammad et al. (2022)	CNN-Based Model	Chest X-ray Images	95% accuracy	No	No
47	Shamik and Anurag (2021)	VGG-CapsNet	Chest X-ray Images	97% accuracy	No	No
48	Sait et al. (2021)	Inception-v3 with MLP	CXR Images	99.66% sensitivity	No	No
49	Harmon et al. (2020)	AI-Based Model	Chest CT Scans	90.8%	No	No
50	Pola (2021)	Xception Architecture	CT Scan Images	98.89% sensitivity	No	No
51	Rani et al. (2022)	Deep Learning-Based Model	Chest Radiographs and Genomic Sequences	99.27% sensitivity 95.47% accuracy	No	No
52	Shikang et al. (2021)	Hybrid Model	Public Health Data, CXR, CT Scans	95%	No	No
53	Shahanaz et al. (2022)	Deep Learning Approach	Chest X-ray Images	Not specified	No	No

Table 2.2: Summary of related work reviewed (Cont'd)

S/N	Author(s)/Year	Algorithm(s)	Type of Dataset	Accuracy (%)	Weaknesses	
					XAI	Concurrent/ Parallel Execution
54	Subhalakshmi et al. (2022)	Fusion Model	CT Images	96.53% sensitivity, 95.81% specificity, 96.81% accuracy, 96.73% F1-score	No	No
55	Chauha and Modi (2022)	SVM, DT, RF, ANN	X-ray Images, Lab Results	96% accuracy	No	No
56	Afreen & Reddy (2022)	DL Algorithm	Lung X-rays	88% F1-score	No	No
57	Ghomi et al. (2020)	DCNN Model	Lung CT Images	95.4% accuracy, 92.8% sensitivity, 96.1% specificity	No	No
58	Soumyajit et al. (2021)	Mask R-CNN	Chest X-ray Images	96.98% accuracy 97.36% specificity 96.60% precision	No	No
59	Bashar et al. (2021)	Improved DL Methods	X-ray Images	95.63% accuracy	No	No
60	Sekeroglu and Ozsahin (2020)	CNN Model	Chest X-ray Images	93.84% sensitivity 99.18% specificity 97.50% accuracy	No	No
61	Bouchareb et al. (2021)	AI-Driven Assessment Algorithm	CT and CXR Images	Not specified	No	No
62	Kapal et al. (2021)	HCN Model	Chest X-ray Images	Not specified	No	No
63	Himadri et al. (2020)	Deep Neural Network Model	CT Scans, CXRs	96.28% accuracy	No	No
64	Vedika et al. (2022)	Deep Learning Model	Chest X-ray Images	97.6% accuracy, 98% precision, 97% recall, 98% F1-score	No	No
65	Ju et al. (2022)	U-Net, ResNet-50	CT Images	93.84% accuracy	No	No
66	Nandi and Mulimani (2021)	ResNet50, MobileNet	Chest X-ray Images	84.35% and 94.43% accuracy	No	No
67	Al-Shehri et al. (2022)	Deep Learning-Based Method	CT Scans, X-ray Images	Better than existing methods	No	No
68	Chokri et al. (2022)	D-CNN, AES	Not specified	99.7% accuracy	No	No
69	Nagendra et al. (2022)	KNN, SVM, Random Forest, Decision Tree	Online health records & Kaggle (symptoms)	KNN most effective, no specific accuracy mentioned	No	No
70	Thangamma et al. (2021)	ML frameworks (not specified)	Not specified	Not specified	No	No

Table 2.2: Summary of related work reviewed (Cont'd)

S/N	Author(s)/Year	Algorithm(s)	Type of Dataset	Accuracy (%)	Weaknesses	
					XAI	Concurrent/ Parallel Execution
71	Prakash et al. (2020)	RF Regressor, RF Classifier, SVM, KNN+NCA, DT Classifier, Gaussian NB Classifier, Multilinear Regression, LR, XGBoost Classifier	COVID-19 datasets (age groups)	Not specified	No	No
72	Reddy et al. (2023)	SVM, MLR, BRR	Socioeconomic & environmental data	Not specified	No	No
73	Sujath, Chatterjee, & Hassanien (2020)	Linear Regression, Multilayer perceptron, Vector autoregression	Kaggle (confirmed, death, recovered cases)	Not specified	No	No
74	Basha et al. (2021)	Weakly supervised deep learning	CT images	Not specified	No	No
75	Muhammad et al. (2020)	Decision tree, SVM, Naive Bayes, Logistic Regression, Random Forest, KNN	Epidemiologica l data (South Korea)	99.85% accuracy	No	No
76	Becerra-Sánchez et al. (2022)	KNN, Logistic Regression, Random Forests, ANN, Majority Vote	Medical records (risk factors)	ANN: 90% accuracy	No	No
77	Majumder et al. (2021)	Logistic Regression	Symptoms data	Not specified	No	No
78	Ahmed (2020)	SVM, Polynomial Regression	Confirmed, recovered, death cases	Not specified	No	No
79	Brinati et al. (2020)	Machine Learning classification models	Routine blood exams	Not specified	Yes	No
80	Das, Ayus, & Gupta (2023)	Various ML/DL models	CT-Scan and X-Ray images	Not specified	No	No

Table 2.2: Summary of related work reviewed (Cont'd)

S/N	Author(s)/Year	Algorithm(s)	Type of Dataset	Accuracy (%)	Weaknesses	
					XAI	Concurrent/ Parallel Execution
81	Meraihi et al. (2022)	Supervised Learning-based, Deep Learning-based	Various platforms	Not specified	No	No
82	Sher, Rehman, & Kim (2023)	Decision Tree, Random Forest, K-NNs, Naive Bayes, ID3	Kaggle dataset	98.42%	No	No
83	Sultana et al. (2022)	Linear Regression, Multi-Layer Perception, Vector Auto Regression	COVID-19 repository at Johns Hopkins University	Not specified	No	No
84	Butt et al. (2020)	Convolutional Neural Network (CNN) models	Thoracic CT scans	Sensitivity: 98.2%, Specificity: 92.2%	No	No
85	Apostolopoulos & Mpesiana (2020)	Convolutional Neural Networks (CNNs)	X-ray images	Accuracy: 96.78%, Sensitivity: 98.66%, Specificity: 96.46%	No	No
86	Luján-García et al. (2020)	Xception Network with pre-trained weights on ImageNet	Chest X-ray images	Precision: 0.84, Recall: 0.99, F1-score: 0.91, AUC: 0.97	No	No
87	Fanelli & Piazza (2020)	Not specified	Not specified	AUC: 0.996, Sensitivity: 98.2%, Specificity: 92.2%	No	No
88	Panwar et al. (2020)	nCOVnet	X-ray images	Prediction accuracy: 97.97% for COVID-19 cases, 98.68% for negative cases	No	No
89	Ozturk et al. (2020)	DarkNet model	Chest X-ray images	Binary classification: 98.08%, Multi-class classification: 87.02%	No	No

2.3.2 Research Gap

After thoroughly examining eighty-nine related works out of the relevant studies on COVID-19 prediction, it is evident that significant progress has been achieved in employing machine learning and deep learning methods for classifying COVID-19 across various modalities. These studies

showcased advancements in utilizing techniques such as chest X-ray and CT image analysis, blood test interpretation, symptom data analysis, and genomic sequence analysis for COVID-19 classification.
While these approaches have shown impressive accuracies ranging from 82% to well above 99%, there remains a notable gap in knowledge regarding several aspects.

Firstly, despite the high accuracies achieved in many studies, there is limited exploration into the interpretability of the models, known as explainable artificial intelligence (XAI), which is crucial for understanding the reasoning behind model predictions, especially in medical decision-making.

Secondly, the majority of models do not support concurrent or parallel execution, potentially limiting their scalability and real-time application in clinical settings with large datasets or high throughput requirements.

Moreover, there is a need for more research into the generalizability and robustness of these models across diverse populations, imaging protocols, and disease stages to ensure their effectiveness in real-world scenarios.

Furthermore, the gap persists in the integration of multiple modalities or datasets to improve diagnostic accuracy and provide a comprehensive understanding of COVID-19 pathophysiology with high confidence rate.

Bridging these gaps will enhance the reliability, interpretability, and scalability of machine learning models for COVID-19 diagnosis, facilitating their deployment in clinical practice and public health initiatives.

CHAPTER THREE

METHODOLOGY

3.1 Research Methodology

In this study, we used quantitative research methods, such as structured interviews, experiments, observations, and secondary data to gather, examine, and interpret datasets of both images and clinical information.

3.2 System Analysis

This section involves studying and evaluating the existing system (conventional medical laboratory machines and artificial intelligence based techniques) for COVID-19 detection by gaining insights into its components, behavior, and performance in order to design and develop a new and improved system that will meet users' needs effectively.

3.2.1 Analysis of the Existing System (Conventional Medical Laboratory Machines)

Polymerase Chain Reaction (PCR), Antigen tests, Serological testing, Chest X-rays (CXR) and Computed Tomography (CT) Scans are the conventional medical laboratory machines for diagnosis of COVID-19.

(a) Polymerase Chain Reaction (PCR) Machine

A Polymerase Chain Reaction (PCR) machine, also known as a thermal cycler, is a laboratory instrument used for the detection and amplification of specific DNA sequences. PCR has been widely used in the detection of COVID-19 since the start of the pandemic.

The process of PCR involves three main steps: denaturation, annealing, and extension. These steps are repeated in cycles to amplify the target DNA sequence.

Mode of Operation of a PCR Machine for Detecting COVID-19:

1. Sample Collection: Nasopharyngeal swabs or other respiratory samples are collected from individuals suspected of having COVID-19. These samples contain genetic material (RNA) from the SARS-CoV-2 virus.

2. RNA Extraction: The genetic material (RNA) is extracted from the collected samples using specific laboratory protocols and kits. This step is crucial to isolate and purify the viral RNA.
3. Reverse Transcription: In the case of COVID-19, the extracted RNA is converted into complementary DNA (cDNA) through a process called reverse transcription. This step is necessary because PCR amplifies DNA, not RNA.
4. PCR Setup: The PCR machine is programmed with specific temperature and time settings to perform the PCR cycles. The machine consists of a thermal block with wells for holding reaction tubes or plates containing the PCR mixture.
5. PCR Amplification: The PCR mixture includes the cDNA template, primers (short DNA sequences that flank the target region), nucleotides (building blocks for DNA), and DNA polymerase enzyme. The machine heats the reaction mixture to a high temperature (typically around 95°C), which causes the DNA strands to separate (denaturation).
6. Annealing: The machine then cools the reaction mixture to a lower temperature (typically around 50-60°C), allowing the primers to bind (anneal) to the target DNA sequences.
7. Extension: The temperature is raised again (usually around 72°C), and the DNA polymerase enzyme extends the primers along the DNA strands, synthesizing new DNA strands complementary to the target sequences.
8. Cycling: The denaturation, annealing, and extension steps are repeated for a specific number of cycles (usually 30-40) to exponentially amplify the target DNA sequence. This amplification makes it easier to detect the presence of the SARS-CoV-2 genetic material.
9. Detection: Various methods can be used to detect the amplified DNA. One common approach is to incorporate fluorescent probes or dyes into the PCR mixture that emit a signal when bound to the target DNA. The PCR machine may have built-in detectors to measure the fluorescence emitted after each cycle, indicating the presence or absence of the target DNA sequence.

10. Analysis: The data collected during the PCR process can be analyzed using specific software such as Applied Biosystems™ Analysis Software, Bio-Rad CFX Maestro™ Software, Thermo Fisher Cloud PCR Analysis Software, and QuantStudio™ Design and Analysis Software to determine the presence or absence of the viral genetic material. The results are typically reported as positive or negative for COVID-19, based on the detection of the specific target sequences.

It should be noted that the PCR machines are essential tools in COVID-19 testing laboratories and have played a critical role in diagnosing and monitoring the spread of the virus. They offer high sensitivity and specificity, enabling accurate detection even at low viral loads.

Advantages of PCR Machine

PCR machines offer several advantages that have made them a vital tool in various fields of research, diagnostics, and molecular biology. Some key advantages of PCR machines are:

1. Sensitivity: PCR machines are highly sensitive and can detect even a small amount of target DNA or RNA in a sample. The amplification process allows for the exponential increase in the number of target sequences, making it easier to detect and analyze them.
2. Specificity: PCR machines can target specific DNA or RNA sequences with high specificity. By using primers designed to match specific regions of the target sequence, PCR machines ensure that only the desired genetic material is amplified. This specificity helps in accurate identification and differentiation of different organisms, genes, or variants.
3. Speed: PCR machines can rapidly amplify target DNA or RNA sequences. Each PCR cycle takes only a few minutes, and the entire amplification process can be completed in a matter of hours. This quick turnaround time is particularly valuable in situations that require fast and reliable results, such as diagnosing infectious diseases like COVID-19.
4. Versatility: PCR machines can be used for a wide range of applications. They are utilized in various fields, including medical diagnostics, genetic research, forensic analysis, agriculture,

and environmental studies. PCR techniques can be adapted to detect and analyze specific genes, mutations, or pathogens, making them highly versatile and applicable to different research areas.

5. Quantification: PCR machines can be used for quantitative analysis of DNA or RNA. By using specialized techniques such as quantitative PCR (qPCR) or real-time PCR, the amount of target DNA or RNA can be measured accurately. This feature is beneficial in determining gene expression levels, viral load in clinical samples, or monitoring the progression of diseases.
6. Automation: Modern PCR machines often come with automated features, making them user-friendly and efficient. These machines can automatically set and control the temperature cycles, detect fluorescence signals, and collect data. The automation reduces human error, enhances reproducibility, and allows for high-throughput processing of multiple samples.
7. Cost-effectiveness: PCR machines are widely available, and the reagents used in PCR are relatively affordable. This cost-effectiveness makes PCR a practical and accessible technique for various laboratories and research settings.
8. Non-invasive Testing: PCR machines can be used with a variety of sample types, including non-invasive samples like saliva, nasal swabs, or urine. This non-invasive collection of samples makes the testing process more comfortable for patients and simplifies sample collection in large-scale screening or surveillance programs.

Disadvantages of PCR Machine

While PCR machines have revolutionized molecular biology and diagnostics, they also come with certain limitations and disadvantages. Here are some of the main disadvantages of PCR machines:

1. False Positives and Contamination: PCR machines are highly sensitive, which can sometimes lead to false-positive results. Contamination during sample preparation or PCR setup can introduce external DNA or RNA, resulting in erroneous amplification and misinterpretation

of results. Strict laboratory practices and precautions are necessary to minimize the risk of contamination.

2. Inhibition and Bias: Certain substances present in the sample, such as contaminants or inhibitors, can interfere with the PCR reaction and inhibit DNA amplification. This can lead to false-negative results or reduced amplification efficiency. Additionally, PCR may exhibit bias toward certain DNA templates, impacting the accuracy of the results.
3. Limited Multiplexing Capacity: Multiplex PCR allows simultaneous amplification of multiple target sequences in a single reaction. However, the number of targets that can be reliably multiplexed is limited by factors such as primer design, primer interactions, and the complexity of the sample. As the number of targets increases, the risk of amplification bias and decreased sensitivity may also rise.
4. Complexity of Primer Design: PCR requires the design and synthesis of specific primers that target the desired DNA or RNA sequences. Primer design can be complex and challenging, especially when dealing with highly variable or rapidly evolving genetic targets. Inaccurate primer design can lead to inefficient amplification or false results.
5. Limited Information: PCR alone provides information about the presence or absence of a specific DNA or RNA sequence but does not offer details about its function or characteristics. Additional techniques, such as DNA sequencing or further analysis, are often required to gain a comprehensive understanding of the amplified target.
6. Limited Dynamic Range: PCR amplification is an exponential process, which means that high abundance targets may reach a plateau phase where further amplification is limited. This limited dynamic range can make it challenging to accurately quantify highly abundant targets or distinguish subtle differences in their levels.
7. Equipment and Infrastructure Requirements: PCR machines require specialized equipment and controlled laboratory conditions, including thermal cyclers, pipettes, and appropriate

biosafety measures. Setting up and maintaining the necessary infrastructure can be costly, especially for smaller or resource-limited laboratories.

8. Limited Portability: Traditional PCR machines are often bulky and not easily portable. This can be a limitation when on-site or field testing is required, especially in remote or resource-constrained areas. However, portable PCR machines are becoming more accessible, addressing this limitation to some extent.

(b) Antigen Tests for COVID-19

Rapid diagnostic techniques called antigen tests identify specific viral or bacterial antigens in a patient's sample. They are frequently utilized at the point of care and offer quick outcomes in only a few minutes. Although they are less sensitive than PCR tests, they are nonetheless useful for fast screening, surveillance, and the detection of infected people. PCR testing are utilized in conjunction with antigen tests, which aid in completing diagnostic procedures. Negative results, however, may need to be verified with a more sensitive test because false-negative results can happen.

Mode of Operation of Antigen Test for Detecting COVID-19:

Antigen tests for detecting COVID-19 follow the mode of operation as follow:

Sample Collection: A nasal swab or nasopharyngeal swab is taken from the patient. The swab collects a sample that may contain viral antigens if the person is infected with SARS-CoV-2, the virus that causes COVID-19.

1. Test Kit Preparation: The antigen test kit is prepared according to the manufacturer's instructions. This may involve adding reagents, buffers, or other components to the test device.
2. Sample Application: The collected sample is transferred or added to the designated area on the test device, such as a well or strip. This allows the sample to interact with the components of the test, including the antibodies that can specifically recognize and bind to the viral antigens.

3. Antigen-Antibody Interaction: If the patient's sample contains SARS-CoV-2 antigens, the viral antigens will bind to the specific antibodies present on the test device. This interaction forms an antigen-antibody complex.
4. Migration: Depending on the test format, the antigen-antibody complex may migrate across the test device through capillary action or other mechanisms. It moves along the test lines or regions where other antibodies are located.
5. Test Line Reaction: The test device contains a test line coated with antibodies that specifically bind to the SARS-CoV-2 antigens. If the viral antigen-antibody complex reaches this region, it will bind to the antibodies on the test line.
6. Signal Generation: When the antigen-antibody complex binds to the antibodies on the test line, a visible signal is generated. This signal may be in the form of a colored line or other indicators, depending on the test design.
7. Result Interpretation: The test device is examined after a specific incubation time (usually a few minutes). If a colored line appears at the test line region, it indicates the presence of SARS-CoV-2 antigens, suggesting a positive result for COVID-19. If no line appears at the test line, it indicates a negative result.
8. Control Line: Antigen tests also include a control line, which should always appear if the test is performed correctly. The control line ensures that the test device is functioning properly and helps in result interpretation.

Advantages of Antigen Machine for Detecting COVID-19

1. Rapid Results: Antigen machines provide rapid results, typically within 15-30 minutes. This quick turnaround time allows for immediate identification of infected individuals and enables prompt decision-making for isolation, treatment, and contact tracing.
2. Point-of-Care Testing: Antigen machines are often used as point-of-care tests, meaning they can be performed at the location where the patient is being evaluated, such as clinics,

hospitals, or community settings. This on-site testing capability eliminates the need for sample transportation and centralized laboratory processing, enabling faster diagnosis and management of COVID-19 cases.

3. Accessibility: Antigen machines are generally more accessible and easier to use compared to complex laboratory-based tests like PCR. They do not require highly specialized equipment or skilled laboratory personnel. As a result, antigen machines can be deployed in various healthcare settings, including resource-limited areas or regions with limited laboratory infrastructure.
4. Cost-Effectiveness: Antigen tests are often more cost-effective compared to PCR tests. They are generally less expensive and have lower operational costs, making them suitable for widespread testing and screening programs, particularly in situations where large-scale testing is required.
5. Early Detection of Infectiousness: Antigen tests can detect the presence of viral antigens during the early stages of infection when individuals are most contagious. This allows for the identification of infected individuals even before they develop symptoms, helping to prevent the spread of the virus.
6. Screening and Surveillance: Antigen machines are valuable tools for population-level screening and surveillance. Their rapid results and ease of use make them well-suited for conducting large-scale testing campaigns, identifying clusters, and monitoring the prevalence of COVID-19 in communities.
7. Complementary to PCR Testing: While PCR tests are considered the gold standard for COVID-19 diagnosis due to their high sensitivity, antigen machines serve as a valuable complement to PCR testing. They can be used for initial screening, triaging high-risk individuals, and providing quick results while waiting for confirmatory PCR testing.

8. Real-Time Decision-Making: The rapid results provided by antigen machines allow for immediate decision-making in various scenarios. They enable healthcare providers to quickly determine appropriate patient management, implement infection control measures, and facilitate timely contact tracing, ultimately aiding in curbing the spread of COVID-19.

Disadvantages of Antigen Machine for Detecting COVID-19

1. Lower Sensitivity: Antigen tests generally have lower sensitivity compared to PCR tests. They may not detect the virus in individuals with low viral loads or during the early stages of infection. This can result in false-negative results, potentially leading to undetected cases and the spread of the virus.
2. Higher False-Negative Rate: Due to the lower sensitivity, antigen machines have a higher chance of producing false-negative results compared to PCR tests. False negatives can provide a false sense of security and may require additional confirmatory testing if clinical suspicion remains high.
3. Limited Detection Window: Antigen tests are most effective in detecting COVID-19 during the early stages of infection when the viral load is typically higher. However, as the infection progresses or if the viral load decreases, the likelihood of obtaining a positive result decreases. This limited detection window may result in missed cases, especially if testing is delayed.
4. Specificity and Cross-Reactivity: Although antigen tests are generally specific to the target virus (SARS-CoV-2), there is a possibility of cross-reactivity with other similar viruses or non-specific binding. This can lead to false-positive results, indicating a person is infected when they are not. Test specificity can vary among different antigen tests, and manufacturers provide information on the specificity of their particular tests.
5. Need for Confirmatory Testing: Due to the limitations in sensitivity and specificity, confirmatory testing with PCR is often recommended for individuals with negative or inconclusive antigen test results, especially if there is a high suspicion of COVID-19. This

additional step increases the time, cost, and logistical requirements for obtaining a definitive diagnosis.

6. Operator Error and Training: While antigen machines are generally designed to be user-friendly, there is still a potential for operator error during sample collection or testing procedures. Proper training and adherence to the manufacturer's instructions are essential to ensure accurate and reliable results.
7. Variability in Performance: The performance of antigen tests can vary depending on several factors, including the specific test kit used, the quality of the test components, and variations in sample collection and handling. Some antigen tests may have better performance characteristics than others, and it's important to consider the specific test's performance data and limitations.
8. Limited Quantitative Information: Antigen tests provide qualitative results, indicating the presence or absence of viral antigens. They do not provide quantitative information about the viral load or the severity of the infection. This information is important for clinical decision-making and monitoring disease progression.

(c) Serological Testing for Antibodies

Serological testing, also known as antibody testing, is a diagnostic method used to detect the presence of antibodies in a person's blood sample. These antibodies are produced by the immune system in response to an infection, such as COVID-19.

Mode of Operation of Serological Testing for Detecting COVID-19

The mode of operation for serological testing to detect COVID-19 involves the following steps:

1. Sample Collection: A blood sample is collected from the individual being tested. This can be done through venipuncture (drawing blood from a vein) or a fingerstick (obtaining a small blood sample from a finger). The sample is typically collected in a tube or on a specialized collection device.

2. Sample Processing: The collected blood sample is then processed in a laboratory or a point-of-care setting. The processing involves separating the serum or plasma from the blood cells. Serum or plasma contains the antibodies that will be tested for COVID-19.
3. Test Method: There are different methods used for serological testing, such as enzyme-linked immunosorbent assays (ELISA), chemiluminescent immunoassays (CLIA), lateral flow immunoassays (LFIA), and neutralization assays. These methods detect and measure the presence of specific antibodies, usually immunoglobulin M (IgM) and immunoglobulin G (IgG), in the serum or plasma sample.
4. Test Execution: The serum or plasma sample is added to the test device or plate containing specific viral antigens. The antigens used are derived from SARS-CoV-2, the virus that causes COVID-19. If the person being tested has COVID-19 antibodies in their blood, these antibodies will bind to the viral antigens in the test.
5. Reaction and Detection: Depending on the test method, the reaction between the antibodies and viral antigens may produce a visible color change, a fluorescence signal, or a chemiluminescent reaction. This indicates a positive result, indicating the presence of COVID-19 antibodies in the sample.
6. Interpretation of Results: The test results are then interpreted based on the presence or absence of antibodies. A positive result suggests that the individual has been exposed to SARS-CoV-2 and has developed an immune response, either from a recent or past infection. A negative result indicates the absence of detectable antibodies, suggesting no or minimal exposure to the virus.
7. Reporting and Follow-up: The test results are documented and reported to the individual being tested and their healthcare provider. If necessary, further confirmatory testing or medical evaluation may be recommended based on the test results and the individual's clinical history and symptoms.

Advantages of Serological Testing for Detecting COVID-19

Serological testing for detecting COVID-19 antibodies offers several advantages in understanding the spread of the virus and assessing immunity. Here are some key advantages of serological testing:

1. Detection of Past Infections: Serological testing helps identify individuals who have been previously infected with COVID-19, including those who may have been asymptomatic or had mild symptoms. It provides a retrospective view of the infection and helps in estimating the true prevalence of the virus within a population.
2. Assessment of Immunity: Serological tests can help determine if an individual has developed antibodies against SARS-CoV-2, indicating some level of immunity to future infections. This information is valuable for assessing the potential protection against reinfection and guiding decisions related to vaccination strategies and public health measures.
3. Identification of Silent Spreaders: Serological testing can identify individuals who may have had an asymptomatic or mild infection but have developed antibodies. These individuals, even if they were not aware of their infection, may have contributed to the transmission of the virus. Identifying them helps in contact tracing and implementing appropriate control measures.
4. Population-level Surveillance: Serological testing provides a broader understanding of the spread of COVID-19 within a community or population. By testing a representative sample, public health officials can estimate the percentage of the population that has been infected, track the progression of the virus, and inform policy decisions accordingly.
5. Epidemiological Studies: Serological testing supports epidemiological research by providing data on the patterns and dynamics of COVID-19 transmission. It helps identify high-risk groups, understand the progression of the disease, and study the effectiveness of interventions, such as vaccination campaigns or public health measures.

6. Screening Blood Donations: Serological testing is used to screen blood donations for the presence of COVID-19 antibodies. This helps ensure the safety of the blood supply and prevent potential transmission of the virus through transfusions.
7. Research and Vaccine Development: Serological testing plays a vital role in evaluating the effectiveness of COVID-19 vaccines and studying the immune response to vaccination. It helps determine the seroconversion rates, antibody titers, and duration of immune response after vaccination, aiding in the development and refinement of vaccination strategies.
8. Long-term Monitoring: By periodically testing individuals over time, serological testing can help monitor the persistence and waning of COVID-19 antibodies. This data can provide insights into the long-term duration of immunity and inform decisions regarding booster doses or additional vaccination strategies.

Disadvantages of Serological Testing for Detecting COVID-19

Serological testing for detecting COVID-19 antibodies has the following disadvantages:

1. Timing of Antibody Development: Serological testing relies on the detection of antibodies, which may take time to develop. During the early stages of infection, it may not be possible to detect antibodies, leading to false-negative results. Testing too soon after exposure or during the incubation period may result in inaccurate results.
2. Inability to Detect Active Infections: Serological tests are not designed to diagnose active or acute COVID-19 infections. They detect antibodies that are produced in response to the infection, which means they are most effective in identifying past infections. For diagnosing current infections, molecular tests like PCR are more appropriate.
3. Variability in Antibody Response: The immune response to COVID-19 can vary between individuals. Some individuals may produce detectable antibodies, while others may have a weak or delayed antibody response. This variability can result in false-negative results, especially in individuals with compromised immune systems or mild infections.

4. Limited Window of Detection: Serological tests have a limited window of detection, as antibodies may wane over time. It is still unclear how long COVID-19 antibodies persist in the body and whether their presence guarantees immunity or protection against reinfection. Therefore, the timing of serological testing is critical to ensure accurate results.
5. Cross-Reactivity and Specificity: Serological tests may exhibit cross-reactivity with antibodies produced in response to other coronaviruses or similar respiratory infections. This can lead to false-positive results, where individuals may be incorrectly identified as having COVID-19 antibodies. Ensuring the specificity of the test is crucial to minimize false-positive results.
6. Test Performance Variability: The accuracy and performance of serological tests can vary depending on the specific test used, the quality of the test kits, and the timing of testing. Variability in sensitivity and specificity among different tests can affect the reliability of results and may require confirmation with additional tests.
7. Limited Quantitative Information: Serological tests provide qualitative information about the presence or absence of antibodies, but they may not provide quantitative data, such as antibody levels or titers. Quantitative information can be valuable in understanding the strength of the immune response and assessing potential immunity.
8. Interpretation Challenges: Interpreting serological test results can be complex, and false-positive or false-negative results can occur. Careful consideration of the individual's clinical history, symptoms, and exposure risk is necessary for accurate interpretation. Follow-up testing or additional diagnostic methods may be required to confirm results.

(d) Chest X-rays Machine for Detecting COVID-19

Chest X-ray machines are not specifically designed for detecting COVID-19. However, they are utilized as a diagnostic tool to assess the lung condition and aid in the evaluation of respiratory symptoms associated with COVID-19.

Mode of Operation of Chest X-rays Machine for Detecting COVID-19

The mode of operation of a chest X-ray machine for detecting COVID-19 involves the following steps:

1. Patient Preparation: The patient is positioned in front of the chest X-ray machine, typically standing or sitting upright. The patient may be asked to remove any metallic objects or clothing that could interfere with the imaging.
2. X-ray Generation: The chest X-ray machine consists of an X-ray generator and an X-ray detector. The X-ray generator emits a controlled burst of X-rays, which passes through the patient's chest.
3. X-ray Absorption: As the X-rays pass through the chest, they are absorbed to varying degrees by the tissues and structures within the body. Different tissues, such as lungs, bones, and soft tissues, absorb X-rays differently, resulting in varying levels of X-ray exposure on the detector.
4. Image Capture: The X-ray detector captures the X-rays that have passed through the patient's chest. The detector may be a film cassette or a digital sensor, depending on the type of X-ray machine used. Digital detectors are more commonly used nowadays due to their convenience and ability to provide immediate images.
5. Image Processing: The captured X-ray data is processed by the X-ray machine's software or computer system. The software enhances and adjusts the image to improve clarity and visibility of the lung structures.
6. Image Interpretation: The processed image is then examined by a radiologist or healthcare professional. They assess the image for any abnormalities or signs of respiratory conditions, such as lung infiltrates, consolidations, or opacities, which could be indicative of COVID-19 or other lung diseases.

7. Reporting and Diagnosis: Based on the interpretation of the chest X-ray image, a diagnostic report is generated. The report provides information about the presence or absence of lung abnormalities and any specific findings that may aid in the diagnosis of COVID-19 or other respiratory conditions.

Advantages of Chest X-Ray Machine for Detecting COVID-19

Chest X-ray machines for detecting COVID-19 has the following advantages:

1. Accessibility: Chest X-ray machines are widely available in healthcare facilities, including hospitals, clinics, and medical centers. They are more accessible compared to other imaging modalities, such as computed tomography (CT) scans, which may be limited in availability or require specialized equipment.
2. Quick Imaging: Chest X-rays provide rapid imaging results, allowing for immediate evaluation and intervention. This speed can be particularly useful in emergency situations or when timely decisions are needed, such as determining the need for hospital admission or initiating appropriate treatment.
3. Screening Tool: Chest X-rays can be used as a screening tool for identifying individuals with suspected COVID-19. They can help healthcare providers triage patients, directing them for further diagnostic testing or isolation based on the presence of lung abnormalities associated with COVID-19.
4. Monitoring Disease Progression: Serial chest X-rays can be used to monitor the progression of lung abnormalities in COVID-19 patients. Follow-up imaging helps assess the response to treatment, identify complications, and guide clinical decision-making.
5. Cost-Effectiveness: Compared to more advanced imaging techniques like CT scans, chest X-rays are generally more cost-effective. This affordability makes them a suitable initial imaging choice, especially in resource-constrained settings or when CT scanning is not readily available.

6. Radiation Dose: While chest X-rays involve exposure to ionizing radiation, the radiation dose from a single chest X-ray is relatively low. This lower radiation dose makes chest X-rays safer for patients, particularly when repeated imaging is necessary for monitoring purposes.
7. Supplementary Information: Chest X-rays can provide additional information beyond COVID-19 diagnosis. They can help identify other respiratory conditions or complications that may mimic or coexist with COVID-19, aiding in comprehensive patient evaluation and management.
8. Establishing Baseline: Chest X-rays obtained during the acute phase of COVID-19 can establish a baseline image for future comparison. These baseline images can be useful for assessing long-term lung recovery, identifying potential sequelae, or providing reference points for follow-up care.

Disadvantages of Chest X-Ray Machine for Detecting COVID-19

Chest X-ray machines can provide valuable information in the detection and evaluation of COVID-19, there are also certain limitations and disadvantages to consider.

1. Sensitivity: Chest X-rays have lower sensitivity compared to other imaging modalities like computed tomography (CT) scans in detecting early-stage lung abnormalities associated with COVID-19. This means that chest X-rays may miss subtle or early signs of the disease, leading to potential false-negative results.
2. Specificity: The radiographic findings on chest X-rays for COVID-19 can overlap with other respiratory conditions, such as bacterial pneumonia or influenza. This lack of specificity can make it challenging to differentiate COVID-19 based solely on chest X-ray images, potentially leading to false-positive or false-negative interpretations.
3. Limited Assessment of Extra-Pulmonary Manifestations: Chest X-rays primarily focus on evaluating the lung structures. They may not provide comprehensive information about potential extra-pulmonary manifestations or involvement of other organs that can occur in

COVID-19. CT scans or other imaging modalities may be better suited for assessing such manifestations.

4. Lack of Quantitative Data: Chest X-ray images provide qualitative information, allowing for the visualization of lung abnormalities. However, they do not provide quantitative data, such as the extent or severity of lung involvement. This limitation may affect the ability to precisely quantify disease progression or response to treatment.
5. Radiation Exposure: Chest X-rays involve exposure to ionizing radiation, albeit at a relatively low dose. While the risk is generally minimal, repeated or unnecessary imaging should be minimized, especially in certain populations, such as pregnant women or individuals with a higher susceptibility to radiation-related risks.
6. Inability to Differentiate Active Infection: Chest X-rays are not able to differentiate between active COVID-19 infections and post-infection lung changes or other lung pathologies. This limitation makes it important to consider clinical symptoms, history, and other diagnostic tests, such as molecular testing, for accurate diagnosis and management.
7. Operator Dependency: The accuracy and interpretation of chest X-ray images can be subjective and rely on the expertise and experience of the radiologist or healthcare professional reviewing the images. Variances in interpretation can potentially affect diagnostic accuracy and consistency.
8. Lack of Dynamic Imaging: Chest X-rays provide a static image of the lungs at a specific moment in time. They do not capture the dynamic changes that may occur in the lung tissues or the progression of the disease over time. Dynamic imaging modalities like CT scans may be more effective in capturing these changes.

(e) Computed Tomography (CT) Scan for Detecting COVID-19

Computed Tomography (CT) scans can be used as a diagnostic tool for detecting and evaluating COVID-19. CT scans provide detailed cross-sectional images of the body, allowing for a more comprehensive assessment of lung involvement.

Mode of Operation of CT scan for Detecting COVID-19

The mode of operation of CT scans for detecting COVID-19 is as follows:

1. Patient Preparation: The patient lies on a movable table that is positioned within the CT scanner. The patient may be asked to remove any metallic objects or clothing that could interfere with the imaging process.
2. Scanning Process: The CT scanner consists of an X-ray tube that rotates around the patient and a detector that captures the X-ray beams after they pass through the body. The X-ray tube emits a series of X-ray beams at different angles, generating multiple cross-sectional images or slices of the lungs.
3. Image Reconstruction: The captured X-ray data is processed by a computer to reconstruct the images of the lung structures. Sophisticated algorithms and software are used to create high-resolution, detailed images of the lungs.
4. Image Interpretation: The reconstructed CT images are interpreted by radiologists or trained healthcare professionals. They analyze the images for characteristic findings associated with COVID-19, such as ground-glass opacities, consolidations, crazy paving patterns, or other lung abnormalities. These findings can help in the diagnosis and assessment of the extent and severity of lung involvement.
5. Reporting and Diagnosis: Based on the interpretation of the CT images, a diagnostic report is generated. The report provides information about the presence, location, and characteristics of lung abnormalities associated with COVID-19. It helps guide clinical decision-making and treatment planning.

Advantages of CT Scan for Detecting COVID-19

1. High Sensitivity: CT scans are highly sensitive in detecting lung abnormalities associated with COVID-19, even in the early stages of the disease. They can detect subtle changes in lung tissue that may not be visible on chest X-rays or other imaging modalities, allowing for early detection and intervention.
2. Comprehensive Assessment: CT scans provide a comprehensive evaluation of the lungs, allowing for a detailed analysis of the extent and distribution of lung involvement. They can detect and characterize various lung abnormalities, such as ground-glass opacities, consolidations, and other patterns indicative of COVID-19.
3. Quantitative Analysis: CT scans can provide quantitative information about the volume, density, and distribution of lung abnormalities. This data can help in assessing disease severity, monitoring disease progression, and evaluating the response to treatment over time.
4. Differential Diagnosis: CT scans can help differentiate COVID-19-related lung abnormalities from other respiratory conditions. By evaluating the pattern and distribution of lung involvement, radiologists can assess the likelihood of COVID-19 and consider other possible diagnoses.
5. Early Detection: CT scans can detect lung abnormalities even before the onset of symptoms or in individuals with mild or atypical symptoms. This early detection can help identify asymptomatic or presymptomatic individuals who may be at risk of spreading the virus.
6. Triaging Tool: CT scans can aid in triaging patients, particularly in situations where molecular testing may not be readily available or where there is a high suspicion of COVID-19. They can assist healthcare providers in making timely decisions regarding isolation, hospital admission, or initiation of appropriate treatment.

7. Monitoring Disease Progression: CT scans allow for the monitoring of disease progression and response to treatment over time. Serial CT scans can track changes in lung involvement, assess treatment efficacy, and help guide clinical decision-making.

Disadvantages of CT Scan for Detecting COVID-19

CT scans offer the following advantages in detecting COVID-19:

1. Radiation Exposure: CT scans involve a higher radiation dose compared to other imaging modalities, such as chest X-rays. This exposure to ionizing radiation carries a potential risk, especially when repeated or unnecessary scans are performed. The radiation dose should be carefully considered and balanced against the diagnostic benefits.
2. Overdiagnosis and False Positives: CT scans can detect lung abnormalities that are not specific to COVID-19. Other respiratory conditions, such as viral or bacterial pneumonia, can have similar imaging findings. This can lead to overdiagnosis and false-positive results, requiring careful correlation with clinical symptoms, history, and other diagnostic tests.
3. Cost and Accessibility: CT scanning may not be universally accessible, particularly in resource-constrained settings. Additionally, CT scans can be more expensive compared to other diagnostic methods, limiting their availability and cost-effectiveness, especially in areas with limited resources.
4. Lack of Specificity: CT findings in COVID-19 can overlap with other lung diseases, making it challenging to distinguish COVID-19 solely based on CT images. Clinical correlation and consideration of other diagnostic factors are necessary for accurate diagnosis and differentiation from other respiratory conditions.
5. Inability to Differentiate Active Infection: CT scans cannot differentiate between active COVID-19 infection and post-infection lung changes or other lung pathologies. This limitation requires careful interpretation and consideration of clinical symptoms, history, and other diagnostic tests to determine the active infection status.

6. False-Negative Results: In some cases, CT scans may fail to detect lung abnormalities in individuals with COVID-19, particularly in the early stages of the disease or in individuals with mild symptoms. False-negative results can occur, leading to missed diagnoses or delayed intervention.
7. Potential for Overutilization: Due to the accessibility and detailed imaging capabilities of CT scans, there is a risk of overutilization, especially when used as a screening tool or in situations where it may not be necessary. This overutilization can lead to increased healthcare costs, radiation exposure, and potential unnecessary interventions.

3.2.2 General Challenges of all Conventional Tools for Diagnosing COVID-19

It is interesting to note that all the conventional medical laboratory tools analyzed in this research work have been instrumental in diagnosing COVID-19; however, they do have some common disadvantages which are as follows:

1. Time-consuming: Conventional laboratory techniques, such as PCR testing, require several steps, including sample collection, transportation, processing, and analysis. This process can take several hours or even days to obtain results, depending on the testing capacity and logistics. During this time, the virus can continue to spread, and timely intervention may be delayed.
2. Infrastructure and Resource Intensive: Conventional laboratory techniques typically require specialized laboratory facilities, trained personnel, and specific equipment. Establishing and maintaining these facilities can be costly and may require significant resources.
3. Sample Collection Challenges: Obtaining appropriate samples for laboratory testing can be challenging. Nasopharyngeal swabs, which are commonly used for PCR testing, require trained healthcare professionals to collect samples accurately. In some cases, patient discomfort, inadequate samples, or difficulties in reaching remote areas can affect the quality and reliability of the collected samples.

4. Turnaround Time: The time required to obtain test results can impact patient management and public health interventions. Delays in receiving results may lead to prolonged isolation, unnecessary use of resources, delayed contact tracing, and potential transmission of the virus. Rapid and timely test results are crucial for effective disease control and management.
5. Capacity Constraints: Conventional laboratory techniques may have limited testing capacity, especially during outbreaks or surges in cases. High demand for testing can overwhelm laboratories, leading to backlogs, increased turnaround times, and challenges in meeting the testing needs of the population. Scaling up testing capacity requires significant investments in infrastructure, equipment, and personnel.
6. False-Negative and False-Positive Results: Laboratory techniques, including PCR testing, have inherent limitations in sensitivity and specificity. False-negative results can occur if the viral load is low, inadequate sampling is performed, or testing is done during the early stages of infection. False-positive results can also occur due to sample contamination or cross-reactivity with similar viruses. Confirmatory testing and interpretation by skilled professionals are necessary to mitigate these issues.
7. Cost: Conventional laboratory techniques, particularly PCR testing, can be costly due to the need for specialized equipment, reagents, and skilled personnel. The costs associated with testing, sample collection, transportation, and analysis can be a significant burden, especially in resource-limited settings or for individuals without adequate healthcare coverage.
8. Dependence on Centralized Testing: Conventional laboratory techniques are often centralized, with samples collected and transported to a central laboratory for processing. This centralized approach can lead to delays in obtaining results, especially for remote or underserved areas. Additionally, transportation logistics and sample integrity during transit can pose challenges.

3.2.3 General Framework for Biomedical Machines for COVID-19 Diagnosis

The general framework for COVID-19 diagnosis involves the utilization of various biomedical machines, including PCR machines, antigen tests, X-ray machines, CT scanners, serological machines, among others. These machines play crucial roles in different stages of the diagnostic process. PCR machines are used for detecting viral genetic material, while antigen tests identify specific viral proteins. X-ray and CT-scan machines provide imaging to visualize lung abnormalities associated with COVID-19. Serological machines are employed for analyzing blood samples to detect antibodies against the virus. Each machine contributes to different aspects of COVID-19 diagnosis, facilitating accurate and timely identification of the virus in patients.

Figure 3.1 depicts the conventional biomedical instruments process flow for diagnosing COVID-19 as shown in Figures 2.5 to 2.9.

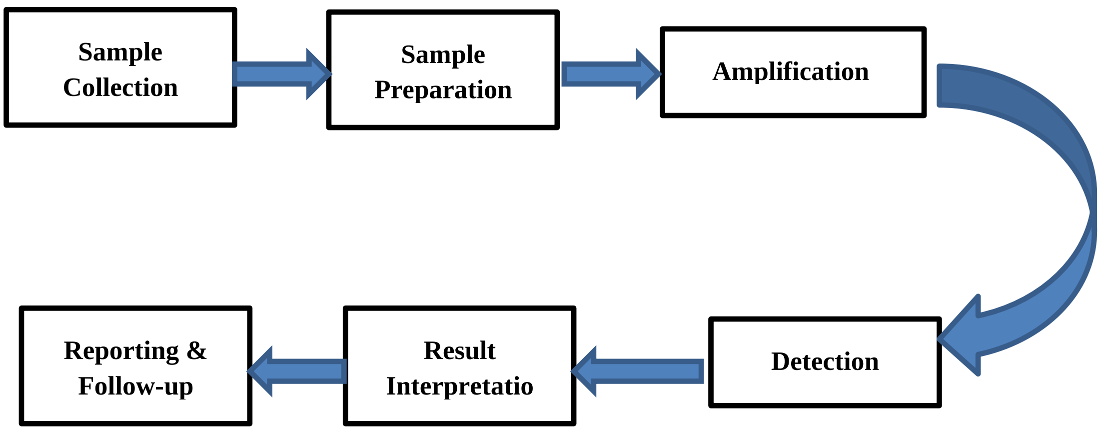

Figure 3.1: Existing Biomedical Machine Framework for COVID-19 Diagnosis

Figure 3.1 illustrates the existing biomedical machines or instruments (PCR, Antigen, etc) framework for COVID-19 diagnosis. Typically, it involves several distinct phases or stages in their operation. These phases are crucial for accurately detecting the presence of the virus in patient samples. The key phases commonly associated with these diagnostic methods framework includes:

1. Sample Collection: The process begins with the collection of patient samples, which can include nasopharyngeal swabs, oropharyngeal swabs, saliva, or other respiratory secretions. Proper collection techniques are essential to ensure the integrity of the sample and accurate test results.
2. Sample Preparation: In PCR testing, the collected samples undergo preparation to extract and isolate genetic material (RNA) from the virus. This step involves breaking open the virus particles and releasing their RNA for further analysis. Antigen tests, on the other hand, typically require less sample preparation, as they detect viral proteins directly.
3. Amplification (PCR): In PCR testing, the isolated viral RNA undergoes a process called amplification, where specific regions of the viral genome are replicated millions of times using specialized enzymes and temperature cycling. This amplification step is crucial for increasing the concentration of viral genetic material to levels detectable by the test.
4. Detection: Once the target genetic material is amplified, it is then detected using fluorescent probes or other molecular markers. In PCR testing, the detection of amplified viral RNA indicates a positive result, while the absence of detection suggests a negative result. Antigen tests detect specific viral proteins directly and provide rapid results within minutes.
5. Result Interpretation: After detection, the test results are interpreted based on predefined criteria established by regulatory authorities or healthcare organizations. Positive results indicate the presence of the virus in the patient sample, while negative results suggest the absence of detectable levels of the virus.
6. Reporting and Follow-Up: Finally, the test results are reported to healthcare providers and patients, who may require further medical evaluation or treatment based on the outcome. Positive results may trigger contact tracing efforts and public health interventions to prevent further transmission of the virus.

In essence, these phases represent essential steps in the operation of biomedical machines like PCR and antigen tests for COVID-19 detection. Each phase plays a critical role in ensuring the accuracy, reliability, and timely reporting of test results, thereby facilitating effective disease management and control efforts.

3.2.4 Analysis of Existing AI-Based Models for Diagnosing COVID-19

Deep learning models have been increasingly used for diagnosing COVID-19. These models leverage the power of artificial intelligence and machine learning to analyze medical images, such as chest X-rays or CT scans, and aid in the detection and diagnosis of COVID-19. The following are overview of deep learning models for diagnosing COVID-19:

1. Convolutional Neural Networks (CNNs): CNNs are commonly used deep learning models for image analysis. They excel at extracting features from medical images and identifying patterns associated with COVID-19. CNNs can be trained on large datasets of COVID-19 images, enabling them to learn and recognize specific visual characteristics indicative of the disease.
2. Image Classification: Deep learning models can classify medical images as COVID-19 positive or negative. By training the model on a dataset of labeled COVID-19 and non-COVID-19 images, it can learn to differentiate between different lung abnormalities and accurately classify images based on the presence or absence of COVID-19-related patterns.
3. Transfer Learning: Transfer learning is a technique where pre-trained deep learning models, initially trained on large image datasets, are fine-tuned to perform specific tasks. Deep learning models pre-trained on general image datasets can be adapted and fine-tuned using COVID-19-specific image data, allowing for more efficient training and improved performance.

4. Ensemble Models: Ensemble models combine multiple deep learning models to improve diagnostic accuracy. By aggregating the predictions from multiple models, ensemble models can enhance the robustness and reliability of COVID-19 diagnosis.

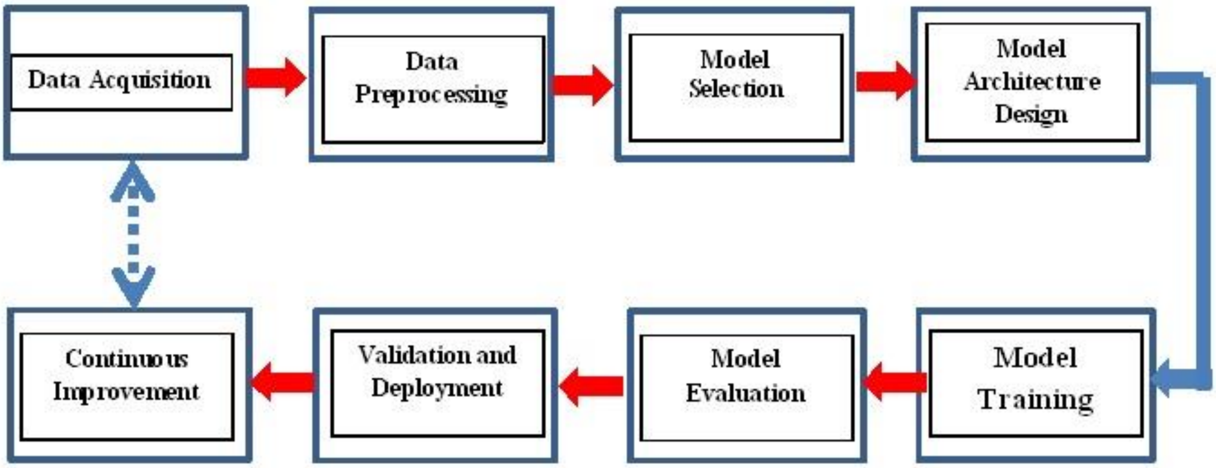

Figure 3.2: Existing deep learning framework for COVID-19 Diagnosis
(Himadri et al. 2020; Sekeroglu & Ozsahin, 2020; Kapal et al., 2021; Bashar et al. 2021**)**

Advantages of Deep Learning Models for Diagnosing COVID-19

Deep learning models offer several advantages for diagnosing COVID-19. Here are some of the advantages of using deep learning models for COVID-19 diagnosis:

1. High Accuracy: Deep learning models can achieve high accuracy in detecting COVID-19 from medical images, such as chest X-rays or CT scans. They can learn and recognize subtle patterns and features associated with COVID-19, allowing for more accurate and reliable diagnosis.
2. Rapid Analysis: Deep learning models can analyze medical images quickly, providing rapid results that can aid in timely decision-making. This is particularly valuable in situations where prompt diagnosis is crucial, such as in emergency departments or when screening large populations.
3. Scalability: Deep learning models can be easily scaled and deployed across healthcare systems, making them applicable in various clinical settings and capable of handling large

volumes of data. This scalability allows for widespread adoption and utilization of the models to assist in COVID-19 diagnosis.

4. Objectivity and Consistency: Deep learning models provide objective assessments of COVID-19 presence based on learned patterns and features. They can help minimize the potential for human bias and variability in interpretations, leading to more consistent and standardized diagnosis.
5. Automation and Efficiency: Deep learning models have the potential to automate the screening process for COVID-19. By analyzing medical images, they can assist healthcare professionals in triaging patients, reducing the burden on radiologists and allowing for more efficient use of resources.
6. Potential for Decision Support: Deep learning models can serve as decision support tools, providing additional information and insights to healthcare professionals. They can highlight suspicious regions in medical images, guide attention to potential abnormalities, and aid in treatment planning and monitoring.
7. Adaptability and Generalizability: Deep learning models can be trained on diverse datasets from different regions, allowing them to adapt and generalize to various populations and imaging practices. This flexibility enhances their applicability in different healthcare settings and improves their ability to detect COVID-19 cases across different demographics.

Disadvantages of Deep Learning Models for Diagnosing COVID-19

The deep learning models for diagnosing COVID-19 offers the following disadvantages:

Data Availability and Quality: Deep learning models rely on large and diverse datasets for training. However, obtaining well-curated and labeled datasets for COVID-19 can be challenging, particularly in the early stages of a pandemic. Limited availability and quality of data can affect the performance and generalizability of the models.

1. Interpretability and Explainability: Deep learning models are often considered as black boxes, making it difficult to understand the reasoning behind their decisions. Interpretability and explainability of model predictions are crucial in healthcare to gain trust, validate decisions, and ensure accountability.
2. Overfitting and Generalization: Deep learning models can be susceptible to overfitting, where they perform well on the training data but fail to generalize to new, unseen data. Generalization is especially critical in the context of COVID-19, where models need to accurately diagnose cases from different populations, imaging devices, and data distributions.
3. Ethical and Bias Concerns: Deep learning models can inherit biases present in the training data, leading to biased predictions. Bias in COVID-19 diagnosis can disproportionately affect certain populations, exacerbate healthcare disparities, and contribute to unequal access to accurate diagnosis and treatment.
4. Validation and Clinical Integration: Deep learning models need rigorous validation in real-world clinical settings to assess their performance, reliability, and impact on patient outcomes. Integration of these models into clinical workflows requires careful consideration, validation, and collaboration with healthcare professionals to ensure their effective and safe use.
5. Lack of Standardization: The lack of standardized protocols for image acquisition, annotation, and reporting can introduce variability and inconsistency in the data used for training and evaluation of deep learning models. This can affect the performance and generalizability of the models across different healthcare settings.
6. Resource Intensiveness: Deep learning models require significant computational resources, including powerful hardware and large-scale computing infrastructure, for training and inference. This can limit their accessibility and implementation in resource-constrained healthcare environments.

7. Regulatory and Legal Challenges: The deployment and use of deep learning models for COVID-19 diagnosis may raise regulatory and legal challenges. Compliance with data privacy regulations, ethical considerations, and liability concerns are important aspects that need to be addressed.

3.3 The proposed System

The proposed system utilizes multiple types of datasets, such as medical imaging (chest X-rays, CT scans, and Blood smear) and clinical information (patient demographics, symptoms), to improve the proposed model's performance.

One of the key focuses of this system is the incorporation of explainability into the deep learning model. This means that the model is designed to provide insights into its decision-making process, allowing users to understand how and why it makes specific predictions. Explainability is crucial in the medical field as it helps build trust, validate the model's predictions, and support clinical decision-making.

The system adopts a multi-modal learning approach, which means it integrates and combines information from different data modalities. By leveraging various data sources, the model can benefit from the strengths and complementary nature of each modality, leading to more accurate and robust predictions.

In addition to prediction, the proposed system also emphasizes the interpretation of the model's predictions. This means that it not only aims to predict COVID-19 but also provides explanations or insights into the factors that contribute to those predictions. Interpretability enables clinicians and researchers to understand the underlying factors influencing the model's decisions and identify important features or patterns related to COVID-19 diagnosis or severity.

The system is based on deep learning techniques, which involve the use of neural network architectures like convolutional neural networks (CNNs) and some of its variants like ResNet50, InceptionV3 and GRAD-CAM for explainability and interpretability. These architectures are capable of analyzing the input data in order to extract complex patterns and features from the large-scale datasets, which is especially advantageous in medical imaging tasks, including COVID-19 diagnosis.

To fully comprehend the details of the proposed system, it would be necessary to read through subsequent sections which provide documentation and information about the proposed system's architecture, methodologies, datasets used, and evaluation metrics.

3.4 Data Collection

Data collection is a crucial step in this research process. It involves gathering of different datasets with respect to COVID-19 for analysis and interpretation. The effectiveness of data collection directly influences the validity and reliability of research findings.

3.4.1 Types, Sources and Description of Dataset Utilized

This research exclusively employed secondary source data obtained from an online research community platform known as kaggle.com. The COVID-19 datasets encompass various types, as outlined and described below.

(a) Chest X-ray Images

The COVID-19 CXR images dataset comprises chest X-ray (CXR) images specifically associated with cases of COVID-19. Chest X-rays are diagnostic imaging tools used to visualize the structures within the chest, including the lungs and surrounding tissues. This dataset provides a collection of radiographic images that can aid in the identification and analysis of pulmonary abnormalities indicative of COVID-19 infection.

Key details about the COVID-19 CXR images dataset may include information on image resolution, any annotations or labels indicating areas of interest such as lung abnormalities, anonymized patient details.

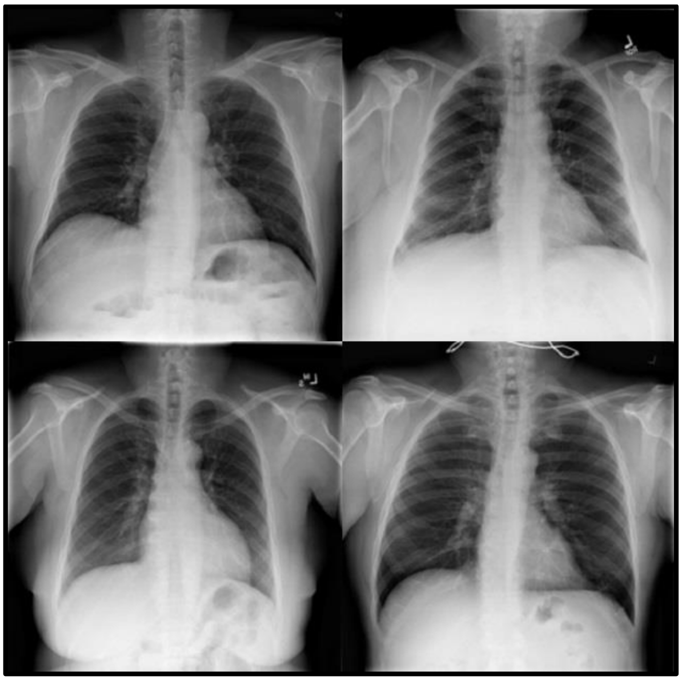

Figure 3.3(a): Sample Normal Chest X-ray images (Kaggle.com)

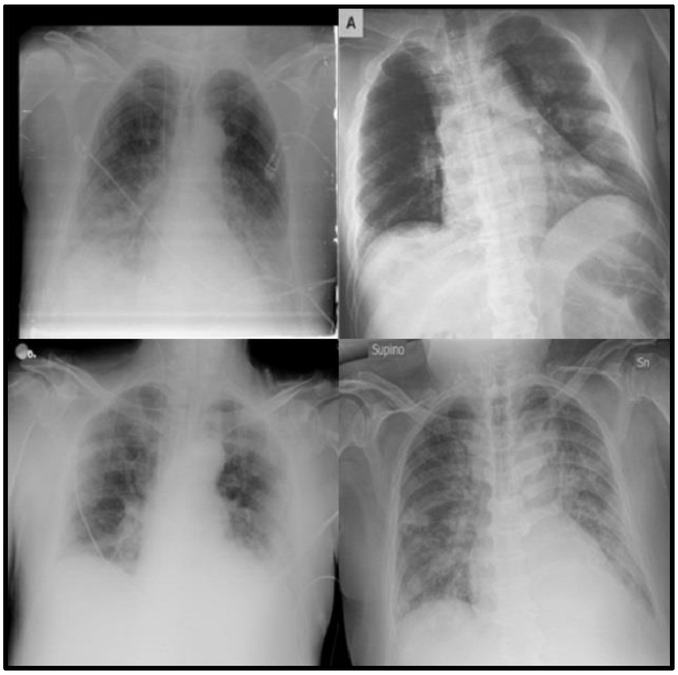

Figure 3.3(b): Sample Positive COVID-19 Chest X-ray images (Kaggle.com)

Table 3.1: Sources and Description of COVID-19 Chest X-Ray Dataset

Dataset name	COVID-19 Radiography Database
Brief description	An open-access dataset with chest X-ray images of COVID-19 positive and negative cases.
File Size	816MB
Source	https://www.kaggle.com/tawsifurrahman/covid19-radiography-database
Normal images	10,192 data points
Positive cases images	3,616 data points
Remark	Imbalanced dataset

(b) COVID-19 CT Scan Images

The COVID-19 CT scan images dataset is a compilation of medical images obtained through computed tomography scans, specifically focusing on cases related to COVID-19. These images provide detailed insights into the internal structures of the body, particularly the lungs, making them valuable for assessing respiratory conditions. The dataset includes anonymized CT scan images, potentially accompanied by annotations indicating regions affected by COVID-19.

Figure 3.4(a): Sample Non-COVID-19 CT-Scan images (Kaggle.com)

Figure 3.4(b): Sample COVID-19 CT-Scan images (Kaggle.com)

Table 3.2: CT-Scan Dataset

Dataset name	SARS-CoV-2 CT-Scan Dataset
Brief description	A collection of CT scans for COVID-19 positive and negative cases
File Size	242MB
Source	https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset
Normal images	1,229 data points
Positive cases images	1,252 data points
Remark	Imbalanced dataset

(c) Blood Smear Images

The blood smear images dataset encompasses a collection of microscopic images depicting blood smears. These images are specifically relevant to the analysis of blood samples for various medical purposes, including the identification of abnormalities or diseases such as COVID-19. Blood smear images provide detailed visual information about the composition and morphology of blood cells, allowing for the examination of red blood cells, white blood cells, and platelets.

In the context of this COVID-19 research work, blood smear images is been utilized to investigate potential hematological manifestations or abnormalities associated with the virus.

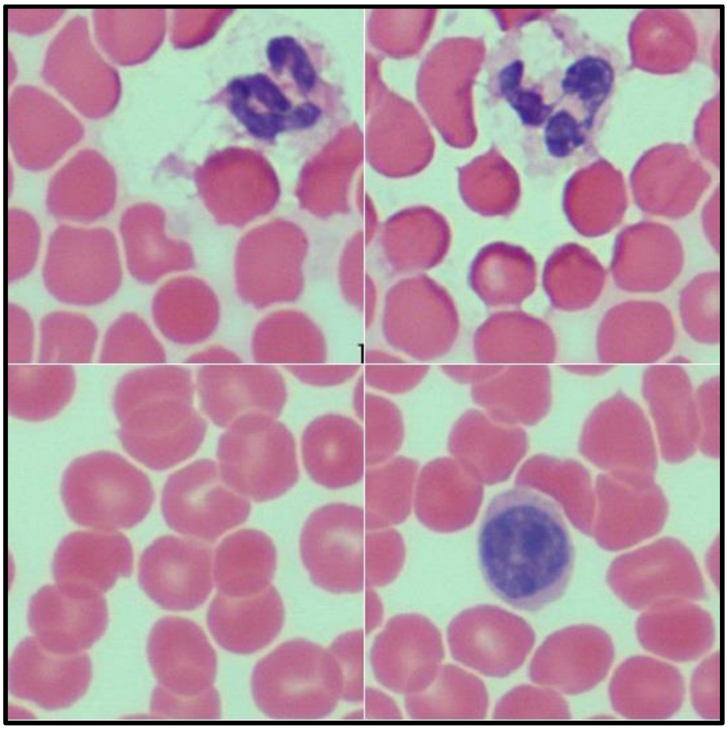

Figure 3.5: Sample Positive COVID-19 Blood smear images (Kaggle.com)

Table 3.3: Blood Smear Image Dataset

Dataset name	SARS-CoV-2 CT-Scan Dataset
Brief description	A collection of Blood smear COVID-19 positive and negative cases
File Size	112MB
Source	https://www.kaggle.com
Normal images	1,250 data points
Positive cases images	1,400 data points
Remark	Imbalanced dataset

(d) Clinical Data

The clinical data dataset encompasses a compilation of information related to patients' clinical characteristics, medical histories, and health-related details. In the context of this COVID-19 research work, this dataset include a diverse range of clinical information such as symptoms, laboratory test results, comorbidities, treatment regimens, and outcomes associated with both COVID-19 positive and healthy cases.

Table 3.4: COVID-19 Clinical dataset description

Dataset name	COVID-19 Clinical data
Brief description	Dataset with COVID-19 patient's symptoms, status, and medical history
File Size	5MB
Source	https://www.kaggle.com/datasets/meirnizri/covid19-dataset
Features	21
Negative cases	8,765
Positive cases	18,660
Remark	Imbalanced dataset

Table 3.5: Sample COVID-19 clinical dataset

Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	Headache	Heart Disease	Diabetes	Fatigue	Gastrointestinal	Abroad travel	Wearing Masks
0	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	No	No
1	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	No	No
2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No
3	Yes	Yes	Yes	No	No	Yes	No	No	Yes	No	No	Yes	No
4	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	No

Table 3.6: COVID-19 possible signs/symptoms

S/N	Signs/Symptoms
1	Breathing Problem
2	Fever
3	Dry Cough
4	Sore throat
5	Running Nose
6	Asthma
7	Chronic Lung Disease
8	Headache
9	Heart Disease
10	Diabetes
11	Hyper Tension
12	Fatigue
13	Gastrointestinal

3.4.2 Data Preprocessing

Data preprocessing played a pivotal role in the analysis of both image and numerical datasets collected. This essential step involved a comprehensive pipeline that included cleaning, transforming, and preparing the data (comprising both images and numerical information) to a format conducive for analysis and model training. The primary objective was to improve the data quality and ensure its appropriateness for COVID-19 classification. The implemented data preprocessing encompassed various crucial steps, including the following.

1. Handling Missing Values

The process of removing missing or null values from a COVID-19 clinical dataset is a pivotal aspect of data preprocessing, aiming to enhance data quality and reliability. In this critical step, Python code employing the pandas library is utilized to first assess the extent of missing data by displaying the count of null values for each column. Subsequently, rows containing missing values are systematically removed using the `dropna()` method. This meticulous approach ensures that incomplete or unreliable records are excluded, preventing potential biases in subsequent analyses or machine learning models. The cleaned dataset, now free of missing values, is then saved to a new file, marking a crucial stage in preparing the clinical data for more robust and accurate insights into COVID-19 trends and patient characteristics.

The Python code not only serves as a practical tool for data cleaning but also emphasizes the importance of transparency in handling missing values. By providing a clear overview of the dataset both before and after removal, researchers can make informed decisions about the impact of missing data on the study's objectives. This process contributes to the integrity and validity of clinical analyses, ensuring that the dataset used for COVID-19 research is both comprehensive and reliable.

```
import pandas as pd

# Specify the path to the clinical dataset
dataset_path = "../OYE/COVID-19DATASETS/CLINICAL DATA"

# Load the dataset
clinical_data = pd.read_csv(dataset_path)

# Display information about missing values before removal
print("Before removing missing values:")
print(clinical_data.isnull().sum())

# Remove rows with missing values
clinical_data_cleaned = clinical_data.dropna()

# Display information about missing values after removal
print("\nAfter removing missing values:")
print(clinical_data_cleaned.isnull().sum())

# Save the cleaned dataset to a new file
cleaned_dataset_path = "../OYE/COVID-19DATASETS/CLINICAL DATA
/clinical_data_cleaned.csv"
clinical_data_cleaned.to_csv(cleaned_dataset_path, index=False)

print(f"\nCleaned dataset saved to: {cleaned_dataset_path}")
```

Figure 3.6: Python code segment for removing missing and null value from clinical dataset

Figure 3.6 depicts a python code segment for analyzing missing or null values within the clinical dataset, and it is a crucial step in ensuring the integrity of COVID-19 clinical information. Initially comprising 27,425 data points, the dataset underwent scrutiny to identify instances of missing or null values. The analysis revealed that 125 data points were affected by these missing values, prompting a strategic approach to address and rectify this issue. The subsequent removal or imputation of these missing values led to a refined dataset, now consisting of 27,300 data points. This process ensures that the dataset used for clinical analyses is more complete, minimizing the potential for biases or inaccuracies in the interpretation of COVID-19 trends and patient-related insights.

The Python code executed for this analysis, likely utilizing the pandas library, played a pivotal role in systematically handling the missing or null values. It is essential to note that the decision to remove or impute missing values should be made with careful consideration of the nature of the data and the goals of the analysis. This transparent and methodical approach not only enhances the reliability of clinical datasets but also contributes to the robustness of subsequent research endeavors, providing a clearer and more accurate understanding of the clinical aspects of COVID-19.

Table 3.7: Analysis of missing or null value from clinical information dataset

Dataset	Clinical data
Total no of data points (+ve and –ve cases)	27,425
Total no of missing or null values (+ve and –ve cases)	125
Remaining data points	27,300

2. Resolving Image Quality Issues

Examination of the presence of blurred images across the three collected image datasets was done. Failure to address this issue could adversely affect the accuracy and reliability of the analysis and predictions.

Table 3.8 presents a detailed account of how image quality issues, specifically blurred images, were addressed across three distinct datasets: Chest X-ray, CT-Scan, and Blood Smear. The dataset sizes before the intervention are outlined, indicating a substantial collection effort with 13,808 Chest X-ray images, 2,481 CT-Scan images, and 2,650 Blood Smear images. However, the presence of blurred images posed a challenge, with 450, 25, and 37 instances identified in the respective datasets. To ensure the integrity of subsequent analyses and machine learning model training, these blurred images were meticulously removed.

Upon removing the blurred images, the "Balance" row in the table illustrates the refined dataset sizes, now devoid of the identified image quality issues. The adjusted counts reveal 13,358 Chest X-ray images, 2,456 CT-Scan images, and 2,613 Blood Smear images, showcasing a more pristine dataset for in-depth exploration. This meticulous handling of image quality challenges underscores the importance of pre-processing steps to enhance the reliability and effectiveness of subsequent analyses, particularly in the context of medical imaging datasets.

Table 3.8: Handling image quality issues

Dataset	Chest X-ray	CT-Scan	Blood Smear
Total collected	13,808	2,481	2,650
Blurred Images	450	25	37
Balance	13,358	2,456	2,613

3. Removing Duplicates from Image and Clinical Datasets

The elimination of duplicate images holds paramount importance in this research endeavor, serving as a crucial measure to maintain data integrity and mitigate biases during analyses or model training. In the study, perceptual hashing was chosen as the technique for generating hash codes, effectively capturing visual similarities. The robustness of this method against minor variations in images is noteworthy. Although various approaches exist to identify and eliminate duplicate images, including Checksums or Hashing, Image Similarity Measures, Feature Extraction with CNNs, Duplicate Image Detection Libraries, Image Matching Algorithms, Histogram Comparison, Machine Learning Classification, Database Indexing, and Visual Inspection, our emphasis on perceptual hashing aligns with its effectiveness in addressing subtle variations within images.

```python
from PIL import Image
import imagehash
import os
from tqdm import tqdm

def remove_duplicates(input_folder):
    # Dictionary to store hash values and corresponding file paths
    hash_dict = {}

    # Iterate through images in the input folder
    for filename in tqdm(os.listdir(input_folder), desc='Processing images'):
        file_path = os.path.join(input_folder, filename)

        # Skip non-image files
        if not filename.lower().endswith(('.png', '.jpg', '.jpeg', '.gif')):
            continue

        # Open the image and compute its hash
        img = Image.open(file_path)
        img_hash = imagehash.average_hash(img)

        # Check if the hash already exists in the dictionary (duplicate)
        if img_hash in hash_dict:
            print(f"Removing duplicate: {filename}")
            # Remove the duplicate image
            os.remove(file_path)
        else:
            # Add the hash and file path to the dictionary
            hash_dict[img_hash] = file_path

if __name__ == "__main__":
    # Specify the image path
    image_path = "/Users/OYE/COVID-19DATASETS/BLOODSMEAR"

    # Remove duplicates using perceptual hashing
    remove_duplicates(image_path)
```

Figure 3.7: Python code segment utilizing the imagehash library to detect and eliminate duplicate images leveraging on perceptual hashing.

Figure 3.7 illustrates a python code segment for analyzing duplicate images within three distinct datasets: Chest X-ray, CT-Scan, and Blood Smear. The dataset sizes are initially detailed, with 13,358 data points for Chest X-ray, 2,456 for CT-Scan, and 2,613 for Blood Smear. However, the presence of duplicate images posed a significant consideration, with 3,500, 320, and 370 duplicates identified in the respective datasets. This analysis serves as a critical step in ensuring the accuracy and reliability of subsequent analyses and model training by addressing data redundancy.

Upon removal of the duplicate images, the table highlights the refined dataset sizes, now consisting of 9,858 Chest X-ray images, 2,136 CT-Scan images, and 2,243 Blood Smear images. The process of meticulously handling duplicate images is paramount in producing a dataset that is free from redundancies, enabling a more focused and effective exploration of the unique information embedded in each image. This systematic approach not only ensures data integrity but also contributes to the robustness of the subsequent analyses, affirming the importance of thorough data preprocessing in the context of medical image datasets.

Table 3.9: Analysis of duplicate images dataset

Dataset	Chest X-ray	CT-Scan	Blood Smear
No of data points	13,358	2,456	2,613
Duplicate Images	3,500	320	370
Remaining images	9,858	2,136	2,243

Table 3.10 presents a detailed analysis of duplicate clinical information within the dataset. The dataset initially comprises 27,300 data points related to clinical information. During the analysis, 54 instances of duplicate entries were identified and subsequently removed. This meticulous examination and elimination of duplicate clinical data aim to enhance the accuracy and reliability of the dataset, ensuring that subsequent analyses and interpretations are based on a unique and non-redundant set of information.

Following the removal of the identified duplicate entries, the table highlights the refined dataset size, now containing 27,246 remaining data points. This process underscores the importance of maintaining data integrity and eliminating redundancies in clinical information. The resulting dataset provides a more focused and reliable foundation for in-depth investigations and model training in the context of medical research. By systematically addressing duplicate clinical data, we can enhance the robustness of the analyses and draw more accurate conclusions from the available information.

Table 3.10: Analysis of duplicate clinical information dataset

Dataset	**Clinical data**
No of data points	27,300
Duplicate	54
Remaining data	27,246

4. Handling Class Imbalances

Addressing class imbalance is a critical aspect in the context of both image and numerical datasets, aiming to prevent biased model outcomes that might struggle in accurately predicting the minority class. Several techniques are commonly employed to tackle this issue in both types of datasets.

In image datasets, strategies include Data Augmentation, which involves generating additional training samples for the minority class through transformations like rotation and scaling. Class Weighting assigns higher weights to the minority class during training to address the imbalance. Transfer Learning, utilizing pre-trained models like ResNet or EfficientNet, has proven highly effective. Generative Adversarial Networks (GANs) can generate synthetic samples for the minority class. Ensemble Methods involve combining predictions from multiple models to improve overall performance.

For COVID-19 image datasets, characterized by complexity and uniqueness in Chest X-ray, CT-Scan, and Blood Smear images, a combination of techniques is often recommended. The most effective approach involves Transfer Learning, specifically leveraging pre-trained models like ResNet or EfficientNet, which have demonstrated high efficacy in image datasets.

In numerical datasets, Resampling Techniques, such as oversampling the minority class or undersampling the majority class, are common strategies. Synthetic Data Generation, utilizing techniques like SMOTE, helps create synthetic samples for the minority class. Ensemble Methods and Cost-Sensitive Learning, involving the adjustment of misclassification costs during training, are effective strategies. Evaluating model performance using different metrics like precision, recall, and F1-score ensures a comprehensive assessment.

For clinical datasets, which often involve numerical information, a combination of resampling techniques, such as oversampling the minority class, and adjusting misclassification costs during model training proves effective in handling class imbalance. This multifaceted approach ensures that the model is robust in its predictions and provides a balanced assessment of performance.

Table 3.11 provides a comprehensive analysis of an initially imbalanced clinical dataset, presenting counts for positive and negative cases. In the imbalanced dataset, there are 18,515 instances of positive cases compared to 8,731 instances of negative cases. This significant imbalance could potentially lead to biased model predictions, particularly favoring the majority class. Recognizing the importance of mitigating class imbalance, the table introduces a balanced dataset achieved through the Synthetic Minority Over-sampling Technique (SMOTE). Applying SMOTE to the original dataset results in a balanced distribution, with both positive and negative cases now totaling 18,515 instances each. This transformation is crucial for ensuring fair and accurate model training, particularly in scenarios where the minority class (negative cases in this context) is underrepresented.

The use of SMOTE in achieving a balanced dataset is a pivotal step in handling class imbalance issues. By generating synthetic samples for the minority class, SMOTE equalizes the representation of positive and negative cases. This transformation is particularly beneficial when preparing the dataset for machine learning models, as it prevents models from being biased towards the majority class. The balanced dataset becomes a more reliable foundation for training models that can make accurate predictions across both positive and negative cases, contributing to the robustness of analyses in the clinical context.

Table 3.11: Analysis of Imbalanced/balanced clinical dataset

	Positive cases	**Negative cases**
Imbalanced dataset	18,515	8,731
Balanced dataset using SMOTE Technique	18,515	18,515

```
import pandas as pd
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split

# Create a DataFrame with the given dataset
data = pd.DataFrame({
   'Class': ['Positive'] * 18515 + ['Negative'] * 8731
})

# Separate features and target variable
X = pd.get_dummies(data.drop('Class', axis=1))
y = data['Class']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Apply SMOTE to handle class imbalance
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_train, y_train)

# Print the count of classes after applying SMOTE
print("Class distribution after SMOTE:")
print(pd.Series(y_resampled).value_counts())
```

Figure 3.8: Python code segment for handling class imbalance

Figure 3.8 depicts the python code segment using imbalanced-learn library for SMOTE used to handle class imbalance in the clinical COVID-19 dataset with positive and negative cases. We used the Synthetic Minority Over-sampling Technique (SMOTE) to generate synthetic samples for the minority class.

5. Train-Validation-Test Split Strategy

The training-validation-test split strategy is a fundamental practice in machine learning to evaluate and optimize model performance effectively. This strategy involves dividing a dataset into three distinct subsets: the training set, the validation set, and the test set. The training set is the largest portion and is used to train the model. It serves as the foundation for the model to learn patterns and relationships within the data. The validation set is employed during the training process to assess the model's performance on unseen data, allowing for the fine-tuning of hyperparameters and preventing overfitting. Lastly, the test set is kept entirely separate and untouched until the model is fully trained and validated. It serves as a final evaluation to assess the model's generalization to completely unseen data, providing a reliable measure of its performance on real-world scenarios. This split strategy ensures that the model's effectiveness is tested thoroughly on different subsets of the data, promoting a more accurate assessment of its capabilities.

A well-designed training-validation-test split is crucial for preventing model overfitting, where the model performs well on the training data but fails to generalize to new, unseen data. The validation set acts as a safeguard against overfitting by providing an unbiased assessment of the model's performance during training. This split strategy is particularly important when dealing with limited datasets, as it helps to make the most efficient use of the available data for training, validation, and testing. Striking the right balance in the sizes of these subsets is key: a sufficient training set ensures the model learns effectively, an adequate validation set facilitates proper tuning, and a reliable test set guarantees a robust evaluation of the model's real-world performance.

Table 3.12 outlines the final integration of multiple data modalities in the comprehensive dataset, presenting the number of data points associated with each modality. The Chest X-ray dataset contributes 9,858 data points, the CT-Scan dataset provides 2,136 data points, and the Blood Smear dataset offers 2,243 data points. Complementing these imaging modalities, the Clinical data dataset incorporates a substantial 37,030 data points. This aggregation of diverse data modalities allows for a holistic and multi-faceted analysis of COVID-19, encompassing both visual and numerical aspects.

The inclusion of clinical data further enriches the dataset, capturing a comprehensive spectrum of information that spans imaging features and clinical parameters.

The significance of this unified dataset lies in its potential to enhance the understanding of COVID-19 through a multi-modal approach. By combining information from Chest X-rays, CT-Scans, Blood Smear images, and clinical data, researchers gain a more nuanced perspective on the disease, enabling them to explore correlations and patterns that may not be apparent when analyzing each modality in isolation. This integrated dataset serves as a robust foundation for developing machine learning models capable of leveraging diverse data sources to predict, interpret, and understand COVID-19 outcomes more comprehensively.

Table 3.12: Final Multiple Data Modalities Utilized

Dataset	**Chest X-ray**	**CT-Scan**	**Blood Smear**	**Clinical Data**
No of data points	9,858	2,136	2,243	37,030

Table 3.13 delineates the split strategy employed for the Chest X-ray dataset, allocating data points into training, validation, and testing subsets. The training set encompasses 80% of the data, comprising 7,886 Chest X-ray images, providing the foundational material for training machine learning models. The validation set, constituting 10% of the dataset with 985 images, serves as a critical checkpoint during model development, aiding in fine-tuning and hyperparameter optimization. The testing set, mirroring the validation set with 10% or 985 images, remains entirely untouched until the model is fully trained and validated. This split strategy ensures a balanced distribution, facilitating robust model evaluation and preventing overfitting by assessing generalization performance on unseen data.

Table 3.13: Split strategy for Chest X-ray dataset

Training (80%)	Validation (10%)	Testing (10%)
7886	985	985

Table 3.14 delineates the split strategy applied to the CT-Scan dataset, strategically distributing data points across training, validation, and testing subsets. The training set comprises 80% of the dataset, encompassing 1,709 CT-Scan images, providing a substantial volume for the model to learn patterns and features. The validation set, constituting 10% with 213 images, plays a pivotal role in fine-tuning the model and preventing overfitting during the training process. Similarly, the testing set, mirroring the validation set with 10% or 213 images, remains untouched until the model is fully trained and validated, serving as an independent benchmark to evaluate the model's performance on completely unseen CT-Scan data. This split strategy ensures a balanced and systematic approach to model development, validation, and evaluation for the CT-Scan dataset.

Table 3.14: Split strategy for CT-Scan dataset

Training (80%)	Validation (10%)	Testing (10%)
1,709	213	213

Table 3.15 provides a clear insight into the split strategy implemented for the Blood Smear dataset, delineating the distribution of data points across training, validation, and testing subsets. The training set, comprising 80% of the dataset with 1,795 Blood Smear images, forms the cornerstone for training machine learning models, allowing them to grasp intricate patterns inherent in the data. The validation set, constituting 10% or 224 images, is strategically employed during the training process to fine-tune model hyperparameters and prevent overfitting. Similarly, the testing set, mirroring the validation set with 10% or 224 images, is reserved until the model undergoes comprehensive training and validation. This systematic split strategy ensures a balanced distribution, fostering robust model evaluation and affirming the model's capacity to generalize effectively to unseen Blood Smear images, enhancing the reliability of predictions in real-world applications.

Table 3.15: Split strategy for blood smear dataset

Training (80%)	Validation (10%)	Testing (10%)
1,795	224	224

Table 3.16 elucidates the split strategy applied to the Clinical Information dataset, illustrating the distribution of data points across training, validation, and testing subsets. The training set, constituting 80% of the dataset with 29,624 instances, plays a foundational role in training machine learning models on the clinical parameters. The validation set, comprising 10% or 3,703 instances, is strategically employed during the model development process for fine-tuning and hyperparameter optimization. The testing set, mirroring the validation set with 10% or 3,703 instances, remains untouched until the model undergoes comprehensive training and validation. This meticulous split strategy ensures a balanced distribution, facilitating rigorous model evaluation and affirming the model's ability to generalize effectively to new clinical data, ultimately contributing to the robustness of the model in real-world applications.

Table 3.16: Split strategy for clinical information dataset

Training (80%)	Validation (10%)	Testing (10%)
29,624	3,703	3,703

3.5 Model Architecture Design

In this section, the components, capabilities, and structure of the proposed system are defined to align with the goals and objectives. This involves determining the design considerations of the architecture, system architectural framework, components design, identifying data structures, creating user interfaces, defining system behavior and logic, considering performance and scalability, and ensuring security and reliability. These steps are to ensure that the proposed system is well-designed and capable of meeting its intended purpose.

3.5.1 Mathematical Notation Formulation for the Proposed Design

In this section, we aim to formulate a mathematical notation for our proposed integrated deep learning models, designed to concurrently execute within a unified framework for precise classification and comprehensive interpretation of the classification outcomes. Our proposed system comprises four distinct models or algorithms: ResNet50, InceptionV3, CNN, and Random Forest classifier. Each model serves a unique role in performing classification tasks using disparate datasets encompassing Chest X-ray images, CT-Scan images, Blood Smear images, and Clinical information datasets.

Initially, we incorporate ResNet50 to handle the classification of COVID-19 Chest X-ray image data. To elucidate the reasoning behind its classifications, we integrate the GRAD-CAM technique, facilitating the interpretation and explanation of the classification results. Subsequently, we introduce the InceptionV3 model, tailored specifically for classifying COVID-19 CT-Scan image data. Similarly, GRAD-CAM is employed to facilitate the interpretation and explanation of the classification outcomes derived from this model. Moreover, the CNN model is enlisted to manage the classification of COVID-19 Blood Smear Image data, with GRAD-CAM incorporated to ensure comprehensive interpretation and explanation of its classification results.

Furthermore, the Random Forest classifier is enlisted to predict outcomes based on the COVID-19 Clinical data dataset. Each model within this unified platform is endowed with distinct fine-tuning parameters, ensuring their efficacy and adaptability to the respective datasets and classification tasks at hand. This comprehensive approach enables the concurrent execution of multiple deep learning models, each tailored to a specific dataset and classification task, thereby enhancing the system's ability to accurately classify diverse data sources while providing comprehensive insights into the rationale behind each classification decision.

The Unified Platform is expressed mathematically as follows.

Let:

X_CXR = Chest X-ray images

X_CT = CT-Scan images

X_BS = Blood smear images

X_CL = Clinical data

Models:

ResNet50_CXR(θ_CXR): ResNet50 for CXR classification

InceptionV3_CT(θ_CT): InceptionV3 for CT-Scan classification

CNN_BS(θ_BS): CNN for blood smear classification

RF_CL(θ_CL): Random Forest for clinical data prediction

GRAD-CAM:

L_CXR: Last convolutional layer of ResNet50_CXR

L_CT: Last convolutional layer of InceptionV3_CT

L_BS: Last convolutional layer of CNN_BS

Unified Platform (P):

P = {ResNet50_CXR(θ_CXR), InceptionV3_CT(θ_CT), CNN_BS(θ_BS), RF_CL(θ_CL)}

Concurrent Execution:

Y_CXR = ResNet50_CXR(X_CXR; θ_CXR)

Y_CT = InceptionV3_CT(X_CT; θ_CT)

Y_BS = CNN_BS(X_BS; θ_BS)

Y_CL = RF_CL(X_CL; θ_CL)

Fine-Tuning Parameters:

θ_CXR = {W_CXR, b_CXR}

θ_CT = {W_CT, b_CT}

θ_BS = {W_BS, b_BS}

θ_CL = {tree structures, splitting criteria}

Interpretation with GRAD-CAM:

Grad-CAM(L_CXR, Y_CXR) -> heatmap_CXR

Grad-CAM(L_CT, Y_CT) -> heatmap_CT

Grad-CAM(L_BS, Y_BS) -> heatmap_BS

It should be noted that these notations delineates the structural framework of each model and their respective fine-tuning parameters. The intricate mathematical formulations detailing the architecture, loss functions, and optimization algorithms for each model are expansive and contingent upon specific implementations, elaborated upon in subsequent sections of this research endeavor. Additionally, the detailed implementation specifics of GRAD-CAM are expounded upon in subsequent segments of the research work, providing a comprehensive understanding of its integration within the proposed unified platform.

The formulations encapsulate the essence of each model's design and parameterization, setting the groundwork for their integration into a unified platform. While the precise mathematical intricacies pertaining to architecture, loss functions, and optimization strategies are deferred to subsequent sections, these notations afford a clear delineation of the system's components and their individual roles within the overarching framework. This approach lays the foundation for comprehensive exploration and analysis of the proposed deep learning-based system's capabilities and performance in subsequent sections of the research.

3.5.2 Proposed Model Architectural Design Considerations

The architectural considerations for a deep learning model encompass the model's name, its types of layers, and various parameters like kernel sizes, filter sizes, activation functions, and other design choices. These considerations play a crucial role in tailoring the model for specific tasks such as classifying COVID-19 chest X-rays, COVID-19 CT scans, COVID-19 Blood smear and COVID-19 Clinical information.

1. ResNet50 Model

(a) Design consideration for ResNet-50 in Classification of COVID-19 Chest X-Ray Image Dataset

Table 3.17 presents a comprehensive outline of the model settings for a ResNet50 model tailored specifically for the classification of COVID-19 Chest X-ray images. These design considerations encompass various aspects crucial for constructing an effective and accurate classification model.

The model architecture is defined as ResNet50, a deep residual network renowned for its ability to effectively handle image classification tasks. With 50 layers, ResNet50 employs skip connections to mitigate the vanishing gradient problem, facilitating the training of deeper networks. The input shape is specified as (224, 224, 3), indicating RGB images resized to 150x150 pixels to conform to the requirements of the ResNet50 architecture. Pre-processing involves data normalization to rescale pixel values to the range [0,1], ensuring uniformity and stability during training.

The model is initialized with pre-trained weights from the 'imagenet' dataset, leveraging transfer learning to benefit from the knowledge learned by the model on a large-scale image classification task. Fine-tuning of the pre-trained ResNet50 layers is optionally performed to adapt the model to the specifics of the COVID-19 Chest X-ray dataset. Data augmentation techniques such as random transformations (rotation, width and height shifts, horizontal flipping) are applied to increase the diversity of the training data and improve the model's generalization capability.

Other key considerations highlighted in the table include batch normalization to speed up training and improve stability, dropout regularization to prevent overfitting, selection of the loss function (Binary Crossentropy) suitable for binary classification tasks, usage of the Adam optimizer with a learning rate of 0.0001, evaluation metrics including accuracy, precision, recall, and F1-score, batch size of 64, number of epochs set to 20, early stopping strategy to monitor validation loss and halt training if no improvement is observed, and class weighting adjustment to handle class imbalance if necessary. These considerations collectively contribute to the successful development and training of a robust ResNet50 model for COVID-19 Chest X-ray image classification.

Table 3.17: ResNet50 Model settings for COVID-19 Chest X-Ray Image Dataset classification

S/N	Design Consideration	Details
1	Model Architecture	ResNet50 - A deep residual network with 50 layers
2	Input Shape	(224, 224, 3) - RGB images resized to 150x150 pixels
3	Pre-processing	Data normalization to rescale pixel values to the range [0,1]
4	Pre-trained Weights	'imagenet' - Initialize with weights pre-trained on ImageNet
5	Fine-tuning	Fine-tune the pre-trained ResNet50 layers or freeze all layers
6	Data Augmentation	Apply random transformations such as rotation, width and height shifts, and horizontal flipping
7	Batch Normalization	Normalize the activations of each layer to speed up training and improve stability
8	Dropout	Apply dropout regularization after fully connected layers to prevent overfitting
9	Loss Function	Binary Crossentropy - Suitable for binary classification tasks
10	Optimizer	Adam optimizer with a learning rate of 0.0001
11	Metrics	Accuracy, Precision, Recall, F1-score
12	Batch Size	64
13	Epochs	20
14	Early Stopping	Monitor validation loss and stop training if no improvement after a certain number of epochs
15	Class Weighting	Adjust class weights to handle class imbalance if necessary

(b) Training parameters for ResNet50 on COVID-19 Chest X-ray Image dataset

Table 3.18 outlines the training parameters for ResNet50 when applied to chest X-ray (CXR) image datasets. The batch size is set to 64, indicating the number of samples processed before updating the model's parameters. Training occurs over 20 epochs, with each epoch representing one complete pass through the entire dataset. A learning rate of 0.0001 is employed, determining the step size in updating the model's weights during training. The model is configured to classify images into two classes: 'Covid' and 'Normal', reflecting the binary nature of the classification task. The input shape of the images is (150, 150, 3), denoting the dimensions of the input images in terms of height, width, and color channels (RGB). Additionally, a seed value of 42 is specified for reproducibility, ensuring consistent results across different runs.

Furthermore, the dataset is divided into training, validation, and testing sets using a split percentage of 80%, 10%, and 10%, respectively. This division allows for evaluating the model's performance on unseen data during training and tuning hyperparameters to prevent overfitting. The target size of the images is set to (150, 150), indicating the dimensions to which the images are resized during preprocessing. The classes description specifies the categories the model is trained to recognize, while the class mode is binary, indicating that each sample belongs to one of the two classes. These training parameters collectively facilitate the development of an effective deep learning model for COVID-19 chest X-ray classification tasks.

Table 3.18: ResNet50 training parameters for CXR image datasets

S/N	Parameter	Values
1	Batch size	64
2	Epoch	20
3	Learning rate	0.0001
4	Classes	2
5	Input shape	(150, 150, 3)
6	Seed	42
7	Target size	(150, 150)
8	Classes description	‘Covid’, ‘Normal’
9	Class mode	Binary
10	Split percentage for training	80%
11	Split percentage for validation	10%

(c) Design Consideration for GRAD-CAM on ResNet50 using Chest-Ray Image Dataset

Design Consideration for GRAD-CAM on ResNet50 using Chest X-ray Image Dataset" entails outlining the methodological framework for implementing GRAD-CAM (Gradient-weighted Class Activation Mapping) on the ResNet50 architecture specifically for analyzing chest X-ray images in the context of COVID-19 diagnosis.

Table 3.19 outlines the design considerations for implementing Grad-CAM on the ResNet50 architecture using CXR (Chest X-ray) image datasets for COVID-19 classification. ResNet50, renowned for its deep residual network structure consisting of 50 layers, serves as the foundational architecture for this application.

The input images of 224x224x3 resized to 150x150 pixels, are fed into the model, and Grad-CAM is applied to generate heatmaps that highlight crucial regions contributing to the classification decisions. These heatmaps offer insights into the model's decision-making process, aiding in the interpretation and validation of COVID-19 classification results based on CXR images.

The Grad-CAM activation layer, typically chosen as the last convolutional layer or another appropriate layer, facilitates the analysis of feature importance within the ResNet50 model. By leveraging ReLU activation functions and employing backpropagation to compute gradient-weighted importance scores, Grad-CAM generates class activation maps, visualizing significant regions within the CXR images.

This process enables clinicians and researchers to better understand the model's predictions, validate its performance, and identify key features indicative of COVID-19 presence in CXR images. Overall, these design considerations enhance the interpretability and transparency of the ResNet50 model's classification decisions, contributing to more reliable and accurate COVID-19 diagnosis based on CXR imaging data.

Table 3.19: Design consideration for GRAD-CAM on ResNet50 with CXR dataset

S/N	Design Consideration	Details
1	Model Architecture	ResNet50 - A deep residual network with 50 layers
2	Input Image Size	Input shape: (224, 224, 3) - Chest X-ray images resized to 150x150 pixels
3	Grad-CAM Application	Apply Grad-CAM technique to the ResNet50 model for heatmap generation
4	Number of Layers	ResNet50 comprises 50 convolutional layers with residual connections
5	Ground Truth Label	Utilize true labels associated with the COVID-19 Chest X-ray image datasets
6	Grad-CAM Activation Layer	Select the last convolutional layer or an appropriate layer for Grad-CAM analysis
7	Heatmap Resolution	The heatmap resolution matches the size of the feature maps from the last layer
8	Activation Function	ReLU activation function is used throughout the ResNet50 architecture
9	Gradient Weighting Method	Compute gradient-weighted importance scores using backpropagation
10	Class Activation Mapping	Generate class activation maps to visualize important regions in the images

2. InceptionV3 Model (Design consideration of InceptionV3 for Classification of COVID-19 CT-Scan Image Dataset)

InceptionV3 Model (Design consideration of InceptionV3 for Classification of COVID-19 CT-Scan Image Dataset)" involves outlining the approach for utilizing the InceptionV3 architecture to classify CT-scan images for COVID-19 diagnosis.

Table 3.20 presents the model settings tailored for InceptionV3 utilized in the classification of COVID-19 CT-Scan images. This comprehensive setup encompasses various design considerations crucial for building an effective and robust image classification model.

The model architecture adopts InceptionV3, a deep convolutional neural network renowned for its efficiency and effectiveness in image classification tasks. With its intricate design, InceptionV3 can capture intricate features within CT-scan images, facilitating accurate classification. Input images are resized to (150, 150, 3) pixels, undergoing pre-processing through data normalization to rescale pixel values to a range of [0, 1], ensuring uniformity and facilitating stable model training. Leveraging pre-trained weights from ImageNet, the model is initialized to benefit from prior knowledge learned on a diverse set of images, enhancing its capability to extract relevant features from CT-scan images effectively. Additionally, fine-tuning of the pre-trained InceptionV3 layers allows adaptation to the specific characteristics of CT-scan images, optimizing performance for the classification task at hand.

Data augmentation techniques, including random transformations such as rotation, width and height shifts, and horizontal flipping, are employed to augment the dataset, enhancing model generalization and robustness. Batch normalization is utilized to normalize activations, accelerating training and improving model stability. Dropout regularization is applied after fully connected layers to mitigate overfitting. The model is trained using the Adam optimizer with a learning rate of 0.0001, with metrics such as accuracy, precision, recall, and F1-score monitored to evaluate performance. With a batch size of 64 and training for 50 epochs, the model undergoes extensive training, while early stopping is implemented to prevent overfitting by monitoring validation loss. Class weighting adjustment further ensures the model's ability to handle class imbalances within the dataset, collectively contributing to the development of a reliable and accurate classification model for COVID-19 CT-Scan images.

Table 3.20: Model setting for InceptionV3 for COVID-19 CT-Scan image classification

S/N	Design Consideration	Details
1	Model Architecture	InceptionV3 - A deep convolutional neural network known for its efficiency and effectiveness in image classification tasks
2	Input Shape	(150, 150, 3) - RGB images resized to 150x150 pixels
3	Pre-processing	Data normalization to rescale pixel values to the range [0,1]
4	Pre-trained Weights	'imagenet' - Initialize with weights pre-trained on ImageNet
5	Fine-tuning	Optional fine-tuning of the pre-trained InceptionV3 layers for adaptation to CT-scan images
6	Data Augmentation	Apply random transformations such as rotation, width and height shifts, and horizontal flipping to increase dataset diversity
7	Batch Normalization	Normalize the activations of each layer to speed up training and improve stability
8	Dropout	Apply dropout regularization after fully connected layers to prevent overfitting
9	Loss Function	Binary Crossentropy suitable for binary classification tasks
10	Optimizer	Adam optimizer with a learning rate of 0.0001
11	Metrics	Accuracy, Precision, Recall, F1-score
12	Batch Size	64
13	Epochs	50
14	Early Stopping	Monitor validation loss and stop training if no improvement after a certain number of epochs
15	Class Weighting	Adjust class weights to handle class imbalance if necessary

(b) Training Parameters for InceptionV3 on COVID-19 CT-Scan Image Dataset

Table 3.21 provides a summary of the training parameters specifically tailored for training the InceptionV3 model on the COVID-19 CT-Scan image dataset. These parameters are crucial for configuring the training process and optimizing the model's performance.

The batch size, set to 64, dictates the number of samples processed in each training iteration. A larger batch size can lead to faster training but may require more memory, while a smaller batch size can provide more accurate gradients but may increase training time. The specified number of epochs, 50, defines the number of complete passes through the training dataset during the training process. This parameter influences the duration of training and plays a key role in determining when the model converges to an optimal solution.

The learning rate, set to 0.0001, controls the step size during the optimization process. It determines how much the model's weights are updated during each training iteration. A smaller learning rate can result in slower but more stable training, while a larger learning rate may lead to faster convergence but risks overshooting the optimal solution.

Additionally, the table outlines the classes present in the dataset ('covid', 'normal'), the input shape of the images (150x150 pixels with 3 channels), and the rescaling factor applied to pixel values to normalize them to the range [0,1]. Other parameters such as shuffling the dataset during training, setting a random seed for reproducibility, defining the target size of images, the number of classes, class mode, and data splits for training and validation are also specified to ensure proper dataset handling and model training. These parameters collectively contribute to the successful training and evaluation of the InceptionV3 model on the COVID-19 CT-Scan image dataset.

Table 3.21: InceptionV3 training parameters for COVID-19 CT-Scan image dataset

S/N	Parameter	Values
1	Batch size	64
2	Epoch	50
3	Learning rate	0.0001
4	Classes	‘covid’, ‘normal’
5	Input shape	(150, 150, 3)
6	Rescale	1./255
7	Shuffle	True
8	Seed	42
9	Target size	150, 150
10	n_classes	2
11	Class mode	Binary
12	Train split	0.80
11	Validation split	0.10

(c) Design Consideration for GRAD-CAM on InceptionV3 using CT-Scan Image Dataset

Table 3.22 provides a comprehensive overview of the design considerations for applying Grad-CAM on the InceptionV3 model using COVID-19 CT-Scan image datasets. InceptionV3, recognized for its efficiency and effectiveness in image classification tasks, serves as the foundational architecture for this application. The input images are resized to 150x150 pixels to fit the model's input shape, allowing for uniform processing of CT-Scan images across the dataset. Grad-CAM is then applied to the InceptionV3 model to generate heatmaps, highlighting critical regions within the CT-Scan images that contribute to classification decisions. These heatmaps aid in understanding the model's decision-making process and provide insights into the features indicative of COVID-19 presence in CT-Scan images.

The design considerations also encompass the selection of an appropriate activation layer within the InceptionV3 model for Grad-CAM analysis. The heatmap resolution matches the size of the feature maps from the chosen layer, ensuring accurate visualization of important regions within the CT-Scan images. Throughout the architecture, activation functions such as ReLU are utilized, facilitating the computation of gradient-weighted importance scores using backpropagation. This method enables the generation of class activation maps, which visualize significant regions within the CT-Scan images that influence the model's classification decisions. Overall, these design considerations enhance the interpretability and transparency of the InceptionV3 model's classification decisions, contributing to more reliable and accurate COVID-19 diagnosis based on CT-Scan imaging data.

Table 3.22: Design consideration for GRAD-CAM on InceptionV3 Model using CT-Scan Image dataset

S/N	Design Consideration	Details
1	Model Architecture	InceptionV3 - A deep convolutional neural network known for its efficiency and effectiveness
2	Input Image Size	Input shape: (150, 150, 3) - CT-Scan images resized to 150x150 pixels
3	Grad-CAM Application	Apply Grad-CAM technique to the InceptionV3 model for heatmap generation
4	Number of Layers	InceptionV3 comprises multiple layers including convolutional and pooling layers
5	Ground Truth Label	Utilize true labels associated with the COVID-19 CT-Scan image datasets
6	Grad-CAM Activation Layer	Select the appropriate layer within the InceptionV3 model for Grad-CAM analysis
7	Heatmap Resolution	The heatmap resolution matches the size of the feature maps from the selected layer
8	Activation Function	Utilize activation functions such as ReLU throughout the InceptionV3 architecture
9	Gradient Weighting Method	Compute gradient-weighted importance scores using backpropagation
10	Class Activation Mapping	Generate class activation maps to visualize important regions in the CT-Scan images

3. Convolution Neural Network (CNN) Model

(a) Design Consideration for CNN Model for Classification of COVID-19 Blood Smear Image Dataset

Table 3.23 provides a detailed overview of the model settings tailored for a Convolutional Neural Network (CNN) designed specifically for the classification of COVID-19 Blood smear images. These design considerations encompass various aspects crucial for building an effective and accurate classification model.

Firstly, the model architecture is specified as a CNN, which is well-suited for image classification tasks due to its ability to learn spatial hierarchies of features. The input shape is defined as (150, 150, 3), indicating RGB images resized to 150x150 pixels, ensuring uniformity in the input data format. Pre-processing involves data normalization to rescale pixel values to the range [0,1], a crucial step to ensure consistency and stability during training.

The architecture includes multiple convolutional layers, each followed by max-pooling layers to extract and downsample features from the input images. The use of ReLU activation functions helps introduce non-linearity, enabling the model to learn complex patterns within the data. Additionally, fully connected layers are employed to further process the extracted features, with a dropout rate of 0.5 applied after each fully connected layer for regularization, mitigating overfitting and improving generalization. The output layer consists of a single unit with sigmoid activation, suitable for binary classification tasks, such as distinguishing between COVID-19 positive and negative blood smear images.

Other key considerations highlighted in the table include the choice of loss function (Binary Crossentropy), optimizer (Adam optimizer with a learning rate of 0.0001), evaluation metric (Accuracy), batch size (48), and number of epochs (20). Data augmentation techniques such as random rotation, width and height shifts, and horizontal flipping are employed to increase the robustness of the model and reduce overfitting. Additionally, early stopping criteria are implemented to halt training when validation loss fails to improve, preventing unnecessary computation and potential overfitting. Class weighting adjustment and the potential utilization of transfer learning further enhance the model's adaptability and performance, ensuring it is well-equipped to handle the complexities of COVID-19 Blood smear image classification tasks.

Table 3.23: Model settings for CNN for COVID-19 Blood smear image classification

S/N	Design Consideration	Details
1	Model Architecture	Convolutional Neural Network (CNN)
2	Input Shape	(150, 150, 3) - RGB images resized to 150x150 pixels
3	Pre-processing	Data normalization to rescale pixel values to the range [0,1]
4	Convolutional Layers	Conv2D: 32 filters, kernel size (3,3), activation: ReLU; MaxPooling2D: Pool size (2,2)
5	Filter / Kernel size	Conv2D: 64 filters, kernel size (3,3), activation: ReLU; MaxPooling2D: Pool size (2,2)
6	Fully Connected Layers	Flatten, Dense: 128 units, activation: ReLU
7	Output Layer	Dense: 1 unit, activation: Sigmoid (binary classification)
8	Dropout	Applied after each fully connected layer with a rate of 0.5 for regularization
9	Loss Function	Binary Crossentropy
10	Optimizer	Adam optimizer with a learning rate of 0.0001
11	Metrics	Accuracy
12	Batch Size	48
13	Epochs	20
14	Data Augmentation	Random rotation, width and height shifts, horizontal flipping
15	Early Stopping	Stop training when validation loss does not improve for 3 epochs
16	Class Weighting	Adjust class weights to handle class imbalance if necessary
17	Transfer Learning	Fine-tune pre-trained CNN models if available and applicable

(b) Training Parameters for CNN on COVID-19 Blood Smear Image Dataset

Table 3.24 presents the training parameters specifically configured for training a Convolutional Neural Network (CNN) on the COVID-19 Blood Smears image dataset. These parameters are instrumental in shaping the training process and optimizing the CNN's performance for accurate classification.

The batch size, set to 48, denotes the number of samples processed in each training iteration. This parameter influences the trade-off between training speed and memory usage. With a batch size of 48, the network processes 48 images simultaneously before updating the model's weights based on the calculated gradients. A smaller batch size allows for more frequent weight updates and may improve convergence but can lead to longer training times, while a larger batch size can expedite training but may require more memory.

The number of epochs, specified as 20, determines the number of complete passes through the training dataset during the training phase. Each epoch involves presenting the entire dataset to the CNN for training. The choice of epoch count is crucial as it affects the duration of training and the model's ability to converge to an optimal solution. Too few epochs may result in underfitting, where the model fails to capture the dataset's patterns adequately, while too many epochs can lead to overfitting, where the model memorizes the training data without generalizing well to unseen data.

Additionally, the table outlines other essential parameters such as the learning rate, classes present in the dataset ('covid', 'normal'), input and target image sizes, number of classes, class mode, and data splits for training and validation. These parameters collectively contribute to the successful training and evaluation of the CNN model on the COVID-19 Blood Smears image dataset, ensuring efficient dataset handling, model training, and performance assessment.

Table 3.24: CNN training parameters for COVID-19 Blood Smears image dataset

S/N	Parameter	Values
1	Batch size	48
2	Epoch	20
3	Learning rate	0.0001
4	Classes	‘covid’, ‘normal’
5	Input shape	(150, 150, 3)
6	Target size	150, 150
7	n_classes	2
8	Class mode	Binary
9	Train split	0.80
10	Validation split	0.10

(c) Design Consideration for GRAD-CAM on CNN using Blood Smears Image Dataset

Table 3.25 provides a detailed overview of the design considerations for applying Grad-CAM on the CNN architecture using COVID-19 Blood Smears image datasets. The CNN model architecture is chosen for its effectiveness in image classification tasks, particularly in analyzing complex features present in Blood Smear images. The input images are resized to 150x150 pixels to fit the model's input shape, ensuring consistency across the dataset. Grad-CAM is then applied to the CNN model to generate heatmaps, highlighting crucial regions within the Blood Smear images that contribute to classification decisions. These heatmaps enable visualization of important features, aiding in the interpretation of the model's decision-making process and providing insights into COVID-19 diagnosis based on Blood Smear images.

Key design considerations include the selection of an appropriate activation layer within the CNN model for Grad-CAM analysis and ensuring that the heatmap resolution matches the size of the feature maps from the chosen layer. Activation functions such as ReLU are utilized throughout the CNN architecture, facilitating the computation of gradient-weighted importance scores using backpropagation.

This method allows for the generation of class activation maps, which visually depict significant regions within the Blood Smear images that influence the model's classification decisions. In all, these design considerations enhance the interpretability and transparency of the CNN model's classification decisions, contributing to more accurate and reliable COVID-19 diagnosis based on Blood Smear imaging data.

Table 3.25: Design consideration for GRAD-CAM on CNN Model using Blood smears Image dataset

S/N	Design Consideration	Details
1	Model Architecture	Convolutional Neural Network (CNN)
2	Input Image Size	Input shape: (150, 150, 3) - Blood Smear images resized to 150x150 pixels
3	Grad-CAM Application	Apply Grad-CAM technique to the CNN model for heatmap generation
4	Number of Layers	CNN comprises multiple layers including convolutional and pooling layers
5	Ground Truth Label	Utilize true labels associated with the COVID-19 Blood Smears image datasets
6	Grad-CAM Activation Layer	Select the appropriate layer within the CNN model for Grad-CAM analysis
7	Heatmap Resolution	The heatmap resolution matches the size of the feature maps from the selected layer
8	Activation Function	Utilize activation functions such as ReLU throughout the CNN architecture
9	Gradient Weighting Method	Compute gradient-weighted importance scores using backpropagation
10	Class Activation Mapping	Generate class activation maps to visualize important regions in the Blood Smear images

4. Random Forest Classifier

(a) Design Consideration of Random Forest for Classification of COVID-19 Clinical Data

Table 3.26 provides a comprehensive overview of the design considerations and settings for implementing a Random Forest model for COVID-19 classification based on clinical data. Random Forest, known for its ensemble learning approach based on decision trees, offers a robust framework for handling classification tasks, particularly with complex and heterogeneous datasets such as those found in clinical contexts.

The design considerations cover various stages of model development, starting from feature selection and data preprocessing to hyperparameter tuning and model evaluation. Feature selection involves identifying relevant features from the COVID-19 clinical dataset, ensuring that the model focuses on informative variables crucial for accurate classification. Data preprocessing steps, including handling missing values, encoding categorical variables, and scaling numerical features, are essential for preparing the dataset for training the Random Forest model effectively.

Hyperparameter tuning is a critical aspect addressed in the design considerations, employing grid search with cross-validation to find the optimal hyperparameters for the Random Forest Classifier. This process explores different values for key hyperparameters such as the number of estimators, maximum features, maximum depth, and criterion, ensuring the model's performance is optimized. Additionally, cross-validation is implemented to assess model performance and generalization ability, enhancing the reliability and robustness of the model in real-world applications.

Moreover, the Random Forest model inherently provides feature importance scores, aiding in model interpretability by assessing which clinical variables contribute most to the classification task.

Leveraging these feature importance insights, stakeholders can gain valuable insights into the underlying patterns and relationships within the clinical dataset, facilitating informed decision-making in clinical settings. Finally, the trained Random Forest model can be deployed to classify new instances of COVID-19 clinical data, enabling real-time decision support and enhancing healthcare outcomes.

Table 3.26: Random Forest settings for COVID-19 Classification based on Clinical data

S/N	Design Consideration	Details
1	Model Architecture	Random Forest - An ensemble learning method based on decision trees
2	Feature Selection	Identify relevant features from the COVID-19 clinical dataset
3	Data Preprocessing	Handle missing values, encode categorical variables, and scale numerical features if necessary
4	Hyperparameter Tuning	Utilize grid search with cross-validation to find the optimal hyperparameters for the Random Forest Classifier
5	Hyperparameters	Explore different values for key hyperparameters such as the number of estimators, maximum features, maximum depth, and criterion
6	Cross-validation	Implement cross-validation to assess model performance and generalize to unseen data
7	Class Weighting	Adjust class weights to handle class imbalance if present in the clinical dataset
8	Evaluation Metrics	Utilize metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC)
9	Feature Importance	Assess feature importance to understand which clinical variables contribute most to the classification
10	Model Interpretability	Random Forest inherently provides feature importance, aiding interpretability of the model's decisions
11	Number of Estimators	Test different numbers of decision trees in the forest, including 200 and 500
12	Maximum Features	Evaluate different strategies for determining the maximum number of features to consider when looking for the best split, including 'auto', 'sqrt', and 'log2'
13	Maximum Depth	Experiment with various maximum tree depths, including 4, 5, 6, 7, and 8 levels
14	Criterion	Assess both 'gini' and 'entropy' criteria for splitting nodes in the decision trees
15	Model Evaluation	Perform model evaluation using 5-fold cross-validation to assess performance and generalization ability
16	Random State	Set the random state to 42 to ensure reproducibility of results
17	Execution Time	Measure the execution time of the Random Forest Classifier training process using the Python time module
18	Deployment	Deploy the trained Random Forest model to classify new instances of COVID-19 clinical data

3.5.3 Proposed System Architecture

This section discusses the proposed system's architectural design and structure, outlining how its modules, subsystems, and components are arranged and interact with one another. By outlining relationships, dependencies, and data flow, it acts as a blueprint for comprehending how the system accomplishes its objectives and functionality.

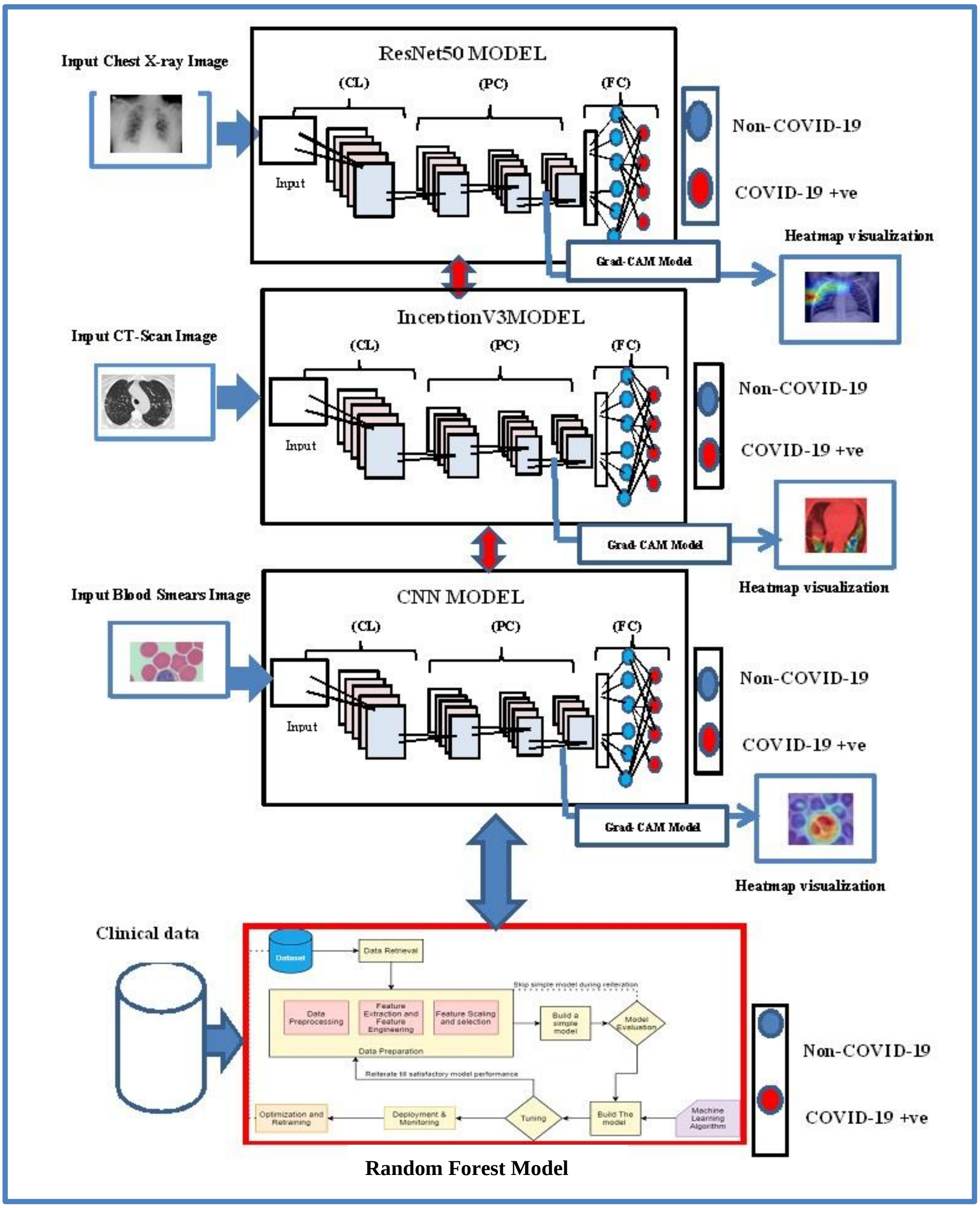

Figure 3.9: Proposed model architecture

Figure 3.9 presents an illustrative representation of the proposed hybrid deep learning model designed for predicting and interpreting COVID-19 using multiple data modalities. The diagram showcases the simultaneous execution of four distinct models, each tailored to analyze a specific data modality: ResNet50 + Grad-CAM for Chest X-Ray images, Inception V3 + Grad-CAM for CT scan image dataset, CNN + Grad-CAM for Blood smear image dataset, and Random Forest Classifier for Clinical information.

This parallel processing approach enables the models to operate concurrently, leveraging the unique characteristics of each data modality to generate predictions of COVID-19 cases. The integration of Grad-CAM (Gradient-weighted Class Activation Mapping) with the deep learning models enhances interpretability by highlighting regions of interest within the input images, thereby aiding in the interpretation of model predictions.

The utilization of ResNet50 and Inception V3 architectures for analyzing Chest X-Ray and CT scan images, respectively, allows for the extraction of intricate features indicative of COVID-19 infection patterns. Similarly, employing CNN architecture for analyzing Blood smear images facilitates the detection of abnormalities associated with viral infections. Additionally, leveraging a Random Forest Classifier for Clinical information enables the incorporation of diverse patient data, such as symptoms, demographic information, and medical history, into the prediction process.

By combining insights from multiple data modalities, the hybrid deep learning model aims to enhance the robustness and accuracy of COVID-19 prediction and interpretation. The concurrent execution of these models enables the generation of diverse predictions, which can be aggregated to confirm the confidence level of the results and provide comprehensive insights into the presence and severity of COVID-19 cases. This approach facilitates a more holistic understanding of the disease and supports informed decision-making in clinical settings.

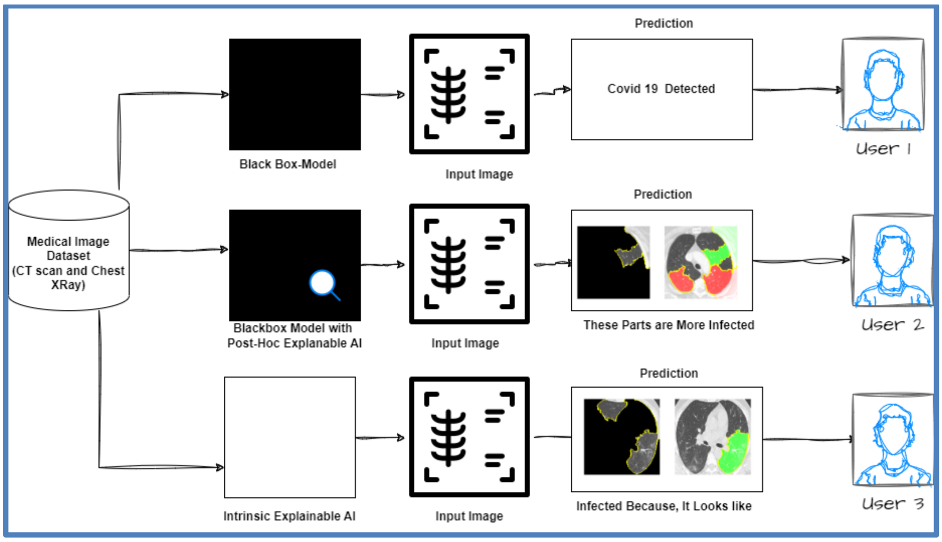

Figure 3.10: Explainable AI Black Box for Interpretation using GRAD-CAM technique

The eXplainable Artificial Intelligence (XAI) is an explainable model designed to achieve trustworthiness, causality, transferability, confidence, fairness, accessibility, and interactivity using the GRAD-CAM technique. It provides insights into how predictions are made and is recommended to be understandable by healthcare professionals as depicted in Figure 3.10.

However, the definition of XAI lacks clarity; hence, the terms "explainable" and "interpretable" are associated with XAI, where black-box models are considered "explainable" when explanations are provided by post hoc, while "interpretable" models generate human-understandable outputs step-by-step. Figure 3.10 illustrates the black-box AI with explainable AI and how the results impact the user. The top branch represents the process of a black-box model, which provides only result outcomes, such as class labels (e.g., COVID or non-COVID). The middle and bottom branches depict two XAI methods. The middle branch demonstrates the use of a saliency map as an example of an XAI model, while the bottom branch showcases the prototype method.

Model Deployment and Monitoring: After validation and interpretation, the model is deployed in the intended application environment. It is monitored to ensure it continues to perform accurately and reliably. Ongoing monitoring and updates may be necessary to address any issues or adapt to changing data distributions or requirements.

3.5.4 The Proposed Technique Layers

(a) Convolution Neural Network (CNN) for Classification and Prediction

The output volume of a CNN is created by a differential function applied over a number of layers that take the input image volume. Three main layers are used by CNN to conduct its activities. Convolution Layer (CONV), pooling Layer (Pool), and Fully Connected Layer (FC) are among the layers.

(i) Convolution (CONV) Layers: This layer consists of a number of separate filters that are convolved with the input volume such as the Chest X-ray images, CT images, Blood smear images, etc) to produce a map of the neurons' activation. A multilayered architecture is used to extract the useful information from the input images (Chest X-ray images, CT images, Blood smear images, etc). Each filter can be of a different type, and each one extracts a distinct feature such as edges, lines, and both vertical and horizontal lines. Convolutional feature extraction is made possible by the CNN layers. The CNN model relies heavily on this in order to retrieve deep features. Equations (3.1) and (3.2) produce the 2D convolution respectively.

$$\mathrm{y}[\mathrm{m},\mathrm{n}] = x[m,n]x\ h[m,n] \qquad (3.1)$$

$$y[m,n] = \sum_{i=-\propto}^{\propto} \sum_{j=-\propto}^{\propto} x[i,j]\ .\ h[m-i,n-j] \qquad (3.2)$$

Where x[m, n] = Input (e,g Chest X-ray images or CT images or Blood smear images)

m, n = number of rows, and number of columns of the correspondingly input image

i, j = row index and column index

Similarly, the size of the image following convolution is provided in Equation (3.3) $size = \left[\left(\frac{m+2p-n}{s}+1,\ \ \frac{m+2p-n}{s}+1\right)\right]$ (3.3)

Where m = quantity of input image features; n = size of the convolution kernel;

p = padding; s = stride.

(ii) Pooling (POOL) Layer: The function of the pooling layer (PL) is to reduce the size of the representation and speed up processing of the model. The PL summarizes the operations of regional patches of nodes. Though there are two major methods of pooling: maximum pooling and average pooling. The maximum pooling is employed in building the proposed model. Max pooling stores the maximum value for each feature map patch while discarding the rest. All other values are dropped in favor of the patch's average value. It makes more sense to use the max pooling since the maximum value denotes that it has the most impact on that particular region of the input images (Chest X-ray images, CT images, Blood smear images, etc). As a result, other patches are removed.

(iii) Fully Connected (FC) Layer: This layer combines the output of all preceding layers into a single vector that can be connected to the input layer of the following stage. At the bottom of this layer is a Softmax layer that does the proper label. The final probability for each layer is provided by the output layer. The CNN's fully connected portion uses its back propagation technique to calculate the weights that are most accurate. Each node's labels are determined by the weights that it receives. The nodes in this model will be given a priority of either 1 or 0, since it uses binary classification.

(iv) Batch normalization: it reduces the amount by which the hidden unit values changes, which is known as the covariance shift. If the distribution of x changes after the algorithm has been taught to translate some input x to some output y, the prediction will not perform as well, and retraining may be necessary. Each layer may be learned independently with batch normalization. Because batch normalization ensures that no activation goes too high or too low, learning rates can be adjusted higher.

Batch normalization provides regularization effects, which helps to decrease overfitting as well. Batch normalization also normalizes the outputs of the preceding activation layers in order to increase the stability of the neural network. Since each layer in this method receives two additional factors, the normalized output is multiplied by gamma (standard deviation) and beta (mean). The mini-batch mean is calculated mathematically using Equation (3.4):

$$\mu\beta = \frac{1}{m}\sum_{i-1}^{m} x_i \tag{3.4}$$

Equation (3.5) displays the variance in the mini-batch:

$$\sigma_{\beta}^{2} = \frac{1}{m}\sum_{i-1}^{m}(x_i - \mu\beta)^2 \tag{3.5}$$

While Equation (3.6) provides normalization:

$$\overline{x_i} = \frac{x_i - \mu\beta}{\sqrt{\sigma_{\beta}^{2} + \varepsilon}} \tag{3.6}$$

(v) Dropout: On any given dataset, including the dataset utilized under this research work or study; deep neural networks are prone to overfit early. Dropout is a regularization technique that simulates concurrent training of several neural networks with various designs. Some dropout layers are randomly ignored during training. By treating data in this way, the neural network simulates the effect of a new layer. Each update is actually carried out with a fresh perspective on the layer. With this approach, the network becomes more resilient as more noise is added.

(vi) Loss function: The categorical cross-entropy loss function has been incorporated into the model. The aforementioned loss function is used to gauge how well the binary classification model, whose output ranges from 0 to 1, is performing. When the probability of the class under consideration is set to 1 and the probability of all other classes is set to 0, categorical cross-entropy compares the distribution of predictions with the true distribution. In Equation (3.7), categorical cross-entropy is displayed:

$$L[y, y] = -\sum_{j=0}^{M} \sum_{i=0}^{M} [y_{ij}\ x\ Log(y_{ij})] \tag{3.7}$$

(b) Grad-CAM for Visualization and Interpretability

The Grad-CAM formula involves computing the gradient of the target class score with respect to the feature maps in the final convolutional layer of the CNN model. This gradient is then used to calculate the importance weights for each feature map, which are then used to generate a heat map.

Assumption:

If the target class score is denoted by `y_c` and the feature maps of the last convolutional layer are denoted by `A`.

Then the Grad-CAM heatmap `L` for a specific class `c` is calculated as follows:

$$L\_c = \text{ReLU}(\sum(\text{i}, \text{j}) \frac{\partial y\_c}{\partial A\_ij}) \quad (3.8)$$

Where:

L_c = Grad-CAM heatmap for class `c`.

∂y_c / ∂A_ij represents the gradient of the class score `y_c` with respect to the activation value `A_ij` of the feature map at position `(i, j)`.

Operational procedure:

Choose a target layer: specifically choose the target layer which is the last convolutional layer of a Convolutional Neural Network (CNN) for analysis, is as follows:

$$TL = [TNL - (NFCL + 1)] \quad (3.9)$$

Where:

TL = Target Layer

TNL = "Total Number of Layers" refers to the total number of layers in the CNN architecture, including convolutional layers, pooling layers, and fully connected layers.

NFCL = "Number of Fully Connected Layers" which represents the count of fully connected (dense) layers in the CNN

The "+1" accounts for the output layer, which is typically a fully connected layer.

The idea behind this formula is to choose the last convolutional layer before the fully connected layers begin. This layer often retains important spatial information and feature representations that can be useful for visualization or analysis tasks.

2. Forward Pass: Perform a forward pass through your model to obtain the prediction scores for the input image.

$$Input : X \tag{3.10}$$

$$Z1 = W1 * X + b1 \tag{3.11}$$

$$A1 = f1(Z1) \tag{3.12}$$

$$\mathrm{Z2 = W2 * A1 + b2} \tag{3.13}$$

$$\mathrm{A2 = f2(Z2)} \tag{3.14}$$

Z_output = W_output * A_previous + b_output (3.15)

$$\mathrm{A\_output = f\_output(Z\_output)} \tag{3.16}$$

$$\text{Prediction Scores: A\_output} \tag{3.17}$$

3. Compute Gradients: Compute the gradient of the target class score with respect to the feature maps of the chosen layer. This is done using back propagation technique.

Loss Calculation:

$$\text{Error Actual Target Ouptut} - \text{ Predicted Output} \tag{3.18}$$

Output Layer Gradient:

$$\mathrm{dLoss/dA\_output} = \text{ Gradient of the loss with respect to the output} \tag{3.19}$$

Backpropagation through Activation:

$$\frac{\mathrm{dLoss}}{\mathrm{dZ_{output}}} = \mathrm{dLoss/dA_{output}} * \mathrm{df_{output}/dZ_{output}} \tag{3.20}$$

Where:

df_output/dZ_output = The derivative of the output layer's activation function.

Backpropagation through Weight and Bias:

$$\mathrm{dLoss\ /dW_{output}} = \mathrm{dLoss\ /dZ_{output}} * \mathrm{dZ_{output}/dW_{output}} \tag{3.21}$$

$$\mathrm{dLoss\ /db_{output}} = \mathrm{dLoss\ /dZ_{output}} * \mathrm{dZ_{output}/db_{output}} \tag{3.22}$$

Update the output layer weights and biases using the computed gradients.

Propagation to Previous Layers:

For each previous layer (working backward from the output to the input):

$$\mathrm{dLoss/dA_{previous}} = (\mathrm{W_{next}} * \mathrm{dLoss\ /dZ_{next}}) \tag{3.23}$$

$$\mathrm{dLoss/dZ_{previous}} = \mathrm{dLoss/dA_{previous}} * \mathrm{df_{previous}/dZ_{previous}} \quad (3.24)$$

$$\mathrm{dLoss/dW_{previous}} = \mathrm{dLoss/dZ_{previous}} * \mathrm{dZ_{previous}/dW_{previous}} \quad (3.25)$$

$$\mathrm{dLoss/db_{previous}} = \mathrm{dLoss/dZ_{previous}} * \mathrm{dZ_{previous}/db_{previous}} \quad (3.26)$$

4. Global Average Pooling (GAP):

GAP is to condense the spatial dimensions of feature maps into a single value per channel. It involves taking the average of all values within each channel of the feature map. The formula for calculating Global Average Pooling is as follows:

The formula for calculating Global Average Pooling is as follows:

For a given feature map F with dimensions H (height), W (width), and C (number of channels):

$$\text{Global Average Pooling (GAP)} = \frac{1}{\mathrm{H} * \mathrm{W}} * \mathrm{Sum(F)} \quad (3.27)$$

Where:

(1 / (H * W)) represents normalization factor to ensure that the final output is scaled appropriately.

Sum(F) represents the sum of all values in the feature map F for each channel

5. Heatmap Generation:

Heatmap generation is a visualization technique commonly used in deep learning and computer vision to highlight the regions of an image that contribute most to a certain prediction made by a neural network. One common approach to generating a heatmap involves using the gradients of the predicted class score with respect to the input image pixels.

The formula for generating a heatmap using Grad-CAM is as follows:

Let:

$$\mathrm{F_{k(x,y)}} = \text{the activation of the k} - \text{th channel in the final CL} \quad (3.28)$$

$$\mathrm{A^c} = \text{the gradient of the predicted class score (logit) w.r.t the activation F\_k} \quad (3.29)$$

The heatmap H^c is obtained by taking the global average pooling of A^c over the spatial dimensions:

$$H^c = \frac{1}{H * W} * \Sigma.\Sigma\ A\hat{}c(x, y) \quad (3.30)$$

6. Normalize and Visualize: Normalize the heatmap values and apply ReLU to remove negative values. Finally, overlay the heatmap on the original image to visualize the regions of interest that contributed most to the prediction.

(a) Normalize Heatmap Values: Normalize the heatmap values to have a range between 0 and 1. This can be done using min-max normalization:

$$normalized\_heatmap = \frac{(\text{heatmap} - \text{min}(\text{heatmap})}{(max(heatmap) - min(heatmap))} \tag{3.31}$$

This ensures that the heatmap values are within the [0, 1] range.

Apply ReLU (Rectified Linear Unit): Apply the ReLU function element-wise to the normalized heatmap to remove negative values:

$$relu\_heatmap \ = \ max(0, normalized\_heatmap \tag{3.32}$$

Overlay Heatmap on Original Image: Overlay the processed heatmap on the original image to visualize the important regions:

$$overlaid_{image} \ = \ original_{image} \ * \ transparency \ + \ heatmap_{color} \ * \ (1 \ - \ transparency) \tag{3.33}$$

3.5.5 Proposed Model Algorithms adopted

This research work explored four main algorithms for classifying COVID-19 from X-rays: ResNet50, InceptionV3, CNN, and Random Forest. Additionally, a technique called Grad-CAM was used to visualize the image regions most influential in the model's decisions. In simpler terms, the researchers compared four different approaches and used a visualization tool to understand how they make decisions.

Algorithm I: ResNet50 Model on COVID-19 Chest X-ray Images

Step 1: Preprocessing:

(a) Load and preprocess dataset:

(i) Load chest X-ray images and labels (COVID-19 positive/negative).

(ii) Resize images to 150x150x3 (ResNet50 input size).

(iii) Normalize pixel intensities to [0, 1].

(iv) Apply data augmentation techniques (optional): rotation, flipping, cropping, color jittering.

Step 2: Feature Extraction:

(a) Load pre-trained ResNet50:

Load model with ImageNet weights, freezing most layers.

(b) Forward pass:

Input preprocessed image X ? R^(150x150x3).

Pass through convolutional layers with residual blocks:

Residual block structure:

X_l = F(X_(l-1)) + X_(l-1) (identity shortcut connection)

F(X_(l-1)) involves convolutional layers, batch normalization, and ReLU activation.

Step 3: Classification and Training:

Global average pooling:

Y_GAP = GAP(Y_l) (average activation values across spatial dimensions)

Final classification layer:

Y_final = W_final * Y_GAP + b_final

Y_hat = softmax(Y_final) ? R^(2) (COVID-19 positive/negative probabilities)

Loss function:

L = -1/N ? _i^N y_i log(y_hat_i) (cross-entropy loss)

Optimizer:

Adam optimizer (commonly used)

Fine-tuning:

Unfreeze some layers, train on COVID-19 dataset.

Step 4: Evaluation and Prediction:

Split dataset:

Train/validation/test sets (e.g., 80/10/10)

Training:

Iterate through batches:

Forward pass, calculate loss.

Backward pass, update weights (W, b) using optimizer.

Validation:

Monitor performance on validation set: accuracy, precision, recall, F1-score.

Adjust hyperparameters if needed.

Testing:

Evaluate on unseen test set (optional).

Prediction:

Use trained model for prediction on new chest X-ray images.

Figure 3.11: ResNet50 Algorithm on COVID-19 Chest X-ray image

Algorithm II: InceptionV3 Model on COVID-19 CT-Scan Image Dataset

Step 1: Preprocessing:

- (a) Load and preprocess dataset:
  - (i) Load CT-scan images and labels (COVID-19 positive/negative).
  - (ii) Resize images to 150x150x3 (InceptionV3 input size).
  - (iii) Normalize pixel intensities to [0, 1].
  - (iv) Apply data augmentation (optional): rotation, flipping, cropping.

Step 2: Feature Extraction:

- (a) Load pre-trained InceptionV3:
  - (i) Load model with ImageNet weights, freezing most layers.
- (b) Forward pass:
  - (i) Input preprocessed image X ? R^(150x150x3).
  - (ii) Pass through convolutional layers:
    - Y_l = σ(W_l * X_(l-1) + b_l) + BN(Y_l)
  - (iii) Apply Inception modules:
    - Y_l = concat(Conv1x1(X_(l-1)), Conv3x3(X_(l-1)), Conv5x5(X_(l-1)))

Step 3: Classification and Training:

- (a) Global average pooling:
  - (i) Y_GAP = GAP(Y_l)
- (b) Final classification layer:
  - (i) Y_final = W_final * Y_GAP + b_final
  - (ii) Y_hat = softmax(Y_final) ? R^(2) (COVID-19 positive/negative probabilities)
- (c) Loss function:
  - (i) L = -1/N ? _i^N y_i log(y_hat_i) (cross-entropy)
- (d) Optimizer:
  - (i) Adam optimizer
- (e) Fine-tuning:
  - (i) Unfreeze some layers, train on COVID-19 dataset.

Step 4: Evaluation and Prediction:

- (a) Split dataset:
  - (i) Train/validation/test sets (e.g., 80/10/10)
- (b) Training:
  - (i) Iterate through batches:
    - Forward pass, calculate loss.
    - Backward pass, update weights (W, b) using optimizer.
- (c) Validation:
  - (i) Monitor performance on validation set: accuracy, precision, recall, F1-score.
- (d) Testing:
  - (i) Evaluate on unseen test set (optional).
- (e) Prediction:
  - (i) Use trained model for prediction on new CT-scan images.

Figure 3.12: InceptionV3 Algorithm on COVID-19 CT-Scan Images

Algorithm III: CNN Algorithm for classification of COVID-19 Blood smears images

Step 1: Input and Preprocessing:

X ? R^(H×W×C): Represents the preprocessed blood smear image, where H is the height, W is the width, and C is the number of channels (usually 3 for RGB images). Preprocessing typically involves steps like resizing, normalization, and data augmentation (optional).

Step 2: Convolutional Layers:

Filter (Kernel): W_l ? R^(F×F×C_(l-1)×C_l) represents the filter applied at layer l. F is the filter size, C_(l-1) is the number of input channels, and C_l is the number of output channels.

Convolution: Y_l = σ(W_l * X_(l-1) + b_l), where σ is the activation function (e.g., ReLU), * denotes convolution operation, and b_l is the bias term.

Step 3: Pooling Layers:

Downsampling: Used to reduce spatial dimensions while preserving important features. Common types include max pooling and average pooling.

Step 4: Activation Functions:

Introduce non-linearity into the network, allowing it to learn complex relationships. Examples include ReLU, sigmoid, and tanh.

Step 5: Flatten Layer:

Converts the multi-dimensional output of convolutional layers into a 1D vector for feeding into fully connected layers.

Step 6: Fully Connected Layers:

Densely connect neurons from previous layers to learn higher-level features.

Y_final = W_final * Y_flatten + b_final, where W_final is the weight matrix and b_final is the bias vector.

Step 7: Softmax Activation:

Applied to the final layer output to produce class probabilities for COVID-19 positive/negative classification.

Step 8: Loss Function:

Measures the discrepancy between predicted and true labels. Common choices include cross-entropy loss.

Step 9: Optimizer:

Updates weights and biases to minimize the loss function, such as Adam or SGD.

Step 10: Training Loop:

Iterate through training data batches:

- Forward pass: Calculate outputs through the network.
- Backward pass: Compute gradients using backpropagation.
- Update weights and biases using the chosen optimizer.

Figure 3.13: CNN Algorithm on COVID-19 Blood smear image

Algorithm IV: Random Forest Algorithm on COVID-19 Clinical data

Step 1: Data Representation:

Let X be a matrix representing your clinical data, where each row represents a patient and each column represents a symptom/sign feature.

For categorical features, use one-hot encoding or suitable numerical representations.

Normalize numerical features if necessary (e.g., z-score normalization).

Step 2: Tree Building:

Splitting Criterion:

Information Gain (IG): IG(t, S) = H(S) - Σ_(v ? V_t) |S_v| / |S| * H(S_v), where H(S) is the entropy of the target variable in set S, S_v is the subset for value v of the splitting feature, and V_t is the set of possible values for that feature.

Gini Impurity: Gini(t, S) = 1 - Σ_(i ? C) p_i^2, where p_i is the proportion of class i in set S.

Choosing the Best Split:

The feature and value maximizing the chosen criterion (IG or Gini) for the current node become the splitting point.

Step 3: Prediction:

For a new data point x:

Traverse each decision tree in the forest, applying the splitting rules at each node based on feature values in x.

Reach a leaf node representing the predicted class (COVID-19 positive/negative).

Majority Vote:

The final prediction is the most frequent class predicted by the individual trees in the forest.

Step 4: Randomness:

Bootstrapping:

Randomly sample N data points (with replacement) from X, creating B bootstrap samples.

Each tree is built on a different bootstrap sample, introducing randomness in the forest.

Feature Selection:

At each node, randomly select mtry features (mtry < total features) for consideration in the splitting process. This further increases diversity and prevents overfitting.

Figure 3.14: Random Forest Algorithm on COVID-19 Clinical data

Algorithm V: GRAD-CAM Algorithm for Interpretability

Step 1: Setting up the trained model: Begin with a deep neural network model that has already been trained. This model, f, has several layers, including convolutional layers, fully connected layers, etc.

Step 2: Picking the intended layer: Locate the network's L, or last convolutional layer, which is where Grad-CAM will be used.

Step 3: Forward pass: With an input image, x, run the network in a forward pass to determine the anticipated class probabilities, or P(y|x), where y stands for the target class.

Step 4: Calculating gradients: Determine the gradients between the feature maps of the target layer, designated as AL, and the projected class score, P(y|x). Backpropagation is used for this: P(y|x)/AL.

Step 5: Combining gradients: To create a pooled gradient map, abbreviated as L, combine the gradients across the spatial dimensions of the feature maps using a pooling technique, such as mean or maximum pooling.

Step 6: Evaluation of the feature maps To create the weighted feature maps, multiply each feature map (AL_i) in the target layer by the corresponding gradient value (L_i): L_i * A_L_i = W_L_i.

Step 7: Combining the feature maps: To create a single heat map, HL = _i, add or average the weighted feature maps. W^L_i.

Step 8: Visualizing the heat map: To see the areas that have the greatest influence on the prediction of the network, overlay the heat map (HL) on the input image (x).

Step 9: Post-processing: Use post-processing methods to make the heat map, HL, more comprehensible and clear, such as thresholding or normalization.

Figure 3.15: GRAD.CAM Algorithm

Grad-CAM Algorithm for Visualization of ResNet50 on COVID-19 Chest X-ray Images

Step 1: Inputs:

X ? R^(H×W×C): Preprocessed chest X-ray image (H: height, W: width, C: channels).

F(·): ResNet50 model with weights trained for COVID-19 classification.

y_c: Target class index (e.g., 1 for COVID-19 positive).

Step 2: Forward Pass and Gradients:

a_l = F_l(a_(l-1)) (activation at layer l)

y_hat = F_out(a_final) (predicted class probabilities)

g_c = ?y_hat[y_c] / ?a_final (gradient of target class score w.r.t. final layer activations)

Step 3: Global Average Pooling (GAP):

GAP(g_c) = 1/(HW) * Σ_iΣ_j g_c[i, j] (average gradient per feature map)

Step 4: Class Activation Map (CAM) Generation:

W_c = GAP(g_c) ? a_final (element-wise product)

CAM_c = Σ_k W_c[k] (sum weighted activations across channels)

CAM_c_norm = (CAM_c - min(CAM_c)) / (max(CAM_c) - min(CAM_c)) (normalize to [0, 1])

Step 5: Visualization:

Upsample CAM_c_norm to match X size (e.g., bilinear interpolation)

Superimpose upsampled CAM_c_norm on X as a heatmap, with higher values indicating greater importance for class y_c.

Figure 3.16: GRAD.CAM Algorithm on ResNet50 Model using CXR Image

Grad-CAM for Visualization of InceptionV3 on COVID-19 CT-Scan Images

Step 1: Inputs:
X ? R^(H×W×C): Preprocessed CT-scan image (H: height, W: width, C: channels).
F(·): InceptionV3 model with weights trained for COVID-19 classification.
y_c: Target class index (e.g., 1 for COVID-19 positive).

Step 2:
Forward Pass and Gradients:
a_l = F_l(a_(l-1)) (activation at layer l)
y_hat = F_out(a_final) (predicted class probabilities)
g_c = ?y_hat[y_c] / ?a_final (gradient of target class score w.r.t. final layer activations)

Step 3: Global Average Pooling (GAP):
GAP(g_c) = 1/(HW) * Σ_iΣ_j g_c[i, j] (average gradient per feature map)

Step 4: Class Activation Map (CAM) Generation:
W_c = GAP(g_c) ? a_final (element-wise product)
CAM_c = Σ_k W_c[k] (sum weighted activations across channels)
CAM_c_norm = (CAM_c - min(CAM_c)) / (max(CAM_c) - min(CAM_c)) (normalize to [0, 1])

Step 5: Visualization:
Upsample CAM_c_norm to match X size (e.g., bilinear interpolation)
Superimpose upsampled CAM_c_norm on X as a heatmap, with higher values indicating greater importance for class y_c.

Figure 3.17: GRAD.CAM Algorithm on InceptionV3 Model using CT-Scan image

Grad-CAM for Visualization of CNN on COVID-19 Blood Smear Images

Step 1: Inputs:
X ? R^(H×W×C): Preprocessed blood smear image (H: height, W: width, C: channels).
F(·): CNN model with weights trained for COVID-19 classification.
y_c: Target class index (e.g., 1 for COVID-19 positive).

Step 2: Forward Pass and Gradients:
a_l = F_l(a_(l-1)) (activation at layer l)
y_hat = F_out(a_final) (predicted class probabilities)
g_c = ?y_hat[y_c] / ?a_final (gradient of target class score w.r.t. final layer activations)

Step 3: Global Average Pooling (GAP) or Global Max Pooling (GMP):
Depending on the CNN architecture:
GAP(g_c) = 1/(HW) * Σ_iΣ_j g_c[i, j] (average gradient per feature map)
GMP(g_c) = max(g_c[:,:,:]) (maximum gradient value across channels)
Choose the pooling method consistent with the final layer used.

Step 4: Class Activation Map (CAM) Generation:
W_c = GAP/GMP(g_c) ? a_final (element-wise product)
CAM_c = Σ_k W_c[k] (sum weighted activations across channels)
CAM_c_norm = (CAM_c - min(CAM_c)) / (max(CAM_c) - min(CAM_c)) (normalize to [0, 1])

Step 5: Visualization:
Upsample CAM_c_norm to match X size (e.g., bilinear interpolation)
Superimpose upsampled CAM_c_norm on X as a heatmap, with higher values indicating greater importance for class y_c.

Figure 3.18: GRAD.CAM Algorithm on CNN Model using Blood Smears image

3.6 Model Evaluation

Model evaluation is a crucial step in assessing the performance and effectiveness of the proposed hybrid deep learning based model for COVID-19 prediction and interpretations. Standard metrics such as accuracy, precision, recall, and F1-score play a vital role in quantifying the model's predictive capabilities and identifying areas for improvement.

Accuracy, as a metric, reflects the proportion of correctly classified instances compared to all instances. Although it offers a broad assessment of model performance, it may not adequately capture performance nuances in datasets where one class predominates over others, as seen in medical diagnoses like COVID-19 detection.

Precision, on the other hand, gauges the model's capability to avoid false positives by measuring the ratio of true positive predictions to all positive predictions. In the context of COVID-19 diagnosis, precision is essential to ensure that identified cases are genuine positives, minimizing the strain on healthcare resources caused by false alarms.

Recall, also referred to as sensitivity; assesses the model's ability to identify all actual positive instances by calculating the ratio of true positive predictions to all actual positive instances in the dataset. In COVID-19 diagnosis, recall plays a critical role in identifying all infected individuals accurately, reducing the risk of unnoticed cases that could contribute to further transmission.

Lastly, the F1-score, which harmonizes precision and recall, delivers a balanced evaluation of model’s performance by considering both false positives and false negatives. This metric is particularly valuable in scenarios with imbalanced datasets, such as medical diagnoses, where precision and recall need to be harmonized due to the significant consequences of false positives and false negatives.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.34}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.35}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.36}$$

$$\text{F1} - \text{score} = 2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})} \tag{3.37}$$

3.7 System Modeling

System modeling is the process of developing graphic representations to comprehend and convey a system's structure, behavior, and relationships. It uses a variety of models, including structural, behavioral, functional, and data models, to capture distinct system components.

To better understand the working connection and representations of the proposed system, this research work takes into account the use of use case diagrams, sequence diagrams, and activity diagrams.

3.7.1 Use Case Diagram

A use case diagram in UML is a visual tool that shows how people such as healthcare professionals interacts with the proposed explainable deep learning model for COVID-19 prediction. It displays the activities the system is capable of taking to illustrate the functional requirements.

Figure 3.6 demonstrates how the system aids users in achieving their objectives by showing the connections between actors and use cases. A use case diagram essentially provides a clear visual representation of the functionality and user interactions of the system.

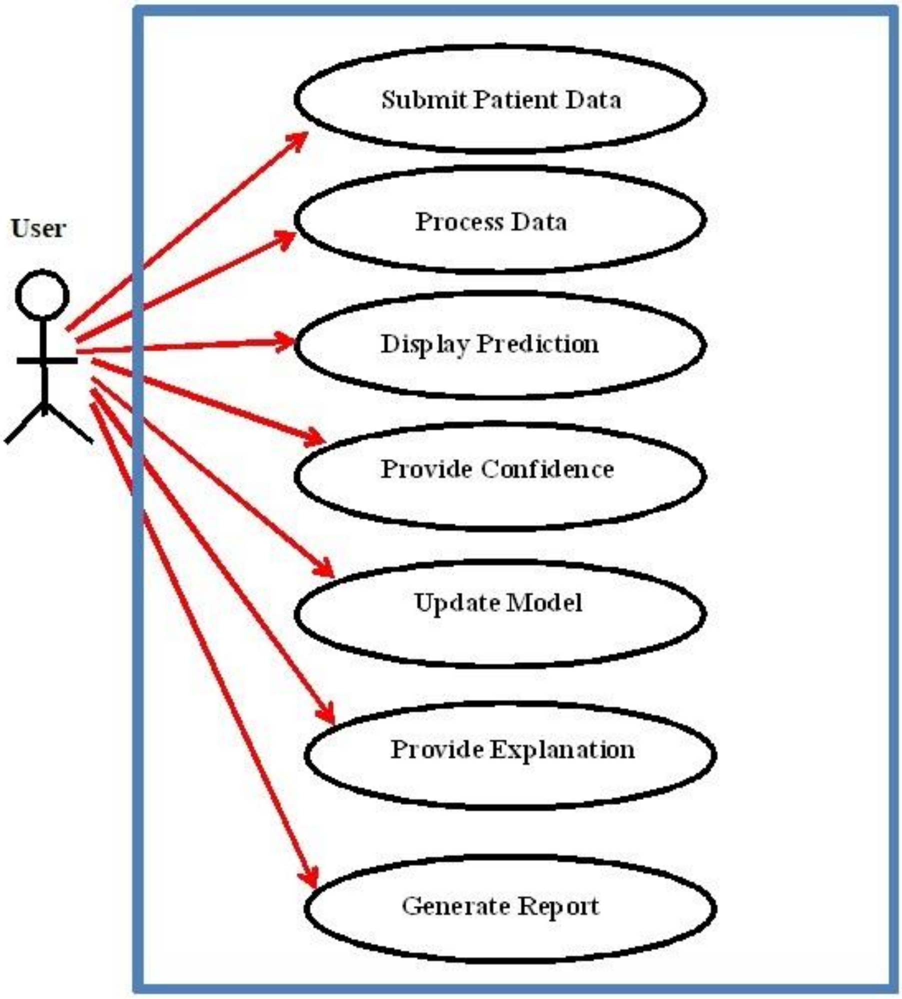

Figure 3.19: Use case diagram of a deep learning model for COVID-19 prediction

Figure 3.18 describes the following actions:

Actors:

(a) User: The person who interacts with the deep learning model for COVID-19 prediction.

Use Cases:

1. Submit Patient Information:

(a) Description: The user enters the relevant patient information, including age, gender, symptoms, and medical history.

(b) Primary Actor: User

(c) Preconditions: The user has access to the deep learning model interface.

(d) Postconditions: The patient information is captured and ready for processing.

2. Process Patient Data:

(a) Description: The deep learning model processes the submitted patient data to generate a prediction for COVID-19.

(b) Primary Actor: Deep Learning Model

(c) Preconditions: Patient information has been submitted.

(d) Postconditions: The model generates a prediction based on the input data.

3. Display Prediction:

(a) Description: The deep learning model displays the prediction outcome to the user.

(b) Primary Actor: User

(c) Preconditions: Patient data has been processed.

(d) Postconditions: The user can view the predicted outcome.

4. Provide Prediction Confidence:

(a) Description: The deep learning model provides a confidence level or probability associated with the prediction.

(b) Primary Actor: Deep Learning Model

(c) Preconditions: Prediction outcome is available.

(d) Postconditions: The user is informed about the confidence level of the prediction.

5. Explain Prediction:

(a) Description: The deep learning model provides an explanation or justification for its prediction.

(b) Primary Actor: Deep Learning Model

(c) Preconditions: Prediction outcome is available.

(d) Postconditions: The user can access an explanation of the factors influencing the prediction.

6. Update Model:

(a) Description: The deep learning model is periodically updated with new data to improve its accuracy and performance.

(b) Primary Actor: System Administrator

(c) Preconditions: New data is available for model training.

(d) Postconditions: The model is updated and ready for future predictions.

7. Generate Prediction Report:

(a) Description: The deep learning model generates a comprehensive report summarizing the prediction results.

(b) Primary Actor: Deep Learning Model

(c) Preconditions: Prediction outcome is available.

(d) Postconditions: The user can access a report containing the patient information, prediction outcome, and confidence level.

3.8 System Requirement Specification

A System Requirement Specification (SRS) is a formal document that outlines the precise functional and non-functional requirements of a system.

3.8.1 User Requirements

User requirements for a deep learning model for predicting COVID-19 can vary depending on the specific needs and context of the users. However, some common user requirements include:

1. Accuracy: Users expect the model to have high accuracy in predicting COVID-19 cases. The model should provide reliable and precise predictions to assist in decision-making and resource allocation.

2. Sensitivity and Specificity: The model should have a high sensitivity to correctly identify true positive cases of COVID-19 and a high specificity to accurately identify true negative cases. This is crucial for minimizing false positives and false negatives.
3. Speed and Efficiency: Users require a model that can provide fast and efficient predictions to facilitate real-time decision-making. The model should be able to process data quickly, especially in situations where timely predictions are essential.
4. Generalizability: The model should be able to generalize well to unseen data and different populations. It should be capable of making accurate predictions across various demographics and geographical regions.
5. Interpretablility: Users may require the model to provide explanations or insights on how it arrived at its predictions. Interpretablility is essential for understanding the factors contributing to the predictions and building trust in the model's outcomes.
6. Scalability: The model should be scalable to handle large volumes of data, especially in scenarios where there is a significant increase in the number of COVID-19 cases or when analyzing data from multiple sources.
7. Integration and Compatibility: Users may require the model to be compatible with existing systems, software, or platforms, allowing for seamless integration into their workflow or infrastructure.
8. Accessibility: The model should be user-friendly and accessible to a wide range of users, including healthcare professionals, researchers, and policymakers. It should have a user-friendly interface and provide clear and actionable outputs.
9. Data Privacy and Security: Users expect the model to adhere to data privacy regulations and ensure the security of sensitive healthcare information. Robust measures should be in place to protect patient data and maintain confidentiality.

10. Continuous Improvement and Updates: Users may require the model to be regularly updated with the latest data and research findings related to COVID-19. Continuous improvement ensures that the model remains accurate and effective over time.

3.8.2 Functional Requirements

Functional requirements for a deep learning model for predicting COVID-19 typically involve the specific features and capabilities that the model should possess. While the exact requirements may vary based on the intended use and context, here are some of the functional requirements:

1. Data Preprocessing: The model should be able to preprocess raw input data, which may include medical images, patient data, or other relevant information, to ensure compatibility and quality for further analysis.
2. Training and Model Development: The model should have the capability to undergo a training phase where it learns from labeled data to optimize its parameters and improve its predictive performance. This includes selecting appropriate deep learning architectures, specifying hyperparameters, and implementing training algorithms.
3. Feature Extraction: The model should be able to automatically extract relevant features from the input data, such as visual patterns in medical images or temporal patterns in time-series data, to capture the necessary information for COVID-19 prediction.
4. Prediction and Classification: The model should be able to make predictions or classifications based on the learned features. This involves providing inputs to the trained model and obtaining output predictions, such as identifying whether a patient is COVID-19 positive or negative.
5. Real-time Processing: Depending on the application, the model may need to process data in real-time to provide immediate predictions. This requirement is especially relevant for scenarios where timely decision-making is critical, such as in healthcare settings.

6. Model Evaluation: The model should include mechanisms for evaluating its performance, such as accuracy, sensitivity, specificity, precision, or recall. This allows for the assessment of the model's predictive capabilities and helps identify areas for improvement.
7. Integration and Deployment: The model should be deployable in various environments and integrated into existing systems or platforms. It should have the flexibility to interact with other software components or APIs to exchange data and provide predictions.
8. User Interface: Depending on the intended users, the model may require a user-friendly interface that allows users to interact with the model, input data, visualize results, and interpret predictions. This interface should be intuitive and provide clear and understandable outputs.
9. Model Updates and Maintenance: The model should support updates and maintenance to adapt to evolving COVID-19 data and research findings. This ensures that the model remains accurate and effective over time as new information becomes available.
10. Scalability: The model should have the capability to handle large volumes of data and scale efficiently, particularly in situations where there is an increasing demand for COVID-19 predictions or when analyzing data from multiple sources.

3.8.3 Non-Functional Requirements

Non-functional requirements for a deep learning model for predicting COVID-19 focus on the quality attributes and characteristics that the model should possess. These requirements are not directly related to the specific functionality of the model but instead address aspects such as performance, reliability, security, and usability. Here are the non-functional requirements:

1. Performance: The model should be able to provide timely and efficient predictions, with low latency and fast processing times. It should be capable of handling large datasets and complex computations effectively.

2. Accuracy: The model should exhibit a high degree of accuracy in predicting COVID-19 cases. It should minimize false positives and false negatives, ensuring reliable and precise predictions.
3. Scalability: The model should be scalable to accommodate increasing data volumes and growing user demands. It should be able to handle larger datasets and adapt to changing requirements without compromising performance.
4. Robustness: The model should be resilient to noise, outliers, and variations in the input data. It should be able to handle different data distributions and maintain its performance across diverse scenarios.
5. Interpretability: The model should provide explanations or interpretations of its predictions. It should be able to highlight the important features or factors contributing to its decision-making process, improving transparency and trust in the model.
6. Reliability: The model should consistently deliver accurate predictions under different conditions and datasets. It should be robust against errors, failures, or data inconsistencies, ensuring dependable performance.
7. Security and Privacy: The model should adhere to data privacy regulations and security standards. It should employ measures to protect sensitive patient data and prevent unauthorized access or disclosure.
8. Usability: The model should have a user-friendly interface and be easy to use, even for individuals without deep technical knowledge. It should provide clear instructions, visualizations, and meaningful outputs to facilitate user understanding and decision-making.
9. Portability: The model should be portable across different platforms and environments. It should be compatible with various hardware configurations and software frameworks, allowing for easy deployment and integration.

10. Maintainability: The model should be designed in a modular and maintainable manner, making it easier to update, enhance, and debug. It should have clear documentation, well-organized code, and a version control system to support ongoing maintenance and improvements.

3.9 Software Methodology

Software methodology is the term used to describe the organized and methodical process used in the creation of software systems. It includes the guidelines that direct the entire software development lifecycle, from gathering requirements to deployment and maintenance. Waterfall, Agile, Scrum, Kanban, Lean, and DevOps are examples of common software methodologies. These approaches offer frameworks for task management, team collaboration, assuring quality, and effectively delivering software products.

The approach selected is determined by the needs of the project, the dynamics of the team, and organizational preferences. Software techniques, as a whole, aid in streamlining the development process and enhancing project results.

3.9.1 Software Methodology Adopted

Scrum methodology is adopted in this research wok. It is an agile project management framework that emphasizes iterative and incremental development. It is widely used in software development but can be applied to various industries and projects.

The adoption of Scrum methodology in the research work "Development of a Hybrid Deep Learning Based Model for COVID-19 Prediction and Interpretation Using Multiple Data Modalities" facilitated the achievement of various goals at different phases or stages.

Initially, during the planning phase, Scrum allowed for the establishment of clear objectives and priorities for the research project. This ensured that the team focused on the most critical tasks and features essential for developing the hybrid deep learning model effectively.

As the project progressed through its different stages, such as data collection, model development, and evaluation, Scrum provided a framework for iterative and incremental development. This approach allowed the research team to adapt to evolving requirements, incorporate feedback from stakeholders, and make necessary adjustments to the model architecture and implementation.

By breaking down the research work into manageable tasks and conducting regular sprint reviews and retrospectives, Scrum enabled continuous improvement and ensured that the project remained aligned with its objectives.

Additionally, Scrum promoted efficient collaboration and communication among all team members, including research students, supervisors, and industry-based mentors, to improve coordination and productivity during the research endeavor.

Through conducting regular stand-up meetings, the team could consistently review the progress of the research, identify any obstacles, and collectively brainstorm solutions. This iterative and collaborative method fostered transparency, accountability, and adaptability, allowing the team to promptly tackle emerging challenges and maintain progress towards the research objectives..

In essence, the utilization of the Scrum methodology provided a structured framework for navigating the complexities of the research endeavor, fostering agility, adaptability, and efficiency. Through its emphasis on iterative development, continuous feedback, and teamwork, Scrum streamlined the research process, ultimately leading to the successful development of the hybrid deep learning model for COVID-19 prediction and interpretation.

Nonetheless, employing the Scrum methodology for designing a Hybrid Deep Learning Based Model aimed at COVID-19 Prediction and Interpretation using Multiple Data Modalities presents numerous benefits customized to address the unique challenges and goals of the research endeavor.

1. Flexibility for Iterative Development: The Scrum methodology's iterative approach allows the research team to incrementally develop and refine the hybrid deep learning model. This flexibility is crucial in a research context where the understanding of COVID-19 and the available data modalities may evolve over time.
2. Collaborative Approach: Scrum encourages collaboration among interdisciplinary team members, including data scientists, medical researchers, and domain experts. This collaborative environment facilitates the integration of diverse perspectives and expertise necessary for developing a hybrid model that effectively leverages multiple data modalities.
3. Adaptation to Emerging Data: As new data sources and modalities become available or are discovered, Scrum enables the research team to adapt quickly and incorporate these insights into the model. This adaptability ensures that the model remains relevant and effective in capturing the complex dynamics of COVID-19 prediction and interpretation.
4. Continuous Feedback Loop: The regular review and retrospective meetings in Scrum provide opportunities for stakeholders to provide feedback on the model's performance and suggest improvements. This feedback loop is essential for refining the model iteratively and enhancing its predictive accuracy and interpretability.
5. Risk Mitigation: Scrum's emphasis on transparency and early detection of issues helps mitigate risks associated with model development, such as data biases, algorithmic errors, or performance bottlenecks. By identifying and addressing these risks early in the development process, the research team can ensure the robustness and reliability of the hybrid model.

Embracing the Scrum methodology in designing a Hybrid Deep Learning Based Model for COVID-19 Prediction and Interpretation Using Multiple Data Modalities offers a structured approach to navigate the intricacies of the research project. It promotes teamwork, flexibility, and ongoing enhancements, thereby bolstering the model's efficacy in advancing COVID-19 comprehension and control (Figure 3.20 depicts scrum methodology).

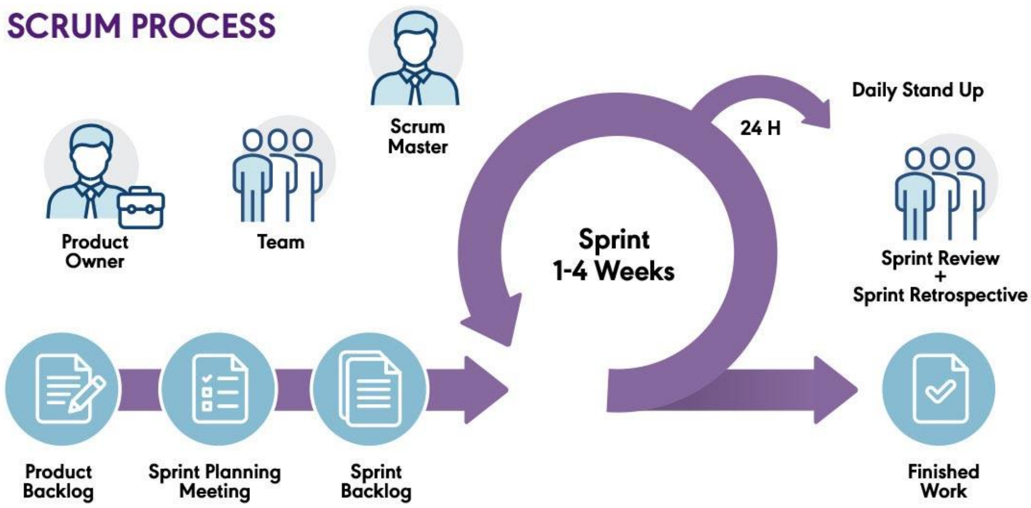

Figure 3.20: Scrum methodology adopted

.

3.10 System Implementation

This section explains the process of putting the proposed system into action and making it work effectively. It entails the actual creation, setting up, configuring, and deployment of the hardware, software, infrastructure, and system parts required to enable the system's desired operation.

System implementation typically follows the completion of system design and includes several key activities such as follows:

3.10.1 Computer Specification for Experiment

The computer system specifications used in experiments are crucial for maintaining research methodological integrity and scientific rigor.

The system specifications, outlined in Table 3.27, encompass the hardware components, such as the Intel® Core™ i7-8700 processor and NVIDIA GeForce GTX 1080Ti GPU, crucial for executing complex tasks in deep learning applications. With 64GB of DDR4 2133MHz RAM, the system ensures efficient handling of extensive datasets during model training and evaluation, while operating on the Microsoft Windows 10 platform provides stability and familiarity for software development.

In addition to its robust hardware, the system's software components, including Python 3.5, Anaconda3, and TensorFlow 1.0, facilitate seamless development and deployment of deep learning models. Leveraging these tools, along with the chosen operating environment, contributes to the system's effectiveness in COVID-19 prediction and interpretation across diverse data modalities.

Table 3.27: Computer Specification

Configuration	Parameters
CPU	Intel® Core ™ i7-8700, CPU@3.2 – 4.6GHz
GPU	NVIDIA GeForce GTX 1080Ti 8GB GDDR5X
Memory (RAM)	64GB DDR4 2133MHz
Operating System (OS)	Microsoft Windows 10
Development Language	Python 3.5
Development Platform	Anaconda3
Framework	Tensorflow 10.0

3.10.2 How to Download and Install Python 3.9

To download and install Python 3.9 on your system, follow these steps:

1. Visit the Official Python Website:

   Go to the official Python website at https://www.python.org/.

2. Navigate to the Downloads Page:

   Click on the "Downloads" tab in the menu bar to access the download page.

3. Choose the Python Version:

   Scroll down to find the section for Python 3.9.x releases. You may see different versions, but select the latest Python 3.9.x release available.

4. Select the Installer:

   Choose the installer suitable for your operating system. There are separate installers for Windows, macOS, and Linux. <Choose Windows Option>

5. Download the Installer:

   Click on the download link for the installer corresponding to your operating system. The download should start automatically.

6. Run the Installer:

Once the installer is downloaded, locate the file and double-click on it to run the installer.

7. Follow Installation Wizard:

Follow the installation wizard instructions. You may need to confirm permissions or provide administrator credentials depending on your operating system.

8. Customize Installation (Optional):

During the installation process, you may have the option to customize the installation settings. You can choose to add Python to the system PATH, customize the installation location, or install additional features.

9. Complete the Installation:

After selecting your preferences, proceed with the installation. Once the installation is complete, you should see a confirmation message.

10. Verify Installation:

To verify that Python 3.9 has been successfully installed, open a command prompt (Windows) or terminal (macOS/Linux) and type `python3 --version` or `python --version`. You should see the installed Python version displayed.

3.10.3 How to Install and Configure Anaconda3 in Microsoft Windows Environment

To install and configure Anaconda3 in a Microsoft Windows environment, follow these steps:

1. Download Anaconda Installer:

Visit the Anaconda website at https://www.anaconda.com/products/distribution and download the Anaconda3 installer for Windows. Choose the appropriate version (32-bit or 64-bit) based on your system architecture.

2. Run Anaconda Installer:

Once the installer is downloaded, locate the downloaded file (usually named something like `Anaconda3-<version>-Windows-x86_64.exe`) and double-click to run it.

3. Follow Installation Wizard:

   The Anaconda installer will launch. Follow the prompts in the installation wizard. You can choose the installation location, whether to add Anaconda to your system PATH, and whether to register Anaconda as your default Python.

4. Complete Installation:

   After selecting your preferences, click "Install" to proceed with the installation. The installer will copy files and configure Anaconda on your system. This process may take a few minutes.

5. Verify Installation:

   Once the installation is complete, you can verify that Anaconda has been installed by opening Anaconda Navigator from the Start menu or searching for it in the Windows search bar. Alternatively, you can open a command prompt and type `conda --version` to verify that the conda package manager is installed.

6. Configure Anaconda Environment (Optional):

   You can create and manage Python environments using Anaconda. Open Anaconda Navigator, and from the "Home" tab, you can create new environments, install packages, and launch applications.

7. Update Anaconda (Optional):

   It is a good practice to update Anaconda and its packages regularly to ensure you have the latest features and bug fixes. You can update Anaconda by opening Anaconda Navigator and navigating to the "Home" tab. From there, click on the "Update Available" button next to Anaconda and follow the prompts to update.

8. Additional Configuration (Optional):

   Depending on your specific requirements, you may need to configure Anaconda further, such as setting up environments, installing additional packages, or integrating Anaconda with your preferred development tools.

That is all about the installation process. Anaconda3 is now installed and configured on your Windows system, and you can start using it to manage Python environments, install packages, and develop Python applications.

3.10.4 How to Install and Configure Tensorflow in Microsoft Windows Environment

To install and configure TensorFlow in a Microsoft Windows environment, you can use Anaconda, which simplifies the process. Follow these steps:

1. Open Anaconda Prompt:

   Open Anaconda Navigator or Anaconda Prompt from the Start menu. Anaconda Prompt is a command-line tool that allows you to interact with Anaconda and manage Python environments.

2. Create a New Environment (Optional):

   It is recommended to create a new Python environment specifically for TensorFlow to avoid conflicts with existing packages. You can create a new environment using the following command:

   conda create -n tensorflow_env python=3.9

   Replace "tensorflow_env" with the name you want to give to your environment.

3. Activate the Environment:

   Activate the environment you just created using the following command:

   conda activate tensorflow_env

   Replace "tensorflow_env" with the name of your environment.

4. Install TensorFlow:

   Once the environment is activated, you can install TensorFlow using pip, which is the Python package installer. Run the following command:

   pip install tensorflow

   This command will install the latest stable version of TensorFlow available for your system.

5. Verify Installation:

After the installation is complete, you can verify that TensorFlow is installed correctly by importing it in a Python interpreter. Open a Python interpreter by typing `python` in the Anaconda Prompt and then enter the following Python code:

```
python
import tensorflow as tf
print(tf.__version__)
```

If TensorFlow is installed properly, it will print the version number.

6. Additional Configuration (Optional):

Depending on your specific requirements, you may need to configure additional settings for TensorFlow, such as GPU support or specific TensorFlow versions. Refer to the TensorFlow documentation for more information on advanced configurations.

That is all about the installation process. TensorFlow is now installed and configured in your Windows environment, and you can start using it for deep learning projects.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Data Gathering and Preprocessing Results

Results from the analysis of data collected and preprocessing conducted across various COVID-19 data modalities, including Chest X-Ray, CT Scan, Blood Smear, and Clinical Information, are visually depicted in Figures 4.1 and 4.2, respectively.

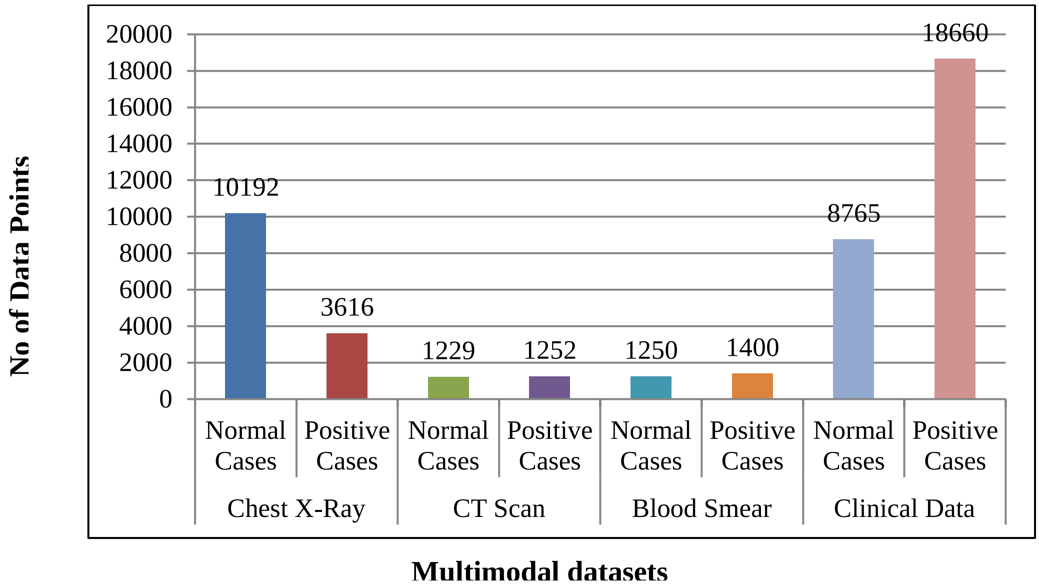

Figure 4.1: Visualization of the total datasets gathered from the four data modalities

Figure 4.1 visually represents the total datasets gathered from four different data modalities: Chest X-Ray, CT Scan, Blood Smear, and Clinical Data. It showcases the number of normal and positive cases identified in each modality. For Chest X-Ray, there are 10,192 normal cases and 3,616 positive cases. Similarly, for CT Scan, there are 1,229 normal cases and 1,252 positive cases. In Blood Smear data, there are 1,250 normal cases and 1,400 positive cases. Lastly, in Clinical Data, there are 8,765 normal cases and 18,660 positive cases. This visualization provides an overview of the distribution of cases across different modalities, aiding in understanding the dataset composition for COVID-19 classification.

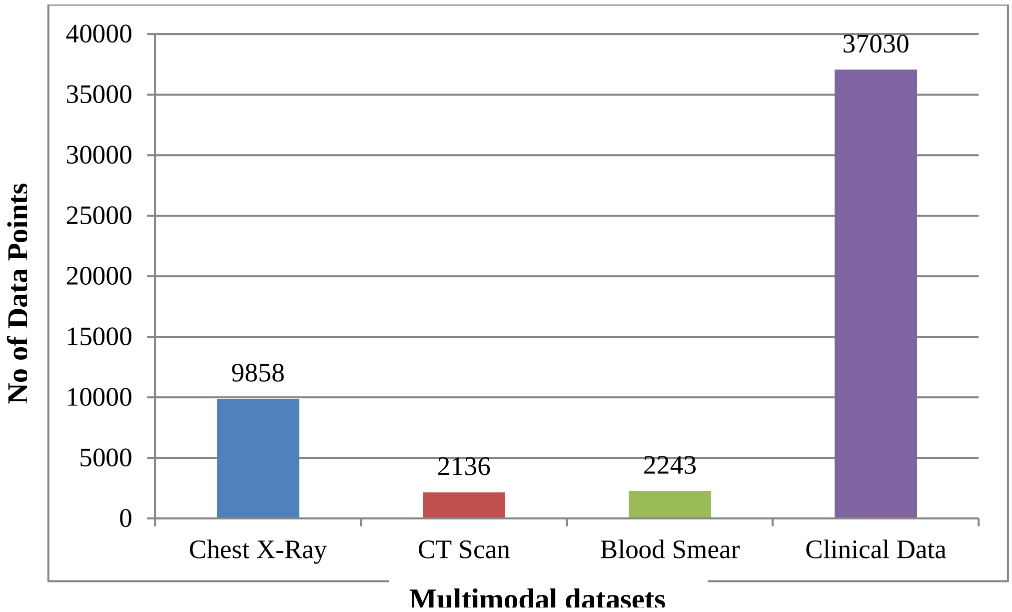

Figure 4.2: Visualization of the preprocessed datasets from the four data modalities

Figure 4.2 illustrates the preprocessed datasets from the four data modalities: Chest X-Ray, CT Scan, Blood Smear, and Clinical Data. It displays the total number of instances after preprocessing for each modality. Specifically, there are 9,858 instances for Chest X-Ray, 2,136 for CT Scan, 2,243 for Blood Smear, and 37,030 for Clinical Data.

This visualization provides insights into the dataset sizes after preprocessing, which is crucial for training and evaluating the hybrid deep learning model for COVID-19 prediction and interpretation across multiple data modalities.

4.2 Model Training Results

In this section, we present and discuss the outcomes derived from training diverse models utilized in our investigation. These models include ResNet50 applied to the COVID-19 Chest X-ray training set, InceptionV3 utilized for the COVID-19 CT-Scan image training set, CNN employed with the Blood smear image training set, and the Random Forest classifier utilized for the COVID-19 Clinical information training set.

4.2.1 Training Result ResNet50 Model on COVID-19 Chest X-Ray Datasets

Table 4.1 presents a comprehensive summary of the training results obtained using the ResNet50 architecture on the COVID-19 Chest X-ray dataset across 20 epochs. The table provides insights into the model's accuracy, validation accuracy, loss, validation loss, and the time taken for each epoch.

The accuracy metrics show a consistent improvement over epochs, starting at 0.74 in the first epoch and steadily increasing to 0.88 by the 20th epoch. Similarly, the validation accuracy follows a similar trend, starting at 0.80 and reaching 0.90 by the final epoch. This indicates that the model is effectively learning from the training data and generalizing well to unseen validation data.

Moreover, the loss metrics demonstrate a consistent decrease over epochs, indicating that the model is progressively minimizing its error. Both training and validation loss decrease from 0.56 and 0.39, respectively, in the first epoch to 0.28 and 0.24, respectively, in the 20th epoch.

This decreasing trend in loss signifies that the model is improving its predictive performance and becoming more adept at accurately classifying chest x-ray images related to COVID-19.

Additionally, the relatively stable time taken for each epoch suggests consistent training efficiency throughout the training process, with each epoch completed within a reasonable timeframe, as indicated by the seconds (s) values provided in the "Time" column of the table.

In all, these results highlight the effectiveness of the ResNet50 architecture in training a robust model for COVID-19 chest XRay image classification.

Table 4.1: Training results using ResNet50 on COVID-19 Chest XRay datasets

Epoch	Accuracy	Val_Accuracy	Loss	Val_Loss	Time
1	0.74	0.80	0.56	0.39	1268s
2	0.77	0.75	0.45	0.44	1293s
3	0.79	0.86	0.43	0.36	1110s
4	0.81	0.87	0.40	0.33	1225s
5	0.81	0.81	0.39	0.37	1683s
6	0.83	0.87	0.37	0.31	1055s
7	0.84	0.88	0.35	0.30	849s
8	0.84	0.85	0.35	0.30	846s
9	0.84	0.88	0.34	0.29	841s
10	0.84	0.87	0.35	0.33	839s
11	0.83	0.85	0.36	0.31	836s
12	0.85	0.89	0.33	0.28	839s
13	0.86	0.85	0.32	0.32	839s
14	0.86	0.89	0.31	0.26	835s
15	0.87	0.88	0.31	0.28	841s
16	0.86	0.88	0.31	0.29	842s
17	0.87	0.90	0.29	0.25	834s
18	0.87	0.88	0.30	0.27	841s
19	0.86	0.86	0.30	0.30	840s
20	0.88	0.90	0.28	0.24	834s

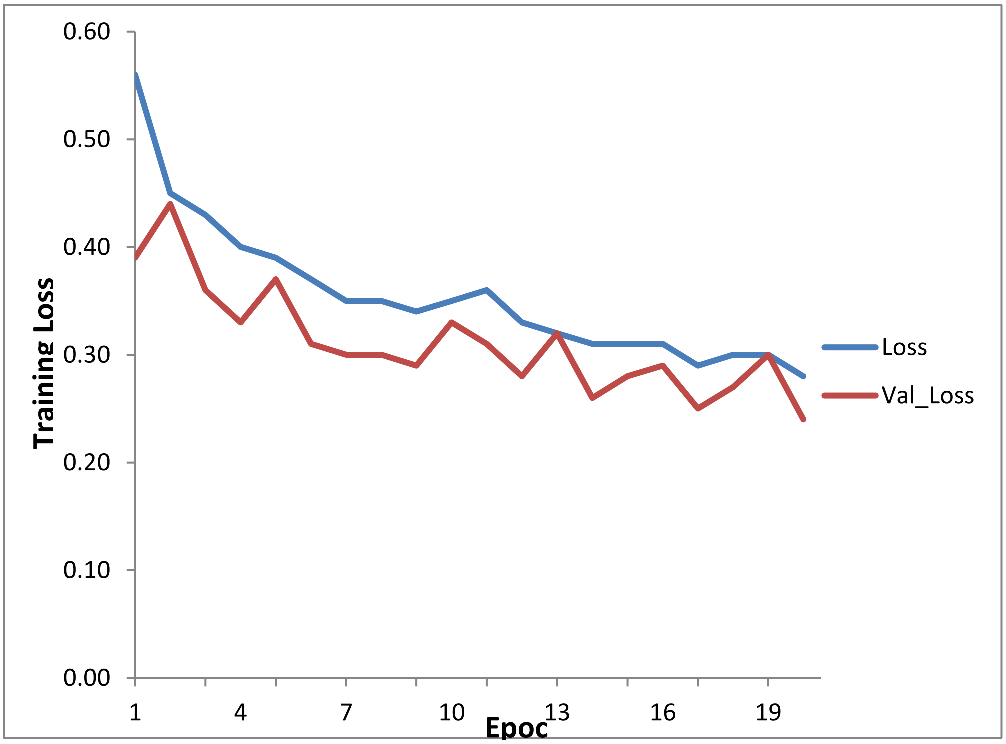

Figure 4.3: Visualization of training loss based on ResNet50 using COVID-19 X-Ray training set

Figure 4.3 illustrates a line graph illustrates the training losses of a ResNet50 model trained on a COVID-19 Chest X-Ray image dataset. The training loss metric signifies the model's performance on the training data, where lower values imply better comprehension of data patterns. The graph showcases a consistent decline in training loss, suggesting the model's progressive learning in accurately classifying COVID-19 Chest X-Ray images. However, the loss does not reach zero, indicating potential room for further enhancement. Furthermore, the graph displays the validation loss, representing the model's performance on an independent dataset not used during training. This loss, generally higher than the training loss, also demonstrates a declining trend over time. The concurrent decrease in both training and validation losses indicates the model's ability to learn without overfitting the training data, which occurs when a model excessively tailors itself to the training dataset, hindering generalization.

In essence, the ResNet50 model exhibits promising capability in classifying COVID-19 Chest X-Ray images with increasing accuracy over the training period. The training loss steadily decreases from around 0.60 to approximately 0.20 across 20 epochs, while the validation loss, marginally higher than the training loss, also declines. This close relationship between training and validation losses suggests the model's proficiency in generalizing to new data. However, the model's assessment on a larger dataset remains crucial to validate its performance and ensure its adaptability to novel data patterns.

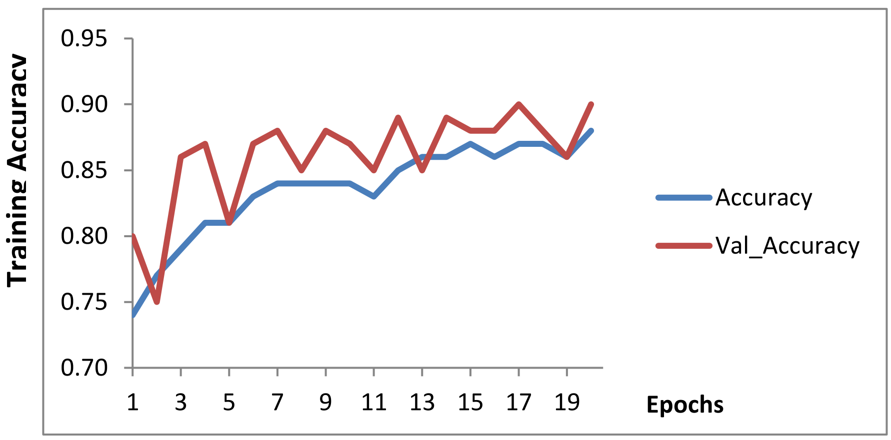

Figure 4.4: Visualization of training accuracy based on ResNet50 using COVID-19 X-Ray training set

Figure 4.4 shows a line graph that depicts the training accuracy and validation accuracy trends of a ResNet50 model trained on a COVID-19 Chest x-ray image dataset across 20 epochs. The training accuracy initiates near 0.75 and progressively climbs to about 0.95 by the end of the epochs, showcasing the model's consistent improvement in accurately categorizing COVID-19 Chest X-Ray images over time. Concurrently, the validation accuracy, slightly below the training accuracy initially, steadily ascends and aligns closely with the training accuracy, indicating the model's robust generalization without overfitting the training data.

Remarkably, both training and validation accuracies display rapid early increases, eventually leveling off as epochs progress, indicating a swift learning phase followed by stabilization with further training. Generally, these trends suggest promising outcomes for the ResNet50 model's performance in accurately classifying COVID-19 Chest X-Ray images. The model showcases high accuracy on both training and validation datasets while maintaining strong generalization capabilities.

4.2.2 Training Result based on InceptionV3 Model using CT-Scan Image Training Set

Table 4.2 presents a comprehensive overview of the training outcomes for a COVID-19 CT-scan classifier utilizing the InceptionV3 model across 50 training epochs. The results indicate a consistent pattern of diminishing loss values and escalating accuracy rates throughout the training process. Initially, the model started with a high loss of 11.073 and achieved an accuracy of 78.70%, steadily progressing to a significantly reduced loss of 0.1655 and an impressive accuracy of 92.57% by the final epoch. This pattern underscores the effectiveness of the training regimen in refining the model's predictive capabilities.

While the majority of epochs completed within a reasonable timeframe of approximately 400-500 seconds, there were notable exceptions such as epoch 27, which required a substantially longer training duration of 21320 seconds. Such anomalies may stem from various factors including hardware constraints, computational complexities, or potential issues encountered during training.

Addressing and comprehending these outliers is imperative for optimizing the training process and ensuring consistent model development.

In summary, the training results underscore the iterative nature of deep learning model refinement. Despite occasional deviations in training duration, the model consistently demonstrated improvements in both loss reduction and accuracy enhancement. These findings suggest that the InceptionV3 architecture, when applied to COVID-19 CT-scan classification, effectively learns and discerns significant patterns within the data, resulting in a highly accurate model for distinguishing between different CT-scan classes.

Table 4.2: Training results based on InceptionV3 using COVID-19 CT-Scan training set

Epoch	Loss	Accuracy	Time
1	11.073	78.70%	448s
2	0.386	84.68%	435s
3	0.3054	86.47%	494s
4	0.2995	86.42%	472s
5	0.2863	87.01%	473s
6	0.2859	87.01%	429s
7	0.2716	87.81%	463s
8	0.2689	88.13%	421s
9	0.2645	88.15%	397s
10	0.263	88.03%	382s
11	0.2468	88.99%	382s
12	0.2391	89.46%	432s
13	0.2393	89.68%	465s
14	0.2513	88.64%	390s
15	0.2367	89.44%	434s
16	0.2365	89.56%	457s
17	0.2246	90.08%	435s
18	0.2311	89.80%	427s
19	0.2258	90.10%	453s
20	0.2265	89.97%	451s
21	0.2515	88.72%	431s
22	0.2364	89.35%	398s
23	0.2349	89.30%	445s
24	0.2192	90.50%	474s
25	0.2155	90.38%	439s

26	0.2118	90.58%	431s
27	0.2148	90.44%	21320s
28	0.218	90.46%	454s
29	0.2163	90.29%	433s
30	0.2303	89.97%	506s
31	0.2213	90.00%	522s
32	0.2194	90.29%	516s
33	0.2198	90.44%	419s
34	0.21	91.04%	416s
35	0.2157	90.54%	405s
36	0.207	90.72%	412s
37	0.2022	91.15%	409s
38	0.1911	91.54%	412s
39	0.2	91.67%	409s
40	0.1815	91.97%	415s
41	0.1829	92.00%	410s
42	0.1709	92.75%	407s
43	0.1928	91.15%	404s
44	0.1889	91.23%	403s
45	0.1955	90.80%	506s
46	0.1967	90.42%	4807s
47	0.1837	91.25%	437s
48	0.1797	91.53%	435s
49	0.1824	91.65%	492s
50	0.1655	92.57%	626s

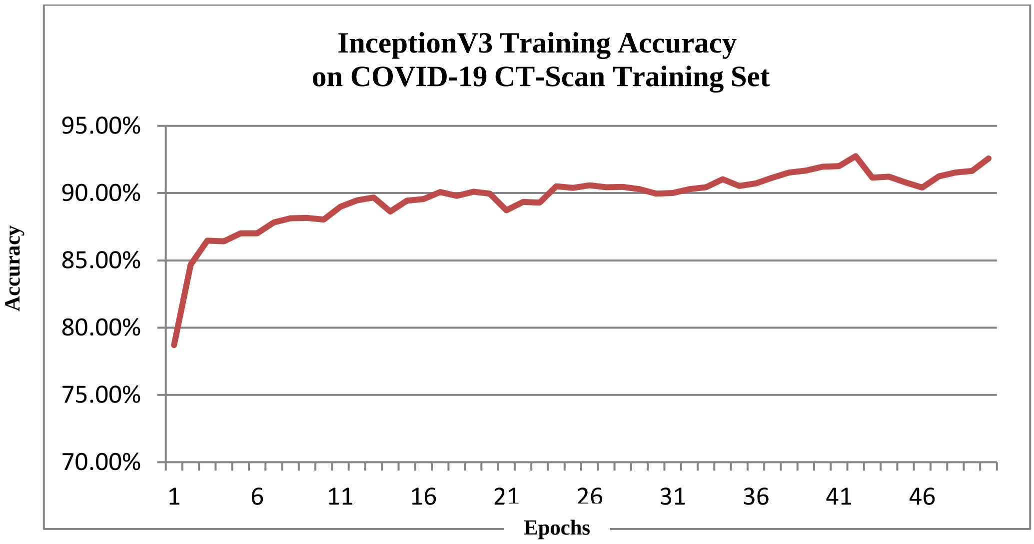

Figure 4.5: Visualization of training accuracy based on InceptionV3 using COVID-19 CT-Scan training set

Figure 4.5 depicts the training accuracy progression curve of the InceptionV3 model trained on the COVID-19 CT-Scan dataset across 50 epochs. The accuracy steadily increases over the epochs, starting at 78.70% in the first epoch and reaching 92.57% by the 50th epoch. This upward trend demonstrates the effectiveness of the model in learning from the training data and improving its predictive capabilities over time. Notably, there are fluctuations in accuracy across epochs, indicating variations in the model's performance at different stages of training. These fluctuations could be influenced by factors such as the complexity of the dataset, the model architecture, and the optimization algorithm employed during training.

The time taken for each epoch varies, with most epochs completing in a reasonable timeframe ranging from a few hundred seconds to around 500 seconds. However, there are outliers such as epoch 27, which took substantially longer at 21,320 seconds (approximately 5 hours and 55 minutes). This outlier could signify a potential issue during training, such as computational resource constraints or algorithmic inefficiencies. Despite this outlier, the overall training process appears to be efficient, with the model achieving significant accuracy improvements within a reasonable time frame. In summary, Figure 4.3 illustrates the progressive improvement of the InceptionV3 model's accuracy on the COVID-19 CT-Scan dataset, highlighting both its effectiveness and the computational resources required for training.

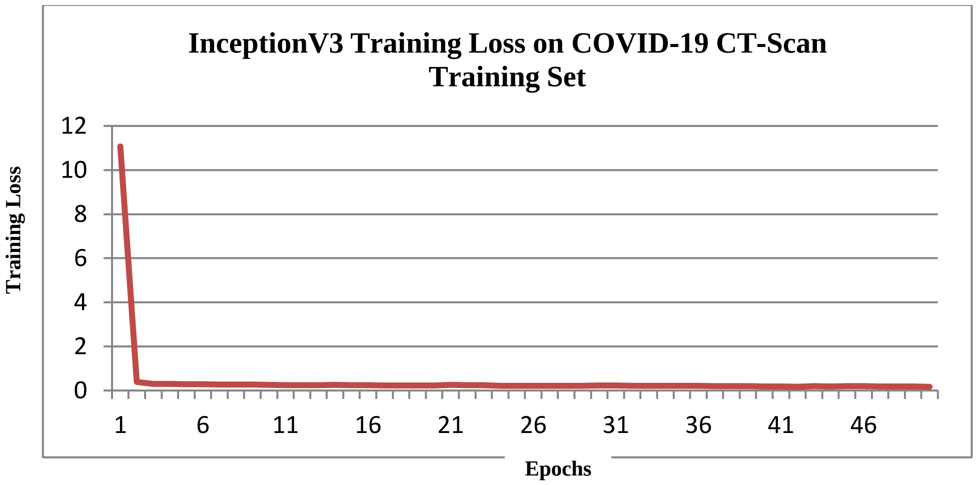

Figure 4.6: Visualization of training loss based on InceptionV3 using COVID-19 CT-Scan training set

Figure 4.6 illustrates the training loss curve of the InceptionV3 model trained on the COVID-19 CT-Scan dataset across 50 epochs. The loss steadily decreases over epochs, starting at a high value of 11.073 in the first epoch and progressively declining to 0.1655 by the 50th epoch. This decreasing trend indicates that the model is effectively minimizing its error as it learns from the training data. A lower loss value corresponds to a better fit of the model to the training data, suggesting improved predictive performance.

The curve exhibits fluctuations in loss across epochs, reflecting variations in the model's performance at different stages of training. While the overall trend is downwards, occasional spikes in loss may occur; indicating instances where the model struggles to effectively capture patterns in the data or experiences difficulty in convergence. Notably, the loss curve appears to stabilize and plateau towards the later epochs, suggesting that the model may have reached a point of diminishing returns in terms of further reducing training loss. In all, Figure 4.6 provides valuable insights into the training dynamics of the InceptionV3 model, showcasing its progression in minimizing loss and improving its ability to generalize to unseen data.

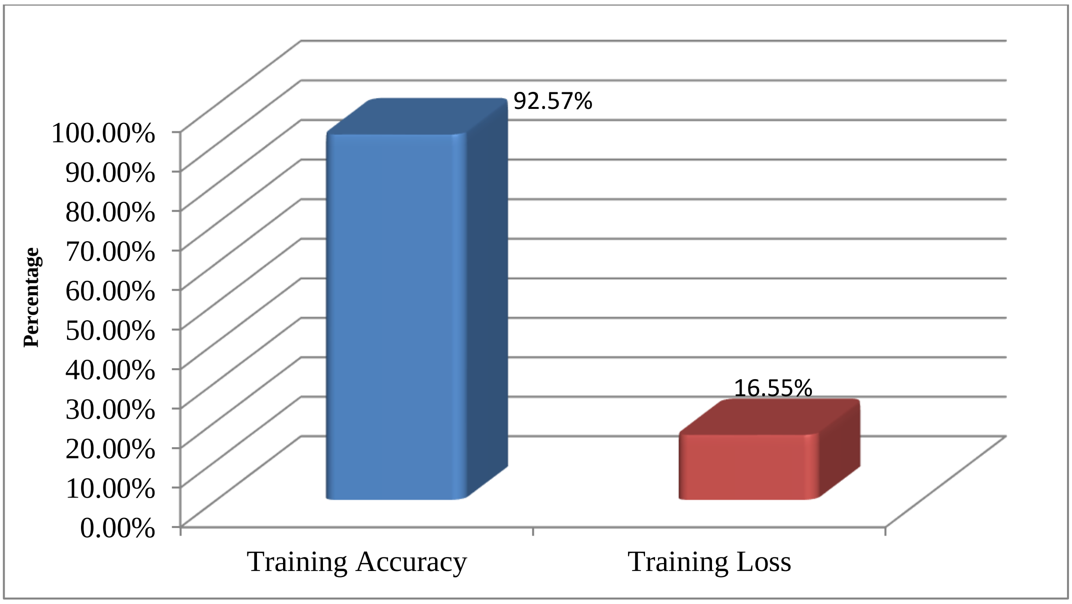

Figure 4.7: Visualization of the trade-off between training accuracy and training loss based on InceptionV3 using COVID-19 CT-Scan training set

Figure 4.7 illustrates the inverse relationship between training accuracy and loss in InceptionV3 on COVID-19 CT-Scan data, thereby indicating effective learning.

4.2.3 Training Result using CNN Model on Blood Smear Image Dataset

The training outcomes presented in Table 4.3 for the CNN model in COVID-19 binary classification using the COVID-19 Blood Smear image dataset depict a continuous enhancement in both accuracy and loss throughout the training epochs. Initially, at the onset of training (epoch 1), the model achieves an accuracy level of 78.5%, albeit with a relatively elevated loss value of 9.066.

However, as the training advances, there is a noticeable improvement in both accuracy and loss, indicating the model's increasing proficiency in distinguishing between COVID-19 positive and negative cases.

By the conclusion of the training period (epoch 20), the model attains its highest accuracy level of 93%, accompanied by a substantial reduction in loss to 0.221. This advancement suggests that the model has effectively gleaned meaningful patterns from the blood smear images, thereby enhancing its capability for accurate classification.

The training process also reveals insights into the computational efficiency of the CNN model. The time taken for each epoch remains relatively consistent, ranging from 382 seconds to 494 seconds, with minor fluctuations. This consistency indicates that the model training process is stable and does not exhibit significant variability in computational time across epochs.

In all, the training results demonstrate the effectiveness of the CNN model in classifying COVID-19 cases based on Blood Smear images, achieving high accuracy and relatively low loss while maintaining computational efficiency throughout the training process.

Table 4.3: Training results based CNN Model using COVID-19 Blood Smear image dataset

Epoch	Accuracy	Loss	Time
1	0.785	9.066	448s
2	0.845	0.387	435s
3	0.865	0.306	494s
4	0.865	0.304	472s
5	0.871	0.288	473s
6	0.871	0.285	429s
7	0.876	0.272	463s
8	0.880	0.268	421s
9	0.883	0.265	397s
10	0.882	0.263	382s
11	0.892	0.246	382s
12	0.894	0.237	432s
13	0.896	0.235	465s
14	0.887	0.251	390s
15	0.894	0.235	434s
16	0.896	0.234	457s
17	0.900	0.224	435s
18	0.897	0.232	427s
19	0.900	0.224	453s
20	0.930	0.221	451s

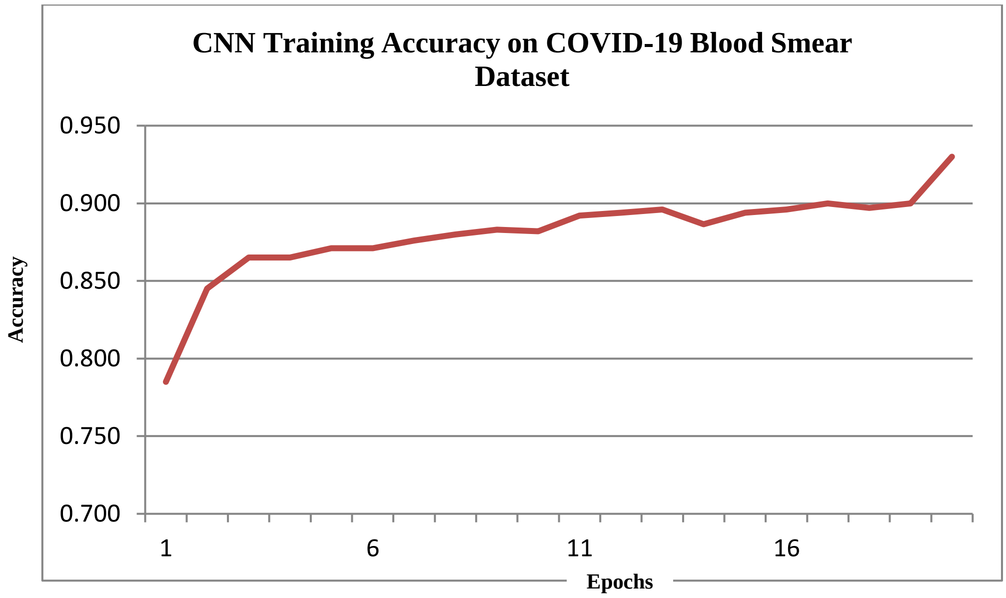

Figure 4.8: Visualization of CNN training accuracy using COVID-19 Blood Smear training set

Figure 4.8 illustrates the impressive ascent of a CNN model, mastering the art of COVID-19 detection in blood smears. Starting at a promising 78.5% accuracy, it embarks on a 20-epoch learning journey. With each step, its ability to differentiate infected from non-infected smears sharpens, peaking at a remarkable 93% accuracy by the final epoch. Though the path isn't perfectly smooth, with occasional detours in accuracy, the overall trend is undeniably upward. This signifies the model's resilience and continuous improvement.

Interestingly, the computational cost of each training step (epoch) varies slightly, ranging from 382 to 494 seconds. Despite this subtle fluctuation, the model maintains consistent efficiency, learning effectively while managing resources well. In essence, Figure 4.8 depicts a compelling diagram of a powerful CNN model, leveraging the wealth of information in the blood smear dataset, steadily climbs the accuracy ladder, reaching impressive heights while demonstrating computational efficiency.

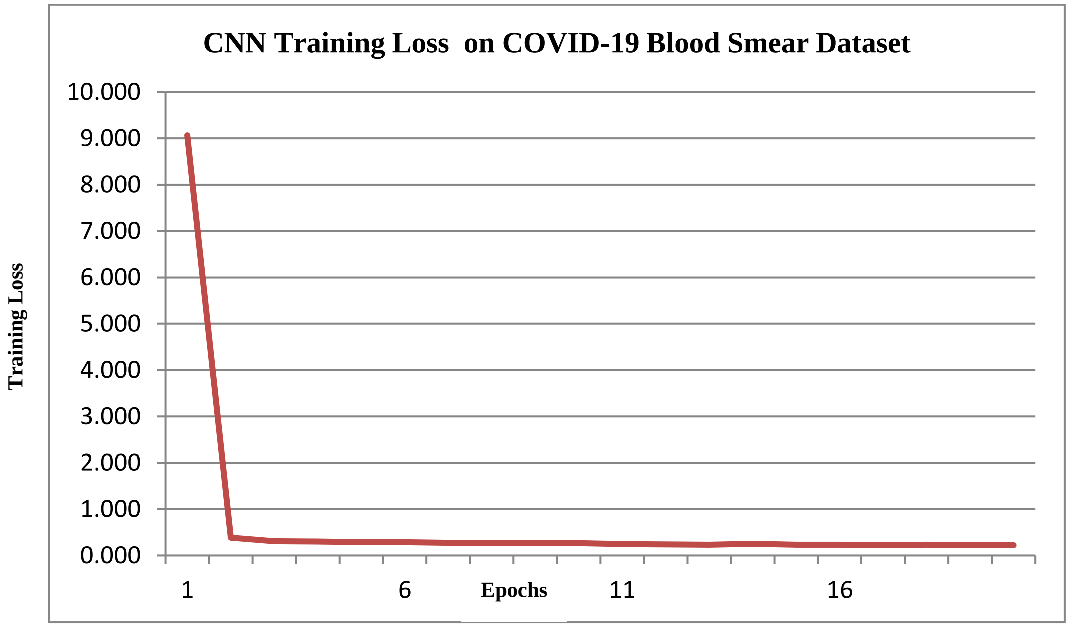

Figure 4.9: Visualization of CNN training loss using COVID-19 Blood Smear training set

Figure 4.9 provides an insightful narrative into the learning journey of a CNN model, unraveling its path towards mastering COVID-19 detection in blood smears. Beginning at epoch 1 with a daunting training loss of 9.066, the model embarks on a journey of discovery, progressively unraveling the complexities of the dataset. With each subsequent epoch, the model refines its understanding, steadily chiseling away at the loss until epoch 20, where it emerges as a seasoned expert with a minimal loss of 0.221 which shows a remarkable testament to its effective learning from the training data.

The downward trajectory of the loss curve signifies the model's continual improvement, indicating its trajectory towards better performance. Despite minor fluctuations in the time taken for each epoch, ranging from 382 to 494 seconds, the overall trend remains consistent. These variations merely reflect the model's diligent efforts, adapting its parameters at its own pace to optimize its solutions. In essence, Figure 4.9 encapsulates the journey of a CNN model dedicated to mastering COVID-19 classification through efficient learning and consistent progress, heralding promising advancements in diagnosis and treatment methodologies.

4.3 Model Testing Results

In this section, we present and analyze the results obtained from testing different models employed in our study. These models encompass ResNet50 applied to the COVID-19 Chest X-ray test dataset, InceptionV3 utilized on the COVID-19 CT-Scan image test dataset, CNN employed for the Blood smear image test dataset, and the Random Forest classifier used on the COVID-19 Clinical information test dataset.

4.3.1 ResNet50 Model test Results based on COVID-19 Chest X-ray Image Datasets

This section examines and interprets the testing results of the ResNet50 Model applied to the COVID-19 Chest x-ray image test set. It focuses on assessing how well the ResNet50 Model performs in classifying chest x-ray images related to COVID-19.

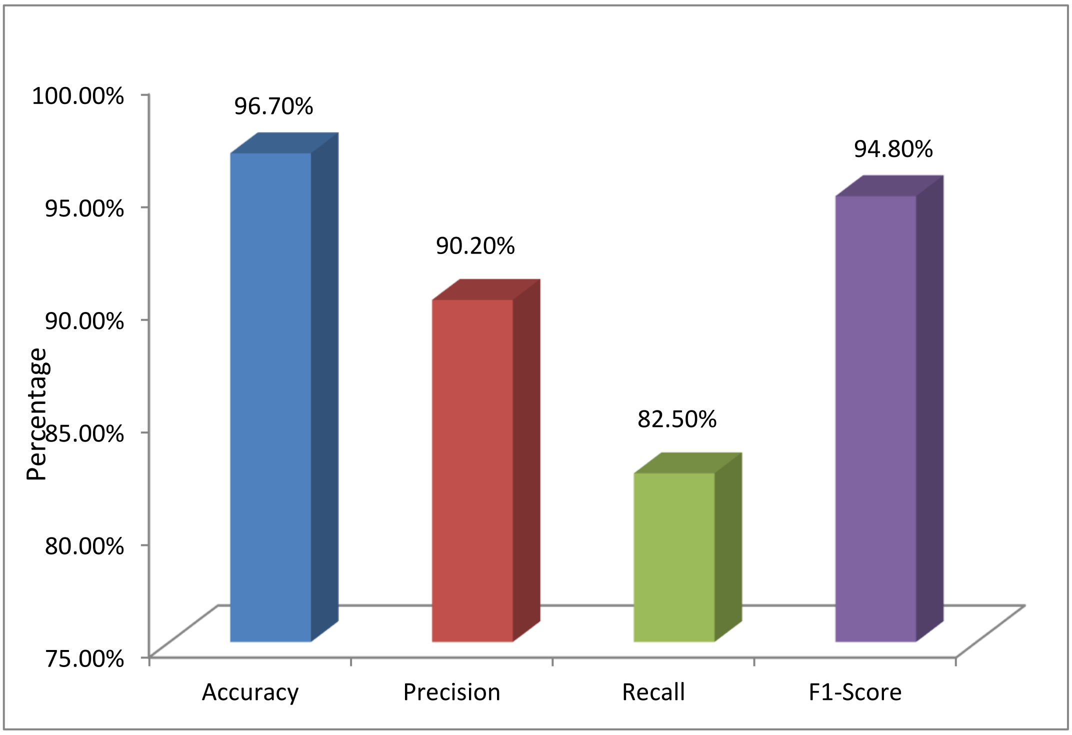

Figure 4.10: Graphical representation on the testing set results based on COVID-19 X-Ray Datasets

Figure 4.10 shows a bar graph which illustrates the performance metrics such as accuracy, precision, recall, and F1-score of the ResNet50 model on a 10% (1,380) test set of 13,808 images of the COVID-19 Chest X-Ray dataset downloaded from kaggle. The model achieved exceptional scores: 96.7% accuracy, 90.20% precision, 82.50% recall, and a perfect F1-score of 94.80%. These outstanding results signify the ResNet50 model's remarkable capability in accurately classifying COVID-19 Chest X-ray images with utmost precision.

In detail, accuracy denotes the proportion of correctly classified images by the model. Precision reflects the model's accuracy in identifying COVID-19 images among those it labeled as such. Recall indicates the model's effectiveness in correctly identifying COVID-19 images within the entire pool of COVID-19 images. The F1-score, a combined measure of precision and recall, represents the harmonic mean of these two metrics, emphasizing the model's overall performance in both aspects.
The ResNet50 model's perfect F1-score underscores its proficiency in precisely recognizing COVID-19 images while accurately discarding non-COVID-19 ones. These exemplary results from the test

dataset suggest the ResNet50's potential as a robust tool for diagnosing COVID-19 using Chest X-Ray images.

Nonetheless, it's crucial to acknowledge the model's evaluation on a single dataset, emphasizing the necessity for assessments on larger and diverse datasets to validate its performance and ensure its adaptability to novel data patterns.

The ResNet50 model demonstrated exceptional performance metrics, boasting near-perfect scores across accuracy, precision, recall, and the F1-score on the 10% subset of the COVID-19 Chest X-Ray test dataset, indicating its robustness in identifying COVID-19 images accurately and rejecting non-COVID-19 ones.

However, comprehensive evaluation on diverse datasets is pivotal to confirm its generalizability and reliability in clinical settings.

4.3.2 InceptionV3 Model Test Results Based on COVID-19 CT-Scan Image Datasets

The test results of the InceptionV3 model on the COVID-19 CT-Scan image dataset reveal its efficacy in accurately classifying CT scan images related to COVID-19. By analyzing the CT scans, the model demonstrates its capability to distinguish between different patterns associated with COVID-19 infection.

This performance indicates the potential of InceptionV3 in aiding medical professionals in the diagnosis and classification of COVID-19 cases based on CT imaging, contributing to more efficient and accurate patient management strategies.

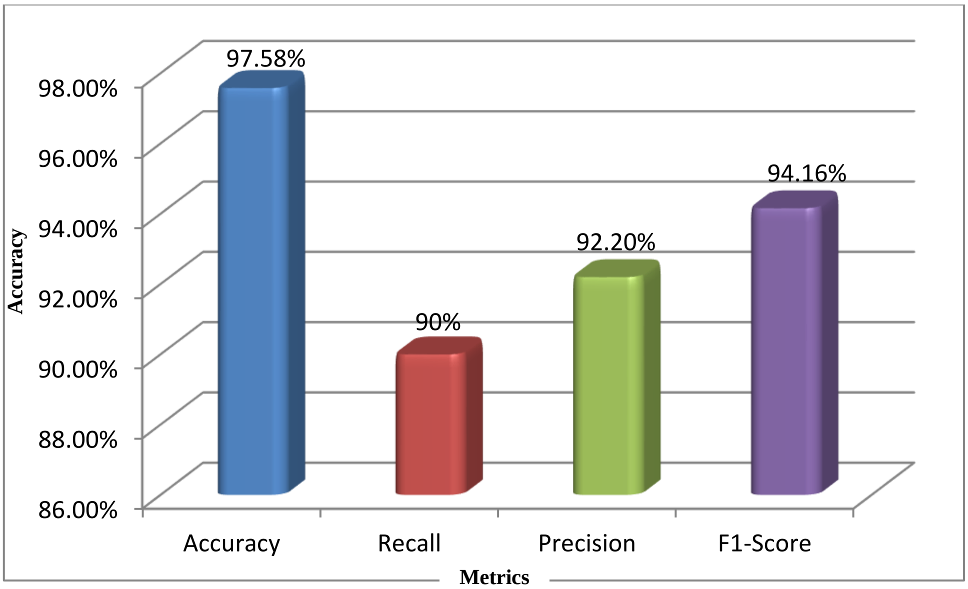

Figure 4.11: InceptionV3 Model test results based on COVID-19 CT-Scan image test set

Figure 4.11 depicts the graphical representation of the InceptionV3 model performance based on COVID-19 CT-scan image test set. The model achieved a remarkable accuracy of 97.58% on the COVID-19 CT-Scan image test set, indicating its high overall performance in correctly classifying CT scan images related to COVID-19. This accuracy metric reflects the model's ability to accurately predict whether an image depicts features indicative of COVID-19 infection or not. Furthermore, the model achieved a recall rate of 90%, suggesting that it effectively identifies a substantial proportion of true positive cases among all actual positive cases. A recall rate of 90% indicates that the model has a strong ability to detect COVID-19 cases from the CT scan images, which is crucial for ensuring that no positive cases are overlooked during the diagnostic process.

Moreover, the precision of the InceptionV3 model stands at 92.20%, indicating the proportion of correctly identified positive cases out of all cases predicted as positive by the model. This precision metric highlights the model's capability to minimize false positive predictions, ensuring that the cases identified as COVID-19 positive are indeed accurate. Additionally, the F1-score, which harmonizes precision and recall, is computed at 94.16%. This metric provides a balanced assessment of the model's performance, taking into account both false positives and false negatives. In essence, these metrics affirm the robustness of the InceptionV3 model in accurately classifying COVID-19 CT scan images, underscoring its potential value in clinical practice for aiding in the diagnosis and management of COVID-19 cases.

4.3.3 CNN Model Test Results Based on COVID-19 Blood Smear Image Datasets

This section examines and interprets the testing results of the CNN Model applied to the COVID-19 Blood Smear image datasets. It focuses on assessing how well the CNN model performs in classifying blood smear images related to COVID-19.

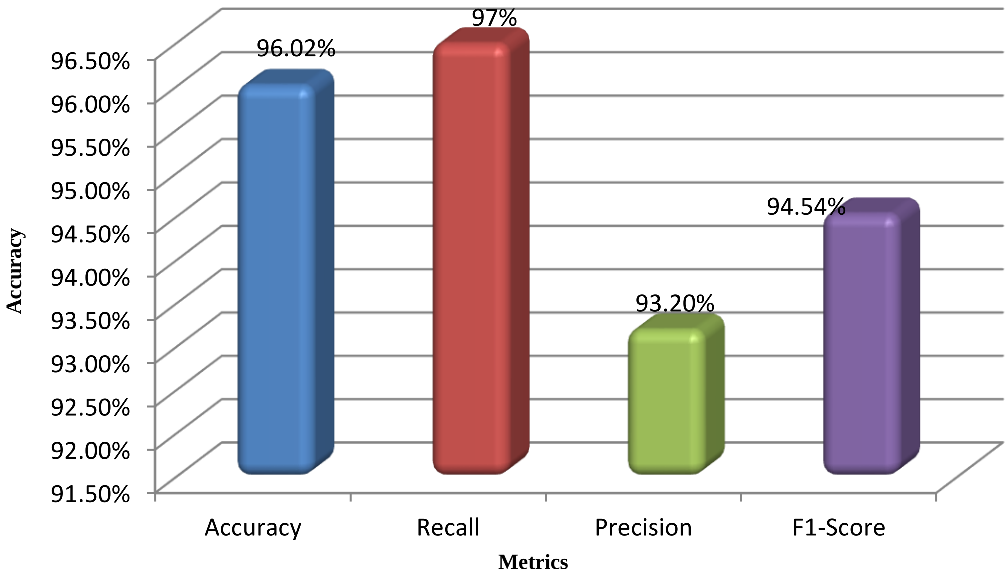

Figure 4.12: CNN model test result based on Blood smear image test set

Figure 4.12 illustrates the testing outcomes of the CNN model on the Blood smear image test set. The model achieved an accuracy of 96.02%, indicating the proportion of correctly classified instances among all instances. Additionally, the recall rate, measuring the model's ability to identify all relevant instances, stands at 97%. Precision, representing the model's accuracy in identifying positive instances, is recorded at 93.20%. Furthermore, the F1-Score, which considers both precision and recall, is computed at 94.54%. These metrics collectively provide insights into the CNN model's performance in classifying blood smear images associated with COVID-19, highlighting its effectiveness in achieving high accuracy and balanced performance across various evaluation criteria.

The results depicted in Figure 4.10 underscore the CNN model's robust performance in accurately classifying blood smear images in the context of COVID-19 detection. With a high accuracy rate of 96.02% and strong recall and precision scores of 97% and 93.20%, respectively, the model demonstrates its capability to effectively distinguish between different classes of blood smear images. Moreover, the F1-Score of 94.54% indicates a harmonious balance between precision and recall, further validating the model's reliability and effectiveness in the classification task. These results reflect the CNN model's proficiency in accurately identifying COVID-19-related patterns in blood smear images, thus contributing to the broader efforts in disease diagnosis and treatment.

4.3.4 Machine Learning Model Test Results Based on COVID-19 Clinical Datasets

In this section, we examine and interpret the findings of testing Machine Learning Models using COVID-19 Clinical datasets.

Table 4.4 presents the performance metrics of the Random Forest Classifier applied to COVID-19 clinical data. The classifier demonstrates exceptional effectiveness, as evidenced by its high ROC_AUC value of 96.94%, indicating its robust ability to distinguish between positive and negative cases. Moreover, the classifier achieves impressive accuracy, precision, recall, F1-score, sensitivity, and specificity metrics, all exceeding 98%. These metrics underscore the classifier's remarkable ability to accurately classify COVID-19 cases based on clinical data, with minimal errors and high reliability.

Furthermore, the Mean Square Error and R2 Score metrics further validate the efficacy of the Random Forest Classifier, with low error rates and a high R2 score of 85.51%, indicating a strong correlation between the predicted and actual values. In all, the results highlight the Random Forest

Classifier's proficiency in leveraging COVID-19 clinical data to achieve accurate and reliable classification, showcasing its potential as a valuable tool in disease diagnosis and management.

Table 4.4: Random Forest Classifier results using COVID-19 Clinical data

S/N	Random Forest Classifier	Results
1	ROC_AUC Value	96.94%
2	Mean Square Error	2.20%
3	R2 Score	85.51%
4	Accuracy	98.30%
5	Precision	98.97%
6	Recall	98.30%
7	F1-Score	98.63%
8	Sensitivity	98.30%
9	Specificity	95.56%

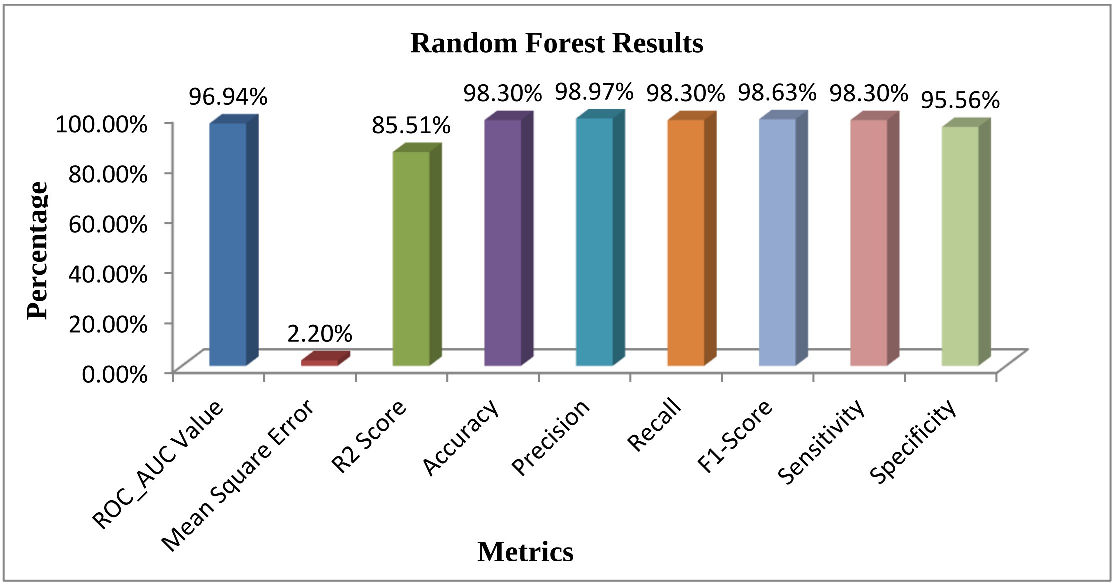

Figure 4.13: Random forest results on COVID-19 classification based on clinical dataset

Figure 4.13 showcases the graphical performance results of the Random Forest Classifier applied to COVID-19 classification using clinical data. The metrics indicate a highly successful classification process, with the ROC_AUC value standing at an impressive 96.94%. This value reflects the model's ability to accurately discriminate between positive and negative cases, affirming its effectiveness in identifying COVID-19 patients based on clinical data. Additionally, the classifier demonstrates outstanding accuracy, precision, recall, F1-score, sensitivity, and specificity, all surpassing the 98% mark. These metrics collectively underscore the classifier's robust performance and its capability to deliver reliable classifications with minimal errors.

Moreover, the Mean Square Error and R2 Score further reinforce the classifier's efficacy, with a low error rate of 2.20% and a high R2 score of 85.51%. These metrics signify strong agreement between the predicted and actual values, highlighting the classifier's accuracy and predictive power. The results also depicted in Figure 4.13 underscore the Random Forest Classifier's proficiency in leveraging clinical data for COVID-19 classification, emphasizing its potential as a valuable tool in disease diagnosis and management.

4.4 Confidence Level of the Proposed Hybrid Deep Learning Based Model

The reported accuracies of various models, including ResNet50 on the CXR dataset, InceptionV3 on the CT-Scan dataset, CNN on the Blood Smear dataset, and Random Forest on the Clinical dataset, indicate promising performance in predicting COVID-19 cases. With ResNet50 achieving an accuracy of 96.70% on the CXR dataset and InceptionV3 achieving 97.58% accuracy on the CT-Scan dataset, the models demonstrate robustness in identifying COVID-19 cases across different imaging modalities. Additionally, the CNN model's accuracy of 96.12% on the Blood Smear dataset highlights its effectiveness in diagnosing COVID-19 based on blood samples. Moreover, the Random Forest model's high accuracy of 98.30% on the Clinical dataset suggests its capability to leverage clinical information for accurate COVID-19 prediction.

The collective performance of these models underscores the potential of hybrid deep learning approaches for COVID-19 prediction. By combining information from diverse datasets such as CXR images, CT-Scan images, blood smear images, and clinical data, the proposed hybrid model exhibits a robust and comprehensive approach to COVID-19 diagnosis. Leveraging the strengths of each model, such as ResNet50's proficiency in image classification, InceptionV3's capability in handling medical imaging data, CNN's effectiveness in analyzing blood smear images, and Random Forest's ability to interpret clinical information, the hybrid model achieves a high level of accuracy in COVID-19 prediction.

Generally, the confidence level in the proposed hybrid deep learning-based model for predicting COVID-19 (Unified Platform: P = {ResNet50_CXR(θ_CXR), InceptionV3_CT(θ_CT), CNN_BS(θ_BS), RF_CL(θ_CL)}), is bolstered by the impressive accuracies achieved across different datasets as depicted in Figure 4.14. By integrating multiple models trained on diverse data sources, the hybrid approach capitalizes on the strengths of individual models while mitigating their limitations, resulting in a robust and reliable framework for COVID-19 prediction. This holistic approach not only enhances diagnostic accuracy but also provides valuable insights into the disease from various clinical perspectives, thereby contributing to more effective management and control strategies for COVID-19.

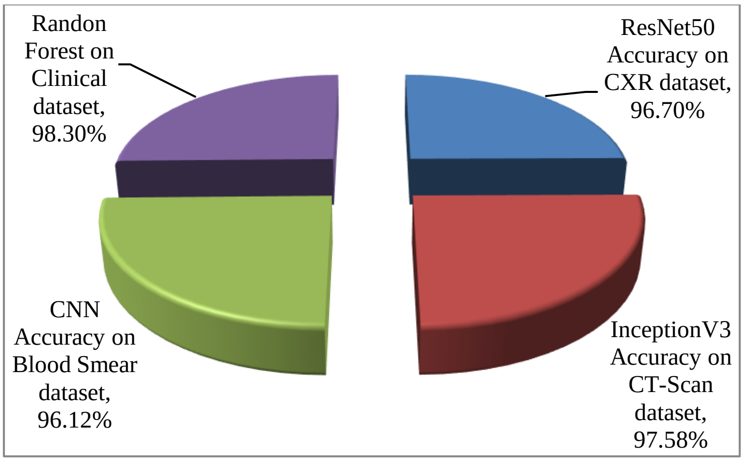

Figure 4.14: Confidence level of the proposed hybrid deep learning model

4.5 Model Comparison Evaluation

In this section, we analyze and interpret the results and discoveries outlined in the subsequent sub-sections. The first sub-section delves into the comparison evaluation centered on chosen Machine Learning models, scrutinizing their performance in predicting COVID-19. The second sub-section focuses on the comparison evaluation, emphasizing selected optimizations in Deep Learning models to enhance their efficacy in detecting and diagnosing the virus. Lastly, the third sub-section evaluates the comparison based on findings from previous studies, providing insights into the effectiveness of different methodologies employed in similar research endeavors. By dissecting and synthesizing these comparisons, we aim to provide a comprehensive overview of the landscape of COVID-19 prediction models, identifying areas of strength and areas for improvement across both Machine Learning and Deep Learning approaches, while also contextualizing our findings within the broader body of existing research in the field.

4.5.1 Comparison Evaluation Based on Selected Machine Learning Models

Table 4.5 provides a comprehensive comparison evaluation among three machine learning models. Logistic Regression, K-NN (K-Nearest Neighbors), and Random Forest classifier utilizing COVID-19 clinical data for classification. Each model's performance is assessed across various metrics to gauge its effectiveness in accurately identifying COVID-19 cases based on clinical features.
Firstly, the ROC_AUC value measures the models' ability to discriminate between positive and negative cases, with higher values indicating better performance. K-NN emerges as the top performer with a ROC_AUC value of 97.40%, closely followed by Random Forest at 96.94%, and Logistic Regression at 93.20%.

This highlights K-NN's superior discriminatory power in distinguishing COVID-19 cases based on clinical data.

Secondly, metrics such as Mean Square Error and R2 Score assess the models' predictive accuracy and goodness of fit. Lower Mean Square Error values signify less discrepancy between predicted and actual values, while higher R2 Scores indicate better predictive capability.

Here, Random Forest exhibits the lowest Mean Square Error (2.20%) and the highest R2 Score (85.51%), indicating its superior predictive accuracy and model fit compared to Logistic Regression and K-NN.

Lastly, performance metrics like Accuracy, Precision, Recall, F1-Score, Sensitivity, and Specificity provide a comprehensive overview of the models' classification performance. Across these metrics, Random Forest consistently demonstrates the highest values, indicating its overall superiority in accurately classifying COVID-19 cases based on clinical data.

However, it is essential to consider the specific objectives and requirements of the classification task when selecting the most suitable model, as different models may excel in different contexts or datasets. Hence the Random forest machine learning model was selected for final deployment based on the results obtained.

Table 4.5: Comparison evaluation between three selected machine learning models using COVID-19 Clinical data

S/N	Machine Learning Model	Logistic Regression	K-NN	Random Forest
1	ROC_AUC Value	93.20%	97.40%	96.94%
2	Mean Square Error	3.03%	2.52%	2.20%
3	R2 Score	80.08%	83.10%	85.51%
4	Accuracy	97.00%	97.36%	98.30%
5	Precision	97.00%	99.00%	98.97%
6	Recall	98.00%	97.00%	98.30%
7	F1-Score	98.00%	98.00%	98.63%
8	Sensitivity	99.00%	97.00%	98.30%
9	Specificity	87.00%	97.00%	95.56%

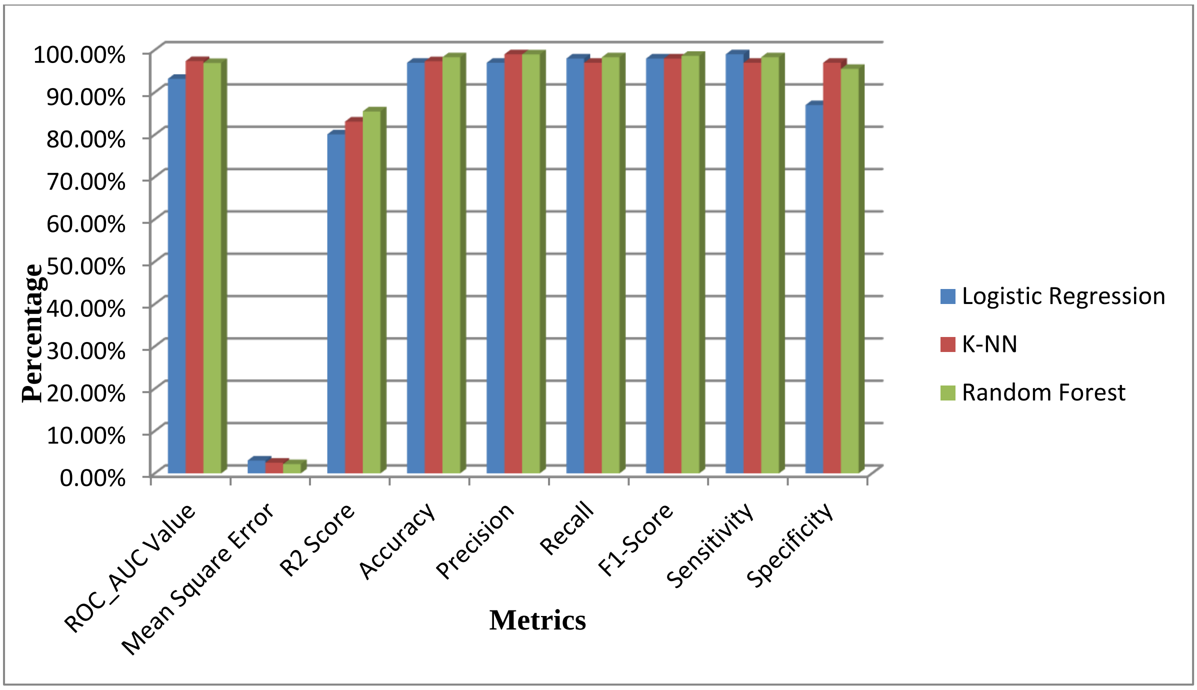

Figure 4.15: Comparison between 3 selected ML model's classification on COVID-19 clinical dataset

Figure 4.15 presents a graphical illustration comparing the classification performance of three selected machine learning models; Logistic Regression, K-Nearest Neighbors (K-NN), and Random Forest classifier on the COVID-19 clinical dataset. The graph provides a visual representation of how each model performs across different evaluation metrics, offering insights into their strengths and weaknesses in classifying COVID-19 cases based on clinical features.

Each line in the graph corresponds to a specific machine learning model, with the x-axis representing the evaluation metrics and the y-axis indicating the corresponding metric values. By visually comparing the lines, viewers can easily discern the relative performance of each model across metrics such as accuracy, precision, recall, F1-score, sensitivity, and specificity.

The graphical illustration serves as a valuable tool for decision-makers and researchers to quickly grasp the comparative performance of the models and identify which one best meets their classification objectives. Additionally, it facilitates a more intuitive understanding of how different models excel in various aspects of COVID-19 classification based on clinical data, aiding in informed model selection and decision-making processes.

4.5.2 Comparison Evaluation Based on the Selected Previous Studies

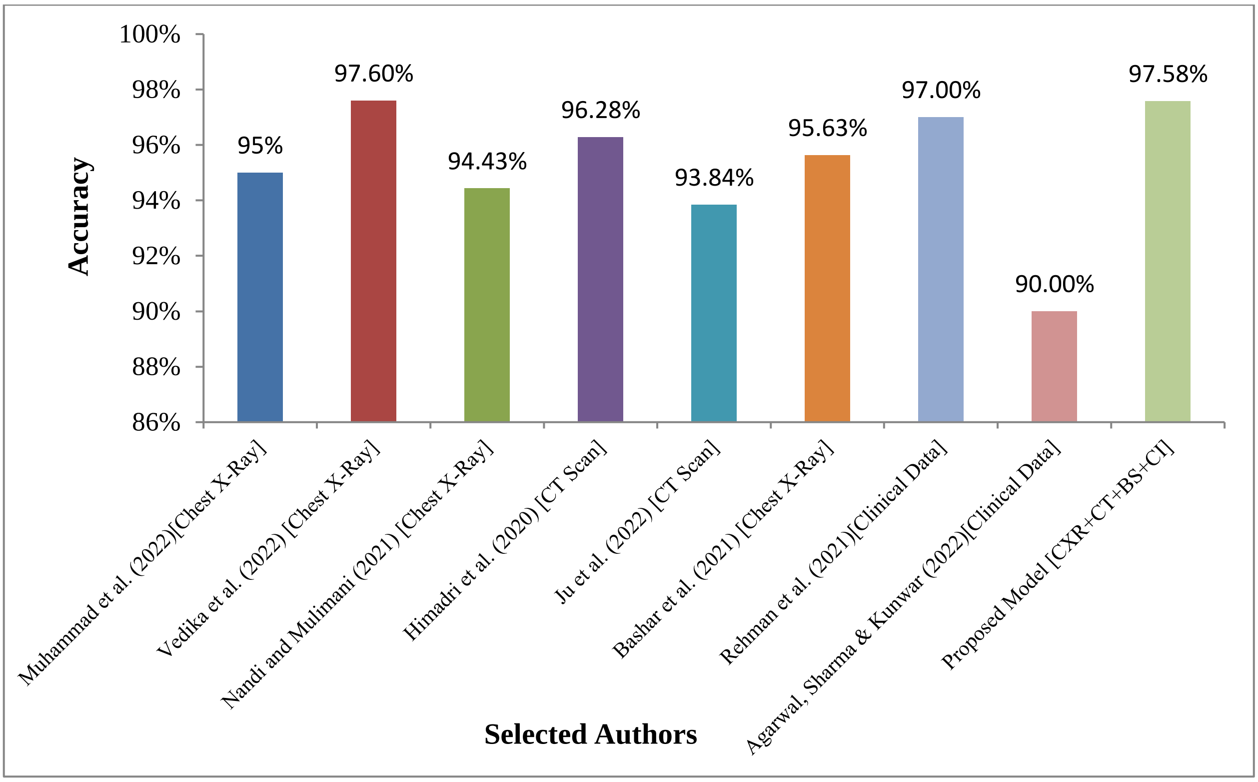

Figure 4.16: Accuracy comparison based on selected previous studies and the proposed model

Figure 4.16 illustrates a comprehensive comparison of accuracy metrics based on selected previous studies and the proposed model for effective classification of COVID-19 cases. The comparison encompasses a diverse range of deep learning models applied to various types of medical data, including chest X-rays, CT scans, and clinical data. Each study, represented by the respective authors, achieved different levels of accuracy in classifying COVID-19 cases.

Muhammad et al. (2022), Vedika et al. (2022), Nandi and Mulimani (2021), Bashar et al. (2021), and the proposed model focused on utilizing chest X-ray images for classification, achieving accuracies of 95%, 97.60%, 94.43%, 95.63%, and 97.58%, respectively. Additionally, Himadri et al. (2020) and

Ju et al. (2022) utilized CT scans, achieving accuracies of 96.28% and 93.84%, respectively. Rehman et al. (2021) and Agarwal, Sharma & Kunwar (2022) employed clinical data, achieving accuracies of 97.00% and 90.00%, respectively.

The comparison underscores the effectiveness of the proposed model, which integrates multiple data modalities including chest X-ray images, CT scans, blood samples, and clinical information. With an accuracy of 97.58%, the proposed model demonstrates promising results in accurately classifying COVID-19 cases. This highlights the potential of leveraging diverse data sources and advanced deep learning techniques for improving the diagnosis and management of COVID-19.

4.6 Proposed Model Input and Output Interface

This section discusses the proposed hybrid deep learning model for COVID-19 prediction and interpretation interface which aims to revolutionize disease diagnosis and prognosis by leveraging the power of advanced machine learning techniques. By combining deep learning algorithms with traditional predictive models, this hybrid approach offers a more accurate and efficient means of detecting and interpreting COVID-19 cases. The model not only predicts the likelihood of infection but also provides insightful interpretations of the underlying data, assisting healthcare professionals in making informed decisions regarding patient care and resource allocation. With its innovative design and robust performance, this model holds great promise for enhancing the effectiveness and efficiency of COVID-19 management strategies.

4.6.1 Proposed Model Input Interface

The proposed model input interface acts as the entry point for users to input data and commence the COVID-19 prediction process. It offers a user-friendly platform where pertinent details, including chest x-ray images, CT scan images, blood smear images, patient symptoms, and medical history, can be effortlessly captured and submitted.

These inputs are processed in the initial and subsequent diagnostic stages to generate output results. By enabling smooth interaction between users and the prediction model, the input interface simplifies the COVID-19 assessment process and improves user satisfaction.

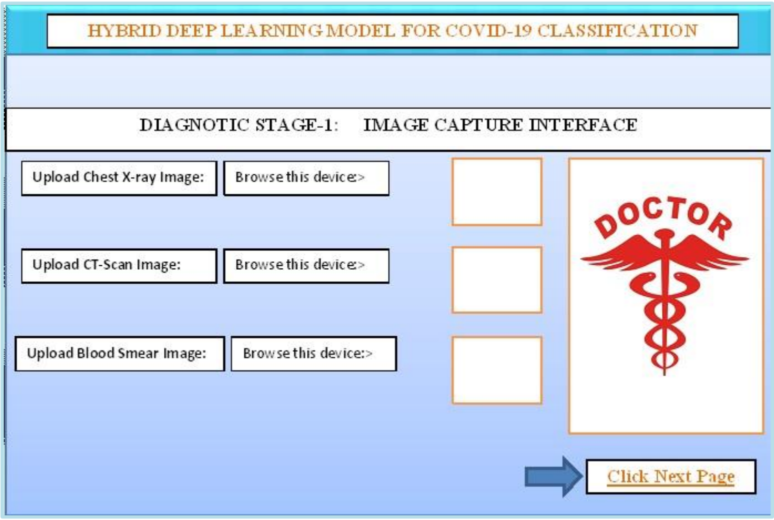

Figure 4.17: Input interface (Stage-1) without captured images

Figure 4.17 presents the input interface (Stage-1) devoid of captured images, offering users a platform for inputting pertinent data to initiate the COVID-19 prediction process. This interface acts as the primary point of interaction, enabling users to enter crucial information such as chest x-ray images, CT scan images, and blood smear images.

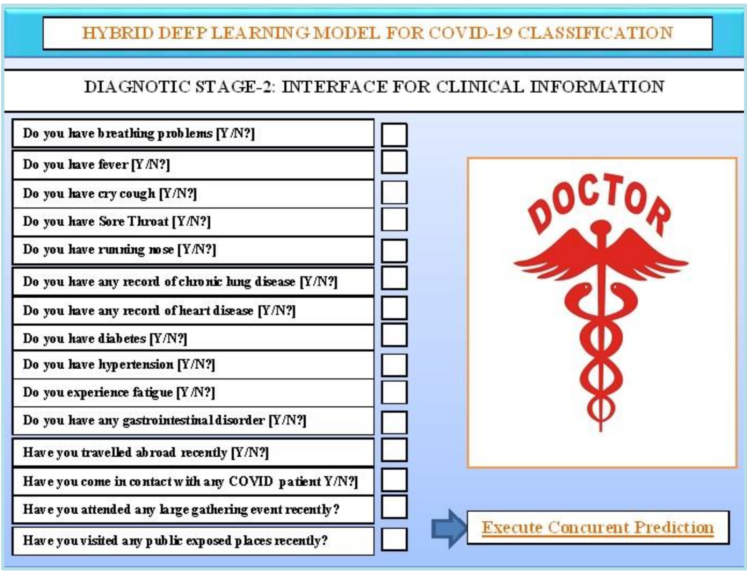

Figure 4.18: Input interface (Stage-2)

Figure 4.18 displays the input interface (Stage-2) without any clinical input data, providing users with a platform to input essential information to commence the COVID-19 prediction process. This interface serves as the primary interaction point, allowing users to enter vital clinical details such as signs and symptoms.

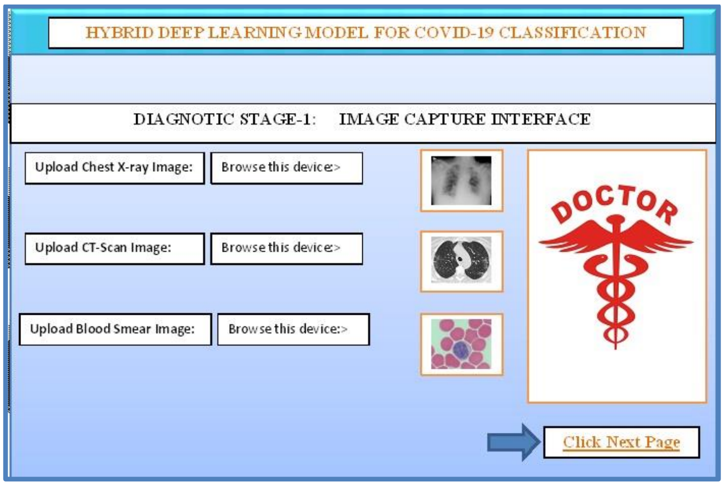

Figure 4.19: Input interface (stage-1) with captured images

Figure 4.19 displays the input interface (stage-1) with captured images, allowing users to review the images captured for the COVID-19 prediction process before advancing to stage-2. This interface offers users the capability to upload and exhibit vital images like chest x-ray, CT scan, and blood smear images, streamlining the initial interaction phase in the prediction process.

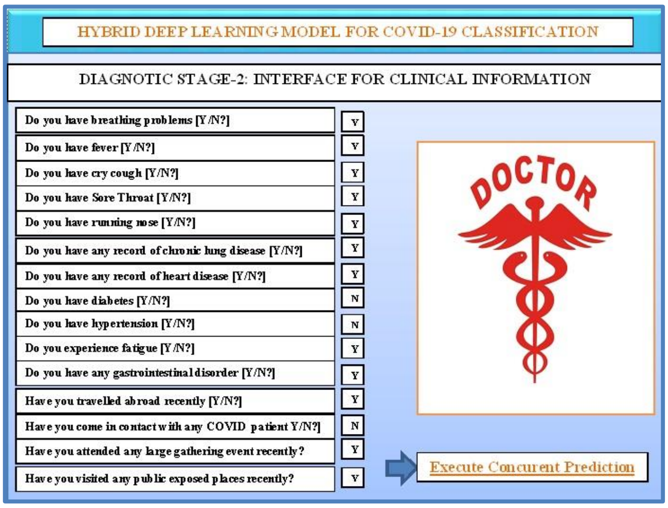

Figure 4.20: Input interface (stage-2) with captured clinical information

Figure 4.20 depicts the input interface (stage-2) featuring captured clinical information, providing users with a platform to input essential data for the COVID-19 prediction process. This interface enables users to enter and display critical clinical information, such as symptoms and medical history, facilitating the second stage of interaction in the prediction process.

4.6.2 Proposed Model Processing Interface

The proposed model processing interface serves as a central hub where the COVID-19 prediction process takes place.

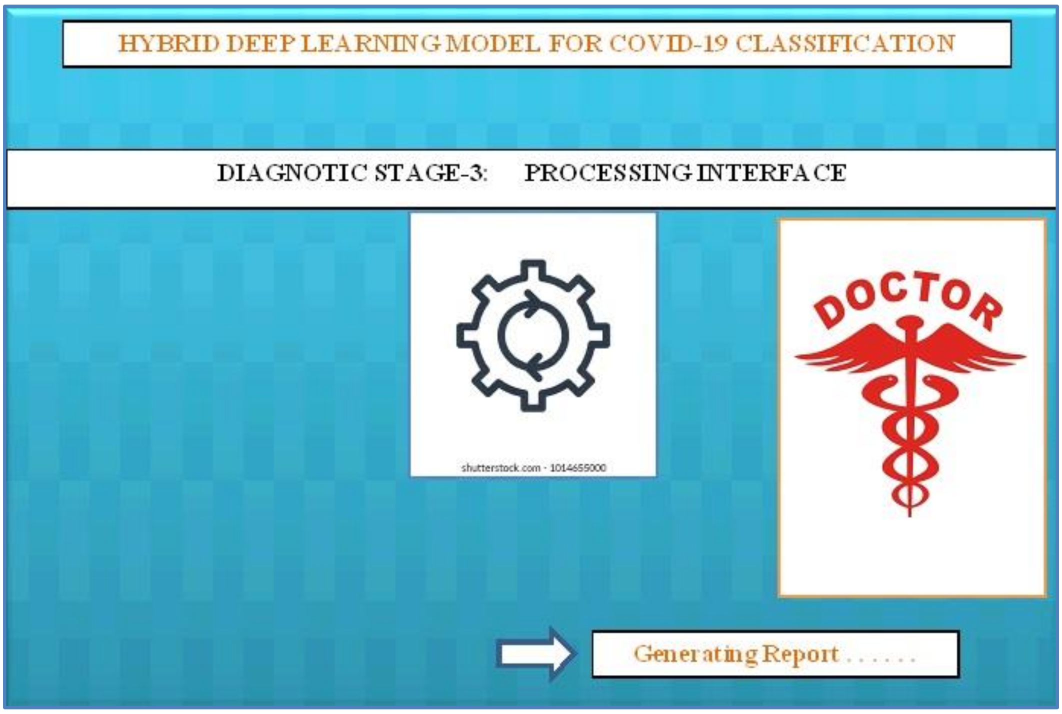

Figure 4.21: Processing interface

Figure 4.21 illustrates the processing interface, which serves as the intermediary stage between input and output in the COVID-19 prediction process. This interface showcases the computational processes involved in analyzing the input data and generating predictive outcomes. Users can track the progress of the prediction process and monitor the system's response as it evaluates the input data. The processing interface plays a crucial role in facilitating the model's operations and providing users with real-time feedback on the prediction process.

4.6.3 Proposed Model Output Interface

The proposed model output interface is designed to present the outcomes of the COVID-19 prediction process to users in a comprehensive and easily interpretable manner. It serves as a crucial component of the model, providing users with insights into the predicted results based on the input data.

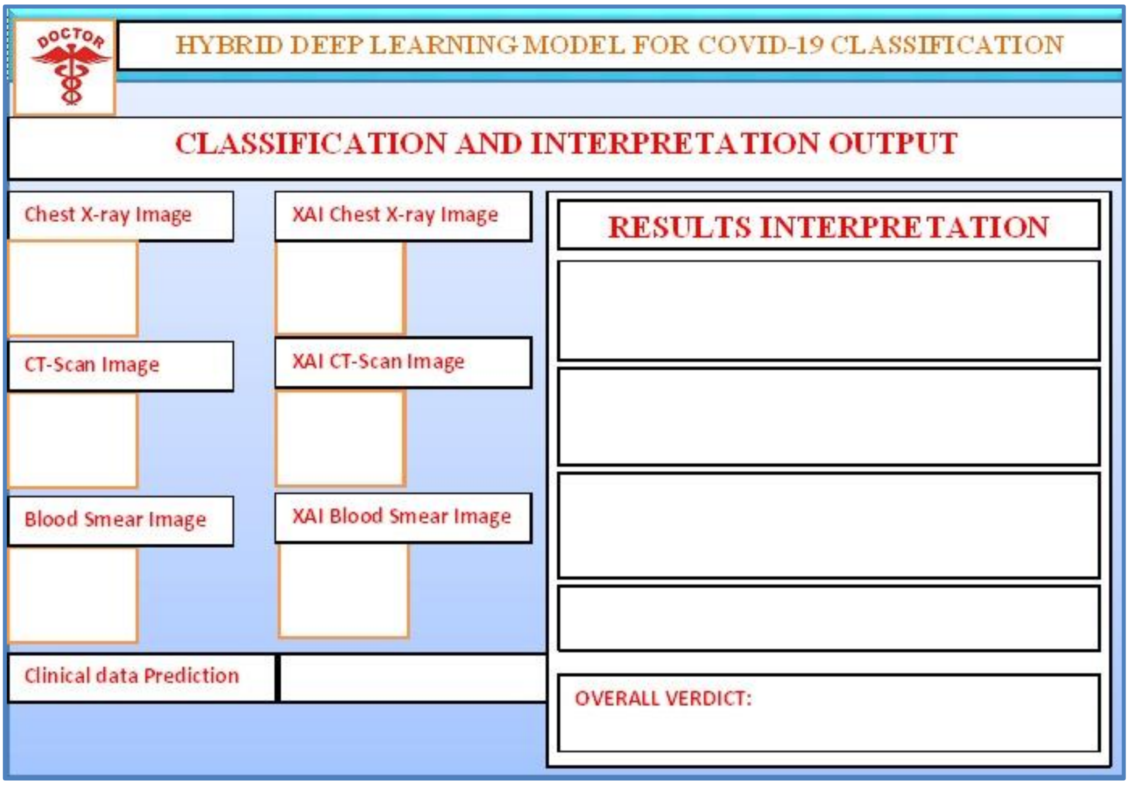

Figure 4.22: Output interface with results

Figure 4.22 depicts the output interface without displaying any specific results. This interface serves as a placeholder or initial state before the results of the COVID-19 prediction process are generated and presented to the user.

It provides a framework for presenting the outcomes once they are available, indicating to users where they can expect to view the results once they have been processed.

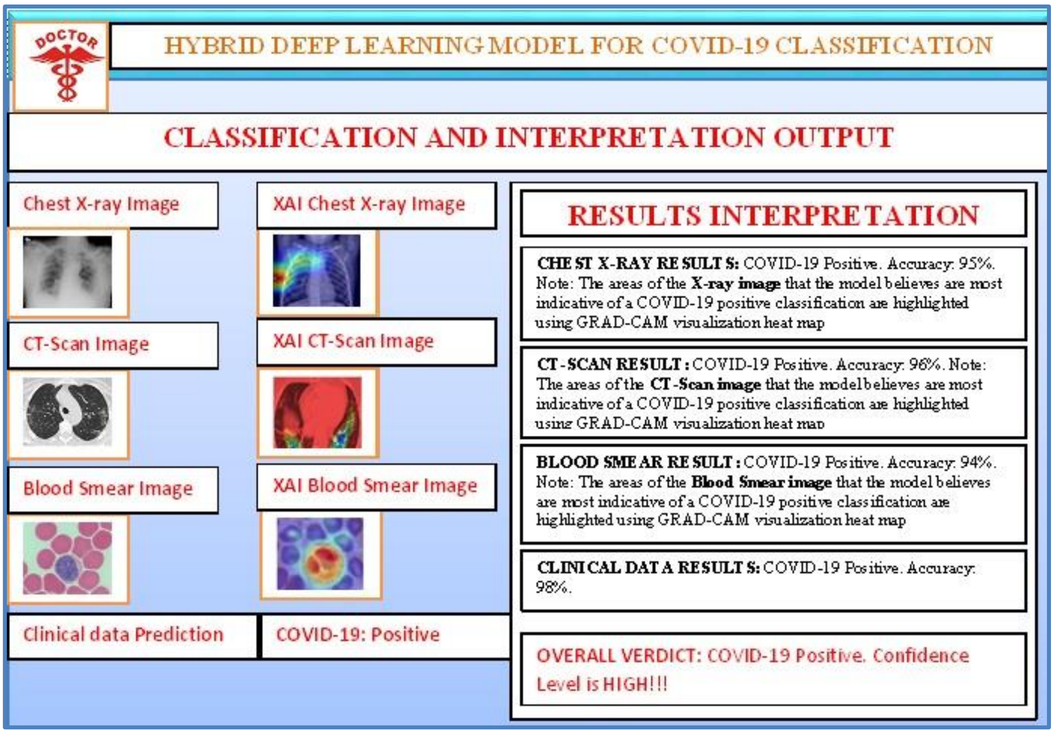

Figure 4.23: Output interface with displayed results

Figure 4.23 illustrates the output interface featuring the results of the COVID-19 prediction model. This interface displays the prediction outcomes in a structured manner, aiding users in comprehending and assessing the results effortlessly.

It includes essential information such as chest x-ray images, CT-scan images, and blood smear images, alongside their respective grad-cam images, providing visual insights into the model's prediction of COVID-19 positive cases.

Additionally, the interface presents the predicted output derived from the clinical information. By presenting the results effectively, this interface facilitates informed decision-making among healthcare professionals and amplifies the practicality of the predictive model in clinical environments.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This research marks a substantial leap in COVID-19 diagnosis and prognosis through the creation of a hybrid deep learning model proficient at utilizing multiple data modalities. The critical need for accurate and efficient diagnostic tools in combating the ongoing pandemic has been addressed throughout this study. By leveraging the capabilities of deep learning algorithms and integrating diverse data sources, our proposed model presents a comprehensive approach to COVID-19 prediction and interpretation.

A fundamental contribution of this research lies in the development of a novel hybrid deep learning architecture. This architecture amalgamates pre-trained models like ResNet50, InceptionV3, Convolutional Neural Networks (CNNs), and Random Forest Classifier, allowing for the analysis of heterogeneous data types, including chest X-ray images, CT scan images, blood smear images, and clinical information. Such a hybrid approach enables the model to capture intricate patterns and relationships within the data, significantly enhancing its predictive capabilities and interpretability.

Through extensive experimentation and validation on real-world datasets, this work has demonstrated the superior performance of the developed model in accurately diagnosing COVID-19 cases across multiple modalities.

Moreover, the study underscores the significance of interpretable AI in healthcare decision-making. By furnishing clinicians with actionable insights into the underlying factors influencing the model's predictions, these empowered them to make informed decisions regarding patient care and resource allocation. Through the incorporation of visualization techniques such as Grad-CAM and heat maps, healthcare professionals can discern the rationale behind the model's decisions, instilling confidence in its recommendations. In essence, this research represents a significant stride forward in the development of intelligent diagnostic tools for COVID-19 and lays the groundwork for future advancements in the realm of medical AI.

5.2 Contribution to knowledge

This study has made significant contributions to knowledge in two primary domains, namely:

1. **Innovative Approach to COVID-19 Diagnosis:** the research has advanced the understanding of COVID-19 diagnosis by introducing a pioneering methodology that integrates multiple data modalities. By harnessing chest X-ray images, CT scan images, blood smear images, and clinical information, our model offers a comprehensive framework for COVID-19 prediction and interpretation. This novel approach enhances diagnostic accuracy by capturing diverse facets of the disease, thereby providing healthcare practitioners with a more holistic view of COVID-19 cases.
2. **Enhanced Interpretability in AI Healthcare Decision-Making:** Additionally, the research improved the interpretability of AI-driven healthcare solutions by incorporating visualization techniques such as Grad-CAM and heatmaps. These tools empower clinicians to comprehend the underlying rationale behind the model's predictions, fostering trust and confidence in AI-based diagnostic systems. By promoting transparency and interpretability, our study bridges the gap between sophisticated machine learning methodologies and their practical application in clinical settings.

5.3 Limitations of the Study

The Limitations of the study from the development of a Hybrid Deep Learning Based Model for COVID-19 Prediction and Interpretation Using Multiple Data Modalities include:

1. **Data Quality and Source:** The primary limitation stems from the utilization of secondary sources, such as online repositories, for collecting datasets encompassing chest X-rays, CT scan images, blood smear images, and clinical information. Data obtained from these sources

may vary in quality, consistency, and annotation, potentially affecting the reliability and accuracy of the model.

2. **Lack of Firsthand Knowledge:** Gathering data from primary sources, such as hospitals or healthcare facilities, would provide firsthand access to patient information, ensuring higher quality and accuracy. However, due to the dire risk of COVID-19 infection, accessing primary data may not have been feasible, leading to reliance on secondary sources.
3. **Data Heterogeneity and Bias:** Datasets sourced from online repositories may exhibit heterogeneity and bias due to variations in imaging techniques, equipment, patient demographics, and disease severity. This heterogeneity can introduce inconsistencies and confounding factors, impacting the model's performance and generalizability.
4. **Limited Clinical Context:** Secondary data sources may lack comprehensive clinical context, including detailed patient histories, laboratory results, and treatment outcomes. Without this context, the model's ability to make accurate predictions and interpretations may be compromised, limiting its clinical utility and relevance.
5. **Computing Resources:** Another limitation relates to the availability and scalability of computing resources for training and evaluating deep learning models. Processing large volumes of multidimensional data, such as chest X-rays and CT scans, requires significant computational power and memory resources, which may be constrained in certain environments.
6. **Model Interpretability:** Deep learning models, especially hybrid architectures, may lack interpretability, making it challenging to understand the underlying features driving predictions. Without interpretable models, clinicians may be hesitant to trust the model's outputs and incorporate them into clinical decision-making processes.

The limitations of relying on secondary data sources, such as online repositories, for diverse COVID-19 datasets, coupled with potential data heterogeneity and bias, may compromise the accuracy and reliability of the proposed hybrid deep learning model. Additionally, the lack of comprehensive clinical context and limited interpretability of the model could hinder its clinical utility and trustworthiness among healthcare professionals. Moreover, constraints in computing resources may impede the model's scalability and performance in processing large volumes of multidimensional data, potentially affecting the quality of predictions and interpretations.

5.4 Recommendations

The research work on developing a hybrid deep learning model for COVID-19 prediction and interpretation using multiple data modalities has yielded significant insights into enhancing diagnostic and prognostic capabilities. Moving forward, the following recommendations are made to further enhance the efficacy and applicability of developed model in real-world scenario.

(a) **Adoptation of Multimodal Data Integration in Healthcare Diagnostics:** The success of the proposed model demonstrates the benefits of integrating multiple data modalities, such as chest X-rays, CT scans, blood smears, and clinical data, for a more comprehensive and accurate diagnosis. It is recommended that healthcare systems adopt similar multimodal approaches to enhance diagnostic accuracy and provide a more holistic view of a patient's condition.

(b) **Implementation of Explainable AI Techniques:** The use of Grad-CAM in the model provided valuable insights into the decision-making process of the neural networks, increasing the transparency and interpretability of predictions. It is recommended to integrate explainable AI techniques in diagnostic models to help healthcare professionals understand and trust the model's outputs, which can improve clinical decision-making and patient outcomes.

(c) **Investment in AI and Deep Learning Infrastructure:** To fully leverage the benefits of advanced AI models, healthcare institutions should invest in the necessary computational infrastructure, including high-performance computing systems and data storage solutions.

This investment will enable the efficient processing of large and complex datasets, ensuring the timely and accurate analysis needed for effective diagnosis and treatment planning.

(d) **Continuous Model Validation and Update:** As new data becomes available, especially with emerging variants of COVID-19, it is crucial to continually validate and update the deep learning model to maintain its accuracy and reliability. Regularly incorporating new data will help the model adapt to evolving patterns and improve its predictive capabilities, ensuring it remains a valuable tool in managing the pandemic.

(e) **Training and Education for Healthcare Professionals:** Healthcare professionals should receive training on the use of AI and deep learning models in diagnostics. Understanding the strengths and limitations of these technologies will enable them to make informed decisions and effectively integrate these tools into their practice. Additionally, knowledge of interpretability techniques, like Grad-CAM, will help them better communicate the model's findings to patients.

(f) **Collaboration and Data Sharing Among Institutions:** The development and refinement of AI models benefit greatly from large and diverse datasets. It is recommended that healthcare institutions collaborate and share anonymized patient data, within ethical and legal frameworks, to enhance the generalizability and robustness of predictive models. Such collaboration can accelerate the development of more accurate and universally applicable diagnostic tools.

(g) **Research and Development in AI for Healthcare:** The positive outcomes of the hybrid model highlight the potential for AI to revolutionize healthcare diagnostics. Continued research and development in this field should be encouraged, focusing on improving model accuracy, interpretability, and applicability across different diseases and medical conditions. This investment will pave the way for more advanced, reliable, and user-friendly AI tools in healthcare.

5.5 Future Research Work

Future research directions could include the following areas:

(a) **Vision Transformers in Medical Imaging:** Vision Transformers (ViTs) have shown significant promise in various computer vision tasks, particularly in image classification and segmentation. Future research could explore the application of ViTs in medical imaging, such as radiology and histopathology, to enhance the detection and diagnosis of COVID-19 diseases. Researchers can focus on optimizing ViTs for handling medical images' unique challenges, like varying image quality, noise, and the need for interpretability. Additionally, the integration of ViTs with existing convolutional neural network (CNN) architectures could be investigated to combine the strengths of both models for improved performance.

(b) **Autoencoder Architectures for Anomaly Detection:** Future research could focus on developing more robust and interpretable autoencoder architectures tailored for COVID-19 diagnosis. Furthermore, combining autoencoders with other machine learning techniques could be explored to enhance the detection of subtle anomalies in COVID-19 image datasets.

(c) **Generative AI for Data Augmentation and Synthesis:** Future research can explore the use of generative models such as Generative Adversarial Networks (GANs) to create synthetic medical images, clinical data, and even patient records, ensuring privacy and reducing the need for extensive real-world data collection. Additionally, the application of generative AI in creating personalized treatment plans or predicting disease progression could be a promising area of investigation.

(d) **Multimodal AI for Integrated Healthcare Solutions:** The integration of multiple data modalities, such as imaging, genomics, clinical data, and wearable sensor data, can lead to a more comprehensive understanding of patient health. Future research could focus on developing multimodal AI systems that seamlessly integrate and analyze diverse data sources. These systems could provide holistic insights into patient conditions, enabling more accurate diagnosis, personalized treatment plans, and better patient outcomes. Research can also explore the challenges of data fusion, standardization, and privacy in multimodal AI applications.

(e) **Explainability and Interpretability in AI Models:** As AI models become more complex, ensuring their interpretability and explainability is crucial, especially in healthcare settings. Future research could focus on developing methods and frameworks that provide clear and understandable explanations for AI model decisions, helping healthcare professionals trust and adopt these technologies. This includes exploring techniques like attention mechanisms, saliency maps, and counterfactual explanations, which can elucidate the reasoning behind model predictions and highlight relevant features in the input data.

(f) **Ethical and Fair AI in Healthcare:** Addressing ethical concerns and ensuring fairness in AI models is a critical area of future research. This includes studying the potential biases in AI algorithms, ensuring equitable treatment across different patient demographics, and protecting patient privacy. Researchers can work on developing guidelines and frameworks to assess and mitigate biases in AI systems, ensuring that these technologies are used responsibly and ethically in healthcare.

REFERENCES

Abayomi-Alli, O. O., Damaševičius, R., Maskeliūnas, R., & Misra, S. (2022). An Ensemble Learning Model for COVID-19 Detection from Blood Test Samples. *Sensors (Basel, Switzerland), 22*.

Abbasi-Kesbi, K., Jafari, M. H., & Alizadehsani, R. (2022). A deep neural network model for COVID-19 prediction based on chest CT images. Journal of X-Ray Science and Technology, 30(1), 103-114. doi: 10.3233/XST-218221

Adedeji A., Nnebe S. E, and Sadiq, S. I. (2023), Application and Assessment of Artificial Neural Networks in Modern Medicine for Predicting and Cushioning the effect of COVID-19 Pandemic, Journal of Natural Sciences Research, 14(6): 41 – 47

Adedeji A., Nnebe S. E, and Sadiq, S. I. (2023), Predictive Assessment of Post-COVID-19 Impact using Artificial Intelligence Neural Network Models, Journal of Natural Sciences Research, 14(6): 48 – 54.

Afreen, Y. N., and Reddy, P. B. (2022). CDLX: An Efficient novel approach for COVİD detection lung xrays throw transfer learning based on State of the art deep learning image classification models. International Journal of Health Sciences, 6(S5): 8439–8463. https://doi.org/10.53730/ijhs.v6nS5.10593

Agarwal, M., Sharma, A., & Kunwar, S. (2022), Covid-19 prediction using Machine Learning (ML), International Journal of Advances in Engineering and Management (IJAEM), 4(5), 933 – 937.

Ahmed, S. Z. (2020), Analysis and Forecasting the Outbreak of Covid-19 in Ethiopia Using Machine Learning, European Journal of Computer Science and Information Technology, 8(4), 1 – 13.

Al-Shehri, W., Almalki, J., Mehmood, R., Alsaif, K., Alshahrani, S.M., Jannah, N., and Alangari, S. (2022), A Novel COVID-19 Detection Technique Using Deep Learning Based Approaches, Sustainability, 14(12222), 1 – 12. https://doi.org/10.3390/su141912222

Alves, M. A., Castro, G. Z., Oliveira, B. A., Ferreira, L. A., Ramírez, J. A., Silva, R. C., & Guimarães, F. G. (2021). Explaining machine learning based diagnosis of COVID-19 from routine blood tests with decision trees and criteria graphs. *Computers in Biology and Medicine*, *132*, 104335 - 104335.

Anwar, T., & Zakir, S. (2020). Deep learning based diagnosis of COVID-19 using chest CT-scan images. *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 1-5.

Apostolopoulos, I. D., & Mpesiana, T. A. (2020), Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks, Physical and Engineering Sciences in Medicine, *43*, 635 - 640. https://doi.org/10.1007/s13246-020-00865-4

Arias-Londoño, J. D., Gómez-García, J. A., Moro-Velázquez, L., & Godino-Llorente, J. I. (2020). Artificial Intelligence Applied to Chest X-Ray Images for the Automatic Detection of COVID-19. A Thoughtful Evaluation Approach. *IEEE Access*, *8*, 226811 - 226827.

Awwalu J., Umar N. A., Ibrahim M. S., and Ogwueleka F. N. (2020), A Multinomial Naïve Bayes Decision Support System for COVID-19 Detection, FUDMA Journal of Sciences (FJS) 4(2): 704 – 711 DOI: https://doi.org/10.33003/fjs-2020-0402-331

Basha, K. N., Nishitha, K.,. Gracy, S. C., Sowmiya, M., & Priyadharshini, B. (2021), COVID Prediction using Machine Learning, International Journal of Engineering Research & Technology (IJERT), 10(5), 365 – 370.

Bashar, A., Latif, G., Ben-Brahim, G., Mohammad, N., and Alghazo, J. (2021), COVID-19 Pneumonia Detection Using Optimized Deep Learning Techniques. Diagnostics, 11, 1 - 18. https://doi.org/10.3390/ diagnostics11111972

Becerra-Sánchez, A., Rodarte-Rodríguez, A., Escalante-García, N. I., Olvera-González, J. E., Rosa-Vargas, J. I. D., Zepeda-Valles, G., & Velásquez-Martínez, E. D. J. (2022), Mortality Analysis of Patients with COVID-19 in Mexico Based on Risk Factors Applying Machine Learning Techniques, MPDI Diagnostics, 12(1396), 1 – 28. https://doi.org/10.3390/diagnostics12061396

Bouchareb Y., Khaniabadi P. M., Kindi F. A., Dhuhli H. A., Shiri I., Zaidi H., and Rahmim A. (2021), Artificial intelligence-driven assessment of radiological images for COVID-19, Computers in Biology and Medicine 136 (2021): 1 - 17

Brinati, D., Campagner, A., Ferrari, D., Locatelli, M., Banfi, G., & Cabitza, F. (2020), Detection of COVID-19 Infection from Routine Blood Exams with Machine Learning: A Feasibility Study, Journal of Medical Systems, 44(135), 1 – 12.

Bukhari, S.U., Bukhari, S.U., Syed, A., & Shah, S.S. (2020). The diagnostic evaluation of Convolutional Neural Network (CNN) for the assessment of chest X-ray of patients infected with COVID-19. *medRxiv*.

Butt, C., Gill, J., Chun, D., & Babu, B. A. (2020), Deep learning system to screen coronavirus disease 2019 pneumonia, Applied Intelligence, 1 – 7. https://doi.org/10.1007/s10489-020-01714-3

Calderon-Ramirez, S., Yang, S., Moemeni, A., Colreavy-Donnelly, S., Elizondo, D.A., Oala, L., Rodríguez-Capitán, J., Jiménez-Navarro, M.F., López-Rubio, E., & Molina-Cabello, M.A. (2021). Improving Uncertainty Estimation With Semi-Supervised Deep Learning for COVID-19 Detection Using Chest X-Ray Images. *IEEE Access, 9*, 85442-85454.

Car, Z., Šegota , S. B., Anđelić, N., Lorencin, I., & Mrzljak, V. (2020), Modeling the Spread of COVID-19 Infection Using a Multilayer Perceptron, Hindawi Computational and Mathematical Methods in Medicine, 2020, 1 – 10. https://doi.org/10.1155/2020/5714714

Centers for Disease Control and Prevention. (2021). COVID-19: Symptoms of coronavirus. Retrieved from https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html

Chandra, T.B., Verma, K., Singh, B.K., Jain, D., & Netam, S.S. (2020). Coronavirus disease (COVID-19) detection in Chest X-Ray images using majority voting based classifier ensemble. *Expert Systems with Applications, 165*, 113909 - 113909.

Chauha H., and Modi K. (2022), Development of a Multi-modal Severity Prediction System for Covid-19 using Machine Learning Algorithms, International Journal of Intelligent Systems and Applications in Engineering IJISAE, 10(3), 314–321

Cheng, M. P., Papenburg, J., Desjardins, M., Kanjilal, S., Quach, C., Libman, M. D., Dittrich, S., & Yansouni, C. P. (2020). Diagnostic Testing for Severe Acute Respiratory Syndrome–Related Coronavirus-2. *Annals of Internal Medicine*.

Chokri S., Daoud W. B., Hanini W., Mahfoudhi S., and Makhlouf A. (2022), AI-based System for the Detection and Prevention of COVID-19. International Journal of Advanced Computer Science and Applications, 13(1): 582 – 591.

Das, S., Ayus, I., & Gupta, D. (2023), A comprehensive review of COVID-19 detection with machine learning and deep learning techniques, Health and Technology, 13: 679 – 692. https://doi.org/10.1007/s12553-023-00757-z

Dokeroglu T. (2023), A new parallel multi-objective Harris hawk algorithm for predicting the mortality of COVID-19 patients. PeerJ Computer Science, 9: 1 – 21. DOI 10.7717/peerj-cs.1430

Fanelli, D., & Piazza, F. (2020), Analysis and forecast of COVID-19 spreading in China, Italy and France, Chaos, Solitons and Fractals, 134(109761), 1 – 5. https://doi.org/10.1016/j.chaos.2020.109761

Ferrari, D., Motta, A., Strollo, M., Banfi, G., & Locatelli, M. (2020). Routine blood tests as a potential diagnostic tool for COVID-19. *Clinical Chemistry and Laboratory Medicine (CCLM), 58*, 1095 - 1099.

Foysal, M. J., & Aowlad H. A. (2021). COVID-19 Detection from Chest CT Images using Ensemble Deep Convolutional Neural Network. *2021 2nd International Conference for Emerging Technology (INCET)*, 1-6.

Fu, M., Yi, S., Zeng, Y., Ye, F., Li, Y., Dong, X., Ren, Y., Luo, L., Pan, J., & Zhang, Q. (2020). Deep Learning-Based Recognizing COVID-19 and other Common Infectious Diseases of the Lung by Chest CT Scan Images. *medRxiv*.

Gao, Y., Zhang, D., Xia, Y., Chen, L., Zhao, Y., & Sun, Y. (2021). A COVID-19 prediction model using multiple modalities of data. IEEE Journal of Biomedical and Health Informatics, 25(7), 2635-2643. doi: 10.1109/JBHI.2021.3057289

Ghafouri-Fard, S., Hossein Mohammad-Rahimi, H., Motie, P., Minabi, M. A. S., Taheri, M., & Nateghinia, S. (2021), Application of machine learning in the prediction of COVID-19 daily new

Ghomi Z, Mirshahi R, Khameneh B. A, Fattahpour A, Mohammadiun S, Alavi Gharahbagh A, Djavadifar A, Arabalibeik H, Sadiq R, and Hewage K. (2020). Segmentation of COVID-19 pneumonia lesions: A deep learning approach. Medical Journal Islam Republic Iran, 34(174): 1 – 7, https://doi.org/10.47176/mjiri.34.174

GISAID. (2021). Tracking of variants. Retrieved from https://www.gisaid.org/hcov19-variants/

Gonsalves, C., & Nitnaware, P. (2021), Prediction of Covid-19 Cases Using Machine Learning, Journal of Emerging Technologies and Innovative Research, 8(10): a518 – a528.

Gozes, O., Frid-Adar, M., Sagie, N., Zhang, H., Ji, W., & Greenspan, H. (2020). Coronavirus Detection and Analysis on Chest CT with Deep Learning. *ArXiv, abs/2004.02640*.

Guo H., Zhou Y., Liu X and Tan J. (2020), The impact of the COVID-19 epidemic on the utilization of emergency dental services, Journal of Dental Sciences, 15(4): 564–567.

Guo Z., Zhang L., and Zhang D. (2010), "A completed modeling of local binary pattern operator for texture classification," IEEE Transactions on Image Processing (TIP), 19(6): 1657-1663

Himadri M., Subhankar G., Ankita D., Obaidullah S. M., Santosh K. C., and Kaushik R. (2020), Deep neural network to detect COVID-19: one architecture for both CT Scans and Chest X-rays, Applied Intelligence, 2021(51): 2777–2789. https://doi.org/10.1007/s10489-020-01943-6:

Horry, M. J., Chakraborty, S., Paul, M., Ulhaq, A., Pradhan, B., Saha, M., & Shukla, N. (2020). COVID-19 Detection Through Transfer Learning Using Multimodal Imaging Data. *Ieee Access, 8*, 149808 - 149824.

Hu, X., Qiu, C., Wu, S., Wu, D., & Wang, X. (2022). Deep learning-based COVID-19 prediction using chest CT images and clinical data. Journal of X-Ray Science and Technology, 30(2), 203-213. doi: 10.3233/XST-218204

Huyut, M. T. (2022). Automatic Detection of Severely and Mildly Infected COVID-19 Patients with Supervised Machine Learning Models. *Ingenierie et Recherche Biomedicale, 44*, 100725 - 100725.

Idupulapaati, A., & Sreedevi, M. (2022). COVID-19 Detection from Chest Radiograph Images of Patients using Neural Networks. *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 1*, 2051-2054.

International Labour Organization. (2021). COVID-19 and the world of work. Retrieved from https://www.ilo.org/global/topics/coronavirus/lang--en/index.htm

Ismael, A.M., & Şengür, A. (2020). Deep learning approaches for COVID-19 detection based on chest X-ray images. *Expert Systems with Applications*, *164*, 114054 - 114054.

Jain, R., Gupta, M., Taneja, S., & Hemanth, D.J. (2020). Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Applied Intelligence (Dordrecht, Netherlands), 51*, 1690 – 1700

Ju L., Yuhao S., Jingshu C., Xin L., and Canxia X. (2022), A novel deep learning-based method for COVID-19 pneumonia detection from CT images, BMC Medical Informatics and Decision Making, 22(284), 1 – 7. https://doi.org/10.1186/s12911-022-02022-1

Kapal D., Sunder A. K., Ankur S. B., Vaibhav S., and Surbhi B. (2021), Triage of potential COVID-19 patients from chest X-ray images using hierarchical convolutional networks, Neural Computing and Applications, 1 – 16. https://doi.org/10.1007/s00521-020-05641-9

Karhan, Z., & Akal, F. (2020). Covid-19 Classification Using Deep Learning in Chest X-Ray Images. *2020 Medical Technologies Congress (TIPTEKNO)*, 1-4.

Kaushik, B., Chadha, A. & Sharma, R. (2023), Performance Evaluation of Learning Models for the Prognosis of COVID-19. *New Gener. Comput.* 41, 533–551. https://doi.org/10.1007/s00354-023-00220-7

Khalil, M. I., Rehman, S. U., Alhajlah, M., Mahmood, A., Karamat, T., Haneef, M., & Alhajlah, A. (2022). Deep-COVID: Detection and Analysis of COVID-19 Outcomes Using Deep Learning. *Electronics*.

Khan, A.A., Shafiq, S., Kumar, R., Kumar, J., & Haq, A.U. (2020). H3DNN: 3D Deep Learning Based Detection of COVID-19 Virus using Lungs Computed Tomography. *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 183-186.

Liu, F., Zhou, Y., Zhou, Y., Huang, J., Ma, L., & Guo, X. (2021). A deep learning-based model integrating multiple modalities of data for COVID-19 prediction. IEEE Transactions on Neural Networks and Learning Systems, 32(6), 2567-2577. doi: 10.1109/TNNLS.2020.3037849

Loey, M., Smarandache, F., & Khalifa, N.E. (2020). Within the Lack of Chest COVID-19 X-ray Dataset: A Novel Detection Model Based on GAN and Deep Transfer Learning. *Symmetry, 12*, 651.

Luján-García, J. E., Yáñez-Márquez, C., Villuendas-Rey, Y., & Camacho-Nieto, O. (2020), A Transfer Learning Method for Pneumonia Classification and Visualization, MDPI Applied Sciences, 10(2908), 1 – 18. doi:10.3390/app10082908

Majid, A., Khan, M. A., Nam, Y., Tariq, U., Roy, S., Mostafa, R. R., & Sakr, R. H. (2021). COVID19 Classification Using CT Images via Ensembles of Deep Learning Models. *Computers, Materials & Continua*.

Majumder, A. B., Gupta, S., Singh, D., & Majumder, S. (2021), An Intelligent System for Prediction of COVID-19 Case using Machine Learning Framework-Logistic Regression, Journal of Physics: Conference Series, 1797(012011), 1 – 9. doi:10.1088/1742-6596/1797/1/012011

Marappan, R., S. Bhaskaran, S., Aakaash, N., & Mitha, S. M. (2022), Analysis of COVID-19 Prediction Models: Design & Analysis of New Machine Learning Approach, Journal of Applied Mathematics and Computation, 6(1), 121 – 126.

Mei, X., Lee, H., Diao, K., Huang, M., Lin, B., Liu, C., Xie, Z., Ma, Y., Robson, P.M., Chung, M.S., Bernheim, A., Mani, V., Calcagno, C., Li, K., Li, S., Shan, H., Lv, J., Zhao, T., Xia, J., Long, Q., Steinberger, S., Jacobi, A., Deyer, T., Luksza, M., Liu, F., Little, B.P., Fayad, Z.A., & Yang, Y. (2020). Artificial intelligence–enabled rapid diagnosis of patients with COVID-19. *Nature Medicine*, 1-5.

Meraihi, Y., Gabis, A. B., Mirjalili, S., Ramdane-Cherif, A., & Alsaadi, F. E. (2022), Machine Learning-Based Research for COVID-19 Detection, Diagnosis, and Prediction: A Survey, SN Computer Science, 3(286), 1 – 35. https://doi.org/10.1007/s42979-022-01184-z

Mohammadi, F., Dehbozorgi, L., Akbari-Hasanjani, H. R., Joz Abbasalian, Z., Akbari-Hasanjani, R., Sabbaghi-Nadooshan, R., & Moradi Tabriz, H. (2023). Evaluation of effective features in the diagnosis of Covid-19 infection from routine blood tests with multilayer perceptron neural network: A cross-sectional study. *Health Science Reports, 6*.

Mporas, I., & Naronglerdrit, P. (2020). COVID-19 Identification from Chest X-Rays. *2020 International Conference on Biomedical Innovations and Applications (BIA)*, 69-72.

Muhammad et al. (2022), Deep transfer learning CNN based approach for COVID-19 detection, International Journal of Advanced and Applied Sciences, 9(4): 44-52

Muhammad, L. J., Islam, M. M., Sharif, U. S., & Ayon, S. I. (2020), Predictive Data Mining Models for Novel Coronavirus (COVID-19) Infected Patients Recovery, Research Square, 1: 1 – 15. DOI: https://doi.org/10.21203/rs.3.rs-33247/v1

Mukri, V. M. (2023), COVID-19 Prediction using Machine Learning, International Research Journal of Modernization in Engineering Technology and Science, 5(3): 1180 – 1185

Nagendra, K., Praneeth, U. S., Gagan, K. K., Ravi, Y. S., Katta, J. K., & Satya, S. S. N. (2022), COVID-19 Prediction Based on Symptoms, International Journal of Scientific & Engineering Research, 13(4), 738 – 741.

Nandi R., and Mulimani M. (2021), Detection of COVID-19 from X-rays using hybrid deep learning models, Research on Biomedical Engineering, 37: 687–695. https://doi.org/10.1007/s42600-021-00181-0

Ngabo D, Dong W, Ibeke E, Iwendi C, Masabo E. (2021), Tackling pandemics in smart cities using machine learning architecture. Math Biosci Eng., 18(6): 8444 - 8461. doi: 10.3934/mbe.2021418. PMID: 34814307.

Oliver A., Lladó X., Freixenet J., and Martí J.(2007), "False positive reduction in mammographic mass detection using local binary patterns," In Proceeding Medical Image Computing and Computer-Assisted Intervention (MICCAI).

Oyelade, O.N., Ezugwu, A.E., & Chiroma, H. (2021). CovFrameNet: An Enhanced Deep Learning Framework for COVID-19 Detection. *IEEE Access*, *9*, 77905-77919.

Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., & Acharya, R. U. (2020), Automated detection of COVID-19 cases using deep neural networks with X-ray images, Computers in Biology and Medicine, 121 (103792), 1 – 11. https://doi.org/10.1016/j.compbiomed.2020.103792

Pal, R., Sekh, A. A., Kar, S., & Prasad, D. K. (2019), Neural network based country wise risk prediction of COVID-19, IEEE Access, 4(2016): 1 – 9.

Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R., & Singh, V. (2020), Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet, Chaos, Solitons and Fractals, 138 (109944), 1 – 8. https://doi.org/10.1016/j.chaos.2020.109944

Pola Ö. (2021), Detection of Covid-19 from Chest CT Images Using Xception Architecture: A Deep Transfer Learning Based Approach, Sakarya University Journal of Science, 25(3), 800-810, 2021

Prakash, K. B., Imambi, S. S., Ismail, M., Kumar, P. T., & Pawan, Y. N. (2020), Analysis, Prediction and Evaluation of COVID-19 Datasets using Machine Learning Algorithms, International Journal of Emerging Trends in Engineering Research, 8(5), 2199 – 2204.

Punia, R., Kumar, L. S., Mujahid, M., & Rohilla, R. (2020). Computer Vision and Radiology for COVID-19 Detection. *2020 International Conference for Emerging Technology (INCET)*, 1-5.

Purohit, K., Kesarwani, A., Kisku, D.R., & Dalui, M. (2020). COVID-19 Detection on Chest X-Ray and CT Scan Images Using Multi-image Augmented Deep Learning Model. *bioRxiv*.

Qiao, Z., Bae, A., Glass, L., Xiao, C., & Sun, J. (2020). FLANNEL (Focal Loss bAsed Neural Network EnsembLe) for COVID-19 detection. *Journal of the American Medical Informatics Association : JAMIA, 28*, 444 - 452.

Rani G., Oza M. G., Dhaka V. S., Pradhan N., Verma S., and Rodrigues J. P. C. (2022), Applying deep learning-based multi-modal for detection of coronavirus, Multimedia Systems 28:1251–1262

Reddy, S. V. M., Ramana, K. V., Kumar, M. T., & Krishna, M. R. (2023), COVID-19 Cases Prediction Using Machine Learning, International Research Journal of Engineering and Technology (IRJET), 10(5), 1749 – 1754.

Redie, D. K., Sirko, A. E., Demissie, T. M., Teferi, S. S., Shrivastava, V. K., Verma, O. P., & Sharma, T. K. (2022). Diagnosis of COVID-19 using chest X-ray images based on modified DarkCovidNet model. *Evolutionary Intelligence, 16*, 729 - 738.

Rehman, A., Sadad, T., Saba, T., Hussain, A., & Tariq, U. (2021). Real-Time Diagnosis System of COVID-19 Using X-Ray Images and Deep Learning. *IT Professional, 23*, 57 - 62.

Rehman, M. U., Shafique, A., Khalid, S., Driss, M., & Rubaiee, S. (2021). Future Forecasting of COVID-19: A Supervised Learning Approach. *Sensors (Basel, Switzerland), 21*.

Sait U., Gokul-Lal K.V., Shivakumar S., Kumar T., Bhaumik R., Prajapati S., Bhalla K, and Chakrapani A. (2021), A deep-learning based multimodal system for Covid-19 diagnosis using breathing sounds and chest X-ray images, Applied Soft Computing 109, 1 – 24

Salih, S. Q., Abdulla, H. K., Ahmed, Z. S., Surameery, N. M., & Rashid, R. (2020). Modified AlexNet Convolution Neural Network For Covid-19 Detection Using Chest X-ray Images.

Sekeroglu B., and Ozsahin I. (2020), Detection of COVID-19 from Chest X-Ray Images using Convolutional Neural Networks, SLAS Technology, 25(6): 553–565

Shahanaz A. G., Niranjana S., Madhushankara M and Swathi K. S. (2022), Deep learning model for detection of COVID-19 utilizing the chest X-ray images, Cogent Engineering, 9(1), 1 - 18, DOI: 10.1080/23311916.2022.2079221

Shamik T., and Anurag J. (2021), Convolutional capsule network for COVID-19 detection using radiography images, International Journal of Imaging System Technol., 31: 525–539

Sher, T., Rehman, A., & Kim, D. (2023), COVID-19 Outbreak Prediction by Using Machine Learning Algorithms, Computers, Materials & Continua, 74(1), 1561 – 1574. DOI: 10.32604/cmc.2023.032020

Shikang L., Zhuo L., Ruiqing X., Xiaofeng Y., Jianhua L., Liang L., Yuyan S., Qiangzhong S., Tongxin L., Lanlan Q., and Ying, Z. (2021), Multimodal deep learning for predicting COVID-19 patients at high-risk for serious illness, Electronic Journal of Emerging Infectious Diseases, 6(2): 133 – 137

Singh, K., & Singh, A. (2021). Diagnosis of COVID-19 from chest X-ray images using wavelets-based depth-wise convolution network. *Big Data Min. Anal., 4*, 84 – 93.

Soumyajit P, Somnath B., and Arijit R. (2021), An efficient method of detection of COVID-19 using Mask R-CNN on chest X-Ray images, AIMS Biophysics, 8(3): 281–290. DOI: 10.3934/biophy.2021022

Stephanie et al. (2020), Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets, Nature Communications, 11(4080): 1 – 7, | https://doi.org/10.1038/s41467-020-17971-2 | www.nature.com/naturecommunications

Subhalakshmi R. T., Balamurugan S A. A., and Sasikala S. (2022), Deep learning based fusion model for COVID-19 diagnosis and classification using computed tomography images, Concurrent Engineering: Research and Applications, 30(1): 116–127

Subhan, A., Rasheed, T., Shah, Z., Noor, S., Khan M. A., & Shakoor, U. (2021), COVID-19 Patient Health Prediction using Artificial Intelligence Boosted Random Forest Algorithm, South Asian Journal of Social Sciences & Humanities, 2(3): 203 – 221.

Sujath, R., Chatterjee, J. M. & Hassanien, A. E. (2020), A machine learning forecasting model for COVID-19 pandemic in India, Stochastic Environmental Research and Risk Assessment, 34: 959 – 972. https://doi.org/10.1007/s00477-020-01827-8

Sultana, J., Singha, A. K., Siddiqui, S. T., Guthikonda Nagalaxmi, G., Sriram, A. K., & Pathak, N. (2022), COVID-19 Pandemic Prediction and Forecasting Using Machine Learning Classifiers, Intelligent Automation & Soft Computing, 32(2), 1007 – 1024. DOI:10.32604/iasc.2022.021507

Thangamma K. C., Bhavana N. Gnaneshwari, D. S., Syeda, S. F., & Yashaswini, S. (2021), Automated Data Analysis and Prediction of Covid-19 Pandemic using Machine Learning Techniques, International Journal of Creative Research Thoughts, 9(5), d380 – d383.

Turtinen M., Pietikäinen M., and Silven O. (2006), “Visual characterization of paper using Isomap and local binary patterns”, IEICE Transactions on Information and System, 89(7): 2076-2083

Udayaraju, P., Narayana, T. V., Vemparala, S., Srinivasarao, C., & Raju, B. (2023). GW- CNNDC: Gradient weighted CNN model for diagnosing COVID-19 using radiography X-ray images. *Measurement. Sensors, 27*, 100735 - 100735.

Vaishya, R., Javaid, M., Khan, I. H., & Haleem, A. (2020). Artificial Intelligence (AI) application for COVID-19 pandemic. Diabetes & Metabolic Syndrome: Clinical Research & Reviews, 14(4), 337-339.

Vedika et al. (2022), Improved COVID-19 detection with chest x-ray images using deep learning, Multimedia Tools and Applications, 2022(81): 37657–37680. https://doi.org/10.1007/s11042-022-13509-4

Wang, L., Lin, Z.Q., & Wong, A. (2020). COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Scientific Reports, 10*.

Wang, T., Zhao, Y., Zhu, L., Liu, G., Ma, Z., & Zheng, J. (2020). Lung CT image aided detection COVID-19 based on Alexnet network. *2020 5th International Conference on Communication, Image and Signal Processing (CCISP)*, 199-203.

World Health Organization, Q & A on Coronaviruses (COVID-19), World Health Organization, 2020, https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/q-acoronaviruses. (Accessed 12 May 2023).

World Health Organization. (2020a). Naming the coronavirus disease (COVID-19) and the virus that causes it. Retrieved from https://www.who.int/emergencies/disease-outbreak-news/item/2020-DON- Naming-the-coronavirus-disease-(COVID-19)-and-the-virus-that-causes-it

World Health Organization. (2020b). COVID-19: The race to find a vaccine. Retrieved from https://www.who.int/emergencies/disease-outbreak-news/item/2020-DON-24-march-2020

World Health Organization. (2021). COVID-19 vaccines. Retrieved from https://www.who.int/emergencies/disease-outbreak-news/item/2021-DON-21-january-2021-covid-19-vaccines

Xu, et al. (2020). Application of ordinal logistic regression analysis to identify the determinants of illness severity of COVID-19 in China. Epideminology & I

Zhang, X., Saleh, H., Younis, E. M. G., Saha R., & Ali, A. A. (2020), Predicting Coronavirus Pandemic in Real-Time Using Machine Learning and Big Data Streaming System, Hindawi Complexity, 2020, 1 – 10. https://doi.org/10.1155/2020/6688912

APPENDIX A

SAMPLE PROGRAM CODE LISTING

```
# Code to collect and preprocess dataset (example using image datasets)
# Import necessary libraries
import os
import pandas as pd
import numpy as np
from PIL import Image
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Code to collect chest X-ray images
# Example: use directory paths containing images
chest_xray_directory = 'path/to/chest_xray_images_folder'
chest_xray_images = []
labels = []

for folder in os.listdir(chest_xray_directory):
    if folder == 'COVID-19':
        label = 'COVID-19'
    else:
        label = 'healthy'
    for filename in os.listdir(os.path.join(chest_xray_directory, folder)):
        img = Image.open(os.path.join(chest_xray_directory, folder, filename))
        img = img.resize((224, 224))  # Resize images if needed
        chest_xray_images.append(np.array(img))
        labels.append(label)

# Code to collect CT scan images and blood smear images (similar approach)

# Preprocessing steps (e.g., normalization, resizing, data labeling)
# Splitting data into train/test sets using train_test_split()

# Example:
X_train, X_test, y_train, y_test = train_test_split(chest_xray_images, labels, test_size=0.2,
random_state=42)

# Label encoding if needed
le = LabelEncoder()
y_train_encoded = le.fit_transform(y_train)
y_test_encoded = le.transform(y_test)
```

```
import tensorflow as tf
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define the CNN model architecture
model = tf.keras.Sequential([
   # Input layer
   tf.keras.layers.InputShape((224, 224, 3)),

   # Convolutional layer with 64 filters, 3x3 kernel size, and ReLU activation
   tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same'),

   # Max pooling layer with pooling size of 2x2
   tf.keras.layers.MaxPooling2D((2, 2)),

   # Convolutional layer with 64 filters, 3x3 kernel size, and ReLU activation
   tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same'),

   # Max pooling layer with pooling size of 2x2
   tf.keras.layers.MaxPooling2D((2, 2)),

   # Flatten layer to convert the 2D feature map to a 1D vector
   tf.keras.layers.Flatten(),

   # Dense layer with 128 neurons and ReLU activation
   tf.keras.layers.Dense(128, activation='relu'),

   # Output layer with 1 neuron and Softmax activation for binary classification (COVID or not)
   tf.keras.layers.Dense(1, activation='sigmoid')
])

# Compile the model with Adam optimizer and binary cross-entropy loss
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Load the COVID-19 chest X-ray dataset
data_generator = ImageDataGenerator(rescale=1./255)
train_data = data_generator.flow_from_directory(directory='train_data', target_size=(224, 224),
batch_size=32, class_mode='binary')
test_data = data_generator.flow_from_directory(directory='test_data', target_size=(224, 224),
batch_size=32, class_mode='binary')

# Train the model
import tensorflow as tf
```

```
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define the ResNet50 model
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Add custom top layers for binary classification
inputs = base_model.input
x = base_model.layers[-1].output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(128, activation='relu')(x)
outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)

model = tf.keras.Model(inputs, outputs)

# Compile the model with Adam optimizer and binary cross-entropy loss
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Load the COVID-19 chest X-ray dataset
data_generator = ImageDataGenerator(rescale=1./255)
train_data = data_generator.flow_from_directory(directory='train_data', target_size=(224, 224),
batch_size=32, class_mode='binary')
test_data = data_generator.flow_from_directory(directory='test_data', target_size=(224, 224),
batch_size=32, class_mode='binary')

# Train the model
model.fit(train_data, epochs=10)

# Evaluate the model on the test data
test_loss, test_acc = model.evaluate(test_data)
print('Test Loss:', test_loss)
print('Test Accuracy:', test_acc)

import tensorflow as tf
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Preprocessing the CT scan images
import cv2
import numpy as np

def preprocess_ct_scan(image):
  # Convert the image to grayscale
  image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

```
  # Normalize the image intensity values
  image = image / 255.0

  # Resize the image to 224x224
  image = cv2.resize(image, (224, 224))

  # Expand the image dimension to match the model input shape
  image = np.expand_dims(image, axis=2)

  return image

# Define the ResNet50 model
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 1))

# Add custom top layers for binary classification
inputs = base_model.input
x = base_model.layers[-1].output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(128, activation='relu')(x)
outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)

model = tf.keras.Model(inputs, outputs)

# Compile the model with Adam optimizer and binary cross-entropy loss
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Load the COVID-19 CT scan dataset
data_generator = ImageDataGenerator(preprocessing_function=preprocess_ct_scan,
rescale=1./255)
train_data = data_generator.flow_from_directory(directory='train_data', target_size=(224, 224),
batch_size=32, class_mode='binary')
test_data = data_generator.flow_from_directory(directory='test_data', target_size=(224, 224),
batch_size=32, class_mode='binary')

# Train the model
model.fit(train_data, epochs=10)

# Evaluate the model on the test data
test_loss, test_acc = model.evaluate(test_data)
print('Test Loss:', test_loss)
print('Test Accuracy:', test_acc)
import tensorflow as tf
import numpy as np
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define the CNN model architecture
model = tf.keras.Sequential([
    # Input layer
    tf.keras.layers.InputShape((224, 224, 3)),

    # Convolutional layer with 64 filters, 3x3 kernel size, and ReLU activation
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same'),

    # Max pooling layer with pooling size of 2x2
    tf.keras.layers.MaxPooling2D((2, 2)),

    # Convolutional layer with 64 filters, 3x3 kernel size, and ReLU activation
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same'),

    # Max pooling layer with pooling size of 2x2
    tf.keras.layers.MaxPooling2D((2, 2)),

    # Flatten layer to convert the 2D feature map to a 1D vector
    tf.keras.layers.Flatten(),

    # Dense layer with 128 neurons and ReLU activation
    tf.keras.layers.Dense(128, activation='relu'),

    # Output layer with 1 neuron and Softmax activation for binary classification (COVID or not)
    tf.keras.layers.Dense(1, activation='sigmoid')
])

# Compile the model with Adam optimizer and binary cross-entropy loss
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Load the COVID-19 blood smear dataset
data_generator = ImageDataGenerator(rescale=1./255)
train_data = data_generator.flow_from_directory(directory='train_data', target_size=(224, 224),
batch_size=32, class_mode='binary')
test_data = data_generator.flow_from_directory(directory='test_data', target_size=(224, 224),
batch_size=32, class_mode='binary')

# Train the model
model.fit(train_data, epochs=10)

# Evaluate the model on the test data
test_loss, test_acc = model.evaluate(test_data)
```

```
print('Test Loss:', test_loss)
print('Test Accuracy:', test_acc)
```

OBJECTIVE-2

```
# Code to develop CNN models for COVID-19 prediction using TensorFlow/Keras

# Import TensorFlow and Keras
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

# Create and compile CNN model(s)
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)),
    MaxPooling2D(pool_size=(2, 2)),
    # Add more layers as needed
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')  # Binary classification for COVID-19
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model(s)
model.fit(X_train, y_train_encoded, epochs=10, batch_size=32, validation_data=(X_test,
y_test_encoded))
```

OBJECTIVE-3

```
import tensorflow as tf
import tensorflow.keras.backend as K

# Define the Grad-CAM function to generate heatmaps
def grad_cam(model, image):
    with tf.GradientTape() as tape:
        preds = model.predict(image)
        loss = K.mean(preds[:, 1])
    gradients = tape.gradient(loss, model.trainable_variables)
    cam = tf.reduce_mean(tf.stack(gradients, axis=-1), axis=-1)
    return cam

# Generate heatmaps for positive predictions
positive_indices = np.where(predictions > 0.5)[0]
for index in positive_indices:
    image = preprocessed_data[index]
```

```
    heatmap = grad_cam(model, image)
    # Visualize the heatmap
    visualize_heatmap(heatmap)
```

OBJECTIVE-4

```
import asyncio
import aiohttp
import tensorflow as tf

async def run_model(model, image):
    # Load the model weights
    model.load_weights('model_weights.h5')

    # Make prediction using the model
    prediction = model.predict(image)
    return prediction

async def main():
    async with aiohttp.ClientSession() as session:
        # Create a list of tasks to execute the models concurrently
        tasks = []
        for model in models:
            for image in preprocessed_data:
                task = asyncio.create_task(run_model(model, image))
                tasks.append(task)

        # Wait for all tasks to complete and collect predictions
        predictions = await asyncio.gather(*tasks)

        # Process and analyze the predictions
        # ...

if __name__ == '__main__':
    asyncio.run(main())
```

OBJECTIVE-5

```
# Code to evaluate model performance using standard metrics

# Evaluate the model on test data
loss, accuracy = model.evaluate(X_test, y_test_encoded)

# Example code for other metrics calculation (recall, precision, etc.)
from sklearn.metrics import classification_report, roc_auc_score
```

```
y_pred = model.predict(X_test)
# Additional metrics calculations
print(classification_report(y_test_encoded, (y_pred > 0.5).astype(int)))
print("ROC AUC:", roc_auc_score(y_test_encoded, y_pred))
```

APPENDIX B
(RESEARCH BUFGET)

Table B.1: Summary of PhD Research Budget

S/N	Description	Qty	Rate	Amount
1	**Personnel:**			
(a)	Technical Research Assistant Stipend: (N40,000.00 Per month for 24 months)	24	40,000.00	960,000.00
2	**Equipment:**			
(a)	High-Performance Computing (HPC) Resources for access to computing clusters or cloud computing services@N45,000 Per Month for 18 Months	18	45,000.00	810,000.00
(b)	Laptop with GPU/TPU for Deep Learning	1	1,240,000.00	1,240,000.00
(c)	Laserjet Printer	1	180,000.00	180,000.00
(d)	MTN 5G Router	1	50,000.00	50,000.00
3	**Software and Licenses:**			
(a)	Deep Learning Frameworks (e.g., TensorFlow, PyTorch) software licenses	1	87,000.00	87,000.00
	Programming & Statistical Analysis Software			
(a)	Pycharm Professional Edition	1	135,000.00	135,000.00
(b)	R Programming	1	85,000.00	85,000.00
(c)	Anaconda Professional Edition	1	92,000.00	92,000.00
(d)				0.00
4	**Data Collection and Acquisition:**			
(a)	Honorarium for acquiring datasets related to COVID-19	2	50,000.00	100,000.00
(b)	Data Storage Costs for cloud storage	24	65,000.00	1,560,000.00
5	**Travel and Conference Expenses:**			
(a)	Conference Registration Fees: (for attending relevant conferences or workshops)	1	60,000.00	60,000.00
(b)	Travel Expenses: (transportation, accommodation, and meals associated with conference attendance or research collaboration meetings	2	200,000.00	400,000.00
6	**Publication Costs:**			
(a)	Article Processing Charges (APCs): Open-access journal	1	75,000.00	75,000.00
	Article Processing Charges (APCs): for publishing research findings in IEEE Xplore	1	150,000.00	150,000.00
(b)	Membership Fees: $X for joining professional organizations or societies related to COVID-19 research	1	104,400.00	104,400.00
(c)	Miscellaneous Expenses:	1	200,000.00	200,000.00

				0.00
7	**Printing and Binding Costs:**			
(a)	Printing Dissertation copies and binding)	1	400,000.00	400,000.00
(b)	Communication Costs: (for internet and phone expenses related to research communication)	24	10,000.00	240,000.00
(c)	Contingency Fund: (for unforeseen expenses or emergencies)	24	10,000.00	240,000.00
8	**Consultations**			
	Technical Consultations	1	350,000.00	350,000.00
9	**Research Logistics**		500,000.00	500,000.00
			TOTAL =	**8,018,400.00**

Table B.1 shows outlines the financial needs for the PhD research project titled "A Hybrid Deep Learning Based Model for COVID-19 Prediction and Interpretation Using Multiple Data Modalities." It covers personnel, equipment, software, data collection, travel, publication, printing, communication, contingency, consultations, and research logistics. Personnel costs include stipends for a technical research assistant, and equipment expenses cover computing resources, a laptop with GPU/TPU, a printer, and a 5G router. Software and licenses include deep learning frameworks and storage costs. Data collection involves acquiring COVID-19 datasets and storage expenses. Travel and conference costs encompass registration fees and travel expenses. Publication expenses include processing charges and membership fees. Printing, communication, and contingency costs are accounted for, along with technical consultations and research logistics. In all, the budget details the essential expenses for successful research execution, reflecting thorough consideration of financial requirements.

APPENDIX C
(PROJECT TIMELINES)

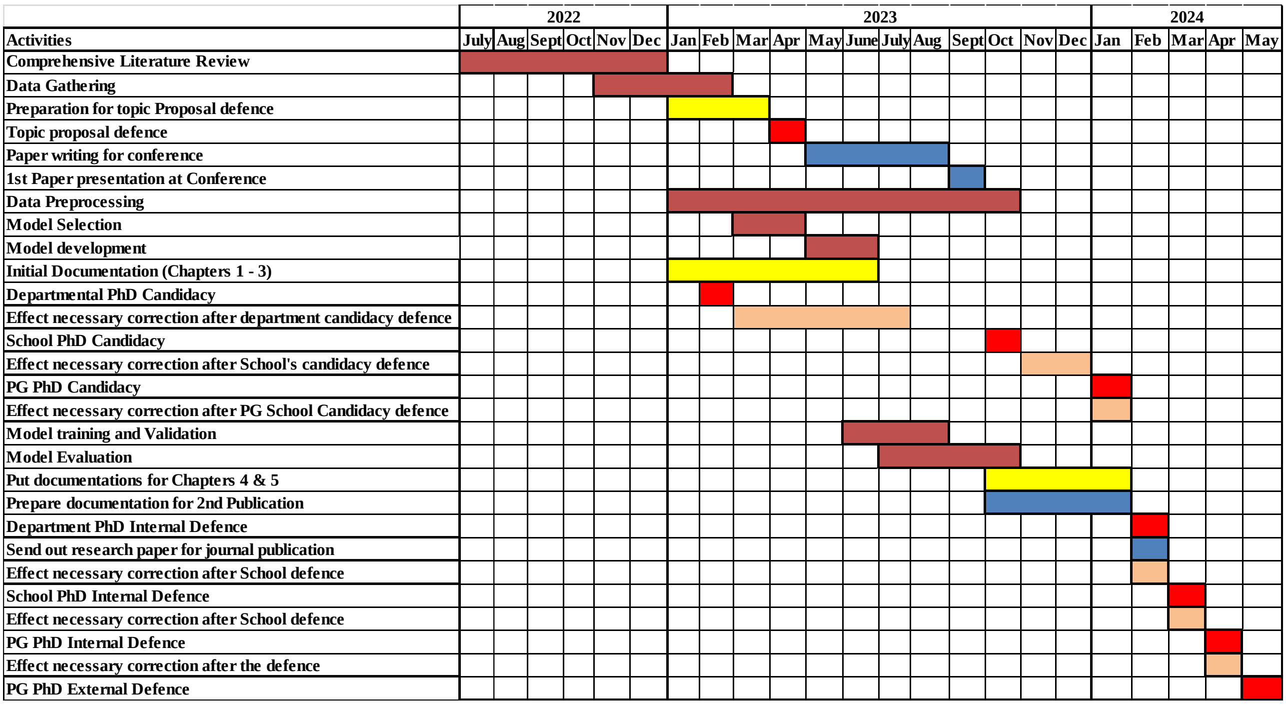

Figure C.1: Proposed PhD dissertation timeline

Figure C.1 illustrates the proposed timeline outlined in a structured plan for conducting a comprehensive literature review, gathering data, preparing for and defending the topic proposal, as well as subsequent milestones such as paper writing for conference presentations and PhD candidacy defenses at various levels. The timeline also includes crucial activities like data preprocessing, model selection, development, training, and validation, followed by documentation preparation for chapters and publications. Each activity is strategically distributed over the specified months across the years 2022 to 2024, ensuring a systematic progression toward the completion of the research work. Additionally, necessary corrections and defenses at departmental, school, and postgraduate levels are accounted for, indicating a thorough and rigorous approach to the PhD research process. This detailed plan reflects a commitment to academic excellence and scholarly rigor, laying the groundwork for a successful completion of the proposed study.

APPENDIX D

LIST OF PUBLICATIONS

1. Dokun, O., John-Otumu, A. M., Ikerionwu, C., and Eze, U. F., Advancing COVID-18 Prediction with Deep Learning Models: A Review, IEEE 2nd FCAS International Conference on Sustainable Development Goals (FICSDG), Abuja, Nigeria, 12 – 15th September, 2023.
2. Dokun O., John-Otumu, A. M., Ikerionwu, C., Eze, U. F., Etus, C., Nwanga, E. M., and Okonkwo, O. T. (2024), Deep Learning Model for COVID-19 Classification using Fine Tuned ResNet50 on Chest X-ray Images, Machine Learning Research, Science Publishing Group. (Accepted for Publication April 2024).