2021-09

IDENTIFICATION OF COVID-19 FROM CHEST RADIOGRAPHIC IMAGES USING DEEP LEARNING

MINICHIL, ABOYE ALEMNEH

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND GRADUATE STUDIES

FACULTY OF COMPUTING

IDENTIFICATION OF COVID-19 FROM CHEST RADIOGRAPHIC IMAGES USING DEEP LEARNING

BY

MINICHIL ABOYE ALEMNEH

BAHIR DAR, ETHIOPIA

SEPTEMBER, 2021

IDENTIFICATION OF COVID-19 FROM CHEST RADIOGRAPHIC IMAGES USING DEEP LEARNING

BY

MINICHIL ABOYE ALEMNEH

A Thesis Submitted to the School of Research and Graduate Studies of Bahir Dar Institute of Technology, BDU in Partial Fulfillment for the Degree of Master of Science
in Information Technology in the Faculty of Computing

ADVISOR: SEFFI GEBEYEHU (ASSISTANT PROFESSOR)

BAHIR DAR, ETHIOPIA

SEPTEMBER 2021

DECLARATION

This is to certify that the thesis prepared by Minichil Aboye, titled: Identification of COVID-19 from Chest radiographic Images using Deep learning and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Information Technology complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

I, the undersigned, declare that the thesis comprises my own work. In compliance with internationally accepted practices, I have acknowledged and refereed all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation / fabrication of any idea/data/fact/source will constitute sufficient ground for disciplinary action by university and can also evoke penal action from the sources which have not been properly cited or acknowledged.

Name of the student: Minichil Aboye Signature________

Date of submission: 10/04/2021

Place: Bahir Dar

This thesis has been submitted for examination with my approval as a university advisor.

Advisor Name: Seffi Gebeyehu (Assistant Professor)

Signature: ________

BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTING
Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student: Minichil Aboye Alemneh Signature ________ Date 10/04/2021

As members of the board of examiners, we examined this thesis entitled "Identification of Covid_19 from Chest Radiographic Images by Using Deep Learning" by Minichil A. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of science in "Information Technology".

Board of Examiners

Name of Advisor	Signature	Date
Seffi G (Assis. prof)		10/5/2021
Name of External examiner	Signature	Date
Adane Letta (Phd)		10/04/2021
Name of Internal Examiner	Signature	Date
Dagnachew M (Assi.prof)		10/05/2021
Name of Chairperson	Signature	Date
Abrham D (Ass. prof)		10/05/2021
Name of Chair Holder	Signature	Date
Derejaw L.		10/5/2021 G.C
Name of Faculty Dean	Signature	Date
Asegahegn E.		Oct 05 - 2021

Faculty Stamp

ABSTRACT

Corona virus disease (COVID-19), which is caused by the severe acute respiratory syndrome corona virus 2 (SARS-CoV-2), is an inflammatory disease that causes respiratory illness (similar to influenza) with symptoms such as cold, cough, and fever, as well as difficulty breathing in more severe cases. The goal of the research is to develop a model that can identify COVID-19. The information is gathered from hospitals. The visual data was analyzed using a deep learning approach (CNN).

Preprocessing, feature extraction, and classification are the three components of the research. We normalize the image to a standard size during image processing. For feature extraction, we employ a convolutional neural network. It is used to identify and choose essential elements that contribute to the disease's symptom. We employ a convolutional neural network for classification. For classifying into a specific class (normal/no findings, pneumonia, and COVID-19), a three-way Softmax is employed. The research was carried out in Python using Keras (with TensorFlow as a backend) and evaluated on a sample image dataset obtained from Tibebe Gion Hospital.

The model achieved a diagnosis accuracy of 98.35% for training and 98% for testing to identify Covid-19. This research work presented different contributions that can be further improved or implemented on the effort to detect and grade related diseases.

Keywords: Covid-19, Deep Learning, CNN, Feature Learning, SoftMax

ACKNOWLEDGMENT

First of all, I would like to thank the Almighty God, for He is the Source of Wisdom, and the Beginning of Knowledge. I praise and thank Him for His great help throughout this work. I would also thank the ever-Vergin, St. Marry, Mother of our Lord.

My deepest gratitude is to my advisor, Seffi Gebeyehu (Assistant Professor). I am deeply thankful to him for giving his insights and constructive comments. Discussions with him always made me think that things are possible. I expect one day I would become as good advisor as him and guide my students as he has been to me.

My deepest gratitude also goes to Mr. Seffi Gebeyehu, for his guidance at the time of problem formulation and proposal writing. I would also thank Dr. Tesfa T., for his teaching during Research Design and Methodology course. He helped a lot in motivating us to do our work with in the schedule. I am also thankful for the constructive comments provided from Mr. Abrham Debasu during my progress report presentation.

My sincere thanks also go to staff Tibebe Gion Hospital, Bahir-Dar, who have provided a collection of pictures. A special thanks goes to Sintayehu Alemu for his kind sharing of the pictures and providing constructive ideas about the identification and grading of Covid-19.

I would also thank my brethern, Sintayehu Alemu (pulmonologist), Yaregal Nigatu, Animut Aboye, Amare Aboye, Workneh Alamrie, and all my sisters who has spent many hours for discussion to make things goes right and within time. Finally, I would like to thank my father Mr. Aboye Alemneh, and my mother, Mrs. Gedamnesh Endalew and all the rest of my families and friends, who in one or another brought me up to success in my academic endeavor. None of this would have been possible without the love and patience of my family and friends. Their support and care help me to stay focused on my graduate study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
CT	Computed Tomography
CONV	Convolution Operation (Layer)
CXR	Chest Radiography
GGO	Ground Glass Opacities
GPU	Graphics Processing Unit
IDE	Integrated Development Environment
ILSVC	ImageNet Large Scale Visual Recognition Challenge
ReLU	Rectifying Linear Unit
RGB	Red Green Blue
ROI	Region of Interest
RT-PCR	Reverse Transcription, Polymerase Chain Reaction
SARS	Severe Acute Respiratory Syndrome
VGG	Visual Geometry Group
WHO	World Health Organization

Chapter One: Introduction

1.1. Background

Corona virus disease (COVID-19), which is caused by a virus termed severe acute respiratory syndrome corona virus 2 (SARS-CoV-2), is an inflammatory disease that causes respiratory ailment (like influenza) with manifestations, for example, cold, cough and fever, and in progressively serious cases, the problem in breathing. The World Health Organization (WHO) has already declared the outbreak as a Public Health Emergency of International Concern and subsequently as a pandemic. COVID-19 is primarily transmitted from symptomatic people to others who are in close contact through respiratory droplets, by direct contact with infected persons, or by contact with contaminated objects and surfaces. COVID-19 is firstly reported in China on Dec 31, 2019 (WHO, 2021). Although China launched an emergency response early in the outbreak, the infection rapidly spread to metropolitan areas in China and around the world. afterward, the disease was exported to many countries. Consequently, many countries have declared total lockdown and asked their population to stay indoors and strictly avoid gatherings. The USA, India, Brazil, UK and Russia are the top five countries that are infected by COVID-19 (Johns Hopkins, 2020).

The main step in the fight against COVID-19 is the effective screening of infected patients, such that those infected can receive immediate treatment and care, as well as the infected patients can be isolated to mitigate the spread of the virus. One of the most widely used laboratory methods for detecting coronaviruses is known as time reverse transcription–polymerase chain reaction (RT-PCR)(WHO, 2021). The test is done on respiratory samples of the patient and the results can be available within few hours to 2 days.

An alternative method to real-time RT-PCR is based on chest radiography images, where chest radiography imaging (chest X-ray (CXR) or computed tomography (CT) imaging) is conducted and analyzed by radiologists to look for visual indicators associated with SARS-CoV-2 viral infection. Various researchers found that the lungs of patients with COVID-19 symptoms have some visual marks like ground-glass opacities (GGO), hazy darkened spots, bilateral abnormalities, and interstitial abnormalities that can differentiate COVID-19 infected patients from non-COVID-19 infected ones. CT findings are observed over a long interval after the onset of symptoms, and patients usually have a normal CT in the first 0–2 days. In a study on lung CT of patients who survived COVID-19 pneumonia, the most significant lung disease is observed ten days after the onset of symptoms (Pan et al., 2020). The researchers believe that a chest radiology-based system can be an effective tool in the detection, quantification and follow-up of COVID-19 cases (Huang et al., 2020).

Deep Learning, which is the state-of-the-art research area of AI, enables the creation of end-to-end models to achieve promised results using input data, without the need for manual feature extraction. Deep Learning and it's variant the Convolutional Neural Network (CNN) techniques have been successfully applied in many medical problems such as arrhythmia detection(Yıldırım et al, 2018), skin cancer classification (Esteva et al., 2017), breast cancer detection (Esteva et al, 2017), brain disease classification (Talo et al., 2019), pneumonia detection from chest X-ray images (Ouchicha et al., 2020), fundus image segmentation and lung segmentation (Vidal et al., 2021). Similarly, in this era of COVID-19 many researchers have published different kinds of papers demonstrating deep learning approaches for COVID-19 detection from chest radiography images (Mahmud et al., 2020).

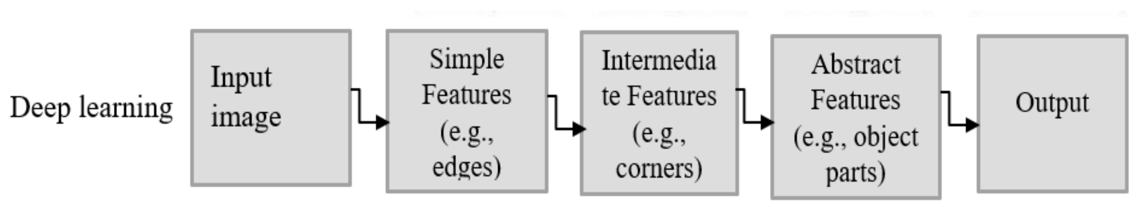

Figure 1.1: Deep Learning Feature Extraction Mechanisms

Deep learning, and explicitly CNN, follow the different way to extract features from images. CNN automatically learns features from the training process. In deep learning, problems are understood in terms of a hierarchy of concepts. As the network learns to classify images into some predefined categories, each layer learns to identify the features that are necessary to do the final classification (Liu et al., 2020). Lower-level layers are used to recognize low order features and higher-level layers combine low-level features into high order features. Finally, the output layer is used to classify the image and obtain the output class label. The great benefit of deep learning and CNN is that it reduces the need of handcrafted feature extraction algorithms and instead focus on process of training our network to learn these filters. However, training a network to get higher accuracy on a given image dataset is not an easy task. Figure 1.1 above simplifies what we have said.

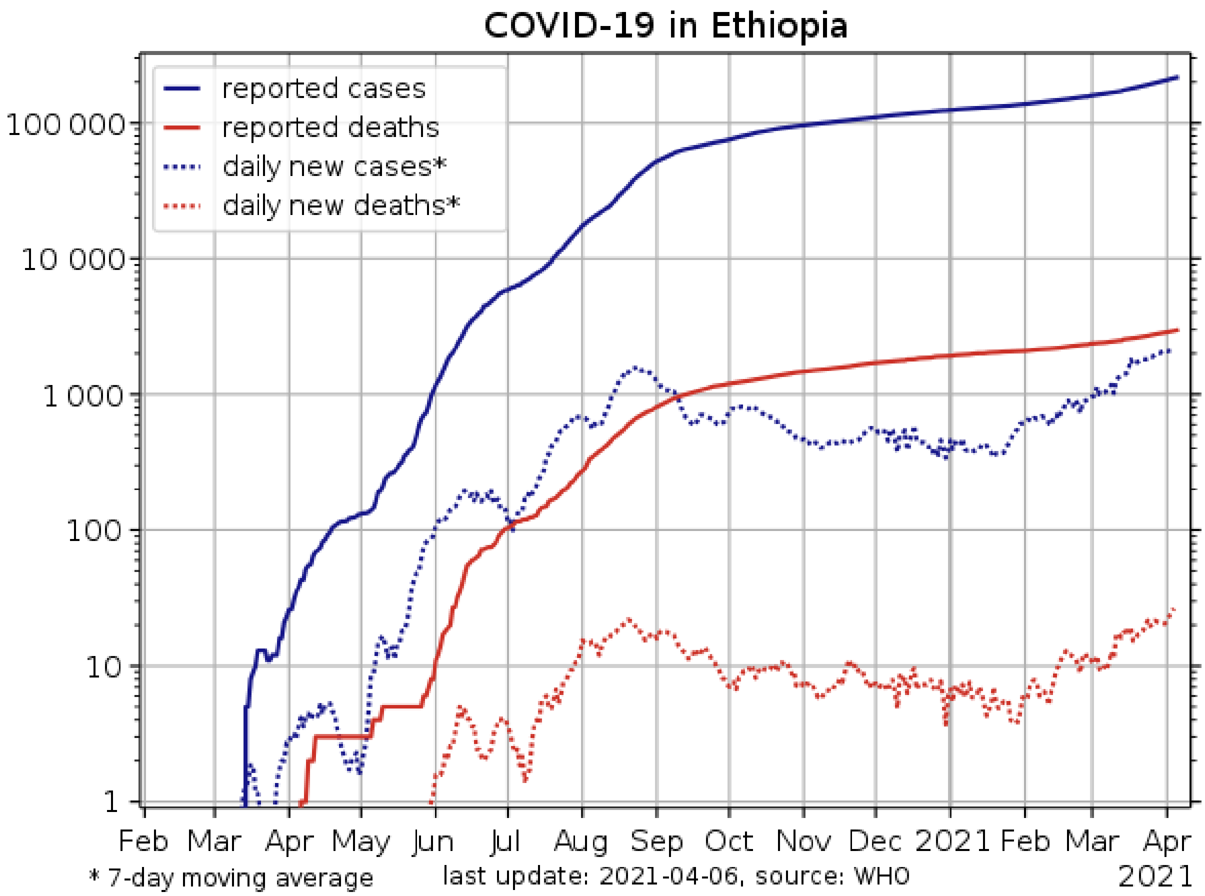

Figure 1.2: Causes of COVID-19 in Ethiopia (Anon, 2021)

1.2. Motivation

According to WHO, if additional resources are not urgently mobilized and efforts are not made, in the future, the global burden of death by Covid-19 could be increased rapidly. One way to avoid covid-19 is early identification of the disease through the current technology, deep convolutional neural networks, which achieve promising results in identifying diseases that consists of related or closed but different features.

Deep learning simplifies the complex feature extraction problem and hence will be used for image (having related features) recognition. Now-adays, by using deep convolutional neural networks, we can diagnose and identify different diseases which have very similar features. Hence, in this research we have given great emphasis on the identification of Covid-19, which is among the preventable causes of death. Noticeably, striving to identify Covid-19 according to its severity will help pulmonologist to apply the treatment based on the severity level of the disease.

The results of this research work would have a great impact on the upcoming efforts to identify the severity of covid-19, which will also be used as a benchmark for identifying and grading disease that affect other parts of our body. The model developed in this study will also overcome the gaps observed in the current related works.

In addition, this research could reduce the amount of time taken (for both patients and pulmonologist) and money spent to be diagnosed. The research work also enables the patients to have early diagnosis and to take treatments accordingly with the severity of covid-19. Consequently, death by covid-19 will be reduced. Due to this and the above reasons, we conduct this research work.

1.3. Statement of the Problem

There are several problems regarding the current COVID-19 screening method, that the previous works amid to uncover, most of the papers address that the rea-time polymerase chain reaction (RT-PCR) is the gold standard for corona virus diagnosis, but it is time-consuming to confirm COVID-19 patients. Due to the low RT-PCR sensitivity, even if negative results are obtained, symptoms can be detected by examining radiological images of patients. It has been reported in(Basturk et al., 2020), that some patients with positive chest CT findings may present with negative results of real-time RT-PCR for COVID-19. And also due to the rapid increasing growth rate of COVID-19 cases, the health system of many advanced countries has come to the point of collapse. In order to test the good complement of RT-PCR, a radiography examination can be conducted faster and have greater availability given the prevalence of chest radiology imaging systems in modern healthcare systems. Moreover, CT is a sensitive method to detect COVID-19, pneumonia, and can be considered as a screening tool with RT-PCR (Bhandari et al., 2021).

However, one of the biggest bottlenecks faced is the need for expert radiologists to interpret the radiography images and it is a challenge task to provide expert clinicians to every

hospital. The researchers advised to extend their work to more detailed covid-19 classifications. We consider their recommendations and try to design and develop a CNN model for identification of Covid-19 (i.e., Normal/no findings, Pneumonia and Covid-19). The approach that most papers used are machine learning techniques, which needs identification
of features by domain experts in order to reduce the complexity of the data and make patterns more visible to learning algorithms to work. They need human intervention only capable of what they are designed for. Those are shallow learnings which needs manual feature
extraction and trained on small amount of image datasets, which will be difficult to generalize
for large population (Patterson and Gibson, 2017).

It is due to the pattern of connections difference in machine learning that uses fully connected neural network and deep learning that uses CNNs(Patterson and Gibson, 2017), (Farnham, et al., 2017). In fully connected neural network, each unit is connected to all of the units (neurons) in the previous layer. This research work mainly uses deep convolutional neural network to learn high-level features from the data in an incremental manner. This study is intended to fill the aforementioned gaps. Finally, this study will answer the following research questions to attain the specific objectives.

- ❖ How to design a deep CNN model that able to identify covid-19 diseases?
- ❖ How to develop a prototype of the proposed model and increase its accuracies using validation phase?
- ❖ How to evaluate the proposed model using test dataset?

1.4. Objectives

1.4.1. General Objective

The general objective of the study is to develop a model that can identify COVID-19 diseases by using a chest radiographic image.

1.4.2. Specific Objectives

In order to achieve the general objective, the following specific objectives are identified.

- ❖ To collect images of chest radiographic images and preprocessing it.
- ❖ To designing CNN model that able to identify Covid-19 diseases.
- ❖ Developing a prototype of the proposed system and increasing its accuracy using validation phase.
- ❖ To test and evaluate the model using test datasets.
- ❖ To Select better Covid_19 disease identification state-of-the-art CNN architecture.

1.5. Scope and Delimitation

The main focus of this research is designing, modeling and prototype development of identification covid-19 by using deep convolutional neural networks. However, this research work does not include the recommendation of treatment at the time of medication.

1.6. Methods

In this study, we follow design science research methodology. It is a research paradigm where the creation of new artifact and evaluation of the artifact is a key contribution(Peffers et al., 2007). We use the process model designed by (Gengler et al., 2006). Following is the description of each phase: problem identification and motivation, objectives for a solution, design and development, evaluation, and communication.

1.6.1. Problem Identification and Motivation

In this phase, the research problem is defined and the value of a solution is justified. Problem definition is used to develop an artifact that provide a solution. Justifying the value of a solution helps to motivate the researcher and to understand the reasoning associated with the researcher's understanding of the problem (Suri et al., 2021). Various literatures are reviewed to acquire knowledge about the state of the problem and the importance of the solution. Research works that have been done to identify covid-19 diseases will be analyzed and evaluated to get an understanding of the various methods in identifying the disease. The gaps in related research works are analyzed and how we fill in the gaps are presented.

1.6.2. Objectives for a Solution

The objectives of a solution are inferred from the problem definition or specification. The objectives of the study that are inferred from the problem specification are explained.

Various resources have been reviewed to know the state of the problem, the state of current solutions and their efficiency.

1.6.3. Design and Development

In this section, the artifactual solution is created. This activity includes determining the artifact's desired functionality and its architecture and then creating actual artifact or model. Keras (using TensorFlow as backend) is used for designing the CNN model. Python is used for writing the required source codes.

We collected images of chest radiography with the help of pulmonologists from, Tibebe gion hospital. Images were taken from real patient's, which were attending treatments. Image datasets are divided into two parts: training dataset used to train the model and to increase the performance of the system through different parameters; testing dataset to evaluate the system. About 20,000 Chest radiographic images was collected. It will be divided in 70/30 format for training and testing respectively. Allocating 2/3rd of the dataset (Dobbin and Simon, 2011) for training is close to optimal for reasonable sized datasets (greater than 100 images).

Keras is a powerful library running on top of TensorFlow. It has its own graph data structure for handling computational graphs and communicating with TensorFlow. Keras has two main types of models to work with: sequential and functional. Sequential model is designed for simple architectures, where we just want to stack layers in a linear fashion. Functional model support more-general models with a diverse layer structure, such as multi-output models (Carlborg, 1976).

TensorFlow, in general terms, is a software framework for numerical competitions based on dataflow graphs (Farnham et al, 2017). In this graph, nodes represent operations (such as addition or division) and edges represent data (tensors) flowing around the system. Tensors are the standard way of representing data in deep learning. Simply put, tensors are just multidimensional arrays, an extension of two-dimensional tables or matrices to data with high dimensionality. RGB images are represented as tensors (three-dimensional arrays), with each pixel having three values corresponding to red, green and blue components.

Our system consists of three main components: preprocessing of images, feature extraction and classification into predefined classes. Classification, in turn, encompasses three main phases to identify covid-19. These are:

- Training phase: consists of a sequence of convolution, activation, pooling, fully-connected and dropout layers. Input to convolution layer is the original image. A set of K learnable filters are applied, where each filter has width and height and are square. Pooling layer is then applied to reduce the spatial size of the input volume and hence will reduce the number of parameters and amount of computation in the network. Finally, fully connected layers are placed at the end of the network before applying the Softmax classifier. Dropout is also applied to prevent overfitting by altering the network architecture at training time. It ensures that no single node in the networks is responsible to learn a pattern.
- Validation phase: different techniques such as data augmentation, dropout at early stages and batch normalization between convolutional layers are applied to better characterize or learn features and have higher accuracy. In this phase, our aim is to increase the performance of our model by increasing the accuracy or by decreasing the loss. Combination of parameters (weight and bias) that provide higher accuracy are used to classify testing datasets.
- Testing phase: images different from training datasets have been given to the learned model to evaluate how well the system responds to new datasets.

1.6.4. Evaluation

The developed system is evaluated to measure how well it supports a solution to the problem. To evaluate the system in a rational method, testing datasets were fed into the developed model. Subsequently, the model was evaluated by comparing its output against the observed data using precision, recall and f1-score values for evaluating diagnostic accuracy. In addition, we conducted a comparison with state-of-the-art models such as AlexNet (Krizhevsky et al., 2012), and GoogLeNet (Gliner et al., 2021) that are regularly used by previous works. State-of-the-art models are those that won ILSVRC (ImageNet Large Scale Visual Recognition Challenges) in different years.

1.6.5. Communication

In this section, the problems, the artifacts of the designed solution, the effectiveness and other related information are communicated to relevant audiences when appropriate.

1.7. Significance of the Study

A chest radiology image-based identification system can have a great role for both the medical and social communities. It can be fast, analyze multiple cases simultaneously, have greater availability and more importantly, such a system can be very useful in hospitals with no or the limited number of testing kits and resources. Moreover, given the importance of radiography in the modern health care system, radiology imaging systems are available in every hospital, thus making the radiography-based approach more convenient and easily available. Furthermore, CXR screening enables to accelerate the process of determining the priority of the COVID-19 suspected patients.

The method can be done in parallel with of RT-PCR testing to help relief the high volumes of patients as well as it can be an option especially when viral testing has low supplies. It can also be quite effective for triaging in geographic areas where patients are instructed to stay home until the onset of advanced symptoms, since abnormalities are often seen at time of presentation when patients suspected of COVID-19 arrive at clinical sites.

Portability is another advantage, the existence of portable CXR systems means that imaging can be performed within an isolation room, thus significantly reducing the risk of COVID-19 transmission during transport to fixed systems such as CT scanners as well as within the rooms housing the fixed imaging systems.

1.8. Organization of the Study

This section presents an overview on the contents of the remaining chapters. The rest of this thesis is organized as follows:

In Chapter Two, the literature review on the concept of covid-19 diagnosis and the approaches used for diagnosis are presented. A brief description of covid-19 diagnosis is discussed. In addition, a detailed description of the building blocks of CNNs are presented. Finally, the most successful CNN architectures which have been used in the field of image processing and pattern recognition are presented and a detailed analysis of different works related to the identification of covid-19 are presented. Only those works whose contributions are related to our work are discussed. In addition, the common gaps of the reviewed works and the way how we fill in the gaps are described exhaustively.

Chapter three, presents a detailed description of the proposed system is discussed. The components that compose the system (preprocessing, feature extraction, feature learning, and classification using CNN), and the responsibility of each component are described in detail.

In Chapter Four, the experimental evaluation of the proposed model for identification of covid-19 is described in detail. The dataset used and the implementation of the proposed model are described thoroughly. Finally, the test results are compared with the state-of-the-art models.

Chapter Five, summarizes the major findings of the research work. In addition, the major contributions of the proposed deep CNN model and the future works will be outlined.

Chapter Two: Literature Review

2.1. Introduction

In this chapter, thorough review of literature and analysis of related works are presented. Literature on the concept of covid-19 diagnosis and the approaches used for identification are discussed. We begin with a brief introduction of covid-19 diagnosis and identification and covid-19 control strategy. The steps in digital image processing and the approaches in pattern recognition are discussed thoroughly. Finally, a detailed description of the building blocks of CNNs is presented.

2.2. Covid-19 Diagnosis

According to (J. CARRIE et al., 2021)COVID-19 can be identified based on symptoms and then confirmed using RT-PCR or another nucleic acid testing of contaminated secretions. Chest CT scans, in addition to laboratory tests, can be useful in diagnosing COVID-19 in people who have a high clinical suspicion of infection. Serological tests may be used to detect a previous infection.

Pneumonia: is the leading cause of death among young children and one of the top mortalities causes worldwide. Pneumonia is detected usually by performing through examination of chest X-ray radiograph by highly-trained specialists. This process is tedious and often leads to a disagreement between radiologists. Computer-aided diagnosis systems showed the potential for improving diagnostic accuracy. In work(Varshni et al., 2019), the researchers develop the computational approach for pneumonia regions detection based on single-shot detectors, squeeze-and-excitation deep convolution neural networks, augmentations and multi-task learning. The proposed approach was evaluated in the context of the Radiological Society of North America Pneumonia Detection Challenge, achieving one of the best results in the challenge(Varshni et al., 2019).

COVID_19: Early studies show patients have abnormalities in chest radiography images that are characteristic of those infected with COVID-19, with some suggesting that radiography examination may be used as a primary tool for COVID-19 screening in epidemic areas. The need for expert radiologists to interpret the radiography images creates a bottleneck due to the subtle visual indicators present in the images. While radiography exams can be performed faster and are more accessible due to the prevalence of chest radiology imaging systems in modern healthcare systems, the need for expert radiologists to interpret the radiography images creates a bottleneck due to the subtle visual indicators present in the images (J. CARRIE et al., 2021). While many artificial intelligence (AI) systems based on deep learning deliver promising results in terms of accurately detecting COVID-19 through radiography imaging, the majority (if not all) of these are closed source and inaccessible to the scientific community and the general public to build upon for a deeper understanding and extension of these systems. The initial network design prototype aims to make one of the following three predictions, such as no findings/normal, pneumonia, and COVID-19 viral infection. These predictions were chosen because the findings will help clinicians determine not only who should be prioritized for PCR testing for COVID-19 case validation, but also which treatment approach to use based on the cause of infection, as each form of infection needs a different treatment plan.

2.3. Pathogens of COVID_19

According to (J. CARRIE et al., 2021) Emerging infectious diseases/pathogens, according to the National Institute of Allergy and Infectious Diseases, are those that have recently arisen in a population or have existed but are increasingly growing in incidence or geographic range. Many of the most harmful emerging pathogens are pathogenic viruses. The length of time that these viruses stay on surfaces will affect disease transmission. The coronavirus that causes COVID-19, SARS-CoV-2, is a pathogenic virus. Since emerging viral pathogens are less common and unpredictable than proven pathogens, few, if any, EPA-registered disinfectant product labels suggest using them against this group of infectious agents. As a result, in 2016, the Environmental Protection Agency (EPA) created a two-stage voluntary mechanism to allow the use of some EPA-registered disinfectant products against emerging viral pathogens not specified on the product label. Based on previous EPA-approved claims for harder-to-kill viruses, a company may apply for an emerging viral pathogen claim even before an outbreak occurs.

2.4. COVID_19 Control: Safe Strategies

As (J. CARRIE et al., 2021) describes , it has been months since COVID-19 threw our lives into disarray. We've gotten used to wearing masks, keeping a safe distance from others, regularly washing our hands, and working and learning from afar. But what do we know about COVID-19 infection prevention? Scientists, physicians, and public health authorities are also trying to figure out how the virus spreads, how to prevent it, and how to handle it effectively. New research may lead to guidance that contradicts what we've previously been told, and keeping track of it all can be difficult. Fortunately, there is still plenty of sound advice available. Being continuously alert and taking precautions, such as wearing a mask and physically distancing, which can be physically and emotionally uncomfortable, says Yale Medicine infectious disease specialist Jaimie Meyer, MD, MS. “However, maintaining these habits is critical to containing this pandemic, particularly before a vaccine is available” Furthermore, the colder weather is taking more of us indoors, which is riskier than being outdoors because there is less ventilation and keeping people 6 feet apart can be more difficult. Furthermore, according to Dr. Meyer, SARS-CoV-2, the virus that causes COVID-19, could be airborne, making ventilation even more important. Seasonal respiratory viruses, such as colds and flu, are expected to arrive in the coming months, raising concerns of a "twin emic" that could overwhelm health-care services already stretched thin by COVID-19. Since the symptoms of these other illnesses are so similar to those of COVID-19, they may confuse.

Meanwhile, COVID-19 has claimed the lives of over 210,000 people in the United States. As we move into the fall season after a hectic spring and summer, it's a good idea to check in with Yale Medicine experts and revisit the standard and most recent advice on how to stay healthy(J. CARRIE et al., 2021).

Wearing your mask: COVID-19 can be prevented from spreading to others by wearing a mask that protects the mouth and nose. Recent research indicates that masks may actually benefit the wearer by providing some form of infection protection. When you can't keep 6 feet away from everyone, the Centers for Disease Control and Prevention (CDC) suggests that anyone over the age of 2 wear masks in public places and with others who don't live in the same household. Masks should be made of two or three layers of washable, breathable fabric and should be able to fit snugly over your face. Stay socially distant: COVID-19 is distributed primarily among people who are within 6 feet (about two arms' length) of one another for an extended period of time (at least 15 minutes). Coughing, sneezing, or talking by an infected person may spread the virus by releasing droplets from the mouth or nose into the air. People may be asymptomatic and spread the virus without even realizing it, which is why it's important to keep a safe distance of 6 feet from others, whether inside or outside. Plus, the more people you communicate with at a meeting and the longer you spend interacting with each of them, the more time you spend interacting with them, the higher your risk of becoming infected with the virus by someone who has it (J. CARRIE et al., 2021).

Keep washing your hands: Handwashing is also an essential part of avoiding COVID-19 infection. The CDC recommends washing your hands with soap frequently, particularly after being in a public place or after blowing your nose, coughing, or sneezing. According to the CDC, you can wash your hands for at least 20 seconds and clean the backs of your hands between all fingers, under all fingernails, and up to the wrist. After washing, make sure they are fully dry (using an air dryer or a paper towel) and don't touch the sink, faucet, door handles, or other items. If soap isn't available, use a hand sanitizer that contains at least 60% alcohol and rub it on your hands until they're dry. Though the CDC believes that the virus is transmitted mainly by close person-to-person contact, it is possible to contract COVID-19 by touching a virus-infected surface or object and then touching your own mouth, nose, or eyes. As a result, you can wash your hands after touching something that may be tainted, such as a public banister or door handle, and before touching your face. The virus can live for a short time on some surfaces, but it is unlikely to spread via mail, products, or packaging, according to the CDC. Similarly, the risk of contamination from food (whether prepared at home, in a restaurant, or as takeout) is considered to be very low, as is the risk from food packaging or bags. Even though there is still a lot we don't know about the virus, it's still a good idea to wash your hands thoroughly after handling any food or items that come into your house (J. CARRIE et al., 2021).

Keep holiday gatherings small: Holidays put many families together in the fall and winter. This can be particularly difficult for those of us who live in areas where it would be difficult to gather outside. “Families might be less likely to do a group Zoom call after months apart during this pandemic,” says Dr. Meyer. “This may be the year we have to get imaginative and reconsider how we celebrate together.” According to Dr. Meyer, this may mean further holiday preparations. Think about quarantining for 14 days prior to the event and/or getting everyone screened for COVID-19 if tests are available in your community," she advises. "If at all possible, hold events to a minimum-perhaps only immediate family and close friends. Encourage the visitors to wear masks inside if it isn't safe to be outside. Spread out the food and dining areas so that people are isolated while eating while keeping their masks down(J. CARRIE et al.,2021).

Dine out carefully: even though many restaurants offer outdoor dining, which experts claim is the better alternative, a recent CDC study found that adults with COVID-19 infections were twice as likely as those without an infection to have visited a restaurant in the two weeks before their illness. The study made no distinction between indoor and outdoor dining, nor did it take into account social distancing or the use of a mask. (Those infected with COVID-19 were more likely to eat at restaurants where few other people wore masks or were socially isolated.) Dr. Ogbuagu points out that if you are dining with others at a restaurant and sharing tables while eating, which does not allow for sufficient social distancing and mask use, it offers opportunities for the virus to spread from person to person. With each additional person you come into contact with, your chances of spreading infection increase, particularly when people congregate(J. CARRIE et al., 2021).

Travel safely: Though you should try to avoid traveling, if at all possible, the CDC recommends staying at home to avoid COVID-19. It is often unavoidable. However, you should check to see if the virus is spreading at your destination before you leave. More cases of the virus at your destination raises the chances of catching it and spreading it to others. On the CDC website, you can see the weekly number of cases for each state. Don't forget to review the quarantining or monitoring laws at your destination or when you return home, Dr. Meyer adds. There are safety precautions you should take if you're traveling by car, plane, bus, or train. The CDC has a comprehensive list of guidelines for each mode of transportation that largely follows the advice above of social distancing, wearing a mask, and washing hands, but also provides unique advice for different scenarios. Differentiate between flu, colds, and COVID-19: Many people may have trouble distinguishing between the flu, a common cold, and COVID-19, which both have similar symptoms. Fever, shortness of breath, exhaustion, headache, cough, sore throat, runny nose, muscle pain, or body aches, as well as vomiting and diarrhea, are all signs of COVID-19 and the flu (though these last two are more common in children). Colds, on the other hand, are typically milder than the flu and are associated with a runny or stuffy nose. COVID-19, on the other hand, is associated with a loss of taste and smell. So, what do you do if you or someone in your family experiences any of these symptoms? First and foremost, you should keep as far away from others as possible and wash your hands before coming into contact with your face, And, of course, if you have severe symptoms including a high fever or shortness of breath, see a doctor or go to the hospital. Otherwise, a COVID-19 test at a nearby testing center will help you determine what kind of respiratory disease you have and how to notify people you've come into touch with. Parents should contact their children's pediatricians about these symptoms, since otherwise, their children would possibly be unable to return to school(J. CARRIE et al., 2021).

Seek routine medical care: You should continue to pursue the appropriate medical attention or procedures, whether normal or emergency. Many health centers and physicians already provide telehealth (video or phone) appointments, and most have procedures in place to reduce the risk of coronavirus infection. It's important to receive emergency treatment when you need it. Pediatric and adult doctors recorded fewer emergency department visits earlier in the pandemic, raising concerns that patients were avoiding treatment out of fear of contracting COVID-19(J. CARRIE et al., 2021).

Be mindful of your mental health: As a result of the pandemic's tension and uncertainty, many people are experiencing anxiety, depression, and other mental health problems. All of this is natural, according to mental health experts, who suggest allowing yourself to tolerate all feelings, including the negative ones, in order to better control them. If the events of the world are too much for you right now, experts recommend practicing mindfulness (even only breathing exercises), eating well, and staying physically active. Parents can support children who are still adjusting to a lack of play dates, cancelled events, and new school schedules by thoroughly listening to their concerns and offering age-appropriate responses to their questions. Parents will be able to decide whether additional emotional support is needed by communicating with children about what they know and how they are doing(J. CARRIE et al., 2021).

Watch your weight: Some people may be gaining weight at a time when habits are interrupted and many people work from home, where snacks are readily accessible (the so-called quarantine 15). Yale Medicine doctors advise that you concentrate on keeping a balanced diet, having enough exercise, getting enough sleep, and finding healthy ways to relieve stress now more than ever(J. CARRIE et al., 2021). Obesity, on the other hand, is emerging as a separate risk factor for serious COVID-19 disease, also in younger patients. Obesity was found to be twice as likely to need hospitalization and much more likely to require emergency care than those without it, according to one study of COVID-19 patients under the age of 60. This is important, given that an estimated 42 percent of Americans are obese (defined as having a BMI of 30 or higher). balance the tension.

Keep up the good (safety) work: COVID-19 is likely to remain with us for a while. "But there is a light at the end of the tunnel with good efforts to continue to pursue public health policies to protect each other, and, hopefully, a better vaccine in the future," Dr. Ogbuagu says. COVID-19 is a preventable disease, according to Dr. Meyer, even before a safe and effective vaccine is available. "What it takes is for all of us to put in the effort to practice the behaviors mentioned above in order to keep our neighborhoods safe and healthy"(J. CARRIE et al., 2021).

2.5. Digital Image Processing

An image can be defined as a two-dimensional function, g (x, y), where x and y are spatial coordinates, and the amplitude of g at any pair of coordinates (x, y) is called the gray level or the intensity of the image at that coordinate or point. Digital image processing (DIP) uses digital computers to process the image to make an image sharp, clean, detailed, and make them "look better"(Davros 2010). DIP involves low, medium, and high-level processes. Low-level processes are primitive operations such as noise reduction, contrast enhancement, and image sharpening. Medium-level processes include operations like segmentation in which the input is an image and its outputs are attributes extracted from images. High-level processes are used to "make sense" of a group of attributes (Ge et al., 2016).

Color information has gained an ever-growing meaning in digital image processing. A color image is treated as a vector function (generally with the three components). The range of the image function is a vector space also known as a color space. Color space (RGB, CMY, and HSI color space) is a mathematical model describing the way colors are represented as tuples of numbers.

The RGB color space is used for representing a color image on a monitor (additive color mixture). The CMY color space is used for printing a color image (subtractive color mixture) (Azad, Hasan, and K, 2017). The fundamental difference between color images and gray-level images is that in a color space, a color vector, which generally consists of three components, is assigned to a pixel of color image, while a scalar gray value is assigned to a pixel of a gray-level image (Azad et al, 2017). For a digital color image in the RGB color space, I, three vector components R, G, B are to be indicated for each image pixel (x, y):

$$I(x, y) = (R(x, y), G(x, y), B(x, y))^T = (R, G, B)^T \tag{2.1}$$

All vectors $(R, G, B)^T$ with integer components $0 <= R, G, B <= 255$ characterize one color in the RGB color space (Azad et al., 2017) as shown in Figure 2.1 below. A vector $q = (R, G, B)^T$ inside the color cube represents exactly one color, where R, G, B, are integers.

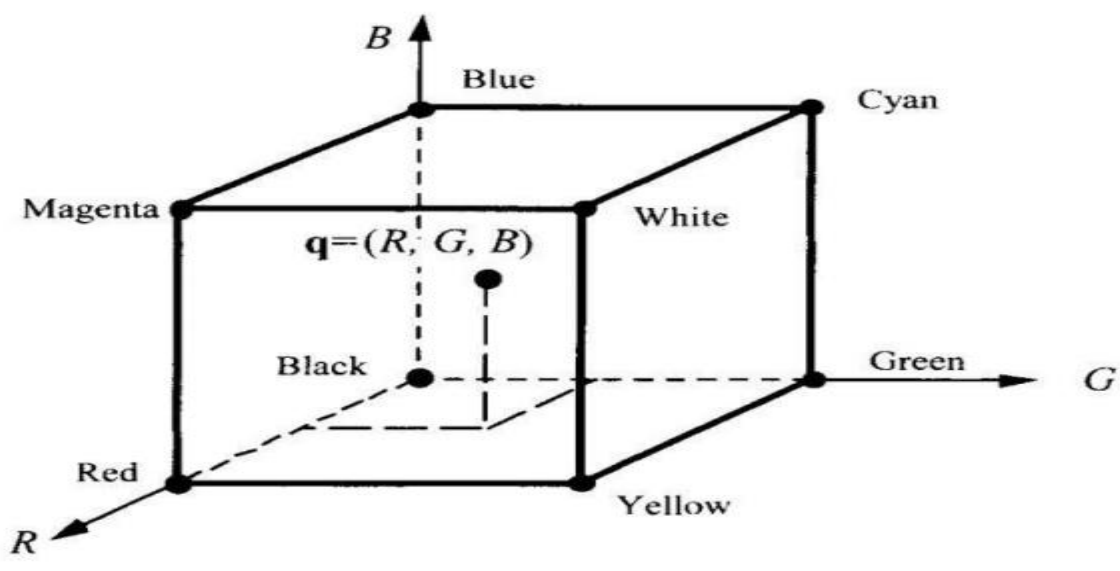

Figure 2. 1: RGB Color Space(CronJ, 2021)

RGB and CMY can be transferred into one another through:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 255 \\ 255 \\ 255 \end{pmatrix} - \begin{pmatrix} C \\ M \\ Y \end{pmatrix} \text{ and } \begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 255 \\ 255 \\ 255 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (2.2)$$

The necessary procedures of digital image processing include image acquisition, preprocessing, segmentation, feature extraction, and pattern recognition or classification.

2.5.1. Image Acquisition

Image acquisition is the first step for further image processing functions, unless no processing is possible. It is the process of taking an image using some hardware appropriate for taking a picture. Image can also be acquired from the database or another source tailored for research purposes. Most of the time an image taken is unprocessed and requires further processing and analysis to be used for specific purposes(P.K.Sinha, 2012).

2.5.2. Preprocessing

This step is mainly used to reduce the noise by smoothing or blurring the image. Image preprocessing used to improve image quality by suppressing unnecessary distortions or enhance features that will be used for further processes. Image enhancement and image restoration are the main activities in image preprocessing. Enhancement techniques are aimed to magnify details or highlight interesting features to better fit the image for further analysis. Image restoration also used to improve the appearance of an image based on mathematical or probabilistic models to restore degraded images(P.K.Sinha, 2012).

2.5.3. Segmentation

Image segmentation classify pixels until it is possible to extract objects or regions from the background (Cai et al., 2019). It is a process of isolating the objects of interest from the rest. Segmentation is the partitioning of an image into meaningful regions in order to differentiate foreground from background. Segmentation approaches are classified based on the features (include brightness, texture, and gradient) and the techniques (contextual and non-contextual) used (Davros, 2010). Non-contextual techniques (like intensity-based thresholding) ignore the relationships between features in the image. It uses pixels' gray value to categorize into a particular group. Contextual techniques, however, considers the relationships amongst the features in addition to the pixel gray value. It includes region-based techniques (such as region growing) and boundary-based techniques. The decision which approaches best segment regions can only be determined by experimentation with the specific image (Azad et al., 2017). The selection of an appropriate segmentation technique also depends on the type of images and applications (Cai et al., 2019).

Segmentation by thresholding: according to each pixel compared to the threshold value can be categorized to foreground if its value is greater than the threshold or a background otherwise. The success of thresholding depends on the careful and critical selection of the threshold value (Azad et al., 2017), (Cai et al., 2019). Thresholding is based on the assumption that clusters in the histogram correspond to either background or foreground (object of interest) that can be extracted by separating these histogram cluster. If the intensity distribution of foreground objects is quite different from the intensity distribution of background, it will be clear to apply thresholding for image segmentation (Cai et al., 2019). The two most commonly used thresholding techniques are optimal thresholding and adaptive thresholding. Optimal thresholding considers the histogram of an image to be a weighted sum of two or more probability densities, which corresponds to the intersection of the two normal distributions (Azad et al., 2017). The goal is to pick a threshold such that each pixel on each side of the threshold is closer in value to the mean of the pixels on that side of the threshold than the mean of the pixels on the other side of the threshold (Azad et al., 2017). Adaptive thresholding is used to segment an image with a variable background (Azad et al., 2017). In this case, it is impossible to segment an image with a single threshold value. It dynamically changes and applies the threshold over the image.

Segmentation by region-based methods: based on the similarity of the pixels (pixel similarity), it is used to produce connected regions that are as large as possible (to produce as few regions as possible). It grows the seed pixel into a region by adding neighbor pixels that have similar properties (brightness, texture, color, gradient, or geometric properties) with the seed (Azad et al., 2017). Region-based methods assume that neighboring pixels within the same region should have similar values (e.g., intensity, color, and texture) (Cai et al., 2019). An optic disc and optic cup features using region growing and mean shift algorithm was used for automatic detection of corona (Turkoglu, 2021). The region-based method may bias the segmentation in favor of the regions that are segmented first. However, they are generally better in segmenting noisy images where edges are extremely difficult to detect (Azad et al., 2017).

Segmentation by boundary-based methods: based on the differences of the pixels (pixel discontinuity), it is used to determine a closed boundary such that the foreground and the background can be defined (Azad et al., 2017). Edge detection, linking and boundary tracking are used in boundary-based methods. Boundary-based methods assume that the pixel properties, such as intensity, color, and texture should change abruptly between different regions (Azad et al., 2017), (Cai et al., 2019). It is liable to fail under conditions of high noise, when the boundary makes abrupt changes of direction which cannot successfully be tracked in this way. In this case, low-pass filtering is needed beforehand to reduce noise.

2.5.4. Feature Extraction[1]

Combining the existing features set into a smaller set of new, more informative features is called feature extraction. In feature extraction, a smaller set of features are computed which are more informative. It addresses the problem of finding the most informative set of features (Xu et al., 2010). Feature extraction is used to extract the descriptive features (can be color, shape or size feature sub-components) of images. Extracted features are used to interpret meaningful information from images. The goal of image feature extraction is to achieve a better classification rate by extracting new features to represent objects from raw pixel data(Cai et al., 2019).

Feature extraction aims to locate significant feature regions on images depending on their intrinsic characteristics and applications. These regions are distinguished by shapes, textures, sizes, intensities, and statistical properties. Feature extraction tends to identify the characteristic features that can form a good representation of the object, to discriminate across the object category with tolerance of variations (Cai et al., 2019).

Feature extraction by using discrete cosine transform (DCT): DCT is used to reduce the dimension of data to avoid singularity and decrease the computational cost. Feature extraction using DCT involves two steps. The first step involves the application of DCT to

[1] Feature extraction is not similar with feature selection in that the latter outputs the smaller set of features selecting k (out of N) important features, ignoring the remaining N-k features, but the premier outputs a new set of features. i.e., feature extraction outputs a new set of k- dimensions that are combination of the original N dimensions.

the entire image to obtain the DCT coefficients, whereas in the second step coefficients are selected to form feature vectors (Dabbaghchian et al., 2010).

Feature extraction by using histogram of oriented gradient (HOG): according to Maesen Churchill and Adela Fedor (Churchill and I n.d.), HOG-based feature extraction involves three steps: (i) Gradient computation: slant change of something between two points is called gradient. Gradient calculation requires filtering the color or intensity data of the image with the kernel [-1, 1, 0] or [-1, 0, 1]. (ii) Spatial and orientation binning: used to create cell histograms. Each pixel within the cell casts a weighted vote for an orientation based on the values found in step (iii). Depending on the gradient, histogram channels spread over 0-180 degree or 0-360 degree. (iv) Normalization and descriptor blocks: the slant change of something between two points must be locally normalized, which requires grouping the cells together into larger and spatially connected blocks.

2.5.5. Pattern Recognition

A pattern is any distinguishable representation or interrelation of data, events, and concepts. Patterns need to be apparent even when they are severely affected by noise (Cai et al., 2019). Classification or pattern recognition is a process to decide whether a new object belongs to a particular group or not, depending on whether its features fall inside the domain of that group or outside of the domain of that group, respectively (Azad et al., 2017). Given images that are typical examples of the number of classes (the training set), the classification task is to categorize or classify a new image (different from the training set) into one of these classes. The training images should have as much variety to have higher accuracy.

2.6. Approaches to Pattern Recognition[2]

2.6.1. Naïve Bayes

Consider the classification problem where sample x belongs to one of two classes, denoted as C1 and C2. Assume the priori probabilities P(C1), and P(C2) are known. The density function, P(Ci|x), is obtained by:

[2] The goal of recognition is to recognize or detect an object and make a (yes/no) decision. Classification, however, goes a step further by sorting objects into one of several groups or classes.

$$P(\text{Ci}|\text{x}) = \frac{P(\text{x}|\text{Ci})\text{P}(\text{Ci})}{\text{P}(\text{x})} \quad (2.3)$$

According to Bayes theory, the probability of the classification error can be minimized by the following rule:

$$x \text{ is classified to } C1, if\ P(C1|x) > P(C2|x)$$

$$x \text{ is classified to } C2, if\ P(C2|x) > P(C1|x) \quad (2.4)$$

Naïve Bayes assumes that the attribute values are conditionally independent to one another. It ignores the possible dependencies among the inputs.

2.6.2. Support Vector Machine

Support vector machine (SVM) separates a set of input pattern vectors into two classes with an optimal separating hyperplane (Cai et al., 2019). Its aims to separate the input patterns by maximizing the distance between the closest vectors to the hyperplane and by separating them without error. SVM produces the pattern classifier by applying a variety of kernel functions such as linear, polynomial functions as the possible sets of approximating functions.

Even though, SVM was originally designed to handle two-class classification, it was later expanded into multiclass classification problems. Different types of SVM classifiers are used depending upon the type of input patterns: a linear maximal margin classifier is used for linearly separable data, a linear soft margin classifier is used for linearly non-separable, or overlapping, classes, and a nonlinear classifier is used for classes that are overlapped as well as separated by nonlinear hyperplanes (Cai et al., 2019).

Linear maximal marginal classifier: used where the training data can be separated by a hyperplane, w * x + b = 0. The goal of SVM is to find the optimal values for w and b. After finding the optimal separating hyperplane, w_0. x + b_0 = 0, an unseen pattern, xt, can be classified by the decision rule for f(x) = sign (w_0. x + b_0). Each unseen pattern, xi, belonging as it does to one of two classes, has a corresponding value yi, where yi = {-1, 1}. Because the hyperplane is w. x + b = 0, the training data can be divided into two classes such that:

$$w.xi + b \geq 1, if\ yi = 1$$

$$w.xi + b \leq -1, if\ yi = -1 \tag{2.5}$$

Linear soft margin classifier: handle input patterns that are overlapping or linearly non-separable. Its objective is to separate the two classes of training data with a minimal number of errors. The linearly separable case in the above case (linear maximal margin classifier), can be rewritten as:

$$w.xi + b \geq 1 - ei, if\ yi = 1$$

$$w.xi + b \geq -1 - e, if\ yi = -1 \tag{2.6}$$

where ei, is non-negative slack variables.

Nonlinear classifier: kernel functions, such as polynomial functions are used to transform the input space to a feature space of higher dimensionality, when the input vectors cannot be linearly separated in the input space.

2.6.3. Artificial Neural Network

As (Patterson and Gibson, 2017) depicts a neural network consists of at least an input layer and an output layer. Some network architectures may include multiple hidden layers between the input and output layers. Each layer can have one or more nodes. Each neuron in the input layer is connected to every output neuron in the next layer.

Two operating phases, training and testing, are always encountered in neural networks. During the training phase, the neural network takes the training dataset as input and adjusts the connection weights to achieve the desired association or classification. During the testing phase, the neural networks is tested with the testing dataset (different from the training dataset) to retrieve corresponding outputs based on the knowledge discovered from the training phase.

A neural network has an input layer, one or more hidden layers, and a single output layer. Each layer can have a different number of neurons and each layer is fully connected to the adjacent layer. The behavior of neural networks is shaped by its network architecture. A network’s architecture can be defined in terms of:

- Number of neurons
- Number of layers
- Types of connections between layers

For the input layer, the input is the raw vector input. The input to neurons of the other layers is the output (activation) of the previous layer’s neurons. As data moves through the network in a feedforward fashion, it is influenced by the connection weights and the activation function type. The basic structure of the neural network is shown in figure 2.2 (taken from (Patterson and Gibson, 2017) below).

Input layer: shows how we get input data into our network. The number of neurons in an input layer is typically the same number as the input feature to the network. Input layers are followed by one or more hidden layers.

Hidden layer: There are one or more hidden layers in a feed-forward neural network. The weight values on the connections between the layers are how neural networks encode the learned information extracted from the raw training data. Hidden layers are the key to allowing neural networks to model nonlinear functions(Patterson and Gibson, 2017).

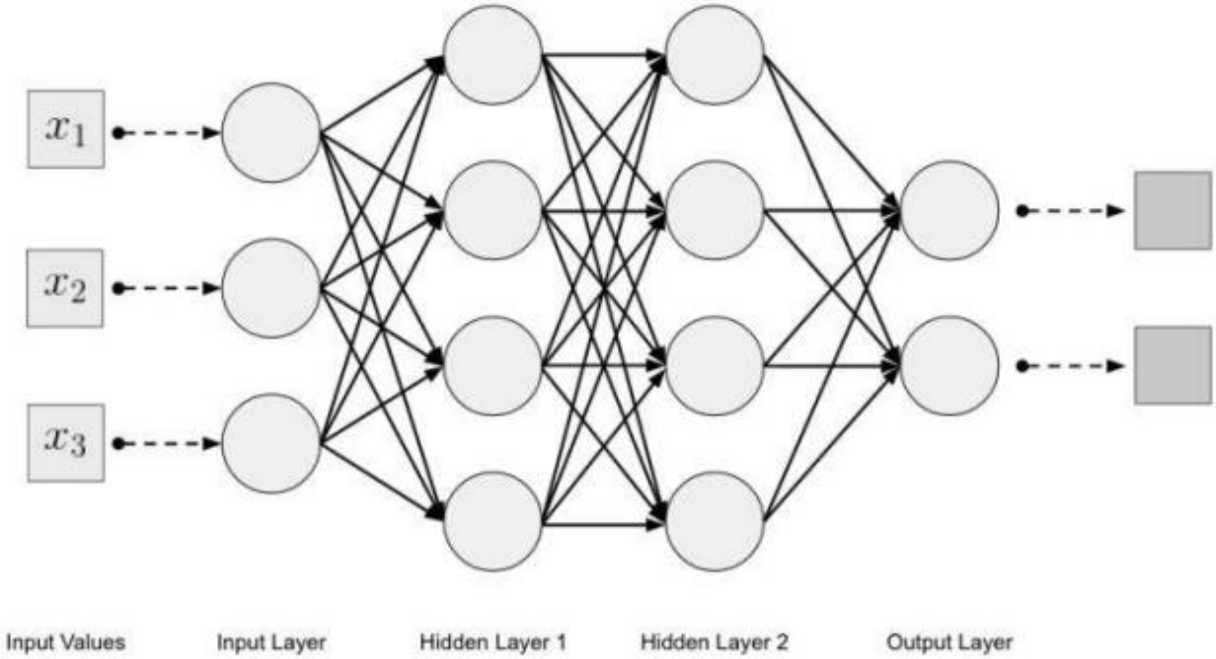

Figure 2. 2: Fully Connected Multilayer Feed-Forward Neural Network Topology(Patterson and Gibson, 2017)

Output layer: output (prediction or classification) of our model is answered from the output layer. The output layer gives us an output based on the input from the input layer. Depending on the setup of the neural network, the final output may be a real-valued output (regression) or a set of probabilities (classification). This is controlled by the type of activation function we use on the neurons in the output layer (Patterson and Gibson, 2017).

Connections between layers: In a fully connected feed-forward network, the connections between layers are the outgoing connections from all neurons in the previous layer to all of the neurons in the next layer. These weights are progressively changed as the algorithm finds the best solution with the backpropagation learning algorithm.

2.6.4. Convolutional Neural Network

In fully connected layers, each unit (neuron) is connected to all of the units in the previous layer. In CNN, however, each unit is connected to a small number of units in previous layer. In addition, all units are connected to previous layer in the same way, with the same weights and structure. CNNs use convolution operation instead of general matrix multiplication in at least one of their layers. Convolution is an element-wise multiplication of two matrices followed by the sum. The basic CNN architecture is given in Figure 2.3 below(García Reyes, 2013).

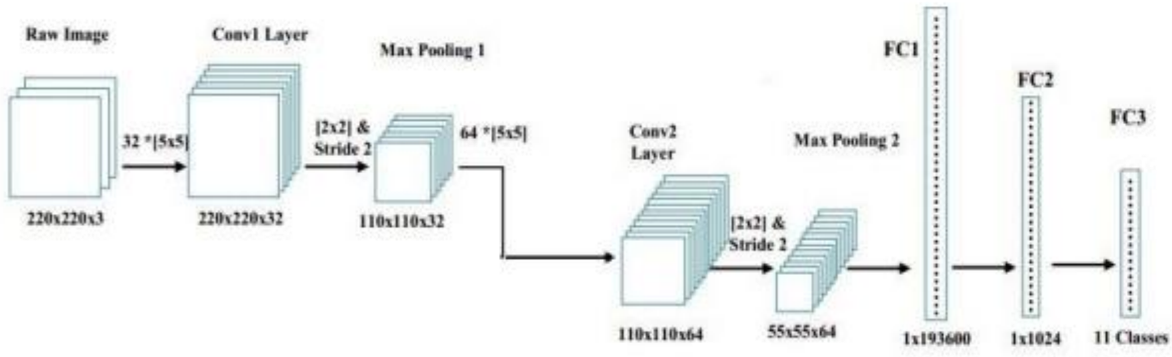

Figure 2.3: Basic CNN Architecture(García Reyes, 2013)

Nearly all machine learning and deep learning libraries use the simplified cross-correlation function as a convolution operation (García Reyes, 2013). Convolution (denoted by * operator) over a two-dimensional input image I and two-dimensional kernel K is defined as:

$$S\,(i,j) = (I * K)\,(i,j) = \sum m \sum n\; K(i + m, j + n)\, I(m,n) \tag{2.7}$$

The kernel K (tiny matrix) sits on the top of the big image I (big matrix) and slides from left-to right and top-to-bottom, applying a mathematical convolution operation at (x, y)-coordinate of the big image I. CNNs are able to learn kernels (filters) that can detect edges in the lower-level layers of the network and then use the edges as building blocks to detect high-level objects (features).

CNN give three key benefits: local invariance (via pooling layers), compositionality, and shared weights (García Reyes, 2013) (Carlborg, 1976). Local invariance is used to detect an image as containing a particular object regardless of its spatial location. When designing CNNs, it should be invariant of translation, rotation, scaling, and any other sorts. A CNN computes the same features of an image across all spatial areas (Farnham et al., 2017).

Compositionality is used to learn more rich features deeper in the network. Each filter composes a local patch of lower-level features into a higher-level representation. A network may learn edges from pixels, shapes from edges, and then complex features from shapes. This property gives CNN to be powerful in computer vision.

To move away from pixel representation in a row by gaining the ability to detect the same feature independently from the location where it is placed in the input image, a simple intuition is to use the same weights for all neurons in the hidden layers. Each layer will then learn a set of position-independent latent features derived from the image (Carlborg, 1976).

2.6.4.1. Building Blocks of CNN

Many layers are used to build CNNs such as convolutional, activation, pooling, fully-connected and dropout layers. Following is a detailed description of each layer.

Convolutional layers: considered as the core building blocks of CNN architectures. Convolution takes input (raw data or feature map output from another convolution), applies a convolutional kernel (filter), and outputs a feature map. Filters are small in terms of their spatial dimension (but extend throughout the full depth of the volume) as compared to the input image and are nearly always square. On images with large spatial dimensions, it is often impractical to connect a neuron to all other neurons in the previous layer. When implementing CNNs, we choose to connect each neuron to only a local region of the input volume. We call this local connectivity and the size of this local region the receptive field of the neuron(García Reyes, 2013).

If the receptive field is of size 5 x 5, then each neuron in the convolution layer will connect to a 5 x 5 local region of the image for a total of 5 x 5 x 3 = 75 weights (if the image has a depth of 3 (one for each RGB channel). Three parameters control the size of an output volume: the depth, stride, and zero-padding size (García Reyes, 2013). The set of filters that are looking at the same (x, y) location of the input is called the depth. The depth controls the number of neurons (filters) in the convolution layer that connect to a local region of the input volume. If the volume size is 8 x 8 x 32 and the receptive field size is of 5 x 5, then each neuron in the convolution layer will connect to a total of 5 x 5 local region of the image for the total of 5 x 5 x 32 = 800 connections to the input volume. Stride defines by how much we skip pixels in the x- and y-coordinates. It controls the spatial movement of the filter across the image or feature map. It introduces the size of skips in the application of the filter (Farnham et al., 2017). Smaller strides lead to overlapping receptive fields and larger output volumes, while larger strides lead to smaller output volumes. As shown in Table 2.1 the size of output volume is reduced much when using stride size of 2.

Table 2.1: Left: a 5 x 5 input image. Right: a 3 x 3 kernel (filter)

12	35	117	209	9
56	123	7	38	136
235	64	111	19	23
67	203	45	182	77
3	55	101	59	221

0	0	1
0	1	0
-1	0	0

Table 2.2: output of convolution with stride size of 1 (left) and 2 (right)

5	152	-64
4	-54	110
311	9	104

5	-64
311	104

Zero-padding mean padding borders of the original image with zero to keep the original image size when applying a convolution. Without zero-padding, the spatial dimensions of the input volume would decrease too quickly. Output of applying a 3 x 3 filter on zero-padded 5 x 5 image is shown in table 2.3 left while the output of applying 3 x 3 filter without zero-padding the original image is shown in the right. Zero-padding helps to preserve the spatial dimensions of the original 5 x 5 image(Farnham et al., 2017).

Table 2.3: Applying Zero-Padding (of size 1) to the Image on Table 2.1 left

0	0	0	0	0	0	0
0	12	35	117	209	9	0
0	56	123	7	38	136	0
0	235	64	111	19	23	0
0	67	203	45	182	77	0
0	3	55	101	59	221	0
0	0	0	0	0	0	0

Table 2.4: Effect of zero-padding on output image size

12	-21	-6	202	-29
91	5	152	-64	117
358	4	-54	110	-159
131	311	9	104	18
206	100	283	136	221

5	152	-64
4	-54	110
311	9	104

Activation layers: because no parameters are learned in activation layers, they are not technical layers. The output of the activation function is always the same as the input dimension since activation function is applied in an element-wise manner (García Reyes, 2013). After linear layers, it is common practice to apply nonlinear activation functions. Rectifying linear unit (ReLU) is the most widely used activation function in CNNs (Heaton, 2018). Deep CNNs that apply ReLU train several times faster as compared to applying other activation such as tanh. ReLUs do not require input normalization to prevent them from saturating (Krizhevsky et al, 2012). Activations are done in-place so there is no need to create a separate output volume. ReLU outputs the maximum of zero, and the number (image pixel). i.e., max (0, x). Table 2. 5 right below shows the output of applying ReLU to a 3 x 3 input volume (table 2.5 left).

Table 2.5: Application of ReLU

5	152	-64
4	-54	110
-11	-9	104

5	152	0
4	0	110
0	0	104

Pooling layers: used to progressively reduce the dimensionality of the input data and help reduce overfitting. This in turn helps to reduce the number of parameters and the amount of computation in the network. It takes a local receptive field and replace the nonlinear activation function at each portion of the field with the max or min or average function (Bharath Ramsundar and Zadeh, 2017). The most common type of pooling layer is max pooling, which is used in the middle of the CNN architecture to reduce the spatial size. If our pool size is 2 x 2 and we apply max pooling, then we keep only the maximum or the largest value in each 2 x 2 block regions and the height and width of the input volume will be reduced by a factor of 2. However, we can further reduce the dimensionality of the input volume by increasing the stride size. Table 2.6 below depicts the effect of max pooling with a stride size of 1 (top-right) and 2 (bottom-right) while applied on a 4x4 input volume (left).

Table 2. 6: Effect of Max Pooling on Different Stride Size

12	35	117	209
156	123	7	203
235	64	141	19
67	203	45	182

156	123	209
235	141	203
235	203	182

156	209
235	182

Applying a max pooling with large stride size (such as 2) will intensely reduce the spatial dimensions of the input volume, discarding 75 % of activations from the previous layer.

Fully-connected layers: neurons in this layer are fully connected to all neurons in the previous layer. They have normal parameters for the layer and hyperparameters. Fully connected layers are used to compute class scores that will be used as the output of the network (Patterson and Gibson, 2017). They are applied at the end of the network before applying the classifier (usually Softmax classifier).

Dropout: is a technique of turning off some fraction of units in a layer, by setting their values to zero during training and turning on these units by setting their values to one during testing. It is a regularization form used to force the network to distribute the learned representations across all neurons and hence force the network to learn a representation that works after the dropout (Farnham et al., 2017). Dropout reduces overfitting by explicitly altering the network architecture at training time. It ensures every single node to be activated when presented with a given pattern. Dropout avoids the dependency of the network on small units of neurons (García Reyes, 2013).

2.6.4.2. CNN Architectures [3]

Stacking different layers (convolutional, pooling, activation, or fully connected) in a particular pattern yields the CNN architecture. Following the successful CNN architectures which have been used in the field of image processing and pattern recognition are presented.

AlexNet

AlexNet won the contest in the ILSVRC-2012 to classify 1.2 million images into 1000 different classes by achieving a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry. The network, which has 60 million parameters and 650,000 neurons, consists of eight learned10 layers including five convolutional layers (some of which are followed by max-pooling) and 3 fully connected layers with the final 1000-way Softmax which produces a distribution over the 1000 class labels. The non-linear

---

[3] Parameter refers to weights and biases learned during training, whereas hyperparameter refers to parameters we want to tune that can not be learned during training.

[4]Learned layers try to learn some parameters such as weights.

ReLU activation was applied at the output of every convolutional and fully connected layer. The network's input is 150,528 (224 x 224 x 3) dimensional. The 1st convolutional layer filters the input image with 96 kernels of size 11 x 11 x 3 with a stride of 4 pixels. The 2nd convolutional layer takes as input the pooled output of the first convolutional layer and filters it with 256 kernels of size 5 x 5 x 48. The 3rd, 4th, and 5th convolutional layers are connected to one another without any pooling or normalization layers. The fully connected layers have 4096 neurons each.

VGG

VGG takes second place in the contest of the ILSVRC-2014 to classify 1.2 million images into 1000 different classes by achieving a top-5 test error rate of 7.3 % (Simonyan and Zisserman, 2015). VGG also secured first place in the localization competition. The network consists of 16 learned layers including thirteen convolutional layers with a filter size of 3 x 3, five pooling layers that follow some of the convolutional layers, and three fully-connected layers with the final 1000-way Softmax which produces a distribution over the 1000 class labels. All learned layers were applied with ReLU nonlinearity. Their main contribution was a detailed evaluation of networks of increasing depth using an architecture with very small (3 x 3) convolution filters, which are convolved with the input at every pixel with a stride size of 1. It also improves the prior-art by pushing the depth to 16-19 weight layers. Small convolutional filter size (such as 3 x 3) has gained great advantage over large convolutional filer size, such as 7 x 7 as used in (Krizhevsky et al., 2012) by reducing the number of parameters and incorporating more nonlinear rectification layers.

GoogleNet

GoogleNet won the contest in the ILSVRC-2014 to classify 1.2 million images into 1000 different classes by achieving a winning top-5 test error rate of 6.7 %, compared to 7.3 % achieved by the second-best entry (Gliner et al., 2021). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. Efficient deep CNN architecture named Inception was used. Its main idea is based on how an optimal local sparse structure in a convolutional vision network can be approximated and covered by readily available dense components.

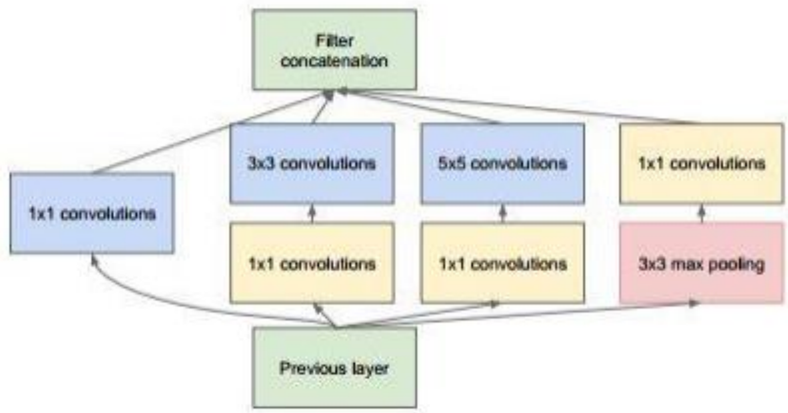

Figure 2.4: Inception Module with Dimension Reduction (source(Das et al., 2021))

The Inception module (as shown in figure 2.5) used 1 x 1 convolutions to compute reductions before the expensive 3 x 3 and 5 x 5 convolutions. In addition to being used as reductions, they also include the use of ReLU which makes them dual-purpose. GoogleNet has 22 learned layers and 5 pooling layers. It used average pooling instead of fully connected layers and it improved the top-1 accuracy by about 0.6 %. All convolutions including those inside the Inception module use ReLU non-linear activation.

ResNet

ResNet won the 1st place in the ILSVRC-2015 classification competition to classify 1.2 million images into 1000 different classes by achieving a winning top-5 test error rate of 3.57% (He et al., 2016). It also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation. Although, the ResNet is the deepest network (consists of 152 layers) ever presented on ImageNet, it still has lower complexity than VGG.

Deep residual networks (ResNets) make the training process faster and achieve higher accuracy compared to the equivalent neural networks by adding a simple skip connection parallel to the layers of convolutional neural networks. ResNets have shortcut connections parallel to their normal convolutional layers. Shortcut connections are those skipping one or more layers and their main function is executing an identity mapping without adding extra parameter and complexity, and their outputs are added to the outputs of the stacked layers.

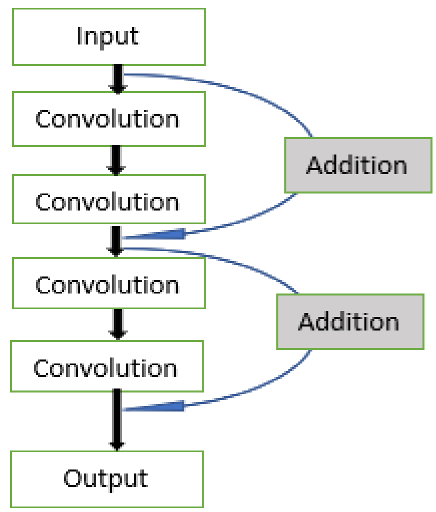

Figure 2.5: ResNet: A Building Block

$$Y = F(X, \{Wi\} + X) \tag{2.8}$$

$$Y = F(x, \{Wi\} + WsX) \tag{2.9}$$

ResNets can be defined as serial connection of multiple basic blocks as shown in figure 2.5 above. There are also shortcut connections parallel to each basic block and it gets added to its output. When the input and output dimensions are equal, the identity shortcuts (Eqn. (2.9)) can be directly used, otherwise, linear projection by shortcut connections (Eqn. (2.10)) is performed to match the dimensions.

2.7. Evaluation Technique

There are different performance metrics that have been used to evaluate the performance of the proposed solution or model. Among these, accuracy, precision, recall and f1-score are used extensively for measuring the performance of proposed solutions.

Accuracy: is the proportion of true positives (include both true positives and true negatives) against the whole population. Accuracy may mislead the quality of the model if the class is not balanced.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{P} + \text{N}) \tag{2.10}$$

Precision: is the proportion of true positives against the whole positives. Mathematically, it is expressed as:

$$\text{Precision} = \text{TP} / \text{P} \tag{2.11}$$

Recall or sensitivity: is the proportion of true positives against the whole true or correct data. It quantifies how well the model avoids false negatives (Patterson and Gibson, 2017). It is also known as true positive rate or hit rate.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \tag{2.12}$$

F1-score: is the weighted average the precision and recall. The relative contribution of precision and recall to the F1-score are equal.

$$\text{F1} - \text{score} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall}) \tag{2.13}$$

Micro-average, macro-average, and weighted-average for all the aforementioned performance metrics can also be calculated and used for additional analysis of results.

Macro-average precision or recall is just the average of the precision and recall (respectively) of the model on different classes.

$$\text{Macro} - \text{average precision} = (\text{P1} + \text{P2} + \dots + \text{PN}) / \text{N} \tag{2.14}$$

$$\text{Macro} - \text{average recall} = (\text{R1} + \text{R2} + \dots + \text{RN}) \tag{2.15}$$

Micro-average precision or recall is calculated by summing up the individual true positives, false positives and false negatives for each class.

$$\text{Micro} - \text{average precision} = \frac{\text{TP1} + \text{TP2} + \dots + \text{TPN}}{(\text{TP1} + \text{TP2} + \dots + \text{TPN}) + (\text{FP1} + \text{FP2} + \dots + \text{FPN})} \tag{2.16}$$

$$\text{Micro} - \text{average recall} = \frac{\text{TP1} + \text{TP2} + \dots + \text{TPN}}{(\text{TP1} + \text{TP2} + \dots + \text{TPN}) + (\text{TN1} + \text{TN2} + \dots + \text{TNN})} \tag{2.17}$$

2.8. Related Works

In this section, a detailed analysis of different works related to the identification of COVID-19 diseases using image processing, machine learning and deep learning techniques are presented. Only those works whose contributions are related to our work are discussed.

Most researchers have pointed out the challenges of identifying both corona virus and other respiratory diseases by taking some radiographical images. Some of the papers that related to this study are summarized in the table below:

Table 2.7: Related Works

No.	Authors	Research title	Methods	Recommendation
1.	Rajpurkar et al.	Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning	They develop an algorithm which detects pneumonia from frontal-view chest X-ray images at a level exceeding practicing radiologist. They also show that a simple extension of their algorithm to detect multiple diseases outperforms previous state of the art on ChestX-ray14, the largest publicly available chest X- ray dataset. Finally, they hope that the technology can improve healthcare delivery and increase access to medical imaging expertise in parts of the world where	

			access to skilled radiologists is limited	
2.	Asmaa Abbas' Mohammed M. Abdelsamea, Mohamed Medhat Gaber	Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network	The authors performed on chest X-ray images consisting of COVID-19, normal and viral pneumonia are presented to demonstrate the classification effectiveness of the hybrid 2D curvelet transform-CSSA-EfficientNet-B0 model proposed for the diagnosis of COVID-19.	The irregularities in annotated data remains the biggest challenge in coping with real COVID-19 cases from CXR images.

			They obtained the results for single EfficientNet-B0 model, combined model consisting of 2D curvelet transform and EfficientNet-B0, and hybrid 2D curvelet transform- CSSA-EfficientNet-B0 model is compared with the results of the studies in literature.	
3.	Ulhaq A, Khan A, Gomes D, Pau M	Computer vision for COVID-19 control	The authors discussed the role of AI and big data in fighting COVID-19. In addition to the existing deep learning architectures available for detection and diagnosis of COVID-19, they discussed the existing SIR (Susceptible, Infected and Removed) models and other deep models for identification, tracking, and out- break prediction.	They did not show any comparative quantitative analysis of the reviewed works.

			They also included different speech and text analysis methods and deep learning algorithms for drug repurposing, big data analysis-based outbreak prediction, virus tracking, vaccine, and drug discovery, etc.	
4.	Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS	Deep learning for visual understanding: A review	The authors reviewed medical imaging techniques in battling COVID-19. Various contact-less image acquisition techniques, deep learning-based segmentation of lungs and lesion, x-ray and CT screenings, and severity analysis of COVID-19 along with some publicly open datasets are included in the work.	They did not provide any quantitative analysis of existing methods either. Also, their discussions on the existing datasets are somewhat inadequate.

5.	M.K. Pandit a, b, S.A. Banday, R. Naaz, M.A. Chishti	Automatic detection of COVID-19 from chest radiographs using deep learning	They have used transfer learning strategy.	They have used small datasets and they have put as limitation. Using small datasets in deep learning approach is not good enough to overcome a good accuracy.

Chapter Three: System Design

3.1. Introduction

In this chapter, a detailed description of the proposed system or model for the identification and grading of COVID-19 is discussed. It requires a series of steps starting from

preprocessing of images, feature extraction, and learning to classification into predefined classes. Classification mainly encompasses three major phases; these are training phase, the testing phase, and the validation phase. In section 3.2, a general description of the proposed system architecture is presented. In the following sections, each process (preprocessing, feature extraction, feature learning, and classification) is described thoroughly.

3.2. System Architecture

The proposed system has three components: preprocessing, feature extraction and classification. In image preprocessing, we normalize the image to a standard size. We used predetermined coordinates, set empirically from the dataset. In feature extraction, we used CNN to identify and select important features that account for the symptom of the disease. For classification, we used convolutional neural network. Classification encompasses three main phases: training, validation, and testing phase. Finally, a SoftMax is used for grading into a specific class (normal/no findings, Pneumonia, and COVID-19). The proposed system architecture is depicted in Figure 3.1.

The training phase encompasses different layers (mainly convolution, activation, and pooling) stacked on top of each other for learning the most characteristic or distinguishing features. Once the features are learned, classification is performed by using the SoftMax classifier.

The validation phase is mainly concerned to optimize (to increase the accuracy or decrease the loss) the learning model. In this phase, different techniques such as batch normalization and dropout (with smaller value) at early stages are applied after each convolution, which are followed by activation layer. Finally, the learning model is constructed from training or validation of samples and will be used for testing other samples.

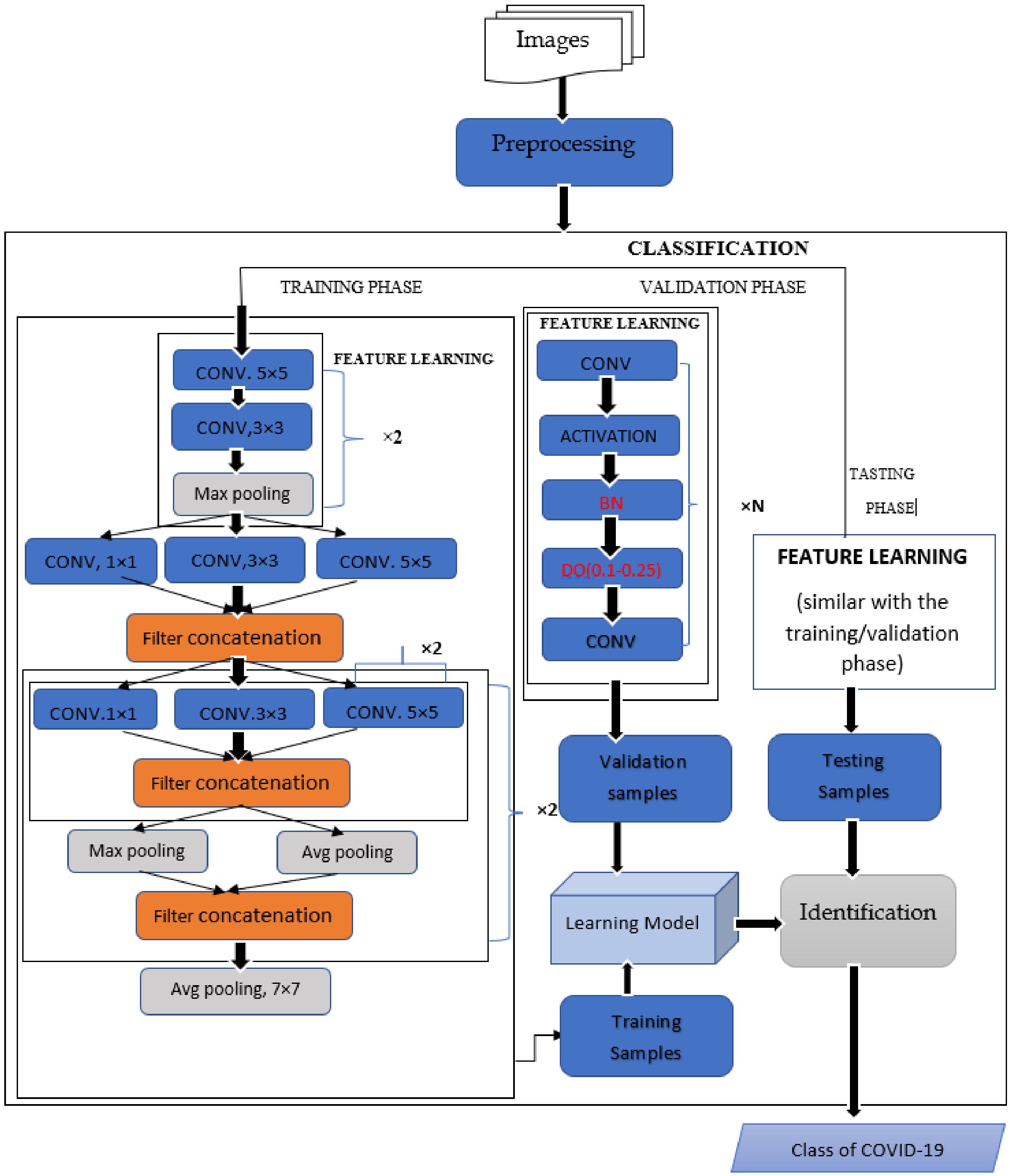

Figure 3.1: System Architecture

The testing phase follows similar procedures with the training or validation phase. It should follow the same way, otherwise, it may produce an inaccurate result or will generate an error.

3.3. Preprocessing

The preprocessing phase involves image resizing (to 224 x 224) and conversion of images into NumPy array that Keras can work with. The state-of-the-art models (Krizhevsky et al., 2012), (Gliner et al., 2021) take as input an image size of (224 x 224). Hence, it is convenient to use an image size similar with previous models as we evaluate our network by comparing with state-of-the-art models. OpenCV is used for image resizing and conversion into NumPy array. The input layer should be divisible by two multiple times after each convolution layer. This will enable the spatial input in our model to be down-sampled through pooling operation in a suitable and efficient manner. An image source in Hospital can be in different size. So, it's mandatory to resize and rescale the image. Before the data is feed into the network, the data should be organized in to suitable size. Process before feeding the dataset into CNN mode as described follows: -

- ✓ Encode the PNG file using Keras package.
- ✓ Decoding the PNG file content into different color channels (RGB) grids pixels.
- ✓ Rescale the pixel values to 254x254 and conversion of the image into NumPy arrays.

Input: A sample image I

Output: An array of resized image (to 224,224)

Begin;

Image = original image I

Image = resize (Image, 224,224)

Image = image_to_array (Image)

Return Image

End

Algorithm3.1: Image Resizing

In the pre-processing stage, the input shape is determined. If the input shape is channels last ordering, then the depth of the input volume needs to be placed at the third coordinate next to height and width. However, if the input shape is channels first ordering, then the depth of the input volume needs to be placed at the first coordinate before height and width. This will ensure our network (or model) to work regardless of image channel ordering.

```
Input: An image data format
Output: InputShape //with the correct depth coordinate
Begin;
    If image data format == channels first:
        Channel dimension =1; //the position of the depth is at the first coordinate
        InputShape = (depth, height, width);
    Else:
        Channel dimension = -1 //the depth will be located next to height and width
        InputShape = (height, width, depth)
    Return InputShape
End
```

Algorithm 3.2: Input Shape Ordering

3.4. Feature Extraction

A human chest radiographic image contains interesting structure and provides abundant texture information. It contains discriminating textural features so that it can be used whether a person is caught by a particular disease or not.

3.6. Classification

Classification is done after the distinctive features are learned. Feature learning encompasses different layers stacked on top of each other. In this section, description of each operation in each phase that is used for learning features and grading into predefined classes (Normal/no findings, Pneumonia and COVID-19) is presented.

3.6.1 Training Phase

3.6.1.1 Feature Learning for Training Phase

Following are descriptions of each layer in each phase to learn the characteristics features that characterize at each grade of covid-19.

Convolution layer: there are 19 convolution layers in the training phase. The input to the first convolution layer is 224 x 224 x 1 image. As discussed in chapter two, convolution operation requires four parameters. The first parameter is the number of filters that are used to control the depth of the output volume. The study model used 32, 64, 96, 128, 160, and 192 filters. The number of filters applied is increased as one moves to the fully connected layers and the Softmax classifier. Different number of convolution layers and different number of filters are tested and those that achieves higher accuracy are selected. The second parameter is the receptive field size, which determines the size of each filter (kernel) and is nearly always square. The study used 5 x 5, 3 x 3, and 1 x 1 filter size at a single layer. A characteristic feature that cannot be detected by 3 x 3 filter size can be detected by a filter size of 5 x 5. The third parameter is stride size, which determines the number of pixels skipped (horizontally and vertically) each time we make convolution operation. The study used stride size of two (2, 2) and one (1, 1). When stride size is two, image dimension is reduced by half vertically and horizontally. The fourth (last) parameter is the amount of zero-padding, which is used to control the size of the output. We have used “same” padding, which means the size of the output is equal to the size of the input if the stride size is one.

Since we don’t want to change the size of an input volume through convolution operation, we have used stride size of one and padding “same”, otherwise we apply stride size of two. We have done convolution operation repeatedly before the input image is down-sampled with the pooling layers (operation). Functioning multiple convolution layers before applying a pooling layer allows the model to develop more complex features before the destructive pooling operation is performed.

Convolution operation (denoted by * operator) over a two-dimensional input image I and two-dimensional kernel K is defined as:

$$S(i,j) = (I * K)(i,j) = \sum_{m}.\sum_{n}.\ K(i+m, j+n)I(m,n) \qquad 4.1$$

Where i and j are image I coordinates, and m, and n are kernel K coordinates. Mathematically,
equation 4.1 is called cross-correlation. However, many neural network libraries implement it as
a convolution operation and call it convolution (Goodfellow and Yoshua Bengio, 2015). We also use this library implemented based on the equation 4.1.

The output of convolution layer after each convolution operation is then W_{output} x H_{output} x D_{output} (Adrian, 2017)where:

$$W_{output} = \left(\frac{W_{input} - F + 2P}{S}\right) + 1$$

$$H_{output} = \left(\frac{H_{input} - F + 2P}{S}\right) + 1$$

$$D_{output} = K \qquad 4.2$$

Where F denotes the filter size, P denotes the number of zero padding, S denotes the number of
stride size, K denotes the number of filters applied.

Selection of filter size: In the convolution module, we have used 1 x 1, 3 x 3, and 5 x 5 filter size at the same layer.

Activation layer: activation function is applied in an element-wise manner. Hence, the output of the activation function is always the same as the size (dimension) of the input. The width, height, and depth of the output layer is the same as the width, height, and depth of the input layer respectively. We have used ReLU activation function in the activation layer throughout our model. ReLU activation function returns zero, if the values in the input layer is negative, otherwise, it returns its existing value. Mathematically, it is defined as:

$$Y = \max(0, x) \qquad 4.3$$

Pooling layer: other operation to change the size of the input volume is pooling. We have applied max pooling operation after two or more consecutive convolution layers. We use pooling operation to progressively reduce the width and height of the input volume. Pooling operation requires two parameters. The first parameter is pool size, which controls by how much we want to reduce the spatial size of the input volume. Since our input image size is greater than 200 pixels, we have applied 2 x 2 pooling size many times after the consecutive convolution layers. The second parameter is the stride size, which determine the number of pixels we skipped while doing the pooling operation. We have used stride size of two each time we do pooling operation.

Since max pooling result in loss of accurate spatial information (Szegedy et al., 2015), we propose to do average pooling (concatenated with max pooling at the same time) to reduce distraction of features. Average pooling, at least, holds the average of all data, thereby the selected data have high relation with all other data rather than only with the maximum. As shown in table 4.2 below, the application of 2x2 max pooling and average pooling with a stride size of 2 on a 4x4 input volume (right) would result in a 2x2x2 output volume (left). Concatenating the result of both max and average pooling hold more information about the features (data) rather than doing only max pooling. Like the convolution module, we apply pooling module at higher layers while keeping the lower layers with the traditional max pooling.

Table 4.1: Application of Pooling Module on a 4x4 Input Volume

12	35	117	209
156	123	7	203
235	64	141	19
67	203	45	182

81.5	134
142	97

156	209
235	182

The output of pooling layer after each pooling operation is then W_{output} x H_{output} x D_{output} (Adrian, 2017)

where:

$$W_{output} = \left(\frac{W_{input} - F}{S}\right) + 1$$

$$H_{output} = \left(\frac{H_{input} - F}{S}\right) + 1$$

$$D_{output} = K \qquad 4.4$$

Fully connected layer: We have used only one fully connected layer to compute the final output probabilities for each class before applying to the Softmax classifier. It is due to the move from fully connected layer to average pooling result in 0.6% increment of accuracy (Szegedy et al., 2015). It holds three nodes (equal to number of classes), which are directly applied to the Softmax classifier.

Dropout layer: is used to reduce overfitting by randomly disconnecting inputs from the previous layer to the next layer in the network architecture. Random disconnection ensure that no single node is responsible for "activation" when presented with a pattern. It enables multiple, redundant nodes to activate when given with similar pattern (inputs), which also helps our model to generalize. We have applied dropout layers with p (dropping probability) = 0.4 immediately before the fully connected layers, which is followed by the Softmax classifier.

Input: preprocessed image I

Output: feature vector

Begin;

Get the processed image I

Initialize the number of filters K, filter size F, stride size S, and zero-padding ZP, pool size PS, the number of nodes N, the number of classes C, and dropout probability P;

//the first block of the model

For i =0; i<2, i++;

Apply convolution operation, Convolution (K, F, ZP, S);

Apply activation function, ReLU on the output of the previous convolution operation;

Apply convolution operation, Convolution (K, F, ZP, S);

End for

// the first block of pooling module

Apply max pooling operation, MaxPool (PS, S)

Apply average pooling operation, AvgPool (PS, S)

Concatenate filter size

// Similarly apply other pooling modules

// the first block of convolution operation

Apply 1 x 1 convolution operation, Convolution (K, (1,1), ZP, S);

Apply 3 x 3 convolution operation, Convolution (K, (3,3), ZP, S);

Apply 5 x 5 convolution operation, Convolution (K, (5,5), ZP, S);

Concatenate filter size

// similarly apply other convolution modules

Apply dropout operation, Dropout (P) // drop around half of the nodes, if P = 0.4

Apply fully connected layer, FC (C); // takes only the number of classes which will be directly applied to the Softmax classifier.

Save (or return) extracted features

End

Algorithm 3.3: Feature Learning in Training Phase

The proposed convolutional neural network is "deep" for two reasons: the depth (the number of levels is higher) and the width (the number of units at same level is more than one). The width of our model is controlled by the convolution module and pooling module. These modules introduce a new level of organization (in the architecture of CNN) different from the basic CNN architecture (figure 2.3) where different layers or operations are stacked on top of each other linearly (Krizhevsky et al., 2012), (Simonyan and Zisserman, 2015). In the basic CNN architecture, only one operation (convolution, pooling, activation, or batch normalization) is done at a single layer.

The convolution module and the pooling module are the building blocks of the model. The convolution module (as shown in figure 3.3 left) allows us to do different convolution operation at a single layer. The pooling module (as shown in figure 3.3 right) also allows us to do both maximum pooling and average pooling at a single layer. The output of each convolution operation in the convolution module is concatenated and taken as an input to the next layer. Similarly, the output of each pooling operation in the pooling module is concatenated and taken as input to the next layer.

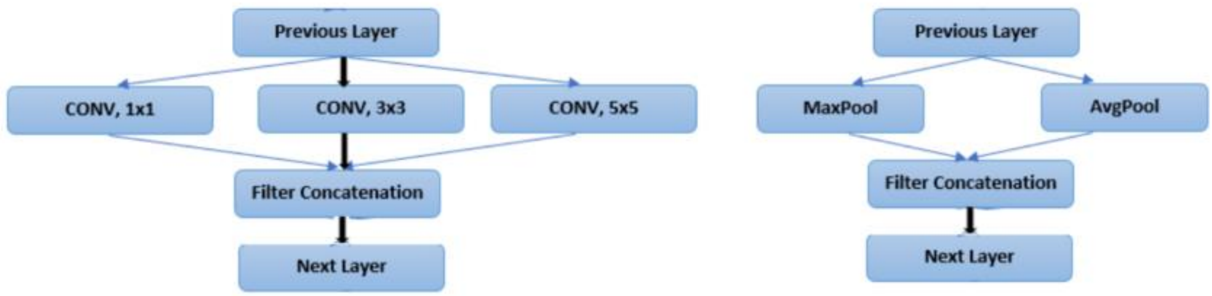

Figure 3.3: Left: Convolution Module; Right: Pooling Module

3.6.1.2 Softmax

The output of the final fully-connected layer is given as input to the Softmax classifier. A 3-way Softmax is used for grading in to specific class (Normal/no findings, Covid-19, and Pneumonia).

Input: extracted or learned features

Output: class label

Begin;

Get the extracted or learned features (**from algorithm 3.3)**

Apply the Softmax classifier on the learned features

Return the class label

End

Algorithm3.4: Applying Softmax Classifier

3.6.2. Validation phase

3.6.2.1 Feature Learning for Validation Phase

In this phase, our aim is to improve the performance of our model by using validation datasets. we can say the performance of the model is improved, if the accuracy of the model is increased or the loss of the model is decreased. Following are the techniques we have used to improve the performance of our model.

Batch normalization: is used to normalize the activations of the given input volume before being applied to the next layer in the network. Even though, it is used to reduce the number of epochs taken to train a neural network and stabilize training, it slows down the wall time (by more than 2x) taken to train our network (Witten et al., 2017). The extra time, however, outweighs the negatives. In addition, it makes learning rate and regularization less volatile to tune.

When we apply batch normalization after each activation layer in our network, we have got more stable loss curve. We also got better accuracy after batch normalization.

Dropout at initial stages: we have seen the effect of dropout at the final stage applied between the last two fully connected layers in the training phase. It is also possible to apply dropout between the convolutional layers and after batch normalization in early stages and improve the performance of our model. In this time, however, we apply dropout layers with smaller probability p = 0.1-0.25 following convolution layers.

Input: preprocessed image I

Output: feature vector

Begin;

Get the processed image I

Initialize the number of filters K, filter size F, stride size S, and zero-padding ZP, pool size PS, the number of nodes N, the number of classes C, and dropout probability P;

//the first block of the model

For i =0; i<2, i++;

Apply convolution operation, Convolution (K, F, ZP, S);

Apply activation function, ReLU on the output of the previous convolution operation;

Apply batch normalization on the activated output in the previous layer

Apply convolution operation, Convolution (K, F, ZP, S);

End for

Apply max pooling operation, MaxPool (PS, S)

Apply average pooling operation, AvgPool (PS, S)

Concatenate filter size

Apply dropout operation, Dropout (P=0.1-0.25)

// Similarly apply other pooling modules of the model

// the first block of convolution operation

Apply 1 x 1 convolution operation, Convolution (K, (1,1), ZP, S);

Apply 3 x 3 convolution operation, Convolution (K, (3,3), ZP, S);

Apply 5 x 5 convolution operation, Convolution (K, (5,5), ZP, S);

Concatenate filter size

// similarly apply other convolution modules

Apply fully connected layer, FC (C); // takes only the number of classes

which will be directly applied to the Softmax classifier.

Save (or return) extracted features

End

Algorithm 3.5: Feature Learning in Validation Phase

Checkpointing: used to serialize our network to disk and visualize it each time there is an improvement during training. Saving model's weights to disk after every epoch, allow us to checkpoint our network and choose the best performing one.

Learning rate: we use smaller learning rate to reduce the weight update. Decaying the learning rate is helpful in reducing overfitting. Keeping learning rate high will lead to overshooting of areas of the low loss since we are taking large steps to descend into these areas. It is better to decrease the learning rate progressively, and hence taking smaller steps. Reduced rate enables the network to descend into lower loss landscape (without missing it) effectively.

Input: learning rate, and number of epochs, momentum

Output: learning rate scheduler setting

Begin;

Initialize learning rate, LR, and number of epochs, EPOCHS

Calculate learning rate decay as, decay = LR / EPOCHS

Select the optimization algorithm; // for example Adam, SGD

opt = Adam (LR, decay, momentum); // Applying the Adam optimization algorithm

Return the LR scheduler setting

End

Algorithm 3.6: Learning Rate Scheduler Setting

According to (Witten et al., 2017) learning rate scheduler generally has two main objectives. These are: (1) finding a set of learning reasonably "good" weights early in the training process with a higher learning rate. (2) tuning these (learned) weights later in the process to find more optimal weights using a smaller learning rate. We have applied learning rate schedulers that reduce gradually based on the epoch number.

Data augmentation: we have applied random transformation (such as translation, rotation, shearing, and resizing) to our dataset to balance the insufficiency of our data. It is clear that increasing the size of our dataset will improve the performance of our model.

Input: preprocessed image

Output: augmented image

Begin;

Initialize rotation range (RR), width shift range (WSR), height shift range (HSR), shear range (SR), zoom range (ZR)

Set horizontal flip (HF) to true;

// Applying augmentation operation

aug = ImageDataGenerator (RR, WSR, HSR, SR, ZR, HF)

Return aug; //augmented images

End

Algorithm 3.7: Data Augmentation

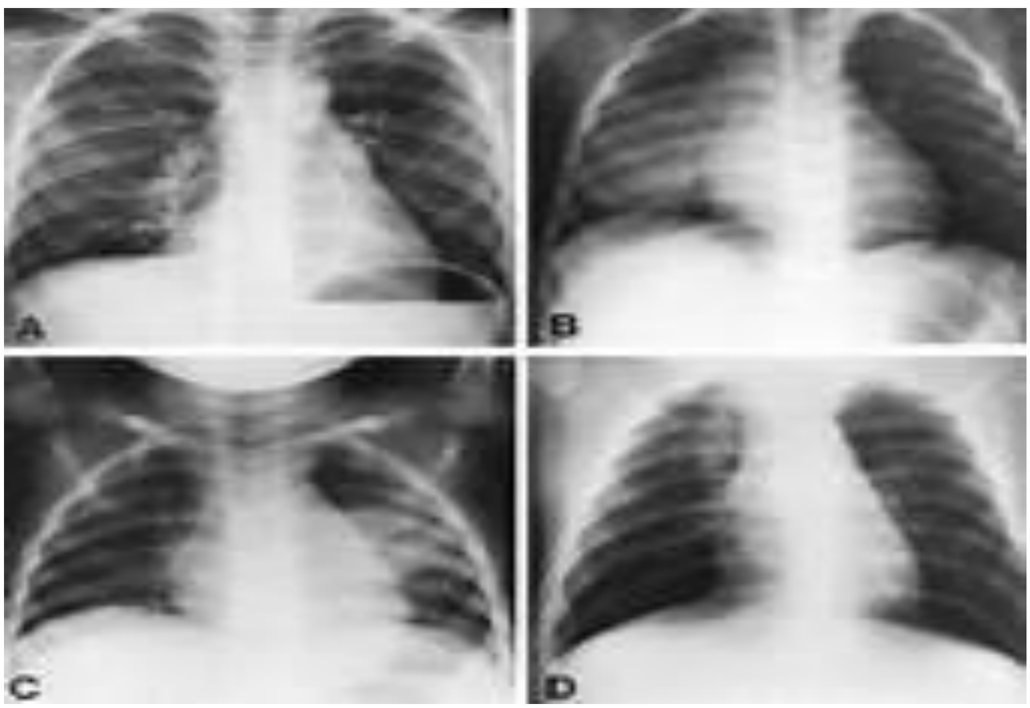

Figure3.2: Data Augmentation

Data augmentation may reduce training accuracy since the input images are changed constantly as compare to training without data augmentation. However, data augmentation dramatically reduce overfitting. In addition, it enables the model to generalize better to new input samples.

The output of the data augmentation algorithm is illustrated in figure 3.2. Notice all images (except the top-left image) are randomly rotated, shifted (vertically and horizontally), sheared, zoomed, and horizontally flipped. In each case the image retains the original class label: COVID-19. However, each image has been modified slightly, thereby giving our neural network new patterns to learn. As shown in Figure 3.2, data augmentation helps us to generate additional training data, thereby reducing the amount of hand-labeled data required to train a deep network.

3.6.3. Testing Phase

In this phase, the study followed the same procedure as the training phase or the validation phase. We have to preprocess the image in the same manner as the training or validation phase. If we follow any other way, it will lead us to incorrect classification since the network may be presented with patterns (inputs) it can't categorize. Feature learning and classification is also done in the same manner as the training or validation phase by using the knowledge base or the model constructed from the training or validation phase. Input images that are different from the training datasets are used, which are called testing dataset.

3.6.3.1 Feature Learning for Testing Phase

In this phase, we follow the same procedure as the training phase or the validation phase. We have to preprocess the image in the same manner as the training or validation phase. Any other approach may result in inaccurate categorization since the network may be faced with patterns (inputs) that it is unable to categorize. Similarly, feature learning is carried out in the same way as training or validation, utilizing the learning model developed during the training or validation phase. The testing dataset is made up of input images that are different from the training datasets.

Identification

As we described in the previous sections, after learnable feature is extracted and build the model, the next step is identification and classification of Covid-19 disease using three classes.

3.7. Summary

In this chapter, the design of the deep convolutional neural network model for identification of COVID-19 is discussed thoroughly. The layers, the activation functions, and the classifier that compose the model are discussed in detail. The design of the model is composed of three phases; training phase, testing phase, and validation phase. The training and validation phase uses the training dataset to form the knowledge base or the model. The testing phase passes via the same procedures as in the training and validation phases.

Chapter Four: Result and Discussion

4.1. Introduction

In this chapter, the experimental evaluation of the proposed model for identification and grading of COVID-19 is described in detail. Experimental evaluation approves the realization of the proposed model or architecture. The dataset used and the implementation of the proposed model are described thoroughly. Finally, the test results are also presented.

4.2. Dataset

For this study, Chest radiographic images classified by Pulmonologists are taken from Tibebe Gion hospital, which is stored in their local database. In the image acquisition phase, normal and diseased (containing COVID-19) images are taken. We have taken images of the normal/no findings, pneumonia and COVID_19. All images were captured using Charge-coupled devices (CCDs). All the images are in png (Portable Graphics Format) and have pixel size of 4288 x 2848.

Table 4.1: Data Description

No.	**Grade of COVID-19**	**Source**	**Special Resolution**	**Image Format**	**quantity**
1	Normal/no findings	Tibebe Gion Hospital	224 x 224	Png	6650
2	Pneumonia	Tibebe Gion Hospital	224 x 224	Png	6650
3	COVID-19	Tibebe Gion Hospital	224 x 224	Png	6650

4.3. Implementation

Experiments are done based on the prototype developed with Keras (TensorFlow as a backend) on Intel Core ™ i5-6200 CPU, and 8 GB of RAM. The model is trained for 50 epochs, a batch size of 32, and a starting or initial learning rate of 0.001 (1e-3). The data is partitioned into training and testing dataset such that 70 percent of the data is assigned for training the model and 30 percent of the data is allotted for testing. Allocating 2/3rd of the dataset (Dobbin and Simon, 2011) for training is close to optimal for reasonably sized datasets (greater than 100 images).

4.4. Test Result

We have used precision, recall and f1-score for measuring the performance of our model. In addition, we have also calculated the micro-average, macro-average, and weighted-average for all the aforementioned performance metrics.

Before comparison of the model with the state-of-the-art models, we have to clearly show the difference on the accuracy of our model in the training phase and validation phase. The training phase is the sequence of convolution, activation function pooling, and fully-connected layers and dropout after the final fully-connected layers and before the Softmax classifier. However, in the validation phase, the model is trained using different parameters such as batch normalization (after the activation function), small dropout values at the initial stages. In addition, the validation phase is also trained using data augmentation.

4.4.1. COVID-19 Model During Training Phase

As clearly shown in figure 4.1 below, our model obtains over 99% training accuracy and testing accuracy. This classification accuracy is obtained when the model is trained with batch normalization, data augmentation, and dropout at initial stages.

```
Test Loss 0.03673916310071945

In [28]: print("Test Accuracy",test_acc)

Test Accuracy 0.9915522933006287
```

Figure 4.1: Training and Testing Accuracies

The overall analysis of our model with-in total parameters, trainable and non-trainable parameters is given below.

```
Stream
  [INFO] compiling model...
  Model: "sequential"
  _________________________________________________________________
  Layer (type)                 Output Shape              Param #   
  =================================================================
  conv2d (Conv2D)              (None, 64, 64, 32)        2432      
  _________________________________________________________________
  activation (Activation)      (None, 64, 64, 32)        0         
  _________________________________________________________________
  batch_normalization (BatchNo (None, 64, 64, 32)        128       
  _________________________________________________________________
  conv2d_1 (Conv2D)            (None, 64, 64, 32)        9248      
  _________________________________________________________________
  activation_1 (Activation)    (None, 64, 64, 32)        0         
  _________________________________________________________________
  batch_normalization_1 (Batch (None, 64, 64, 32)        128       
  _________________________________________________________________
  conv2d_2 (Conv2D)            (None, 32, 32, 32)        25632     
  _________________________________________________________________
  activation_2 (Activation)    (None, 32, 32, 32)        0         
  _________________________________________________________________
  batch_normalization_2 (Batch (None, 32, 32, 32)        128       
```

```
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
activation_3 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization_3 (Batch (None, 32, 32, 32)        128
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 15, 15, 64)        18496
_________________________________________________________________
activation_4 (Activation)    (None, 15, 15, 64)        0
_________________________________________________________________
batch_normalization_4 (Batch (None, 15, 15, 64)        256
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 15, 15, 64)        36928
_________________________________________________________________
activation_5 (Activation)    (None, 15, 15, 64)        0
_________________________________________________________________
batch_normalization_5 (Batch (None, 15, 15, 64)        256
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 15, 15, 64)        36928
```

```
_________________________________________________________________
activation_6 (Activation)    (None, 15, 15, 64)        0
_________________________________________________________________
batch_normalization_6 (Batch (None, 15, 15, 64)        256
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 15, 15, 64)        36928
_________________________________________________________________
activation_7 (Activation)    (None, 15, 15, 64)        0
_________________________________________________________________
batch_normalization_7 (Batch (None, 15, 15, 64)        256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 7, 7, 64)          0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 4, 4, 128)         73856
_________________________________________________________________
activation_8 (Activation)    (None, 4, 4, 128)         0
_________________________________________________________________
batch_normalization_8 (Batch (None, 4, 4, 128)         512
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 4, 4, 128)         16512
_________________________________________________________________
activation_9 (Activation)    (None, 4, 4, 128)         0
```

```
_________________________________________________________________
batch_normalization_9 (Batch (None, 4, 4, 128)         512
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 2, 2, 128)         147584
_________________________________________________________________
activation_10 (Activation)   (None, 2, 2, 128)         0
_________________________________________________________________
batch_normalization_10 (Batc (None, 2, 2, 128)         512
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 2, 2, 128)         16512
_________________________________________________________________
activation_11 (Activation)   (None, 2, 2, 128)         0
_________________________________________________________________
batch_normalization_11 (Batc (None, 2, 2, 128)         512
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 1, 1, 128)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 1, 1, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 1024)              132096
_________________________________________________________________
activation_12 (Activation)   (None, 1024)              0
```

```
max_pooling2d_2 (MaxPooling2 (None, 1, 1, 128)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 1, 1, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 1024)              132096
_________________________________________________________________
activation_12 (Activation)   (None, 1024)              0
_________________________________________________________________
batch_normalization_12 (Batc (None, 1024)              4096
_________________________________________________________________
dropout_3 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 30)                30750
_________________________________________________________________
activation_13 (Activation)   (None, 30)                0
=================================================================
Total params: 600,830
Trainable params: 596,990
Non-trainable params: 3,840
_________________________________________________________________
[INFO] training network...
```

Figure 4.2: Analysis of Our Model

As clearly depicted in figure 4.3 below, our model obtains 98.35% training accuracy and 98% validation accuracy.

```
Stream
  [INFO] training network...
  Epoch 1/50
  99/99 [==============================] - 97s 971ms/step - loss: 4.0684 - accuracy: 0.0748 - val_loss: 3.6397 - val_accuracy: 0.0325
  Epoch 2/50
  99/99 [==============================] - 96s 966ms/step - loss: 3.1291 - accuracy: 0.1852 - val_loss: 4.0363 - val_accuracy: 0.0496
  Epoch 3/50
  99/99 [==============================] - 95s 960ms/step - loss: 2.2817 - accuracy: 0.3600 - val_loss: 3.9668 - val_accuracy: 0.1161
  Epoch 4/50
  99/99 [==============================] - 95s 958ms/step - loss: 1.6618 - accuracy: 0.5059 - val_loss: 1.3059 - val_accuracy: 0.6021
  Epoch 5/50
  99/99 [==============================] - 98s 996ms/step - loss: 1.2267 - accuracy: 0.6321 - val_loss: 1.1597 - val_accuracy: 0.6871
  Epoch 6/50
  99/99 [==============================] - 95s 956ms/step - loss: 0.9436 - accuracy: 0.7025 - val_loss: 0.9986 - val_accuracy: 0.7581
  Epoch 7/50
  99/99 [==============================] - 95s 956ms/step - loss: 0.7258 - accuracy: 0.7713 - val_loss: 0.4167 - val_accuracy: 0.8499
  Epoch 8/50
  99/99 [==============================] - 95s 960ms/step - loss: 0.6233 - accuracy: 0.8049 - val_loss: 0.3621 - val_accuracy: 0.8942
  Epoch 9/50
  99/99 [==============================] - 95s 956ms/step - loss: 0.5546 - accuracy: 0.8227 - val_loss: 0.2844 - val_accuracy: 0.9090
  Epoch 10/50
  99/99 [==============================] - 95s 957ms/step - loss: 0.4560 - accuracy: 0.8487 - val_loss: 0.3466 - val_accuracy: 0.8831
  Epoch 11/50
```

```
99/99 [==============================] - 94s 949ms/step - loss: 0.0836 - accuracy: 0.9762 - val_loss: 0.0632 - val_accuracy: 0.9808
Epoch 40/50
99/99 [==============================] - 94s 952ms/step - loss: 0.1019 - accuracy: 0.9699 - val_loss: 0.1586 - val_accuracy: 0.9571
Epoch 41/50
99/99 [==============================] - 94s 950ms/step - loss: 0.1053 - accuracy: 0.9689 - val_loss: 0.3142 - val_accuracy: 0.9209
Epoch 42/50
99/99 [==============================] - 94s 953ms/step - loss: 0.1059 - accuracy: 0.9673 - val_loss: 0.1164 - val_accuracy: 0.9712
Epoch 43/50
99/99 [==============================] - 94s 950ms/step - loss: 0.1195 - accuracy: 0.9667 - val_loss: 0.0981 - val_accuracy: 0.9675
Epoch 44/50
99/99 [==============================] - 94s 950ms/step - loss: 0.0923 - accuracy: 0.9718 - val_loss: 0.1225 - val_accuracy: 0.9704
Epoch 45/50
99/99 [==============================] - 94s 950ms/step - loss: 0.0769 - accuracy: 0.9753 - val_loss: 0.0789 - val_accuracy: 0.9793
Epoch 46/50
99/99 [==============================] - 94s 947ms/step - loss: 0.0763 - accuracy: 0.9775 - val_loss: 0.0476 - val_accuracy: 0.9882
Epoch 47/50
99/99 [==============================] - 94s 948ms/step - loss: 0.0669 - accuracy: 0.9765 - val_loss: 0.0610 - val_accuracy: 0.9845
Epoch 48/50
99/99 [==============================] - 94s 948ms/step - loss: 0.0897 - accuracy: 0.9699 - val_loss: 0.0747 - val_accuracy: 0.9778
Epoch 49/50
99/99 [==============================] - 94s 948ms/step - loss: 0.1135 - accuracy: 0.9657 - val_loss: 0.1243 - val_accuracy: 0.9660
Epoch 50/50
99/99 [==============================] - 94s 951ms/step - loss: 0.0811 - accuracy: 0.9727 - val_loss: 0.0522 - val_accuracy: 0.9845
```

Figure 4.3: Iteration to Calculate Loss and Accuracies of the Model

As clearly shown in the training loss and accuracy curve in figure 4.4 and figure 4.5 below, training and validation accuracy increases while training and validation loss decreases. The following figure (figure 4.3) illustrates the loss and validation accuracy of our model.

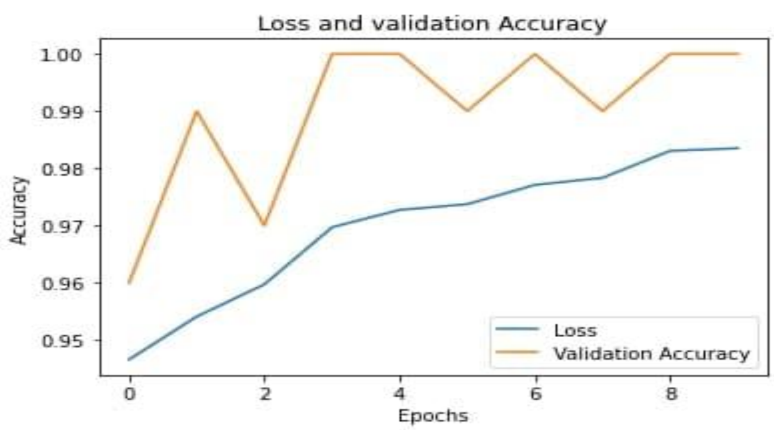

Figure 4.4: Loss and Validation Accuracy

Figure 4.4 illustrates the loss and validation loss of our model.

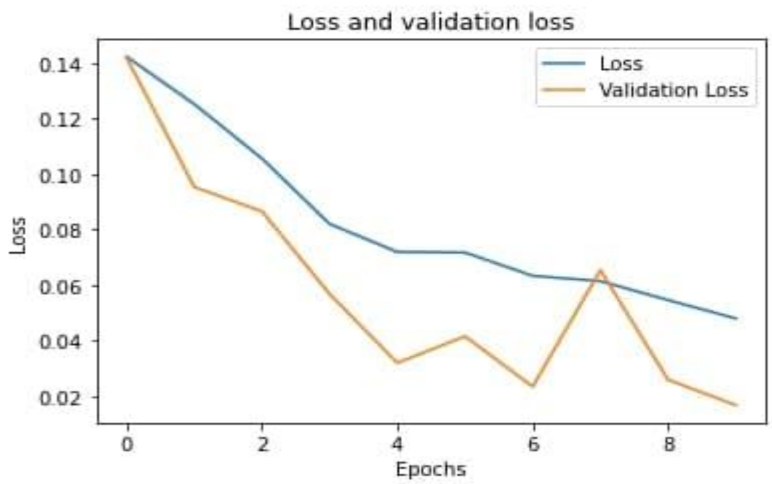

Figure 4.5: Loss and Validation Loss

Figure 4.6 below illustrates the precision, recall, f1-score and support values for no findings, covid-19 and pneumonia classes of our model.

	precision	recall	f1-score	support
Covid-19	0.96	0.86	0.91	28
No_findings	0.89	0.94	0.91	109
Pneumonia	0.88	0.85	0.87	88
micro avg	0.89	0.89	0.89	225
macro avg	0.91	0.88	0.89	225
weighted avg	0.89	0.89	0.89	225

Figure 4.6: Evaluation of Our Model by Testing Datasets

It can be noted from figure 4.6 that the proposed model has achieved an average accuracy of 96% in identifying covid-19.

4.5. Discussion

The challenges in the experiments are overfitting and oscillations in the training and validation loss or accuracy. It is due to a random sample from our dataset: the dataset at each evaluation step is different, so is the validation loss. In addition, at each epoch there (70% of the training dataset, divided by the batch size) iterations. At each iteration, different samples are taken, trained and tested, thereby oscillation or overfitting occurred. It also occurs due to the result of the

SGD overshooting in the optimal direction or overshooting of areas of the low loss (Adrian, 2017), (A.Ng, 2018).

In this chapter, experimental evaluation of the study model for identification of Covid-19 is described in detail. The dataset used and the implementation of the study model are described in detail. In addition, the test results are presented and compared with the state-of the-art models like AlexNet and GoogleNet.

Chapter Five: Conclusion and Future work

5.1 Conclusion

This research work identified CNN as a major contributor and recent research area for the identification of Covid-19 diseases. CNNs are local invariant so that it can be used to identify an image as having a particular sign of disease regardless of its spatial location. A CNN computes the same features of an image across all spatial areas. After a thorough review of literatures and related works, we have interested in identification of Covid-19 is given tiny emphasis and are unresolved problems yet. We have also identified that different approaches such as image processing, machine learning and deep learning techniques are highly applied for detection of the diseases but not for grading the severity of the diseases.

This research work solved the aforementioned problems by using convolution module and pooling module to the basic or default CNN architecture. Hence, a deep CNN system for identification of Covid-19 using images of normal and infected people was developed. Hence, dependency only on the skill and experience of human experts is reduced.

The significance of the developed system will bring change in the progress of diagnosis in terms of accuracy and efficiency. The system has learned a 98.35% training accuracy and 98% testing accuracy in Covid-19 identification. Hence, our system would ease the problems that the field faced before in its dependency on expert skills and delay in diagnosis.

5.2 Contribution

As a contribution to the scientific world or the knowledge, the proposed deep CNN model offered a systematic approach in identification of Covid-19. The study has used multiple datasets to train the model. Considering the CNN architecture, we propose a convolution and pooling module that weighs less, train faster and classify more accurately than other CNN (state-of-the-art) models. In this work, therefore, the following vital contributions are made.

- ✓ **Datasets:** we have trained our model by using maximum amount of chest radiographic images compared with other related works.

- ✓ **Pooling module:** we propose maximum pooling and average pooling to be done at the same layer and to concatenate their filters. Since max pooling is distractive by its nature, we propose to do average pooling to reduce distraction of data or features. average pooling, at least, holds the average of all data, thereby having high relation with all data rather than only with the maximum.
- ✓ **Performance improvement:** the proposed model achieves better result in terms of accuracy, loss, training time and model size. In terms of accuracy, we have got 98% training accuracy and 97.9% testing accuracy that are far above state-of-the-art models. The model weighs very less and trains faster as compared to previous models.

5.3. Future Work

The deep CNN model proposed in this research work can be used in identification of other diseases. We would say the proposed deep CNN model is fit to the current requirement though there are some issues that need additional work. In this section, therefore, we insight the key points that remain a challenge and of course were limitation of this research work is lack of getting a lot of chest radiographic images to train the CNN model and improves the identification accuracies. The researchers believe that an attempt in the future study should consider it to achieve better result in identifying and Covid-19. Hence, this research work presented different contributions that can be further improved or implemented on the effort to identify related diseases.

5. References

A.Ng, Andrew. 2018. "Supervised Learning." in *Supervised learning*. Stanford.

Adrian. 2017. "Deep Learning for Computer Vision with Python."

Anon. 2021. "Causes of COVID-19 in Ethiopia."

Azad, M., M. Hasan, and M. K. 2017. "Color Image Processing in Digital Image." *International Journal of New Technology and Research* 3(3):263334.

Basturk, S. Bahar, Claire E. J. Dancer, and Tony McNally. 2020. "Jo Ur Na l P Re." *Pharmacological Research* 104743. doi: 10.1016/j.tjog.2021.03.005.

Bhandari, Ranjana, Garima Khanna, and Anurag Kuhad. 2021. "Pharmacological Insight into Potential Therapeutic Agents for the Deadly Covid-19 Pandemic." *European Journal of Pharmacology* 890(October 2020):173643. doi: 10.1016/j.ejphar.2020.173643.

Bharath Ramsundar, and Reza Bosagh Zadeh. 2017. *TensorFlow for Deep Learning*. Vol. 42.

Cai, Wei, Xiaodong Wen, Qiu Tu, and Xiujuan Guo. 2019. "Research on Image Processing of Intelligent Building Environment Based on Pattern Recognition Technology." *Journal of Visual Communication and Image Representation* 61:141–48. doi: 10.1016/j.jvcir.2019.03.014.

Carlborg, B. 1976. "Complications When Drugs Accidentally Dissolve in the Oesophagus and the Airways (Swedish)." *Lakartidningen* 73(48):4201–4.

Churchill, Maesen, and I. n.d. "Histogram of Oriented Gradients for Detection of Multiple Scene Properties II . Previous Work III . HOG Descriptors for Image Classification."

CronJ. 2021. "RGB Color Space."

Dabbaghchian, Saeed, Masoumeh P. Ghaemmaghami, and Ali Aghagolzadeh. 2010. "Feature Extraction Using Discrete Cosine Transform and Discrimination Power Analysis with a Face Recognition Technology." *Pattern Recognition* 43(4):1431–40.

doi: 10.1016/j.patcog.2009.11.001.

Das, Amit Kumar, Sayantani Ghosh, Samiruddin Thunder, Rohit Dutta, Sachin Agarwal, and Amlan Chakrabarti. 2021. “Automatic COVID-19 Detection from X-Ray Images Using Ensemble Learning with Convolutional Neural Network.” *Pattern Analysis and Applications* 24(3):1111–24. doi: 10.1007/s10044-021-00970-4.

Davros, William. 2010. “ Digital Image Processing for Medical Applications .” *Medical Physics* 37(2):948–49. doi: 10.1118/1.3285412.

Dobbin, Kevin K., and Richard M. Simon. 2011. “Optimally Splitting Cases for Training and Testing High Dimensional Classifiers.” *BMC Medical Genomics* 4. doi: 10.1186/1755-8794-4-31.

Esteva, Andre, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. 2017. “Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks.” *Nature* 542(7639):115–18. doi: 10.1038/nature21056.

Farnham, Boston, Sebastopol Tokyo, Beijing Boston, Farnham Sebastopol, Tokyo Beijing, Tom Hope, Yehezkel S. Resheff, and Itay Lieder. 2017. *Learning TensorFlow A Guide to Building Deep Learning Systems Revision History for the First Edition: Second Release*.

Farnham, Boston, Sebastopol Tokyo, Beijing Boston, Farnham Sebastopol, Tom Hope, Yehezkel S. Resheff, and Itay Eder. 2017. *Learning TensorFlow A Guide to Building Deep Learning Systems Revision History for the First Edition: Second Release*.

García Reyes, Luis Enrique. 2013. “Modeling.” *Journal of Chemical Information and Modeling* 53(9):1689–99.

Ge, Yufeng, Geng Bai, Vincent Stoerger, and James C. Schnable. 2016. “Temporal Dynamics of Maize Plant Growth, Water Use, and Leaf Water Content Using Automated High Throughput RGB and Hyperspectral Imaging.” *Computers and*

Electronics in Agriculture 127:625–32. doi: 10.1016/j.compag.2016.07.028.

Gengler, Charles E., Matti Rossi, Wendy Hui, and Johanna Bragge. 2006. "The Design Science Research Process : A Model for Producing and Presenting Information System Research." *In Proceedings of the First International Conference on Design Science Research in Information Systems and Technology* (May 2014):83–106.

Gliner, Jeffrey A., George A. Morgan, Nancy L. Leech, Jeffrey A. Gliner, and George A. Morgan. 2021. "Measurement Reliability and Validity." *Research Methods in Applied Settings* 319–38. doi: 10.4324/9781410605337-29.

Goodfellow, Ian J., and Aaron Courville Yoshua Bengio. 2015. "Deep Learning [Draft of March 30, 2015]-MIT Press (2016)."

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Deep Residual Learning for Image Recognition." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem:770–78. doi: 10.1109/CVPR.2016.90.

Heaton, Jeff. 2018. "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep Learning." *Genetic Programming and Evolvable Machines* 19(1–2):305–7. doi: 10.1007/s10710-017-9314-z.

Huang, Chaolin, Yeming Wang, Xingwang Li, Lili Ren, Jianping Zhao, Yi Hu, Li Zhang, Guohui Fan, Jiuyang Xu, Xiaoying Gu, Zhenshun Cheng, Ting Yu, Jiaan Xia, Yuan Wei, Wenjuan Wu, Xuelei Xie, Wen Yin, Hui Li, Min Liu, Yan Xiao, Hong Gao, Li Guo, Jungang Xie, Guangfa Wang, Rongmeng Jiang, Zhancheng Gao, Qi Jin, Jianwei Wang, and Bin Cao. 2020. "Clinical Features of Patients Infected with 2019 Novel Coronavirus in Wuhan, China." *The Lancet* 395(10223):497–506. doi: 10.1016/S0140-6736(20)30183-5.

J. CARRIE, MACMILLAN, and JEREMY LEDGER. 2021. "The 12 Best COVID-19 Prevention Strategies." *Yale Medicine*.

Krizhevsky, By Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2012. "Cnn实际训练的." *Communications of the ACM* 60(6):84–90.

Liu, Jiaye, Xuejiao Liao, Shen Qian, Jing Yuan, Fuxiang Wang, Yingxia Liu, Zhaoqin Wang, Fu-sheng Wang, Lei Liu, and Zheng Zhang. 2020. “Community Transmission of Severe Acute Respiratory.” *Emerging Infectious Diseases* 26(6):1320–23.

Mahmud, Tanvir, Md Awsafur Rahman, and Shaikh Anowarul Fattah. 2020. “CovXNet: A Multi-Dilation Convolutional Neural Network for Automatic COVID-19 and Other Pneumonia Detection from Chest X-Ray Images with Transferable Multi-Receptive Feature Optimization.” *Computers in Biology and Medicine* 122(May):103869. doi: 10.1016/j.compbiomed.2020.103869.

Ouchicha, Chaimae, Ouafae Ammor, and Mohammed Meknassi. 2020. “CVDNet: A Novel Deep Learning Architecture for Detection of Coronavirus (Covid-19) from Chest x-Ray Images.” *Chaos, Solitons and Fractals* 140. doi: 10.1016/j.chaos.2020.110245.

P.K.Sinha. 2012. “Image Acquisition and Preprocessing for Machine Vision Systems.”

Pan, Feng, Tianhe Ye, Peng Sun, Shan Gui, Bo Liang, Lingli Li, Dandan Zheng, Jiazheng Wang, Richard L. Hesketh, Lian Yang, and Chuansheng Zheng. 2020. “Time Course of Lung Changes at Chest CT during Recovery from Coronavirus Disease 2019 (COVID-19).” *Radiology* 295(3):715–21. doi: 10.1148/radiol.2020200370.

Patterson, Josh, and Adam Gibson. 2017a. *Deep Learning: A Practionar Approach*. Vol. 521.

Patterson, Josh, and Adam Gibson. 2017b. *Deep Learning: A Practionar Approach*. Vol. 521.

Peffers, Ken, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. 2007. “A Design Science Research Methodology for Information Systems Research.” *Journal of Management Information Systems* 24(3):45–77. doi: 10.2753/MIS0742-1222240302.

Simonyan, Karen, and Andrew Zisserman. 2015. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” *3rd International Conference on Learning*

Representations, ICLR 2015 - Conference Track Proceedings 1–13.

Suri, Jasjit S., Sushant Agarwal, Suneet K. Gupta, Anudeep Puvvula, Mainak Biswas, Luca Saba, Arindam Bit, Gopal S. “A Narrative Review on Characterization of Acute Respiratory Distress Syndrome in COVID-19-Infected Lungs Using Artificial Intelligence.” *Computers in Biology and Medicine* 130(January). doi: 10.1016/j.compbiomed.2021.104210.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. “Going Deeper with Convolutions.” 1–9.

Talo, Muhammed, Ozal Yildirim, Ulas Baran Baloglu, Galip Aydin, and U. Rajendra Acharya. 2019. “Convolutional Neural Networks for Multi-Class Brain Disease Detection Using MRI Images.” *Computerized Medical Imaging and Graphics* 78:101673. doi: 10.1016/j.compmedimag.2019.101673.

Turkoglu, M. 2021. “COVID-19 Detection System Using Chest CT Images and Multiple Kernels-Extreme Learning Machine Based on Deep Neural Network.” *Irbm* 1:1–8. doi: 10.1016/j.irbm.2021.01.004.

U. Johns Hopkins. 2020. “Top Five Infected Countries.”

Varshni, Dimpy, Kartik Thakral, Lucky Agarwal, Rahul Nijhawan, and Ankush Mittal. 2019. “Pneumonia Detection Using CNN Based Feature Extraction.” *Proceedings of 2019 3rd IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2019*. doi: 10.1109/ICECCT.2019.8869364.

Vidal, Plácido L., Joaquim Moura, Jorge Novo, and Marcos Ortega. 2021. “Multi-Stage Transfer Learning for Lung Segmentation Using Portable X-Ray Devices for Patients with COVID-19.” *Expert Systems with Applications* 173:114677. doi: 10.1016/j.eswa.2021.114677.

WHO. 2021a. “RT-PCR.” *WHO*. Retrieved February 6, 2021 (https://www.who.int/emergencies/diseases/novel-coronavirus-2019/media-resources/science-in-5/episode-14---covid-19---

tests?gclid=Cj0KCQjw2NyFBhDoARIsAMtHtZ4dI6uPqBWF-https://www.who.int/emergencies/diseases/novel-coronavirus-2019?adgroupsurvey=%7Badgroups).

WHO. 2021b. “Who-Timeline for Covid-19.”

Witten, Ian H., Christopher J. Pal, and Mining Fourth. 2017. “Batch Normalization Deep Learning Deep Learning for Multimodal Data Fu- Sion.”

Xu, Yong, David Zhang, and Jing Yu Yang. 2010. “A Feature Extraction Method for Use with Bimodal Biometrics.” *Pattern Recognition* 43(3):1106–15. doi: 10.1016/j.patcog.2009.09.013.

Yıldırım, Özal, Paweł Pławiak, Ru San Tan, and U. Rajendra Acharya. 2018. “Arrhythmia Detection Using Deep Convolutional Neural Network with Long Duration ECG Signals.” *Computers in Biology and Medicine* 102(September):411–20. doi: 10.1016/j.compbiomed.2018.09.009.