جمهورية العراق
وزارة التعليم العالي و البحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات
قسم البرمجيات

اكتشاف كوفيد- 19 في صور مفراس الرئة باستخدام شبكة VGG16 المعدلة

رسالة

مقدمة إلى مجلس كلية تكنولوجيا المعلومات جامعة بابل وهي جزء من متطلبات نيل درجة الماجستير في تكنولوجيا المعلومات / البرامجيات

من قبل

قصي عوده جعفر حسين

بإشراف

أ.د. اسراء هادي علي حسين

1444 هـ

2022 م

Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Department of Software

DETECTION OF COVID-19 IN LUNG CT IMAGES USING MODIFIED VGG16 NEURAL NETWORK

A Thesis

Submitted to the Council of the College of Information Technology for Postgraduate Studies of University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology - Software

By

QUSAY AUDAH JAFAR HUSSEIN

Supervised by

Prof. Dr. Israa Hadi Ali Hussein

2022 A.D. **1444A.H.**

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

فَتَعَٰلَى ٱللَّهُ ٱلۡمَلِكُ ٱلۡحَقُّۗ وَلَا تَعۡجَلۡ بِٱلۡقُرۡءَانِ مِن قَبۡلِ أَن يُقۡضَىٰٓ إِلَيۡكَ وَحۡيُهُۥۖ وَقُل رَّبِّ زِدۡنِي عِلۡمٗا (١١٤)

صدق الله العلي العظيم

سُورَةُ طه: الآية 114

Acknowledgements

All the praises and thanks are to ***Allah***, the most beneficent, the most merciful, for his graces that made me able to continue the requirements of my study.

I would like to express my deep appreciation to my supervisor ***Prof. Dr. Israa Hadi Ali*** for her invaluable guidance, supervision and untiring efforts during the course of this work.

Thanks for *everyone* who encourages me with at least one word.

Qusay Audah

Dedication

I dedicate this effort to

My parents, my wife, my beautiful daughter Zahra and all the members of my family.

To all the defenders who love the homeland and who have sacrificed their lives for it, everyone who dreams with a beautiful future for the children of this country, and my dear bleeding country (Iraq) with my hope of rapid healing.

Qusay Audah

Declaration

The work in this thesis, **Detection of Covid-19 in Lung CT Images using Modified VGG16 Neural Network**, is original and no portion of the work referred to here has been submitted in support of an application for another degree or qualification of this or any other university or institution of learning.

Signature:

Date: / / 2022

Qusay Audah Jafar

Supervisor Certification

I certify that the thesis entitled “Detection of Covid-19 in Lung CT Images using Modified VGG16 Neural Network” was prepared under my supervision at the Department of Software/ College of Information Technology/ University of Babylon as partial fulfillment of the requirements of the degree of Master in Information Technology - Software.

Signature:
Supervisor Name: Prof. Dr. Israa Hadi Ali
Date: / /2022

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled “Detection of Covid-19 in Lung CT Images using Modified VGG16 Neural Network” for debate by the examination committee.

Signature:
Prof. Dr. Ahmed Saleem Abbas
Head of Software Department
Date: / /2022

Certification of the Examination Committee

We, the undersigned, certify that (Qusay Audah Jafar) candidate for the degree of Master in Information Technology - Software, has presented his thesis of the following title (Detection of Covid-19 in Lung CT Images using Modified VGG16 Neural Network) as it appears on the title page and front cover of the thesis that the said thesis is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:

Signature:
Name: Mohammed Sahib Mahdi Altaei
Title: Prof.
Date: / / 2022
(Chairman)

Signature:
Name: Amina Atiya Dawood
Title: Assistant professor
Date: / / 2022
(Member)

Signature:
Name: Wadhah Razooqi Baiee
Title: Lecturer
Date: / / 2022
(Member)

Signature:
Name: Israa Hadi Ali
Title: Prof.
Date: / / 2022
(Member) // Main supervisor

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: Hussein Atiya Lafta
Title: Professor
Date: / / 2022
(Dean of Collage of Information Technology)

Index

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Acronym	Meaning
AAN	Artificial Neural Networks
AI	Artificial Intelligence
CNN	Convolutional Neural Network
CT	Computed Tomography
DL	Deep Learning
DLNNs	Deep Learning Neural Networks
GGO	Ground-glass opacities
Grad-CAM	Gradient class activation map
SARF	Size-Aware Random Forest
ML	Machine Learning
ReLU	Rectified Linear Activation
RT-PCR	Real-Time Polymerase Chain Reaction
SNN	Synthetic Neural Networks
SVM	Support Vector Machine
VGG16	Visual Geometry Group
WHO	World Health Organization

List of Publications

(Paper No.1)

1st al Hikma International conference on netural and applied sciences (HICNAS2022).

Paper Title: Deep Learning for Covid-19 Classification Using CT Scan Slices Of Lung

Authors:

Prof.Dr. Israa Hadi Ali

Student. Qusay Audah Jafar

Software dept. –Information Technology-Babylon University

Email: Israa_hadi@itnet.uobabylon.edu.iq

Email: qussaya.msc.sw@student.uobabylon.edu.iq

(Paper No.2)

International conference on Data science and intelligent computing 2022 [ICDSIC2022].

Paper Title: "Detection Covid-19 Infection of Lung CT scan slices images based on a Transfer Learning and GRAD-CAM"

Authors:

Prof.Dr. Israa Hadi Ali

Student. Qusay Audah Jafar

Software dept. –Information Technology-Babylon University

Email: Israa_hadi@itnet.uobabylon.edu.iq

Email: qussaya.msc.sw@student.uobabylon.edu.iq

Chapter One

General Introduction

Chapter Two

Theoretical Fundamentals

Chapter Three

Proposed System Design

Chapter four

The Experimental Results

Chapter five

Conclusions and Future Works

References

Abstract

World Health Organization categorized Covid-19 as a pandemic (WHO). This virus was spreading direct or indirectly among people. The long timeline of PCR tests and lack of test tool kits in many hospitals lead to infection worsening according to the slow diagnosis. The Various experiences of radiologists cause deferent in accurate detection. Moreover, the detection of infection areas in the lungs is a difficult mission. The aim of this thesis is to detect and visualize the infection of covid-19 in the lungs.

The proposed system detects the Covid-19 CT scans and predicts the infected lungs. The preprocessing stage applied to the dataset such as converting from nii to png format, segmentation to extract the region of interest area from the images (lungs area), normalization, and resizing.

The next stage is the proposed model, Transfer learning with Adam optimizer utilized to detect the infection and many architectures had been trained (VGG16, Resnet50, and inception). After training the previous architectures the VGG16 provides the best results, therefore, manipulating on VGG16 has been accomplished to achieve the best accuracy.

The tuning on the last three layers of VGG16 architecture (dense layers) by replacing them with three layers (global average layer, batch_narmalization, and dense layer with binary classification). The last stage of the system is to localize the infection area. GRAD-CAM technique applied on the feature map of the last convolution layer of model which was used to visualize these areas in the lungs. The last convolution layer produces the discriminative features of the target class in a two dimensions matrix.

Two datasets had been used the first dataset “Large COVID-19 CT scan slice dataset”. The second dataset is the “COVID-19 CT Lung and Infection

Segmentation Dataset". This dataset contains the lungs mask to help to extract lungs from images.

The results obtained from the first dataset reached 99.7% of test accuracy with a test loss of 0.0162. The localization of infection with this dataset was not obtained in this thesis. The second dataset reached 99.82% of test accuracy with test lose 0.0044. With this dataset, the GRAD-CAM obtained the visualization of the infection area.

CHAPTER ONE
GENERAL INTRODUCTION

1.1 Introduction

On December 31, the COVID-19 pandemic was discovered when unexamined pneumonia appeared in Wuhan, Hubei, China 2019 [1-3]. In the start, the spreading virus was called SARS-CoV-2, World Health Organization later labeled it as COVID-19 (WHO). In the first stage, the COVID flare-up was announced as a pandemic on 30 January 2020, WHO subsequently declared it a pandemic on March 11, 2020. [4]. There are approximately 14,692,600 patients of C0VID-19 where urgent about 89, 044,900 killed 6, 014,200 people worldwide till the 21st July 2020 pandemic [5]. The virus was admiringly contagious and can be through direct or indirect connection with infected people due to respiratory droplets when they cough, sneeze or even speak [6]. The most important symptoms of the COVID-19 pandemic are headache, cough, throat swelling, sneezing, fever, sore throat, weakness, malaise, and breathlessness [7]. The real-time polymerase chain reaction (RT-PCR) exam is the first formal method for ensuring COVID-19, However, as the number of infected persons rises, most nations lament a lack of testing kits [8]. In addition, RT-PCR tests have lengthy turnaround times and a high false-negative rate. Thus, to combat this pandemic, another testing technique that is speedier is required. Chest computed tomography (CT) is a complementary approach to the RT-PCR testing tool and has been playing an important function in analyzing the early COVID-19 infections [9]. In some cases, several patients had negative PCR examinations; nevertheless, validation was based on their CT results [10]. The duration of the pandemic can be noted by looking at the anomalies caused by COVID-19 lesions from CT image by radiologists [11]. Specifically, Ground-Glass Opacity (GGO) is one of the most frequently assessed parameters to detect and review the COVID-19. [12]. Detection of COVID-19 from chest CT images by radiologists

west time, and manually reviewing every CT image might not be possible in emergency cases. Moreover, not all these radiologists have the same experience [6]. Therefore, there is a need to detect the infection lesion automatically. This was performed by using the advantage of Artificial Intelligence (AI) and deep learning [13, 14].

Convolutional Neural Network (CNN) was a famous field in machine learning techniques that had the most common strategy in the (AI) field later. CNN have been successfully improved on images of medical field investigation such as CT scans [15], X-ray [16], MRI[17], Ultrasonography[18], etc. CNN has commonly used in computer vision, audio recognition, speech recognition, and natural language processing. A neural network also is a chain of algorithms that examines associations in a set of data similarly the human brain function. This algorithm is highly practical in image processing and pattern recognition. CNN builds a model by extracting features from input image through processing operations to recognize some pattern. CNN uses patterns to determine the nearness of new examples as accurately as attainable. This algorithm is characterized by its subordinate sophistication of the network model, adaptability, simple structure, and reduced training network parameters [9]. Recent products in neural network architecture design and its training have helped researchers to acclimate complex tasks of Deep Learning (DL) methods.

In the past, machine learning has some limitations such as its inability to deal with raw input data like feature extraction task [19]. DL can overcome this problem by operating on a huge number of images, therefore, it was an effective and useful strategy of machine learning. However, this needs a lot of data to be trained and avert overfitting. To overcome this problem, transfer learning methods are introduced [19]. In traditional machine learning, training data and testing data most have the same data distribution and same input feature space. A predictive learner can be degraded as a result when there was a difference in data

distribution among the training and testing data. Because of the difficulty and expense of capturing training and predicted data, there was a need to apply transfer learning. Transfer learning means enhancing the learner core from one domain by transferring experience from a related domain [20].

1.2 Problem Statement

The (RT-PCR) test is the standard way to confirm COVID-19. However, most countries are running out of testing kits because the number of people infected is growing so quickly. RT-PCR testing also takes a long time and has a high rate of false negatives. Since the number of people diagnosed with COVID-19 is growing quickly and there are not enough experienced radiologists, mistakes can be made when interpreting the images. So, the main problem is the difficulty to heatmap on infection area because of the similarity between features of these areas and the rest of CT image.

1.3 Literature Review

Some of works which are relative to the proposed work are reviewed below:

1.3.1 COVID or Non COVID Classification

The majority of related works concentrate on proposed or developed models that help to quickly, cheaply, and accurately diagnose COVID-19 in order to prevent the spread of this epidemic, and do so by using deep learning that achieves high accuracy with medical images, particularly the convolutional neural network. Some of the methods create models for 2D or 3D diagnoses. In this province, some similar works have been presented to explain the domain of this work.

1. ***Ohata et al. (2020)***: propose an automatic approach for detecting COVID-19 infection using chest X-rays. CNNs are integrated with consolidated machine

learning techniques, including k-Nearest Neighbor, multilayer perceptron (MLP), Random Forest, Bayes, and support vector machine (SVM). The results show that for one of the datasets, the pair of extractors and classifiers that work best is the MobileNet design with the SVM classifier employing a linear kernel, which yields an F1-score of 98.5% and an accuracy of 99.5%. DenseNet201 and MLP are the best pair for the second dataset, earning 95.6% accuracy and an F1-score of 95.6%. [21]

2. ***Mangal et al. (2020):*** it is recommended that contemporary AI methods be used COVID-19 patients can be found using X-ray images in an automated way. This is especially important in environments in which radiologists are not readily available, and it will also assist the suggested testing system become scalable. Introducing the CovidAID: COVID-19 AI Detector is an innovative approach based on deep neural networks that triages patients for suitable testing. Regarding the covid-chestxray-dataset that is open to the public dataset, model has an accuracy of 90.5 % with 100% sensitivity (recall) in relation to the COVID-19 infection. [22].
3. ***A.I. Khan et al. (2020):*** CoroNet is a Deep Convolutional Neural Network model is made to automatically find COVID-19 infection from chest X-ray images. The proposed model is based on Xception architecture that has been trained on the ImageNet dataset and trained end-to-end on a dataset made up of COVID-19 and other chest pneumonia X-ray images from two different public sources. The prepared dataset has been used to train and test CoroNet, and the experimental findings demonstrate that the suggested model obtained an overall accuracy of 89.6%, with a precision and recall rate for COVID-19 instances of 93% and 98.2%, respectively (COVID vs Pneumonia bacterial vs pneumonia viral vs normal). For 3-class categorization (COVID vs. Pneumonia vs. normal), the suggested approach achieved 95 % accuracy [23].
4. ***Sedik et al. (2020):*** this study presents two data-augmentation models to make it easier for the Convolutional Neural Network (CNN) and the Convolutional Long

Short-Term Memory (ConvLSTM)-based deep learning models to learn (DADLMs), hence enhancing the accuracy of COVID-19 identification. The experimental findings demonstrate an improvement in terms of detection precision, logarithmic loss, and testing time compared to DLMs without such data enhancement. In addition, the suggested data-augmented deep learning models are reported to improve COVID-19 detection accuracy by an average of 4 to 11 % to machine learning approaches. Therefore, the suggested methodology is useful for conducting a quick and consistent diagnosis of Corona virus, which is primarily intended to aid clinicians in making correct identification of the virus [24].

5. ***Wang et al. (2020):*** in this study, they discuss their experience to developing and implementing an AI system that analyzes CT scans automatically and calculates the risk of infection in order to swiftly diagnose COVID-19 pneumonia. The suggested approach, which comprises of classification and segmentation, would reduce physician detection time by 30 to 40% and improve COVID-19 detection performance. Specifically, an interdisciplinary team of approximately 30 individuals with medical and/or AI backgrounds, geographically split between Beijing and Wuhan, is able to overcome a number of obstacles (e.g. testing time-effectiveness of the model, data discrepancy, data security, etc.). The suggested AI approach prioritizes each CT scan based on the risk of infection, allowing clinicians to confirm and separate sick patients in a timely manner. Using 1,136 training cases (723 positives for COVID-19) from five hospitals, the authors were able to obtain a sensitivity of 0.974 and a specificity of 0.922 on a test dataset containing a range of pulmonary illnesses [25].
6. ***Apostolopoulos et al. (2020):*** the purpose of this study is to analyze the performance of convolutional neural network designs for medical image categorization that have been suggested in recent years. Specifically, the Transfer Learning approach is utilized. Transfer learning makes the detection

of a variety of anomalies in tiny medical image datasets a realistic goal that frequently yields spectacular results. There are two datasets used in this investigation. First, a group of 1427 X-ray scans, comprising 224 with proven Covid-19 illness, 504 with normal conditions, and 700 with confirmed common bacterial pneumonia. Secondly, a dataset containing 504 images of ordinary situations and 224 images with verified Covid-19 disease, 714 images with bacterial and viral pneumonia that has been confirmed, and 714 images with confirmed Covid-19 disease. The data is acquired from X-ray images found in public medical archives. The results indicate that Deep Learning using X-ray imaging may extract important biomarkers associated with the Covid-19 illness, with the greatest accuracy, sensitivity, and specificity being 96.78%, 98.66%, and 96.46 %, respectively. Based on the results, doctors could figure out how likely it is that X-rays could be used to diagnose the disease, and more research could be done to look at the X-ray method from different points of view., given that all diagnostic tests currently have failure rates that are the causes for concern [26].

7. ***Jose Francisco and Hernandez Santa Cruz (2020):*** this study contributes a model based on ensemble techniques and 2-stage transfer learning to detect COVID-19 instances based on CT scan images, depending on a basic architecture with a sufficiently complicated model description to achieve a competitive performance. The accuracy of the suggested model was 86.70 %, the F1 score was 85.86 %, and the AUC was 90.82 %. They trained six pre-trained ImageNet models in two phases using a dataset of 746 CT scan images, 349 of which were COVID-19 positive and 397 of which were COVID-19 negative. The ensemble was constructed by adding all of the results together of the models into a single activation neuron. They discovered that using threshold ranges to estimate the risk of a COVID-19-positive case results in improved performance without sacrificing specificity and accuracy. This, augmented by integrated applications, might expedite

COVID-19 testing by delivering findings in a very short amount of time and deriving only scores near to the categorization threshold for a specialist [27].

8. ***Rahaman et al. (2020):*** Deep transfer learning techniques were applied, and 15 pre-trained CNN models were evaluated to determine the most fit for this job. A total of 860 photos (260 COVID-19 cases, 300 pneumonia cases, and 300 healthy) were used to evaluate the performance of the proposed algorithm, where 70% of images from each class are allowed for training, 30% are rejected. This research illustrates the efficacy of deep transfer learning approaches for identifying COVID-19 instances from CXR images utilized for validation, and the remainder for testing. The VGG19 achieves the greatest classification accuracy of 89.3% with average precision, recall, and F1 scores of 0.90, 0.89, and 0.90, respectively [28].
9. ***Asnaoui and Chawki (2020):*** this work compares the detection of coronavirus pneumonia using the most recent deep learning models (VGG16, VGG19, Inception ResNet V3, MobileNet V2, DenseNet201, Inception ResNet V2, and Resnet50). Experiments were done utilizing a chest X-ray and CT dataset of 6087 images (2,780 images of bacterial pneumonia, 1,493 of coronavirus, 231 of Covid19, and 1,583 normal), and confusion matrices were utilized to assess model performances. Using inception Resnet V2 with Densnet201 yields superior outcomes than other models utilized in this study, according to the findings (92.18% accuracy for Inception-ResNetV2 and 88.09% accuracy for Densnet201) [29].
10. ***Zheng et al. (2020):*** weakly supervised deep learning was used to develop a software system for identifying COVID-19 in 3D CT volumes. In order to predict the possibility of COVID-19 infection, segmented the lung region of each patient using a pre-trained UNet, and then fed that 3D lung region into a 3D deep neural network. 499 CT volumes gathered between December 12, 2019 and January 22, 2020 were utilized for training, whereas 131 CT volumes gathered between January 24, 2020 and February 6, 2020

were used for testing. The algorithm for deep learning achieved 0.959 ROC AUC and 0.976 PR AUC. In the ROC curve, there was an operating point with a sensitivity of 0.907 and a specificity of 0.911. Using a probability threshold of 0.5 to identify COVID-positive and COVID-negative individuals, the system achieved an accuracy of 0.901%, a positive predictive value of 0.840, and an extremely high negative predictive value of 0.982. The method only needed 1.93 seconds to process the CT volume of a single patient using a dedicated GPU. Without labeling the lesions for training, our weakly supervised deep learning model can reliably be predicted the possibility of a COVID-19 infection based on chest CT volumes [30].

11. ***Halder and B. Datta (2021):*** to address this pandemic, the suggested technique classifies COVID-19 (positive) and COVID-19 (negative) patients using transfer-learning pre-trained models. They explain the construction of KarNet, a DL framework whose backbone consists of pre-trained models (ResNet50V2, VGG16, DenseNet201, and MobileNet). KarNet, a simple two-dimensional DL framework, demonstrates excellent diagnostic performance in detecting COVID-19 patients from lung CT-scan images. The KarNet model is pre-trained through transfer learning. In this investigation, each model is trained on both unaugmented and augmented datasets to learn more about what the framework can do. DenseNet201 seems to do the best, as it achieved a 97% accuracy rate on the testing and validation datasets for the two models, i.e. those trained using augmented and unaugmented CT-scan datasets. On the test images, MobileNet, VGG16, and ResNet50V2 also shown promising accuracy. The proposed architecture significantly enhances the model's diagnostic capability and achieves an outstanding AUC score [31].
12. ***Debaditya Shome et al. (2021):*** the suggested transformer model distinguishes COVID-19 from normal chest X-rays with 98% accuracy and a

99% AUC score in the binary classification task. In the Multi-class classification challenge, it separates COVID-19, normal, and pneumonia patient X-rays with an accuracy of 92% and an AUC score of 98%. As baselines for evaluating a data set, they fine-tuned several widely used models from the literature, including DenseNet-121, MobileNetV3, InceptionV3, Resnet50, Xception, and fficientNetB0. In every metric, the proposed transformer model surpassed the competition. Also, a GRAD-CAM-based visualization is made that shows how their method works interpretable by radiologists and can be utilized to track the evolution of the disease in the affected lungs, thereby aiding healthcare professionals [32].

13. ***Sneha Kugunavnd et al. (2021)***: using X-ray and CT images of COVID-19 patients, this study describes the application of CNNs for the diagnosis and prognosis of the disease. The CNN models mentioned in this article were primarily designed for COVID-19 image detection, classification, and segmentation. VGG16, AlexNet, residual network, GoogLeNet, DensNet, MobileNet, Inception, and extreme Inception were the foundational models utilized for detection and classification. For segmentation, U-Net and voxel-based wide learning networks were utilized. These techniques proved effective for detecting the presence of COVID-19, even with limited datasets. To further corroborate these findings, an experimental investigation employing a basic CNN framework for the binary classification of COVID-19 CT images was done. It was achieved 93% accuracy with an F1-score of 0.93. Consequently, with the availability of enhanced medical image datasets, it is obvious that CNNs are extremely valuable for the accurate diagnosis and prognosis of COVID-19. [33].
14. ***Wang et al. (2021):*** by gathering 1000 CT scans of pathogen-confirmed COVID-19 patients as well as those previously diagnosed with conventional viral pneumonia. To establish the method, they modified the inception architecture, internal and external validation following. The external testing

dataset exhibited an overall precision of 79.3%, with a specificity of 0.83 and a sensitivity of 0.67. The overall accuracy of internal validation was 89.5%, with a specificity of 0.88 and a sensitivity of 0.87. In addition, the first two nucleic acid test results were negative in 54 COVID-19 images, whereas the algorithm accurately predicted that 46 images were positive for COVID-19 in 85.2 % [34].

15. ***Shi et al. (2021):*** in this study, 1,658 COVID-19 patients and 1,020 CAP persons who received thin-section CT were recruited. Every image was preprocessed to produce infection and lung field segmentation. Compared to the traditional CT severity score (CT-SS) and radionics characteristics, a collection of handmade location-specific features was presented to represent the COVID-19 distribution pattern. A size-aware random forest approach was developed to distinguish COVID-19 from CAP. Experimental findings indicate that the suggested technique performed best when employing handcrafted features, with a sensitivity of 90.7%, a specificity of 87.2%, and an accuracy of 89.4% compared to state-of-the-art classifiers. Tests on 734 more participants with thick slice images indicate a high degree of generalizability. The suggested approach is predicted to facilitate clinical decision making [35].
16. ***E. A. Abbood and T. A. Al-Assadi (2022):*** A innovative and efficient technique for extracting relevant and meaningful characteristics from CT scans and X-Ray COVID-19 images is presented. The suggested technique relies on the GLCM method to extract statistical texture information from images. The GLCM matrices are recovered from three separate quantized copies of the source image at various distances and orientations. New multi-inputs 1D CNN architecture of the deep neural network is built to directly extract effective features from GLCM matrices using PCA to reduce their dimensions. Their technique is evaluated using three datasets, including COVID-CT, SARS-CoV-2 CT scan, and DLAI3 Hackathon COVID-19

Chest X-Ray datasets. The suggested system demonstrated a classification that is increase in terms of accuracy, F1 score, and AUC metrics in comparison to existing approaches, exceeding 98%, 89%, and 93%, respectively, for three datasets [36].

1.3.2 Localizing the infection in the lungs

1. ***H. Panwar et al. (2020):*** utilizing chest X-ray and CT-scan images, they present a deep transfer learning system that speeds the detection of COVID-19 patients. They evaluated three datasets referred to as 1) Chest X-Ray Images (Pneumonia), 2) SARS-COV-2 CT-scan, and 3) COVID-chest X-ray. They have employed VGG19 with a color visualization technique based on Grad-CAM to provide clear interpretation of radiologist image identification and subsequent action [37].
2. ***Umair, Muhammad et al. (2021):*** for the identification and categorization of COVID-19 in this study, a transfer learning approach depends on fine-tuning. Four architectures models i.e., DenseNet-121, ResNet-50, VGG16, and MobileNet were employed. Previous deep neural networks were trained using a dataset of 7222 chest X-ray images (COVID-19 and normal) (available on Kaggle). A dataset of 450 chest X-ray images of Pakistani patients was compiled for testing and prediction purposes. To validate the correctness of the models, several crucial measures such as recall, specificity, F1-score, precision, loss graphs, and confusion matrices were produced. MobileNet, ResNet-50, VGG16, and DenseNet-121 have attained accuracies of 92.48 %, 83.27 %, 96.49 %, and 96.48 %, respectively. For the purpose of displaying feature maps that represent the breakdown of an input image into distinct filters, intermediate activations are visualized. The GRAD-CAM technique was then used to generate class-specific heatmap images to emphasize the retrieved characteristics from the X-ray images [38].
3. ***S. Ahuja et al. (2021):*** COVID-19 transfer learning is applied to CT scan images decomposed using stationary wavelet. To increase detection accuracy,

a three-phase methodology is proposed: Phase1: Adding to the data with stationary wavelets. Phase2: Finding COVID-19 with a pre-trained CNN model. Phase3: Finding abnormalities in CT scan images. This work evaluated SqueezeNet, ResNet101ResNet18, and ResNet50. 70% of images are used to train and 30% to verify the network [39].

4. ***T. Zebin et al. (2021):*** used transfer learning to categorize COVID-19 chest X-rays from two public datasets1,2. The classifier distinguishes COVID-19 and pneumonia lung inflammation from non-infected lungs (normal). They employed several pre-trained convolutional backbones as feature extractors and obtained 90%, 94.3, and 96.8% detection accuracy for the EfficientNetB0, VGG16, and ResNet50 backbones correspondingly. They trained an ycleGAN to produce and augment the minority COVID-19 class. For visual explanations and interpretation, the researchers used GRAD-CAM to emphasize relevant image regions. These visualizations can also be used to evaluate disease development and severity [40].
5. ***H. Alshazly et al. (2021):*** this article looks at how well deep learning models trained on CT scans of the chest can automatically position COVID-19 infections. They used sophisticated deep network topologies and custom-sized input to obtain the greatest performance. They ran extensive tests COVID19-CT. The study reached accuracy, precision, sensitivity, specificity, and F1-score values of 92.94%, 91.34%, 93.74%, 92.24%, and 92.54% respectively on the COVID19-CT dataset. These algorithms can detect COVID-19 cases and give accurate localization of COVID-19-associated areas, according to well-trained radiologists [41].

1.4 Challenges of Thesis

The lack of open-source datasets on COVID-19 CT images provides a significant challenge to the development of more advanced artificial intelligence algorithms for improved CT detection in COVID-19 testing.

Moreover, the images of these dataset were combined from different devices in different luminance and intensity information. That made difficult challenge to work with.

Another challenge was the localizing of the infection area in lungs due to the Deep Learning technique as considered as black box, moreover, the features of infection area quite similar to the features of surrounding body feature.

1.5 Aims of Thesis

The aim of this thesis presents a better solution for the problem of detection the infection area in CT images of lung with heatmap, through the following steps:

1. Design an intelligent model with modified VGG16 to detect and localize the infected area in CT images of lung.
2. Eliminating some layers from the Deep Learning network to reduce its complexity and resource usage.
3. Detecting the infection of COVID-19 in lungs from CT scans images.
4. Localizing the infection area. This was accomplished by using the Deep Learning technique and GRAD-CAM method.

1.6 Thesis Layout

In addition to chapter one the remaining of the thesis arranges as follow:

Chapter Two:

It represents general information machine learning, and deep learning. The chapter also explains the CNN components. Finally, it explains the localization method by using GRAD-CAM technique.

Chapter Three:

It explains the proposed system steps and the techniques used and displays the results of each algorithm used in the proposed system. The three phases of proposed system explained, preprocessing, model extracting, and infection localizing. Moreover, the algorithms that used in this thesis was illustrated.

Chapter Four:

It depicts the results obtained from the stages in each phase of the proposed system. It also discusses the evaluation of the system's performance that used for this purpose.

Chapter Five:

Mentioned precisely the conclusions, it includes what was discovered in this thesis, and potential future research directions to improve this work.

CHAPTER TWO
THEORETICAL FUNDAMENTALS

2.1 Introduction

A multidisciplinary field, computer vision allows computers to cope with high-level knowledge of digital images or videos automatically in a similar way to the human visual system. To accomplish these tasks, you need to acquire, process, analyze, and understand digital images or videos, as well as extracting a high-dimensional data from the actual world. Driver drowsiness, surveillance, and emotion detection can all benefit from the usage of computer vision techniques [42].

Machine learning is the study of how to make a computer make a decision without explicitly teaching it how to do it. Problems with machine learning may be found in a range of fields, including medicine (diagnosis, drug discovery, prediction), the internet (search engine optimization) (text categorization), robotics, and pattern recognition (speech, handwriting, and face recognition). To do this, a model is fed a massive amount of data, along with all of the projected outcomes. This model learns a mathematical model of how to link the response, or label, to the input, so it can anticipate and generalize the proper answer for unknown inputs. Thus, machine learning relies on a large amount of data to discover the relationship between patterns and outcomes. As a result, this type of learning is referred to as supervised learning [43].

In Deep learning models, additional layers are used to extract more features from raw data, it is possible to make complex features. For example, in image processing the boundaries may be detected at the lower levels, while the relevant elements like digits, letters, or faces can be identified at the upper layers.

Artificial neural networks (ANNs) are the basis of the majority of current deep learning models, which are sometimes referred to as Deep Learning Neural Networks (DLNNs) (CNN) [44].

This chapter will discuss some concepts such as Medical Images, Computer Vision, Machine Learning, Neural Networks, Deep Learning, Gradient Class Activation Map (GRAD-CAM), Region of interest extracting, and Evaluation Measures.

2.2 Medical images

Utilizing imaging technologies to get a better view at what is going on within a patient's body enables medical professionals to more accurate and quickly diagnose disease. It makes be feasible to perform keyhole surgery, which allows surgeons to access internal organs without having to completely open up the patient's body [45]. There are many types of medical images that used to diagnosis diseases. Now, we can describe the types of medical images:

2.2.1 Computer Tomography (CT)

Computed tomography (CT) employs x-rays to provide several views of a specific area of the body. These images are used to build a cross-sectional image by the computer. A two-dimensional image is the end outcome of this process. Several slides will be created as a result of this procedure. It is possible to identify lung infections using CT scans once a variety of clinical signs and

symptoms have been seen [46]. Figure (2.1) illustrate CT scan images.

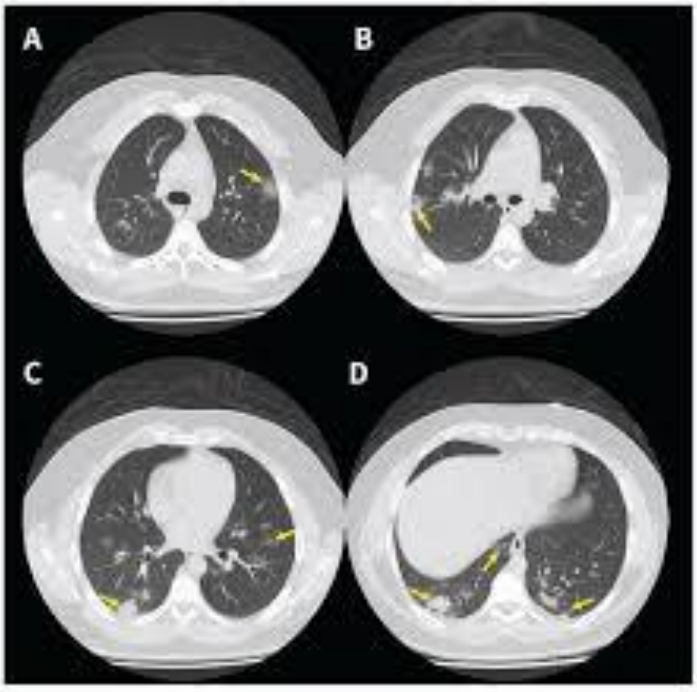


Figure (2.1) CT scan images [46]

2.2.2 X-Ray Images

The X-ray was one of the discoveries that Conrad Roentgen made. It was altogether new kind of radiation that could pass through almost everything. In 1901, the Nobel Prize in Medicine was awarded to him for describing the diagnosis the capabilities of X-rays to provide a visual representation of the human body. Radiation is being emitted through the body, and the resulting images can be recorded on film or stored in a computer. This is the most straightforward approach to diagnostic imaging in the medical field. Utilized frequently. With regard to the X-ray, photoelectric, and Compton absorptions, the term "effects" is used to refer to the attenuation of X-rays caused by soft tissue, bones, and other obstacles. Tissues, as well as air X-ray wavelengths are far shorter than those of UV or visible light. The electromagnetic spectrum includes

something called the wavelength. It is critical to use X-rays whose wavelengths have linear attenuation coefficients for the materials they are aimed at human body in medical applications [47]. Figure (2.2) represent x-ray images.

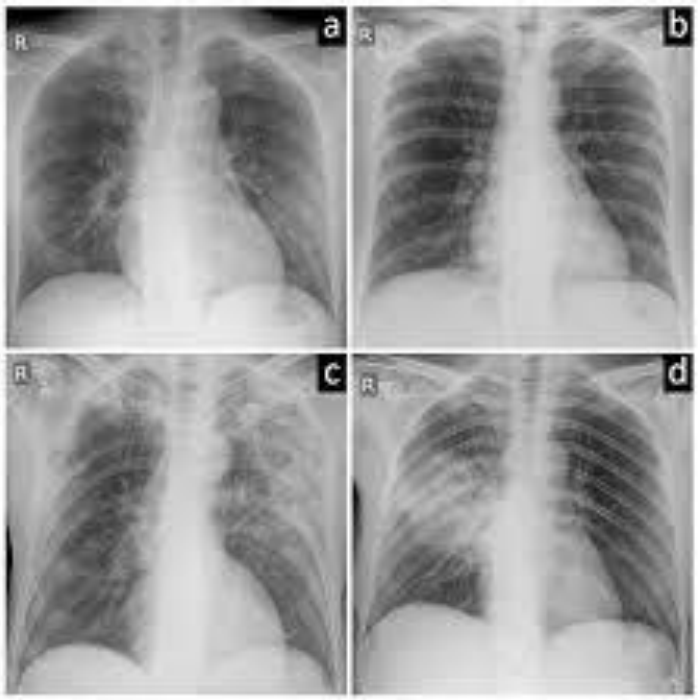


Figure (2.2) X-ray images [47]

2.2.3 MRI Images

MRI is a type of imaging that is mostly used in medical settings to get clear images of what is going on inside a person's body. Unlike traditional X-ray imaging or CT scans, it makes pictures that show the X-ray attenuation of tissues and measures the amount of water and fat in tissues. Nuclear magnetic resonance (NMR) is a spectroscopic method that can be used to find out about the chemical and physical properties of molecules at the molecular level. In the late 1970s, the word "nuclear" had a bad reputation, so this technology was called MRI instead of NMRI. Tomography, which came before MRI, is used to look at NMR signals and make slices through the human body [48]. Figure (2.3) explain MRI scan.

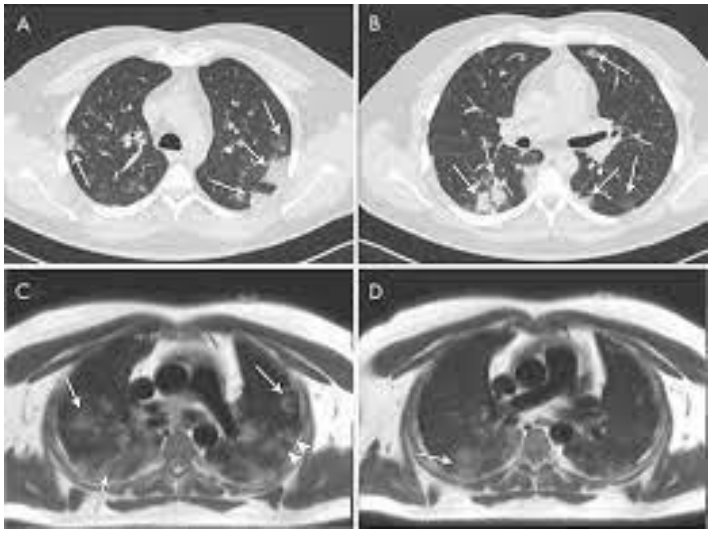


Figure (2.3) MRI scan [48]

2.3 Computer Vision

Computer vision tasks include getting, processing, analyzing, and understanding digital images. They also include getting high-dimensional data from the real world to give numerical or symbolic information, like judgments. In this circumstance, understanding means turning visual images (retinal input) into descriptions of the world that make sense to cognitive processes and can lead to the right action. This image comprehension can be viewed as the separation of symbolic information from visual data utilizing models developed with the aid of geometry, physics, statistics, and the theory of learning [49,50].

The field of computer vision focuses on the theory underlying artificial systems that glean data from images. The image data can come in a variety of formats, including video sequences, images from several cameras, multidimensional data from a 3D scanner, and medical scanning devices. Computer vision tries to apply its theories and concepts to the development of computer vision systems [51].

Object identification, learning, Scene reconstruction, event detection, video tracking, 3D scene modeling, object recognition, 3D pose estimation,

motion estimation, visual serving, and image restoration are all sub-domains of computer vision [52].

Computer-assisted diagnosis (CAD) of lung CT images has been a significant and innovative advance in the identification of lung abnormalities in their earliest and most treatable stages. The CAD systems incorporate technologies for the automated identification of lung nodules and the 3D modeling of lung systems, which aid the radiologists in making their final conclusions. On the images, various image processing methods are employed to clarify and improve the image before isolating the region of interest from the rest of the image [49].

2.4 Machine Learning

Machine learning (ML) is the study of how computers and other electronic devices can "learn," or make use of available data in order to improve their performance on a given task or set of tasks. The term "learning" is used to refer to the process of improving performance. It is generally considered to be a part of artificial intelligence [43]. The goal of the algorithms used in machine learning is to be able to make predictions or choices without being explicitly programmed to do so. This is accomplished by building a model on the basis of sample data, which is also referred to as training data. ML algorithms are utilized in a vast array of applications, including medicine, computer vision, speech recognition, and email filtering. These applications are utilized in areas where it is challenging or impossible to design traditional algorithms to carry out the necessary tasks [53].

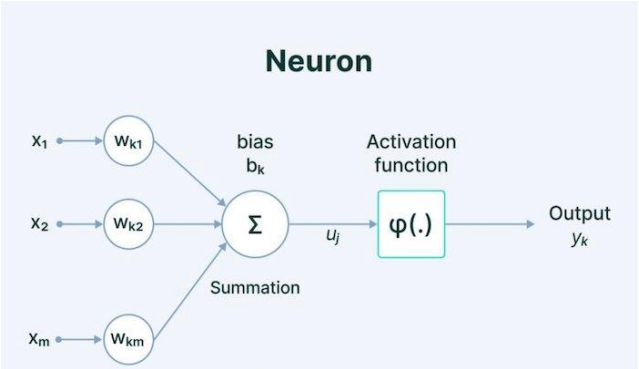


Figure (2.4) Neural network [57].

Not all machine learning is the same as statistical learning, but a subset of machine learning is computational statistics, which is focused on making predictions using computers. The study of mathematical optimization leads to new methods, theories, and application domains that can be used in the field of machine learning. Some applications of machine learning make use of neural networks and data in a fashion that is intended to simulate the way in which a biological brain performs its functions. Machine learning, which applies ML techniques to a wide range of business challenges, is sometimes referred to as predictive analytics [54].

2.5 Neural Network

Two forms of deep learning algorithms based on machine learning are artificial neural networks (ANNs) and synthetic neural networks (SNNs). (ANNs) are machine learning approaches and techniques that are often used to represent the learning processes of biological organisms. Neurons are the cells that make up the central nervous system of humans. Synapses are the points where axons and dendrites meet. In response to external sensory inputs, the frequency of synaptic connections fluctuates greatly. This transformation took place in the way lifeforms think [55].

In an ANN, a node layer is usually consisting of three parts: an input layer, one or more hidden layers, and an output layer. As shown in Figure (2.1), each node, or artificial neuron, is linked to the other nodes and given a weight and a threshold. When a node's performance hits a certain level, it is turned on and data is sent to the next layer of the network. If not, no data has moved to the next layer [56]. Figure (2.1) shows the neural network.

2.5.1 Perceptron Neural Network:

The Perceptron neural network is the earliest model of the ANN. To put it another way, the perceptron contains one neuron, which takes in numerous inputs, each of which has its own weight, and outputs one output (0 or 1, etc.). A single neuron is seen in Figure (2.2) (perceptron). The dot product of the input xi, the weights wi, and the bias neurons b used to make the model fit better is represented by the assembly symbol (∑). Finally, apply the signal activation function is shown by the symbol (⊣) to the output node to be aggregated to provide the final class label corresponding to the input data, with bias set to a constant value of 1. To determine the output, use the activation function as given in equations (2.1) and (2.2), which add the product account. The equations (2.1) and (2.2) have been used for computing the activation function [58].

$$\mathbf{net} = \sum_{i=1}^{n}(x_i w_i) + b \qquad \ldots\ldots\ldots (2.1), [58].$$

$$f(\mathbf{net}) = \mathbf{O} = f(\sum_{i=1}^{n}(x_i w_i + b)) \qquad \ldots\ldots\ldots\ldots (2.2), [58].$$

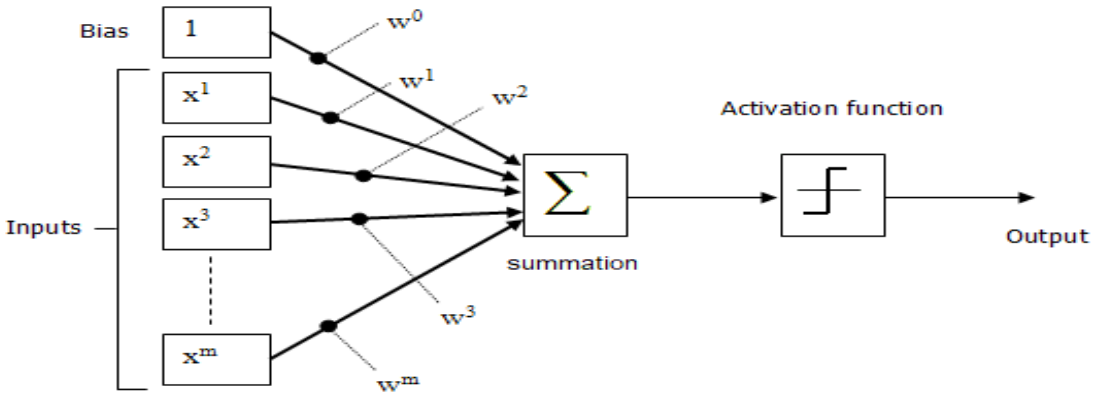


Figure (2.5) Sample perceptron Neural Networks [59].

2.5.2 Multi-Layer Perceptron

As shown in Figure (2.2), a shallow neural network (including at least one hidden layer) may be employed to tackle simple tasks. This neural network has three layers: the input layer, the hidden layer, and the output layer. Each layer of this diagram has one or more nodes, which are shown as small circles. Information only moves from the input to the output of this neural network (that is, from left to right) [58].

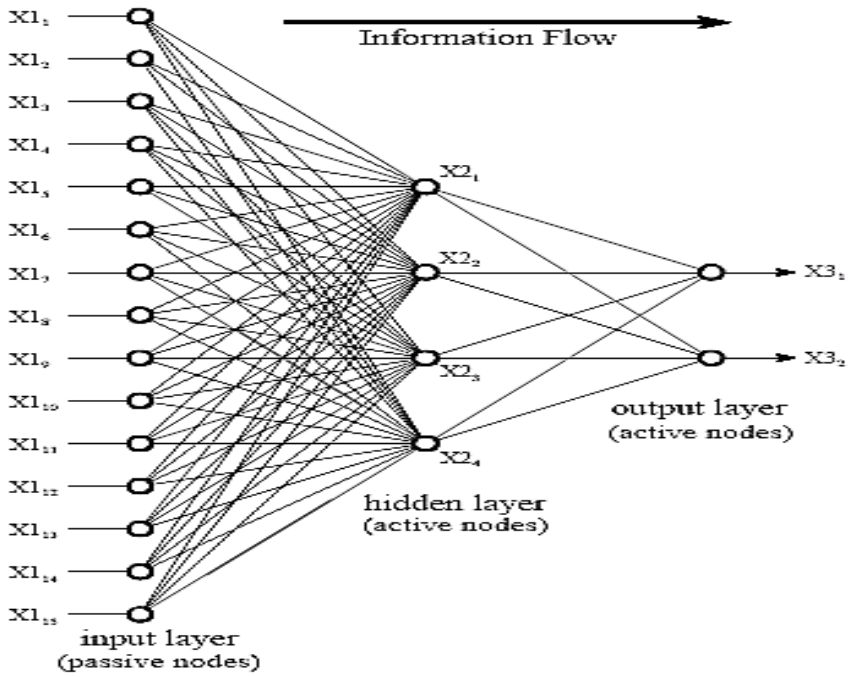


Figure (2.6) Neural Networks have a structure with three layers that are all connected. [58].

The nodes of the input layer are passive. On the other side, the output and hidden layer nodes are engaged. This indicates that they modify the data as indicated in Figure (2.3). The data to be evaluated is included in X11, X12,..., X15 Pixel values from an image. As illustrated in Figure (2.4), the values of the input layer that enter the hidden node are multiplied by the weights, which are a collection of in-software-stored predetermined integers. The sum of the weighted

elements results in a single digit. This is represented by the symbol (∑) on the graph. Before passing the node, this single digit is processed by the sigmoid, a nonlinear mathematical function.

This "S"-shaped curve regulates the performance of the node. A value between - and + is fed into the sigmoid, but the product can only be between 0 and 1 as illustrated in the equation (2.3). In Figure (2.4), the variables X21, X22, X23, and X24 reflect the outputs of the hidden layer. As previously, each of these values is replicated and added to the following layer. Active output layer nodes gather and modify input to create X31 and X32, the network's two output values. If it is necessary to detect a complicated pattern, such as a visual pattern, a deep neural network (DNN) with many hidden layers is required [58].

2.6 Deep Learning

Deep learning algorithms have become the most popular, with significant gains. In deep learning, CNN is a prominent method. CNN is a neural network approach composed of layers. Each of these layers is in charge of a specific action, such as convolution, loss, pooling, calculation, and so on. The output of the previous layer is the input of the next layer [60]. For a long time, CNN has been widely employed in the field of computer vision but they were not used optimally until the ImageNet competition in 2012, this comes with revolution in the efficient utilization of (GPUs) and effective data augmentation [61].

Deep learning is a series of processing layers that builds computer models for identifying subtle structures in huge datasets by learning representations of data with various levels of abstraction. There is a clear concern in how a machine may change the parameters needed to generate the representation of each layer

from the preceding one. Bringing machine learning and artificial intelligence together, this is a novel method to machine learning [62].

The step of extracting meaningful and salient features is essential in machine learning. But traditional machine learning is limited in processing natural information and dealing with it in its raw form, and this limited ability makes it difficult to solve problems in artificial intelligence, and to solve these problems, Deep Learning is designed[63, 64].

Deep learning had numerous medical applications. In [65] the authors introduced a deep learning-based model for classifying images based on age and gender. In [66] the deep learning-based approach developed to classify human facial expressions. In [67] based on deep learning, the scientists created a model to detect head irregularities while driving the car. In [68] deep learning was used to classify human eyes as open or closed while driving in order to prevent accidents. In [69] deep learning was used to classify head movement in order to diagnose a disease.

2.6.1 Activation Function

In artificial neural networks, activation functions take an input signal and turn it into an output signal. That output signal is then sent to the next layer as an input signal. Measure the number of input products and their related weights in an ANN, then add an activation function to obtain the output of that layer and feed it as an input to the next layer [70].

Linear activation functions and nonlinear activation functions are the two main types of activation functions. When the linear activation function is used, the output is similar to the input. This means that it can only adapt to linear changes and can only be used to solve simple problems with some error. Since

the derivative value of the input has to stay the same, it can not solve complex and nonlinear problems that are needed for the backpropagation process [71].

The nonlinear activation functions whose plotted curve has more than one degree. A multi-layer neural network must be able to learn, represent, and interpret any data and complicated issues that relate inputs to outputs. In order to generate non-linear mappings between inputs and outputs, an activation function must be used. A key feature of the nonlinear activation function is that it can decrease errors through backpropagation [70]. Important activation functions for usage in hidden layers include the following.

2.6.1.1 Rectified Linear Activation (ReLU)

Rectified Linear Units (ReLU) it is often employed in Neural Networks, particularly in Deep Learning models, and operates by setting the threshold to 0. Simply said, it returns 0 when x is zero or negative and returns the same value in all other cases. The function denoted by the equation (2.3). Figure (2.6) displays the plot of the ReLU Activation Function [72].

$$ReLU(\mathbf{x}) = max(0, x) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.3), [72].$$

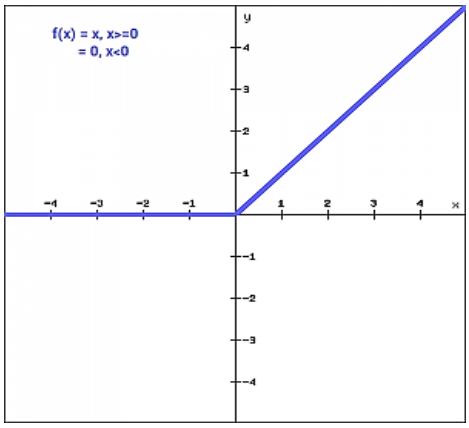


Figure (2.7) ReLU Activation Function [73].

2.6.1.2 Sigmoid

Due to the fact that sigmoid is a nonlinear function, it is the most frequently employed, particularly in binary classification. The sigmoid function alters values in the interval 0 to 1 as seen in Figure (2.7), it may be expressed as equation (2.4) [74]:

$$f(x) = 1/(1 + e - x) \qquad \text{... ... ...(2.4), [74].}$$

The sigmoid function is a smooth, continuously differentiable S-shaped function [70]. The derivative of the function is as an equation (2.5), figure (2.7) shows the sigmoid activation function.

$$f'(x) = 1 - sigmoid(x) \qquad \text{... ... ... (2.5), [74].}$$

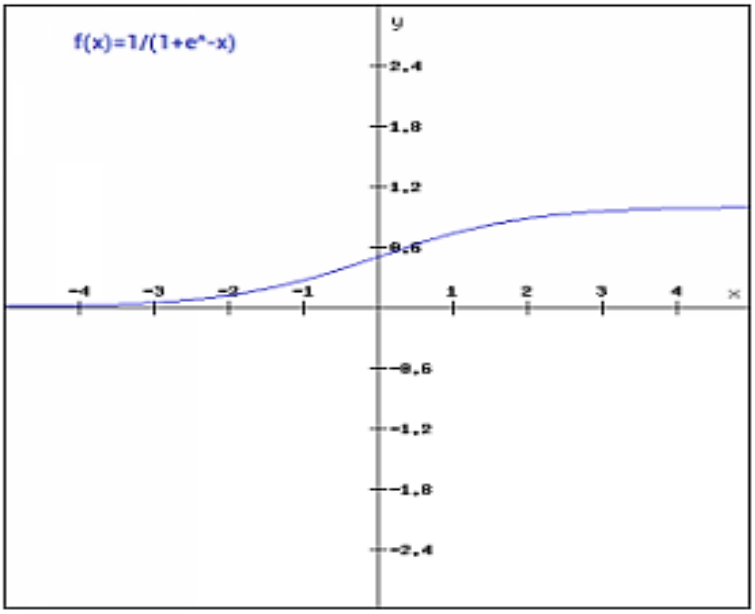


Figure (2.8) Logistic (Sigmoid) Activation Function [70].

2.6.1.3 SoftMax

The SoftMax function is one of the activation function types, and it is commonly employed in neural computing at the last layer to calculate the multiple probability distributions of multi-classes with more than two classes by employing a vector of real values. The result of the Softmax function is a number between 0 and 1 where the total of the probabilities equals 1, and the class with the highest value is the target class. Softmax represented in equation (2.7) [71].

$$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_i)} \qquad \text{... ... ...(2.7), [71].}$$

2.6.2 Loss Function

Early models of neural networks used the difference between the actual output and the predicted output to measure error. For calculating neural network error, numerous formulae have emerged, known as Loss Functions [75].

If the network is trained with a variety of loss functions, each one will provide a different error value for the same prediction. Loss functions can be divided into three categories: Classification Loss Functions, Regression Loss, and Embedding Loss Functions [76].

The classification loss functions are utilized for classification issues. In regression issues, regression loss functions are utilized. While the Embedding loss functions are utilized for jobs that require measuring the similarity between two inputs. Training a neural network with some of the most widely used loss functions [77].

2.6.3 Optimization Algorithms

Choosing an algorithm to optimize a neural network is one of the most important steps. Batch or deterministic gradient techniques, which handle all training examples in a large batch at once, and stochastic or online methods,

which deal with only one instance at a time, are the primary types of optimization methods in machine learning [76].

Adaptive learning algorithms and constant learning rate algorithms, such as SGD, are two types of optimization algorithms. There is a manual selection of a learning rate in the first group. In this kind of algorithm, picking the learning rate is a challenging problem. Choosing a low learning rate slows down the learning process and increases the training duration. The loss value may fluctuate around the minimum value if you choose a reasonably high learning rate, which slows down the convergence process. The algorithms in the second group, on the other hand, do not need to have the learning rate set by hand. Instead, they use a heuristic approach that takes care of adjusting the value of the learning rate [78]. Several algorithms that fit into these two groups have been developed, with the most important ones shown below:

2.6.3.1 Adam

This adaptive learning rate optimization technique (Adam) assesses the rate at which each person learns a certain parameter (or set of parameters). In order to set Adam's learning parameter, Adam used the first and second-moment estimations. A random variable at the power of n is the moment's expectation. The moment can be shown mathematically [79]:

$$m_n = E[X^n] \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(2.9), [79].}$$

Where: m is the_moment, and X is a random_variable.

The following equations are used to estimate, the first and second moment Adam [79].

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.10), [79].$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.11), [79].$$

Where m t is the previous first moment and v t is the previous second moment. In the first step, both of these values are set to 0.

β1 and β2 are two new parameters that have been added to the algorithm. They are set to 0.9 and 0.999 as their default values.

After figuring out the value of the first and second moments, the following equation (2.12) is used to update the network weights [79].

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.12), [79].$$

Where W is network weights, η is Step size, $\boldsymbol{\epsilon = 10^{-8}}$

2.6.4 Back Propagation technique in Neural Networks

Back-Propagation (BP) is the most important part of getting a neural net ready. It is a technique for fine-tuning the weights of a neural network using the error rate from the previous epoch (i.e., iteration). By reducing the number of errors and making the model more general, adjusting the weights improves the model's precision [77]. The BP algorithm, which uses the weights update method, is a popular way to train a network with more than one layer to recognize a certain pattern. Using a function known as the optimization function, the BP algorithm attempts to reduce the difference between the desired output and the predicted output by modifying the initial weights [80]. Most optimization algorithms measure the gradient, which is the partial loss function's derivative

with respect to weights. Based on the measured gradient, the weights are changed in the opposite direction. This loop is done over and over until the model stops working well enough [76]. Figure (2.9) shows how to find the gradient and how the loss function is related to the weights of the network.

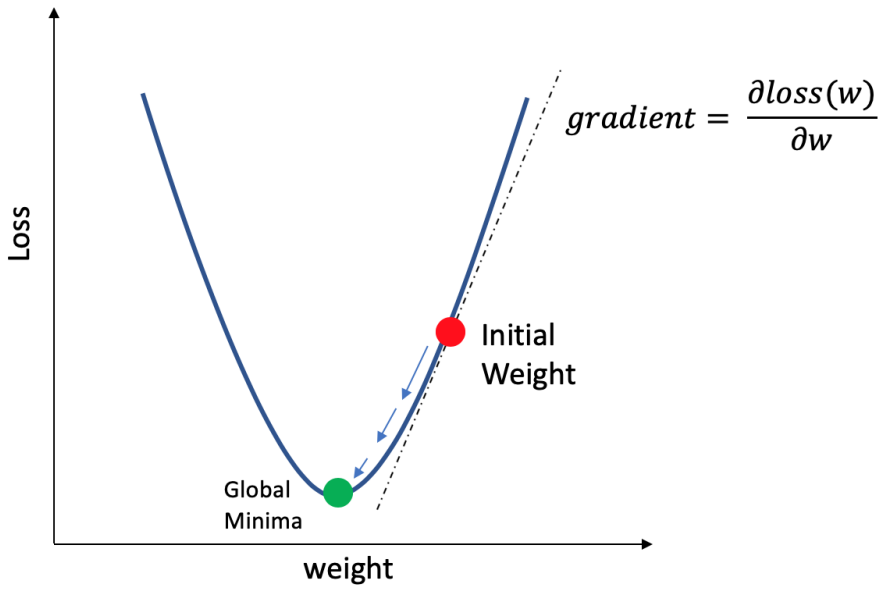


Figure (2.9) Figuring out the relationship between the gradient and the loss function [81].

The BP algorithm consists of forward and reverse steps:

a. Forward step: The procedure is depicted in perceptron Figure (2.4).

1- Input patterns are sent forward into the network.

2- Compute output predictions.

- Compute the net, which is the sum of product weights and input, as shown in equation: (2.1).
- Using the activation function, compute f(net), which is the expected output referred to as 1 out or 0 for the net as in equation (2.2).

3- Compare desired (target) and predicted (out) outputs, error differences between the desired and predicted output, for each neural output must be calculated using loss function such as cross-entropy, which is the preferred

one to be used with the output that is distributed according to the probability of each class, therefore cross-entropy loss function is a suitable loss function for softmax or sigmoid classifier that returns the probability of each class. The equation appears below in formula (2.13) [81].

$$H(out) = -\sum_i (\boldsymbol{target})\ \mathbf{log}(\boldsymbol{out}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(2.13), [81].}$$

b. Reverse step: the procedure is depicted in Figure (2.10) below.

1- Errors that have been precalculated must be propagated in reverse form.

2- Gradient descent, one of the most common optimization techniques employed with the back-propagation method, is used to get the projected output closer to the desired output. Total error is largely derived from E(total) to make adjustments. The error in regard to a specific weight is also known as gradient descent with respect to w1, for instance, using the chain rule as shown in the equation below: (2.14) [81].

$$\frac{\partial\, \boldsymbol{E(total)}}{\partial\, \boldsymbol{w1}} = \frac{\partial\, \boldsymbol{E(total)}}{\partial\, \boldsymbol{out(o1)}} * \frac{\partial\, \boldsymbol{out(o1)}}{\partial\, \boldsymbol{net(o1)}} * \frac{\partial\, \boldsymbol{net(o1)}}{\partial\, \boldsymbol{w(i)}} \ldots\ldots\ldots \text{(2.14), [81].}$$

3- Weight difference value (ΔW) is obtained by multiplying the resultant gradient descent with regard to a particular weight value by the learning rate (η), as shown in the following equation (2.15) [82].

$$\Delta W = \eta \frac{\partial\, \boldsymbol{E(total)}}{\partial\, \boldsymbol{w(i)}} \quad \ldots\ldots\ldots\ldots\ldots \text{(2.15), [82].}$$

4- The present weight will be modified by subtracting the weight difference, as stated in the following equation (2.16) [82].

$$W(i)_{new} = W(i) - \Delta W \quad \ldots\ldots\ldots\ldots \text{(2.16), [82].}$$

By substituting (2.16) with (2.17), the new weight can be obtained by applying the following equation (2.17) [81].

$$W(i)new = W(i) - \eta\ (\partial\ E(total))/(\partial\ w(i)) \text{ ..... (2.17), [81].}$$

5- In a multi-layer or deep network, the weight-adjustment process continues to return to the hidden layer in order to update the weights, and then continues to return into the input layer from the output layer until the error-difference falls below the upper error bound (Emax). This phase is named backward phase because a gradient learning process begins at the output node and continues going backward [83].

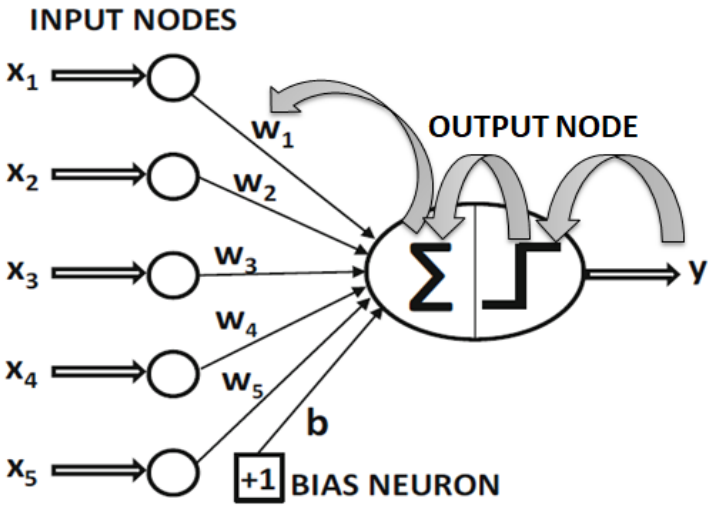


Figure (2.10) Backpropagation process in neural network [83]

It is vital to notice that the most significant back-propagation concern is the gradient vanishing. Vanishing gradient problem can be characterized as the difficulty encountered when training a neural network due to a small weight update received from the preceding layer, resulting in a small weight shift that, in the worst scenario, can totally halt the training process. Due to the range of activation functions, some can compound the problem while others can solve it. ReLU is now the most essential or default activation function due to its ability to maintain only positive values while keeping negative values close to zero [84].

2.6.5 Dimensionality Reduction

Many fields such as numerical analysis, data processing, and machine learning are faced with the problem of curse dimensional. The popular theme of curse dimensional problems is that the amount of space increases so rapidly that the data available are sparse with the dimensionality increases. Deep learning has overcome this problem due to the enormous ability of its algorithms to handle multiple dimensions [85].

2.6.6 Types of Learning

Deep learning models are classified into three categories, each with its own network architecture and application.

2.6.6.1 Supervised Learning

In Supervised Deep Learning methods, the model is trained on data that has already been labeled. During the testing phase, the model should decide the proper response without depending on a label determined by the learning algorithm. The two primary applications of supervised deep learning are classification and regression tasks. Convolution neural networks are among the most frequently used supervised models (CNN). The CNN-based pervised deep learning architectures include VGGNet, InceptionNet, AlexNet, LeNet-5, GoogleNet, Resnet, and others [86].

2.6.6.2 Unsupervised Learning

In unsupervised Deep Learning techniques, the model is trained on unlabeled data and attempts to automatically extract patterns and features. Restricted Boltzmann Machine (RBM), Deep Belief Network (DBN), and other unsupervised deep learning architectures are some examples [87].

2.6.6.3 Semi Supervised Learning

Semi-Supervised Deep Learning is an intermediate learning step between supervised and unsupervised learning. A small quantity of categorized data is required to support a greater volume of unlabeled data. This strategy is useful when obtaining important data characteristics is difficult and labeling samples is time-consuming. General adversarial networks are a prevalent use of this strategy (GANs) [88].

2.6.7 CNN with 2D Architecture

This topic focuses on the architecture that must be implemented to manage a certain type of data, Convolutional Neural Networks (CNN) are a biologically inspired form of feed-forward networks in which the connections between neurons try to record input data distortion or shift pattern invariances. Most CNN architectures assumed the networks would operate with two-dimensional input data (usually images). Each layer of a conventional CNN transforms one volume of activations into another. Figure 2.15 depicts a CNN architecture that recognizes items in an image [89]. Figure (2.11) illustrates how CNN works with images in computer vision field.

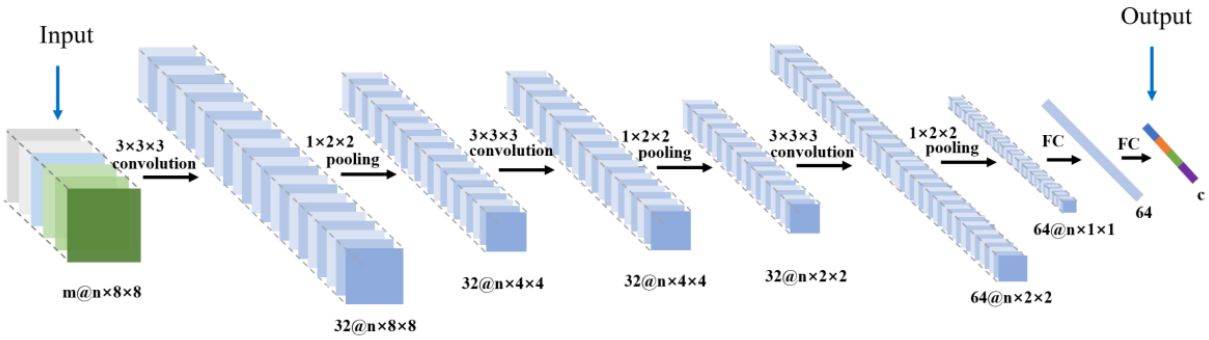


Figure (2.11) illustrates how CNN works with images in computer vision field [90].

a) Convolution Layer

Focus on discrete 2D-convolution in the computer vision field, as it is the most used kind of convolution in digital image processing. The convolution technique determines the value of an image's pixel's intensity. To measure the pixel that corresponds [69, 63].

The input's spatial dimensions (height and width) as well as the depth, the depth is the third dimension of a volume of activation. [64] Neurons within a given layer will only connect to a limited portion of the layer below. A convolution layer is an essential component of CNN architecture that performs feature extraction. It often consists of a combination of nonlinear and linear processes, including convolution and activation functions.

Convolutional is a mathematical procedure for combining two groups of information. It uses convolution filters to extract features from the input image and learn these features using input data arrays, while generating feature maps that enable the growth of the spatial relationship between each feature in the image.

By sliding the filter/kernel across an input image, the convolution process was completed. At each spatial point, the element-wise array is multiplied and the resulting value is computed. This result will then be added to the feature map. The 2D convolution layer extracts features from the input image and generates feature maps using a 2D kernel (filter) [91].

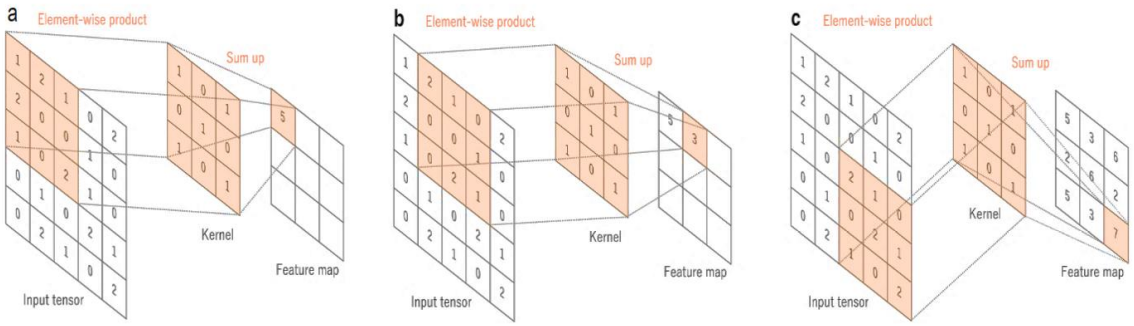


Figure 2.12 Convolution operation perfuming [92]

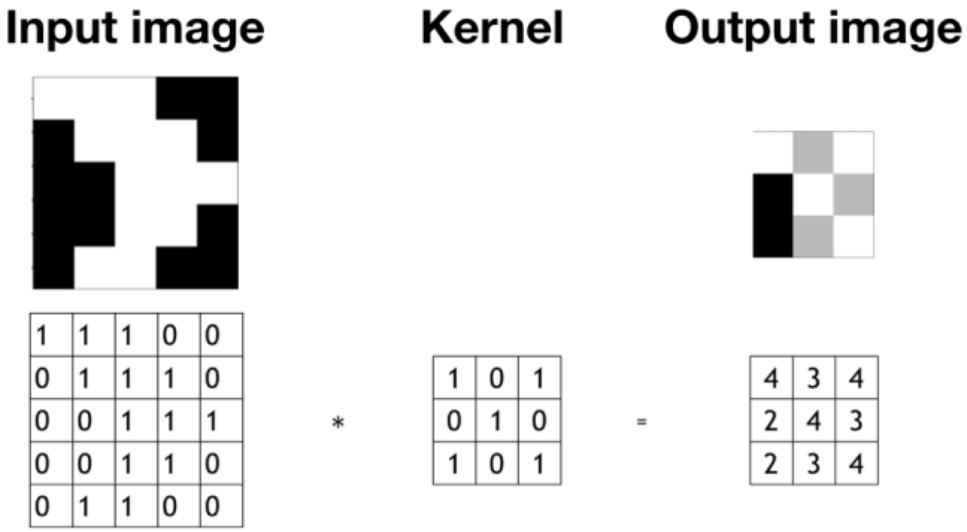


Figure (2.13) Convolution kernal matrix with input and output image matrix [93].

The output size of the convolution layer is controlled by a number of hyperparameters in the CNN design. The following are some important CNN hyperparameters:

- Number of Filters: Different reasonable number of filters can be used with different sizes.
- Filter size: The squared filter or kernel size must be less than the input image.
- Padding: is an effective method for regulating the dimensionality of output volumes by filling at the input boundary.
- Stride: the number of cells (pixels) to move the kernel simultaneously through or down the input map [74].

b) Pooling Layer

The pooling layer transforms the Representation of the common feature into one that is more usable by retaining essential information and removing irrelevant information; then, in the pooling layer, the resolution of the feature maps is decreased in addition to the inputs' deformation stability being enhanced

[94]. The two most common ways to pooling are maximal pooling and average pooling. Similar to how the Max Pooling algorithm generates the maximum value of the input map section covered by the Kernel, the average pooling algorithm generates the mean value of the image portion covered by the Kernel [75].

Global Max Pooling and Global Average Pooling are further forms of pooling layer. Where the pool size is equal to the size of the input map, and the maximum value of the whole input map is calculated as the output in Global Max Pooling, whereas the average of the entire input map values is computed as the output in Global Average Pooling [76]. As seen in Figure (2.19) average and max pooling.

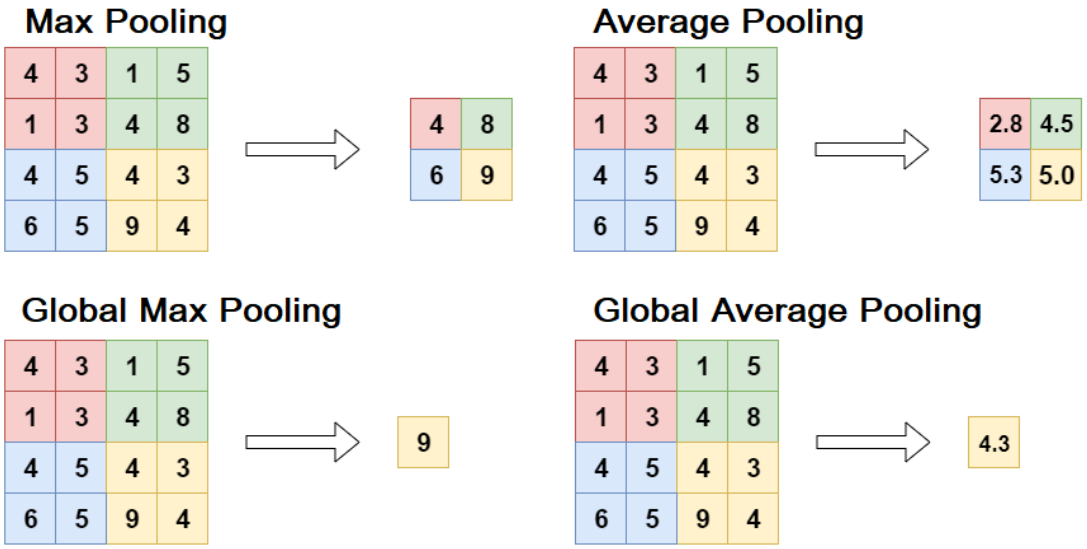


Figure (2.14) Types of Pooling layer [95].

2.7 Transfer Learning (TL)

Transfer learning is one of the most common and widely used strategies in computer vision today. By using transfer learning rather than one of the other methods, we are able to attain a high level of classification performance with datasets that are of a relatively small size [28].

In the discipline of deep learning, transfer learning is an approach that is used to increase the efficiency of current problems on a specific domain by

transferring the information held in other but related source fields [29]. That is, it means not learning the present model from scratch, which is different from the strategies typically used in machine learning. These days, the powers on a machine are appropriate and are obtainable in plenty, but in order to train a CNN model from scratch, we need a significant amount of labeled time, data, and GPU resources speed [30].

It was challenging to train the CNN model from start because to the limited availability of data because the process required a high-speed GPU computer and computationally heavy jobs, and it may take many days or even weeks [31]. Transfer learning is a method that stores the bias and weights values of the early layers of a neural network, which are in charge of feature extraction, in order to anticipate the behavior of subsequent layers deeper inside the network [32].

Transfer learning is a technique for predicting later layers in the deep of a network by saving the bias and weights values of the initial layers that are responsible for feature extraction. Furthermore, transfer learning is employed to overcome the problem of model overfitting that occurs when limited data is used. Transfer learning is applied to a variety of real-world applications, including image categorization [34, 35], name-entity recognition problems [36], NLP problems [37].

2.7.1 VGG16 Architecture

ConvNet is a convolutional neural network, a type of artificial neural network. A convolutional neural network consists of an input layer, an output layer, and multiple hidden layers in between [96]. CNN models such as VGG16 are considered to be among the most effective computer vision models to date. Using an architecture with very small (3x3) convolution filters, the creators of this model analyzed the networks and made them more robust. This was a

significant improvement over previous efforts. VGG16 is an object detection and classification algorithm that can classify 1000 images into 1000 distinct categories with a 92.7% degree of accuracy. It is a popular algorithm for classifying images, and it is easy to use with transfer learning. Figure (2.15) illustrates the architecture of this network [97].

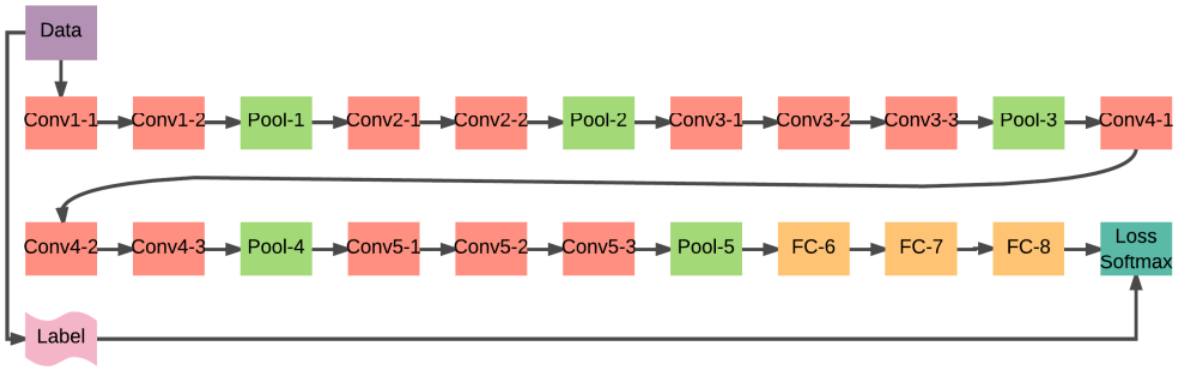


Figure (2.15) VGG16 architecture [98]

The sixteen in VGG16 represents the sixteen weighted layers. VGG16 contains a total of 21 layers, consisting of 13 convolutional layers, 5 Max Pooling layers, and 3 Dense layers. However, there are only 16 weight or learnable parameter layers [99].

VGG16 is compatible with 224x224 tensors with three RGB channels. Instead of having a large number of hyper-parameters, VGG16 focuses on having convolution layers of 3x3 filter with stride 1 and always uses the same padding and maxpool layers of 2x2 filter with stride 2 [98].

The convolution and maximum pool layers are consistently combined in the same manner. Conversion Layer 1 has 64 filters, Conversion Layer 2 has 128 filters, Conversion Layer 3 has 256 filters, Conversion Layers 4 and 5 each have 512 filters. Three Fully-Connected (FC) layers follow a stack of convolutional

layers. The first two have 4096 channels each, while the third utilizes 1000-way classification and therefore has 1000 channels (one for each class). The soft-max is the final layer [99].

2.7.2 Resnet50 Architecture

Shaoqing Ren, Kaiming He, Jian Sun, and Xiangyu Zhang introduced Residual Network (ResNet), a well-known deep learning model, in their article. ResNet is one of the most popular and successful deep learning models to date. These Residual blocks, which are also the basis of the ResNet model, have solved the problem of training very deep neural networks. Figure (2.16) represents the skip connection of this network [100].

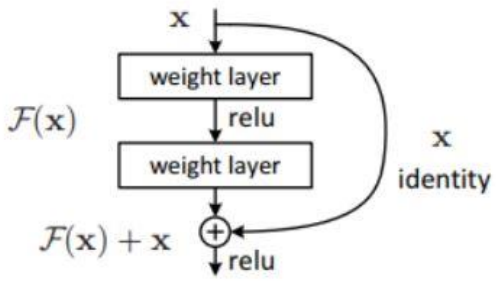


Figure 2.16 skip connection of Resnet50 [101]

The problem of training, it was very deep networks has been solved by these Residual blocks, and the ResNet model is made up of these blocks.

The first thing we can see in the image above is that there is a direct connection that skips over some of the model's layers. This connection is called the "skip connection," and it is the main cause of residual blocks. The output is different because of this skip connection. Without the skip connection, the input

value X is multiplied by the weights of the layer, and a bias term is added on top of that. Then there is the function that turns it on, f(x) and we get the output as H(x) [102].

$$H(x) = f(wx + b) \quad or \quad H(x) = f(x) \qquad \ldots\ldots \text{ (2.18), [102].}$$

With the advent of a new skip connection technology, the output is H(x) which is changed to

$$H(x) = f(x) + x \qquad \ldots\ldots\ldots\ldots\ldots\ldots \text{ (2.19), [102].}$$

This "skip connections" method in ResNet finds a way to deal with gradients that is vanishing. In deep CNNs by letting the gradient flow through an alternate, shorter path. Also, the skip connection is helpful because if a layer hurts the performance of the architecture, regularization will skip it. The architecture of this network illustrated in figure (2.16) [103].

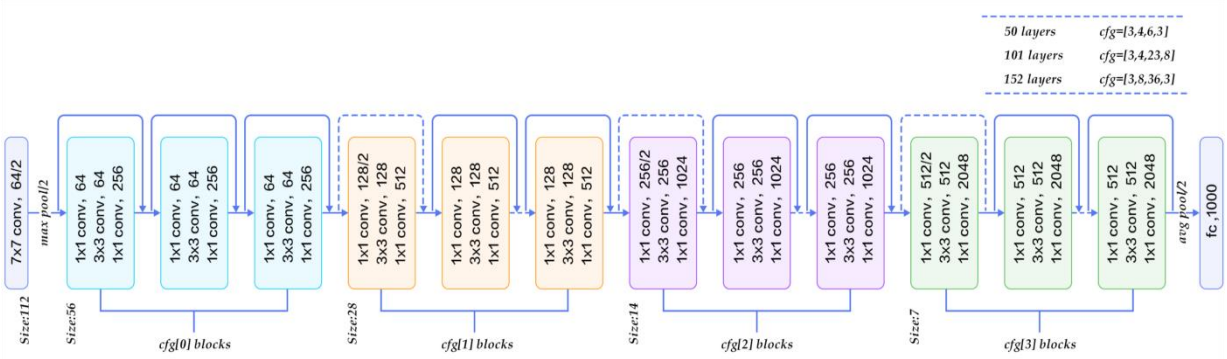


Figure (2.17) represent the architecture of Resnet50 [104]

2.8 Gradient Class Activation Map Technique (GRAD-CAM)

A lot of effort was being done to make deep learning more comprehensible and explainable. Making the deep learning model more interpretive in various deep learning applications connected to medical imaging is critical. The Gradient Weighted Class Activation Mapping (GRAD-CAM) approach, developed. It was

gives an explainable image of deep learning models. GRAD-CAM gives a visual explanation for any deeply linked Neural Network, which aids in learning more about the model while performing detection or prediction tasks. This technique applied and proved on dataset that contains mask to focus on the RoI area in images. These masks in dataset can be generated by training some deep Learning network such as UNET.

GRAD-CAM whose architecture is depicted in Fig. (2.18), employs the gradients of any target perception in a classification network's final convolutional layer to generate a crude heatmap emphasizing the crucial regions of the image for concept prediction. This model can be used for many different CNN model families. GRAD-CAM employs the gradients of any target topic, which flow into the final convolutional layer to provide a coarse heatmap emphasizing the concept-predicting areas of the image. Using GRAD-CAM, helps to visually evaluate where the network is looking, confirming that it is looking at the proper class pattern and activating around them [105].

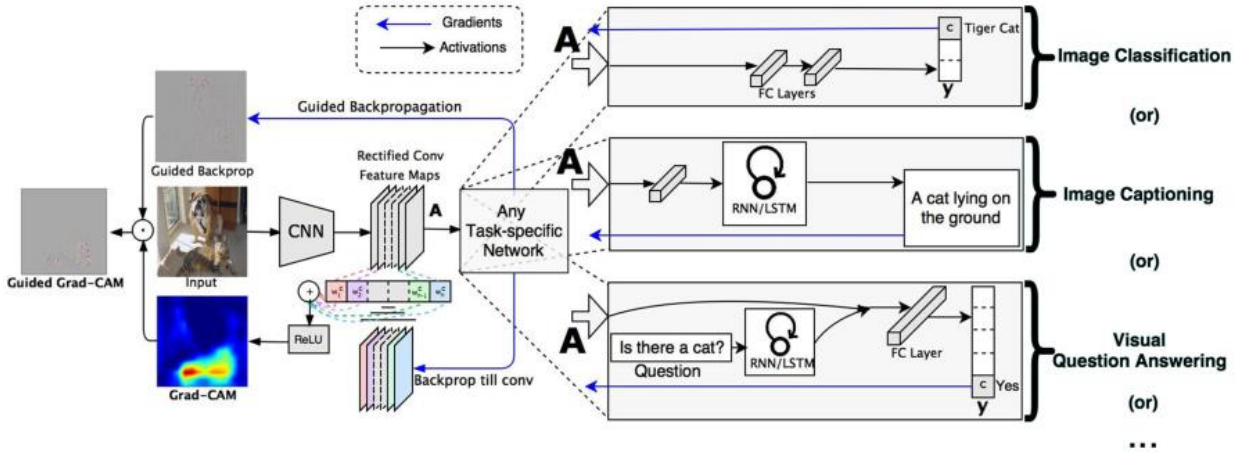


Figure (2.18) Grad-CAM in brief: given an image and a class of importance. All of the gradients are set to 0 except for the one we want (tiger cat), which is set to 1 [105].

Class Activation Model (CAM), on the other hand, is a good way to understand how CNN works and build confidence in the applications that have

been made, but it has some limitations. CAM has some flaws, like the fact that it needs feature maps to work that precede the SoftMax or sigmoid layers. Therefore, it may be used with CNN designs of a certain kind, specifically those that carry out global average pooling across convolutional maps just before to prediction, another limitation to CAM that focus only one time of the object [106].

To acquire the discriminatory localization map based on class, GRAD-CAM Lc GRAD-CAM ∈ R^{uv} of width u and height v for any class c, first, figure out the score for class c, y^c with respect to the feature map activations A^k of a convolutional layer. This is done before the softmax, $\frac{\partial y^c}{\partial A^k}$. The neuron significance weights are obtained by global-average-pooling these flowing-back gradients over the width and height dimensions (indexed by i and j, respectively).α_k^c:

$$\alpha_k^c = \frac{1}{z}\sum_i \quad \sum_j \frac{\partial y^c}{\partial {A^k}_{ij}} \qquad \ldots\ldots\ldots\ldots\ldots\text{(2.20), [120].}$$

When backpropagation gradients with respect to activations, the exact computation c of k amounts to successive matrix products of the weight matrices and the gradient with respect to activation functions until the final convolution layer where gradients are sent; thus, this weight α_k^c is a partial linearization of the deep network after A. It demonstrates the "importance" of feature map k for a target class c, followed by a weighted combination of forward activation maps and a ReLU to produce a classification model:

$$L^c_{GRAD-CAM} = ReLU\Big(\ \sum \alpha_k^c\ A^k\Big) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\text{(2.21), [106].}$$

This gives in a coarse heatmap with the same dimensions as the convolutional featuremaps. Applying a ReLU to the linear combination of maps because the features that have a positive effect on the class of interest, were the only ones of

interest. Pixels whose intensity must be increased to increase yc. Most likely, negative pixels belong to other image types. As anticipated, without this ReLU, heatmap occasionally highlight more than the appropriate class and perform worse when localized [105]. Grad-CAM visualizations of Covid-19 are illustrated in Figure (2.19).

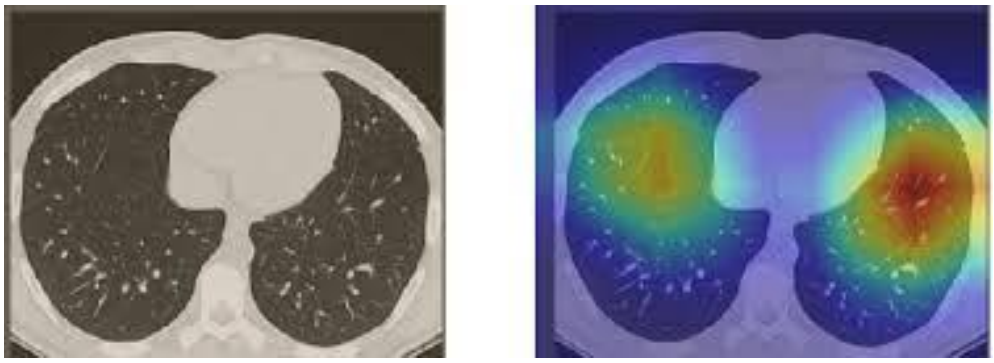


Figure (2.19) Explanation of GRAD-CAM to visualize infection area [105].

2.9 Grad-CAM as a Generalization of CAM

Grad-CAM does not necessitate a specific CNN architecture. Grad-CAM is a more general edition of CAM, which stands for class activation mapping. CAM is a method that does needed the application of a certain architecture.

CAM necessitates an architecture that employs global average pooling on the predictions that are generated by the final convolutional feature maps, followed by a single fully connected layer:

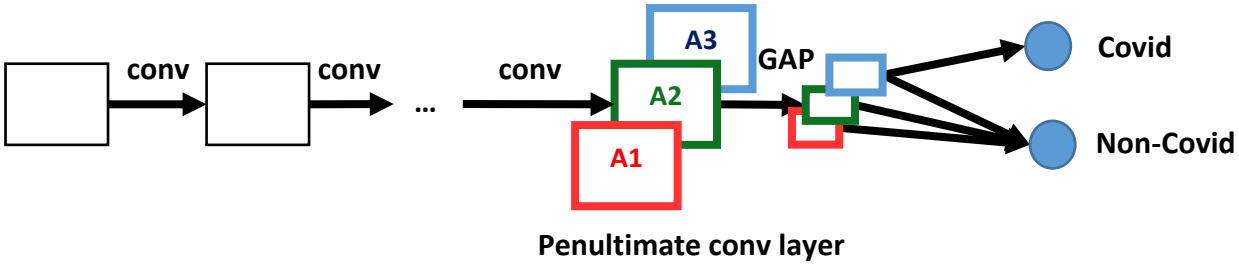


Figure 1 Figure (2.20) Feature map with global average pooling layer of CNN [106].

$$CAM^{covid} = W_1A^1 + W_2A^2 + W_3A^3, i.e. \sum_k W_k^{covid}.A^k \ldots(2.22),$$ [106].

Where W is the average weights in the featuremap.

2.10 Image Segmentation

Segmentation of an image is the process of dividing it into a number of distinct regions that can be compared to one another and are meaningful in some way. The pixels in each segmented area must share the same set of qualities or attributes. Textural features, spectral values, contrast levels, and grayscale levels are some examples of the sets of qualities that can be found in an image. The process of image segmentation results in the creation of several similar regions, each of which is assigned a unique label. These are the regions that have been designated as the Regions of Interest (RoI) for this image [107]. The strategies for image segmentation are displayed in figure (2.20).

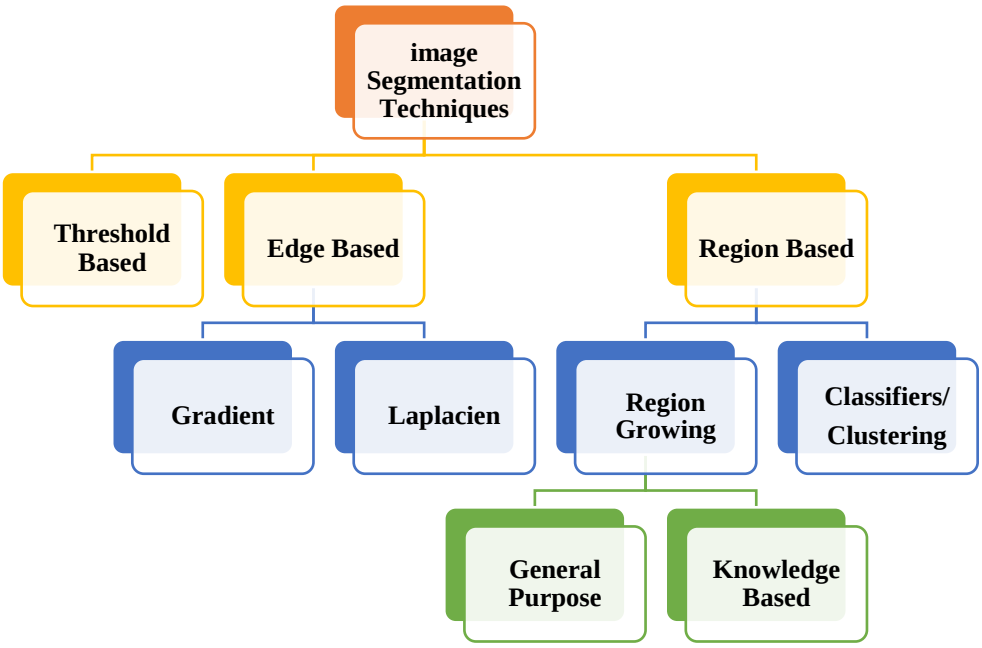


Figure (2.21) The Image Segmentation Techniques. [108]

Techniques for image segmentation are dependent on two basic characteristics: discontinuity in the gray level values and similarity among the pixels.

The first form segments the image based on sharp variations in the level of gray, whereas the second type segments the image based on the similarity of the pixels contained inside the region.

The following are some of the more common methods of image segmentation:

1. Histogram-established thresholding.
2. Region growing.
3. Region merging and splitting.
4. Clustering/Classification.
5. Graph theory approach.
6. Rule-based or knowledge-driven approach [109].

Another method of image segmentation that involves quantizing the colors present in the image. There was a delay in the saving of the information on the colors in the image pixels. This information is stored in discrete chunks of data referred to as palettes. A palette is just an array of colored components, with each component expressing a color and being indexed according to its position in the palette. This method is utilized for indexing images with an eight-bit indexing images [110]**.**

In order to solve the problem of multi-level thresholding (MTH) segmentation, meta-heuristic algorithms like Cuckoo Search, Bat, Artificial Bee Colony, and Social-Spider optimization are also utilized [111]**.** It is more useful to apply ANDing method to extract ROI of image by utilize mask of this area,

this method multiply each pixel of image with corresponding pixel of mask, this lead to extract ROI and leave other pixels with zero value (black) [112]. The graphic demonstrates three distinct approaches to threshold segmentation techniques (2.21)

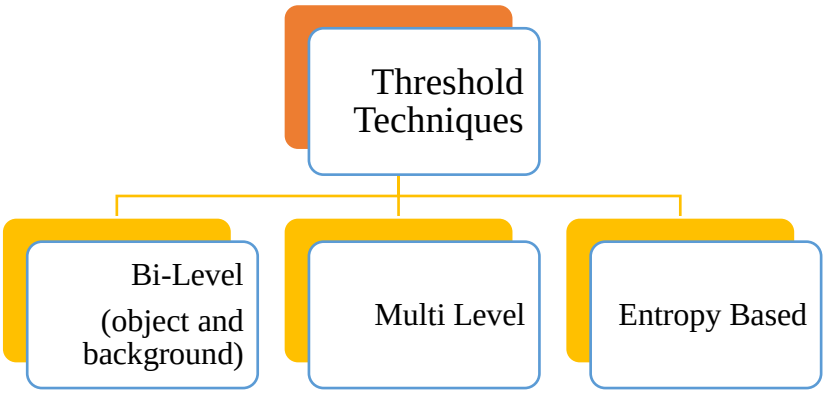


Figure 2.22 Thresholding Techniques Types. [112]

2.11 Evaluation Measures

When constructing a machine learning model, the evaluation of performance and efficiency is crucial. For the machine learning model to be trustworthy, an assessment method must be selected that is proportional with the model's work. Frequently, while assessing machine learning models, many scales are employed to guarantee accurate evaluation. In machine learning, there are three primary types of evaluation measures: those used to assess classification and clustering tasks. Classification tasks may be evaluated using a variety of metrics, including Accuracy, Confusion Matrix, Recall, Precision, and F1 Score [113].

1) Accuracy Measure

As demonstrated in equation (2.22), Accuracy is the proportion of average correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(2.22), [113]}$$

2) Confusion Matrix (CM)

CM is one of the most significant tools for providing a full explanation of the classification model's performance. The confusion matrix for a binary classification system is depicted in Figure (2.22).

	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Figure 2.23 confusion matrix [113]

Each prediction will fall into one of four categories:

• True positive (TP): a correctly predicted positive outcome.

• False positive (FP): a positive prediction that is inaccurate.

• True negative (TN): the correct forecast of a negative outcome.

• False negative (FN): a negative forecast that is incorrect.

The matrix accuracy can be obtained by averaging the main diagonal values using the equation (2.23) [113].

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad \ldots\ldots\ldots\ldots \text{(2.23), [113].}$$

3) Recall

Recall is the proportion of true positives to the sum of true positive and false negative; it is used to quantify the classifier's capacity to recognize all positive cases; and it is one of the most essential metrics utilized with models including unbalanced datasets. This metric is calculated using the following formula: (2.24).

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(2.24), [113].}$$

4) Precision

Precision is defined as the capacity of a classifier to not label a negative instance as positive and is described by the ratio of right positive cases to the total of right and incorrect positive. A measure of precision that can be determined using the equation shown below (2.25) [113].

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \quad \ldots\ldots\ldots\ldots\ldots \text{(2.25), [113].}$$

5) F1-score

The F-score is represented as the arithmetic mean of recall and precision. The F1 seeks to strike a balance between recall and precision, and is used to quantify a test's accuracy, which determines how many instances it correctly classifies, and robustness by preventing the model from ignoring a substantial number of cases. F1 Score ranges between 0 and 1, with the bigger value indicating superior performance. The following equation (2.26) reflects the F-score measure [113].

$$\mathbf{F_1\ score} = \frac{\mathbf{2\times Precision\times Recall}}{\mathbf{Precision+Recall}} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(2.26), [113].}$$

CHAPER THREE
PROPOSED SYSTEM DESGIN

3.1 Introduction

In this chapter, the practical stages of the proposed system are described in a series of procedures. The proposed system illustrates the processes used for image converting from nii files to png format, segmentation, normalization, resizing images, training, testing the model to classify dataset images into Covid and Non-Covid images and infection localization. In addition to discussing the rationale behind selecting the dataset that will be implemented in the system, this chapter focus into the fundamental ideas and techniques that are associated with the various stages of the model.

3.2 Used dataset

In this thesis two datasets had been used, the first dataset with name "Large COVID-19 CT scan slice dataset" [122]. This dataset was combined from seven common datasets, these datasets have been previously used and improved their efficiency in deep learning to detect COVID-19. Therefore, the variety of these CT images from different devices improves the generalization ability against the problem of deep learning by training with different images from different resources. In total, this dataset contains 6,893 normal case from 600 patients, 7,593 COVID-19 case from 465 patients. Figure (3.1) illustrates samples from this dataset.

The second dataset with name "COVID-19 CT Lung and Infection Segmentation Dataset" [123]. This dataset contains four groups of images for Left lung, right lung.

Lung mask has been applied to extract the ROI (the lungs) before the training operation as a preprocessing step. The infections are labeled by two

radiologists and confirmed by a radiologist with extensive experience [114]. Figure (3.2) illustrates samples from this dataset.

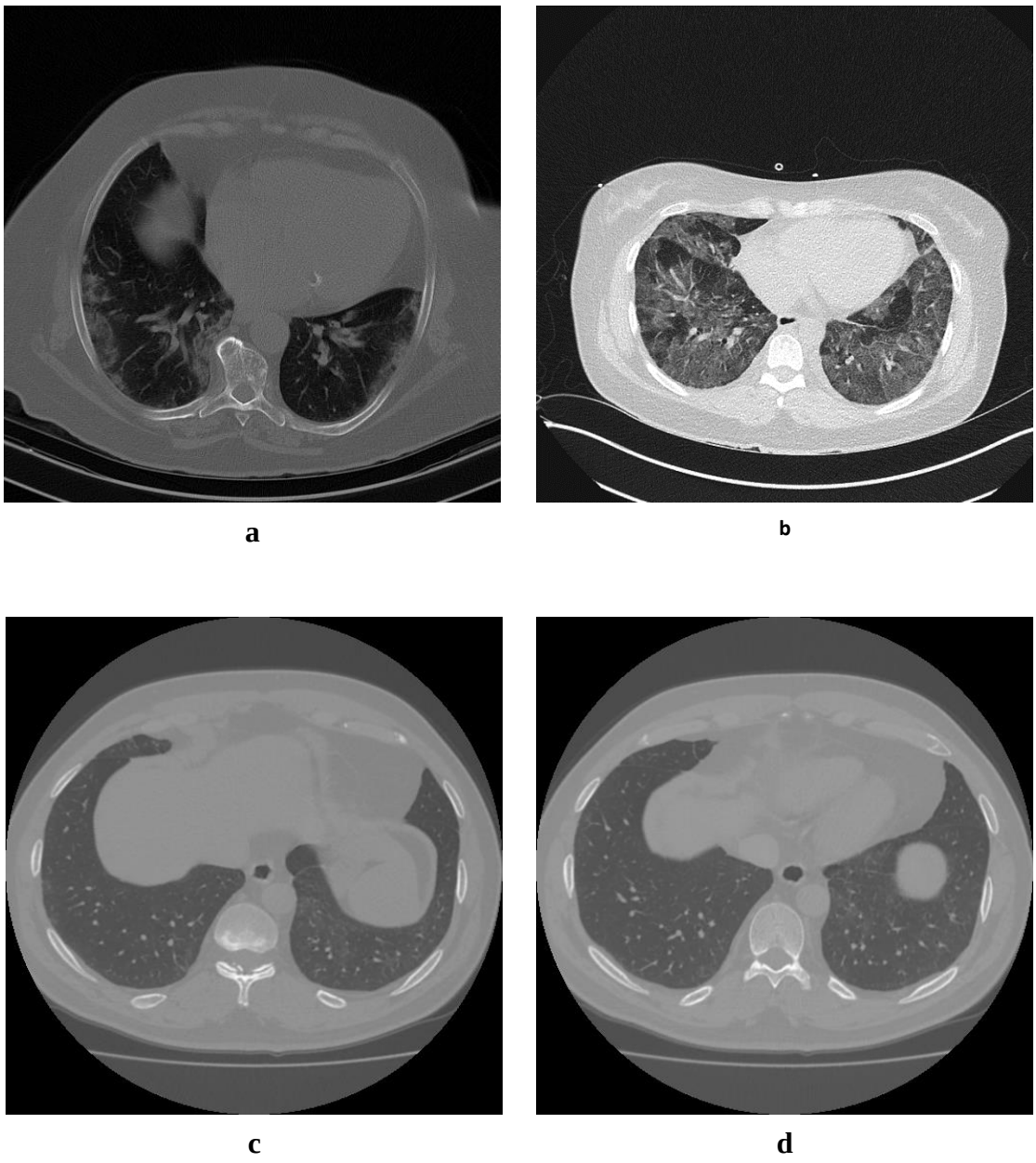
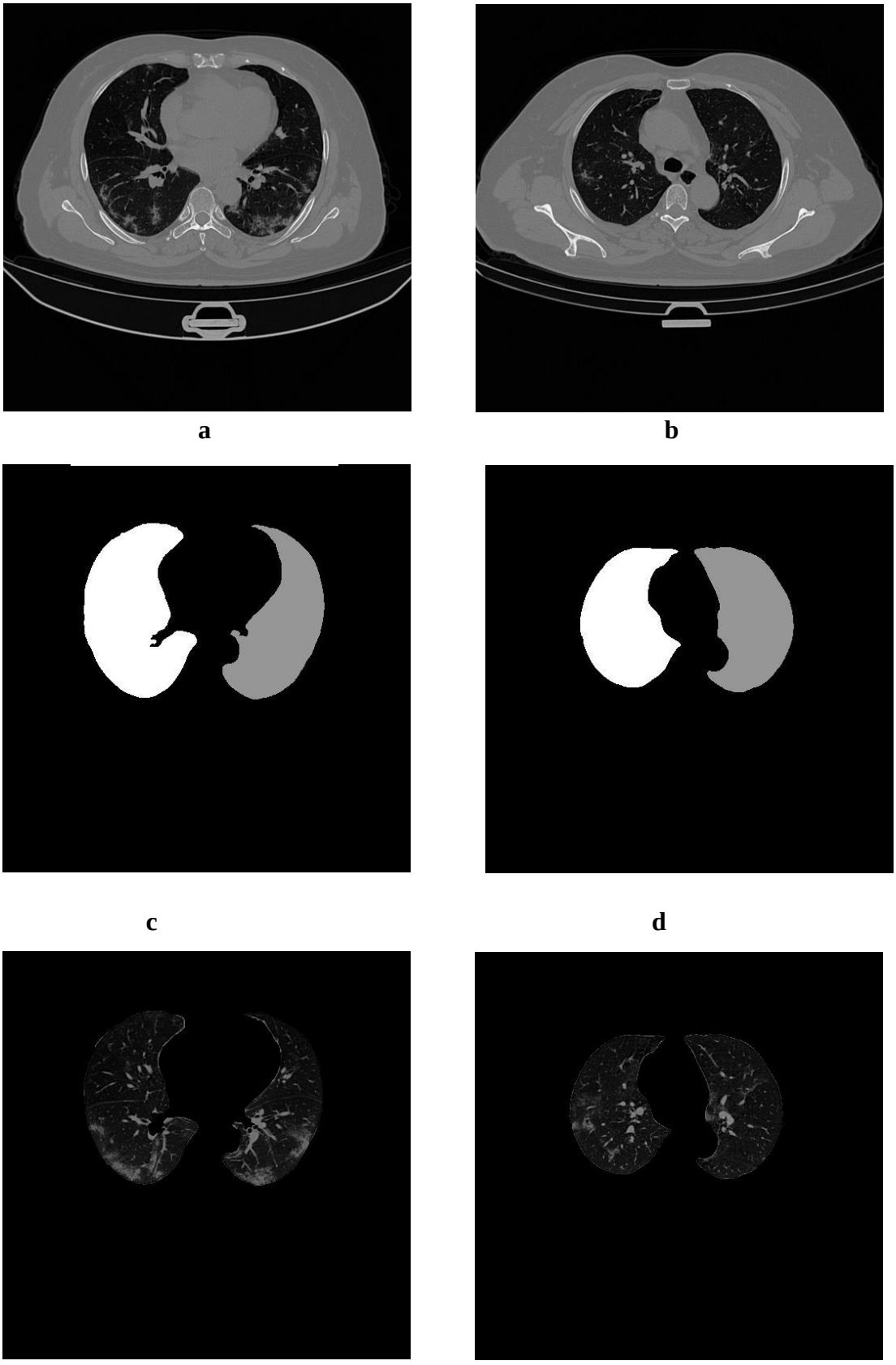


Figure (3.1) Samples from first dataset Covid-19 images
(a,b) Covid image, (c,d) Non Covid image



a **b**

c **d**

e **f**

Figure (3.2) Samples from second dataset Covid-19 images (a,b) CT image,(c,d) mask image, (e,f) lung image (ROI).

3.3 Proposed System

The proposed system is split into three phases, each phase includes some of stages. The proposed system phases explained in figure (3.3).

The first phase consists of data preprocessing while the second phase includes modified VGG16 CNN model to obtain features and make decision (Covid, Non-Covid). Finally, third phase that localizes infection area according to feature map from the last convolution layer in modified VGG16 model in phase two.

3.3.1 Data Preprocessing

This phase regarded with image pre-processing as explained as follows:

3.3.1.1 Converting Image

Medical image such as Nifti (Neuroimaging Informatics Technology Initiative) file with nii extension consists of approximately 100-300 number of slices. Nifti file is a commonly used file format in medical research and related fields for storing medical imaging data, a NIfTI file conceptually incorporates multidimensional numerical data with additional metadata that describes the image's real-space resolution and physical orientation. These files have to convert to another format such as png file format. The reason of converting images is

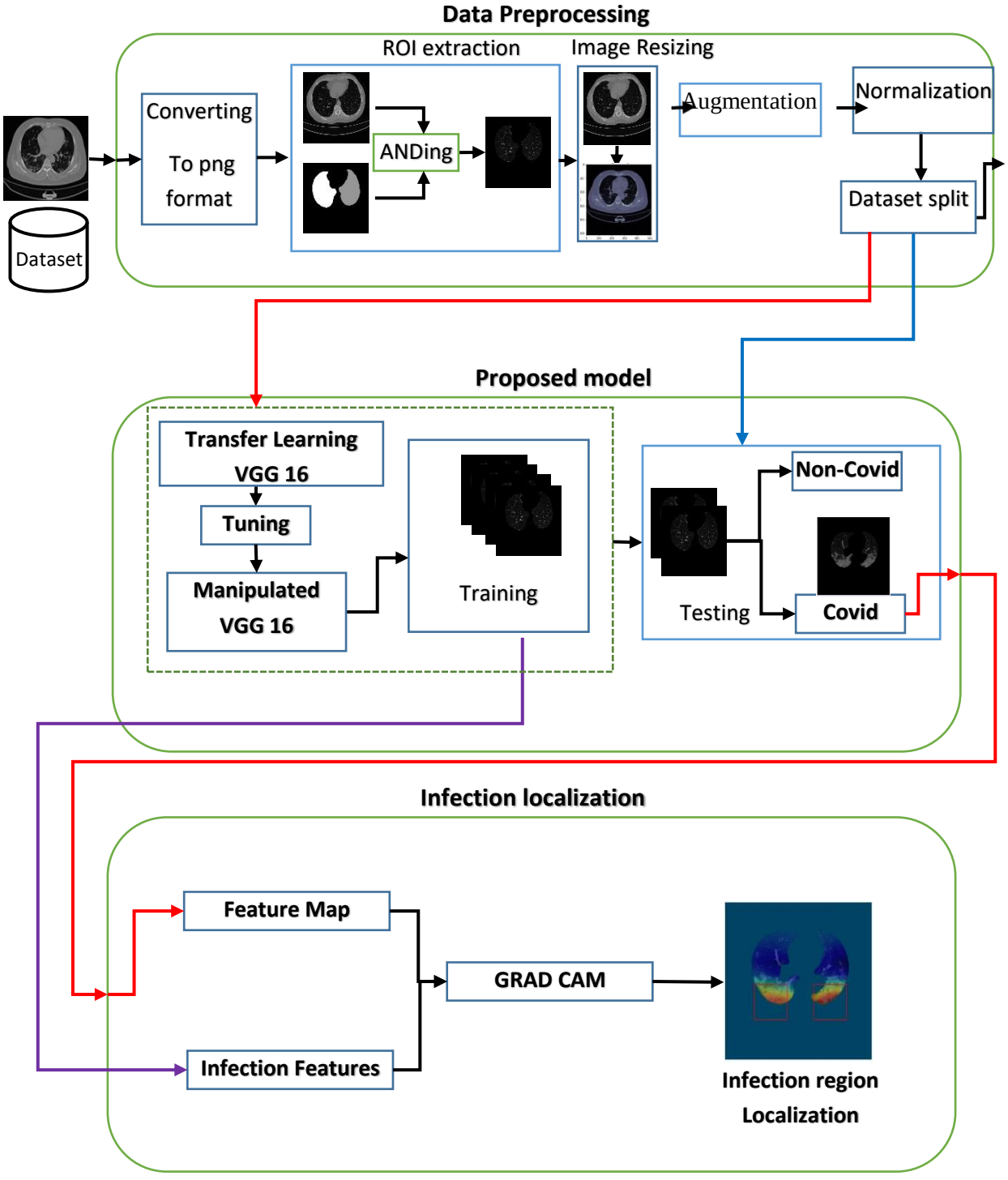


Figure (3.3) Proposed System Block Diagram.

that the result images small in size and easy to implement. Algorithm (3.1) illustrates convert Nifti file to set of png images.

Algorithm (3.1) nii to png convertion

Algorithm name: nii to png convertion

Input: Nifti file.

Output: png images.

Begin

1. Load Nifti file.
2. Define Z as set of slices. // empty array
3. Getting the data of images from Nifti slices to Z. //slices from file to Z
4. Create name for each slice.
5. For each slice in Z
6. Getting image shape from the slice. // transform slice to png image.
7. Apply PNG format to the image shape. // write the header of PNG image
8. store the result with the path and file name.
9. End for

3.3.1.2 Region Of Interest Extraction

Region of interest (ROI) is a neuropsychological term that refers to the parts of an image that are of interest to humans. Generally, regions of interest in images represent the image sections that are significant and relevant. For areas of interest in medical images, for instance, are the focal regions,

including Covid-19 infection in lung. The use of ROIs helps optimize operations. It can lower calculation; the time, Moreover, efficiency of processing because irrelevant data are not considered, performance can be enhanced account. There are a number of strategies for extracting regions of interest in images of nature that mentioned in section 2.9.

CNN models in feature extraction stage will extract the features from the whole image whatever features are useful or not and the extracted features out of RoI. Image segmentation process used to extract RoI from the whole image to avoid training outside RoI feature.

ANDing segmentation technique has been used to extract lung region from the raw image and store them in a new image with black background and lung as foreground object. In this process each pixel in image will multiplied (ANDing) with equivalent pixel in the mask of image in dataset that produce an image contain only the pixel representing the lung in image and the rest of image pixel set to zero value. Algorithm (3.2) illustrates ANDing segmentation algorithm.

3.3.1.3 Image Resizing

Scaling the dimensions of cropped images in the images is a massive step toward presenting generality to samples and trying to make them suitable for the deep learning prediction model.

As the name suggests, "image scaling" refers to the process of recreating an image from one grid to another, either by increasing or decreasing the number of pixels in a sample of an image, "Image scaling" is the term for this technique. For example, when an image was resized, it uses an image scaling technique called interpolation, which estimates missing values by interpolating

known data (the values at adjacent pixels). Other image scaling methods do not produce as good of results as this one.

The bi-cubic interpolation approach was applied to the pixel's sixteen near neighbors to get the appropriate intensity value for that position in the image.

The final images produced from raw images vary in size depending on the original raw image size and scale during image capture resulted in this difference. The 224 X 224 image size can be achieved using a variety of resizing approaches.

Algorithm (3.2) Region of Interest Extraction Algorithm

Algorithm name: Region of Interest Extraction process

Input: raw image, lung mask. // images

Output: lung image with black background. // image

Begin

1. Load image (raw image).
2. Load image (lung mask).
3. Construct lung image with same size of raw image. // create blank image
4. For i=1 to lung image width
5. For j=1 to lung image height
6. Lung image (i,j) = raw image(i,j) AND lung mask (i,j)
7. End for j
8. End for i
9. Return lung image

End

3.3.1.4 Image Normalization

The Image data is made up of pixel values, which are whole numbers from 0 to 255 that show how bright each pixel is. Models of neural networks deal with weight values that are not very high when they process inputs. Large integer values can either throw off the learning process or make it go much more slowly.

Normalization is a process that changes the intensity range of pixel values so that the range of possible values for each pixel value is between 0 and 1. No matter what range of pixel values are in the image, the values of all pixels are made the same by dividing each pixel's value by the largest pixel value in the image, which is 255. This was done repeatedly for each channel. During the normalization process, a grayscale image with intensity values that fall between (Min, Max) looks like this:

$I: \{X \subseteq \mathbb{R}n\} \longrightarrow \{min, \dots, max\}$ generates a new image with intensity values that are within the range of (newMin, newMax) as:

$$IN: \{X \subseteq \mathbb{R}n\} \longrightarrow \{0, \dots, 1\}$$

$$X_{norm} = \frac{X - min\,(X)}{max\,(X) - min(X)} \quad \dots\dots\dots\dots\dots\dots \text{(2.26), [43].}$$

3.3.1.5 Dataset Augmentation

In this step the dataset was augmented to increase the number of images and give the model more different orientation of images. The augmentation processes that used are (shear, height shift, width shift, and rotation). This step increase the dataset from 2760 images into 9640 images.

3.3.1.6 Dataset Splitting

When data is split into two or more smaller groups, this is called "splitting the data" One part of a two-part split is often used to test or evaluate

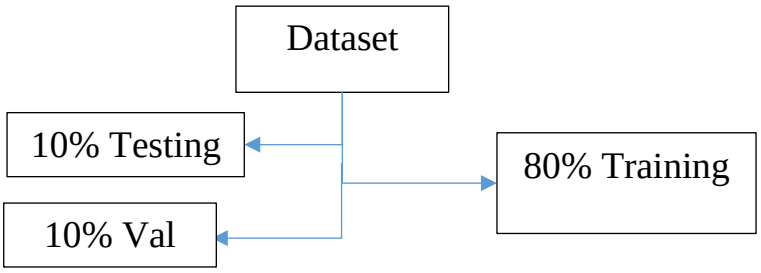


Figure (3.4) Dataset division into train, val, and test set.

the data, while the other parts is used to train the model. Data splitting is a crucial component in data science, especially when building models from scratch. Eighty percent of the images are used for training, while twenty percent are used for validation and testing the neural network models to classify the images as Covid-19 or non-Covid as shown in Figure (3.4).

3.3.2 Model Constructing

This phase consists of choosing the valuable architecture of transfer learning. Many TL networks had been trained and evaluated to construct a model, this phase made up of steps as follow:

3.3.2.1 Features Extracting

Features extracting consist of three steps, each of which is broken down into a substep that has been represented by group of convolution layers, a nonlinear layer, and one max pooling layer. The convolution layer of a CNN is equivalent with local filters that were applied to the input data. The training process has been used to define the size of the filters (3 × 3) and the filter kernel coefficient. The first convolution layer has the ability to extract a set of

primitive patterns, which are low-level characteristics in the input images such as edges and lines. These patterns can be represented by the layer. Using a combination of these fundamental features, such as corners, the second convolution layer is able to identify patterns inside patterns.

The deeper convolution layer is responsible for extracting higher-level features by integrating the secondary features obtained from the previous layer with the primary features detected in the prior layer's patterns, and so on.

The term "Rectified Linear Units" refers to a non-linear "trigger" function that is utilized to differentiate signal from usable characteristics on each hidden layers (ReLUs). The pooling and subsampling layer makes the features resistant to noise and blurring. To do this, the layer decreases the features' resolution by applying the max pooling function. In this, thesis four networks (VGG16, Inception V3, Xception, Resnet50) were trained to choice best architecture.

3.3.2.2 Tuned VGG16

The whole structure of VGG16 that was used in this study contains the following layers:

1. The initial and second convolutional layers are comprised of 64 kernel filters of size 3x3 that were used to extract features. In the first and second convolutional layers. The output was send to the subsequent max-pooling layer.
2. The third and fourth convolutional layers consist of 112x112x128 3x3 kernel filters. After these two layers, there is a max pooling layer. The output dimensions are now 112x112x64.
3. Convolutional layers five, six, and seven with kernel filters of size 3x3 and 56x56x256. A max pooling layer follows these layers. The output image dimensions were 28x28x256.

4. The eighth, ninth, and tenth Convolutional layers are constructed of 3x3 28x28x512 kernel filters. After reducing the output to 14x14x512 dimensions, a max-pooling layer was applied.
5. The eleventh, twelfth, and thirteenth convolutional layers each have a 14x14x512 kernel of size 3x3.
6. Global average pooling layer is the next layer.
7. Batch_normalization layer.
8. The last layer consists of a dense layer, which makes the prediction.

Figure (3.3) illustrates the tuned VGG16 architecture according to origin VGG16 model.

A. Convolution Process

It is possible to represent as an algorithm the convolution process that is utilized by the networks. Imagine that the input image has indices of i and j, and that it has received k weights, Kernel W of size k1*k1 of m,n indices, and it moves with strides S1,S2. The convolution process requires four nested

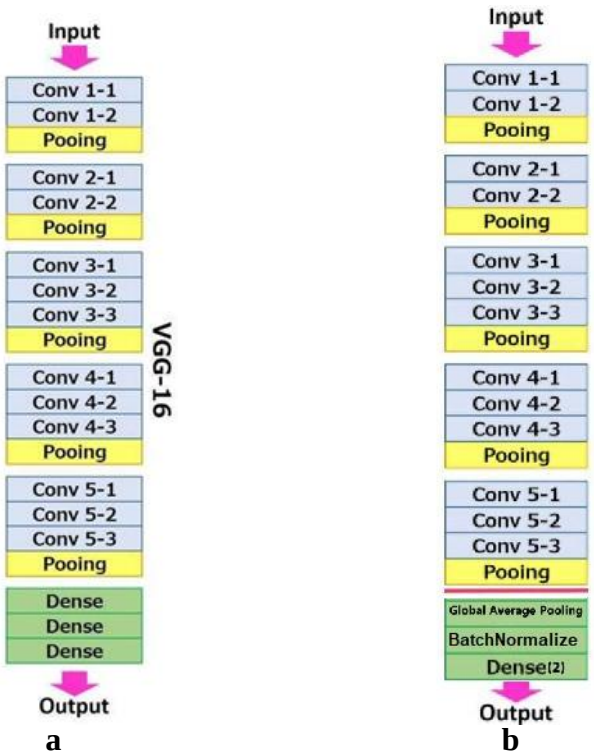


Figure (3.5) VGG16 a. standared VGG 16, b.moified VGG16 [97]

loops for 2D convolution, as indicated in algorithm (3.3), the architecture of VGG16 and modified VGG16 illustrated in figure (3.5).

B. Max Pooling Process

The pooling process is a sample-based procedure. When learning parameters are reduced, both the input dimensions between layers and computing costs are reduced as a result of the pooling layer's goal. The robust features were selected by the MaxPooling level, which is in charge of making the characteristics noise-resistant. The MaxPooling operation explained in Figure (3.6) Algorithm (3.4).

Algorithm (3.3) Convolution Algorithm

Algorithm name: convolution process

Input: Image
Output: Feature map of the image

```
        Row: image row
        Col: image column
        k: number of kernels
        S: kernel size, W =[ ] // W is weights filters or kernel
   For k from 1 to k
       Initialize kernel (W) randomly
       Initialize bias (b) ← 1 // b is a constant which is b equal to 1
       For i ← 1 to Row
          For j ← 1 to Col
            Net=0
            For m from i-d to i+d // rows and cols. of kernel, d kernel size
              For n from j-d to j+d // d=1 equal to L*L kernel (W) of size
3*3
                net ← net +[ I(m,n) * W(m,n) + b]
              End n
            End m
            F(net) ← Max(net, 0) // Relu activation function
            Feature-map[i,j,k] = f(net) // k feature map resulted that equal to
the number of filters
            Batch normalize the Feature-map[i,j,k] as equation (2.8 )
            End j
          End i
       End k
```

C. Global Max Pooling process

CNN is switching to using global average pooling in favor of the more traditional fully connected layers. In the final convolution layer, the objective is to generate one featuremap for each category matching the classification task. Averaging each feature map and feeding the resulting vector directly into a softmax layer save time over just putting additional fully connected layers on top of each one. Because it enforces correspondences between feature maps and categories, global average pooling is more convenient than fully connected layers for the convolution structure.

In addition, because the global average pooling contains no parameters to overfitting, optimize was prevented at this layer. Global average pooling, on the other hand, encapsulates the spatial information, making it more resistant to spatial translations of the input data. GlobalAveragePooling procedure, the pool size is also set to the whole input map size, but the average of the input map is computed one time in this process. Figure (3.7) and algorithm (3.5) illustrate this layer.

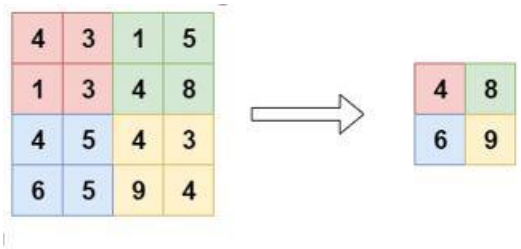


Figure (3.6) Max pooling operation.

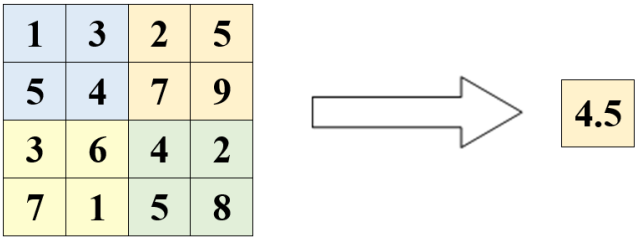


Figure (3.7) Global average pooling operation.

Algorithm (3.4) Max Pooling algorithm

Algorithm name: **Max_ pooling process.**

Input: feature_map

Output: downsampling_feature_map

begin

```
1. Max=feature_map [1,1]  // put the first element in variable.
2. For i = 1 to feature_map.rows
3.     For j =1 to feature_map.columns
4.         Max = feature_map(i,j)
5.               For  m = i to i+1
6.                    For n = j to j+1
7.                      If  feature_map(m,n) > Max
8.                           Max = feature_map(m,n)
9.                     End-n
10.                End-m
11.         downsampling_feature_map [L1, L2] = max
12.         L2 = L2 + 1
13.      End-j
14.   L1 = L1 + 1
15. End-i
16. End
```

D. Adam Optimizer

Adam optimizer had been used to modify the layers weights of model to improve performance of it. This optimizer work in backpropagation process when loss function is high.

Algorithm (3.5) Global Average pooling algorithm

Algorithm name: **Global Average_ pooling process.**

Input: feature_map

Output: average_map

1. AVG=0 // initial value of avarage
2. For i = 1 to feature_map.rows
3. For j =1 to feature_map.columns
4. AVG = AVG+feature_map(i,j)
5. End-j
6. End-i
7. average_map= AVG / (feature_map.rows * feature_map.columns)
8. **Return** average_map
9. **End**

Algorithm 3.6 Adam Optimizer algorithm

Algorithm name: Adam optimizer

Input: network weights

Output: updated network weights

Begin

- $m = 0$, this is the initial moment vector as described in Momentum
- $v = 0$, this is the second moment vector, which is handled the same way in RMSProp.
- $t = 0$
- Update t, t := t + 1
- • Find the gradients divided by derivatives (g) with regard to t. In this case, g is the same as (dw and db respectively).
- Make changes to the first moment mt.
- Keep the second moment up-to-date
- Compute the bias-corrected mt.
- Figure out the bias-adjusted vt.
- Modify the variables θ

Adam's loop will keep going until it finds a solution.

3.3.3 Infection Localization

In this stage, the infected area has been located as a heatmap using GRAD-CAM technique after the detection in previous stage.

3.3.3.1 Gradient Class Activation Map (GRAD-CAM)

In this thesis, GRAD-CAM has been utilized to determine which aspect of the chest CT scan was the most important in determining whether or not the patient was infected with COVID-19. GRAD-CAM take the feature map of the last convolution layer of the model and explain the higher probability of prediction class. It applied the equation (2.20) to extract the score of each class, then multiplied these scores with feature map pixel according equation (2.21). Deep neural models based on the CNN have enabled unprecedented advances in a wide range of computer vision tasks. Although these tasks allow for improved performance, it is difficult to analyze them since they cannot be decomposed into components that can be understood on an individual level. Therefore, when today's intelligent systems fail, they frequently fail in a spectacular and embarrassing manner without providing a warning or explanation. This leaves the user staring at a nonsensical output and asking why the system behaved as it did.

3.3.3.2 GRAD-CAM Fundamental

The fundamental concept behind GRAD-CAM is the same fundamental concept that underpins CAM: the idea was to make use of the spatial information that is retained by the use of convolutional layers, in order to gain an understanding of which aspects of an input image were crucial for a decision regarding a classification.

GRAD-CAM, which is conceptually similar to CAM, utilizes the feature maps produced by a CNN's final convolutional layer. With GRAD-CAM the final convolutional layers provide the optimal balance between semantics at a high level and spatial information.

The following is a diagram depicting the components of a neural network model that are relevant to GRAD-CAM:

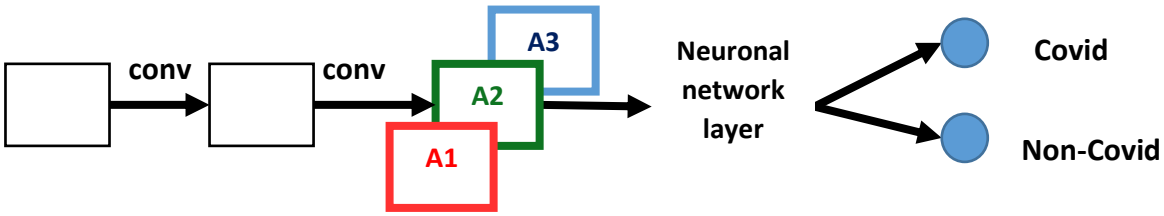


Figure (3.8) Featuremap of CNN layers [106]

The CNN is made up of several convolutional layers, which are represented in the design by the letter "conv." The last convolutional layer was responsible for producing the A1, A2, and A3 feature maps, which are displayed in the same manner as the CAM drawing.

The result of running GRAD-CAM essentially a heatmap in which each "hot" area belongs to a specific class.

$L^{c}_{GRAD-CAM} \in \mathbb{R}$ class-discriminative localization map

Width u, height v, class c

If there are ten distinct output classes, it can generate ten distinct GRAD-CAM heatmaps from a single input image by creating one heatmap for each of the ten output classes, and this is useful because infection may be appear more than ones these heatmaps will point to each one of them. The steps of GRAD-CAM have been listed in algorithm (3.7).

Algorithm 3.7 GRAD-CAM algorithm

Algorithm name: GRAD-CAM process

Input: feature map (last convolution layer)

Output: heatmap of the target class

Begin

1- Find the Gradient of y_c with respect to A^k (the feature map activation) of a convolutional layer according to equation (2.20).
2- Find the Global average bool the gradient the width, height dimensions to find out how important each neuron (α_k^c)
3- Figure out the final GRAD-CAM heatmap by taking a weighted combination of A^k where the weight are α_k^c previously calculated by applie equation (2.21).

End

CHAPTER FOUR
THE EXPERIMENTAL RESULTS

4.1 Introduction

The proposed system results have been presented in this chapter. The 2760 images contain 1380 for both two classes Covid and Non-Covid that trained and tested on this system. The dataset represented as a balance dataset where accuracy per class did not require because the sampling process will be balanced between the two classes. The proposed system consists of three stages. Each stage is with multiple steps. The first stage is the preprocessing step that contains converting from nii file to png file, image segmentation, image resizing, and image normalization. The second stage contains the CNN tuned VGG16 model where the data have been trained and tested to obtain the result. The final stage is the infection localization process that applied the GRAD-CAM algorithm to localize the target class features (infection area).

4.2 Requirement

The system has been built using python as a programming language and the extraction of RoI from images was done using C# 2019 programming language. The hardware requirement of this system is google colab pro.

4.3 The Experimented Dataset

This thesis uses the datasets mentioned in the previous chapter in Section 3.2 when training and testing the proposed model, the proposed-system uses the first dataset “Large COVID-19 CT scan slice dataset”. Witch consist of 7,590 COVID-19 cases from 466 persons, 6,880 normal cases from 604 persons. And the second dataset “COVID-19 CT Lung and Infection Segmentation Dataset”. Witch consist of 1024 Covid images and 1736 Non Covid images. An

augmentation such as shear, height shift, width shift, and rotation applied on this dataset, and the number of images was increased to 9677 images.

The result obtained from using the first dataset with reached 99.7% of accuracy, loss of 0.0085, and validation loss of 0.0162. With this dataset, the localization of infected areas is not accurately accomplished in this thesis. The result with second dataset, the system reached test accuracy of 99.82%, 0.0044 test loss, and the localization was accurately accomplished.

4.4 Proposed System Results

The proposed system consists three phases and each phase consist of steps, in this section the whole system phases as follow:

4.4.1 Preprocessing Phase Results

The preprocessing stage mentioned in Section 3.3.1 includes three steps and all dataset images go through it. These preprocessing steps include:

4.4.1.1 Converting Nifti file to png files results

The second dataset contains image format nii files. These images or information are designated for medical analysis because, in average 80% of users utilize standard images such as JPEG, PNG, and TIFF. However, in the medical area, these images are useless since further information about the patient and how the image was captured is required for a more exact interpretation.

This information is saved in the same file, either Dicom or Nifti, and can call them using something called tags, so to utilize these tags to retrieve the required information, such as the patient ID, name, and so on.

That was information exclusively for the patient, but there is another side that is unique to the image itself. They named them by tags as well, although they are a little different. Converting nii files to png images has been explained in section

3.3.1.1. the Nifti file consists of a set of slices. Each slice should be converted to png image. Figure (4.1) represents nii image, figure(4.2) explain png image.

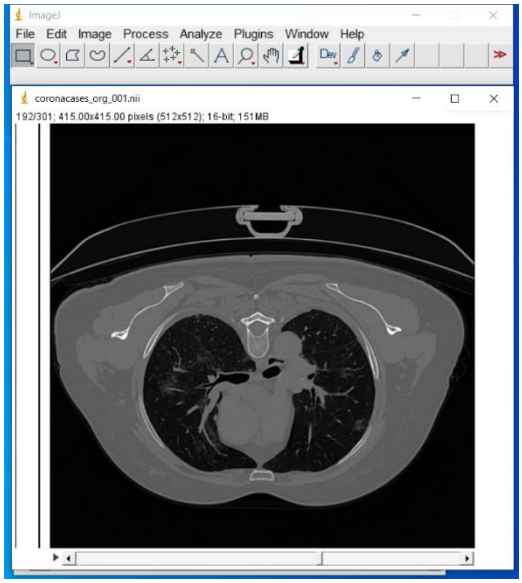


Figure (4.1) Sample of nii image

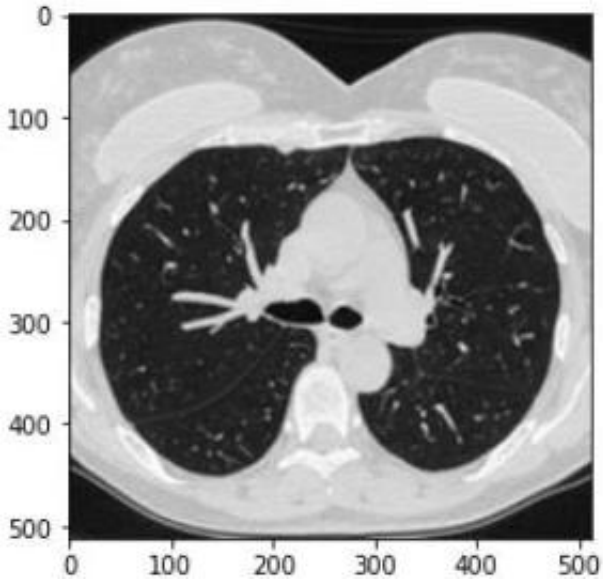


Figure (4.2) Sample of png image

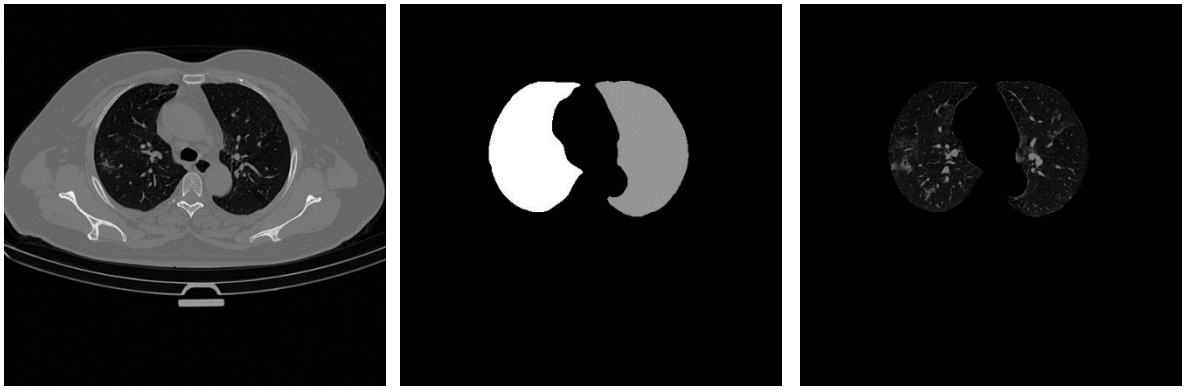
4.4.1.2 ROI Extraction Results

This process concerned with ANDing the raw image with the corresponding lungs mask in order to extract lung image region from raw CT scan image. The segmentation process applied to remove the outside lung tissue and prevent them from sharing decision making. This process explained in section 3.3.1.2. Figure (4.2) shows the ROI extraction process.

The impotence of ROI extraction has been explained in figure (4.3) when the localization process has been classified region outside the lung as an infected area. This means that region shared decision making.

a **b** **c**

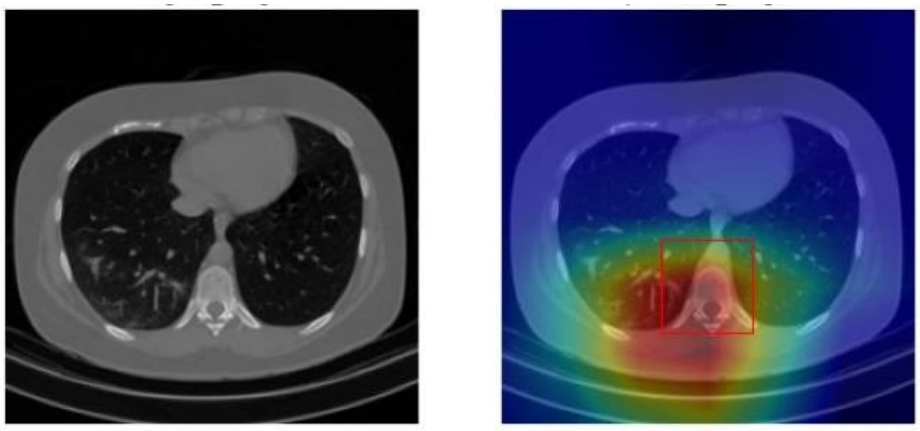
a **b** **c**



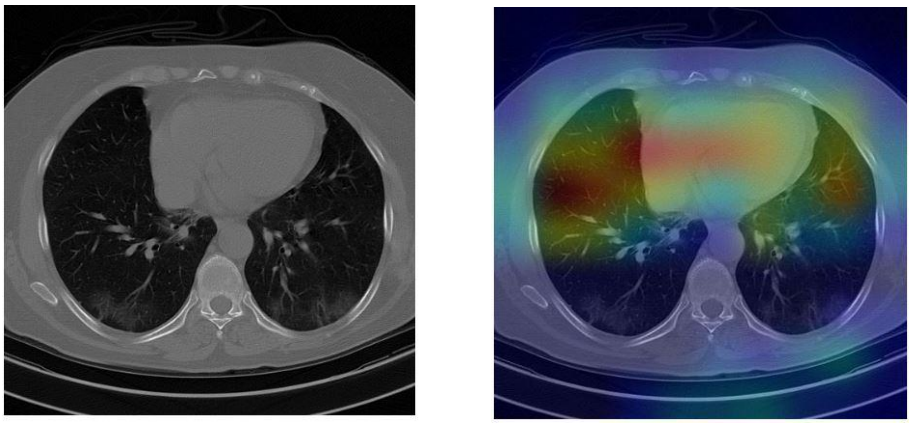
a **b** **c**

Figure (4.3) ROI extraction using ANDing technique:

(a) CT raw image, (b) lung mask, (c) the result of ANDing process



a **b**



a **b**

Figure (4.4) Heatmap appoints to tissue outside the lung.

(a) CT image, (b) localization of infection area.

4.4.1.3 Image Resizing Results

Deep learning with all its models receives images with fixed size that depend on the model. CNN models in general consist of layers, where each layer represents a set of image filters that accepting image with fixed size according to model designation, where each model accepts images with special size. VGG16

model accepts images with of 224*224 size. Figure (4.4) shows the results of resizing image process.

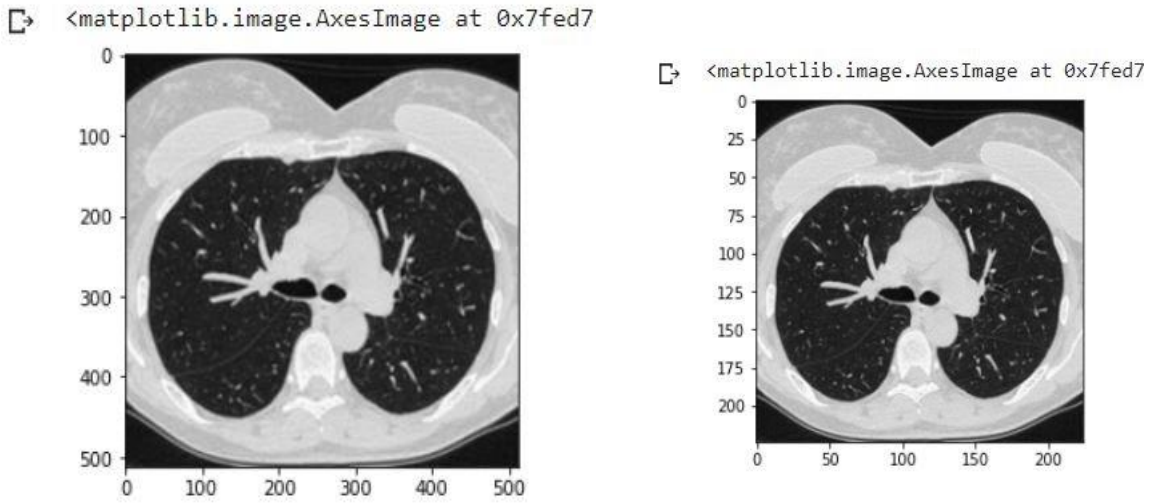


Figure (4.5) Image resizing process.

4.4.1.4 Image Normalization Results

The result of the fourth step preprocessing in section (3.3.1.4) is image normalization. The image normalization puts all the image pixels with the value of 0-1. Figure (4.5) explains the normalization process.

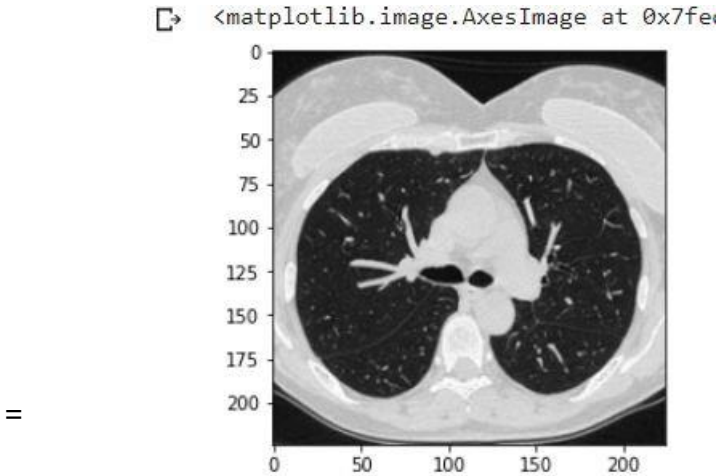


Figure (4.6) Normalization Process.

4.4.1.5 Results Of Training Model

The dataset has been divided to 80: 10:10 (train: valid: test) which means 7741 images for training and 388 images for testing and validation. This process explained in 3.3.1.5. Figure (4.6) shows the percentage splitting process.

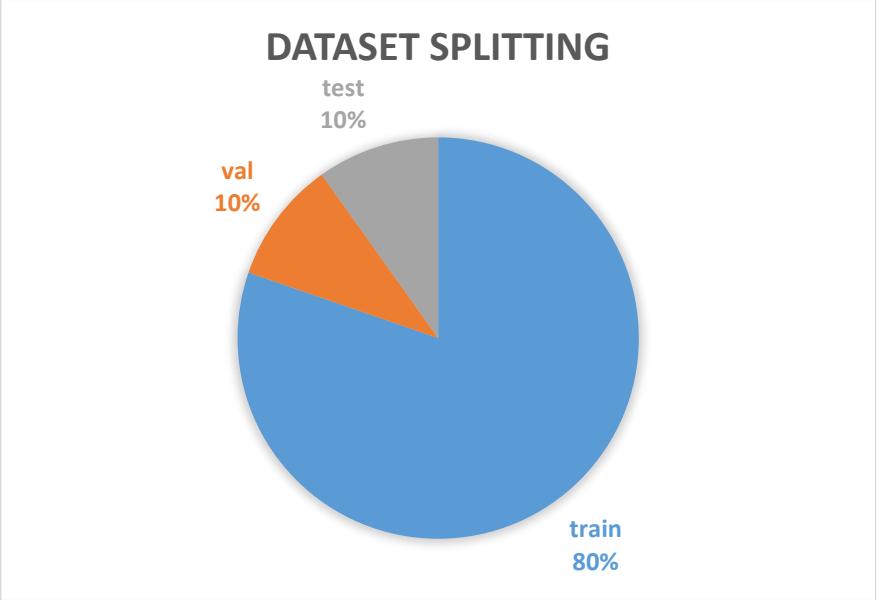


Figure (4.7) Dataset splitting ratio.

4.4.2 Proposed Model Phase Results

Different models of CNN have been applied like (VGG16, Inception v3, Xception, Resnet50) on the COVID-19 CT Lung and Infection Segmentation Dataset with the same circumstances such as resources and number of epochs. The observed results various in accuracy and other criteria all these will be explained separately as follows.

4.4.2.1 Results of Resnet 50

The Resnet50 has been trained on the current dataset, the observed result can be summarized as follows in table (4.1). Figure (4.7) explains the training, validation loss while figure (4.8) shows the training and validation accuracy for the Resnet50 model.

Table (4.1) illustrate the result of Resnet50 architecture.

Train loss	Test loss	Train accuracy	Test accuracy	Total parameters	Trainable parameters	Non-trainable parameters
0.4689	0.4489	0.7708	0.7939.	24,637,820	1,050,113	23,587,710

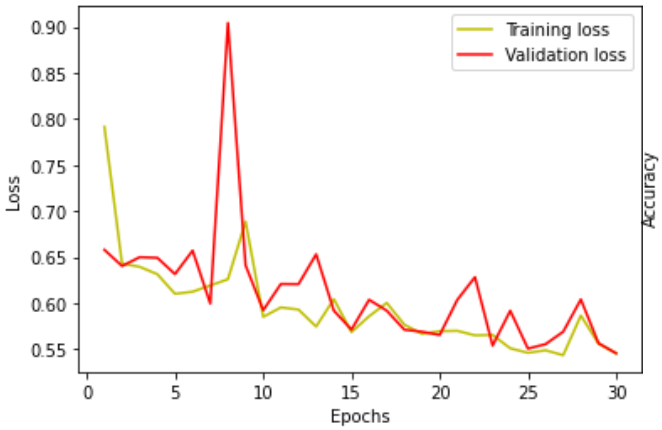


Figure (4.8) Resnet50 training and testing loss.

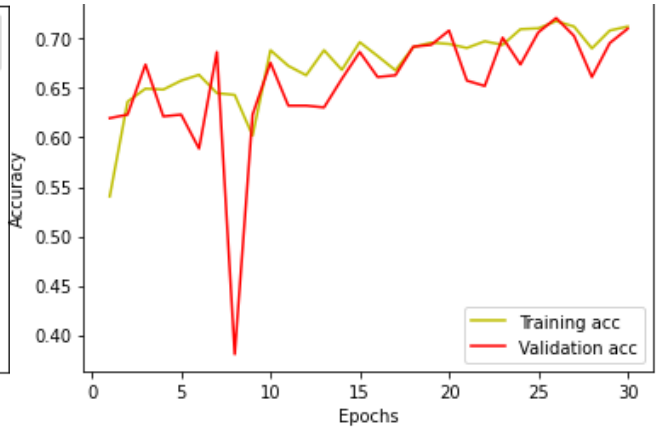


Figure (4.9) Resnet 50 training and testing accuracy.

4.4.2.2 Result of Standard VGG16 Architecture

The standard VGG16 has been trained on the same dataset, the obtained result can be summarized as follows in table (4.2). Figure (4.9) explains the training and validation loss while figure (4.10) shows the training and validation accuracy for the standard VGG16 model.

Table (4.2) illustrate the result of VGG16 architecture.

Train loss	Test loss	Train accuracy	Test accuracy	Total parameters	Trainable parameters	Non-trainable parameters
0.0031	0.0120	0.9999	0.9955	14,780,610	4,785,538	9,995,072

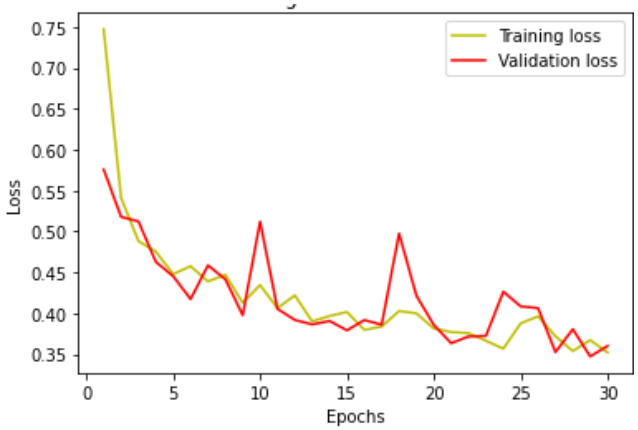


Figure (4.10) Standard VGG16 training and testing loss.

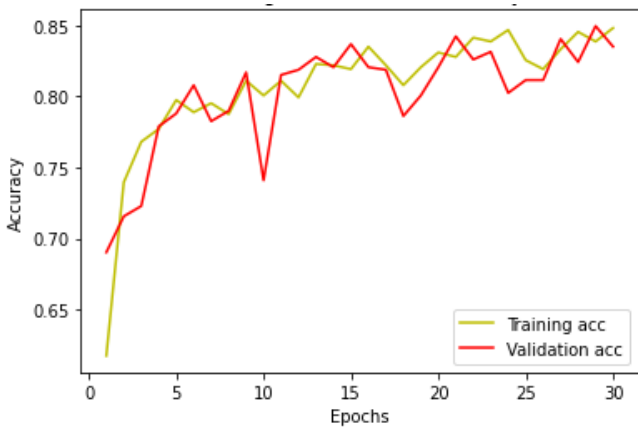


Figure (4.11) Standard VGG16 training and testing accuracy.

4.4.2.3 Result of Xception Architecture

The Xception has been trained on the same dataset, the obtained result can be summarized as follows in table (4.3).

Table (4.3) illustrate the result of Xception architecture.

Train loss	Test loss	Train accuracy	Test accuracy	Total parameters	Trainable parameters	Non-trainable parameters
0.0546	0.4896	0.9903	0. 9529	123,625,002	105,930,242	17,694,760

Figure (4.11) explains the training accuracy, training loss, validation loss, and validation accuracy for the Xception model.

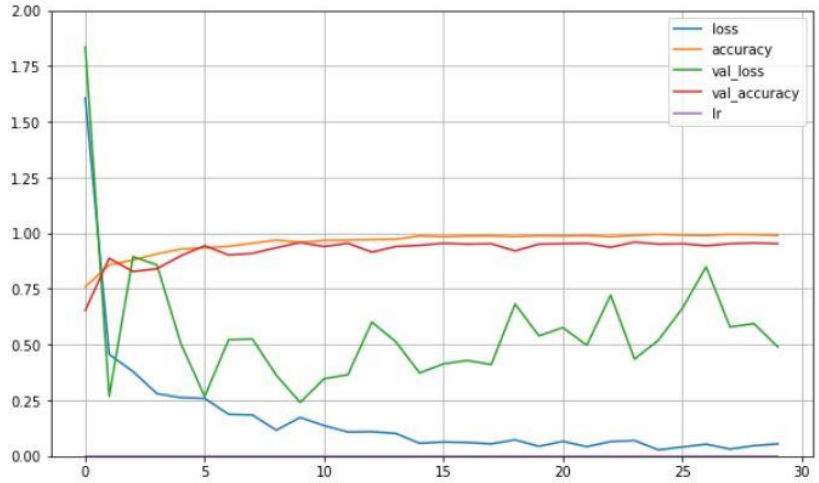


Figure (4.12) XceptionV3 the **training accuracy, training loss, testing loss, and testing accuracy.**

4.4.2.4 Result of Inception Architecture

The Inception has been trained on the same dataset, the obtained result can be summarized as follows in table (4.4). Figure (4.12) explains the training accuracy, training loss, testing loss, and testing accuracy for the Inception model.

Table (4.4) illustrate the result of Inception architecture.

Train loss	Test loss	Train accuracy	Test accuracy	Total parameters	Trainable parameters	Non-trainable parameters
0.4787	0.2656	0.8058	0.8915	74,234,658	52,431,874	21,802,784

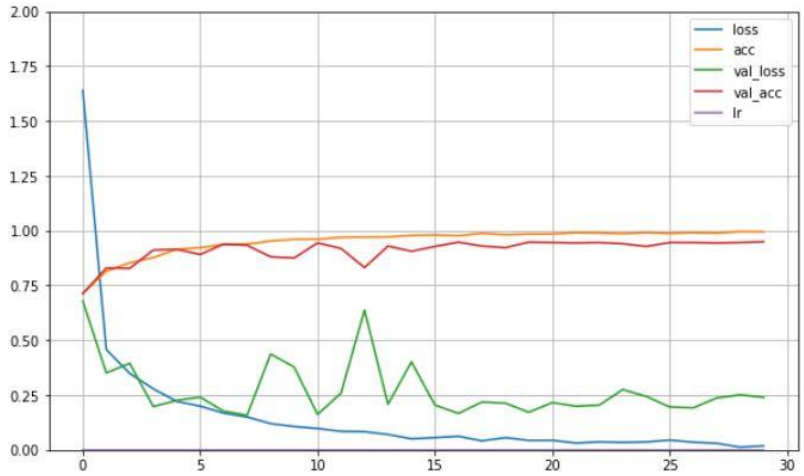


Figure (4.13) InceptionV3 the **training accuracy, training loss, testing loss, and testing accuracy.**

The summary of the four model has been listed in table (4.5).

Table (4.5) the summery of the four CNN models

	Resnet50	**Standard VGG16**	**Xception**	**Inception**
Training loss	0.4689	0. 0031	0. 0546	0.4787
Training accuracy	0.7708	0.9999	0.9903	0.8058
Validation loss	0.4489	0.0120	0.4896	0.2656
Validation accuracy	0.7939	0.9955	0.9529	0.8915
Total parameters	24,637,820	14,780,610	123,625,002	74,234,658
Trainable parameters	1,050,113	4,785,538	105,930,242	52,431,874
Non-Trainable parameters	23,587,710	9,995,072	17,694,760	21,802,784
Number of epochs	30	30	30	30

4.4.2.5 Result of Modified VGG16 Architecture

The results of Rtesnet50 model are shown in figure (4.8), standard VGG16 figure (10.4) and Inception V3 figure (4.11). The first two models show an instable training accuracy and validation accuracy during the 30 epochs training. The training loss and validation loss were instable in the third model figure (4.11)

The observation of table (4.5) shows that the VGG16 has the highest accuracy. The plan is to tune the VGG16 in order to gain best accuracy.

After tuning the VGG16 model training process has been applied second dataset that contain 9677 images. The modified VGG16 obtained result as explained in table (4.6).

Table (4.6) illustrate the result of modified VGG16 architecture.

Train loss	**Test loss**	**Train accuracy**	**Test accuracy**	**Total parameters**	**Trainable parameters**	**Non-trainable parameters**
0.0001	0.0048	100.0	0.9982	14,717,762	7,081,474	7,636,288

Figure (4.12) explains the training accuracy, training loss, testing loss, and testing accuracy for the modified VGG16 model.

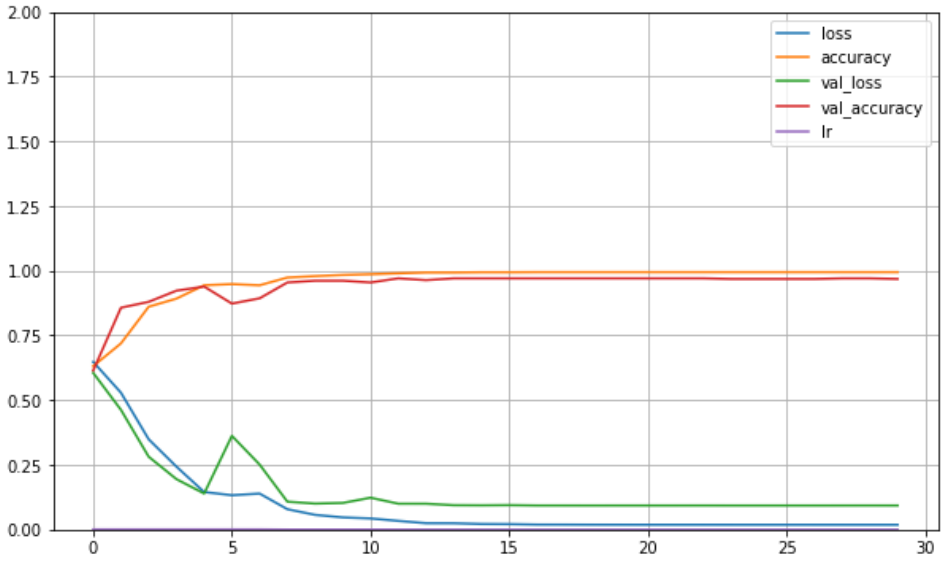


Figure (4.14) Tuned VGG16 explain the training accuracy, training loss, validation loss , and validation accuracy.

4.4.2.6 Other Metrics Results

As follows table (4.7) represent the precision, recall, F1_score for the above trained architectures.

Table (4.7) Precision, recall, F1_score metrics results

Condition form	TP	FP	TN	FN	Precision	Recall	F1_score
VGG16	194	1	191	2	99.48	98.97%	99.22%
Resnet50	176	46	126	40	79.27%	81.48%	80.35%
Xception	159	7	211	11	95.78%	93.52%	94.63%
Inception	151	15	195	27	90.96%	84.83%	87.78%
Modified VGG16	222	0	165	1	100%	99.55%	99.77%

4.4.3 Infection Localization Phase Result

This phase of the proposed system concerned with localizing the feature of infection area on the test image depending on training process. The feature map from the last convolution layer has been used by the GRAD-CAM method for heatmap the RoI. This technique has been summarized in section 3.3.3.1. Figure (4.13) shows the localization process.

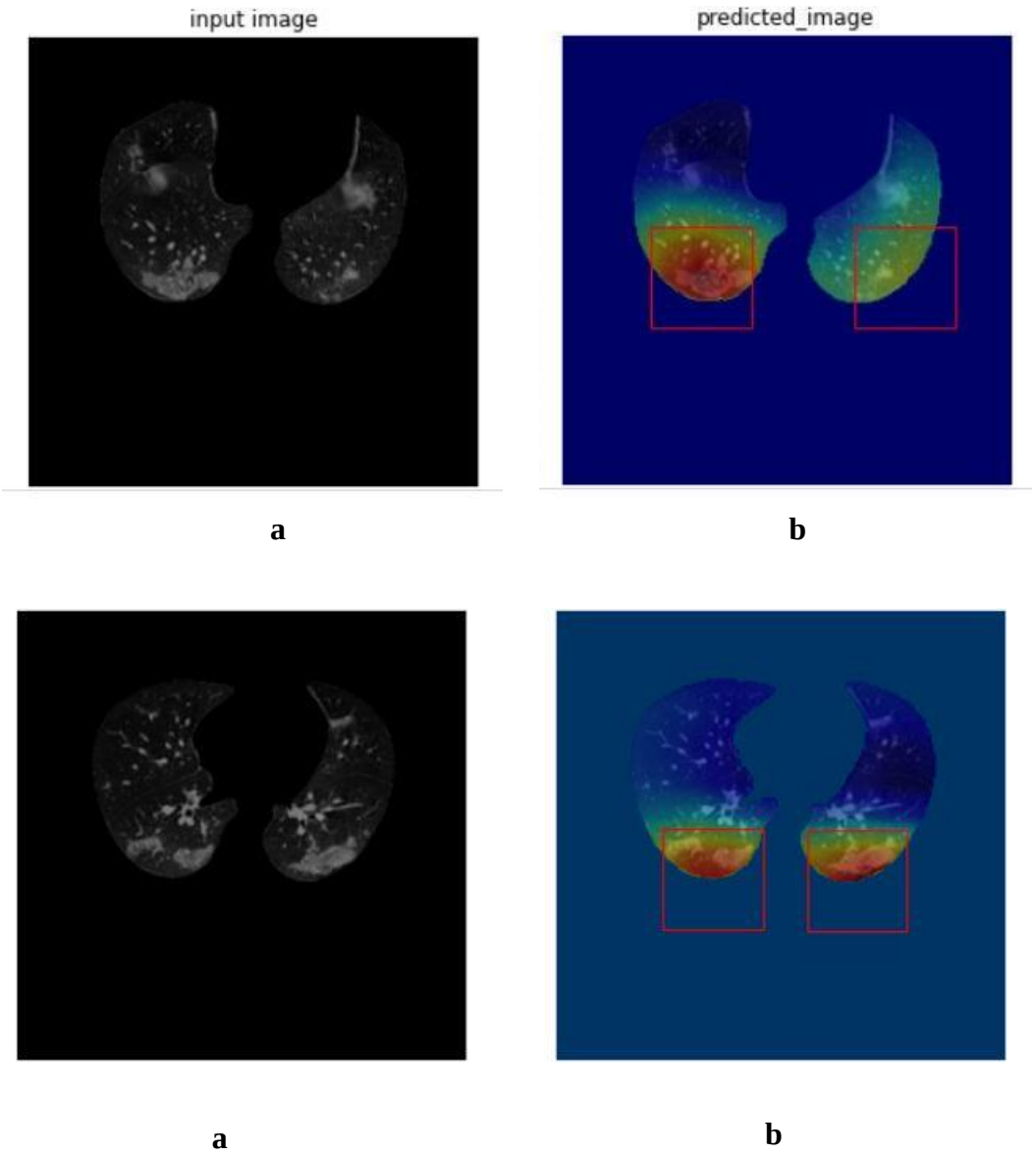


Figure (4.15) Image with infection localization: (a) input image, (b) predicted image.

4.5 Comparison With Other Works

Table (4.8) illustrates the comparison of this work with the other works.

Study	Year	Image type	Methods	Dataset	localization	accuracy
s.wang et al [115]	2021	CT	modified the inception transfer-learning	collected CT images from three hospitals	No	85.2%.
Pathak et al. [116]	2020	CT	ResNet50 architecture used for the detection and classification	One dataset	No	94.78%
Tsiknakis et al. [117]	2020	X-ray	Modified InceptionV3	One dataset	No	76%
Elasnaoui and Chawki [29]	2020	X-ray	InceptionResNetV2	One dataset	No	92.60%
Rahimzadeh and Attar [118]	2020	X-ray	modified architecture of on the combination of ReNet50V2 and Xception	One dataset	No	91.4%
Abraham and Nair [119]	2020	X-ray	Five architectures used (Squeezenet, Darknet-53, MobilenetV2, Xception, and Shufflenet)	Two datasets	No	First dataset 91.16% Second dataset 97.44%
F. Sadik et al [120]	2022	CT	modified DenseNet	COVID-19 CT Lung and Infection	Yes	87.5%

				Segmentation Dataset		
X. He et al [121]	2021	CT	MNas3DNet41	COVID-19 CT Lung and Infection Segmentation Dataset	No	87.14%
Proposed Methodology	2022	**CT**	**Resnet50**	COVID-19 CT Lung and Infection Segmentation Dataset	**Yes**	79.39%
			Standard VGG16		**Yes**	99.55%
			Xception		**Yes**	95.29%
			Inception		**Yes**	89.15%
			modified VGG16		**Yes**	99.82%

The above research work on different datasets to achieve diagnosis Covid-19 of lung. s.wang et al. (2021) modify Inception network and used one dataset to classify Covid-19 or Non Covid, an accuracy of 85.2% without localization of infection area. Tsiknakis et al. (2020) also modify Inception network and used one dataset to for binary classification without localizing. Elasnaoui and Chawki (2021) they used InceptionResNetV2 to classification. Rahimzadeh and Attar (2020) they used modified architecture on the combination of ReNet50V2 and Xception. Abraham and Nair (2020) they used Five architectures used (Squeezenet, Darknet-53, MobilenetV2, Xception, and Shufflenet) for classification with two datasets. F. Sadik et al (2022) they used modified DenseNet with same dataset of this thesis, this research reach an accuracy of

87.5% and used GRAD-CAM to localize infection area. X. He et al (2021) they used MNas3DNet41 network and same dataset of this thesis and reach an accuracy of 87.14% without localization of infection area. This thesis differ from above researches in accuracy and localizing the infection area.

CHAPTER FIVE
CONCLUSION AND FUTURE WORKS

5.1 Conclusions

Several conclusions during the design and implementation of the proposed system have been drawn. Following the results of this thesis:

1. ANDing segmentation between image and lung mask prevent useless information from being a part of decision making, because of the similarity between the infection feature and the rest of image.
2. Training the model using the ROI of images provides better target class localization because the training is applied on interesting features only.
3. The architecture of VGG16 has been modified to improve the performance to classify the current dataset images. The modification of this architecture is done in the last three layers by replacing them with another two layers (global average pooling and dense layer). This modification is applied on the deeper layers because these layers are class discriminative.
4. The model is with little numbers of layers less time-consuming to modify.

5.2. Future Works

There are some suggestion for future works:

1. The proposed system can be developed to detect and localize more than one disease if the dataset was available.
2. The system can also be developed by utilizing information from persons they are getting well from the virus.

3. The system can also be developed to give a severity score of infection.
4. Build a model used to extract the lung from the input images instead of using a masked dataset by utilizing Deep Learning such as Unet architecture.

References

[1] Wu F, Zhao S, Yu B, et al. A new coronavirus associated with human respiratory disease in China. Nature 2020;579(7798):265–9.

[2] Huang C, Wang Y, et al. Clinical features of patients infected with 2019 novel coronavirus in Wuhan. China Lancet 2020;395(10223):497–506.

[3] World Health Organization. Pneumonia of unknown cause–china. Emergencies preparedness. Response, Disease Outbreak News, World Health Organization (WHO); 2020.

[4] S. Wang et al., . "A deep learning algorithm using CT images to screen for Coronavirus disease (COVID-19)," Eur. Radiol., vol. 70, no. December 2020, p. 102987, 2021, DOI: 10.1016/j.bspc.2021.102987.

[5] (15 October 2020). WHO Coronavirus Disease. Available: https://covid19.who.int/.

[6] S. Huang, J. Yang, S. Fong, and Q. Zhao, "Artificial intelligence in the diagnosis of covid-19: Challenges and perspectives," Int. J. Biol. Sci., vol. 17, no. 6, pp. 1581–1587, 2021, doi: 10.7150/ijbs.58855.

[7] Singhal T. A review of coronavirus disease-2019 (COVID-19). Indian J Pediatr 2020;87:281–6.

[8] Zu ZY, Jiang MD, Xu PP, Chen W, Ni QQ, Lu GM, Zhang LJ. Coronavirus disease 2019 (COVID-19): a perspective from China. Radiology 2020.

[9] P. given, A. Shalbaf, and M. Vafaeezadeh, "Automated detection of COVID-19 using an ensemble of transfer learning with deep convolutional neural network based on CT scans," Int. J. Comput. Assist. Radiol. Surg., vol. 16, no. 1, pp. 115–123, 2021, DOI: 10.1007/s11548-020-02286-w.

[10] Yoon SH, Lee KH, Radiol J, et al. Chest radiographic and CT findings of the 2019 novel coronavirus disease (COVID-19): analysis of nine patients treated in Korea. Korean 2020;21(4):494–500.

[11] A. Rehman, T. Saba, U. Tariq, and N. Ayesha, "Deep Learning-Based COVID-19 Detection Using CT and X-Ray Images: Current Analytics and Comparisons," IT Prof., vol. 23, no. 3, pp. 63–68, 2021, DOI: 10.1109/MITP.2020.3036820.

References

[12] E. Hussain, M. Hasan, A. Rahman, I. Lee, and T. Tamanna, "Since January 2020 Elsevier has created a COVID-19 resource center with free information in English and Mandarin on the novel coronavirus COVID- 19. The COVID-19 resource center is hosted on Elsevier Connect, the company's public news, and information," no. January 2020.

[13] Chen J, See KC. Artificial Intelligence for COVID-19: Rapid Review. J Med Internet Res. 2020; 22: e21476.

[14] Vaishya R, Javaid M, Khan IH, Haleem A. Artificial Intelligence (AI) applications for COVID-19 pandemic. Diabetes Metab Syndr. 2020; 14:337-9.

[15] Zhao X, Liu L, Qi S, Teng Y, Li J, Qian W. Agile convolutional neural network for pulmonary nodule classification using CT images. Int J CARS 2018;13:585–95.

[16] Liu C, et al. TX-CNN: Detecting tuberculosis in chest x-ray images using convolutional neural network. In: IEEE International conference on image processing (ICIP), Beijing; 2017. p. 2314–18.

[17] Zou L, Zheng J, Miao C, Mckeown MJ, Wang ZJ. 3D CNN-based automatic diagnosis of attention deficit hyperactivity disorder using functional and structural MRI. IEEE Access 2017;5:23626–36.

[18] Liu, Li W, Zhao N, et al. Integrate domain knowledge in training CNN for ultrasonography breast cancer diagnosis Medical image computing and computer-assisted intervention – MICCAI 2018. MECCA 2018. Lecture Notes in Computer Science, vol 11071. Cham: Springer; 2018.

[19] W. Q. Yan, Computational Methods for Deep Learning - Theoretic, Practice and Applications. Springer, 2020.

[20] A. Zhang, Zachary, C. Lipton, M. Li, and A. J. Smola, "Dive into Deep Learning," Distill, pp. 345–391, 2021, [Online]. Available: https://doi.org/10.48550/arXiv.2106.11342.

[21] E. F. Ohata et al. Automatic detection of COVID-19 infection using chest X-ray images through transfer learning. IEEE/CAA Journal of Automatica Sinica. 2020.

References

[22] A. Mangal et al., "CovidAID: COVID-19 Detection Using Chest X-Ray," arXiv preprint arXiv:2004.09803, 2020.

[23] A.I. Khan J.L. Shah M.M. Bhat "Coronet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images," Computer Methods and Programs in Biomedicine 2020 105581.

[24] A. Sedik et al., "Deploying machine and deep learning models for efficient data-augmented detection of COVID-19 infections," Viruses, vol. 12, no. 7, 2020, doi: 10.3390/v12070769.

[25] Wang, B.; Jin, S.; Yan, Q.; Xu, H.; Luo, C.; Wei, L.; Zhao, W.; Hou, X.; Ma, W.; Xu, Z.; et al. AI-assisted CT imaging analysis for COVID-19 screening: Building and deploying a medical AI system. Appl. Soft Comput. 2021, 98, 106897.

[26] Apostolopoulos, I.D.; Mpesiana, T.A. Covid-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys. Eng. Sci. Med. 2020, 43, 635–640.

[27] J. F. Hern, "An ensemble approach for multi-stage transfer learning models for COVID-19 detection from chest CT scans," Intell. Med. J., no. January 2020.

[28] Rahaman M M et al 2020 Identification of COVID-19 samples from chest x-ray images using deep learning: a comparison of transfer learning approaches J. X-Ray Sci. Technol. 28 821–39

[29] Asnaoui K E and Chawki Y 2020 Using x-ray images and deep learning for automated detection of coronavirus disease J. Biomol. Struct. Dyn. (https://doi.org/10.1080/07391102.2020.1767212).

References

[30] Zheng C, Deng X, Fu Q, Zhou Q, Feng J, Ma H, Liu W and Wang X 2020 Deep learning-based detection for COVID-19 from chest CT using a weak label medRxiv (https://doi.org/10.1101/2020.03.12.20027185)

[31] A. Halder and B. Datta, "COVID-19 detection from lung CT-scan images using transfer learning approach COVID-19 detection from lung CT-scan images using transfer learning approach," 2021.

[32] D. Shome et al., "Covid-transformer: Interpretable covid-19 detection using vision transformer for healthcare," Int. J. Environ. Res. Public Health, vol. 18, no. 21, 2021, DOI: 10.3390/ijerph182111086.

[33] S. Kugunavar and C. J. Prabhakar, "Convolutional neural networks for the diagnosis and prognosis of the coronavirus disease pandemic," Vis. Comput. Ind. Biomed. Art, vol. 4, no. 1, 2021, DOI: 10.1186/s42492-021-00078-w.

[34] Wang, S.; Kang, B.; Ma, J.; Zeng, X.; Xiao, M.; Guo, J.; Cai, M.; Yang, J.; Li, Y.; Meng, X.; et al. A deep learning algorithm using CT images to screen for Coronavirus disease (COVID-19). Eur. Radiol. 2021, 1–9.

[35] Shi, F.; Xia, L.; Shan, F.; Song, B.; Wu, D.; Wei, Y.; Yuan, H.; Jiang, H.; He, Y.; Gao, Y.; et al. Large-scale screening to distinguish between COVID-19 and community-acquired pneumonia using infection size-aware classification. Phys. Med. Biol. 2021, 66, 065031.

[36] E. A. Abbood and T. A. Al-Assadi, "GLCMs Based multi-inputs 1D CNN Deep Learning Neural Network for COVID-19 Texture Feature Extraction and Classification," Karbala Int. J. Mod. Sci., vol. 8, no. 1, pp. 28–39, 2022, DOI: 10.33640/2405-609X.3201.

[37] H. Panwar, P. K. Gupta, M. K. Siddiqui, R. Morales-Menendez, P. Bhardwaj, and V. Singh, "A deep learning and grad-CAM based color

visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images," Chaos, Solitons and Fractals, vol. 140, p. 110190, 2020, doi: 10.1016/j.chaos.2020.110190.

[38] Umair, Muhammad et al. "Detection of COVID-19 Using Transfer Learning and Grad-CAM Visualization on Indigenously Collected X-ray Dataset." Sensors (Basel, Switzerland) vol. 21,17 5813. 29 Aug. 2021, doi:10.3390/s21175813.

[39] S. Ahuja, B. K. Panigrahi, N. Dey, V. Rajinikanth, and T. K. Gandhi, "Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices," Appl. Intell., vol. 51, no. 1, pp. 571–585, 2021, doi: 10.1007/s10489-020-01826-w.

[40] T. Zebin and S. Rezvy, "COVID-19 detection and disease progression visualization: Deep learning on chest X-rays for classification and coarse localization," Appl. Intell., vol. 51, no. 2, pp. 1010–1021, 2021, doi: 10.1007/s10489-020-01867-1.

[41] H. Alshazly, C. Linse, E. Barth, and T. Martinetz, "Explainable COVID-19 Detection Using Chest CT Scans and Deep Learning," Sensors, vol. 29, no. 7553, pp. 1–73, 2021, [Online]. Available: http://deeplearning.net/.

[42] R. Szeliski, Computer Vision Algorithms and Applications Second Edition, no. June. Springer, 2022.

[43] H. Jiang, Machine Learning Fundamentals: A Concise Introduction. 2021.

[44] N. Buduma and J. Papa, Fundamentals of Deep Learning, vol. 1542. O'Reilly Media, 2022.

References

[45] S. N. and A. Z. Muhammad Imran Razzak, "Deep Learning for Medical Image Processing: Overview, Challenges and the Future," springer, pp. 323–349, 2018.

[46] F. Altaf, S. M. S. Islam, N. Akhtar, and N. K. Janjua, "Going deep in medical image analysis: Concepts, methods, challenges, and future directions," IEEE Access, vol. 7, pp. 99540–99572, 2019, doi: 10.1109/ACCESS.2019.2929365.

[47] V. W. Yam, H. Kong, and H. Kong, X-ray Free Electron Lasers. Royal Society of Chemistry, 2018.

[48] J. Liu et al., "Applications of deep learning to MRI Images: A survey," Big Data Min. Anal., vol. 1, no. 1, pp. 1–18, 2018, doi: 10.26599/BDMA.2018.9020001.

[49] H. Fujita, "AI-based computer-aided diagnosis (AI-CAD): the latest review to read first," Radiol. Phys. Technol., vol. 13, no. 1, pp. 6–19, 2020, doi: 10.1007/s12194-019-00552-4.

[50] M. K. Bhuyan, Computer Vision and Image Processing: Fundamentals and Applications, vol. 5, no. 1. CRC Press, 2020.

[51] W. E. Snyder and H. Qi, Fundamentals of computer vision. 2017.

[52] H. T. Sencar, L. Verdoliva, and N. Memon, Multimedia Forensics. Springer, 2022.

[53] P. Agrawal, C. Gupta, A. Sharma, V. Madaan, and N. Joshi, Machine Learning and Data Science, vol. 7, no. 1. 2022.

[54] T. Trappenberg, fundamentals of machine learning. Oxford University Press, 2020.

[55] B. Mehlig, "Machine learning with neural networks," 2021.

[56] C. Aggarwal, Neural Networks and Deep Learning. Springer, 2018.

References

[57] R. Dastanian, E. Abiri, M. R. Salehi, and A. Akbari, "A new approach

[58] S. W. Smith, "Neural Networks," Digit. Signal Process., pp. 451–480, 2003, doi: 10.1016/b978-0-7506-7444-7/50063-7.

[59] R. W. Pettit, R. Fullem, C. Cheng, and C. I. Amos, "Artificial intelligence, machine learning, and deep learning for clinical outcome prediction," Emerg. Top. Life Sci., vol. 5, no. 6, pp. 729–745, 2021, doi: 10.1042/ETLS20210246.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, 2017, DOI: 10.1145/3065386.

[61] M. H. Hesamian, W. Jia, X. He, and P. Kennedy, "Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges," J. Digit. Imaging, vol. 32, no. 4, pp. 582–596, 2019, DOI: 10.1007/s10278-019-00227-x.

[62] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[63] A. B. Bulsari and S. Palosaari, "Application of neural networks for system identification of an adsorption column," Neural Computing & Applications, vol. 1, no. 2. pp. 160–165, 1993, doi:10.1007/BF01414435.

[64] B. Wicht, "Deep Learning Feature Extraction for Image Processing," University of Fribourg, 2017.

[65] K. R. Hassan and I. H. Ali, "Age and Gender Classification using Multiple Convolutional Neural Network," IOP Conf. Ser. Mater. Sci. Eng., vol. 928, no. 3, 2020, doi: 10.1088/1757- 899X/928/3/032039.

[66] K. R. Raheem and I. H. Ali, "Classifying Facial Expression using Convolution Neural Network," IOP Conf. Ser. Mater. Sci. Eng., vol. 928, no. 3, 2020, doi: 10.1088/1757- 899X/928/3/032036.

[67] N. D. Al-Shakarchy and I. H. Ali, "Detecting abnormal movement of driver's head based on spatial-temporal features of video using deep neural network DNN," Indones. J. Electr. Eng. Comput. Sci., vol. 19,

no. 1, pp. 344–352, 2020, doi: 10.11591/ijeecs.v19.i1.pp344-352.

[68] N. D. Al-Shakarchy and I. H. Ali, "Open and closed eyes classification in different lighting conditions using new convolution neural networks architecture," J. Theor. Appl. Inf. Technol., vol. 97, no. 7, pp. 1970–1979, 2019.

[69] N. D. Al-Shakarchy and I. H. Ali, "Abnormal head movement classification using deep neural network DNN," AIP Conf. Proc., vol. 2144, no. August, 2019, doi: 10.1063/1.5123123.

[70] S. Sharma, S. Sharma, and A. Anidhya, "Understanding Activation Functions in Neural Networks," Int. J. Eng. Appl. Sci. Technol., vol. 4, no. 12, pp. 310–316, 2020.

[71] J. Feng and S. Lu, "Performance Analysis of Various Activation Functions in Artificial Neural Networks," J. Phys. Conf. Ser., vol. 1237, no. 2, 2019, doi: 10.1088/1742-6596/1237/2/022030.

[72] R. H. R. Hahnioser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex- inspired silicon circuit," Nature, vol. 405, no. 6789, pp. 947–951, 2000, doi: 10.1038/35016072.

[73] A. F. M. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," arXiv, no. 1, pp. 2–8, 2018.

[74] S.-H. Han, K. W. Kim, S. Kim, and Y. C. Youn, "Artificial Neural Network: Understanding the Basic Concepts without Mathematics," Dement. Neurocognitive Disord., vol. 17, no. 3, p. 83, 2018, doi: 10.12779/dnd.2018.17.3.83.

[75] S. I. Granshaw, Neural networks and neurodiversity, vol. 36, no. 175. 2021.

[76] S. Theodoridis, "Neural Networks and Deep Learning," in Machine Learning, 2015.

[77] S. P. Siregar and A. Wanto, "Analysis of Artificial Neural Network Accuracy Using Backpropagation Algorithm In Predicting Process (Forecasting)," IJISTECH (International J. Inf. Syst. Technol., vol. 1,

no. 1, p. 34, 2017, doi: 10.30645/ijistech.v1i1.4.

[78] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, 2016, [Online]. Available: http://arxiv.org/abs/1609.04747.

[79] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–15, 2015.

[80] C. Fougstedt, M. Mazur, L. Svensson, H. Eliasson, M. Karlsson, and P. Larsson-Edefors, "Time-domain digital back propagation: Algorithm and finite-precision implementation aspects," Opt. InfoBase Conf. Pap., vol. Part F40-O, 2017, doi: 10.1364/OFC.2017.W1G.4.

[81] D. Rengasamy, M. Jafari, B. Rothwell, X. Chen, and G. P. Figueredo, "Deep learning with dynamically weighted loss function for sensor-based prognostics and health management," Sensors (Switzerland), vol. 20, no. 3, 2020, doi: 10.3390/s20030723.

[82] S. I. Granshaw, "Neural networks and neurodiversity," Remote Sens. Photogramm. Soc. John Wiley Sons Ltd, vol. 36, no. 175, pp. 192–196, 2021, doi: 10.1111/phor.12376.

[83] K.-L. Du and M. N. S. Swamy, Neural Networks and Statistical Learning. 2019.

[84] S. Mangrulkar, "Artificial neural systems," ISA Trans., vol. 29, no. 1, pp. 5–7, 1990, doi: 10.1016/0019-0578(90)90024-F.

[85] P. Gang et al., "Dimensionality reduction in deep learning for chest X-ray analysis of lung cancer," Proc. - 2018 10th Int. Conf. Adv. Comput. Intell. ICACI 2018, no. c, pp. 878–883, 2018, doi: 10.1109/ICACI.2018.8377579.

[86] M. Doan et al., "Deepometry, a framework for applying supervised and weakly supervised deep learning to imaging cytometry," Nat. Protoc., vol. 16, no. 7, pp. 3572–3595, 2021, doi: 10.1038/s41596-021-00549-7.

[87] J. Cui et al., "PET image denoising using unsupervised deep learning," Eur. J. Nucl. Med. Mol. Imaging, vol. 46, no. 13, pp. 2780–2789, 2019, doi: 10.1007/s00259-019-04468-4.

[88] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised

learning," Mach. Learn., vol. 109, no. 2, pp. 373–440, 2020, doi: 10.1007/s10994-019-05855-6.

[89] C. Yu, R. Han, M. Song, C. Liu, and C. I. Chang, "A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial-spectral fusion," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 13, pp. 2485–2501, 2020, doi: 10.1109/JSTARS.2020.2983224.

[90] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology https://doi.org/10.1007/s13244-018-0639-9," Springer, vol. 195, pp. 21–30, 2018.

[91] M. I. Razzak, S. Naz, and A. Zaib, "Deep learning for medical image processing: Overview, challenges and the future," Lect. Notes Comput. Vis. Biomech., vol. 26, pp. 323–350, 2018, doi: 10.1007/978-3-319-65981-7_12.

[92] A. Patil and M. Rane, "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition," Smart Innov. Syst. Technol., vol. 195, pp. 21–30, 2021, doi: 10.1007/978-981-15-7078-0_3.

[93] H. Lin and M. Li, Introduction to Data Science, vol. 195. scientistcafe, 2021.

[94] Z. Hajduk, "Hardware implementation of hyperbolic tangent and sigmoid activation functions," Bull. Polish Acad. Sci. Tech. Sci., vol. 66, no. 5, pp. 563–577, 2018, doi: 10.24425/124272.

[95] Z. Li, S. H. Wang, R. R. Fan, G. Cao, Y. D. Zhang, and T. Guo, "Teeth category classification via seven-layer deep convolutional neural network with max pooling and global average pooling," Int. J. Imaging Syst. Technol., vol. 29, no. 4, pp. 577–583, 2019, doi: 10.1002/ima.22337.

[96] J. M. Valverde et al., "Transfer learning in magnetic resonance brain imaging: A systematic review," J. Imaging, vol. 7, no. 4, pp. 1–21, 2021, doi: 10.3390/jimaging7040066.

[97] E. Rezende, G. Ruppert, T. Carvalho, A. Theophilo, F. Ramos, and P.

de Geus, “Malicious Software Classification Using VGG16 Deep Neural Network’s Bottleneck Features,” Adv. Intell. Syst. Comput., vol. 738, pp. 51–59, 2018, doi: 10.1007/978-3-319-77028-4_9.

[98] H. Qassim, A. Verma, and D. Feinzimer, “Compressed residual-VGG16 CNN model for big data places image recognition,” IEEE, vol. 2018-Janua, pp. 169–175, 2018, doi: 10.1109/CCWC.2018.8301729.

[99] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[100] E. Mocsari and S. S. Stone, “Densely Connected Convolutional Networks Gao,” Comput. Vis. Found., vol. 39, no. 9, pp. 1442–1446, 2016.

[101] A. Sai Bharadwaj Reddy and D. Sujitha Juliet, “Transfer learning with RESNET-50 for malaria cell-image classification,” Proc. 2019 IEEE Int. Conf. Commun. Signal Process. ICCSP 2019, pp. 945–949, 2019, doi: 10.1109/ICCSP.2019.8697909.

[102] O. Wichrowska et al., “Learned optimizers that scale and generalize,” 34th Int. Conf. Mach. Learn. ICML 2017, vol. 8, no. 1987, pp. 5744–5753, 2017.

[103] K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks,” Springer Int. Publ. AG, vol. 9914 LNCS, p. V, 2016, doi: 10.1007/978-3-319-46493-0.

[104] A. Rastogi, “RESNET50,” Medium, 14-Mar-2022. [Online]. Available: https://blog.devgenius.io/resnet50-6b42934db431. [Accessed: 02-Aug-2022]

[105] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” Int. J. Comput. Vis., vol. 128, no. 2, pp. 336–359, 2019, doi: 10.1007/s11263-019-01228-7.

[106] S. K. Vuppala, M. Behera, H. Jack, and N. Bussa, “Explainable Deep Learning Methods for Medical Imaging Applications,” 2020 IEEE 5th Int. Conf. Comput. Commun. Autom. ICCCA 2020, pp. 334–339, 2020, doi: 10.1109/ICCCA49541.2020.9250820.

References

[107] R. Shilpa Kamdi, "Image Segmentation and Region Growing Algorithm," Int. J. Comput. Technol. Electron. Eng., vol. 2, no. 1, pp. 103–107, 2012.

[108] T. Acharya and A. K. Ray, Image Processing: Principles and Applications. JOHN WILEY & SONS, MC., PUBLICATION, 2005.

[109] V. Kovalevsky, Modern Algorithms for Image Processing. Library of Congress, 2019.

[110] D. Oliva, M. A. Elaziz, and S. Hinojosa, Metaheuristic Algorithms for Image Segmentation : Theory and Applications. Springer Nature Switzerland, 2019.

[111] J. Hemanth and V. Emilia, Nature Inspired Optimization Techniques for Image Processing Applications. Springer, 2019.

[112] W. K. Pratt, DIGITAL IMAGE PROCESSING. John Wiley & Sons, Inc., 2001.

[113] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Inf. Process. Manag., vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.2009.03.002.

[114] Jun, M., Cheng, G., Yixin, W., Xingle, A., Jiantao, G., Ziqi, Y., Minqing, Z., Xin, L., Xueyuan, D., Shucheng, C., Hao, W., Sen, M., Xiaoyu, Y., Ziwei, N., Chen, L., Lu, T., Yuntao, Z., Qiongjie, Z., Guoqiang, D. and Jian, H., 2020. COVID-19 CT Lung and Infection Segmentation Dataset. [online] Zenodo. Available at: <https://zenodo.org/record/3757476#.Yyh6OHZByUn> [Accessed 7 April 2022].

[115] M. Akif Cifci, "Deep Learning Model for Diagnosis of Corona Virus Disease from CT Images," Int. J. Sci. Eng. Res., vol. 11, no. 4, pp. 273–278, 2020, [Online]. Available: http://www.ijser.org.

[116] Y. Pathak, P. K. Shukla, A. Tiwari, S. Stalin, and S. Singh, "Deep Transfer Learning Based Classification Model for COVID-19 Disease," IRBM, vol. 43, no. 2, pp. 87–92, 2022, doi: 10.1016/j.irbm.2020.05.003.

[117] N. Tsiknakis et al., "Interpretable artificial intelligence framework for COVID-19 screening on chest X-rays," Exp. Ther. Med., vol. 20, no. 2,

pp. 727–735, 2020, doi: 10.3892/etm.2020.8797.

[118] M. Rahimzadeh and A. Attar, “A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2,” Informatics Med. Unlocked, vol. 19, p. 100360, 2020, doi: 10.1016/j.imu.2020.100360.

[119] B. Abraham and M. S. Nair, “Computer-aided detection of COVID-19 from X-ray images using multi-CNN and Bayesnet classifier,” Biocybern. Biomed. Eng., vol. 40, no. 4, pp. 1436–1445, 2020, doi: 10.1016/j.bbe.2020.08.005.

[120] F. Sadik, A. G. Dastider, M. R. Subah, T. Mahmud, and S. A. Fattah, “A dual-stage deep convolutional neural network for automatic diagnosis of COVID-19 and pneumonia from chest CT images,” Comput. Biol. Med., vol. 149, no. March, p. 105806, 2022, doi: 10.1016/j.compbiomed.2022.105806.

[121] X. He et al., “Automated Model Design and Benchmarking of Deep Learning Models for COVID-19 Detection with Chest CT Scans,” 35th AAAI Conf. Artif. Intell. AAAI 2021, vol. 6A, pp. 1–13, 2021.

[122] Maftouni, M., Law, A.C, Shen, B., Zhou, Y., Yazdi, N., and Kong, Z.J. “A Robust Ensemble-Deep Learning Model for COVID-19 Diagnosis based on an Integrated CT Scan Images Database,” Proceedings of the 2021 Industrial and Systems Engineering Conference, Virtual Conference, May 22-25, 2021.

[123] Ma Jun, “COVID-19 CT Lung and Infection Segmentation Dataset”. Zenodo, Apr. 20, 2020. doi: 10.5281/zenodo.3757476.

المستخلص

تم تصنيف فايروس كورونا على انه جائحة من قبل منظمة الصحة العالمية حيث يتميز هذا الفايروس بسرعة انتشاره بصورة مباشرة او غير مباشرة بين الاشخاص. ن عملية اكتشاف الاصابة بهذا الفايروس في بداية انتشارة تتم عن طريق تحليل (RT-PCR)، ومن اهم مساويء هذا النوع من التحليل هي تاخر النتيجة حيث تصل الى اسبوعين تقريبا وكذلك قلة توفر المواد اللوجستية لهذا الفحص بالاضافة الى تباين خبرات الفيزيائيين، مما ادى الى عدم القدرة علىالسيطرة على هذه الجائحة. ان الهدف الاساسي من هذا البحث هو تحديد وتشخيص الاصابة في الرئة.

يمكن تقسيم النظام المقترح الى ثلاثة مراحل اساسية وهي: مرحلة عمليات المعالجة الي تتم على الصور والتي تشمل عدة خطوات منها تحويل نظام الصورة من (nii) الى النظام (png) وتليها خطوة استخلاص منطقة الرئة من الصور وذلك بعملية (Region of Interest Extraction) وبعدها خطوتي تغيير الحجم (resizing) و (Normalization)، اما المرحلة التالية هي مرحلة بناء وتدريب النظام، وفي هذه المرحلة تم استخدام طريقة (Transfer Learning) والتي هي جزء من مفهوم التعلم العميق (Deep Learning) حيث تم تدريب عدة معماريات منها (VGG16, Resnet50, Inception, and Xception) .

وبعد عملية التدريب تم الحصول على اعلى دقة من تدريب معارية (VGG16) لذلك تم التعديل على الشبكة للحصول على نتائج افضل.

ان عملية التعديل على الشبكة VGG16 تمت على الطبقات الثلاثة الاخيرة حيث تم استبدالها بطبقات (global average layer, Batch_Normalization layer, and dense layer with binary classification) .

اما المرحلة الاخيرة من النظام تقوم بتحديد منطقة الاصابة في الرئة حيث تتم بواسطة استخدام تقنية (GRAD-CAM)، حيث يتم تطبيق هذه التقنية على الطبقات الاخيرة من النظام وفق معادلات رياضية تساعد على تحديد منطقة الاصابة.

في هذا البحث تم استخدام اثنين من قواعد البيانات الاولى "Large COVID-19 CT scan slice dataset" وفي هذه القاعدة لم يتم تنفيذ عملية تشخيص الاصابة بشكل. ومن الجدير بالذكر فان اعلى دقة تم التوصل لها عند استخدام قاعدة البيانات المذكورة هي 99.7% وبمعامل خطا يصل الى 0.0162.

اما قاعدة البيانات الثانية "COVID-19 CT Lung and Infection Segmentation Dataset" فانها تحتوي على قالب لمنطقة الرئتين لذلك تم استخلاص صورة الرئتين بعملية (ROI) ليتم تدريب الشبكة على الصور المستخلصة وبالتالي فان عملية التشخيص كانت جيدة. حيث تم الوصول لاعلى دقة تصل الى 99.82% وبمعامل خطا 0.0044.