Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Science for Women
Department of Computer Science

Classification of Cardiomegaly Disease Using CNN-VGG19 Deep Learning Method

A Thesis

Submitted to the Council of College of Science for Women, the University of Babylon in a Partial Fulfilment of the Requirements for the Degree of Master in Sciences\ Computer Sciences

By

Dena Ahmed Mohammed Hussein

Supervised By

DR. Enas Hamood Al-Saadi

2023 A. D. **1445 A.H.**

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

﴿ نَرْفَعُ دَرَجَاتٍ مَنْ نَشَاءُ ۗ وَفَوْقَ كُلِّ ذِي عِلْمٍ عَلِيمٌ ﴾

صَدَقَ اللَّهُ العَظِيمُ

يوسف﴿ 76 ﴾

Supervisor Certification

I certify that the project entitled **"Classification of Cardiomegaly Disease Using CNN-VGG19 Deep Learning Method"** was prepared at the Department of Computer Sciences / College of Science for Women/ University of Babylon, by **(Dena Ahmed Mohammed Hussein)** as a partial fulfillment of the requirements for the degree of Master in Computer Science

Signature:
Name: Dr. Enas Hamood Al-Saad
Date: / /2023
Address: Department of Mathematical, College of Education for Pure Sciences, University of Babylon, Iraq

Head of the Department Certification

In view of the available recommendations, I forward the research entitled **"Classification of Cardiomegaly Disease Using CNN-VGG19 Deep Learning Method"** for debate by the examination committee.

Signature:
Name: Asst. Prof. Dr. Saif M. Kh. Al-Alak
Date: / / 2023
Address: University of Babylon/College of Science for Women

Dedication

I dedicate this thesis to the one who sought and struggled to enjoy comfort and contentment, to the one who was stingy with nothing in order to push me on the path of happiness, to the one who taught me to ascend the ladder of life with wisdom and patience...to the soul of my dear father.

To the spring that is not bored with giving, to the one who woven my happiness with threads woven from her heart, and who completed the path after my father, to the one who waited for this moment impatiently...my dear mother.

To my dear martyr brother, whom I wished would be present to witness my success and be proud of me. If it were not for you, I would not have reached where I am.

To someone who was keen to elevate my ambitions, knowledge, and culture, and be proud of my success and achievement...my husband and my life partner.

To whom their love flows in my veins, and their remembrance fills my heart, to the joy of my heart, the joy of my happiness, and the forgetfulness of my life... my sister .

To whom was the blessing of their presence in my life, and whose beautiful laughter filled my life... my beloved daughters.

To every person who supported me from my family and stood by my side, I am very grateful. You were the best support.

To those who walked together as we paved the way together toward success and creativity...my friends and colleagues.I dedicate this research.

Dena Ahmed

Acknowledgments

In the name of Allah most gracious and merciful. I am thankful to my creator who blessed me with the ability to complete this thesis and for everything.

I would like to express my deepest appreciation and gratitude to my supervisor ***Dr. Enas Hamood*** for guidance, valuable advice, helpful discussions, thoughtful comments, continuous encouragement, and support throughout the year of study and research at every step of this thesis.

Dena Ahmed

Abstract

Cardiomegaly can be viewed as an indication of an underlying medical condition rather than a separate ailment. Within the scope of diagnostic assessments, the identification of an enlarged heart within a chest X-ray image signifies the presence of cardiomegaly. The etiology of cardiomegaly encompasses a spectrum of causative factors, including hypertension, coronary artery disease, infections, hereditary abnormalities, and cardiomyopathies.

The objective of the thesis is to offer a deep-learning-based automated method (CNN-VGG19) for the classification of Cardiomegaly, using the chest X-ray (CXR) dataset.

There are four phases to the suggested system. Preprocessing comprises the following steps: resizing the chest x-ray to 128*128 and cropping it, optimizing the chest x-ray image in terms of brightness, contrast, and sharpness; and lastly, eliminating the noise.

To identify the regions of interest that correlate to cardiomegaly in the chest x-ray images, the second phase involves creating masks for cardiomegaly images. To improve the visibility and isolation of cardiomegaly areas, this procedure involves the use of threshold and filters, such as Gaussian and morphological filters.

The Third phase is the accurate extraction of pertinent features from the cardiomegaly images by utilizing the CNN-VGG19 model.

The fourth phase includes the classification of chest X-ray images, which is done in two distinct categories. The first category is binary classification, which divides images into normal and cardiomegaly classifications. The second category is multi-classification, which focuses on the stages of cardiomegaly, which include mild, moderate, and severe.

To assess the efficiency of the introduced model, evaluation criteria including precision, sensitivity, accuracy, and F-score were employed as essential metrics.

The results and the test of all the facets of the suggested system and all the phases were discussed in-depth and the results demonstrated great accuracy in the classification of cardiomegaly, for binary class and multiple categories and reached (99%) and (97%) respectively.

The proposed method is very fast in diagnosis, and the Chest x-ray image prediction time takes approximately (0.06) seconds for one image.

List of Contents

List of Figures

List of Algorithms

List of Abbreviations

Abbreviation	Meaning
Acc	Accuracy
ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CT	Computed tomography
CXR	chest X-rays
DL	Deep Learning
DNN	Deep neural network
FNNs	feedforward neural networks
FN	False Negative
FP	False Positive
GPU	Graphical Processing Unit
ML	Machine Learning
MLP	Multi-Layer Perceptron
Pre	Precision
ReLU	Rectified Linear Unit
RNN	Recurrent neural networks
ROI	Region Of Interest
Sen	Sensitivity
T	Threshold
TN	True Negative
TP	True Positive
VGG	Visual Geometry Group

List of Symbols

Symbol	Meaning
$\bullet$	Morphological Closing Operation
$\circ$	Morphological Opening Operation
$\oplus$	Dilation
$\ominus$	Erosion
μ	learning rate
$\odot$	Hadamard product

List of Table

Chapter One

General Introduction

CHAPTER ONE

1 GENERAL INTRODUCTION

1.1 Introduction

Cardiomegaly is a medical condition characterized by the enlargement of the heart's dimensions or the thickening of the walls of the ventricles [1]. Accurately determining the prevalence of cardiomegaly is challenging due to underdiagnosis, underreporting, and the absence of obvious symptoms [2] [3].

The diagnosis of cardiomegaly often requires a comprehensive range of diagnostic tests. Among the imaging techniques used are chest X-rays (CXR), Echocardiograms(Echo), Electrocardiograms (ECG/EKG), Magnetic resonance imaging(MRI),Computed tomography (CT) scans, and Cardiac catheterization [4].

Chest X-rays have gained popularity in assessing cardiomegaly due to their accessibility and cost-effectiveness [5][6].

When examining chest X-rays, radiologists often rely on the cardio-thoracic ratio (CTR) values obtained from anteroposterior chest radiographs to evaluate heart size [7].The CTR serves as a simple clinical indicator. Therefore, the CTR derived from a chest X-ray can assist in making clinical management decisions [8][9].

On the other hand, this manual technique necessitates a substantial amount of time, and the diagnostic outcomes rely on the reader's subjective interpretation and prior experience [10]. However, the manual delineation of organ boundaries and computation of the cardiothoracic ratio (CTR) are laborious processes that are prone to errors.

Early identification of cardiomegaly is critical in treatment and can reduce the risk of heart failure. Providing an automatic system for cardiomegaly detection is a difficult task. Because there are few indications of illness in the

early stages, automated and effective diagnostic algorithms based on chest X-ray images are desperately needed.

Deep neural networks were an important tool to solve many problems such as object detection, speech recognition, and image classification, and have seen significant and rapid growth in recent years. Convolution neural networks (CNNs) in particular, produced exceptional image classification results [11].

This thesis proposes an automated approach for early-stage detection of cardiomegaly from chest X-ray images depending on its features. This system should achieve a desirable level of accuracy in determining the heart's condition and distinguishing between a healthy heart and one with cardiomegaly, in addition to grading of cardiomegaly (mild, moderate, or severe).

The suggested system aims to minimize the number of parameters necessary in comparison to current methods, thereby enhancing the accuracy of the deep learning model while reducing processing power.

1.2 Literature Review

Several investigations have been proposed in the associated study involving the finding of cardiomegaly. These researches are mostly concerned with extracting attributes from chest X-ray pictures. The majority of the studies employed CNN models such as VGG Net and Res Net to extract the properties, while the others used image processing approaches. The majority of the strategies utilized to classify cardiomegaly were based on machine learning methodologies. The following is a synopsis of these studies:

- **In 2018, Que et al [12]:** The authors offer a deep learning-based automated approach for detecting cardiomegaly on CXR images. CardioXNet, the proposed algorithm, diagnoses cardiomegaly from CXR images, using the deep learning methods U-NET CTR. This approach achieved an accuracy of 93.75.

- **In 2020, Chamveha et al [13]:** The study offered an algorithm for estimating the (CTR) from CXR images. This approach obtains lung and heart masks from CXR images using a deep learning model based on U-Net and a VGG16 encoder and estimates CTR based on the extent of the masks obtained. This technique achieved accuracy rates of 67.1% and 69.8% on two distinct datasets, NIH and CheXpert. This proposed study needs many training samples to improve the accuracy.
- **In 2020, Arsalan et al [14]** implemented two CXR segmentation networks, X-RayNet-1 and X-RayNet-2, which were constructed with backward multilayer networks to achieve accurate segmentation performance with fewer trainable parameters. The proposed methods used semantic segmentation to aid in the diagnosis of relevant disorders. X-RayNet-1 outperformed X-RayNet-2 across all datasets with a 75% coefficient drop.
- **In 2020, Cardenas D et al [15]:** In this study they worked on checking five convolutional neural networks(EfficientNet,DenseNet, MobileNet, XceptionNet , InceptionNet) for automatically detecting cardiomegaly and training the models with a customized multilayer perceptron. Experiments used radiographs from two publicly available datasets, one of which was only used for external validation. The images were pre-processed so that only the chest cavity was visible. EfficientNetB2 had the best accuracy of 91%, followed by MobileNet at 90% and InceptionV3 at 88%, according to their finding. The main constraint of this investigation was the lack of clinical data from public sources.
- **In 2020,M Rathi et al [16]:** In the study conducted by the researchers, two approaches were employed for disease classification: the utilization of a standalone CNN model and a CNN model with transfer learning, specifically employing the VGG16 model with frozen layers to extract fundamental features. The first method achieved an accuracy of 55%,

while the second method, incorporating transfer learning, yielded an improved accuracy of 60%.

- **In 2020, Bouslama A et al [17]:** They investigated the end-to-end technique for detecting Cardiomegaly disease that employs Deep Convolutional Neural Network U-Net architecture. The learning step is accomplished using Chest X-ray images obtained from the ChestX-ray8 open-source medical dataset. To improve the contrast and brightness of the original images, the Adaptive Histogram Equalization (AHE) approach is used. These latter are compressed before going through a training step to reduce calculation time. The limitation of this study is that working with a large number of images in deep learning processes is difficult, especially if the images are preserved at their original size. Furthermore, the ChestX-ray8 dataset lacks image masks, resulting in a poorly supervised learning problem. This approach achieved a diagnostic accuracy of more than 93%.
- **In 2021, Bougias H et al [18]:** The study conducted by the authors aimed to investigate the feasibility of employing four distinct transfer learning methods, namely (InceptionV3, VGG16, VGG19, and SqueezeNet), for the identification of cardiomegaly presence in CXR images, and compare the diagnostic performance of these approaches using radiologists' reports. they used a dataset consisting of two thousand CXR images. The findings of the study demonstrated that VGG19 outperformed the other models, exhibiting an accuracy of 84.5% a sensitivity of 84%, and a specificity of 83%.
- **In 2021, YOO H et al [19]:** This work presents a cardiomegaly diagnosis assistance model based on CNN-ResNet and an explainable feature map. To begin configuring the model, the ResNet cardiomegaly diagnosis model is created, and a chest X-ray data set is used and learned. The resulting system demonstrated an accuracy rate of 80%.

- **In 2021, Mu Sook Lee et al [20]:** To evaluate the viability of explainable computer-aided identification of cardiomegaly in ordinary clinical practice, used segmentation-based techniques. PA CXRs from two public datasets, as well as in-house PA CXRs, were included in the training dataset .Two fully automated segmentation-based approaches for lung and heart segmentation were developed using cutting-edge DL models. The first method achieved an accuracy of 91%, whereas the second achieved an accuracy of 92%.
- **In 2023,Mohammed Innat et al [21]:** The authors presented in this work Cardio-XAttentionNet, a convolutional attention mapping deep learning model,to accurately detect and locate cardiomegaly, They use the advantages of several of the well-performed state-of-the-art ConvNet architectures (DenseNet-121, DenseNet-201, InceptionResNet_V2, and ResNet50) to evaluate the empirical evidence. They achieved that Cardio-XAttentionNet (CXA_Dense121) appears as the well-generalized model and provided outstanding results in addressing cardiomegaly class information and localization simultaneously, and achieved an accuracy of 85%.

Table (1.1) Summary of the Literature Review

#	Ref.	Techniqus	Dataset	Accuracy (%)
1	[12]	CardioXNet	ChestX-ray8 dataset	93.75
2	[13]	U-Net with VGG16 encoder	NIH dataset	67.1
			CheXpert dataset	69.8
3	[14]	X-RayNet_1 X-ray Net_2	JSRT Dataset, Montgomery County (MC) and Shenzhen X-Ray sets (SC) Dataset.	
4	[15]	EfficientNet	PadChest,OpenI-NIH dataset	91

		DenseNet		
		MobileNet		90
		XceptionNet		
		InceptionNet		88
5	[16]	CNN with vgg16	NIH dataset	60
		CNN		55
6	[17]	U-Net	ChestX-ray8 dataset	94
7	[18]	VGG19	Chest x-ray dataset	84,5
		VGG16		71%-81.3
		SqueezeNet		
		InceptionV3		
8	[19]	ResNet and Explainable Feature Map	NIH dataset	80
9	[20]	Standard U-Net and XLSor Model: Method1 segmentation	JSRT,Montgomery, in-house PA dataset	91
		Method1 segmentation		92
10	[21]	CXR-ResNet50		75
		CXA_IncepRes2		84.75
		CXR_DenseNet201		80.58
		CXA_Dense121		85

1.3 Problem Statement

The presence of mild to moderate heart enlargement often lacks evident symptoms, and numerous patients remain unaware of this condition. Without timely detection and intervention, this condition can progress and eventually result in cardiac arrest. On the other hand, existing methods for the detection and assessment of cardiomegaly depend on manual techniques, which require a lot of labor, time, and cost and necessitate the expertise of experienced radiologists. Therefore, it is essential to design an effective automatic detection system that is capable of rapidly performing diagnosis to detect cardiomegaly at early stages and other stages.

The Cardiomegaly dataset comprises several X-ray images that exhibit poor contrast and high noise, thereby adversely affecting the accuracy of the diagnostic process.

Previous techniques frequently employed manual approaches as traditional image processing techniques for feature extraction, but feature extraction in deep learning is carried out by the model itself.

1.4 Aims of the Thesis

1. Building a classification system for cardiomegaly disease by grading cardiomegaly severity automatically by utilizing the chest x-ray image based on deep learning techniques specifically a CNN-VGG19 model.
2. Identifying and implementing an effective technique for enhancing the quality of input images reducing noise artifacts and preserving important image details which is accepted as an important step in the preprocessing stage.
3. Automatic features extraction using a Deep Convolutional Neural Network based on the VGG19 architecture to capture intricate patterns and features from the chest X-ray images.

1.5 Contributions of the Thesis

1. This study seeks to advance a deep learning system utilizing a convolutional neural network in the context of a binary classification system, distinguishing between cardiomegaly and normal heart, to a multi-classification model is proposed, intending to effectively recognize and categorize each phase of cardiomegaly (mild, moderate, and severe) through the analysis of chest X-ray images.
2. By utilizing a reduced number of parameters compared to conventional deep learning models, particularly CNNs, the proposed approach achieves a remarkable prediction time of approximately 0.06 milliseconds per image. This computational efficiency enables real-time deployment of the system in clinical settings.
3. The performance evaluation of the proposed technique showcased superior results when compared with previous researchers; this indicates the effectiveness and competitiveness of the proposed approach in accurately detecting and classifying cardiomegaly disease.

1.6 Layout of the Thesis

This thesis has been restructured to contain four chapters in addition to chapter one, which is described briefly and is organized as follows:

- **Chapter Two,"Theoretical Background":** It describes image prepossessing approaches ,Segmentation ,Deep learning classification methods, and assessment criteria used in this thesis.
- **Chapter Three," Proposed System":** A description of the structure and operation of the classification algorithms utilized in the proposed diagnostic.
- **Chapter Four," Results and Discussion":** This chapter provided an overview of the many experiments that were conducted for each stage of the study. A description of the assessments and outcomes attained after

applying the advised strategy is also included.

- **Chapter Five," Conclusion and Future Work":** In this part, a synopsis of the research endeavor is provided. Additionally, this section outlines the upcoming tasks in this regard.

Chapter Two

Theoretical Background

CHAPTER TWO

2 THEORETICAL BACKGROUND

2.1 Introduction

This chapter provides an overview of the diverse methodologies and procedures utilized in the proposed cardiomegaly detection system. The chapter's structure is as follows: firstly, an introduction to cardiomegaly is presented, along with a delineation of the fundamental stages of medical image processing, and explains Otsu's technique, and Mathematical morphology for heart segmentation and mask construction, which facilitates subsequent feature extraction. Secondly, a comprehensive exposition of artificial neural networks is given, incorporating the presentation of equations governing convolutional neural networks. Furthermore, the chapter expounds upon the elucidation of performance measures, namely sensitivity, accuracy, precision, and F1 score.

2.2 Cardiomegaly Disease

Cardiomegaly, also known as cardiac enlargement, is a medical condition characterized by an increase in the size of the heart or thickening of the myocardium [22]. It can be caused by various underlying cardiac pathologies or factors that impose an excessive workload on the heart. Cardiomegaly can manifest as a response to physiological demands [23], such as intense physical exertion, or it may indicate the presence of a pathological cardiac condition [24].

The diagnosis of cardiomegaly requires a comprehensive assessment, involving a meticulous review of the patient's medical history, thorough clinical examinations, and the utilization of various diagnostic modalities, such as X-ray radiography, which is preferred due to its availability and cost-effectiveness [25]. Cardiomegaly can be classified into three stages:

- Mild Cardiomegaly.
- Moderate Cardiomegaly.
- Severe Cardiomegaly.

During the early stages of cardiomegaly, individuals are often oblivious to its presence as there are no signs of it, and it may resolve spontaneously or it may progress and worsen. Hence, it becomes necessary to identify an enlarged heart in its initial stages to facilitate appropriate strategies for optimizing patient care and enhancing clinical outcomes. Figure (2.1) depicts various levels of cardiomegaly In Figure (2.1) (a), the early stage of cardiomegaly is represented, wherein the heart exhibits a slight increase in size, readily discernible from the norm. Moving further, Figure (2.1) (b) portrays the intermediate stage of cardiomegaly, highlighting the progression of the disease. Finally, Figure (2.1) (c) portrays the advanced stage of cardiomegaly, demonstrating a substantial enlargement of the heart.

(a) mild **(b) moderate** **(c) severe**

Figure (2.1) The Progression of Cardiomegaly Stages.

2.3 Image Processing Methods

This section outlines the image processing methods employed in this thesis for pre-processing and enhancing the images used as inputs into the system.

2.3.1 Remove the Noise

During this procedure, the images undergo cleaning to remove standard recognized noises and imperfections. Image noise, characterized by variations in brightness and color intensity among the image's pixels, often occurs during the image capture or scanning process, induced by artificial sources like flashlights or natural factors like strong sunlight[26].To mitigate these imperfections, filters were applied, resulting in a considerable reduction in image noise, which proved to be highly effective. The presence of such noise can significantly impact the extraction and categorization of features from images. Among the most prevalent noises are the following:

1. **Salt and pepper noise:** commonly referred to as impulse noise, spike noise, random noise, or independent noise is recognized in the field. This type of noise is characterized by abrupt and rapid fluctuations in the visual signal, resulting in the appearance of isolated white and black pixels. Figure (2.2) presents a comparative illustration, wherein the left side showcases of normal CXR image of the heart, while the right side depicts an identical image subjected to the introduction of salt and pepper noise [27].

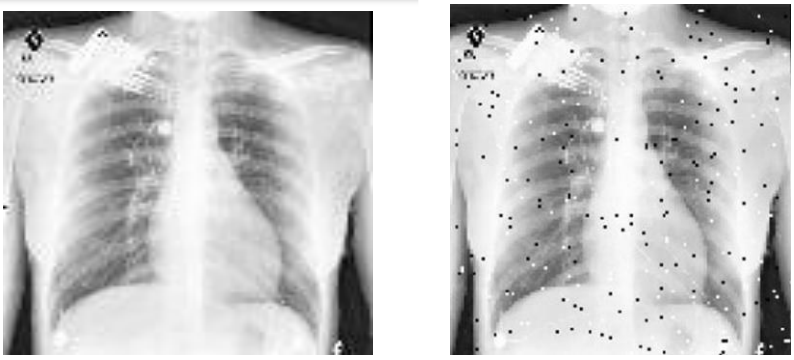


Figure (2.2) The Presence of Salt and Pepper Noise within the Image.

2. **Gaussian Noise:** The alternative nomenclature for normal noise or white noise is Gaussian noise, owing to its density resembling that of similar to the normal distribution (Gaussian distribution). In practice, inadequate lighting conditions often contribute to the presence of this type of noise. Figure (2.3) visually represents the impact of Gaussian noise, where the left side portrays a standard CXR image while the right side displays the same image contaminated with Gaussian noise [28].

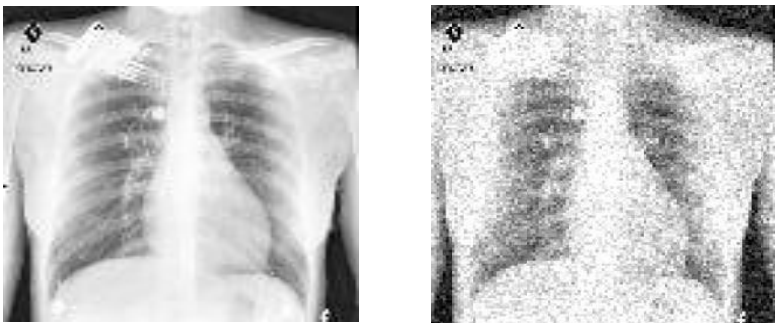


Figure (2.3) Displays the presence of Gaussian noise within the image.

2.3.2 Image noise reduction filters:

To enhance image quality and mitigate the adverse effects of noise, the application of filters, also known as kernels, becomes necessary. Filters consist of

small matrices that are multiplied with the image in order to diminish noise. Among the types of filters used are:

Median filter: The median filter, which as its name implies replaces the value of the pixel with the average intensity levels close by, is the most well-known filter for demand statistics in equation (2.1):

$$\begin{cases} x_{(n+1)/2} & \text{if n is odd} \\ \dfrac{x_{n/2}+x_{(n/2)+1}}{2} & \text{if n is even} \end{cases} \quad (2.1)$$

Let's denote a set of values as $\{x_1, x_2, x_3, ..., xn\}$, where n is the number of values. Essentially, it functions as a nonlinear spatial filter. The pixels contained in the region of the picture that the filter is applied to are ordered or arranged to produce the response. The margins of the image are preserved but salt and pepper noise is reduced with the medium filter. This filter replaces each pixel's value with the average of the values of the neighboring pixels by sliding a window of a specific size across each pixel of the picture. A set of values has a median when half of the values are greater than or equal to and half are less than that value. We first sort the nearby pixel values, choose the median, and then assign this value to the center responsive pixel in the filtered image [29].

Gaussian filter

The Gaussian filter is a popular image-processing filter that is used to smooth or blur images [30], because the filter's kernel is based on the Gaussian function, it is named after the Gaussian distribution or bell curve. The Gaussian filters

work by convolving each pixel in image with a Gaussian distribution kernel. The kernel is a tiny matrix or window of values centered on the pixel being processed, often with odd dimensions [31].

Upon application of the filter, each pixel's value is multiplied by the corresponding kernel value, and the resultant products are aggregated. This procedure is iteratively conducted for every pixel in the image, generating a new image wherein each pixel represents a weighted average of its neighboring pixels. The degree of blurring or smoothing achieved through the Gaussian filter hinges on both the kernel size and the standard deviation (sigma) parameter. A larger kernel and a higher sigma value yield a more pronounced blur, whereas a smaller kernel and a lower sigma value produce a more subtle effect [32].

The mathematical foundations of the Gaussian filter are elucidated in Equation (2.2) [33]

$$\boldsymbol{G(x,y) = (1/(2\pi\sigma^2)) * e\text{^}(-(x^2+y^2)/(2\sigma^2))} \qquad \textbf{(2.2)}$$

G(x, y) represents the Gaussian function, σ which represents the standard deviation or spread of the distribution, and (x, y) are the coordinates in the (x and y) directions respectively, (e) represents the base of the natural logarithm.

2.4 Threshold-based Segmentation.

One of the main and simplest methods to segment the image (separating the objects from the background) is using the threshold. By choosing a threshold value, the original grayscale image that contains a certain number of brightness levels is converted into a black-and-white image, which results in a binary image b(x, y) containing only two values (1, 0) [34].

It is significant to choose an acceptable gray-level threshold for separating the object from the background. The Otsu thresholding method (Minimum variance inside segments) is a segmentation technique that is an automatic threshold determination area dependent and converts the gray value from less than Threshold to 0 and the gray value from greater than Threshold to 255. Otsu's method is illustrated in Equation (2.3)

$$\sigma w\,2 \;=\; wc1\,(t) * \sigma c1\,2 \;+\; wc2\,(t) * \sigma c2\,2 \qquad (2.3)$$

This method has fast and simple processing speed characteristics because its main objective is to reduce the average error in appointing pixels to categories. It depends on the probability density function of the gray levels for each category. Otsu technique intends to optimize the variance inside categories. The main principle is that a good thresholded category should have distinct grey-level values. On the contrary, a threshold providing the biggest difference between categories in terms of grey-level values will be optimum [35].

2.5 The Morphological operators

Mathematical morphology constitutes a versatile technique employed for the characterization of area forms, encompassing boundaries, convex objects, and image component extraction. The essence of morphological operations lies in the utilization of a structural element, often referred to as the kernel, which is applied to the input image, resulting in an output image of equal dimensions. The determination of each pixel's value in the output image involves a comparison of the corresponding pixel with its neighboring elements in the input image.

Within the domain of mathematical morphology, various image-processing operations hold paramount importance, including erosion, dilatation, opening, and closing. These morphological processes necessitate two distinct types of

data: the input image, which may assume a binary or grayscale form depending on the specific procedure, and the structuring element, which rigorously defines the operation's influence on the image [36].

The structuring element is most often referred to as the kernel, though this term is intended for particular objects used during convolution operations. It comprises a pattern denoted by coordinates representing multiple points centered on a specific origin, effectively described as a small image [37]. Figure (2.4) illustrates an array of structural elements.

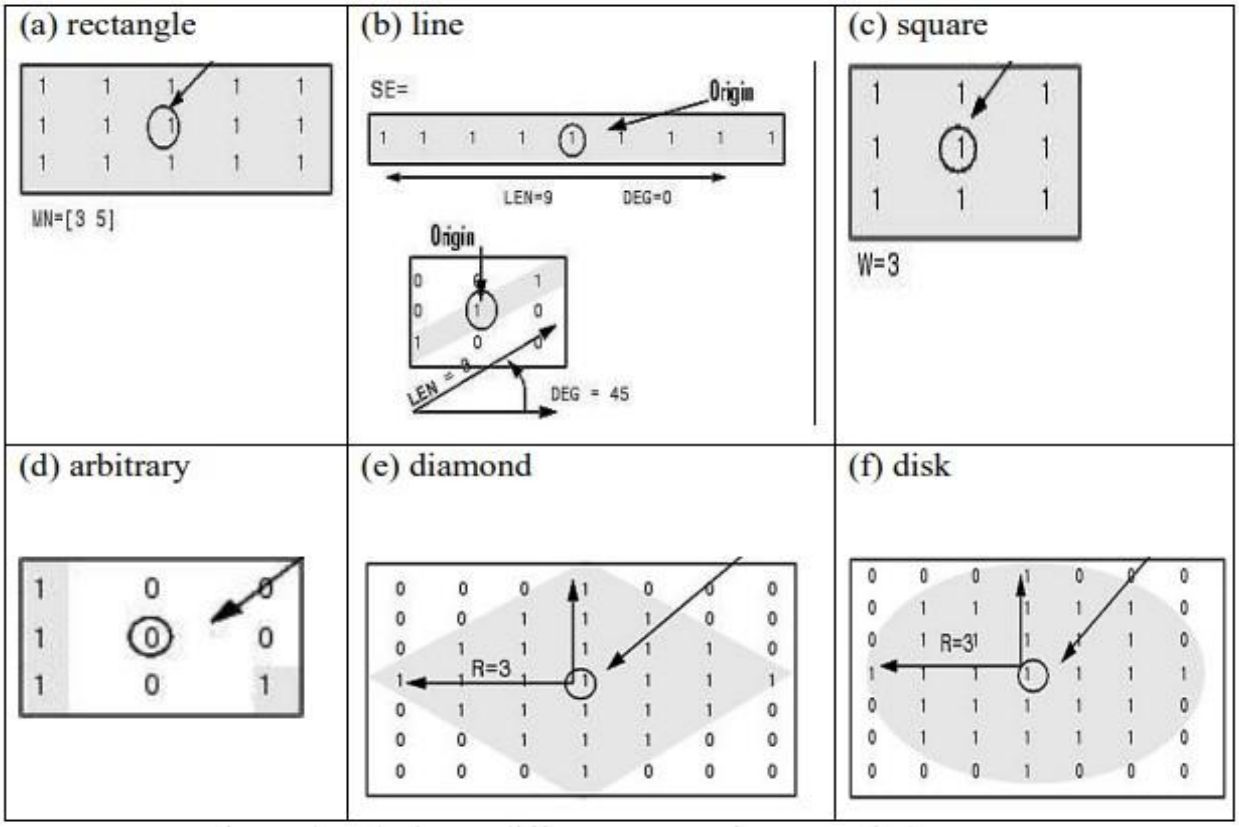


Figure (2.4) Shows different types of structural elements [38].

2.5.1 The Dilation and Erosion

Mathematical morphology emerges as a foundational technique in low-level image processing. Many subsequent morphological transformations, including

opening and closing operations, stem from the initial morphological processes of erosion and dilation. Dilation, an operation aimed at augmenting the size of objects, achieves this by appending pixels to both their inner and outer boundaries. Consequently, it bridges gaps within a region and mitigates minor protrusions that lie within the boundaries of the region.In the context of grayscale images, dilation enhances the brightness of objects by selecting the highest intensity value in the neighborhood as the kernel traverses the image pixels. Conversely, erosion, when applied to grayscale or binary images, results in the contraction of regions, reducing the extent of pixels at both their inner and outer boundaries. This process accentuates gaps within a single zone, consequently increasing the distance between distinct components. It is beneficial to reducing minor noise present in the image and eliminating small protrusions at the peripheries of objects [39]. Following the elucidation of dilation and erosion, the next step is the mathematical description of dilation and erosion [40].

The following are the steps in the mathematical description of dilation:

1. If any of the structural elements' pixels equal zero, the image's associated pixels are ignored.
2. If any of the pixels in the structural elements equal one, at least one of the related image pixels must also equal one.

The following are the steps in the mathematical description of erosion:

1. If any of the structural elements' pixels equal zero, the image's associated pixels are ignored.
2. If any of the structural element's pixels are equal to one, then each of the related

3. Image pixels must likewise be equal to one. If at least one of the structural element's pixels falls over a pixel with the value zero in the image, the pixel (origin) in the image must be altered to zero.

Equation (2.4) represents the operator for dilation (increase) [41]:

$$A \oplus B = \{X|(B^{\wedge})_x \cap A \neq \Theta \quad \} \tag{2.4}$$

A describes a collection of pixels in the input image that belong to an item. It is checked by structural element pixels, ($B^{\wedge)}$ is for the collection B supplement, and (B_x) is a translation of B derived from x. As a result, the dilation of A by B is a collection of all pixels of (x) such that the junction of (B_x) with A is not empty, as illustrated in Figure (2.5).

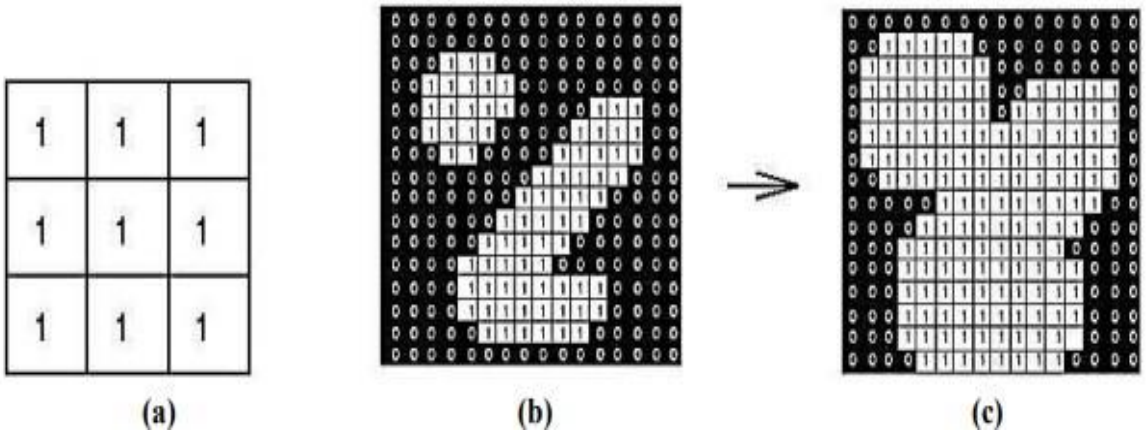


Figure (2.5) Explains the process of dilation :(a) structue element (b) input image (c) the output of dilation.

Equation (2.5) represents the operator for erosion (reduction):

$$A \ominus B = \{X|(B)_x \cap A^c \neq \emptyset \quad \} \tag{2.5}$$

Where A defines a group of pixels in the input image that belongs to an object, it is verified by structural element pixels B, A^c for the collection A's supplement, and (B_x) is a translation of B that originated from x. As a result, erosion of A by B is a set of all x pixels; hence Bx is a subset of A, as shown in Figure (2.6).

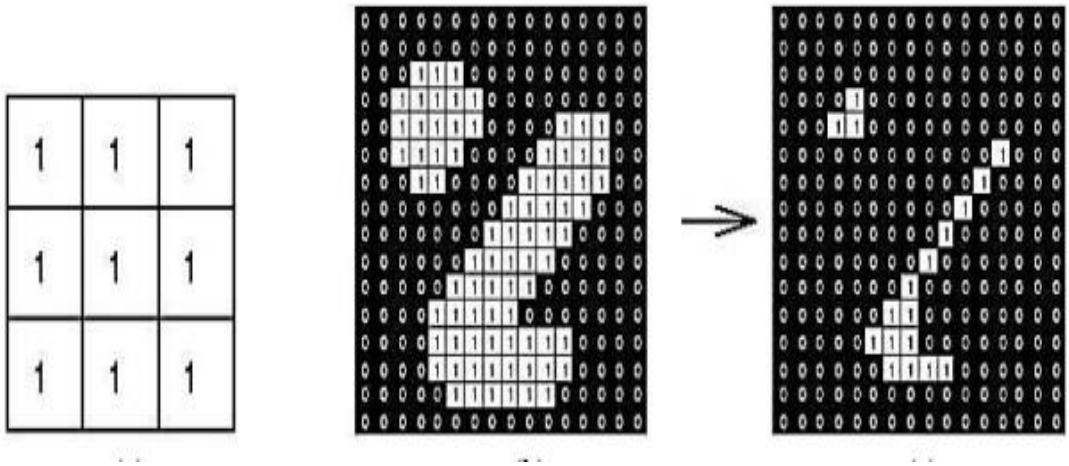


Figure (2.6) Depicts the erosion process: (a) the structural element (b) the input image (c) erosion result[42].

2.5.2 The Opening and Closing Operation

The two primary procedures, namely erosion, and dilation, are typically executed sequentially, often employing a single structuring element. The closing and opening operations operate in a complementary manner. As seen in the following Equation (2.6), the operation of opening involves a process of erosion followed by a phase of dilation.

$$A^{\circ}B = (A \ominus B) \oplus B \qquad (2.6)$$

Where ***A*** represents an input image and ***B*** represents a structural element. The opening technique has the advantage of separating and removing unwanted little things from the front of the image.

Whereas the action of closure comprises a dilatation and erosion process, it entirely reverses the process of opening, as seen in the following Equation (2.7).

$$A \bullet B = (A \oplus B) \ominus B \qquad (2.7)$$

Where ***A*** is the input image, and ***B*** is the structuring element, the basic usage of the Process of closing is to smooth the objects in the input image and to remove or decrease small gaps in the front.

2.6 Machine Learning Models

It is one of the Artificial Intelligence sections that depend on computer science, statistics, and mathematics.The basic goal of Machine Learning and statistical modelling is to enable computer programs to learn from data, and then make appropriate decisions based on the model that has been learned by a prior experience or prior skills. The machine learns directly from the fundamental input data structure and becomes more intelligent [43].

2. 6 .1 General Introduction to Machine Learning

Machine learning methodologies can be broadly categorized into three distinct classes: supervised learning, unsupervised learning, and reinforcement learning. Figure (2.7) illustrates a network of machine-learning techniques.

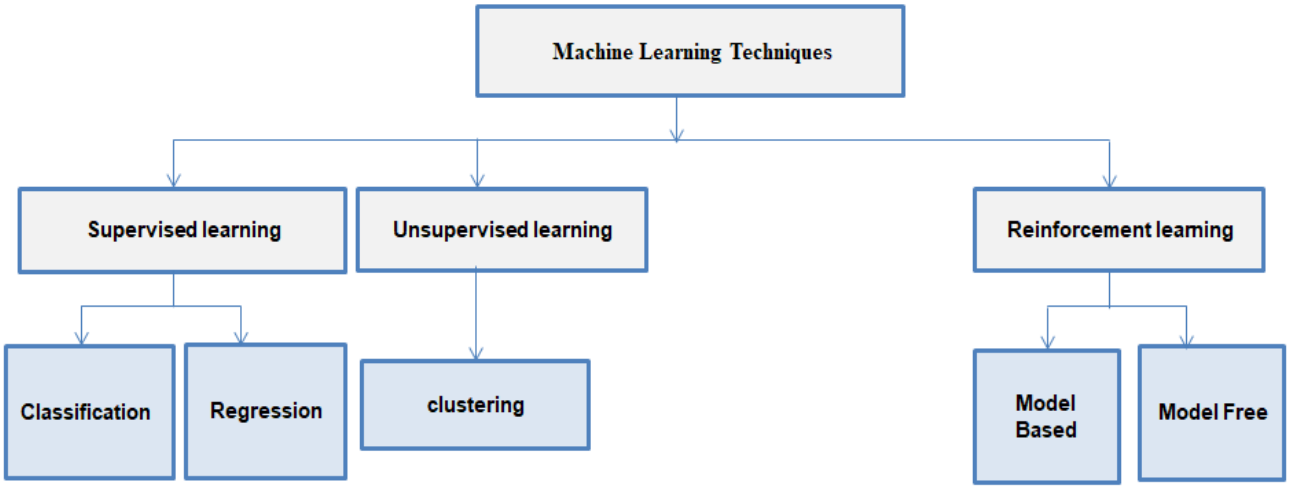


Figure (2.7) Illustrates the machine learning Techniques.

2.6.1.1 Supervised Learning

Supervised learning involves training models using input data (X) and corresponding output data (y), to predict future outputs for unseen input. This learning paradigm relies on the supervision or monitoring of the learning process through the availability of labeled ground truth output data. The algorithm iteratively performs predictions based on these labels until it achieves a satisfactory level of performance. In the context of supervised learning, generic algorithms can be broadly classified into two categories [44]:

- **Classification:** This category encompasses techniques aimed at assigning data points to predefined categories based on specific properties of the data. Examples of classification algorithms include Neural Networks, Support Vector Machines, k-nearest-neighbors, Random Forest, and Naive Bayes.
- **Regression:** Regression techniques are employed to forecast continuous values, also known as actual values. These approaches, such as Decision Trees, Linear

Regression, and Ensemble methods, operate under the guidance of target values (y) provided in the training dataset. The supervised nature of regression algorithms allows them to be directed towards the correct solution during the training phase.

In this thesis, supervised learning methodologies will be employed, specifically utilizing a deep learning technique known as Convolutional Neural Networks, for the purpose of categorizing and diagnosing cardiomegaly of CXR images.

2.6.1.2 Unsupervised learning

The clustering problem algorithm, k-mean, the A priory algorithm for association rule learning, and the dimensional reduction algorithm are examples of unsupervised learning algorithms. In this paradigm, models are trained solely on the input data they receive, without the provision of ground truth labels during the training phase. In contrast, supervised learning involves the partitioning of input data into subsets, each characterized by a single attribute [45].

2.6.1.3 Reinforcement learning

Reinforcement learning is defined as action-based learning. An agent who, in a given situation, takes steps to optimize rewards. The agents are expected to determine the fastest possible route to obtain the reward. This type of algorithm does not require any data for learning. In reinforcement learning, the input required is instead a function that can calculate the reward. One of the most important reinforcement learning applications is the path exploration function, which is used

in computer vision to locate a particular room or location. In video games, another application is to find the right movements to gain the game [46].

2.6.2 Artificial Neural Networks

Artificial Neural Networks are computational methods that have been designed through massive processing of basic units that are densely interconnected in groups to simulate how the human brain works on a specific task. These units are the computational elements that are known as (nodes and neurons) with a neuronal property. Experimental data and practical knowledge are processed to make it accessible to the user by changing weights [47]. There are several various types of ANN, such as Multi-Layer Perception (MLP), recurrent neural networks (RNNs), and Convolutional Neural Networks (CNNs), which are the most widely used in recent years and are illustrated in this section.

2.6.2.1 Multi-Layer Perceptron (MLP) Structure

Multi-Layer Perceptron is an ANN feedforward structure. The main components of an MLP contain a series of computational units (nodes) with trainable weights and biases at several layers (one input layer, one or more hidden layers, and one output layer). Thus, each node (neuron) in these layers connects with all nodes in the next layer, as shown in Figure (2.8). The MPL will map an input vector (X) to the desired output vector(Y) via a feedforward approach, as it is trained in two phases: backpropagation and forward-propagation. By the former, the network's biases and weights are obtained to approximate the complicated relationship between the desired output and the input vector, and employing a certain cost function, the training purpose is to decrease the difference between the desired outputs and the predictions. This method is renewed until this difference becomes zero or near the minimum [48].

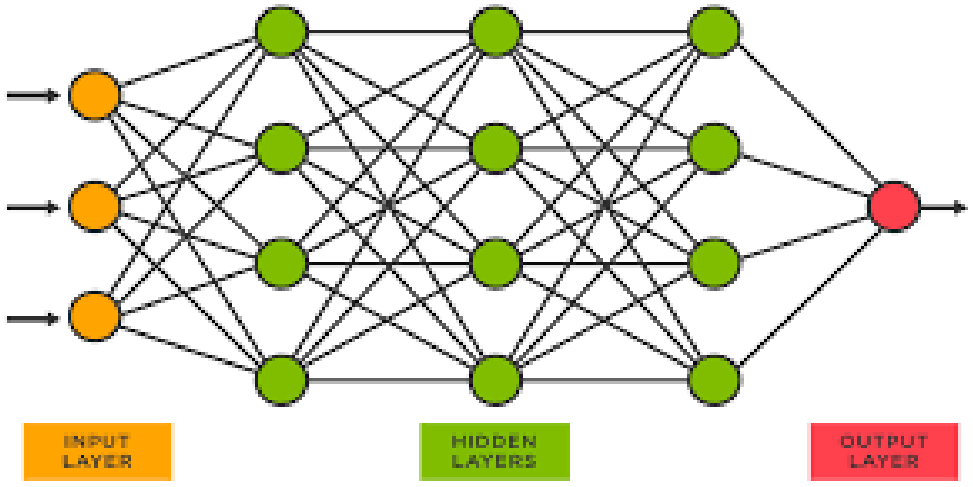


Figure (2.8) Shows the basic architecture of an MLP

2.6.2.2 The Activation Function

Neural networks rely on the transmission of information to acquire knowledge and address complex tasks, There are various kinds of activation functions, some linear and others nonlinear, and the output value typically ranges between [0,1] for unipolar activation function or [-1,1] for bipolar activation function. In the hidden layers, the sigmoid function as illustrated in Equation (2.8), and the Rectified Linear Unit (ReLU) function, as shown in Equation (2.9), are the most widely used activation functions in these layers, as shown in Figure (2.9)[49]

$$\boldsymbol{A = 1/(1 + e\text{^}(-x))} \quad \textbf{(2.8)}$$

where A is the output value , and x is the input value.

$$\boldsymbol{g(z) = max(z, 0)} \quad \textbf{(2.9)}$$

Where (z) is the input value.

Sigmoid:is a non-linear activation function. It is mostly used in models where we need to predict the probability of something. The range of sigmoid is from 0 to 1**.**

ReLU: is a non-linear function that replaces all image pixels with zero values in the activation map when their values are negative. This has the benefit of speeding up calculations and lowering the likelihood of overfitting.

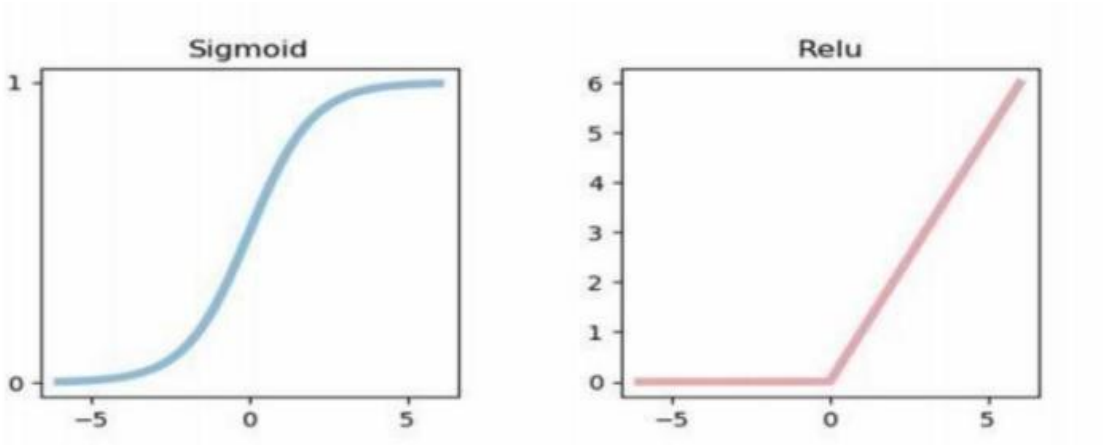


Figure (2.9) Shows the activation functions [50].

The output layer of a neural network may employ the soft max function as an activation function, which is widely utilized to measure probabilities and execute multi-class classifications because soft max ranges are between (0 and 1) and they have a value of 1 if all the features are used according to the following Equation (2.10)[51]

$$\boldsymbol{softmax}(\boldsymbol{x_i}) = \frac{\boldsymbol{e^{z_j}}}{\sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} \boldsymbol{e^{z_k}}} \quad \boldsymbol{for} \;\; \boldsymbol{j = 1 \ldots K} \qquad \textbf{(2.10)}$$

Where $\boldsymbol{K}$ is the number of classes, $\boldsymbol{^{z}j}$ is the production corresponding to class $\boldsymbol{j}$

2.6.2.3 Loss Function

The evaluation of a neural network's capability to accurately predict output values for a given dataset is accomplished through the utilization of a cost function, commonly referred to as the loss function. Within the field of machine

learning, various loss functions are employed, among which are cross-entropy and mean squared error (MSE) [52]. The cross-entropy is the most specific loss function of classification since it determines the classification algorithms perform depending on the probability of the class falls within the range (0, 1) and is described as follows:

$$\boldsymbol{loss} = -\sum_{i=1}^{n} \boldsymbol{y_i} \log(\hat{\boldsymbol{y}}_i) \quad \textbf{(2.11)}$$

The symbol "n" represents the number of classes, $\boldsymbol{y}$ denotes the true value, and ($\hat{\boldsymbol{y}}$) represents the predicted value.The cross-entropy cost and the sigmoid function are called the binary cross-entropy loss and are used when there are only two classes of classifications (0, 1). In addition to the Softmax function, the cross-entropy cost is called categorical cross-entropy loss and is used where two or more label classes exist (labels are given in the one-hot expression). The Sparse Categorical Cross-entropy loss function is used if the labels are presented as integers [53].

2.6.3 Deep Learning Technique:

The Deep Learning Technique, a subset of machine learning, employs a multi-layer architectural design, typically relying on neural networks and non-linear transforms within its algorithms, to simulate intricate abstract concepts present in data. The ultimate objective of these techniques is to achieve a form of real artificial intelligence, wherein machines can learn and execute highly complex tasks akin to the functioning of the human brain, through layers of neurons. Computationally, construction and training, deep learning are intensive. Recent, developments in applications based on general-purpose graphic processing units (GPU s), the rapid advancement of machine learning algorithms, the processing of signals and information, and the growing amount of data that is used in

training are all reasons that have increased the popularity and success of deep learning [54]. Figure (2.10) visually presents the structural composition of a deep neural network, showcasing its multiple layers that confer its deep nature.

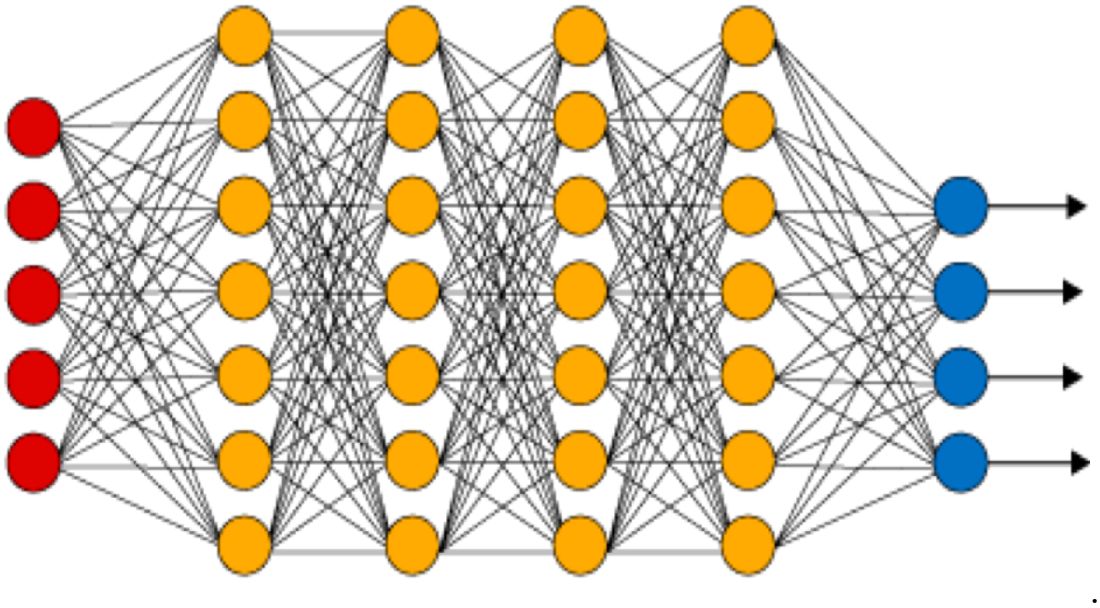


Figure (2.10) Depicted in Deep network architecture [55].

2.6.3.1 Deep Neural Networks (DNN)

The Deep Neural Network (DNN) is an artificial neural network comprising multiple hidden layers. MLP is the most general ANN architecture that is used for DNN. Neural networks are made up of interconnected neurons with several levels of connections. DNN needs long calculation periods to implement and many data feeds at training samples, so the number of weights in these networks will reach thousands or equal millions. There are different architectures for deep learning networks, the most important of which are: convolution Neural Network (CNN), which is used to classify images, and Recurrent Neural Network (RNN) used with texts and continuous data [47].

2.6.3.2 Convolution Neural Networks (CNN)

CNN a prevalent form of deep neural network finds extensive application in the realm of machine vision. It bears similarity to multi-layer perceptron except that the variation lies in its ability to combine several regionally interconnected layers used for feature extraction, accompanied by some fully connected layers used for classification [43]. It is an effective technique for image recognition, image visualization, and object identification. CNNs are widely adopted in AI, particularly in the domain of deep learning and medical image processing. Their proficiency stems from the ability to process large volumes of data without requiring manual feature extraction [51].

In contrast to feed-forward neural networks (FFNs), where input is in the form of a vector of pixel values, convolutional neural networks utilize input data in the format of a matrix of pixel values [56].

One of the most powerful features of convolution neural networks is their use of shared weights, a group of connections that share the same weights instead of using different weights for each connection. The other feature is the CNN has a local connection where each neuron does not contact all the neurons in the previous layer. Still, it only contacts a specific group of neurons to see if they contain the object's feature instead of contacting all cells. This produces strong responses to obtain local characteristics in an image Input (such as ridges, edges, curves). These two features exceedingly reduced the number of parameters in the network, and thus the training time will be reduced.

The standard architecture of a CNN comprises three distinct layers, as depicted in Figure (2.11), forming the foundation for constructing various convolutional neural network models:

A. Convolution layer.

B. Max pooling layer (or Sub Sampling layer).

C. Fully Connected layer (Classification layer).

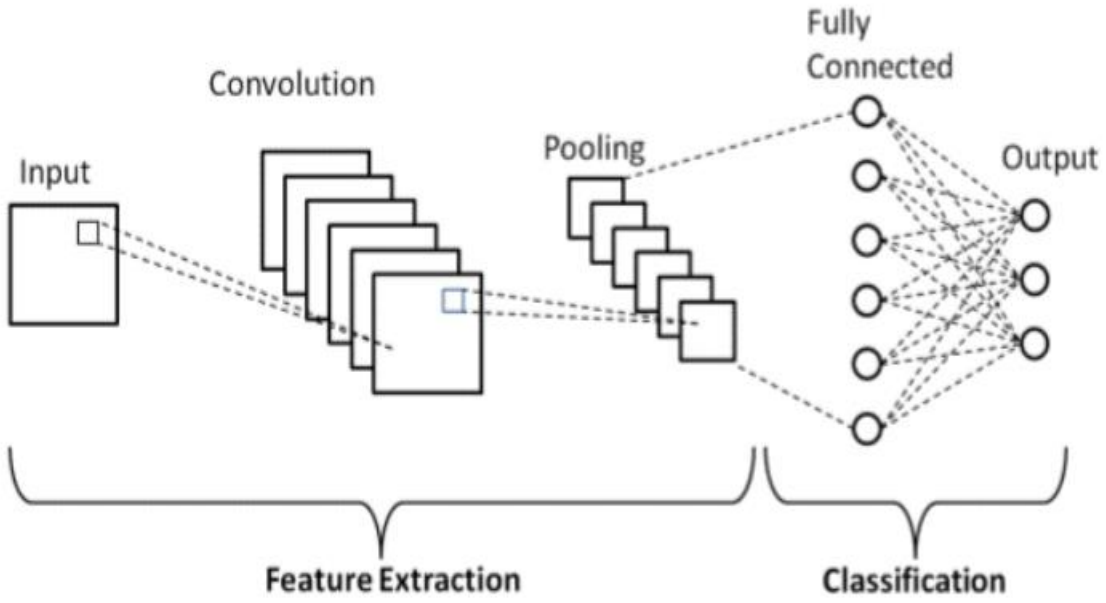


Figure (2.11) Depicts the structure of a CNN [57].

A. The Convolution Layer:

The initial layer serves as the foundational component in constructing a convolutional neural network (CNN) model. Its primary objective is to extract salient features from the original input image, accomplished through the mathematical process known as convolution. This layer comprises three matrices: the first represents the input image, which is transformed into a matrix either three-dimensional or gray scale two-dimensional, the second is called the filter matrix, also referred to as the feature detector or kernel, and the third matrix results from sliding the filter matrix horizontally and vertically across the input image, calculating the dot product, thereby generating the Feature Map or Activation Map, this is the convolution process as illustrated in Figure (2.12)[58].

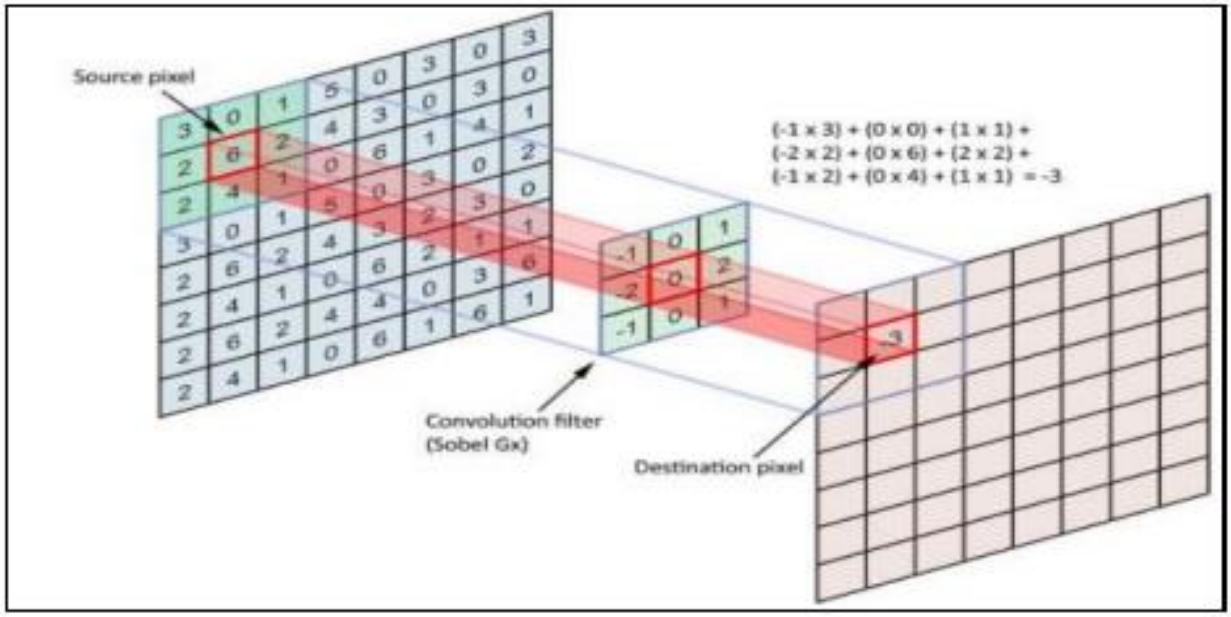


Figure (2.12) Shows image convolutional operations [57]

In a single convolutional layer, multiple convolution filters are employed to process a single input. The resulting activation maps are subsequently combined to yield the overall output of the convolutional layer, which serves as the input for the subsequent layer. Each value of the filter matrix is a weight that is given by default, and to impart diverse characteristics or features to each filter, these weight values must vary across different filters [59].The general technique for a convolutional layer in image processing is described by Equation (2.12) below:

$$Z_{ij} = (X * K)_{(i,j)} \sum_{i=1}^{K1} \sum_{j=1}^{K2} \sum_{k=1}^{C} X_{(i+l,j+a,b)} K_{(l,a,b)} \qquad (2.12)$$

In the above equation, the convolution process between the input X and the kernel weights K in convolution layer l to produce the feature map Z and K1 is the height of the kernel, K2 is the width of the kernel, C is the number of channels [60]. Following each convolution layer, the results are sent through a non-linear activation function (ReLU), as shown in equation (2.9).

The size of the feature map is determined by three parameters, which are as follows:

1. **Depth**: The number of filters used in the convolution process is represented by the depth. If the original image was complicated using three filters, the 'depth' of the activation maps will be three.
2. **Stride**: represents the number of pixels in the filter matrix that leaves during the convolution process. If a stride is equal to 1, the kernel will move by 1. If it equals 2, the kernel moves by 2. As the number of steps increases, the function maps diminish.
3. **Zero-padding**: surrounds the input matrix with zero values so that the number of rows and columns increases by 2; if p = 1. It is possible to make larger padding If p = 2, then the number of rows and columns increases by 4, and so on. This technique has been used to process gradual image fading across multiple layers in the CNN. Another problem is that the pixels on the outside of the image are not used in the convolution operation except very few times, which reduces the efficiency of the model [61].

   There are two types of convolution: a valid convolution, which is a Convolution that does not use padding, and therefore the size of the matrix, will gradually decrease; zero or same convolution, which is the convolution in

which the image size does not change before or after the convolution, and this type uses the technique of padding [62].

The following equation (2.13) illustrates how to calculate the feature map's size in the same convolution.

$$\mathbf{D_{out}} = \frac{D_{in} - D_k + 2p}{S} + 1 \quad \textbf{(2.13)}$$

Where D_{out} is the output size of the Activation map, D_{in} ,D_k represent the input size and kernel size, respectively, S is a stride, and p is the zero padding [57].

B. Max pooling layer

The process of reducing a collection of neighboring pixels to a singular pixel is referred to as pooling. Primarily applied subsequent to the convolutional layer, this technique effectively diminishes the dimensions of image activation maps. Spatial layers used various common types, such as average and maximum pooling, etc. as depicted in Figure (2.13). In the case of Max-Pooling, a defined spatial region, like a 2×2 window, is delineated, and the maximum value is selected from the rectified activation maps within each respective window. This operation leads to a downsizing of the activation map image from (4×4) to (2×2). Conversely, average pooling computes the mean value within each specified window and similarly contributes to the downsampled representation of the activation map image.

$$s_i = \max_{i \in R_j} h_i \quad \textbf{(2.14)}$$

Where $\boldsymbol{h}$ denotes a pixel in the window (or sub-region) $\boldsymbol{R_j}$ from the corrected Activation map [63].

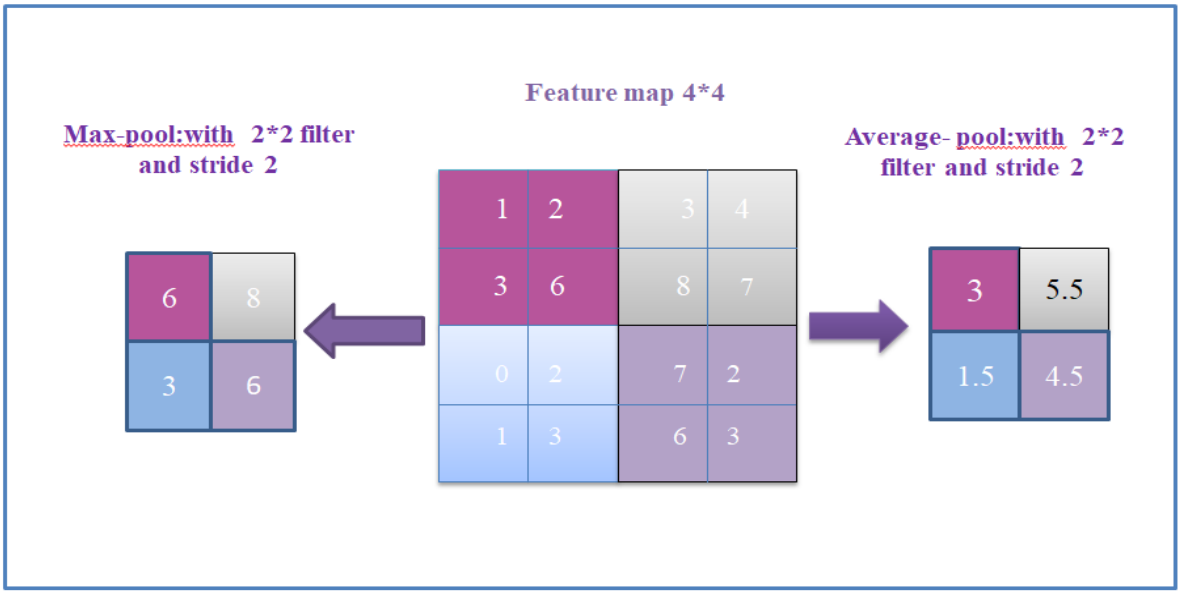


Figure (2.13) Max Pool with a kernel size of (2*2) and a stride of 2[64].

C. Fully Connected Layer (Classification layer)

This layer works in a similar way to the traditional MLP network that is explained in section (2.6.2.1) as each node has a complete connection to all the nodes in the following layer (Fully Connected or Dense layers). It's used as a classifier in the last layers of the CNN structure to assess the probability of the object in the image by using the **Sigmoid** or **Softmax** function in the output layer illustrated in section (2.6.2.2) **[62].**

2.6.3.2.1 Visual Geometry Group Model (VGG19)

VGG19 is deep convolutional neural network architecture, which is an extension of the original VGG16 model. It was developed by the Visual Geometry Group (VGG) at the University of Oxford and published as part of the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). VGG19 is renowned for its simplicity and effectiveness in image classification tasks [18].

The VGG in its name refers to the Visual Geometry Group, and the number 19 indicates the total number of layers in the network[64]. VGG19 consists of 19 layers, including 16 convolutional layers and 3 fully connected layers. The convolutional layers mostly use 3*3 filters, and the model's depth enables it to learn complex patterns and features from images [65].

Due to its straightforward architecture and remarkable performance, VGG19 has become a popular choice for transfer learning, where the pre-trained model on a large dataset like ImageNet can be fine-tuned for various other computer vision tasks, such as object detection, segmentation, and more. Figure (2.14) shows a network structure of VGG19, and Algorithm 2.1 shows the VGG19 works.

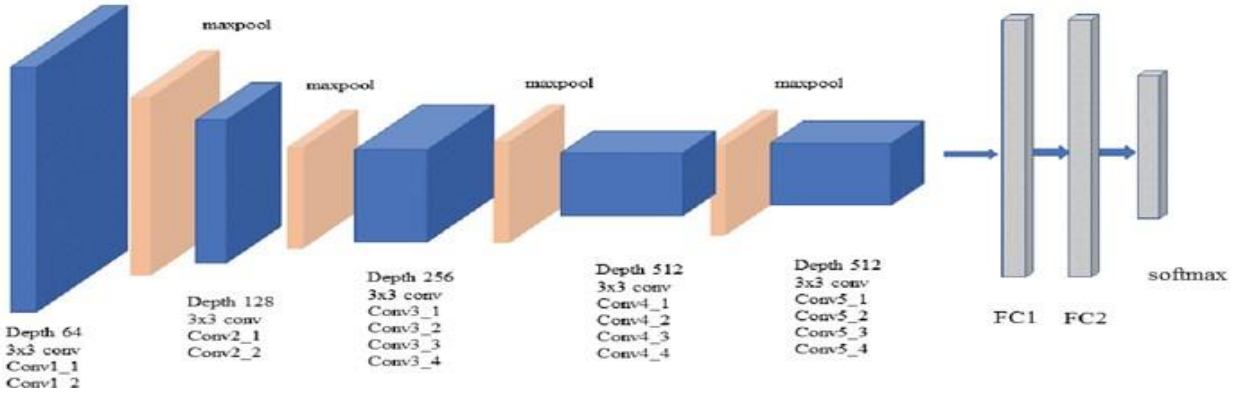


Figure (2.14) Illustrates the architecture of VGG19[67].

Algorithm (2.1): VGG19 for Feature Extraction	
Input: Masks CXR Images.	
Output: Most Significant Features	
Begin	
Step 1: Input Masks CXR Images.	**# 128*128**

Step 2: Build VGG19 model of five conv blocks

Step 3:Build block1

Add Block1_conv1 (Conv2D) **#(64, 3, 3, activation='relu')**

Add Block1_conv2 (Conv2D) **#(64, 3, 3, activation='relu')**

Add MaxPooling2D

Step 4: Build block2

Add Block2_conv1 (Conv2D) **#(128, 3, 3, activation='relu')**

Add Block2_conv2 (Conv2D) **#(128, 3, 3, activation='relu')**

Add MaxPooling2D

Step 5: Build block3

Add Block3_conv1 (Conv2D) **#(256, 3, 3, activation='relu')**

Add Block3_conv2 (Conv2D) **#(256, 3, 3, activation='relu')**

Add Block3_conv3 (Conv2D) **#(256, 3, 3, activation='relu')**

Add Block3_conv4 (Conv2D) **#(256, 3, 3, activation='relu')**

Add MaxPooling2D

Step 6: Build block4

Add Block4_conv1 (Conv2D) **#(512, 3, 3, activation='relu')**

Add Block4_conv2 (Conv2D) **#(512, 3, 3, activation='relu')**

Add Block4_conv3 (Conv2D) **#(512, 3, 3, activation='relu')**

Add Block4_conv4 (Conv2D) **#(512, 3, 3, activation='relu')**

Add MaxPooling2D

Step 7: Build block5

Add Block5_conv1 (Conv2D) #(**512, 3, 3, activation='relu**')

Add Block5_conv2 (Conv2D) #(**512, 3, 3, activation='relu**')

Add Block5_conv3 (Conv2D) #(**512, 3, 3, activation='relu**')

Add Block5_conv4 (Conv2D) #(**512, 3, 3, activation='relu**')

Add MaxPooling2D

Step 8: return feature extracted.

End.

2.6.3.3 Training the Convolution Neural Network

Back propagation is the fundamental component of ANN, and it is more complex in CNNs because of the numerous types of layers. This technique computes gradient descent for all network weights in two stages: the first is known as forward propagation because the training begins with the inputs of the first layer in the network and progresses to the last layer, where the error between the outputs and the desired value is calculated using the loss function shown in section (2.6.2.3).The second method is known as backward propagation, and it begins with the last layer in the hierarchy network to the first layer [62].

1. Forward Propagation:

To generate a prediction value, training samples are distributed throughout all network layers from input to output, using various equations that vary depending on the layer type.

- Using Equation (2.12), the convolution layer in forward propagation implements a convolution process between all of its inputs and filters. The output is then supplied to the ReLU function through equation (2.9).
- The Max pooling layer in the forward propagation is explained in the equation (2.14).
- A fully Connected Layer in the forward propagation works in a similar way to the MLP network that is illustrated in Equation (2.15)[67]

$$s = \bar{X} . \bar{W} = \sum^{n} x_i \, w_i + b \qquad \textbf{(2. 15)}$$

A neuron computes the dot multiplication between the input vector X = [x1, x2,..., xn] with their identical weights W = [w1,w2,...,wn] and then adds the bias value(b).Finally, the value (s) is transferred to the activation function.

The SoftMax and Sigmoid function shown in Equation (2.10),(2.8) was used to assess the probability of the object in this layer. After that, the error between the desired value and the output is calculated using the loss function that has been defined in Equation (2.11)[64].

2. Backward Propagation:

Backward propagation begins at the network's last layer (output) and proceeds to the first layer (input), operating as follows depending on the layer type:

- In backward propagation, the Fully Connected Layer functions similarly to the MLP network. In the output layer, the delta (error) is determined using the Equation (2.16).

$$\delta^L = \nabla_x{}^L \odot \acute{g}(z^L) \qquad (2.16)$$

Where δ^L is the matrix of all the neurons' deltas in the layer, L is the network's last layer, and ∇x^L is the gradient of the loss for x, where x is output of the activation function's output. And $\odot$ is the Hadamard product (element- wise product of matrices).

2.6.3.4 Regularization Techniques:

Several techniques have been proposed to prevent the overfitting problem known as regularization.

- **Dropout**

The dropout strategy involves randomly losing a set of neurons at each training cycle with a predefined probability value. This strategy, proposed in 2014 by Nit-ish Stradivari and Geoffrey Hinton, dramatically enhanced neural networks' performance in solving the overfitting problem, as demonstrated in Figure (2.16). The dropout layer equation can be executed as follows [51]:

$$r^l{}_i = \sim(\text{Bernoulli (p)}) \qquad (2.17)$$

$$\tilde{y}^l = r^l \odot y^l \qquad (2.18)$$

Where r^l represents a vector of independent Bernoulli random variables, each with a probability of being 1 of p, the output of l layer, and $\mathbf{y}^{\sim l}$ is lowered output that will be used as input x in Equation (2.15) for the next layer $l+1$ [64].

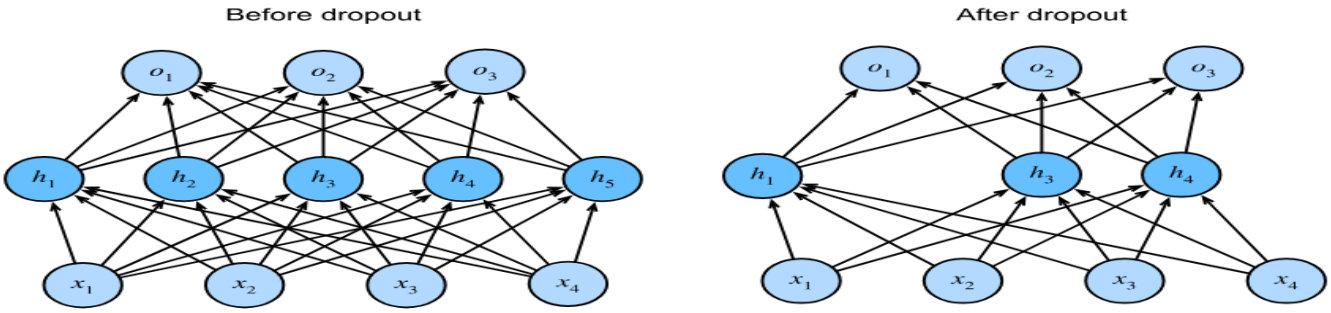


Figure (2.15) Demonstrates the dropout effect on a network.

- **Early stop**

The Early stop is an effective regularization approach used in deep learning network training to determine the appropriate number of epochs. When the loss increases in the validation sample and decreases in the training sample as the number of epochs increases, the training process must be stopped after a few subsequent periods using early stopping, as shown in Figure (2.16), because increasing the number of epochs leads to overfitting and decreasing the number of epochs leads to underfitting Early Stopping[66].

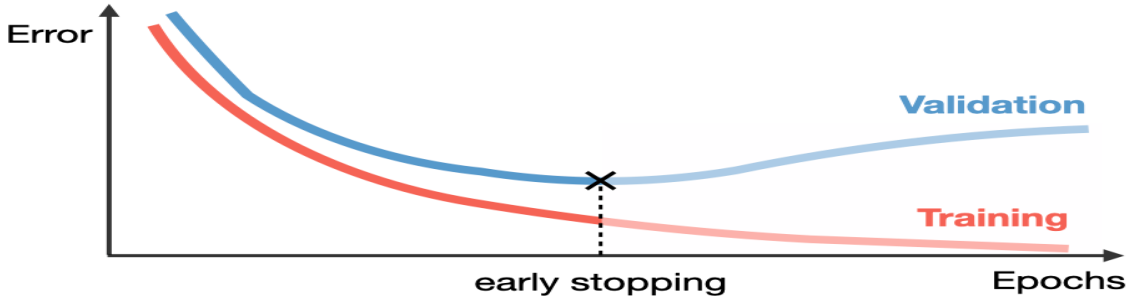


Figure (2.16) Shows the early stopping.

- **Optimization Algorithm**

The optimizer helps to reduce the output error of the loss function by changing the weights and bias values in the model. The Adam Optimizer is one of the most essential performance optimization tools used in deep neural networks. Adaptive Moment Estimation is what Adam means, a tool that calculates each parameter's adaptive learning rate and conserves the previous square gradients' average exponential decay rate while maintaining the prior scales' average exponential decline. Adam outperforms all other optimization techniques, making it suitable for the demand for rapid convergence and highly complicated neural networks, Algorithm 2.2 shows how Adam algorithm works **[67].**

Algorithm (2.2): Adam algorithm	
Input:	α: stepsize $\beta_1, \beta_2 \in [0,1)$: Exponational decay rates for the moment estimate $f(\theta)$: stochastic objective function with parameter θ θ_0: initial parameter vector
Output: optimized model parameters	
Begin **Step1:** $m_0 = 0$ (initialize 1st moment vector) **Step2:** $v_0 = 0$ (initialize 2nd moment vector) **Step3:** $t = 0$ (initialize timestep) **Step4: while** θ_t not converged **do**: **Step5:** $t = t + 1$ **Step6:** $g_t = \nabla_\theta f_t(\theta_{t-1})$ **Step7:** $m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ **Step8:** $v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ **Step9:** $\hat{m}_t = \frac{m_t}{(1-\beta_1^t)}$ **Step10:** $\hat{v}_t = \frac{v_t}{(1-\beta_2^t)}$ **Step11:** $\theta_t = \theta_{t-1} - \alpha \cdot \frac{\hat{m}_t}{(\sqrt{\hat{v}_t}+\epsilon)}$ (update parameter) **Step12: End while** **End**.	

2.7 Performance Measurements

Several criteria have been used to evaluate the classification algorithms' performance [68][69] :

1. **Accuracy** is the average correct prediction.Which is calculated by dividing the correct prediction by the total predictions.

$$\boldsymbol{Accuracy = TP + TN/TP + TN + FP + FN} \quad \textbf{(2.19)}$$

2. **Sensitivity:** is the rate of successfully identifying positive samples.

$$\boldsymbol{Sensitivity = TP/TP + FN} \quad \textbf{(2.20)}$$

3. **Precision:** The model's precision is determined by testing the true positive from the expected positives.

$$\boldsymbol{Precision\ (Pre) = TP/TP + FP.} \quad \textbf{(2.21)}$$

4. **F1-score:** shows a compound of precision and sensitivity for computing A balanced mean output.

$$\boldsymbol{F1 - score = 2 * (Precision * Sensitivity / Precision * Sensitivity)} \quad \textbf{(2.22)}$$

Table (2.1) shows confusion matrix of classifier system.

Table (2.1) The Confusion Matrix of the Classifier System

Confusion Matrix		**Predict**	
		Positive	**Negative**
Actual	**Positive**	**TP**	**FN**
	Negative	**FP**	**TN**

The prediction error is recorded by four parameters :

- **True Positive (TP)** is the positive states that are correctly labeled as positive states.
- **False Positive (FP)** denotes the negative states that are incorrectly labeled as positive states.
- **True Negative (TN)** represents the right classification of negative diagnosis.
- **False Negative (FN)** indicates the positive cases that are incorrectly classified as negative.

Chapter Three

Proposed System

CHAPTER THREE

PROPOSED SYSTEM

3.1 Introduction

This chapter provides an extensive exposition of the methodologies and techniques employed in the construction of the proposed system for cardiomegaly diagnosis using CXR images. The proposed system encompasses four distinct phases, The First phase, is the prepossessing phase which begins with resizing to a standardized dimension of (128×128) and cropping the CXR images, optimizing CXR images such as brightness, contrast, and sharpness. Finally, noise reduction through the application of a median filter on the processed images.

In the Second phase, Otsu's algorithm and mathematical morphology are used to segment a heart as a region of interest (ROI) detection algorithm.

The Third phase is the extraction of pertinent features from the cardiomegaly images.To facilitates this task, the system uses the (CNN-VGG19) model.Finally, the fourth phase encompasses the classification of CXR images, which is performed in two distinct categories.

The first category involves binary classification, where the images are classified into normal and cardiomegaly classes. The second category involves multi-classification for cardiomegaly stages, namely Mild, Moderate, and Severe.

This thesis utilized (70 %) of the available database for training the system, employing the four aforementioned steps. Subsequently, the remaining (30%) of the database was allocated for testing the trained model.

3.2 The Cardiomegaly Diagnosis System

The proposed system undergoes a series of sequential processes, encompassing image preprocessing operations specific to CXR, creating the mask, feature extraction, classification, and, finally, model performance evaluation employing multiple criteria. The proposed system is demonstrated in Figures (3.1) and (3.2)

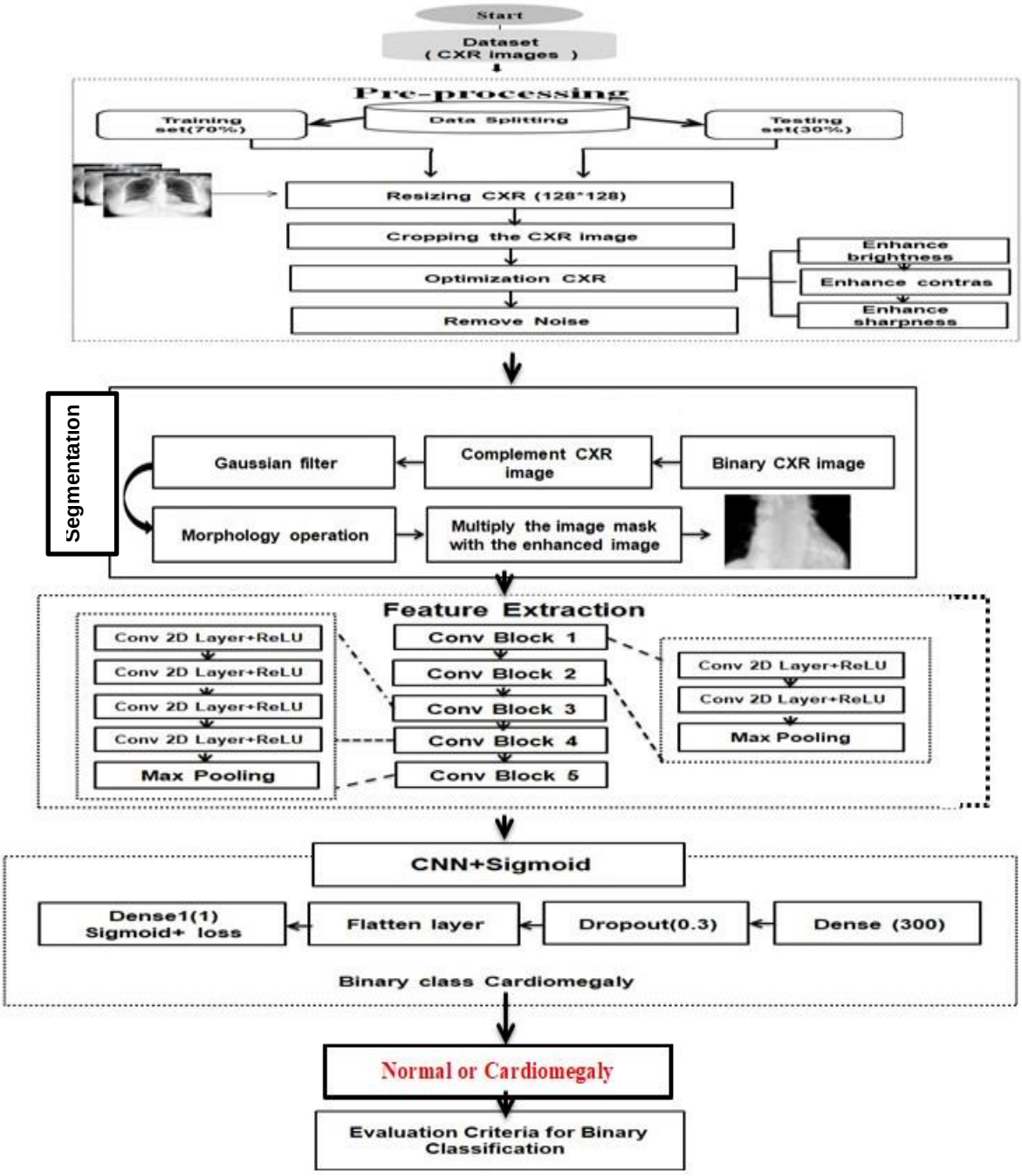


Figure (3.1) The block diagram of the Suggested System.

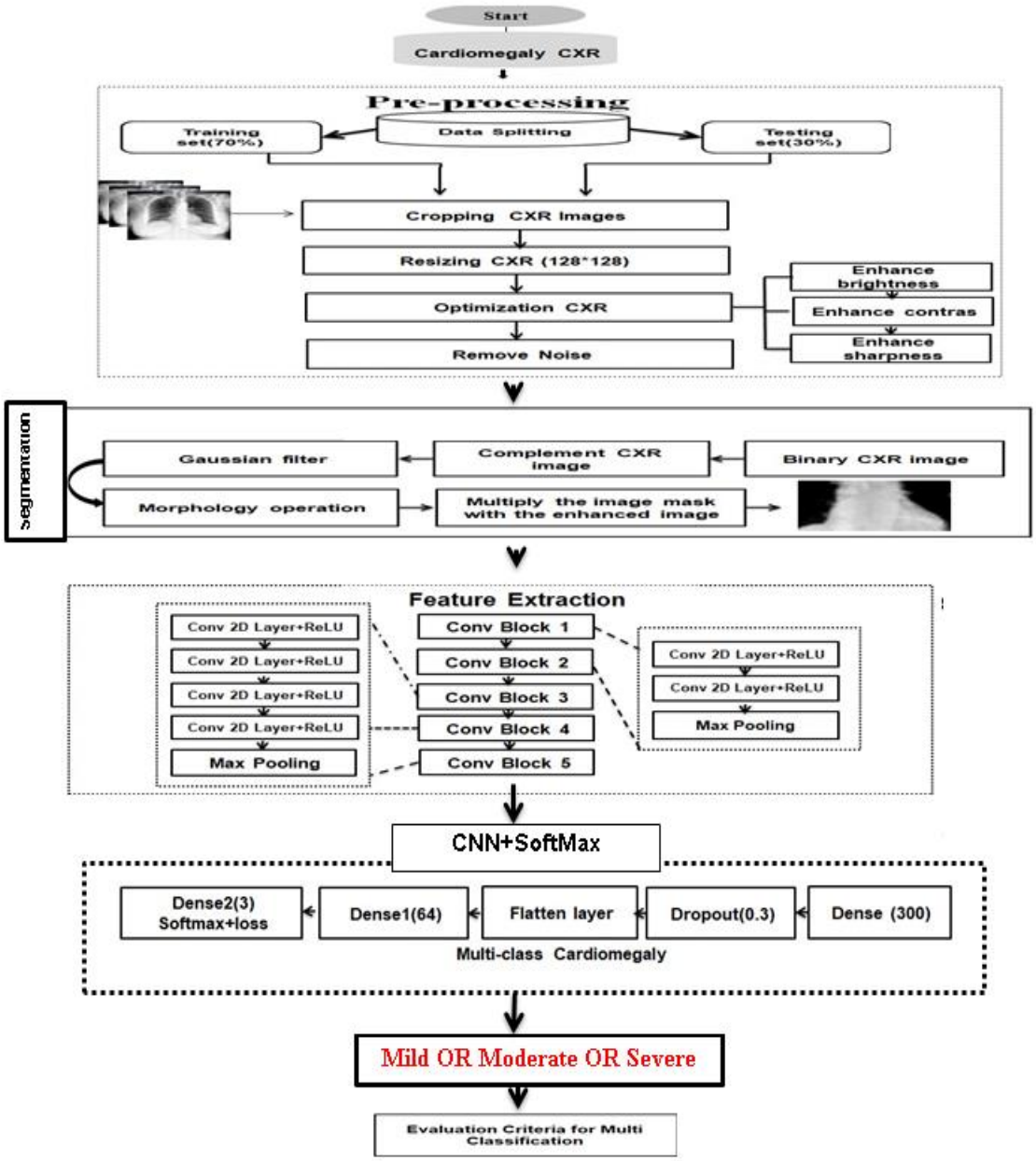


Figure (3.2) The block diagram of the Suggested System.

3.3 Preprocessing Stage

The classification system utilized for cardiomegaly incorporates a series of fundamental steps referred to as pre-processing operations. These operations encompass initial methodologies aimed to initialize data for the next stage.

The pre-processing operations encompass four steps, which involve resizing the CXR images to dimensions of (128*128) and cropping them, optimizing the CXR images, and applying a Median filter to de-noise the optimized images.

The steps of the preprocessing phase are presented in the figure (3.3) and the algorithm (3.1).

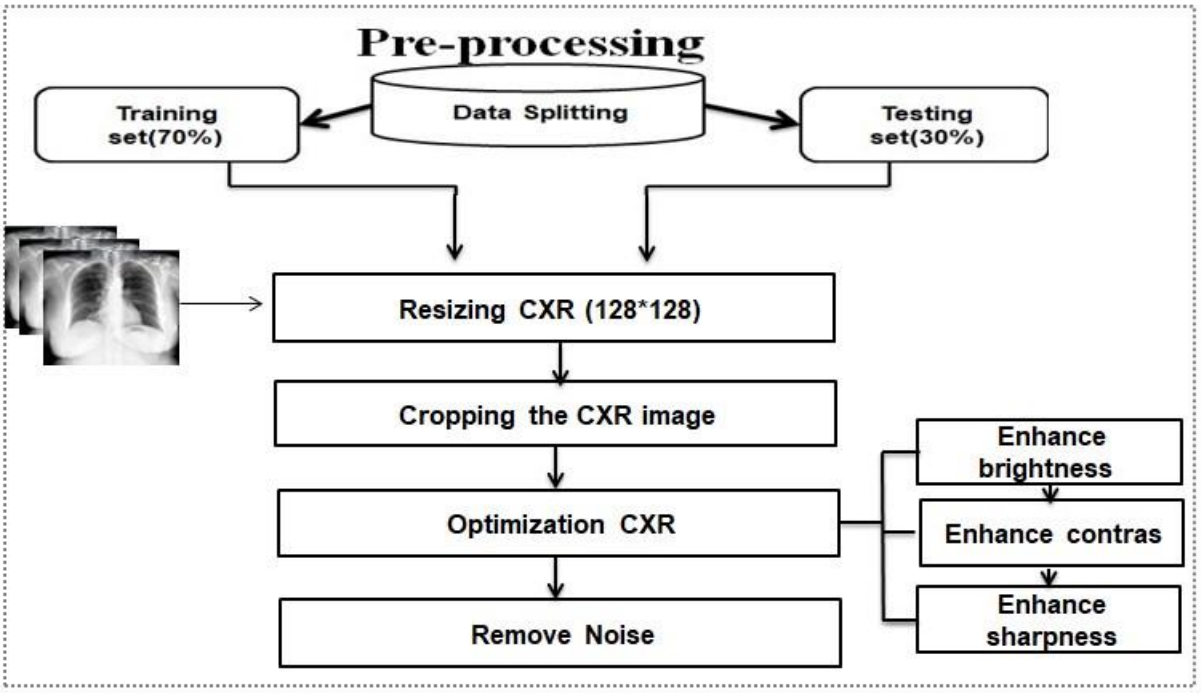


Figure (3.3) The steps in the preprocessing stage.

Algorithm (3.1): Prepossessing operations

Input: CXR images.

Output: Enhanced CXR images**.** **# (128 * 128) gray scale 8-bit images**

Begin

Step 1: Resize the CXR images to (128*128).

Step 2: Cropping the CXR images (left (30), right (100), top (20), and bottom (110))**.**

Step 3: Optimize the CXR Images by :

a) Reducing the intensity of the CXR image by a factor of 0.5.

b) Enhance the CXR image contrast by increasing the contrast by a factor of 1.5.

c) Enhance the CXR image sharpness by increasing the sharpness by a factor of 1.5.

Step 4: The noise was removed from CXR image using the Median Filter, in equation (2.1).

End.

3.3.1 CXR Images Resizing

Chest X-ray images should be resized to 128*128 to help standardize input sizes, simplify the training process and avoid problems caused by different image dimensions, and improve computational efficiency because processing smaller-sized images requires fewer computational resources, which can lead to

faster training and inference times, and manage memory resources by minimizing the memory footprint needed to store and process the images.

3.3.2 CXR Images Cropping

The act of cropping chest X-ray images serves to delineate and emphasize specific regions of interest (ROI) for analysis and classification purposes. Via cropping, irrelevant information can be effectively eliminated, thereby standardizing image sizes and enhancing computational efficiency. As a result, the deep learning model is empowered to focus solely on the most informative aspects of the image, thus augmenting its proficiency in detecting abnormalities or discerning specific patterns .

In the proposed system implemented a cropping approach with precise dimensions for the images, specifically with left (30), right (100), top (20), and bottom (110) margins.

3.3.3 CXR image optimization

CXR image optimization comprises a set of techniques with the aim of improving the quality and visual clarity of radiographic images acquired from the chest region. To enhance the diagnostic value of the images by implementing several adjustments, the brightness is enhanced by adjusting the brightness of the CXR image, reducing its intensity by a factor of (0.5), enhancing contrast and sharpness by a factor of (1.5) to augment the distinction between dark and light regions, accentuating edges, and highlighting intricate details.

3.3.4 CXR image De-noise

During this phase, the CXR images undergo a noise reduction and eliminate common and recognized imperfections that may have been introduced during image collection and transmission.

To address the impact of noise, a filtering technique known as the median filter, as referenced in the chapter two, has been used in a window size (3 × 3) to remove the noise in the CXR images.

3.4 Segmentation of Heart

In this section, heart area segmentation as a region of interest (ROI) by using a threshold with morphological operations has been proposed.

- **Binary CXR Image(Otsu Thresholding)**

The enhanced Grayscale Images is converted to a binary image using (Otsu Thresholding) as shown in Figure (3.4). This method are described in the section (2.4).

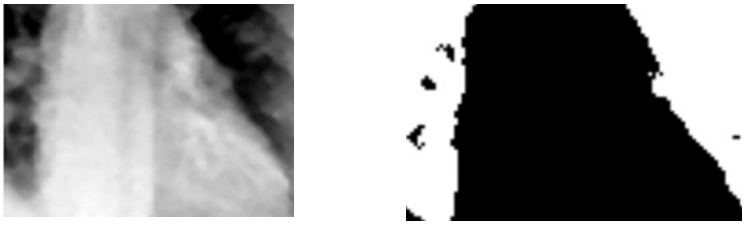


Figure (3.4) Shows binary image (a) Enhancement image (b) Binary image.

- **The Complement of Binary CXR Image**

Take the inverse of the binary image to convert the pixel of the heart image from the black pixel to the white pixel, as shown in Figure (3.5).

Figure (3.5) The complement of binary CXR image.

- **The Gaussian Filter on the Complement Image**

The Gaussian filter has been used, denoted by equation (2.1) in chapter two, with window size (5 ×5) for bluring and remove the noise in the CXR image.

Figure (3.6) Gaussian filter.

- **A Morphological Closing and Opening Operation.**

The morphological closing operation is implemented, which includes a process of dilation accompanied by a process of erosion as in equation (2.8), and the opening operation is implemented, which includes a process of erosion accompanied by dilation as in equation (2.7) for connect the pixels areas within the heart tissue. In these operations, the size of the structure element is (5*5).

(a) **(b)**

Figure (3.7) Morphological operation (a) Closing, (b) Opening.

- **Multiply the image mask with the enhanced image**

In order to the extraction of heart tissue only, the final extracted mask will be multiplied with the enhanced image, as shown in the figure (3.8).

Figure (3.8) Multiplying the enhanced image with extracted mask image.

The algorithm (3.2) presents this step:

Algorithm (3.2): CXR Segmentation	
Input: Enhanced CXR images**.**	**# (128 * 128) gray scale 8-bit images**

Output: heart Segmentation.

Begin

Step 1: Convert enhanced CXR image into binary image by using Otsu Thresholding in Equation (2.3).

Step 2: Take the complement to the binary image.

Step3: Implement a Gaussian filter on the binary image with Kernel (5*5) in equation (2.2)

Step 4: Applying a morphological closing and opening operation with kernel (5*5) in equations (2.7) and (2.6)**.**

Step 5: Multiply the image mask from step **4** with the enhanced image.

End.

3.5 CXR Feature Extraction Step

The feature extraction process holds paramount importance in deep learning approaches. This process encompasses the identification and extraction of valuable patterns, structures, and attributes from chest X-ray images, the extracted feature will be the entrance to the classification step. To achieve feature extraction objectives the VGG19 architecture is utilized. The features extracted in the proposed structure of a convolution Neural Network technique consist of five convolution blocks of VGG19. The fully connected layers of the VGG19 are frozen to focus exclusively on feature extraction, rather than classification, and all layers as non-trainable, use the pre-existing initial weights, trained on the ImageNet dataset as initial weights, and proceed to extract precise features by generating feature maps from the input images.

3.6 The Convolutional Neural Network Structure

The creation of a Convolutional Neural Network (CNN) model encompasses a two-step process: Initially, the foremost step involves formulating the model design the second step necessitates configuring the training parameters, which play a crucial role in fine-tuning the model's performance .which play a crucial role in fine-tuning the model's performance.

A. Design the Convolution Neural Network

This step involves positioning the various layers to achieve the best feature extraction method.The structure for extracting the features was presented in Table (3.1) from the first row to the twenty-three, while the remaining rows of the table represent the classification stage.

Input Layer: The artificial input neurons that make up this layer, which serves as the neural network's first point of entry, are in charge of taking in and sending raw data to later layers for additional processing. The new model's input shape is (128, 128, 3).

Rescaling Layer: is in charge of turning input pixel values into a normalized range of 0 to 1.

VGG19: The model consists of a series of convolutional layers organized into five convolutional blocks as shown:

•**Block 1:** consist of two layer of Convolution2D(64, 3, 3, activation='relu') and ZeroPadding2D (1,1),input shape (3,None,None), MaxPooling2D with size (2,2) and strides (2,2).

•**Block2**: consist of two layer of Convolution2D (128, 3, 3, activation='relu') and ZeroPadding2D (1, 1), MaxPooling2D with size (2, 2) and strides (2, 2).

•**Block 3:** consist of four layer of Convolution2D (256, 3, 3, activation='relu') and ZeroPadding2D (1, 1) with MaxPooling2D with size (2, 2) and strides (2, 2).

•**Block 4:** consist of four layer of Convolution2D (512, 3, 3, activation='relu') and ZeroPadding2D (1, 1) with MaxPooling2D with size (2, 2) and strides (2, 2).

•**Block 5:** consist of four layer of Convolution2D (512, 3, 3, activation='relu') and ZeroPadding2D (1, 1) with MaxPooling2D with size (2, 2) and strides (2, 2).

The number of filters progressively increases from 64 in the initial layers to 512 in the subsequent layers. The VGG19 architecture spans from the third layer to the twenty-three layer, as indicated in Table (3.1).

Dense: With 300 units and a rectified linear activation function, this produces a new densely connected layer.

Dropout_3: To avoid overfitting, it adds a dropout layer with a rate of 0.3.

Flatten_3: It flattens the previous layer's output so that it may be connected to the final output layer, which turns the previous feature map matrix into a single column or vector of values, The feature maps (three dimensions) will be converted to a vector (one dimension) so the output is 4*4*300=4800.

Dense_1: In order to categorize the condition as either Normal or Cardiomegaly, this layer constructs output layer with one unit and a sigmoid activation function.

Rectified Linear Unit (ReLU): This is used after each convolutional layer ReLU is a common nonlinear function that produces an output equal to the input value if positive and zero otherwise. This activation function is popular in neural networks because of its simplicity and capacity to deliver superior training results.

Table (3.1) The suggest CNN structure for binary classification

Num	Layer Name	Output layer	Param#
1.	layer_1(Conv 2D + ReLU)	(None,128,128,3)	/
2.	Layer_2(rescaling)	(None,128,128,3)	/
3.	layer_3(Conv 2D + ReLU)	(None, 128, 128, 64)	1792
4.	layer_4(Conv 2D + ReLU)	(None, 128, 128, 64)	36928
5.	layer_5(Max-Pooling 2D)	(None, 64, 64, 64)	0
6.	layer_6(Conv 2D + ReLU)	(None, 64, 64, 128)	73856
7.	layer_7(Conv 2D + ReLU)	(None, 64, 64, 128)	147584
8.	Layer_8(Max-Pooling 2D)	(None, 32, 32, 128)	0
9.	layer_9(Conv 2D + ReLU)	(None, 32, 32, 256)	295168
10.	layer_10(Conv 2D + ReLU)	(None, 32, 32, 256)	590080
11.	layer_11(Conv 2D + ReLU)	(None, 32, 32, 256)	590080

12.	layer_12(Conv 2D + ReLU)	(None, 32, 32, 256)	590080
13.	layer_13(Max-Pooling 2D)	(None, 16, 16, 256)	0
14.	layer_14(Conv 2D + ReLU)	(None, 16, 16, 512)	1180160
15.	layer_15(Conv 2D + ReLU)	(None, 16, 16, 512)	2359808
16.	layer_16(Conv 2D + ReLU)	(None, 16, 16, 512)	2359808
17.	layer_17(Conv 2D + ReLU)	(None,16,16,512)	2359808
18.	layer_18(Max-Pooling 2D)	(None,8,8,512)	0
19.	layer_19(Conv 2D + ReLU)	(None,8,8,512)	2359808
20.	layer_20(Conv 2D + ReLU)	(None,8,8,512)	2359808
21.	layer_21(Conv 2D + ReLU)	(None,8,8,512)	2359808
22.	layer_22(Conv 2D + ReLU)	(None,8,8,512)	2359808
23.	layer_23(Max-Pooling 2D)	(None,4,4,512)	0
24.	layer_24(Dense + ReLU)	(None,4,4,300)	153900
25.	layer_25(dropout(0.3))	(None,4,4,300)	0
26.	layer_26(flatten)	(None,4800)	0
27.	layer_27(Dense_1+Sigmoid)	(None,1)	4801

Trainable parameters: 158701

Non Trainable parameters:20024384

B. Hyper-Parameter Training

The development of an algorithm capable of self-training to discern images of normal and cardiomegaly hearts from CXR images entails the establishment of numerous parameters and sub-techniques.

1. **Dataset split:** The dataset is uploaded to the suggested system in this stage before the completion of the image pre-processing process. 30% of the dataset is used for testing, while 70% is for training.

2. Loss function: Since we are dealing with binary and multiple class's classification problems, and memory is needed to do the training fast, so choose the Binary Cross-Entropy and Sparse Multiclass Cross-Entropy Loss (as described in the chapter two). Notice with this method, the labels provided to the loss function must be integer.

3. Optimizer: The optimization process is about finding the best parameters and values for the kernels and biases, to set the training operation correctly (as described in the chapter two). In our approach, the Adam optimizer is chosen. Due to its high performance. The number of epochs is set to **20**, the batch size is set to 120, and the starting learning rate is set to 0.001.

4. **The early stopping strategy**: Early stopping technique is used to regulate the training of models (as described in the chapter two).

In our model training, the validation dataset is set as the stopping criteria. When the performance of the model comes to a halt increase decreases on the plateau, the principle of early stopping is adopted, when it can be demonstrated that the value of accuracy of training or of validation stayed constant and maintained a specific level after 18 periods, training is terminated.

Algorithm (3.4) illustrate how the suggested model's training is carried out for binary classification.

Algorithm (3.3): Training CNN model –Binary class

Input: Heart segmented, epoch .

y: set of labelled CXR.

Output: training CNN model

Begin

N = number of images in Labeled_training_dataset.

For i=1 to epoch

Call Algorithm (2.1) to extract the most Significant features.

Pass the most Significant features from previous step into dense layer.

 # Dense with (300)unit.

Drop 30% randomly selected connections.

Convert the multidimensional feature map into 1-D vector (Flatten layer).

Pass the vector from previous step into Dense1 layer.

#Dense 1 with (1) unit

Pass output of Dense1 layer into the equation (2.8) for computing S.

#(S) Sigmoid- function was used to calculate the probabilities vector.

Pass (S and y) into the equation (2.11) to compute the value of the loss.

Call (2.2) algorithm to update the weights, filters, and biases of CNN_model.

End i

End.

3.7 System Extension

The previous model has been upgraded to cover all stages of Cardiomegaly (Mild, Moderate, and Severe) cases.

To perform all stages, the new strategy is identical to the previous one, with a few adjustments in the design. In order to produce better outcomes.

A.The New Design: New layers are added to extend the design.

Layer_27 (Dense _1): A fully connected layer with 64 units.

Layer_28 (Dense _2): is a dense layer with 3 units and softMax activation is added to obtain the output probabilities for each class.

The modified expanded design of the model (**Three Classes**) is shown the detailed structure of the new layers is shown in Table (3.2). Algorithm (3.5) illustrates how the suggested model's training is carried out.

Table (3.2) Show Structure of the new design CNN.

	Layer Name	Output layer	Param#
1.	layer_1(Conv 2D + ReLU)	(None,128,128,3)	/
2.	Layer_2(rescaling)	(None,128,128,3)	/
3.	layer_3(Conv 2D + ReLU)	(None, 128, 128, 64)	1792
4.	layer_4(Conv 2D + ReLU)	(None, 128, 128, 64)	36928
5.	layer_5(Max-Pooling 2D)	(None, 64, 64, 64)	0
6.	layer_6(Conv 2D + ReLU)	(None, 64, 64, 128)	73856
7.	layer_7(Conv 2D + ReLU)	(None, 64, 64, 128)	147584
8.	Layer_8(Max-Pooling 2D)	(None, 32, 32, 128)	0
9.	layer_9(Conv 2D + ReLU)	(None, 32, 32, 256)	295168

10.	layer_10(Conv 2D + ReLU)	(None, 32, 32, 256)	590080
11.	layer_11(Conv 2D + ReLU)	(None, 32, 32, 256)	590080
12.	layer_12(Conv 2D + ReLU)	(None, 32, 32, 256)	590080
13.	layer_13(Max-Pooling 2D)	(None, 16, 16, 256)	0
14.	layer_14(Conv 2D + ReLU)	(None, 16, 16, 512)	1180160
15.	layer_15(Conv 2D + ReLU)	(None, 16, 16, 512)	2359808
16.	layer_16(Conv 2D + ReLU)	(None, 16, 16, 512)	2359808
17.	layer_17(Conv 2D + ReLU)	(None,16,16,512)	2359808
18.	layer_18(Max-Pooling 2D)	(None,8,8,512)	0
19.	layer_19(Conv 2D + ReLU)	(None,8,8,512)	2359808
20.	layer_20(Conv 2D + ReLU)	(None,8,8,512)	2359808
21.	layer_21(Conv 2D + ReLU)	(None,8,8,512)	2359808
22.	layer_22(Conv 2D + ReLU)	(None,8,8,512)	2359808
23.	layer_23(Max-Pooling 2D)	(None,4,4,512)	0
24.	layer_24(Dense + ReLU)	(None,4,4,300)	153900
25.	layer_25(dropout(0.3))	(None,4,4,300)	0
26.	layer_26(flatten)	(None,4800)	0
27.	Layer_27(Dense_1+Relu)	(None,64)	307264
28.	layer_28(Dense_2+softmax)	(None,3)	260

Total Parameter: 20,485,808

Trainable parameters: 461,424

Algorithm (3.4) Training CNN model –multi class

Input: Heart segmented,epoch .

y: set of labelled CXR.

Output: training CNN model

Begin

N = number of images in Labeled_training_dataset.

For i=1: epochs

Call Algorithm (2.1) to extract the most Significant features.

Pass the most Significant features from previous step into dense layer.

 # Dense with (300)unit.

Drop 30% randomly selected connections.

Convert the multidimensional feature map into 1-D vector (Flatten layer).

Pass the vector from previous step into Dense1.

#Dense1 with (64)unit.

Pass output of Dense1 layer into Dense 2.

#Dense 2 with (3) unit.

Pass output of Dense2 layer into the equation (2.10) for computing S1.

#(S1) softmax- function was used to calculate the probabilities vector.

Pass (S1 and y) into the equation (2.11) to compute the value of the loss.

Call (2.2) algorithm to update the weights, filters, and biases of CNN_model.

End i

End.

During the test stage of the CNN, the test is carried out on the unnoticed test data and starts with preprocessing according to the algorithm (3.1). The next step is to extract the heart Segmentation images according to the algorithm (3.2) and use them in the convolutional neural network structure in the forward direction only to extract the features, and then classify these images into binary classification (Normal and Cardiomegaly) or categorization (Cardiomegaly cases) using the CNN-VGG19 deep learning.

Chapter Four

Results and Discussion

CHAPTER FOUR

4 RESULTS AND DISCUSSION

4.1 Introduction

The present chapter aims to elucidate the performance outcomes yielded by the system. The system's performance results will be discussed in this chapter, which is divided into two phases: training and testing. Each of these phases encompasses three steps: (1) the preprocessing steps results, (2) the outcomes of the build of the masks of chest x-rays and segmented CXR images, and (3) the Results of the classification phase of binary and multi of cardiomegaly. Moreover, this section meticulously covers the chest X-ray image database and its segregation, provides a complete breakdown of the outcomes for every stage within the suggested system, and assesses the proposed system's performance by computing metrics for performance. These findings were acquired with the Python programming language version 3.8.3.

4.2 Hardware and Software requisites

The proposed system for detecting cardiomegaly is operated through the use of a personal computer (specifically, a Think Pad with an Intel (R) Core i7- 8565U @ 1.80 GHz for CPU, 8 GB RAM, and a 64-bit OS), as well as Google Colab as a collaborative tool. Google Colab, an open Jupiter Notebook environment that functions entirely in the cloud, is utilized to leverage free GPUs for model training. This platform is highly suitable for deep learning tasks, allowing for the rapid training of large databases and complex models on Google's computing resources (GPU). The implementation of the system is facilitated through the use of

TensorFlow, an open-source programming library developed by Google that is focused on utilizing tensors to achieve effective results, and Keras, a neural network library based on TensorFlow. Furthermore, the system's implementation of the CNN code employs the open-source programming language, Python, and utilizes several open-source libraries such as Open, Sci-kit Learn, and Pandas. These libraries are specifically designed to address machine learning and data analysis tasks.

4.3 CXR dataset Preparations

The data for this thesis is sourced from Kaggle and consists of (5,552) CXR images obtained from the NIH Clinical Center. This dataset encompasses both normal cases and instances of cardiomegaly. To facilitate model training and evaluation, these CXR images are divided into a 70% training set (3,886) consisting of (1,943) CXR images for normal hearts and (1,943) CXR images for cardiomegaly. Additionally, a 20% testing set (1,110) CXR images is created, consisting of (555) normal heart CXR images and (555) cardiomegaly CXR images, and 10% for validation set (556) CXR images is created, consisting of (278) normal heart CXR images and (278) cardiomegaly CXR images.

Within the subset focusing on the Cardiomegaly class of (2776) CXR images, a finer division is performed based on the severity of the condition. This results in a 70% training set (1943) CXR images consisting of (647) CXR images representing the mild stage, (648) CXR images representing the moderate stage, and (648) CXR images representing the severe stage, and the 20% testing set (556) CXR images. This set of cardiomegaly cases is further subdivided into (185) CXR images for the mild stage, (185) CXR images for the moderate stage, and (186) images for the severe stage, and validation set (277) CXR images consisting of (92) CXR images

representing the mild stage, (93) CXR images representing the moderate stage, and (92) CXR images representing the severe stage.The dataset is partitioned for both binary and multi-class models, and separate training and testing phases are established as shown in Tables (4.1) and (4.2), respectively. Visual representations of the original chest X-ray images are presented in Figure (4.1), depicting examples of both normal cases and cardiomegaly cases across different severity levels (mild, moderate, and severe).

Table (4.1) Statistics of CXR images division (binary class).

	Normal	Cardiomegaly	Total
Train	**1943**	**1943**	**3886**
Test	**555**	**555**	**1110**
Validation	**278**	**278**	**556**
Total	**2776**	**2776**	**5552**

Table (4.2) Statistics of CXR images division (multi-class).

	Mild	Moderate	Severe	Total
Train	**647**	**648**	**648**	**1943**
Test	**185**	**185**	**186**	**556**
Validation	**92**	**93**	**92**	**277**
Total	**924**	**926**	**926**	**2776**

(a)

(b)

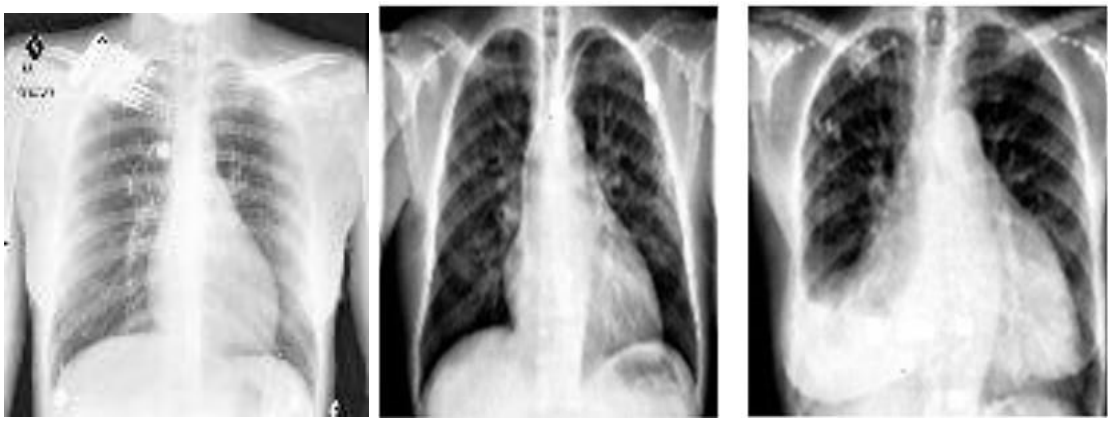


Figure (4.1) Samples of the CXR dataset :(a) samples of the normal dataset, (b) Samples of the Cardiomegaly dataset.

4.4 Results of the Proposed System's Implementation.

This section describes the detailed output of the steps of the cardiomegaly proposed system. The outputs are organized into two phases: the outputs of the training phase, and the outputs of the test phase.

4.4.1 Outputs of the Training Phase

To be trained, all training samples (3886 of heart segmentation images) for binary class and (1943 heart segmentation images) for multi-class with their labels

must pass through the system at the same time during the training phase. The following are the results and output of the preprocessing steps, mask CXR and segmented CXR images generation steps, and classification steps.

4.4.1.1 Results of Applying Prepossessing of CXR Images

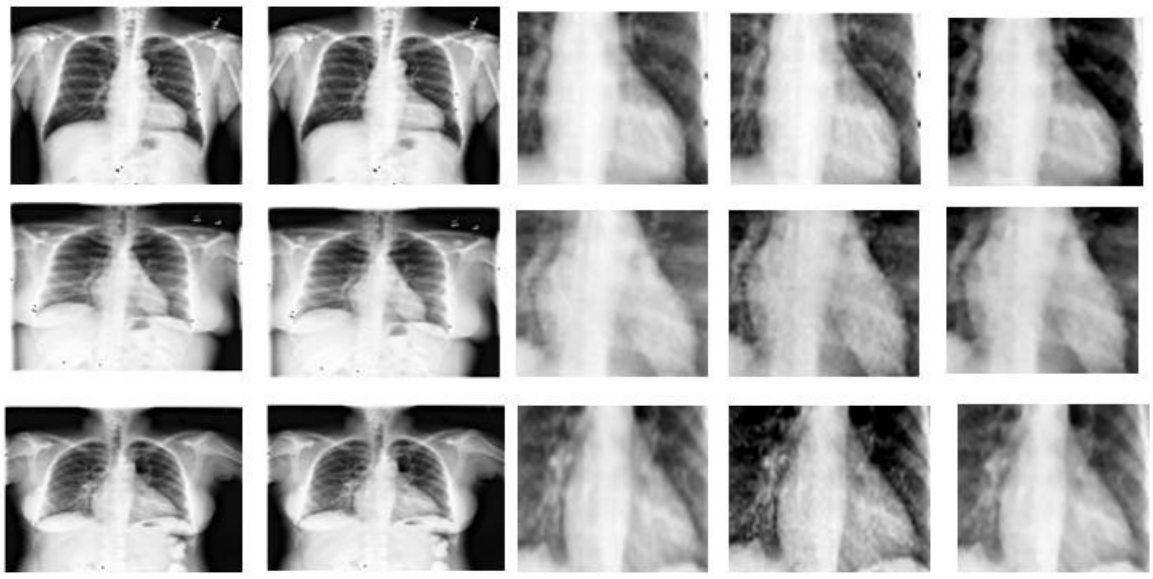
The preprocessing stage encompasses four principal stages. The outcomes of employing the preprocessing techniques to CXR image databases are presented in Figure (4.2), based on the following steps.

- Resize CXR to (128*128).
- Cropping CXR images.
- CXR Image Optimization (enhances brightness (0.5), enhance contrast (1.5), enhance sharpness (1.5).
- Remove Noise (Median Filter).

In order to see the effect of the pre-processing operations on the CXR images in the dataset, a sample of three images is selected to see the effect of the images' pre-processing operations. These three images are displayed first in their original form. After that, the operation of the pre-processing operation is shown in all three images to see the effect of the operation.

Figure (4.2.a) shows the original CXR image before pre-processing. After that, the CXR images are resized into 128*128 pixels in Figure (4.2.b),Figure (4.2.c) shows the cropping of the CXR images, this procedure entails choosing and extracting a specific region of interest within the images In order to exclude unnecessary information,. Figure (4.2.d) shows the CXR images after applying enhancement steps, including enhancing the brightness, contrast, and sharpness. This optimization process targets the augmentation of the diagnostic value of the

images by amplifying contrast, sharpness, and adjusting brightness. Figure (4.2.e) shows the CXR images after applying a median filter, Notice that images are generally clearer after this step.

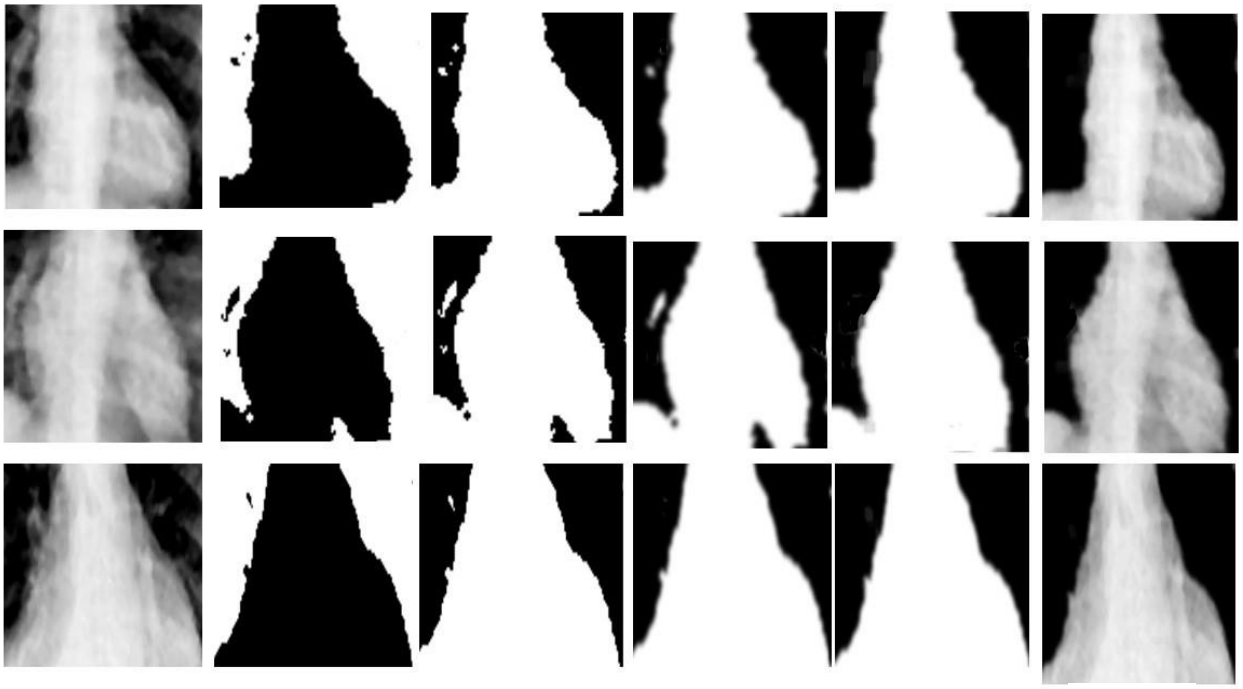


(a) **(b)** **(c)** **(d)** **(e)**

Figure (4.2) Depicts the CXR images before and after pre-processing.

4.4.1.2 Result of Build The Mask

Morphological operation and intensity thresholding are used for CXR image segmentation processes. Figure (4.3) depicts the heart segmentation results from CXR images as follows: Binary image (the result of Otsu thresholding), Inverse image, Gaussian filter on complement image, morphological closing and opening operation, and multiply the image mask by the enhanced image to produce the ROI image.



(a) (b) (c) (d) (e) (f)

Figure (4.3): illustrate the stage of building the mask, (a) Enhanced CXR image,(b) Binary image,(c) Inverse image,(d) Gaussian filter,(e) close and open operation,(f) Multiplication mask

4.4.1.3 Results of The Classification Phase

In this thesis, two classifiers (CNN-Sigmoid and CNN-SoftMax) are utilized to detect Cardiomegaly infected instances:

4.4.1.3.1 Results of the Binary Classification Steps

In the current phase, there are two distinct classes (normal and cardiomegaly), and the convolutional neural network architecture consists of the remaining layers:

- **Dense layer:** adds a new 300-unit with a rectified linear activation function.
- **Dropout:** adds a 0.3 dropout layer to prevent overfitting.
- **Flattening Layer:** This stratum is responsible for the conversion of feature maps, which are inherently three-dimensional, into a singular vector that exists in a one-dimensional plane. This transformation yields an output of dimensions 4 × 4 × 300, resulting in a vector length of 4800.
- **The dense layer 1** is fully connected stratum comprising a one-unit configuration. Within this layer, the sigmoid activation function which computes the probability for each class on the basis of the given input image. The probability is sent to the loss function sparse categorical cross-entropy to calculate the error value, which is then used to modify (update) the weights using the back propagation method.

  In the training phase, the proposed methodology uses the Adam optimization, learning rate reduction to improve the value of the validation loss when model performance stops increasing (reduce on the plateau), batch size = 122, and early stopping to find the best number of epochs (maximum = 20). Figure (4.4) shows the evaluation of CNN's accuracy performance the evaluation of CNN's performance in accuracy score is the total amount of errors the model predicted since it can be noticed that the value of accuracy for training or validation did not change and stabilized at a certain value after 18 epochs, so the training was stopped.

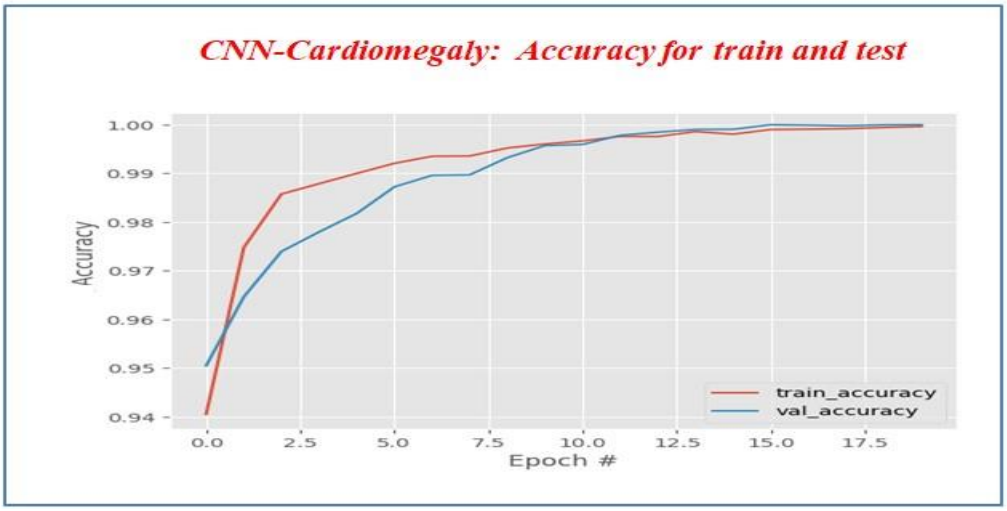


Figure (4.4) Depicts the accuracy curve for the training and validation samples (with mask).

Figure (4.5) depicts the curves of learning loss of training and loss of validation when a model has been trained on images of CXR masks and segmented CXR images.

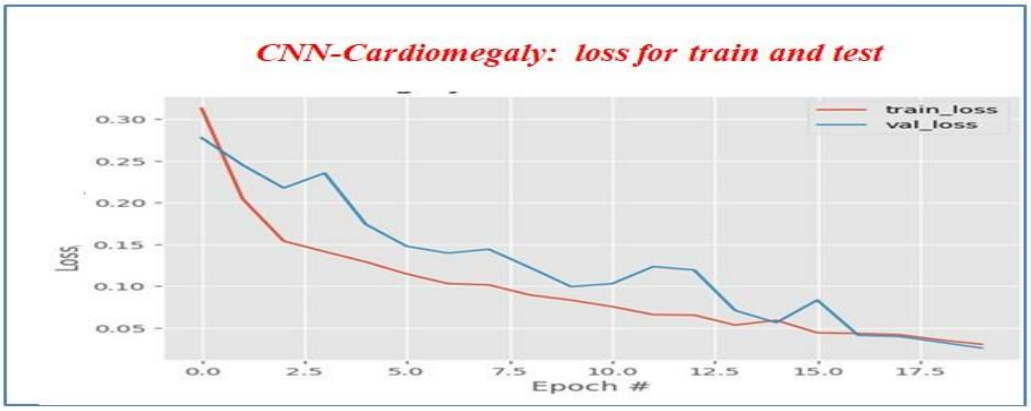


Figure (4.5) Shows the curve of loss for training and validation sample(with mask).

When tested, this training resulted in a high-performance model. The confusion matrix extracted by the test phase is depicted in Figure (4.6).

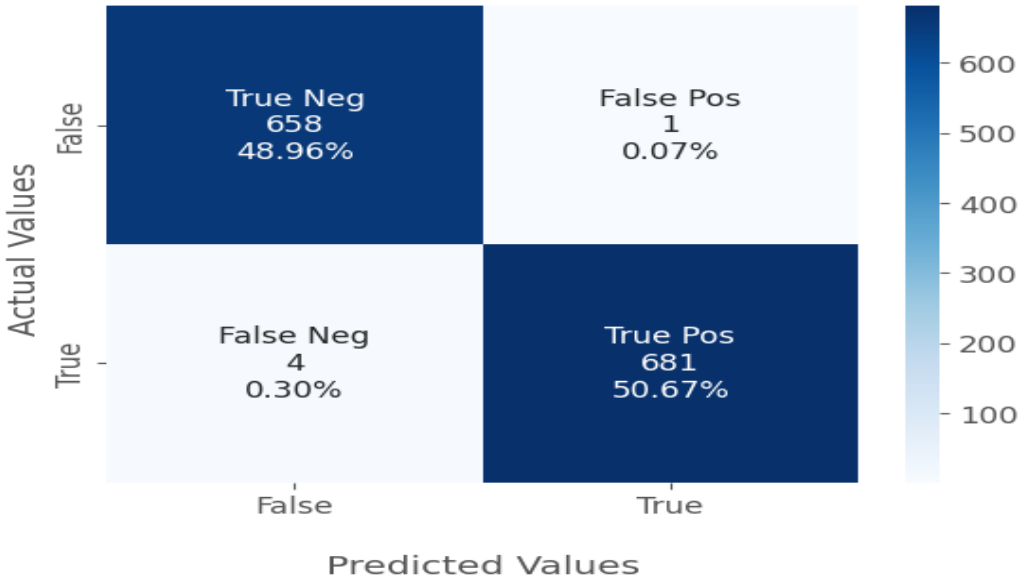


Figure (4.6) Confusion matrix obtained during system test phase (with mask).

When the model is trained on the processed CXR images (without Mask), the Performance metrics decrease, as shown in Table (4.3) shows the performance of the CNN training model on the Mask of CXR images and without the mask CXR images, in terms of Sensitivity, precision, F-score, and Accuracy.

Table (4.3) Shows the result of the performance for the binary class.

	Accuracy	Sensitivity	Precision	F-score
With mask	**99%**	**99.4%**	**99.8%**	**99.6%**
Without mask	**94.5%**	**97.5%**	**91.9%**	**94.6%**

The results presented above demonstrate the superior efficacy of the binary classification model with the mask in the detection of cardiomegaly by the X-ray

images when compared to its counterpart the model lacking the mask. Figure(4.7) show the confusion matrix for this model without mask.

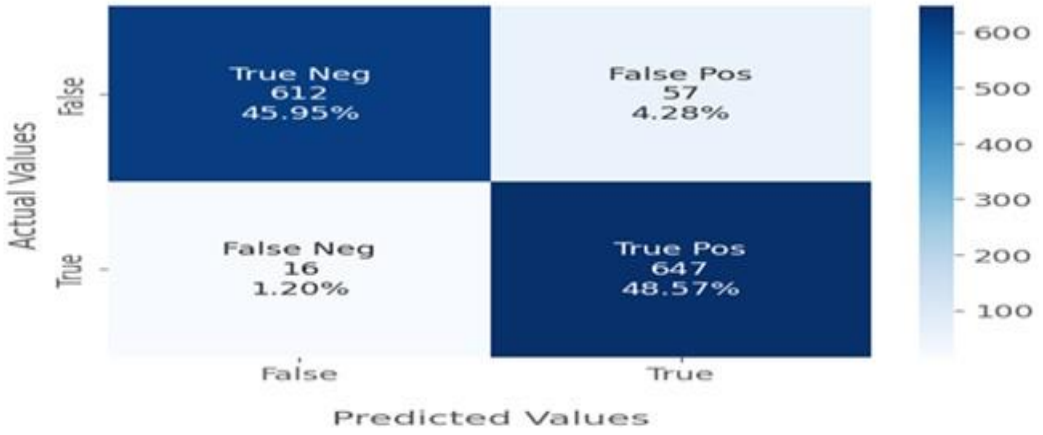


Figure (4.7) The confusion matrix of the test sample(without mask)

4.4.1.3.2 Results of the Multi- Classification Steps

In this phase of the system's development, the (Softmax) function is used to classify cardiomegaly conditions, there are three distinct classes (mild, moderate, and severe), and the convolutional neural network architecture consists of the remaining layers:

- **Dense layer**: adds a new 300-unit with a rectified linear activation function.
- **Flattening Layer:** This stratum is responsible for the conversion of feature maps, which are inherently three-dimensional, into a singular vector that exists in a one-dimensional plane. This transformation yields an output of dimensions 4 × 4 × 300, resulting in a vector length of 4800.
- **Dense layer 1:** fully connected, and this layer uses a rectified linear unit ReLU function for activation with 64 units.

- **Dense layer 2:** output layer, fully connected layer with three units. This layer's output is fed, to the SoftMax activation function which computes the probability for each class on the basis of the given input image. The probability is sent to the loss function sparse categorical cross-entropy to calculate the error value, which is then used to modify (update) the weights using the back propagation method.

In the training phase, the proposed methodology uses the Adam optimization, learning rate reduction to improve the value of the validation loss when model performance stops increasing (reduce on the plateau), batch size = 122, and early stopping to find the best number of epochs (maximum = 20). Figure (4.8) shows the evaluation of CNN's accuracy performance the evaluation of CNN's performance in accuracy score is the total amount of errors the model predicted since it can be noticed that the value of accuracy for training or validation did not change and stabilized at a certain value after 18 epochs, so the training was stopped.

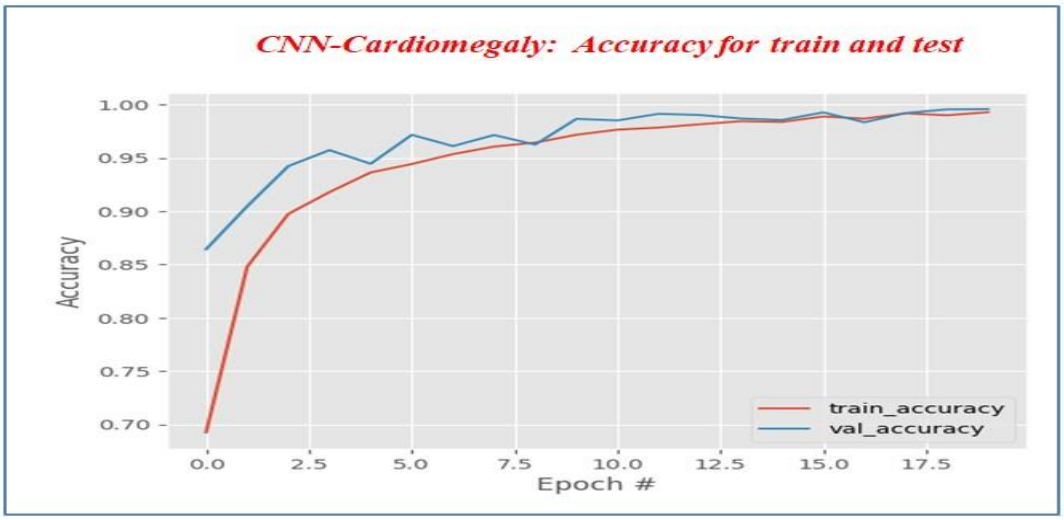


Figure (4.8) Depicts the accuracy curve for the training and validation samples (with mask).

Figure (4.9) depicts the curves of learning loss of training and loss of validation when a model has been trained on images of CXR masks and segmented CXR images.

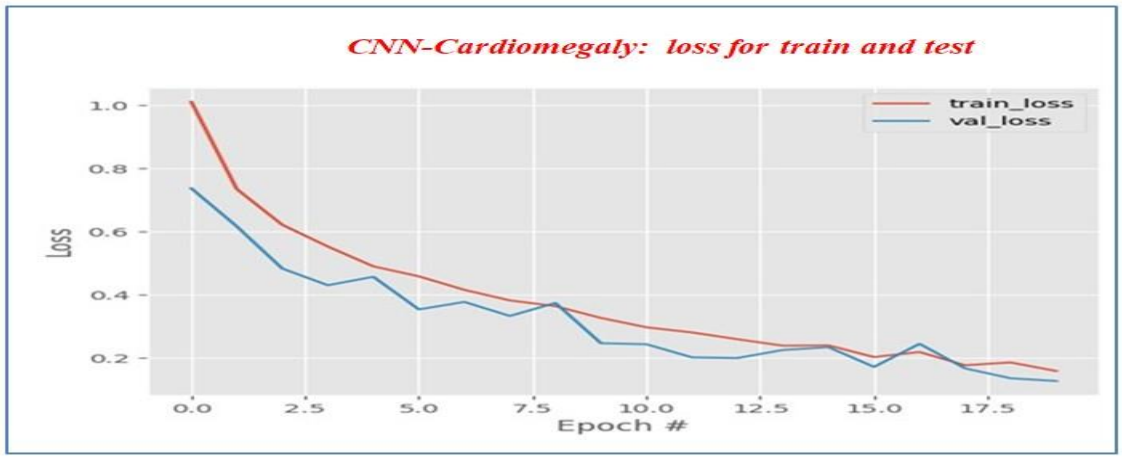


Figure (4.9) The curve of loss for training and validation samples(with mask).

This training produced a high-performance model when tested. Figure (4.10) depicts the confusion matrix obtained during the testing phase.

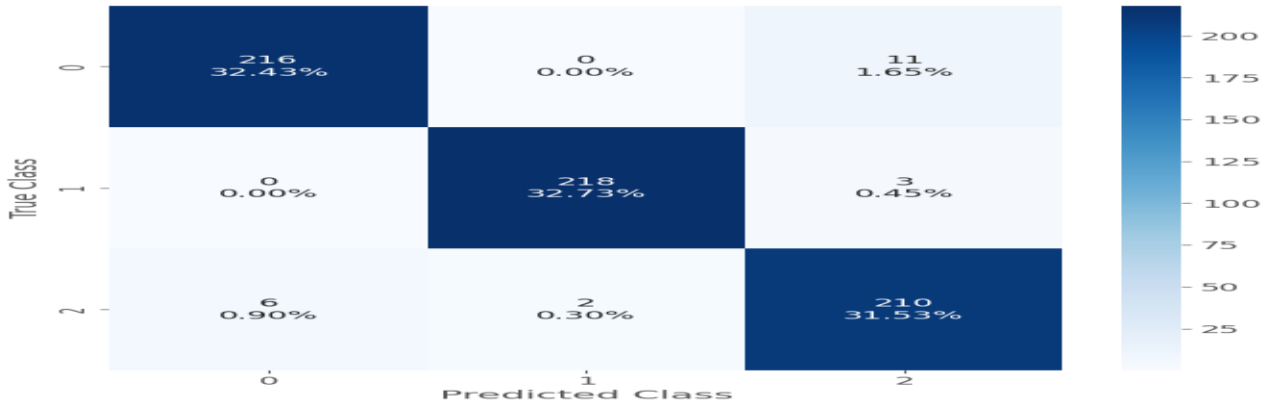


Figure (4.10) Confusion matrix obtained during the multi-class system test phase (with mask).

When the model is trained on the processed CXR images (without Mask), the Performance metrics decrease, as shown in Table (4.4) shows the performance of the CNN training model on the Mask of CXR images and without the mask CXR images, in terms of Sensitivity, precision, F-score, and Accuracy.

Table (4.4) Shows the result of performance for the Multi - class.

	Accuracy	Sensitivity	Precision	F-score
With mask	**97%**	**95.6%**	**96.3%**	**95.2%**
Without mask	**92%**	**91.6%**	**92.3%**	**91.2%**

The results presented above demonstrate the superior efficacy of the multi-classification model with the mask in the detection of cardiomegaly by the X-ray images when compared to its counterpart the model lacking the mask. Figure (4.11) show the confusion matrix for this model without mask.

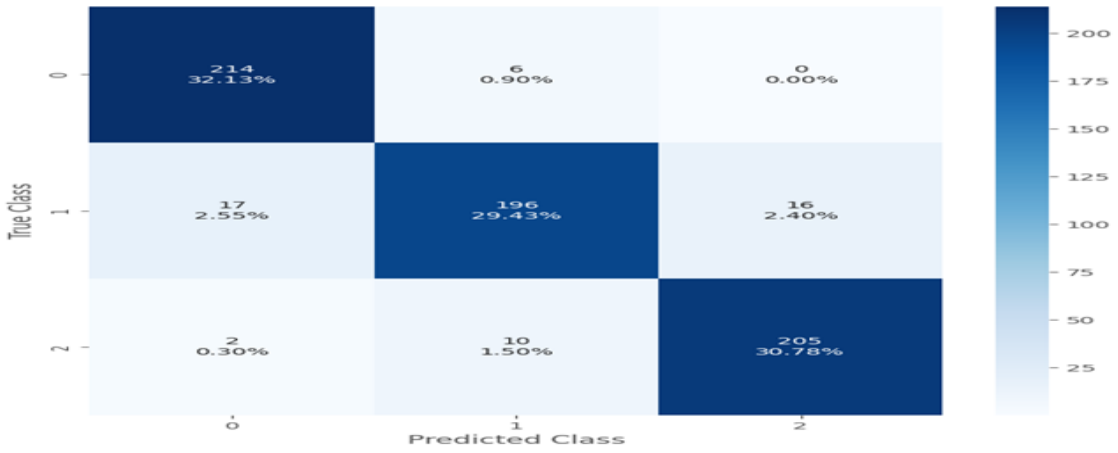


Figure (4.11) Confusion matrix obtained during the multi-class system test phase (without mask).

4.4.2 The Results of Testing Phase

During the testing phase, the proposed system will utilize 30% of the Kaggle chest X-ray image database (1666 CXR images) as test samples for binary classification, and (833 CXR images) as test samples for multi- classification. These samples will be processed by the system without their labels to evaluate the system's performance. The test phase follows the same feature extraction steps as the training phase, but only in a forward direction as the images are passed through the designated blocks for feature extraction in the convolutional neural network architecture -VGG19 blocks. The system will then classify the test images into (Normal and Cardiomegaly) for binary classification, and classify cardiomegaly CXR images into (mild, moderate, and severe). At the classification stages, the system will utilize the Sigmoid and Softmax functions to identify cases of cardiomegaly and output the results of the proposed system. The Figure (4.12) displays samples of the test data.

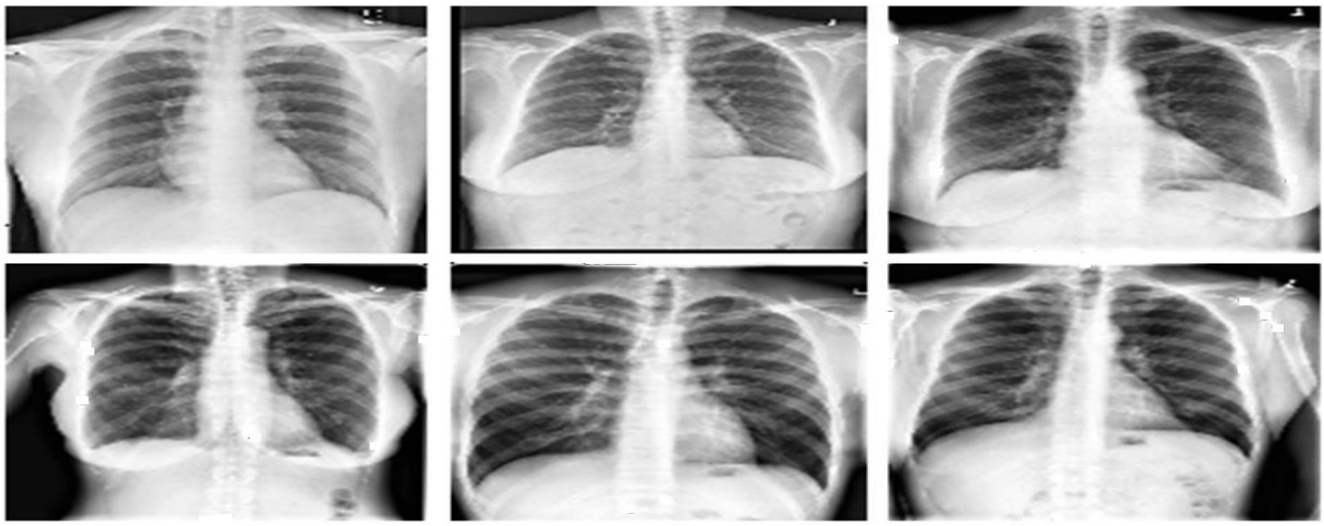


Figure (4.12) Show the CXR samples for testing.

To undertake comprehensive validation for the proposed system, a comparative analysis between the outcomes attained through the aforementioned approaches and those reported in the extant literature and using the same dataset, Table (4.5) summarizes the findings, and the two approaches achieved high performance compared to other studies.

Table (4.5) Comparison of the suggested method to other methods

Author Study	Techniques	Dataset	Performance	
			Accuracy (%)	*Sensitivity (%)*
Isarun.Chamveha,et al.2020[13]	U-Net with VGG16 encoder	NIH	67.1	0.81
Diego.A.Cardona ardenas,et.al. 2020 [1]	EfficientNet,	NIH	0.910	0.760
	DenseNet			
	MobileNet		0.901	0.822
	XceptionNet			
	InceptionNet		0.886	0.869
Megha.Rathi.et.al. 2020 [16]	CNN CNN with transfer learning VGG16	NIH	55 60	
Hyun.Yoo,et.al.2021[19]	ResNet and Explainable Feature Map	NIH	80	
Mohammed Innat,et al.2023[21]	CXR-ResNet50	NIH	75	
	CXA_Incep Res2		84.75	
	CXR_Dense Net201		80.58	

	CXA_Dense 121		**85**	
Proposed system- Binary classification	**DCNN**	**NIH-Kaggle**	**99**	**99.4**
Proposed system/Multi-classification	**DCNN**	**NIH-Kaggle**	**97**	**95.6%**

Chapter Five

Conclusion and Future Work

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1 Conclusions

The following are the essential conclusions of the results obtained from using the suggested method for cardiomegaly diagnosis in chest x-ray (CXR) images:

1. Prepossessing is an important phase in which images are optimized to be suited for the subsequent phases. It is also the most crucial and fundamental step in image classification. It is used to increase image quality.
2. The proposed system is fast, the CXR images prediction time takes approximately (0.06) milliseconds for one image when using the suggested system which reduces the problems of handling more Cardiomegaly cases and allows the system to be used in real-time.
3. The design and training of the proposed system are upgraded to include all the different classes of Cardiomegaly (mild, moderate, and severe), with results of 97% accuracy,95.6% sensitivity, 96.3 % precision, and an F-score of 95.2%, our model showed high performance.
4. The performance of this system is compared with other research that used the same dataset (Kaggle-NIH) and with other techniques, which showed the proposed system has higher performance than the previous researches.
5. The same systems of (binary and multi-class) cardiomegaly are trained and tested on the preprocessed CXR images bypassing the mask creation phase. The result of accuracy is 94.5% for binary classification and 92% for muli-class.
6. The process of feature extraction from cardiomegaly-afflicted chest X-ray (CXR) images is more accurate by using the VGG19 architecture, and this is reflected in the efficiency of the system.

5.2 Suggestion for Future Works

1. Develop the proposed system further to include more heart diseases like Coronary Artery Disease, Congenital heart disease, Heart Valve Disease, and Heart failure, Cardiomyopathy.
2. Using another deep learning technique instead of a Convolution Neural Network.

3. Training and testing of the proposed system on another dataset, such as CT scan.

References

References

[1] T.Ciarambino, G.Menna, G.Sansone , M.Giordano," Cardiomyopathies: an overview. International", Journal.of.Molecular.Sciences" ,in2021,MDPI. Journal,22.14:7722, https://doi.org/10.3390/ijms22147722 .

[2] X. Zhang, R. Zhang "Identifying cardiomegaly in ChestX-ray8 using transfer learning", In MEDINFO 2019: Health and Wellbeing e-Networks for All (pp. 482-486). IOS Press, https://doi.org/10.3233/SHTI190268.

[3] S.Sarpotdar."Cardiomegaly Detection using Deep Convolutional Neural Network with.U-Net".arXiv.preprint.arXiv:2205.11515,DOI: https://doi.org/10.48550/arXiv.2205.11515 .

[4] H.Amin, H & Siddiqui, "Cardiomegaly",In 2022,In StatPearls [internet], Treasure Island (FL): StatPearls Publishing.

[5] M. Abdul .Aowal, A.Tahseen. Minhaz, K.Ashraf," Abnormality Detection and Localization in Chest X-Raysusing Deep Convolutional Neural Networks",in 2017, arXiv:1705.09850v3, https://doi.org/10.48550/arXiv.1705.09850 .

[6] W.H. Hsu, F.J. Tsai , G. Zhang , C.K. Chang , P.H. Hsieh, S.N. Yang, S.S. Sun, K.Liao , E. TC Huang,"Development of a deep learning model for chest X-ray screening", in 2019, MEDICAL PHYSICS INTERNATIONAL, 7(3), 314, Taiwan.

[7] YB.Mensah,K.Mensah,S.Asiamah,H.Gbadamosi,EA.Idun,W.Brakohiapa, A.Oddoye,"Establishing the cardiothoracic ratio using chest radiographs in an indigenous ghanaian population: a simple tool for cardiomegaly screening",in 2015,Ghana medical journal, 49(3): https://doi.org/159–164,0.4314/gmj.v49i3.6.

[8] K.Dimopoulos,et.al,"Cardiothoracic ratio from postero-anterior chest radiographs: a simple, reproducible and independent marker of disease severity

and outcome in adults with congenital heart disease",in 2013, International journal.of.cardiology,166(2):453–457, ttps://doi.org/10.1016/j.ijcard.2011.10.125

[9] J.Ebenezer and A.Rao," Computer aided analysis of chest x-ray images for early detection of cardiomegaly using euler numbers", In 2017.

[10] T.Gupte , M.Niljikar, M.Gawali,V.Kulkarni,A.Kharat,"Deep learning models for calculation of cardiothoracic ratio from chest radiographs for assisted diagnosis of cardiomegaly". In 2021 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD) (pp. 1-6). IEEE,Durban,SouthAfrica, https://doi.org/10.1109/icABCD51485.2021.9519348

[11] D.GRAUPE "Deep learning neural networks: design and case studies",In 2016, World Scientific Publishing Company,USA.

[12] Q.Que, Z.Tang, R.Wang; Z. Zeng; J.Wang, M.Chua, T.Sin .Gee,X.Yang, B.Veer,"CardioXNet: automated detection for cardiomegaly based on deep learning", In 2018 ,40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 612-615). IEEE, Honolulu, HI, USA, https://doi.org/10.1109/EMBC.2018.8512374 .

[13] I. Chamveha, T Promwiset,"Automated cardiothoracic ratio calculation and cardiomegaly detection using deep learning approach". arXiv preprint arXiv:2002.07468,2020. DOI: https://doi.org//10.48550/arXiv.2002.07468 .

[14] M.Arsalan,M.Owais,T.Mahmood,J.Choi .K.Ryoung ," Artificial intelligence-based diagnosis of cardiac and related disease",In 2020, Journal of Clinical Medicine, 9(3), 871. https://doi.org/10.3390/jcm9030871 .

[15] D.Cardenas, J.F.Junior, R.Moreno, M.Rebelo, J.Krieger, M.Gutierrez, "Multicenter validation of convolutional neural networks for automated detection of cardiomegaly on chest radiographs.In 2020, In Anais do XX Simpósio Brasileiro de Computação Aplicada à Saúde (pp. 179-190). SBC,

https://doi.org/10.5753/sbcas.2020.11512 .

[16] M.Rathi,A.Mittal.” Prediction of Thorax Diseases Using Deep and Transfer Learning”. In: 2020 Research, Innovation, Knowledge Management and Technology Application for Business Sustainability (INBUSH). IEEE, 2020. p. 236-240, https://doi.org/10.1109/INBUSH46973.2020.9392161.

[17] A.Bouslama, Y.Laaziz, A.Tali,” Diagnosis and precise localization of cardiomegaly disease using U-NET”,In 2020,Informatics in Medicine Unlocked, 19, 100306, https://doi.org/10.1177/0284185120973630.

[18] H.Bougias, E.Georgiadou, N.Stogiannos,” Identifying cardiomegaly in chest X-rays: a cross-sectional study of evaluation and comparison between different transfer learning methods ”, In.2021,Acta-Radiologica,62 (12),1601- 1609, https://doi.org/10.1016/j.imu.2020.100306 .

[19] H.Yoo,S.Han, K.Chung ,”Diagnosis support model of cardiomegaly based on CNN using ResNet and explainable feature map”,IEEE Access, 2021, 9: 55802-55813, doi: https://doi.org/10.1109/ACCESS.2021.3068597.

[20] L,MS, Y.S.Kim, M.Kim, M.Usman, S.Sub Byon, S.Hyun Kim, B.Il Lee & B-Dai Lee,”Evaluation of the feasibility of explainable computer-aided detection of cardiomegaly on chest radiographs using deep learning”,In 2021, Scientific Reports , 11 (1), 16885, https://doi.org/10.1038/s41598-021-96433-1.

[21] M.Innat, F.Hossain, “A convolutional attention mapping deep neural network for classification and localization of cardiomegaly on chest X-rays”,In 2023, Scientific Reports, 13.1: 6247, https://doi.org/10.1038/s41598-023-32611-7.

[22] B.P.Doppala, A.Al Bataineh, B.Vamsi,“An Efficient, Lightweight, Tiny 2D-CNN Ensemble Model to Detect Cardiomegaly in Heart CT Images”,In 2023, Journal of Personalized Medicine, 13(9), 1338, https://doi.org/10.3390/jpm13091338.

[23] M.Revathi,"Review of Heart Disease Prediction using Data Mining Techniques",In 2016, IJSTE - International Journal of Science Technology & Engineering | Volume 2 | Issue 10 |ISSN (online): 2349-784X.

[24] S. B Priya, M Rajamanogaran,S. Subha," Prediction of chest diseases using transfer learning. Machine Learning for Healthcare Applications",in 2021, 199-213. https://doi.org/10.1002/9781119792611.ch1 .

[25] J. Wu,"Chest X-ray image analysis with combining 2D and 1D convolutional neural network based classifier for rapid cardiomegaly screening",IEEE Access, 10, 47824-47836, https://doi.org/10.1109/ACCESS.2022.3171811.

[26] C.Tian, L. Fei, W. Zheng,et al," Deep learning on image denoising: An overview",in 2020, Neural Networks,vol :131, 251-275, China, https://doi.org/10.1016/j.neunet.2020.07.025.

[27] L. Fan1, F. Zhang, H.Fan, and C.Zhang," Brief review of image denoising techniques",in 2019, Visual Computing for Industry, Biomedicine, and Art, 2, 1-12,China, https://doi.org/10.1186/s42492-019-0016-7.

[28] A.Khmag ,"Additive Gaussian noise removal based on generative adversarial network model and semi-soft thresholding approach",in 2023, Multimedia Tools and Applications, 82.5: 7757-7777, Libya, https://doi.org/10.1007/s11042-022-13569-6.

[29] J.Azzeh, B.Zahran, Z.Alqadi,"Salt and pepper noise: Effects and removal",in 2018, JOIV: International Journal on Informatics Visualization, 2(4), 252-256, Jordan, http://dx.doi.org/10.30630/joiv.2.4.151

[30] P.Singh, D .V.Sharma,"Pre-Processing of Mobile Camera Captured Images for OCR",in 2023,International Journal of Intelligent Systems and Applications in Engineering, 11(2s), 147-155, India.

[31] G. Ramesh, J. Logeshwaran, J. Gowri , A,Mathew,"The management and

reduction of digital noise in video image processing by using transmission based noise elimination scheme",ICTACT Journal on Image & Video Processing, 2022, 13.1, India, https://doi.org/10.21917/ijivp.2022.0398.

[32] N.R.Soora, S.Vodithala, J.Sri .Hari. Badam, "Filtering Techniques to remove Noises from an Image". In: 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI). IEEE, 2022. p. 1-9, Chennai, India, https://doi.org/10.1109/ACCAI53970.2022.9752476.

[33] D. Moura JC, D. Barros DM, Ângelo G, da Nóbrega S, & de Medeiros Valentim RA,"Vessel Segmentation in Fundus Images using Frangi Filter and K-Means: Preliminary results". Anais do I Simpósio de Inovação em Engenharia Biomédica-SABIO 2017, 33.

[34] M. Nixon, A.Alberto ,"Feature extraction and image processing for computer vision",Academic press, 2019,ISBN:978-0-12-814976-8,London.

[35] R. C. Gonzalez, R.E. Woods," Digital Image Processing",in 2008, Edn. 3, Pearson Education. Prentice Hall, PP. 742-747, pages(36).

[36] H.J.A.M.Heijmans,"mathematical morphology",in 2020,vol:216,ISSN:1076-5670.

[37] J.Serra ,"Mathematical morphology",in 2022, Encyclopedia of Mathematical Geosciences,Cham: Springer International Publishing, p. 1-16, France, https://doi.org/10.1007/978-3-030-26050-7_22-2.

[38] A.V.Mühlen, André,"Computer image analysis of malarial Plasmodium vivax in human red blood cells." California Polytechnic State University, San Luis Obispo (2004).

[39] M.A.Kumar,et al,"Image processing based on adaptive morphological technique" , in 2019 International Conference on Emerging Trends in Science and.Engineering(ICESE), India,IEEE,p.1-4,

https://doi.org/10.1109/ICESE46178.2019.9194641.

[40] A.H.Abbas, Aryan A. Kareem,, M.Y. Kamil,"Breast cancer image segmentation using morphological operations." Inte,rnational Journal of Electronics and Communication Engineering & Technology 6.4 (2015): 8-14. 2.

[41] A.M.Raid, W.M.Khedr,, M.A.El-dosuky, M.Aoud, “Image restoration based on morphological operations”,in 2014,International Journal of Computer Science, Engineering and Information Technology (IJCSEIT) 4.3 (2014): 9-21.

[42] E.R.Davies,“Computer vision: principles, algorithms,applications,learning”, Academic Press, London,2017.

[43] J.Heidari,“Classifying Material Defects with convolution Neural Networks and Image Processing”,in 2019, ISSN: 1401-5757, UPTEC F 19026,Uppsala Universitet, Sweden.

[44] S.Badillo, B.Banfai, F.Birzele, I.I. Davydov, L.Hutchinson, T.Kam-Thong, J.Siebourg-Polster, B.Steiert, J.David. Zhang,”An introduction to machine learning. Clinical pharmacology & therapeutics”,in 2020,Clincal Pharmacology&Therapeutics,107(4), 871-885, https://doi.org/10.1002/cpt.1796.

[45] R. Gentleman and V. J. Carey, “Unsupervised Machine Learning,” in Bioconductor Case Studies, New York, NY: Springer New York, 2008, pp. 137–157.

[46] C. Szepesvári, “Algorithms for Reinforcement Learning,” Synth. Lect. Artif. Intell. Mach. Learn., vol. 4, no. 1, pp. 1–103, Jan. 2010, https://doi.org/10.2200/S00268ED1V01Y201005AIM009.

[47] S.Xavier,“Face recognition using Deep Learning.”,in 2017, Catalonia: Polytechnic University of Catalonia 78 (2017).

[48] F,D,Frate, et al,"Use of neural networks for automatic classification from high-resolution images", IEEE transactions on geoscience and remote sensing 45.4

(2007): 800-809, https://doi.org/10.1109/TGRS.2007.892009.

[49] B.Karlik, A .Vehbi Olgac, “Performance analysis of various activation functions in generalized MLP architectures of neural networks”, International Journal of Artificial Intelligence and Expert Systems 1.4 (2011): 111-122.

[50] L.Wu,“Biomedical Image Segmentation and Object Detection Using Deep Convolutional Neural Networks”, in 2019,Diss. Purdue University Graduate School, Indiana, https://doi.org/10.25394/PGS.8051702.v1.

[51] S.Kaliyugarasan, “Deep transfer learning in medical imaging”,in 2019, MS thesis, The University of Bergen.

[52] E.Calli,“Faster Convolutional Neural Networks”,in 2017, M.Sc. Thesis, Faculty of Social Sciences, Radboud University, Nijmegen.

[53] C.alli, E. (2017). Faster Convolutional Neural Networks., M.Sc. Thesis, Faculty of Social Sciences, Radboud University, Nijmegen.

[54] N.Mattsson,et al, “Plasma tau in Alzheimer disease”,In 2016, Neurology, 87(17), 1827-183, https://doi.org/10.1212/WNL.0000000000003246.

[55] M.Legaard, S. Thomas ,S. Gerald , D. Ján, F. Basak , G. Cláudio, I.Alexandros , A. Mahdi, G.Larsen.Peter, “Constructing neural network-based models for simulating dynamical systems”,In 2021, arXiv preprint arXiv:2111.01495, https://doi.org/10.48550/arXiv.2111.01495.

[56] A.Al-Waisy, R.Qahwaji, S.Ipson, S.Al-Fahdawi & T.A. M. Nagem, “A multi-biometric iris recognition system based on a deep learning approach”,In 2018, Pattern Analysis and Applications, 21(3).https://doi.org/10.1007/s10044-017-0656-1 .

[57] X.Dou, I. Hasa, D. Saurel, C.Vaalma, L.Wu, D.Buchholz, D.Bresser, S.Komaba, S.Passerini”, Hard carbons for sodium-ion batteries: Structure, analysis, sustainability, and electrochemistry”,In 2019, Materials Today, 23, 87-

104,https://doi.org/10.1016/j.mattod.2018.12.040

[58] T.Tran, et al, “Impacts and interactions of COVID-19 response involvement, health-related behaviors”,In 2020, health literacy on anxiety, depression and health-related quality of life among healthcare workers: a cross-sectional study. BMJ open, 10(12), e041394, http://dx.doi.org/10.1136/bmjopen-2020-041394 .

[59] M.Xia, T Li, L Xu, L Liu, CW De Silva,”Fault diagnosis for rotating machinery using multiple sensors and convolution neural networks”,In 2017, IEEE/ASME transactions on mechatronics”,23(1),101-110, https://doi.org/10.1109/TMECH.2017.2728371.

[60] I.Goodfellow, Y.Bengio, & A.Courville, “ Deep learning”,In 2016, MIT.

[61] U.Karn, U. “An intuitive explanation of convolution neural networks”,In 2016, The data science blog.

[62] I.Moravcik, I., et al,” Microstructure and mechanical properties of Ni1”,In 2017, 5Co1, 5CrFeTi0, 5 high entropy alloy fabricated by mechanical alloying and spark plasma sintering. Materials & Design, 119, 141-150, https://doi.org/10.1016/j.matdes.2017.01.036.

[63] M.Alradad, “ Robust classification with convolutional neural networks”,In 2015, (Doctoral dissertation, University of Missouri-Columbia).

[64] W.Gong,H.Chen, Z.Zhang,M.Zhang,R.Wang ,C.Guan ,Q.Wang,” Anovel deep learning method for intelligent fault diagnosis of rotating machinery based onimproved CNN-SVM and multichannel data fusion”, Sensors,in 2019, 19(7), 1693,https://doi.org/10.3390/s19071693.

[65] M.Shaha, M. Pawar,“ Transfer learning for image classification”. In 2018 second international conference on electronics, communication and aerospace technology (ICECA) (pp. 656-660), IEEE, Coimbatore, India, https://doi.org/10.1109/ICECA.2018.8474802.

[66] I.Goodfellow, Y. Bengio, A.Courville, “Regularization for deep learning”,In 2016, Deep learning.

[67] A. Saleh, M. B. W. “ Spoken Arabic digits recognition using deep learning”,In 2018, Malaya, https://doi.org/10.1109/I2CACIS.2019.8825004.

[68] M .Khalaf, B.N. Dhannoon,” MSRD-Unet: MultiscaleResidual Dilated U-Net for Medical Image Segmentation”. Baghdad Science Journal.2022, DOI: https://doi.org/10.21123/bsj.2022.7559 .

[69] S M. Zaki, M M Jaber, M A. Kashmoola, “Diagnosing COVID-19 Infection in Chest X-Ray Images Using Neural Network”.Baghdad Science Journal.2022’DOI: https://doi.org/10.21123/bsj.2022.596 .

الخلاصـــة

يمكن النظر إلى تضخم القلب على أنه مؤشر على حالة طبية وليس مرضًا منفصلاً. في نطاق التقييمات التشخيصية ، تشير الزيادة في حجم القلب لصورة الصدر بالأشعة السينية إلى وجود تضخم في القلب. تشمل مسببات تضخم القلب مجموعة من العوامل المسببة ، بما في ذلك ارتفاع ضغط الدم ومرض الشريان التاجي والالتهابات والتشوهات الوراثية واعتلال عضلة القلب. بسبب عدم وجود مؤشرات واضحة خلال المراحل المبكرة ، يمكن أن تتراوح عواقب تضخم القلب من مؤقتة إلى دائمة ، مع مضاعفات خطيرة بما في ذلك السكتة القلبية والموت المفاجئ وفشل القلب. إن التحديد والتشخيص الدقيق لتضخم القلب في الوقت المناسب لهما أهمية قصوى في تسهيل أساليب العلاج الفعالة وتحسين نتائج المرضى.

الهدف من الرسالة هو تقديم طريقة آلية قائمة على التعلم العميق (CNN-VGG19) لتصنيف تضخم القلب، وذلك باستخدام مجموعة بيانات الصدر بالأشعة السينية لـ (المعاهد الوطنية للصحة)، وهي متاحة على Kaggleوالذي يتكون من (5552) صورة أشعة سينية للصدر, تشمل مجموعة البيانات هذه كلاً من الحالات العادية (2776) و (2776) حالة تضخم القلب.

يحتوي النظام المقترح على أربع مراحل, المرحلة الأولى هي مرحلة المعالجة المسبقة ، والتي تبدأ بتغيير حجم صورة الصدر بالأشعة السينية إلى (128 * 128) وقصها ، وتحسين صورة الصدر بالأشعة السينية في (التباين ، والحدة ، والسطوع) ، وأخيراً إزالة الضوضاء.

تتضمن المرحلة الثانية إنشاء أقنعة لصور تضخم القلب ، مما يتيح تحديد المناطق ذات الأهمية التي تتوافق مع تضخم القلب في صور الأشعة السينية للصدر. تستلزم هذه العملية استخدام العتبة والمرشحات ، بما في ذلك مرشحات Gaussian و Morphology ، لتعزيز وضوح وعزل مناطق تضخم القلب.

المرحلة الثالثة هي الاستخراج الدقيق للميزات ذات الصلة من صور تضخم القلب باستخدام نموذج (VGG19).

تتضمن المرحلة الرابعة تصنيف صور الصدر بالأشعة السينية، والتي تتم في فئتين متميزتين. الفئة الأولى هي التصنيف الثنائي، الذي يقسم الصور إلى تصنيفين "عادي" و "تضخم القلب". الفئة الثانية متعددة التصنيفات، والتي تركز على مراحل تضخم القلب، والتي تشمل خفيفة، ومتوسطة، وشديدة.لتقييم كفاءة النموذج المقدم، تم استخدام معايير التقييم بما في ذلك الدقة والحساسية والدقة ودرجة Fscore كمقاييس أساسية. تمت مناقشة النتائج واختبار كافة جوانب النظام المقترح وجميع مراحله بشكل متعمق وأظهرت النتائج دقة كبيرة في تصنيف تضخم القلب للفئة الثنائية والفئات المتعددة ووصلت إلى (99%) (97%) على التوالى.

الطريقة المقترحة سريعة جدًا في التشخيص، يستغرق وقت التنبؤ لصورة الأشعة السينية للصدر حوالي (0.06) ثانية لصورة واحدة.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة بابل-كلية العلوم البنات

قسم علوم الحاسبات

تصنيف مرض تضخم القلب باستخدام طريقة CNN-VGG19 للتعلم العميق

رسالة مقدمة الى

مجلس كلية العلوم للبنات- جامعة بابل

وهي جزء من متطلبات نيل درجة الماجستير في علوم الحاسبات

من قبل

دينا احمد محمد حسين

باشراف

د.ايناس حمود السعدي

1445هـ

2023م