ALMAKI ABDUSALAM SAAD SHELAG

CLASSIFICATION OF DISEASES ON CHEST X-RAYS
USING DEEP LEARNING

NEU
2018

CLASSIFICATION OF DISEASES ON CHEST X-RAYS USING DEEP LEARNING

A THESIS SUBMITTED TO THE
GRADUATE SCHOOL OF APPLIED SCIENCES
OF
NEAR EAST UNIVERSITY

By
ALMAKI ABDUSALAM SAAD SHELAG

In Partial Fulfillment of the Requirements for
the Degree of Master of Science
in

Electrical and Electronics Engineering

NICOSIA, 2018

CLASSIFICATION OF DISEASES ON CHEST X-RAYS USING DEEP LEARNING

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By
ALMAKI ABDUSALAM SAAD SHELAG

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Electronics Engineering

NICOSIA, 2018

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, last name:

Signature:

Data:

ACKNOWLEDGMENTS

All praises and thanks to Allah. It is by His grace that I have been able to access this point in my life.

I would like to express my sincere gratitude to my supervisor, Assist. Prof. Dr. Sertan Kaymak who has supported and directed me with his vast knowledge and also for his patience that ensured the completion of this thesis.

I dedicate my success to the pure spirit of my father, who always supported me in my studies.

I would like to thank the Ministry of Higher Education Tripoli, Libya, for affording me the opportunity of studying in Near East University.

My appreciation also goes to all the lecturers in Near East University who taught me during my master's study period at the university.

Finally, I thank to my friends who supported me in every possible way.

To my family and parents.......

ABSTRACT

Doctors and radiologists are still using manual and visual manners in ordert to diagnose the chest radiographs. Thus, there is a need for an intelligent and automatic system that has the capability of diagnosing the chest X-rays. This thesis aims to employ a deep neural network named as stacked auto-encoder for the classification of chest X-rays into normal and abnormal images. The stacked auto-encoder is trained and tested on chest X-rays obtained for publoc databses which contain normal and abnormal radiographs. A perfromance based camparison is carried out between two networks where the first one uses input chest X-rays without processing or enhancement and the other one uses input images that are processed and enhancd using histogram equalization.

Experimentally, it is concluded that the Stacked auto-encider chieved a good genralization power in diagnosing the unseen chest X-rays into normal or abnormal. Moreover, it is seen that the enhancement of images using hitogram equzalition helps in improving the learning and performance of network due to the rise in the accuracy achieved when image are enhanced.

Keywords: Deep network; stacked auto-encoder; radiographs; classification; generalization; intelligent

ÖZET

Doktorlar, göğüs filmlerine teşhis koyabilmek için halen elle kullanılan ve görüntülü cihazlar kullanmaktadırlar. Fakat, bu tarz göğüs filmlerine teşhis koyabilecek akıllı ve otomatik sistemli sistemlere ihtiyaç duyulmaktadır. Bu tez aracılığı ile normal ve anormal göğüs X-ray filmlerinin sınıflandırılması için düşünülmüş oto-kodlayıcı isimi verilen bağımsız çalışma ağının detaylı olarak incelenmesi hedeflenmiştir. Bu oto kodlayıcı cihaz devlet verilerinden elde edilmiş olan normal ve anormal röntgen filmlerini içeren vakalar üzerinde denenmiş ve test edilmiştir. Karşılaştırmalı uygulama yöntemiyle, işlenmeden veya yükseltilmeden verilen göğüs X-ray filmleri ile histogram dengeleme kullanılarak işlenmiş ve yükseltilmiş görseller arasındaki bağ incelenmiştir.

Deneysel çalışma ile oto-kodlayıcı cihazın, normal ve anormal görünmez X-ray filmlerini teşhisinde genelleme yönünden güzel bir başarı elde ettiği ortaya konmuştur. Histogram dengeleme ile yükseltilen görseller aracılığı ile yükseltilip geliştirilmiş olan göğüs filmleri ile aynı doğrultuda tutarlılık oranın da arttığı ve bu bağı öğrenmede ve uygulamada yardım sağladığı gözlemlenmiştir.

Anahtar Kelimeler: Detaylı ağ; oto-kodlayıcı; röntgen filmleri; sınıflandırma; genelleme; akıllı

TABLE OF CONTENT

ACKNOWLEDGMENTS .... **iv**

ABSTRACT .... **vi**

ÖZET .... **vii**

TABLE OF CONTENT .... **viii**

LIST OF TABLES .... **xi**

LIST OF FIGURES .... **xii**

LIST OF ABBREVIATIONS .... **xiii**

CHAPTER 1: INTRODUCTION **1**

1.1 Introduction 1

1.2 Aims of Thesis 2

1.3 Significance of the Study 2

1.4 Thesis Structure 3

CHAPTER 2: RADIOGRAPHY OVERVIEW **4**

2.1 Introduction 4

2.2 Chest X-rays 5

2.3 Chest Abnormalities 6

2.3.1 Pleural disease 6

2.3.2 Pneumothorax 7

2.3.3 Asbestos plaques 8

2.3.4 Pleural effusions 8

CHAPTER 3: LITERATURE REVIEW **10**

3.1 Introduction 10

3.2 Related Works 10

3.3 Features Extraction in Medicine 11

3.4 Image Processing 13

3.4.1 Image enhancement 14

3.5 Artificial Neural Network (ANN) 15

3.6 Structure of ANN 16

3.6.1 ANN Layers 16

3.7 Supervised and Unsupervised Learning 18

3.7.1 Supervised learning rules 19

3.8 Learning Parameters for the Back Propagation Algorithm 21

3.9 Deep Learning 24

3.9.1 Stacked Auto-encoder 25

3.9.2 Training 29

CHAPTER 4: NETWORKS TRAINING AND PERFORMANCE EVALUATION 30

4.1 Methodology 30

4.2 Database 31

4.3 Training the Deep Models 33

4.3.1 SAE1 Training 35

4.3.1 SAE2 Training 38

4.4 Deep Models Testing 41

4.5 Results Discussion 43

4.6 Results Comparison 44

CHAPTER 5: CONCLUSION.. 46

5.1 Conclusion.. 46

REFERENCES.. 46

APPENDICES

Appendix 1: Image Processing Code 50

Appendix 2: Neural Networks Code 56

LIST OF TABLES

Table 1: Dataset 1 and data division…………………………………………….………..46

Table 2: Dataset 2 and data division………………………………………………………...46

Table 3: Training and testing data from the two databases…………………………….………47

Table 4: Learning parameters of networks during pre-training and fine-tuning……………..48

Table 5: Training network performance……………………………………………….……..50

Table 6: Learning parameters during pre-training and fine-tuning of SAE2………………...52

Table 7: Learning results of SAE2…………………………………………………….…..54

Table 8: Classification rates of both networks during testing……………….……………56

Table 9: Performance comparison with and without enhancement…………..………………57

Table 10: Results comparison…………….……………………………………….…………58

LIST OF FIGURES

Figure 1: Pleural thickening 6

Figure 2: Pneumothorax 7

Figure 3: Asbestos related pleural plaques 8

Figure 4: Pleural effusion 9

Figure 5: Medical Image Enhancement 15

Figure 6: Artificial Neural Network's Basic Structure 16

Figure 7: The ANNs structure showing the three layers 17

Figure 8: Deep network structure 24

Figure 9: Auto-encoder step 1 26

Figure 10: Auto-encoder step 2 27

Figure 11: Softmax classifier 28

Figure 12: Stacked Auto-encoders 28

Figure 13: Flowchart of the proposed methodology 31

Figure 14: A sample of the database images 32

Figure 15: The deep network model 1 (SAE1) structure 35

Figure 16: Learning curve of SAE1 during pre-training 36

Figure 17: Learning curve of SAE1 during fine-tuning 37

Figure 18: SAE2 network architecture 38

Figure 19: Learning curve of SAE2 during pre-training 39

Figure 20: Learning curve of SAE2 during fine-tuning 40

Figure 21: A sample of testing chest X-rays 41

Figure 22: A sample of some misclassified images 42

LIST OF ABBREVIATIONS

ANN:	Artificial Neural Network
AE:	Auto-Encoder
SAE:	Stacked Auto-Encoder
BPNN:	Back Propagation neural network
MSE:	Mean Square Error
SEC:	Second
SVM:	Support Vector Machine
CNN:	Convolutional Neural Network

CHAPTER 1
INTRODUCTION

1.1 Introduction

Medical X-rays are images are generally used to diagnose some sensitive human body parts such as bones, chest, teeth, skull, etc. Medical experts have used this technique for several decades to explore and visualize fractures or abnormalities in body organs (Er et al., 2010). This is due to the fact that X-rays are very effective diagnostic tools in revealing the pathological alterations, in addition to its non-invasive characteristics and economic considerations. Chest diseases can be shown in CXR images in the form of cavitations, consolidations, infiltrates, blunted costophrenic angles, and small broadly distributed nodules. The interpretation of a chest X-ray can diagnose many conditions and diseases such as pleurisy, effusion, pneumonia, bronchitis, infiltration, nodule, atelectasis, pericarditis, cardiomegaly, pneumothorax, fractures and many others (Er et al., 2010).

Classifying the chest x-ray abnormalities is considered a tough task for radiologists. Hence, over the past decades, computer aided diagnosis (CAD) systems have been developed to extract useful information from X-rays to help doctors in having a quantitative insight about an X-ray. However, those CAD systems haven't achieved a significance level to make decisions on the type of conditions of diseases in an X-ray (El-Solh et al., 1999). Thus, the role of them was left as visualization functionality that helps doctors in making decisions.

Recently, accurate images classification has been achieved by deep learning based systems. Those deep networks showed superhuman accuracies in performing such tasks. This success motivated the researchers to apply those networks on medical images for diseases classification tasks and the results showed that deep networks can efficiently extract useful features that distinguish different images classes (Ashizawa et al., 2005). Convolutional neural networks have been applied to various medical images diagnosis and classification due to its power of extracting different level features from images.

Traditional networks have been also used in classifying medical diseases, however, their performance was not as efficient as the deep networks in terms of accuracy, computation time, and minimum square error achieved. In this work, deep learning based networks are employed to classify most common thoracic diseases. Two stacked auto-encoder are examined in this study to classify the chest X-rays into two common classes: normal and abnormal which may have different types of diseases that may be found in chest X-ray, i.e, Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumothorax, Consolidation, Edema, Emphysema, Fibrosis. In this work, we aim to train the deep network on the same number of chest X-ray images and evaluate their performances in classifying different chest X-rays. The data used in obtained from two public databases which are the Shenzhen Hospital X-ray Set / China data set: X-ray images in this data set (Fushman et al., 2005), and the Montgomery County X-ray Set (Fushman et al., 2016).

1.2 Aims of Thesis

Doctors and radiologists are still categorizing the chest X-rays in a manual manner based on some visual examinations. Therefore, There is a need for an automatic and intelligent systems that has the capability of accurate classification of chest X-rays into normal and abnormal images. Thus, in this work we aim to use a powerful deep network in such classification tasks. The deep network that is selected to be used in called the stacked auto-encoder which showed a great efficacy in different classification tasks in the medial field. Furthermore, the network is also examined on enhanced and unenhanced images, which aims to demonstrate the effects of medical image processing and enhancement on the performance of neural network.

1.3 Significance of the Study

As most of the previous work are conducted using convolutional neural networks (CNN) and support vector machine (SVM), this work is ought to investigate the performance of other types of deep networks in classifying the abnormality of chest X-rays. Thus, a deep network named as stacked auto-encoder is selected to be used as the brain behind this study. This network is expected to well perform since it has been applied to various medical classification tasks where it achieved high accuracies and because the data available are enough for training it and achieving a small error.

1.4 Thesis Structure

This thesis is structures as the following:

Chapter one is a general introduction of the thesis where the aims and significance of thesis are discussed.

Chapter two is a general overview of the radiology X-rays and types of chest diseases that may be found in a radiograph.

Chapter three is a detailed review of the artificial neural network and its working principles, in addition to its training algorithm including backpropagation learning technique. Moreover, this chapter discusses the deep learning in particular, stacked auto-encoder and the techniques used for training it.

Chapter four presents the training and testing phase of the network. In this chapter network is trained and tested on images with and without image enhancement and the results are shown and explained. This chapter is also a discussion of the results obtained from the two network simulations. In this chapter, the network that outperformed is discussed.

Finally, **chapter five** is a conclusion of the thesis. it also shows the future recommendations that can be done to enhance the work.

CHAPTER 2
RADIOGRAPHY OVERVIEW

2.1 Introduction

In the territory of human services diagnostics, therapeutic image preparing has assumed a contributory part. From the different scopes of accessible radiological images created from ultrasound, x-beams, attractive Resonance imaging, Computed Tomography, Positron Emission Tomography and so forth, every has its own particular technique for catching the images. Be that as it may, even in the wake of narrowing the focal point of the image catch, just a couple of segments of the radiological images are of clinical significance to the counseling doctor (Fushman et al., 2015). Be that as it may, there are different purposes behind which the pathologist and also radiologist trusted that image produced by such radiological test doesn't yield 100% exact data. For less minor types of the perilous ailment, such blunders may not make any difference much, but rather it does conceivably make a difference generally. In any case, presenting the patient to destructive radiological beams is restoratively not prudent and might be a significant costly issue for both specialist and patient. Consequently, from the previous decades utilization of image handling is progressively used to recognize the issues and settle it. The initial phase in such issue distinguishing proof is to perform image improvement. As though the image with clinical significance isn't upgraded it might possibly prompt exceptions in cutting edge investigation of medicinal information (Fushman et al., 2012).

Henceforth, image upgrade assumes a pivotal part in uncovering the malady with more data to the specialist or to the procedure of further investigation of the illness. This paper talks about the chest x-beam images and proposed an answer with greater activity and lower computational cost for improving the chest x-beams. The radiological images particularly chest x-beam images experiences following issues i.e. I) numerous outline, ii) nearness of rib confines (bones), iii) shadows of bosom in female subjects, iv) stomach and so on. In spite of the fact that there are propelled adaptations of radiological images however chest x-beam image is thought to be essential finding factor by the clinicians. Subsequently, if the chest x-beam images are covered with different relics or issues, post diagnosis will dependably prompt anomalies. Subsequently, it is vital that chest x-beam images ought to be appropriately pre-prepared even before subjecting it to propel investigation.

Therefore, this paper presents a very simple and cost effective chest X-rays classification system using deep networks for chest x-ray images with and without operations of enhancements.

2.2 Chest X-rays

A chest X-beam test is an extremely normal, non-obtrusive radiology test that creates an image of the chest and the inward organs. To deliver a chest X-beam test, the chest is quickly presented to radiation from a X-beam machine and an image is created on a film or into a computerized computer (Jaeger et al., 2014).

Chest X-beam is additionally alluded to as a chest radiograph, chest roentgenogram, or CXR. Contingent upon its thickness, every organ inside the chest pit retains fluctuating degrees of radiation, creating distinctive shadows on the film. Chest X-beam images are highly contrasting with just the brilliance or haziness characterizing the different structures. For instance, bones of the chest divider (ribs and vertebrae) may assimilate a greater amount of the radiation and along these lines, seem more white on the film.

Then again, the lung tissue, which is for the most part made out of air, will enable the majority of the radiation to go through, building up the film to a darker appearance. The heart and the aorta will seem whitish, however typically less brilliant than the bones, which are denser.

Chest X-beams tests are requested by doctors for an assortment of reasons. Numerous clinical conditions can be assessed by this basic radiology test. A portion of the basic conditions recognized on a chest X-beam include:

- pneumonia,
- enlarged heart,
- congestive heart disappointment,
- lung mass,
- rib cracks,

- fluid around the lung (pleural radiation), and
- air around the lung (pneumothorax).

All in all, a chest X-beam test is a straightforward, brisk, economical, and moderately innocuous system with insignificant danger of radiation. It is additionally broadly accessible.

2.3 Chest Abnormalities

2.3.1 Pleural disease

- The pleura and pleural spaces are just noticeable when abnormal
- There ought to be no noticeable space between the instinctive and parietal pleura
- Check for pleural thickening and pleural emanations
- If you miss a strain pneumothorax you hazard your patient's life – and in addition your outcome at finals!
- The pleura just wind up noticeable when there is an abnormality show. Pleural abnormalities can be unpretentious and it is critical to check precisely around the edge of every lung where pleural abnormalities are normally more effortlessly observed (Figure 1). A few illnesses of the pleura cause pleural thickening, and others prompt liquid or air assembling in the pleural spaces (Xue et al., 2015).

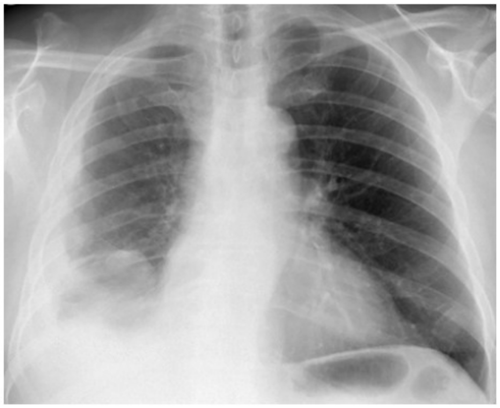


Figure 1: Pleural thickening (Xue et al., 2015)

2.3.2 Pneumothorax

A pneumothorax shapes when there is air caught in the pleural space. This may happen precipitously, or because of hidden lung illness. The most widely recognized reason is injury, with slash of the instinctive pleura by a broken rib (Figure 2).

On the off chance that the lung edge measures in excess of 2 cm from the inward chest divider at the level of the hilum, it is said to be 'substantial. If there is tracheal or mediastinal move far from the pneumothorax, the pneumothorax is said to be under 'pressure.' This is a restorative crisis! Missing a pressure pneumothorax may not just mischief your patient; it is likewise the snappiest method to fizzle the radiology OSCE at finals!

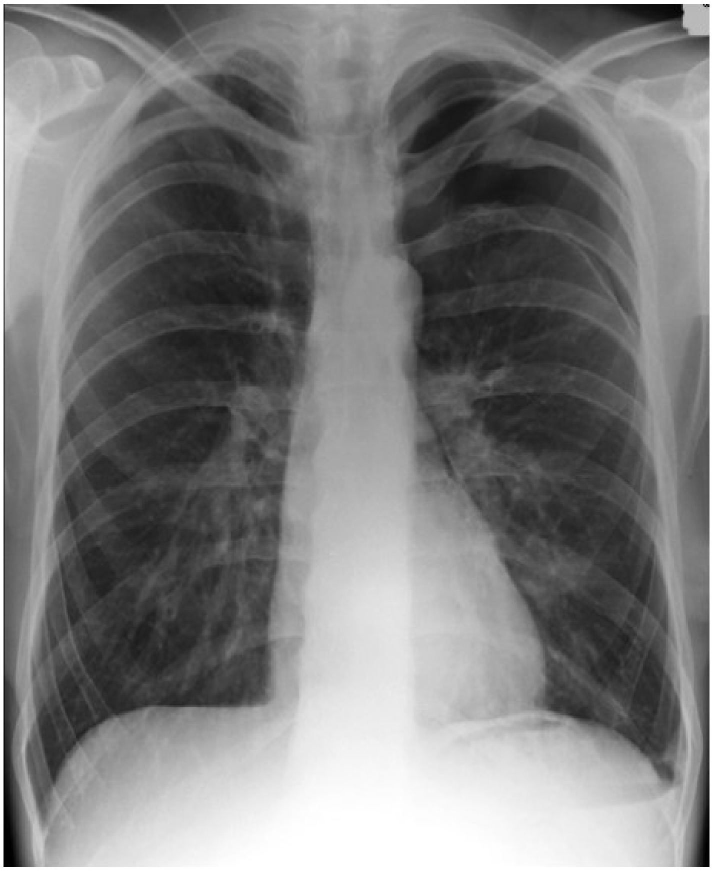


Figure 2: Pneumothorax (Xue et al., 2015)

2.3.3 Asbestos plaques

Calcified asbestos related pleural plaques have a trademark appearance, and are for the most part thought to be favorable. They are sporadic, all around characterized, and traditionally said to look like holly takes off (Candemir, et al., 2014).

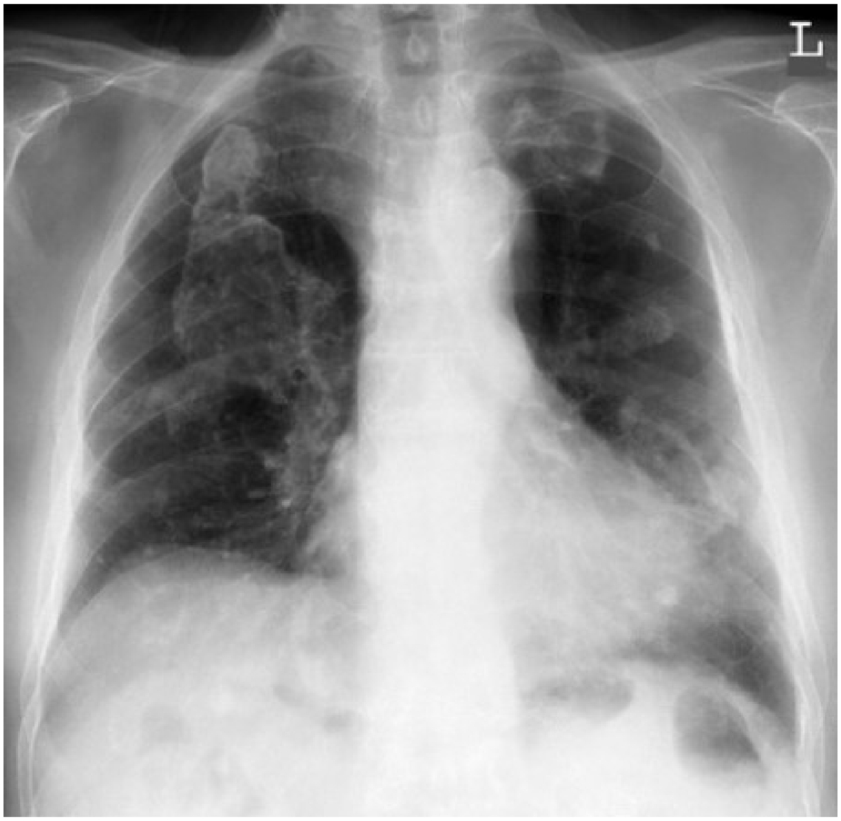


Figure 3: Asbestos related pleural plaques (Candemir, et al., 2014)

2.3.4 Pleural effusions

A pleural emission or effusion is a gathering of liquid in the pleural space. Liquid accumulates in the most minimal piece of the chest, as per the patient's position. In the event that the patient is upright when the X-beam is taken, at that point liquid will encompass the lung base framing a 'meniscus' – a sunken line clouding the costophrenic edge and part or the majority of the hemidiaphragm (Figure 4). On the off chance that a patient is recumbent, at that point a pleural radiation layers along the back part of the chest pit and ends up hard to see on a chest X-beam (Shiraishi et al., 2000).

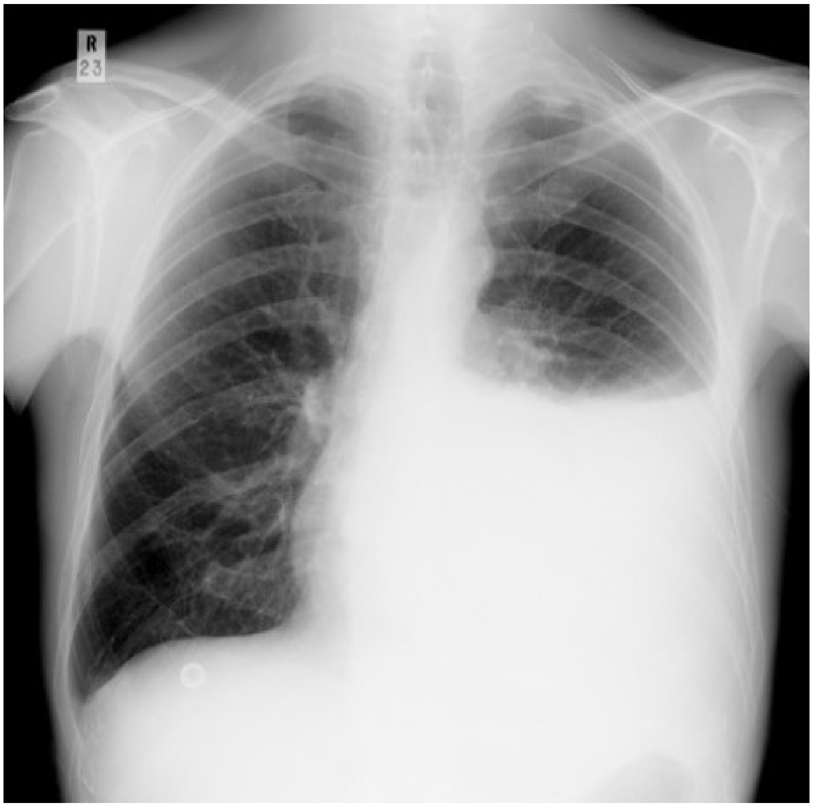


Figure 4: Pleural effusion (Shiraishi et al., 2000)

CHAPTER 3
LITERATURE REVIEW

3.1 Introduction

In this chapter, a brief review of radiography, its application in medical imaging, and diagnosis is presented. Also, the applied approach to the classification of the segmented images, pattern recognition, is introduced. Furthermore, background on related image processing and feature extraction techniques as are considered in this thesis are discussed sufficiently. Artificial neural networks, the backbone of machine learning, and which has also been used extensively in this work for the classification phase are introduced; including the particular algorithms for the supervised and unsupervised learning.

3.2 Related Works

In a past work, (Cernazanu and Holban, 2012), described the segmentation of chest X-ray using convolutional neural network. In their work, they introduced image segmentation into bone tissue and non-bone tissue. The aim of their work was to develop an automatic or an intelligent segmentation system for chest X-rays. The system was established to have the capability to segment bone tissues from the rest of the image.

They were able to achieve the aim of the research by using a convolutional neural network, which was tasked with examining raw image pixels and hence classifying them into “bone tissue” or “non-bone tissue”. The convolutional neural networks were trained on the image patches collected from the chest X-ray images.

It was recorded in their work that the automatic segmentation of chest X-rays using the convolutional neural networks, and approaches suggested in their research produced plausible performance.

In another recent research, “lung Cancer Classification using Image Processing”, presented the application of some image processing techniques in the classification of patients chest X-rays into whether cancer is present or not (benign or malignant). In this work, it was shown that by extracting some geometric features that are essential to the classification of the images such area, perimeter, diameter, and irregularity; an automatic classification system was developed.

Furthermore, in the same research, texture features were considered for a parallel comparison of results on the classification accuracy. The texture features used in the work are average gray level, standard deviation, smoothness, third moment, uniformity, and entropy. The back propagation neural network was used as the classifier, and an accuracy of 83% was recorded in the work (Patil and Kuchanur, 2012).

In this thesis the classification of chest X-ray radiographs into two classes has been achieved using artificial neural networks. The two classes are the normal one which has no disease or conditions, and the abnormal one which may have any types of diseases that may encounter the chest organs including heart, lungs. Chest etc… A deep network called Stacked Auto-encoder is used for (SAE), which relies on a supervised and unsupervised learning algorithms was used to train the network on the images collected for the research.

3.3 Features Extraction in Medicine

Pattern recognition is the process of developing systems that have the capability to identify patterns; while patterns can be seen as a collection of descriptive attributes that distinguishes one pattern or object from the other. It is the study of how machines perceive their environment, and therefore capable of making logical decisions through learning or experience. During the development of pattern recognition systems, we are interested in the manner in which patterns are modeled and hence knowledge represented in such systems. Several advances in machine vision have helped revamp the field of pattern recognition by suggesting novel and more sophisticated approaches to representing knowledge in recognition systems; building on more appreciable understanding of pattern recognition as achieved in the human visual processing.

Typical pattern recognition as the following important phases for the realization of its purpose for decision making or identification.

- Data acquisition: This is the stage in which the data relevant to the recognition task are collected.

- Pre-processing: It is at this stage that the data received in the data acquisition stage is manipulated into a form suitable for the next phase of the system. Also, noise is removed in this stage, and pattern segmentation may be carried out.
- Feature extraction/selection: This stage is where the system designer determines which features are significant and therefore important to the learning of the classification task.
- Features: The attributes which describe the patterns.
- Model learning/ estimation: This is the phase where the appropriate model for the recognition problem is determined based on the nature of the application. The selected model learns the mapping of pattern features to their corresponding classes.
- Model: This is the particular selected model for learning the problem, the model is tuned using the features extracted from the preceding phase.
- Classification: This is the phase where the developed model is simulated with patterns for decision making. The performance parameters used for accessing such models include recognition rate, specificity, accuracy, and achieved mean squared error (MSE).
- Post-processing: The outputs of the model are sometimes required to be processed into a form suitable for the decision making phase stage. Confidence in decision can be evaluated at this stage, and performance augmentation may be achieved.
- Decision: This is the stage in which the system supplies the identification predicted by the developed model.

There exist several approaches to the problem of pattern recognition such as syntactic analysis, statistical analysis, template matching, and machine learning using artificial neural networks.

Syntactic approach uses a set of feature or attribute descriptors to define a pattern, common feature descriptors include horizontal and vertical strokes, term stroke analysis; more compact descriptors such as curves, edges, junctions, corners, etc., which is termed geometric features analysis. Generally, it is the job of the system designer to craft such rules that distinguish one pattern or object from another. The designer is meant to explore attribute descriptors which are unique to identify each pattern, and where there seems to a conflict of identification rules such as can be observed in identifying Figure 6 and 9; they have same geometric feature descriptors save that one is the inverted form of the other, the system designer is meant to explore other techniques of resolving such issues (Yumusak and Temurtas, 2010).

Statistical pattern analysis uses probability theory and decision to infer the suitable model for the recognition tasks.

Template pattern matching uses the technique of collecting perfect or standard examples for each distinct pattern or object considered in the recognition task. It is with these perfect examples that the test patterns are compared. It is usually the work of the system designer to craft the techniques with which pattern variations or dissimilarities from the templates are measured, and hence determine decision boundaries as to accept or reject a pattern being a member of a particular class. Euclidean distance is a common used function to measure the distance between two vectors in n-dimensional space.

Template matching can either be considered as global or local depending on the approach and aim for which the recognition system is designed. In global template matching, the whole pattern for recognition is used to compare the whole perfect example pattern; whereas in local template matching, a region of the pattern for classification is used to compare a corresponding region in the perfect template.

Artificial neural networks, on the other hand, are considered intelligent pattern recognition systems due to their capability to learn from examples in a phase known as training. These systems have sufficed in lots of pattern recognition systems; the ease with which same learning algorithms can be applied to various recognition tasks is motivating.

In this approach, the designer is allowed to focus on determining features to be extracted for learning by the designed systems, rather than expending a huge amount of time, resources, and labour in understanding the whole details of the application domain; instead, the system learns relevant features that distinguish one pattern from the other (Yumusak and Temurtas, 2010).

3.4 Image Processing

An image can be considered as a visual perception of a collection of pixels; where, a pixel can be seen as the intensity value at a particular coordinate in an image. Generally, pixels are described in 2D, such as f(x,y).

The pixel values can vary in an image depending on the number gray levels used in the image. The range of pixels can be expressed as 0 to 2^m, for an image with gray level of m. Image processing is a very important of computer vision, as image data can be suitably conditioned before machine learning.

3.4.1 Image enhancement

Image processing has been extensively used in medicine. Image enhancement is always the most common process needed in this field. A medial image contains many parts and may have lot of noise. This makes it very tough for doctors to find the correct diagnosis of it. Image processing can be useful tool in this case as it helps in detecting and enhancing the images since all parts in image including noise differ from each other's in terms of brightness and intensities. Thus, in this work, image processing tools are used in order to enhance the chest X-ray images and remove the noise that may be found in them. This is done by using many techniques for image enhancement such as filtering, histogram equalization, and intensity adjustment. An example of the working principle of the proposed algorithm is shown in Figure 5 (Yumusak and Temurtas, 2010).

In case of filtering, many filters can be used such as median, mean, Gaussian filters. For median filters, the images are filtered as some of them have noise artifacts which should be removed to enhance the quality of images. Median filter is a good technique for removing noise as it provides good rejection of the Salt and Pepper noise which is found in some medical images.

Moreover, image intensities adjustment can be also used for enhancing the quality of images. This technique involves the mapping of the pixels intensity distribution form one level to another level. To highlight the images more and more, the intensities of pixels are increased by mapping them into other values. This ended up with brighter images where the cells are clearer; including the cancerous cells.

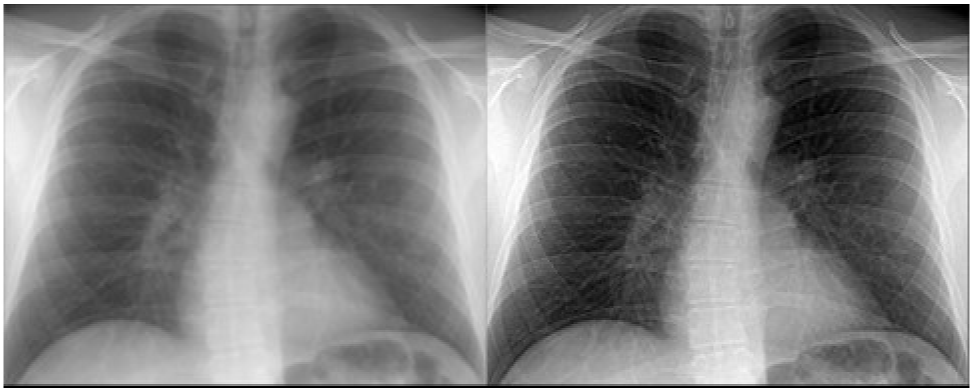


Figure 5: Medical Image Enhancement (Yumusak and Temurtas, 2010)

3.5 Artificial Neural Network (ANN)

Artificial neural systems are structure that originated from the cerebrum of the human brain that is used for reasoning. The structure has been used to deal with troublesome issues in science. The vast majority of the structures of neural systems are like the organic mind in the requirement for preparing before having the capacity to complete a required assignment (Yumusak and Temurtas, 2010). Like the standard of the human neuron, neural system processes the aggregate of every one of its data sources.

On the off chance that that aggregate is more than a decided level, the journalist yield would then be able to be enacted. Something else, the yield isn't going to the actuation work. Figure 6 illustrates the principal assembly of the neural system where the source of the weight and info on summation of work is shown. The quantity of neuron that is find in a structure can is referred to as the yield work. The equation that is used in the calculation of initial work is precisely explained in (Santos et al., 2004):

$$TP = \sum x_n \omega_n \qquad (3.1)$$

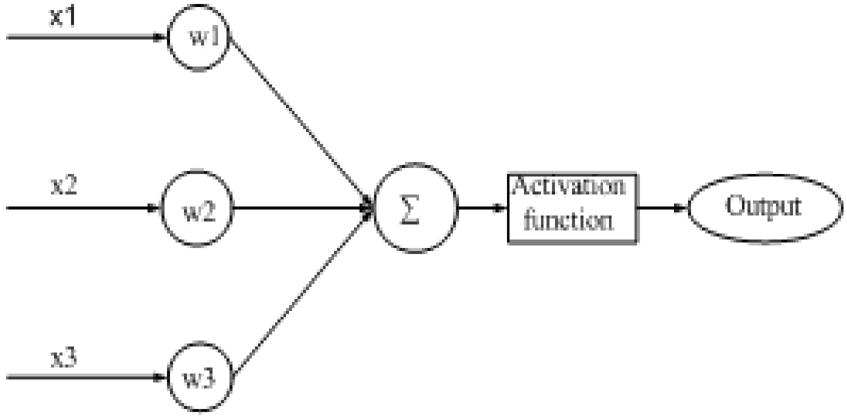


Figure 6: Artificial Neural Network's Basic Structure (Santos et al., 2004)

3.6 Structure of ANN

The ANNs structure contains three layers despite the learning technique. These angles are the layers, weights, and initiation capacities. Every last one of these three sections play an imperative lead in the ANN limit. The three sections or segment works collectively to ensure proper working of the system (Santos et al., 2004).

3.6.1 ANN Layers

The mutual relationship that occurs between the layers of ANN is the major derivative to its creations. The layers interact by sending information between each other using the synaptic weight. The Ann structure can be subdivided into three layers that is listed in the subsequent section below.

1. Input layer: This is the first layer that is found in the neural system of ANN. This layers is major that send information or data to other layers in the neural system. It can be regarded as sensors because it doesn't process later but only pass information processed by other layers.

2. Hidden layers: this can be regard as the central bit of the neural system. It involves no less than one of the layers which is the input layer and the neural layer. This layer transmits the data to the output layers. The Hidden layer can be regards as the intermediate layers or as a principal layer because the synaptic weights found in it are reliable.

3. Output layer: This layer is regard as the output layers because its last contact where the results of the neural system are gotten, the output layer got its information that is processed from the Hidden layer.

Figure 7: The figure below shows the neural system and the interactions that occurs between its three layers. The first layer which is the input layers is the source of the data that is passed to the hidden layer and later to the output layer. The yield or result of the neural system is gotten from the output layer.

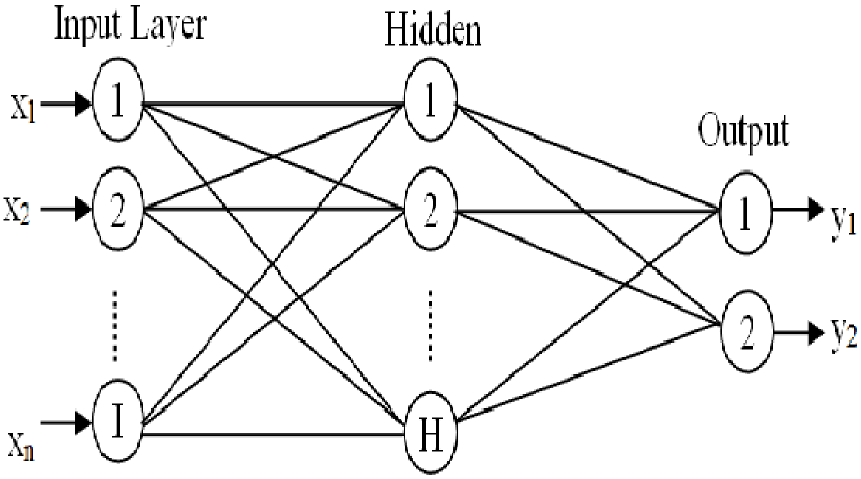


Figure 7: The ANNs structure showing the three layers (Santos et al., 2004)

3.7 Supervised and Unsupervised Learning

The phase of building knowledge into neural networks is called learning or training. The three basic types of learning paradigms are:

-**Supervised learning**: The network is given examples and con-currently supplied the desired outputs; the network is generally meant to minimize a cost function in order to achieve this, usually an accumulated error between desired outputs and the actual outputs.

- Training data includes both the input and the desired results.

• For some examples the correct results (targets) are known and are given in input to the model during the learning process.

• These methods are usually fast and accurate.

• Have to be able to generalize: give the correct results when new data are given in input without knowing a priori the target.

Error per training pattern = desired output - actual output

Accumulated error= ∑(error of training patterns)

-**Unsupervised learning**: The network is given examples but not supplied with the corresponding outputs; the network is meant to determine patterns between the input attributes (examples) according to some criteria and therefore group the examples thus.

• The model is not provided with the correct results during the training.

• Can be used to cluster the input data in classes on the basis of their statistical properties only.

• Cluster significance and labeling.

• The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

3.7.1 Supervised learning rules

- Perceptron learning rule

There are several different models of supervised learning that have been implemented in artificial neural networks (Santos et al., 2004).

$$T.P = \sum_{i=1}^{m} w_{ji} x_i \tag{3.2}$$

$$\begin{aligned} &\mathit{If}\ T.P \geq \theta, \quad then\ y = 1 \\ &else\ \ T.P < \theta,\ then\ y = 0 \end{aligned} \tag{3.3}$$

Where T.P is known as the total potential of the neuron, Ө is the threshold value, w_{ji} is the weight connection from input x_i to neuron j, m is the number of inputs, and y is the output of the neuron. If the total potential is greater than or equal to the threshold value, then the neuron fires; if otherwise, then the neuron does not fire.

The perceptron learning rule is given below; the weights of the network are updated using the equation.

$$w_j = w_j(t) + \eta(d - y)x \tag{3.4}$$

Training patterns are presented to the network's inputs; the output is computed. Then the connection weights w_j are modified by an amount that is proportional to the product of the difference between the actual output, y, and the desired output, d, and the input pattern, x.

- Delta learning rule

An alternative but related approach to the perceptron learning rule is known as the delta rule. While the perceptron training rule is based on the idea of modifying weights according to some fraction of the difference between the output and target, the delta rule is based on the more general idea of "gradient descent". For example, consider the task of training a single TLU with a set of input patterns p, each with a desired target output t_p. The global error E is a function of the weights w. That is, as weights change, the error changes. The goal is to move in "weight space" down the slope of the error function with respect to each weight. The size of the step should be proportional to the magnitude of the slope. How is the slope calculated? Using calculus the slope may be expressed as the partial derivative of the error with respect to the weight:

$$\Delta w_j = -\alpha \frac{\delta E}{\delta w_j} \tag{3.5}$$

where α is the learning rate

If e_p is the error produced by a network processing a particular pattern p, then the global error E is the mean error produced over all the different patterns in the training set:

$$E = \frac{1}{N} \sum_{p=1}^{N} e^p \tag{3.6}$$

where N is the number of patterns in the training set.

The simplest way of determining the pattern error e_p is simply the target output minus the actual output:

$$e^p = t^p - y^p \tag{3.7}$$

where y is the neuron output and t is the target for training pattern p.

However, the above equation has several problems. First the subtraction means that the term may be either positive or negative rather than a simple magnitude and may therefore complicate further calculations. This issue is managed by squaring the term:

$$e^p = (t^p - y^p)^2 \tag{3.8}$$

The second problem we encounter is more subtle. In order to perform gradient descent values must be continuous. This can be remedied by substituting activation a rather than the output y. Though when doing this the target should be carefully defined - if the threshold is set to 0 then one target should be set as positive and the other negative e.g. -1 and 1.

$$e^p = \frac{1}{2}(t^p - y^p)^2 \tag{3.9}$$

The final modification is to divide the entire term by 2 simply to make differentiation easier.

Since E is the mean of all patterns one cannot technically calculate dE/dw_i until the entire set of patterns is available. However, this is very computationally intensive, so de/dw_i is usually performed individually with each training pattern as an approximation as shown below.

$$\delta e^p / \delta w_i = -(t^p - y^p)x^p \tag{3.10}$$

3.8 Learning Parameters for the Back Propagation Algorithm

- Learning rate

The learning rate is a very important parameter in supervised learning; it is used to control how fast the network learns the training examples. The value of this parameter varies between 0 to 1. Learning rate determines the step size with which network weights are updated during training. If the value set for the learning rate is too high, the network runs a high risk of only memorizing the training data, as learning is completed in fewer epochs (possible that the network weights have not been properly tuned to the examples); a situation referred to as over-fitting. If the value set for the learning rate is too low, then the network runs a risk of not insufficient learning of the training data by the time the set number of maximum epochs is reached. It there follows that using a value that is too small for the learning rate makes the learning much slower, and the network may not converge to the set MSE goal before training is stopped. Generally, the suitable value for the learning rate is determined heuristically (through a trial and error method). Low values are usually preferred (Santos et al., 2004).

- Momentum rate

The momentum parameter is often optional for supervised learning, its sole purpose is to help reduce the possibility of the network getting trapped in a poor local minimum during training. Its value also ranges between 0 to 1. The momentum rate parameter can be seen as kind of inertia being introduced into the network. It helps push the learning past poor local minima during network training; and also dampens oscillations that may occur during learning, hence the learning curve is usually smoother compared to when the momentum rate parameter is not used in the learning algorithm.

- Goal of cost function (MSE)

Generally, for any supervised learning algorithm, since a cost function relating the deviation of the actual response of the network from the desired is to be minimized, it then follows that there should be a specified value for the goal of the cost function being minimized.

When the network reaches this specified value for the MSE goal, the training of the network is stopped ().

- Maximum epochs

Since neural networks learn by examples, the forward pass of an example from the input and the back pass of the computed error constitute what is referred to as an epoch. This process is repeated for each training example till all the examples have been propagated through the network; after which the process repeats in such manner, while the set value for the MSE goal is monitored. When training neural networks, it is very important to specify the maximum number of iterations allowed in the training. This has the effect of not allowing the network to continue training indefinitely in a situation where the learning has not converged to the set MSE goal, hence is used as an important stopping criterion in training.

- Number of hidden neurons

For the back propagation neural network and most other networks, the network is made of at least three layers, which are the input layer, hidden layer, and output layer.

The input layer is where the training examples are supplied to the network, the hidden layer learns the features represent in the input, and the output layer allows the actual output of the network to be obtained. Also, the input layer neurons are non-processing, they basically serve to supply the input features to the network.

The output layer in a supervised learning, allows the computation of the error between the desired output and actual output of the network; and therefore back propagation of errors into the network for weights adjustment or tuning (Patil and Kuchanur, 2012).

The hidden layer is very important, considering that it is where the main knowledge representation of the features present in the training examples is achieved. Hence, it is very crucial that a suitable number of neurons are used at the input layer of neural networks to ensure proper learning of a task.

If the number of hidden neurons is too few, then the network may not have enough power to accommodate the feature representation present in the training examples; a situation also referred to as low degree of freedom.

Conversely, if the number of neurons used is too many, then the network develops a far more complex model to the training examples than is required, the network has too much representation power, such that it may begin to model features that too peculiar to the training examples, hence the network is likely to over-fit. A situation also referred to as too high degree of learning freedom.

It is therefore desirable that the number of neurons used in the hidden layer should not be too few or too many. Generally, during training, the number of suitable hidden neurons is determined through a trial and error approach.

- Activation function

Activation functions are used to squash the output of artificial neurons to within a certain range of values. It is conceived that the output of neurons should not be infinite. The weighted sum of the inputs to a neuron is computed, the value referred to as the total potential, which is then passed through an activation function.

Common activation functions used in neural networks include the Signum, linear, Log-Sigmoid, and the Tan-Sigmoid.

During the design of neural networks, the type of application determines the activation function to be used in each layer of the network. The layer that is so application specific is the output layer. The type of activation used in the output layer depends on the range or type of values expected at the output.

For real values problems such as regression tasks, the linear activation function is used, for classification tasks, output values are generally integers, and hence the Log-Sigmoid or Tan-Sigmoid can be used.

3.9 Deep Learning

Deep Learning is another and progressed documented of Machine Learning. It has been produced and enhanced keeping in mind the end goal to move the moving Machine Learning to be nearer its primary and unique objective Artificial Intelligence (Glavan, Holban, 2012).

Deep Learning is called "Deep" because of its structure of the neural systems. Prior, neural systems used to have two layers deep on the grounds that it was not computationally plausible to assemble bigger systems. These days, a neural system with in excess of 10 layers and considerably more layers are being started and constructed. These sorts of systems are called deep neural systems. Figure 8 shows engineering of a deep neural system. It demonstrates that the system comprises of numerous layers which makes it deep (Deng, 2014).

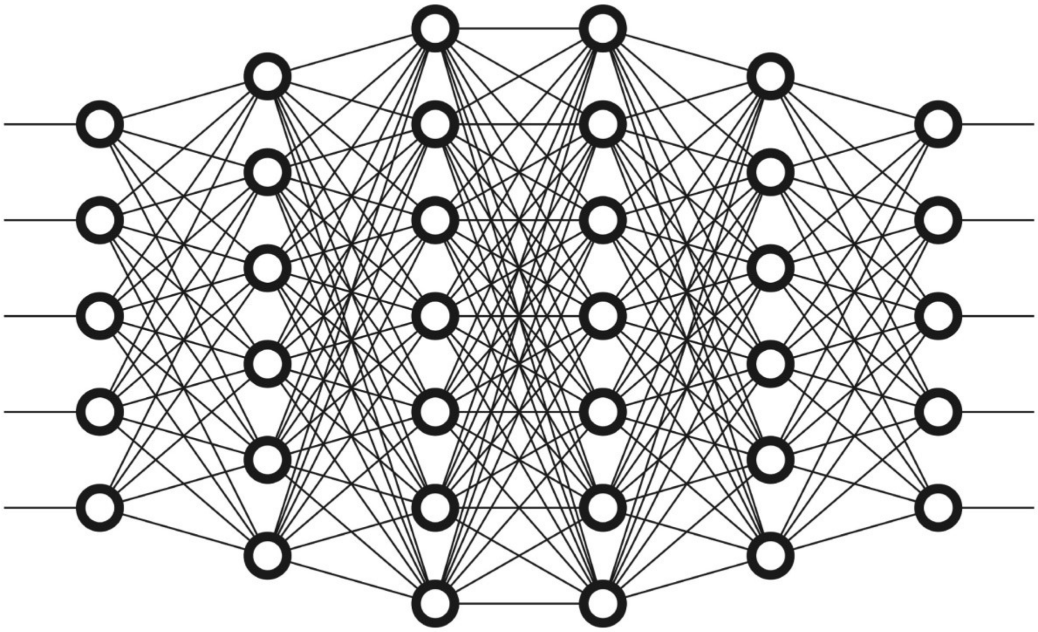


Figure 8: Deep network structure (Glavan, Holban, 2012)

3.9.1 Stacked Auto-encoder

Stacked auto-encoder is one sort of deep systems that is prepared utilizing another calculation called covetous layer wise training. The eager layer wise approach for pre-training a deep system works via training each layer thus. In this page, you will discover how auto-encoders can be "stacked" in an insatiable layer wise design for pre-training (instating) the weights of a deep system (Glavan and Holban, 2012).

A stacked auto-encoder is a neural system comprising of different layers of meager auto-encoders in which the outputs of each layer are wired to the inputs of the progressive layer. Formally, consider a stacked auto-encoder with n layers. Utilizing documentation from the auto-encoder area, let W(k,1),W(k,2),b(k,1),b(k,2) signify the parameters W(1),W(2),b(1),b(2) for kth auto-encoder. At that point the encoding venture for the stacked auto-encoder is given by running the encoding advance of each layer in forward request:

$$a^{(l)} = f(z^{(l)})$$
$$z^{(l+1)} = W^{(l,1)}a^{(l)} + b^{(l,1)}$$

The decoding step is given by running the decoding stack of each auto-encoder in reverse order:

$$a^{(n+l)} = f(z^{(n+l)})$$
$$z^{(n+l+1)} = W^{(n-l,2)}a^{(n+l)} + b^{(n-l,2)}$$

The data of intrigue is contained inside a(n), which is the actuation of the deepest layer of hidden units. This vector gives us a representation of the input as far as higher-arrange highlights.

The highlights from the stacked auto-encoder can be utilized for classification issues by nourishing a(n) to a Softmax classifier.

To give a solid illustration, assume you wished to prepare a stacked auto-encoder with 2 hidden layers for classification of MNIST digits, First, you would prepare a scanty auto-encoder on the crude inputs x(k) to learn essential highlights h(1)(k) on the crude input.

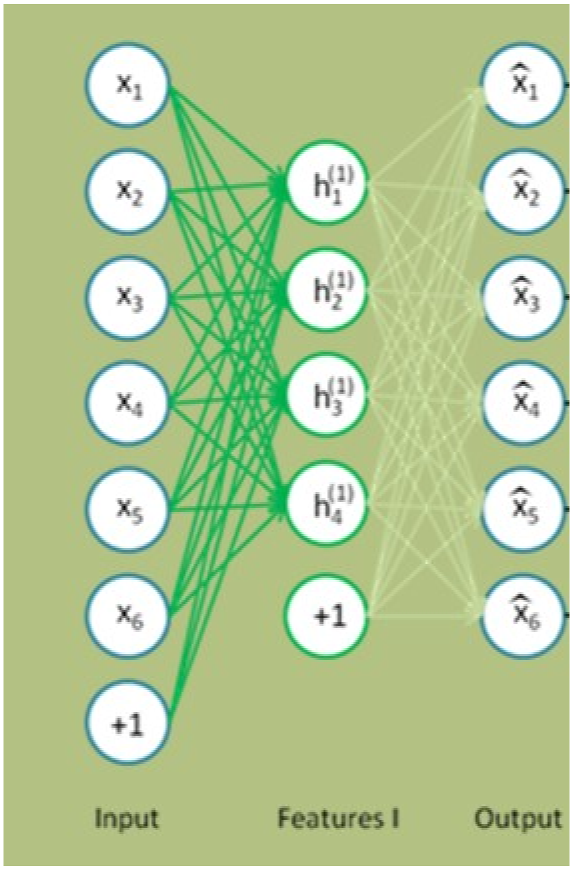


Figure 9: Auto-encoder step 1

Next, you would encourage the crude input into this trained auto-encoder, getting the essential element initiations h(1)(k) for every one of the inputs x(k). You would then utilize these essential features as the "crude input" to another meager auto-encoder to learn optional features h(2)(k) on these essential features.

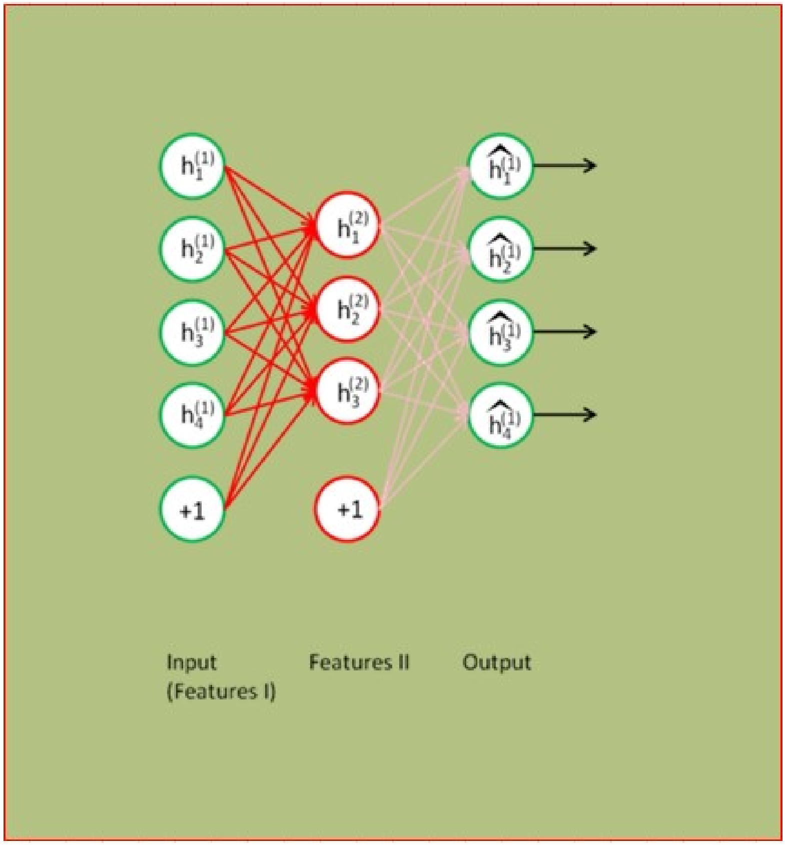


Figure 10: Auto-encoder step 2 (Glavan, Holban, 2012)

Following this, you would feed the primary features into the second sparse auto-encoder to obtain the secondary feature activations $h^{(2)(k)}$ for each of the primary features $h^{(1)(k)}$(which correspond to the primary features of the corresponding inputs $x^{(k)}$). You would then treat these secondary features as "raw input" to a Softmax classifier, training it to map secondary features to digit labels.

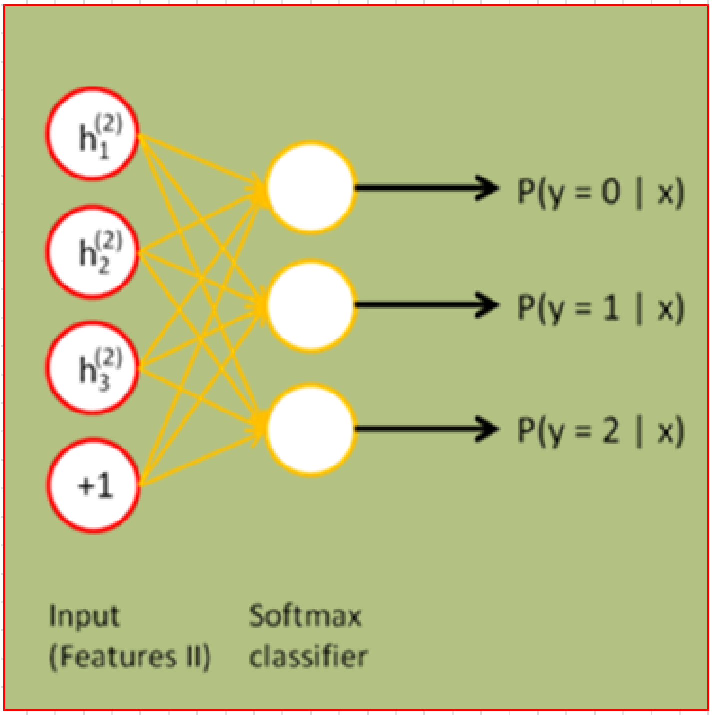


Figure 11: Softmax classifier (Glavan, Holban, 2012)

At long last, you would join every one of the three layers together to frame a stacked auto-encoder with 2 hidden layers and a last Softmax classifier layer equipped for ordering the MNIST digits as wanted.

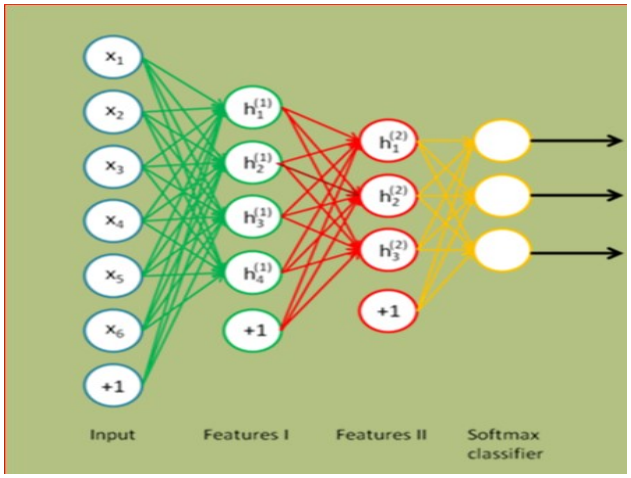


Figure 12: Stacked Auto-encoders (Glavan, Holban, 2012)

3.9.2 Training

A decent method to acquire great parameters for a stacked auto-encoder is to utilize ravenous layer-wise training. To do this, first prepare the primary layer on crude input to get parameters W(1,1),W(1,2),b(1,1),b(1,2). Utilize the primary layer to change the crude input into a vector comprising of enactment of the hidden units, A. Prepare the second layer on this vector to acquire parameters W(2,1),W(2,2),b(2,1),b(2,2). Rehash for ensuing layers, utilizing the output of each layer as input for the resulting layer (Albarqoun et al., 2016).

This strategy prepares the parameters of each layer exclusively while solidifying parameters for the rest of the model. To create better outcomes, after this period of training is finished, fine-tuning utilizing backpropagation can be utilized to enhance the outcomes by tuning the parameters of all layers are changed in the meantime (Bengio et al., 2007).

On the off chance that one is just intrigued by fine-tuning for the reasons for classification, the normal practice is to then dispose of the "disentangling" layers of the stacked auto-encoder and connection the last hidden layer a(n) to the Softmax classifier. The angles from the (Softmax) classification error will then be backpropagated into the encoding layers.

CHAPTER 4
NETWORKS TRAINING AND PERFORMANCE EVALUATION

4.1 Methodology

This study presents an original research for the diagnosis of chest X-rays using deep learning. A deep network named as stacked auto-encoder (SAE) is selected to be used as the brain this work. This selection came from the few researches that were conducted for the chest X-rays classification using this kind of networks. Thus, there is a need to investigate the effectiveness and performance of stacked auto-encoder in classifying the chest X-rays and detecting whether a radiograph has a disease or it is normal (healthy).

Two auto-encoder networks were used to build the proposed stacked auto-encoder that is then used to be as the intelligent classifier of the chest X-ray images. The auto-encoder was first trained layer by layer using greedy layer wise training until a network of two hidden layer, one input, and one output network is formed. Therefore, these trained auto-encoders were all stacked together and the proposed stacked auto-encoder is formed.

The proposed network is trained to classify chest images into normal which have no abnormalities or diseased images regardless of the type of the disease. A sample of the database normal and abnormal chest X-rays is shown in Figure 14.

Note that in this work, two deep models are employed. Both models are stacked auto-encoders with the same learning parameters, however, for the first model, which we call SAE1, the chest X-rays are fed directly into network, without processing and enhancement. The second network model, which is called SAE2, was trained on images that are processed and enhanced before being fed into network. The aim of the use of two models is to investigate the effects of processing and image enhancement on the auto-encoder training and testing performance.

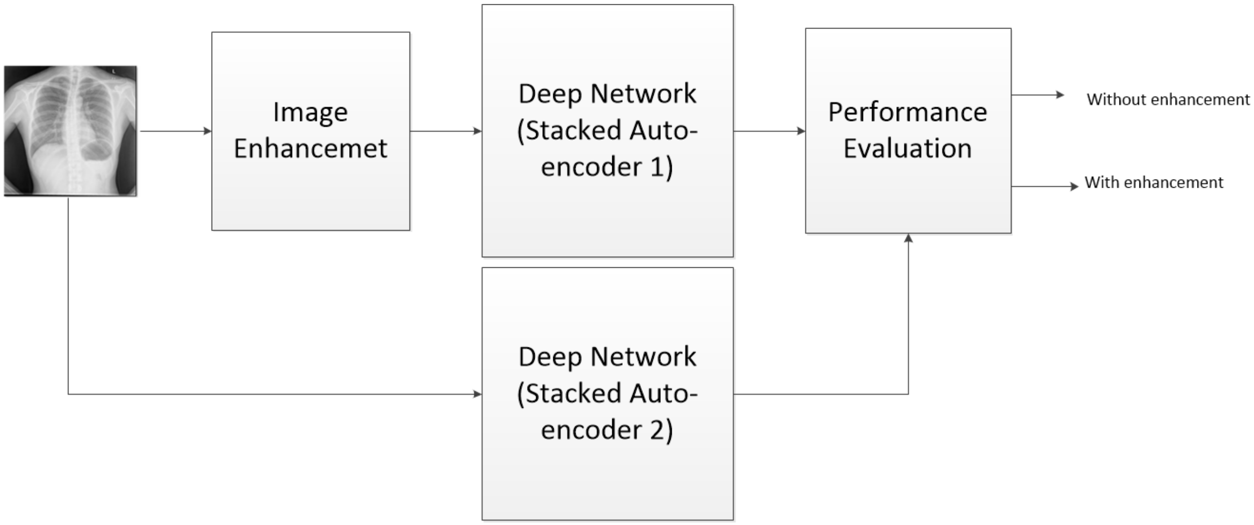


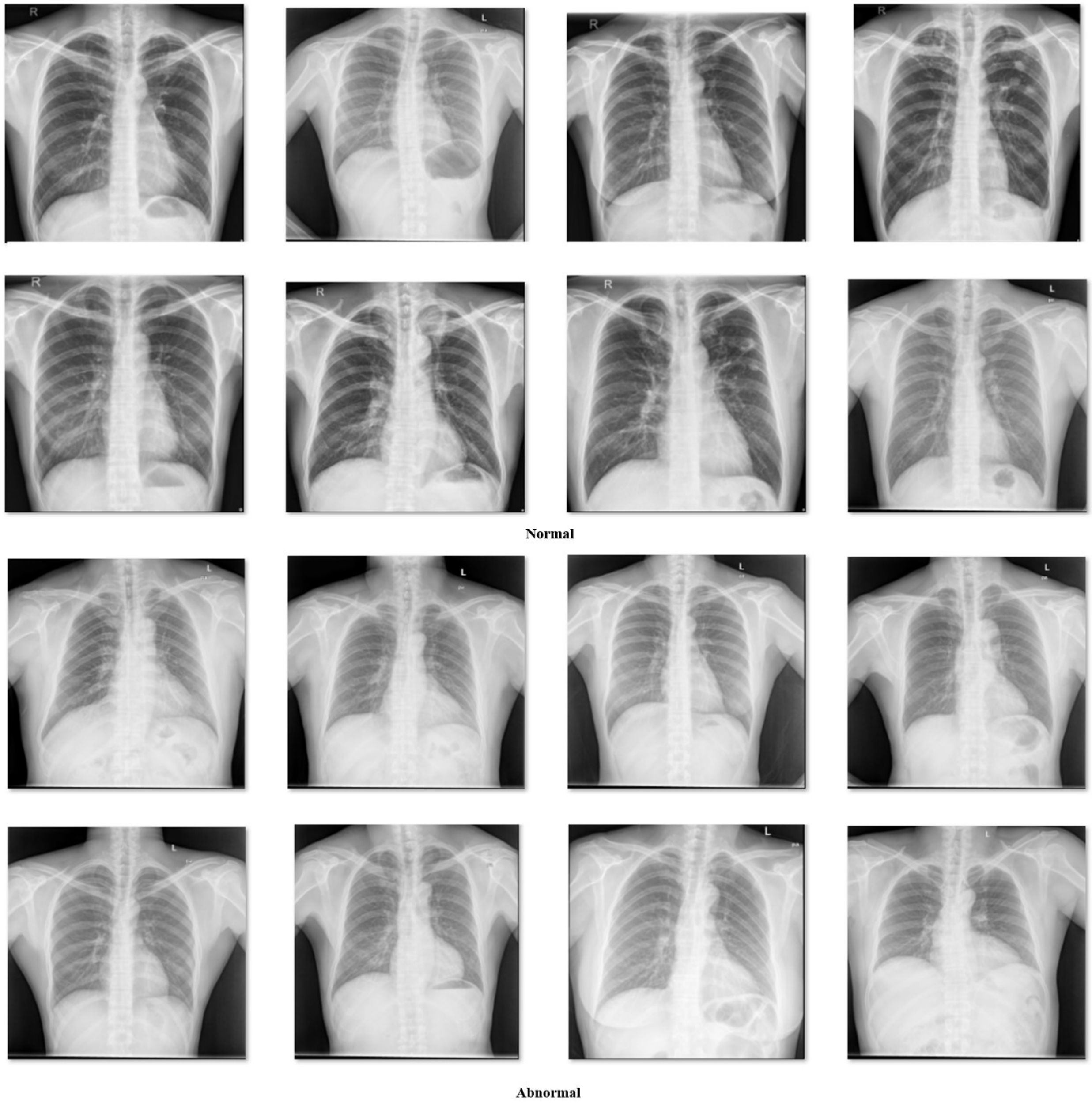
Image Enhancemet
Deep Network (Stacked Auto-encoder 1)
Performance Evaluation
Without enhancement
With enhancement
Deep Network (Stacked Auto-encoder 2)

Figure 13: Flowchart of the proposed methodology

Figure 13 shows the workflow of the proposed methodology. As seen, the network model is trained first on the chest images without enhancement and the network is then tested and the performance is evaluated. Same network is then trained and tested on same images but here they are enhanced using image Histogram equalization and similarly, the network is also evaluated and tested in order to investigate the one that outperforms in terms of accuracy and less error achieved.

4.2 Database

A deep network an intelligent classifier that is hungry for data. The more data it is trained on the more intelligent it will be. Therefore, there is need for a good database that has good number of normal and abnormal images to train and test the developed network. Therefore, the images in this work are all obtained from two public and well-known databases. The first one is Shenzhen Hospital X-ray Set / China data set: X-ray images in this data set (Fushman et al., 2005), while the other database is the Montgomery County X-ray Set. The first database contains chest X-rays of both normal and abnormal cases and they were acquired as part of the routine care at Shenzhen Hospital. The images are of JPEG format and there 340 normal x-rays and 275 abnormal x-rays showing various aspects of tuberculosis. The second database contains 58 abnormal x-rays and 80 normal images.



Normal

Abnormal

Figure 14: A sample of the database images

Figure 14 shows a sample of the chest X-rays found in the database used for training and testing the employed models. Note that the figure shows the two classes of the database images: the normal and the abnormal.

Table 1 shows the number of images found in the database in addition to the training and testing ratios used.

Table 1: Dataset 1 and data division

Image sets	Number of images	Normal images	Abnormal Images
Training set	400	200	200
Testing set	215	140	75
Both sets	615	340	275

Table 2 shows the number of X-rays in the dataset 2 and its data division.

Table 2: Dataset 2 and data division

Image sets	Number of images	Normal images	Abnormal Images
Training set	70	40	30
Testing set	68	40	28
Both sets	138	80	58

4.3 Training the Deep Models

In this section the training of the two deep models which are SAE1 and SAE 2 is discussed. Note that the SAE1 is the stacked auto-encoder network that uses X-ray images without enhancement while SAE2 is the same network but with enhanced images as inputs. It is important to mention that SAE1 and SAE2 are both trained on the same number of images which is 470 images; among them 240 are normal and 230 are abnormal.

For output classes coding it was considered as the following:

- Abnormal output class →[1 0],
- Normal output class →[0 1]

Note that the networks are first pre-trained as they are deep networks. Pre-training means that the networks are first trained layer by layer using Greedy-layer wise training (Hinton, 2006). In this phase there is no output labeling because the network is trained here to reconstruct its input from the extracted features in the hidden layer, which is why the number of output neurons is equal to the number of input neurons which is 4096.

Once the networks finish pre-training, it is then fine-tuned using the conventional backpropagation algorithm. Here, the input images are labeled therefore, output neurons are two which means that network is being trained to classify the images into two classes: normal and abnormal.

Table 3 shows the training and testing data from the two datasets. It is seen that the network is trained on 470 images and tested on 283 chest X-rays.

Table 3: Training and testing data from the two databases

Image sets	Number of images	Training	Testing
Dataset 1	615	400	215
Dataset 2	138	70	68
Both sets	753	470	283

4.3.1 SAE1 Training

SAE1 is a stacked auto-encoder that is trained on 470 images that are fed into it without any processing or enhancement technique. This deep model is composed of one input layer of 4096 neurons since the input images size is 64*64 pixels; two hidden layers of 100 and 65 neurons, respectively. Also, it has an output layer of two neurons as the output classes are only two.

Figure 15 shows the architecture of the SAE1. Table 4 shows the values of the learning parameters of the SAE1 when it is trained on 200 images.

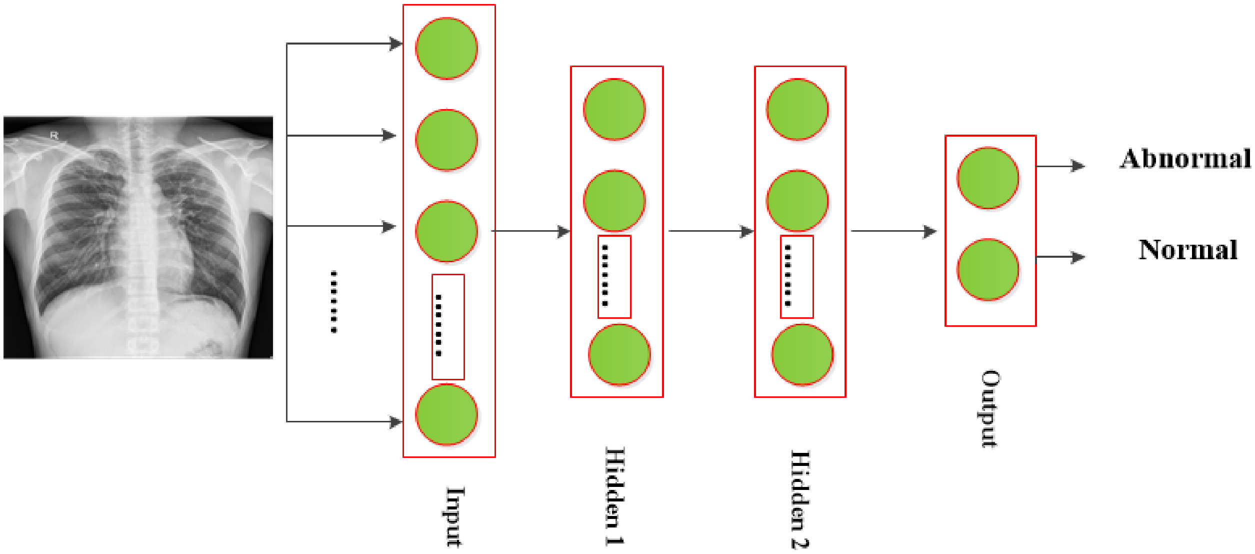


Figure 15: The deep network model 1 (SAE1) structure

Table 4: Learning parameters of networks during pre-training and fine-tuning

Learning parameters	Values (Pre-training)	Values (Fine-training)
Number of training images	470	470
Number of layers of the network	4	4
Number of hidden layers	2	2

Learning rate	0.27	0.15
Maximum number of iterations	500\500	323\400
Transfer function	Sigmoid	Sigmoid

Figure 16 shows the learning curve of the network during pre-training. It is seen that the network error was decreasing with the number of iterations, but it couldn’t reach a very small error (0.04).

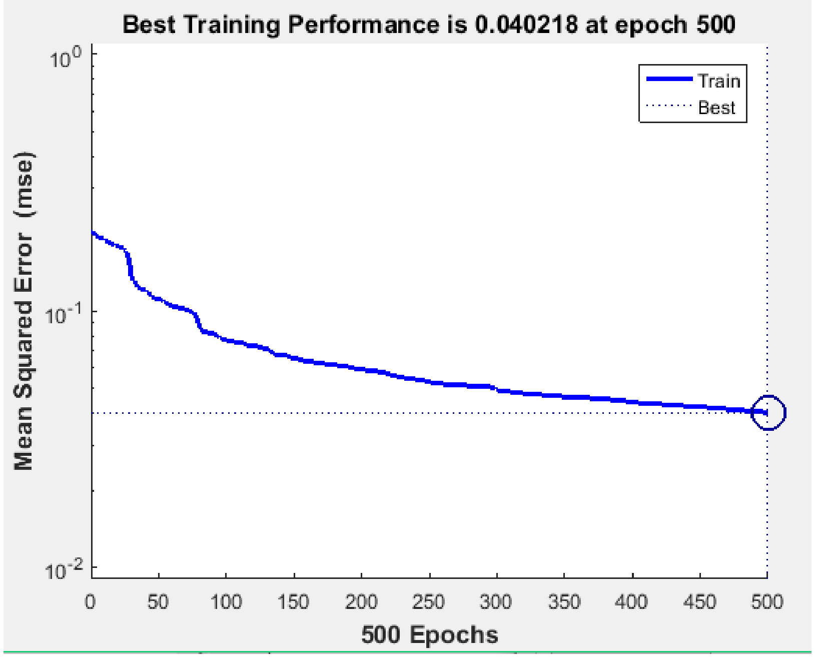


Figure 16: Learning curve of SAE1 during pre-training

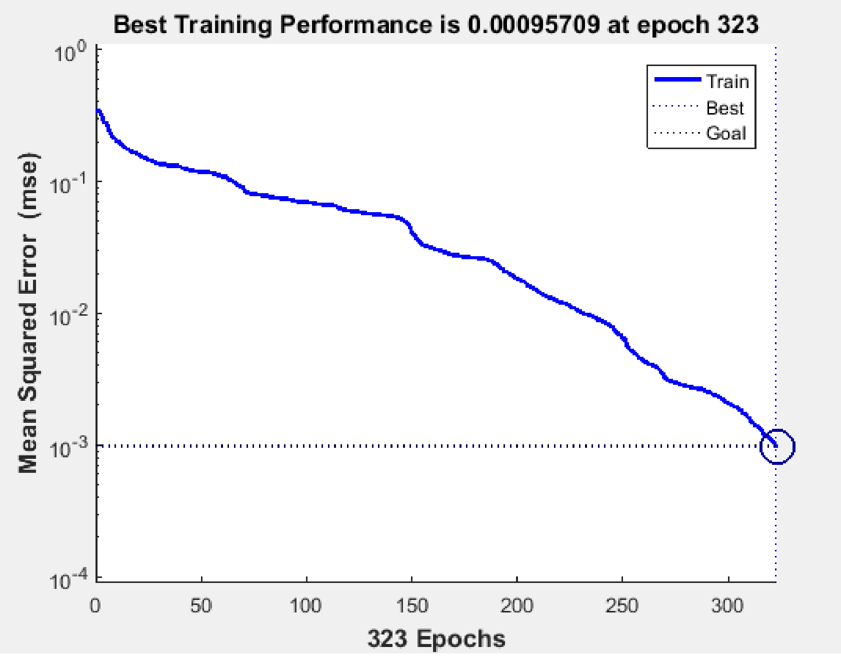


Figure 17: Learning curve of SAE1 during fine-tuning

The Figure 17 depicts the learning of the network SAE1 during fine-tuning. It is also seen that the network error is diminishing sharply and it reaches a very small error of 0.0009 at iteration 323 which indicates a good learning results of the network during this stage.

Table 5: Training network performance

Learning results	Pre-training	Fine-tuning
Number of training images	470	470
Training recognition rate	87%	100%
Minimum square error achieved (MSE)	0.0402	0.0009
Iterations required	500	323
Training time	250 secs	38 secs

Table 5 summarizes the performance of the SAE1 during pre-training and fine-tuning. It is seen that the network SAE1 performed very good during fine-tuning where it achieved a high recognition rate of 100% with a very short time (38 seconds) and with small number of iterations (323). Moreover, it is seen that the network reached a very small error during fine-tuning (0.0009). However, the network couldn't achieve similarly in the pre-training since it achieved a low recognition rate with long time and high error margins. This is not an issue because the network performance is evaluated in the fine-tuning stage where it is trained to classify however, in the pre-training the network is just trying to get the good and right weights that can be used on the fine-tuning.

4.3.1 SAE2 Training

SAE2 is the stacked auto-encoder network that is trained on the same chest X-rays, however, the images are processed and enhanced using histogram equalization. This image enhancement technique is an image processing tool that is used to enhance the contrast of the image by mapping or transforming the values of the intensities of pixels in image into different ranges so that the histogram becomes flat. This results in better quality images where the pixels are sharper and brighter. The result of histogram equalization is shown in Figure 18.

Table 6 shows the parameters setting of the SAE2 network when pre-training and fine-tuning.

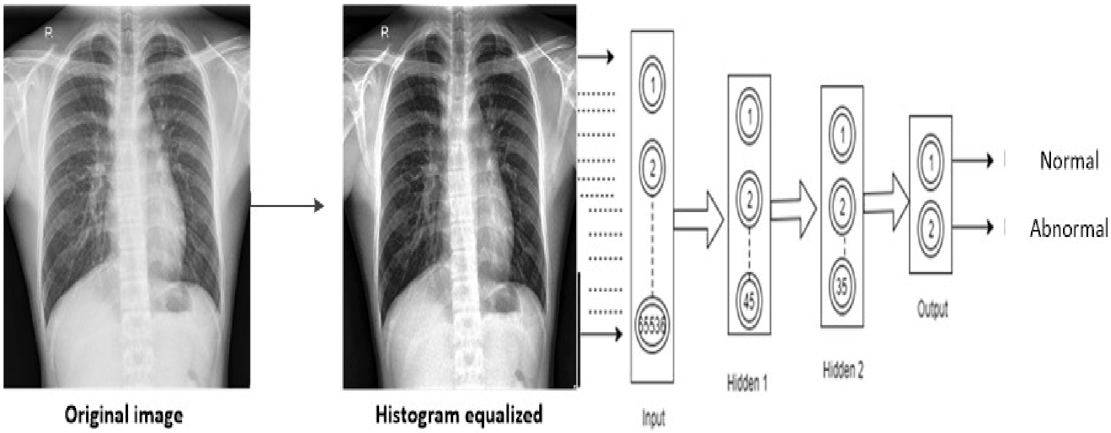


Figure 18: SAE2 network architecture

For the Figure 18 it is seen that the histogram helped in brightening the image which may be good in highlighting the important features that distinguish the abnormality of the chest X-rays

Table 6: Learning parameters during pre-training and fine-tuning of SAE2

Learning parameters	**Values (Pre-training)**	**Values (Fine-training)**
Number of training images	470	470
Number of layers of the network	4	4
Number of hidden layers	2	2
Learning rate	0.27	0.15
Maximum number of iterations	491\500	195\400
Transfer function	Sigmoid	Sigmoid

Figure 19 shows the learning curve of the network during pre-training. It is seen that the SAE2 network error was decreasing with the number of iterations, but it couldn't reach a very small error (0.037).

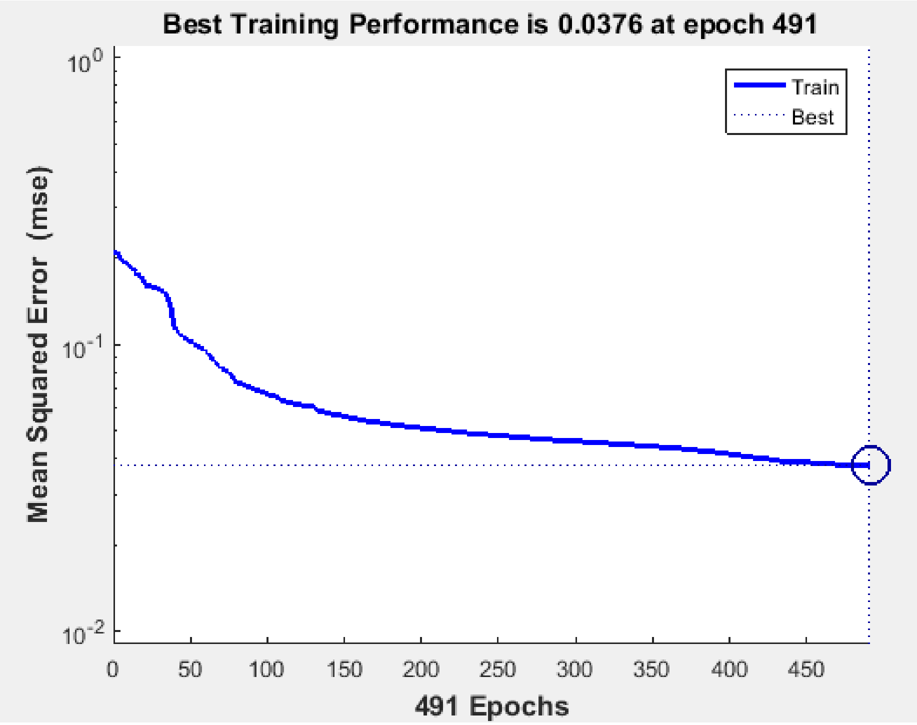


Figure 19: Learning curve of SAE2 during pre-training

Figure 20 shows the learning curve of the SAE2 network during fine-tuning. It is seen that the network achieved a very small error of 0.0008 within a short time (139 secs) and small number of iterations (195).

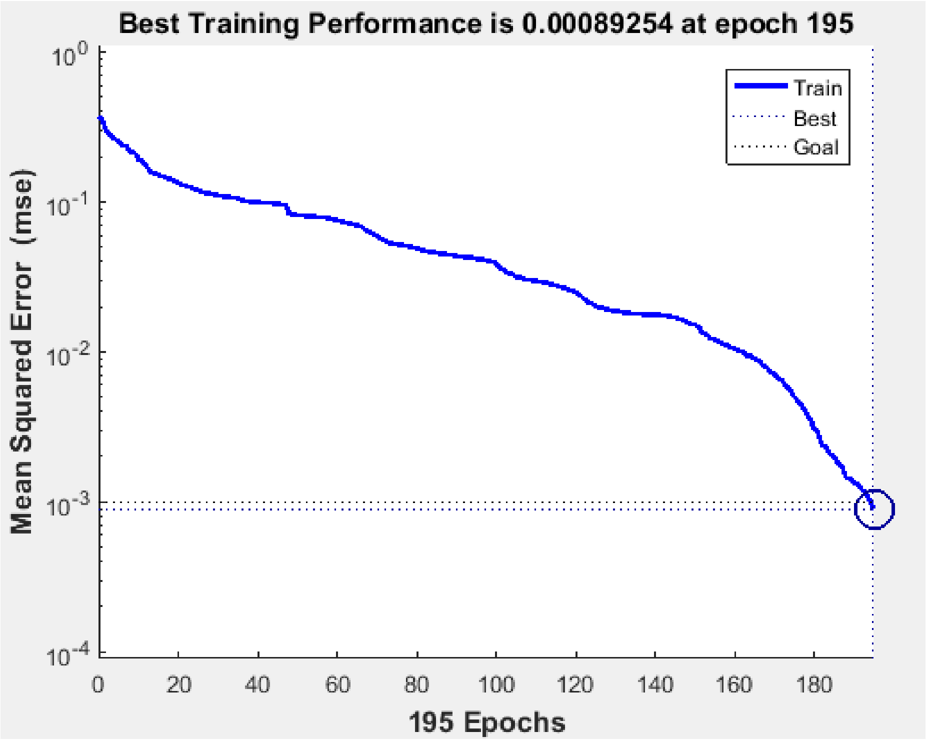


Figure 20: Learning curve of SAE2 during fine-tuning

The Table 7 shows the comparison of the network performance of SAE2 which uses image that are enhanced using histogram equalization. It is noticeable that this image processing technique has contributed to enhancing the network performance in both pre-training and fine-tuning. This is seen in the recognition ration during pre-training which is 90%, greater that of SAE1. Moreover, this network has achieved a smaller error rates (0.0376 and 0.008) than that obtained when histogram equalization is not used. Nevertheless, the use of histogram equalization has resulted in longer training time of the network (524 and 139 secs).

Table 7: Learning results of SAE2

Learning results	**Pre-training**	**Fine-tuning**
Number of training images	470	470
Training recognition rate	90%	100%
Minimum square error achieved (MSE)	0.0376	0.0008
Iterations required	491	195
Training time	524 secs	139 secs

4.4 Deep Models Testing

Once trained, both networks (SAE1 and SAE2) are tested using 283 chest X-rays. Among them, 180 are normal and the rest are abnormal. Figure 20 shows a sample of the chest images used for testing the networks. Table 8 shows the classification rates of networks during testing.

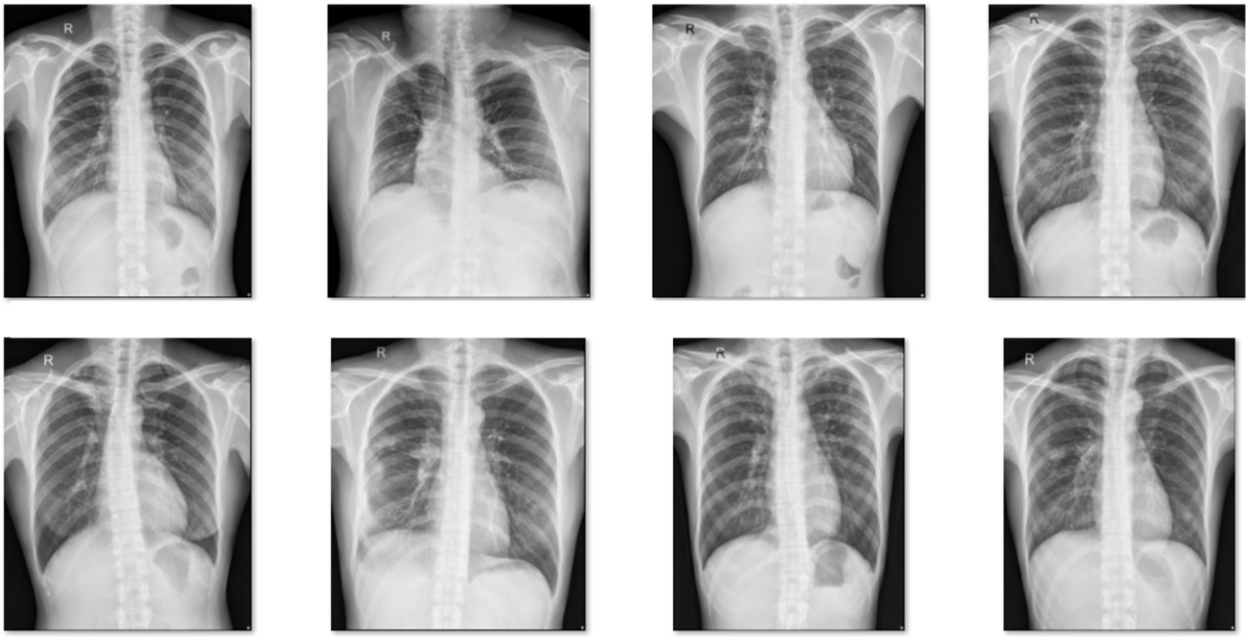


Figure 21: A sample of testing chest X-rays

From Table 8, it can be seen that the stacked auto-encoder which was trained on enhanced images has outperformed the one that used unenhanced images (SAE1) in terms of classification rates.

Table 8: Classification rates of both networks during testing

Deep Networks	Number of testing images	Classification rates
SAE1	283	89.5%
SAE2	283	93%

The Figure 22 shows some misclassified images when both models are tested. Note that these images were classified as abnormal while they are normal

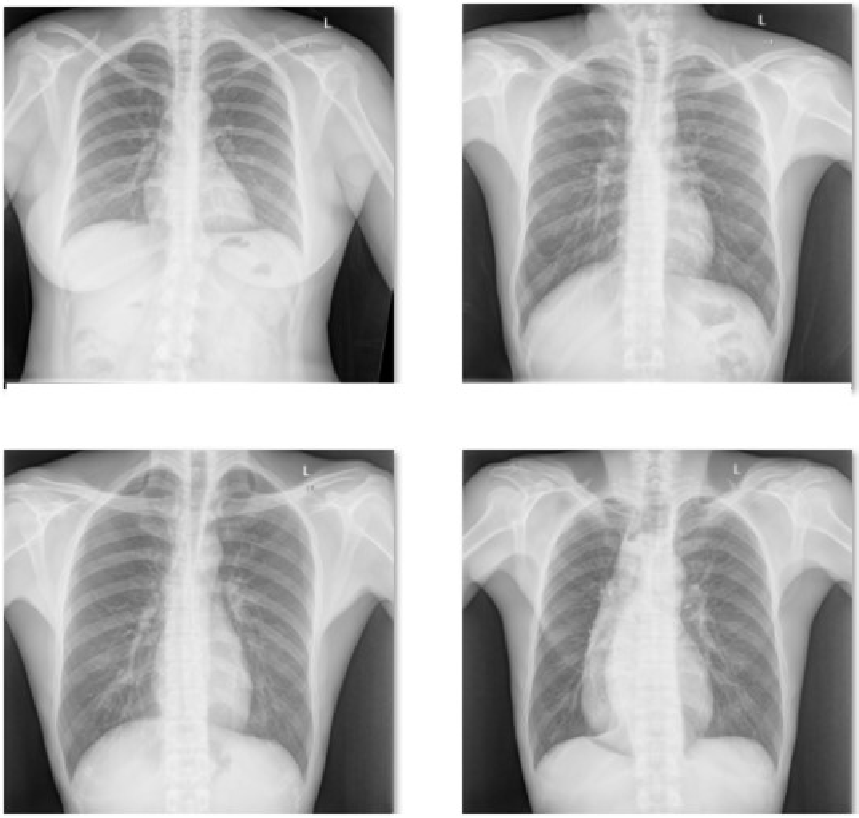


Figure 22: A sample of some misclassified images

4.5 Results Discussion

This thesis presents a deep learning approach for the classification of chest X-rays into normal or abnormal images. This work is based on a stacked auto-encoder that is trained to classify the chest X-rays images to learn the useful features that can distinguish both classes of images. This type of networks is trained using an algorithm called Greedy-layer wise training which is meant to train the network layer by layer in an unsupervised manner. This allows the network to gain the power of extracting of important features that will be used in the next training phase which is called Fine-tuning. Fine-tuning is to train pre-trained network to classify X-rays using the conventional backpropagation learning algorithm by using the weights obtained for the pre-training phase.

In this work, images are first fed into network without enhancement and the network was trained and tested. Then, same images are enhanced using histogram equalization and they are fed into network which is also trained and tested on these processed images. The aim of this is to study if the use of image enhancement technique in particular, histogram equalization affects the learning and performance of the network. Table 9 shows a comparison of the deep networks performances with and without enhancement. Moreover, this table shows the results of the backpropagation neural network (BPNN) when it is trained and tested on images with enhancement.

Table 9: Performance comparison with and without enhancement

Performance parameters	**SAE1**	**SAE2**	**BPNN**
Number of training images	470	470	470
Number of testing images	283	283	283
Training recognition rates	100%	100%	99%
Testing recognition rates	89%	93%	84.2%
Minimum square error achieved (MSE)	0.0009	0.0008	0.052
Iterations required	323	195	1230
Training time	38 secs	139 secs	123 secs

From Table 9, it can be seen that the network that was trained on images which were enhanced has achieved a 93% recognition rate which is higher than that obtained by the network that uses images without enhancement (89%). Moreover, the SAE2 has achieved a smaller error of 0.0008 smaller than that achieved by SAE1 (0.0009). It is also important to mention that the SAE has achieved such high recognition rate with smaller number of iterations however, with a longer training time. Furthermore, it is noted that the BPNN achieved lower accuracies and higher error rates compared to the other employed deep networks.

Overall, it can be concluded that the enhancement of images using image histogram equalization helps in achieving a higher recognition rate and error; however it requires longer training time which can be due to the process which needs more time.

4.6 Results Comparison

Chest classification using neural network have been conducted extensively. Different types of networks and approaches have been used as the classifiers that are trained to classify the chest x-rays. Table 10 shows a comparison of our network performance with other researches, in terms of accuracy. It is seen that our proposed work has achieved a greater accuracy (93%) than other researches.

Table 10: Results comparison

Paper titles	Accuracy
Learning to Read Chest X-Ray Images from 16000+ Examples Using CNN (Dong et al., 2017)	91.67%
Deep Learning for abnormality detection in Chest X-Ray images	80%
Stacked Auto-encoder	93%

CHAPTER 5
CONCLUSION

5.1 Conclusion

This thesis presents the employment of deep network in particular stacked auto-encoder in a medical field challenging task which is the classification of chest X-rays into normal and abnormal images. Such a classification medical system is needed as it makes the radiologist's job faster and easier.

A stacked auto-encoder is employed in this work and it was trained and tested on 470 and 283 images, respectively. The network is first trained on images taken directly from the database, without processing and enhancing. Then same network was tested and performance was evaluated in terms of training time, error, and accuracy. The same network was trained again on the same images but here the images were enhanced using histogram equalization. Also, this network was tested and an evaluation of its performance was carried out.

The performance of both networks was discussed and a comparison of the two network performance was shown, in terms of accuracy, error reached, training time, and number of iterations needed. After this comparison it was seen that the network that uses enhanced images outperformed the one that used unprocessed images as it achieved a higher recognition rate during testing.

In conclusion, the testing of stacked auto-encoder showed that it gained a good capability of diagnosing the new unseen chest X-rays and correctly classifying them into normal or abnormal images. Thus, it can be stated that the SAE can be a good classifier for the chest X-rays classification with a small margin of errors. Moreover, it is seen that the enhancement of chest X-rays using histogram equalization has a good role in improving the learning of the stacked auto-encoder, which results in a better accuracy during the testing of the network.

REFERENCES

Demner-Fushman, D., Kohli, M. D., Rosenman, M. B., Shooshan, S. E., Rodriguez, L., Antani, S., & McDonald, C. J. (2015). Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, *23*(2), 304-310.

Demner-Fushman, D., Antani, S., Simpson, M., & Thoma, G. R. (2012). Design and development of a multimodal biomedical information retrieval system. *Journal of Computing Science and Engineering*, *6*(2), 168-177.

Jaeger, S., Karargyris, A., Candemir, S., Folio, L., Siegelman, J., Callaghan, F., & Thoma, G. (2014). Automatic tuberculosis screening using chest radiographs. *IEEE transactions on medical imaging*, *33*(2), 233-245.

Xue, Z., You, D., Candemir, S., Jaeger, S., Antani, S., Long, L. R., & Thoma, G. R. (2015). Chest x-ray image view classification. *In Proceedings of the 28th International Symposium on Computer-Based Medical Systems (CBMS)* (pp. 66-71). Brazil: Ribeião Preto.

Candemir, S., Jaeger, S., Palaniappan, K., Musco, J. P., Singh, R. K., Xue, Z., & McDonald, C. J. (2014). Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. *IEEE transactions on medical imaging*, *33*(2), 577-590.

Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, T., Kobayashi, T., Komatsu, K. I., & Doi, K. (2000). Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. *American Journal of Roentgenology*, *174*(1), 71-74.

Dong, Y., Pan, Y., Zhang, J., & Xu, W. (2017). Learning to Read Chest X-Ray Images from 16000+ Examples Using CNN. *In the Proceedings of the IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies* (pp. 51-57). USA: Philadelphia.

Cernazanu-Glavan, C., & Holban, S. (2013). Segmentation of bone structure in X-ray images using convolutional neural network. *Advanced Electrical and Computer Engineering*, *13*(1), 87-94.

Patil, S. A., & Kuchanur, M. B. (2012). Lung cancer classification using image processing. *International Journal of Engineering and Innovative Technology*, *2*(3).

Er, O., Yumusak, N., & Temurtas, F. (2010). Chest diseases diagnosis using artificial neural networks. *Expert Systems with Applications*, *37*(12), 7648-7655.

El-Solh, A. A., Hsiao, C. B., Goodnough, S., Serghani, J., & Grant, B. J. (1999). Predicting active pulmonary tuberculosis using an artificial neural network. *Chest*, *116*(4), 968-973.

Ashizawa, K., Ishida, T., MacMahon, H., Vyborny, C. J., Katsuragawa, S., & Doi, K. (1999). Artificial neural networks in chest radiography: application to the differential diagnosis of interstitial lung disease. *Academic radiology*, *6*(1), 2-9.

Santos A.M., Pereira B.B., Seixas J.M., Mello F.C.Q., Kritski A.L. (2007) Neural Networks: An Application for Predicting Smear Negative Pulmonary Tuberculosis. In JL. Auget, N. Balakrishnan, M. Mesbah, G. Molenberghs (Eds), *Advances in Statistical Methods for the Health Sciences* (pp. 131-139). Boston: Birkhäuser.

Albarqouni, S., Baur, C., Achilles, F., Belagiannis, V., Demirci, S., & Navab, N. (2016). Aggnet: deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE transactions on medical imaging*, *35*(5), 1313-1321.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation, 18*(7), 1527-1554.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *In the Proceedings of the Twenty-First Annual Conference on Advances in neural information processing systems* (pp. 153-160). Canada: Cancouver.

Demner-Fushman, D., Kohli, M. D., Rosenman, M. B., Shooshan, S. E., Rodriguez, L., Antani, S., & McDonald, C. J. (2015). Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, *23*(2), 304-310.

Er, O., Yumusak, N., & Temurtas, F. (2010). Chest diseases diagnosis using artificial neural networks. *Expert Systems with Applications*, *37*(12), 7648-7655.

El-Solh, A. A., Hsiao, C. B., Goodnough, S., Serghani, J., & Grant, B. J. (1999). Predicting active pulmonary tuberculosis using an artificial neural network. *Chest*, *116*(4), 968-973.

Ashizawa, K., Ishida, T., MacMahon, H., Vyborny, C. J., Katsuragawa, S., & Doi, K. (1999). Artificial neural networks in chest radiography: application to the differential diagnosis of interstitial lung disease. *Academic radiology*, *6*(1), 2-9.

APPENDICES

APPENDIX 1
IMAGE PROCESSING CODE

```
clear

clc

close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Abnormal X-rays

PATTERNS = [];

for k = 1: 336

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Makki\ChinaSet_AllFiles\ChinaSet_AllFiles\Abnormal1 X-ray';

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName);

  fprintf(1, 'Now reading %s\n', fullFileName);

  img = im2double(imread(fullFileName));

 im1=imresize(img, [64 64]);

 v=im1(:);

PATTERNS =[PATTERNS v];

end

save('AbNormalDATA.mat','PATTERNS');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%Normal X-rays

PATTERNS2 = [];

for k = 1: 326

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Makki\ChinaSet_AllFiles\ChinaSet_AllFiles\Normal X-ray';

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName);

  fprintf(1, 'Now reading %s\n', fullFileName);

  img = im2double(imread(fullFileName));

 im1=imresize(img, [64 64]);

 v=im1(:);

PATTERNS2 =[PATTERNS2 v];

end

save('NormalDATA.mat','PATTERNS2');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%

TrainImages1=PATTERNS(:,1:200);

TrainImages2=PATTERNS2(:,1:200);
```

```
TrainImages=[TrainImages1, TrainImages2];

TestImages1= PATTERNS(:,201:336);

TestImages2= PATTERNS2(:,201:326);

TestImages=[TestImages1, TestImages2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Targets

a=ones(2,662);

a(2,:)=0;

t1=a(:,1:336);

b=a(:,337:end);

b(1,:)=0;

b(2,:)=1;

t2=b;

Targets=[t1,t2];

  %the desired output data


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%Targets

t2=Targets(:,1:200);

t3=Targets(:,337:536);

TrainTargets=[t2,t3];

t5=Targets(:,201:336);

t6=Targets(:,537:662);
```

```
TestTargets=[t5,t6];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%

%Autoencoder training usin Greedy wise layer training:

net = newff(minmax(TrainImages),[100 4096],{'logsig','logsig'},'trainscg');

% TRAINING THE NETWORK

net.trainParam.lr = 0.27;  % Learning Rate.

net.trainParam.show = 300; % Frequency of progress displays (in epochs).

net.trainParam.epochs =500;% Maximum number of epochs to train.

net.trainParam.mc = 0.7 % Momentum Factor.

TARGETS=TrainImages;                         % OUTPUTS = INPUTS

[net,tr] = train(net,TrainImages, TARGETS);

 train1=sim(net, TrainImages);

 Hidden_weights = net.LW{2,1};

 inp_weights=net.IW{1,1};                        %%%%

%------------------------------

save('AE','net');                        %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

% Fine tuning of the Autoencoder

net = newff(minmax(TrainImages),[65 2],{'logsig','logsig'},'trainscg');

%  net.IW{1,1}=inp_weights;                              %%%%%%%
%  net.LW{2,1}=0.01*rand(60,4096)                          %%%%%%

% TRAINING THE NETWORK

net.trainParam.lr = 0.15;  % Learning Rate.

net.trainParam.show = 300; % Frequency of progress displays (in epochs).

net.trainParam.epochs =400;% Maximum number of epochs to train.

net.trainParam.mc = 0.5 % Momentum Factor.

net.trainParam.goal = 0.001

[net,tr] = train(net,TrainImages,TrainTargets);

train = sim(net,TrainImages);

%RECOGNITION RATE OF TRAIN DATA

%target max indices

[M,I_t]=max(TrainTargets);% row vector

%dimensions of target matrix

[u,v]=size(TrainTargets);

%actual output matrix sim_net

sim_net=sim(net,TrainImages);

[N,I_sim_net]=max(sim_net);% row vector

%comparison of target and actual outputs

result = I_t==I_sim_net;% row vector

%sum of all elements,1s, to know how many corrects

corrects=sum(result);
```

```
%recognition rate,

w=double(corrects*100/v); %let recognition rate be w

fprintf('train recognition rate is %d\n',w);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%T
esting

%target max indices

[M,I_t]=max(TestTargets);% row vector

%dimensions of target matrix

[u,v]=size(TestTargets);

Test=sim(net,TestImages);

%actual output matrix sim_net

[N,I_sim_net]=max(Test);% row vector

%comparison of target and actual outputs

result = I_t==I_sim_net;% row vector

%sum of all elements,1s, to know how many corrects

corrects=sum(result);

%recognition rate,

w=double(corrects*100/v); %let recognition rate be w

fprintf('Test Accuracy is %d\n',w);
```

APPENDIX 2
NEURAL NETWORKS CODE

```
clear

clc

close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Abnormal X-rays

PATTERNS = [];

for k = 1: 336

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Makki\ChinaSet_AllFiles\ChinaSet_AllFiles\Abnormal1 X-ray';

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName);

  fprintf(1, 'Now reading %s\n', fullFileName);

  img = im2double(imread(fullFileName));

  %% Enhance contrast using histogram equalization

im1 = histeq(img);

 im1=imresize(im1, [64 64]);

 v=im1(:);

PATTERNS =[PATTERNS v];

end
```

```
save('AbNormalDATA.mat','PATTERNS');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%Normal X-rays

PATTERNS2 = [];

for k = 1: 326

myFolder = 'C:\Users\TOSHIBA\imp. documents\students
projects\Makki\ChinaSet_AllFiles\ChinaSet_AllFiles\Normal X-ray';

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName);

  fprintf(1, 'Now reading %s\n', fullFileName);

  img = im2double(imread(fullFileName));

    %% Enhance contrast using histogram equalization

im1 = histeq(img);

 im1=imresize(im1, [64 64]);

 v=im1(:);

PATTERNS2 =[PATTERNS2 v];

end

save('NormalDATA.mat','PATTERNS2');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%

TrainImages1=PATTERNS(:,1:200);

TrainImages2=PATTERNS2(:,1:200);

TrainImages=[TrainImages1, TrainImages2];

TestImages1= PATTERNS(:,201:336);

TestImages2= PATTERNS2(:,201:326);

TestImages=[TestImages1, TestImages2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Targets

a=ones(2,662);

a(2,:)=0;

t1=a(:,1:336);

b=a(:,337:end);

b(1,:)=0;

b(2,:)=1;

t2=b;

Targets=[t1,t2];

 %the desired output data


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%Targets

t2=Targets(:,1:200);
```

```
t3=Targets(:,337:536);

TrainTargets=[t2,t3];

t5=Targets(:,201:336);

t6=Targets(:,537:662);

TestTargets=[t5,t6];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%

%Autoencoder training usin Greedy wise layer training:

net = newff(minmax(TrainImages),[100 4096],{'logsig','logsig'},'trainscg');

% TRAINING THE NETWORK

net.trainParam.lr = 0.27;  % Learning Rate.

net.trainParam.show = 300; % Frequency of progress displays (in epochs).

net.trainParam.epochs =500;% Maximum number of epochs to train.

net.trainParam.mc = 0.7 % Momentum Factor.

TARGETS=TrainImages;                        % OUTPUTS = INPUTS

[net,tr] = train(net,TrainImages, TARGETS);

 train1=sim(net, TrainImages);

 Hidden_weights = net.LW{2,1};

 inp_weights=net.IW{1,1};                   %%%%

%------------------------------
```

```
save('AE','net');                    %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

% Fine tuning of the Autoencoder

net = newff(minmax(TrainImages),[65 2],{'logsig','logsig'},'trainscg');

%  net.IW{1,1}=inp_weights;                        %%%%%%%

%  net.LW{2,1}=0.01*rand(60,4096)                      %%%%%%

% TRAINING THE NETWORK

net.trainParam.lr = 0.15;  % Learning Rate.

net.trainParam.show = 300; % Frequency of progress displays (in epochs).

net.trainParam.epochs =400;% Maximum number of epochs to train.

net.trainParam.mc = 0.5 % Momentum Factor.

net.trainParam.goal = 0.001

[net,tr] = train(net,TrainImages,TrainTargets);

train = sim(net,TrainImages);

%RECOGNITION RATE OF TRAIN DATA

%target max indices

[M,I_t]=max(TrainTargets);% row vector

%dimensions of target matrix

[u,v]=size(TrainTargets);
```

```
%actual output matrix sim_net

sim_net=sim(net,TrainImages);

[N,I_sim_net]=max(sim_net);% row vector

%comparison of target and actual outputs

result = I_t==I_sim_net;% row vector

%sum of all elements,1s, to know how many corrects

corrects=sum(result);

%recognition rate,

w=double(corrects*100/v); %let recognition rate be w

fprintf('train recognition rate is %d\n',w);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Testing

%target max indices

[M,I_t]=max(TestTargets);% row vector

%dimensions of target matrix

[u,v]=size(TestTargets);

Test=sim(net,TestImages);

%actual output matrix sim_net

[N,I_sim_net]=max(Test);% row vector

%comparison of target and actual outputs

result = I_t==I_sim_net;% row vector
```

```
%sum of all elements,1s, to know how many corrects

corrects=sum(result);

%recognition rate,

w=double(corrects*100/v); %let recognition rate be w

fprintf('Test Accuracy is %d\n',w);
```