A BACKPROPAGATION NEURAL NETWORK FOR THE LEFT VENTRICLE DETECTION

A THESIS SUBMITTED TO THE

GRADUATE SCHOOL OF APPLIED SCIENCES

OF

NEAR EAST UNIVERSITY

By
ANWAR A IBRA

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

Electrical and Electronics Engineering

NICOSIA, 2018

ANWAR A IBRA

A BACKPROPAGATION NEURAL NETWORK
FOR THE LEFT VENTRICLE DETECTION

NEU
2018

A BACKPROPAGATION NEURAL NETWORK FOR THE LEFT VENTRICLE DETECTION

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

By
ANWAR A IBRA

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, Asst.Prof. Dr. Elbrus Imanov 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

Machine learning has proved its effectiveness through its aplication in medicine. neural networks have been used for solving different dilemma in the medical field such as image analysis and diagnosis. Object detection is a very hot topic in computer vision due to the need of detecting some importnant objects in images or videos. In medicine, there is also a need for object detection wich is in this field, organ detection. Thus, different types of deep networks have been used in medicine for the detection of organs or parts in some organs such as the detection of ventricles in heart images. This thesis presents a neural network based system for the automated detection of left ventricles in MRI cardiac images. The work is based on a backpropagation neural network with a sliding window used to go through the images in order to find and detect the left ventricle. A network is first trained to classify left ventricles and non-left ventricles images using a backpropagation neural network. The trained network is then validated to gain a good generalization capability by testing it on unseen data. Upon training, the network was capable of achieving a classification rate of 100% and 88% on the training and test set, respectively. A sliding window of size 40×40 was used for the detection task, where the target images containing left ventricles are sampled using a sliding window of 1600 pixels. The trained network is then used to determine if a left ventricle is contained or not in the sampled region of the target image. Moreover, the developed system seems to perform effectively even when the target images are somewhat noisy as it is seen to be capable of detecting left ventricles in target images with up to 10% salt & pepper noise.

Keywords: Sliding windows; machine learning; Backpropagation; neural network; generalization; intelligent system; salt & pepper noise

ÖZET

Yapay zeka ile öğrenim etkisini tıptaki uygulamalarda göstermektedir. Sinir ağları geçmişte tıp alanındaki görüntü analizi ve teşhisi gibi farklı ikilemleri çözmekte kullanılırdı. Nesne saptama, görüntülerde veya videolarda yer alan bazı önemli nesneleri saptama yönünden bilgisayar görüntüleme konusu altındaki en gözde başlıklardan biridir. Tıp alanında nesne saptama özellikle organ saptaması yönünden de çok gereklidir. Böylece, farklı tarzda ağlar organ saptama veya kalp karıncıkları gibi organ bölümlerinin de saptanabilmesi için tıpta kullanılmaktadır. Bu tez, sol kalp kapakçıklarının MRI kalp görüntülerinde saptanmasında kullanılan otomatik sinir ağı sistemine dayanmaktadır. Bu çalışma sol kalp kapakçığının bulunması ve saptanması için slayt görüntülerinden yararlanarak geri yayılım sinir ağı görüntülerinin üzerinden gidilmesi şeklinde yapılmıştır. İlk olarak sinir ağı geri yayılımı kullanılarak sol ve sol olmayan kapakçıkların sınıflandırılması yapılmıştır. Daha sonra üzerinde çalışılmış bu ağın, iyi bir genelleme kapasitesi kazanması için görünmez veri üzerinde test edilmiştir. Bunun üzerine, bu ağ sınıflandırma açısından %100, çalışma ve test ortamı yönünden %88lik oranlara ulaştırılmıştır. 40x40 boyutlarında, üzerinde sol kapakçıkların sergilendiği hedef görüntüler olan 1600 piksellik bir slayt penceresi, saptama işlemi için kullanılmıştır. Bu çalışılmış ağ sonra da sol kapakçığın hedef görüntüden bir örnek içerip içermediğini saptamak üzere kullanılmıştır. Bunlar yanında geliştirilmiş bu sistemin hedef görüntülerin sol kapak saptamasında görüldüğü üzere %10'a kadar bir kalp atışı sesine benzer bir sebepten gürültülü olması halinde bile etkili olarak işlediği görülmektedir.

***Anahtar Kelimeler**:* Slayt pencereleri; yapay zekâ ile öğrenim; Geri yayılım; sinir ağı; genelleme; akıllı system; kalp atış sesi

TABLE OF CONTENT

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN:	Artificial Neural Network
BPNN:	Back Propagation neural network
MSE:	Mean Square Error
RGB:	Red, Grey, Blue
SEC:	Second
SVM:	Support Vector Machine
CNN:	Convolutional Neural Network

CHAPTER 1
INTRODUCTION

1.1 Introduction

Machine networks have been extensively used in medicine to solve different conditions in various areas: tumor diagnosis (Argyrou et al., 2012), image classification (Kumar et al., 2015), and image segmentation (Huang et al., 2009). Machine Learning is a new and advanced field of Machine Learning (Argyrou et al., 2012). It has been developed and improved in order to move the Machine Learning to be closer to its main and original goal: Artificial Intelligence.

Object identification is an errand that generally has a place with the class of PC vision issues. It is critical that while people are exceptionally compelling and proficient in distinguishing different complex objects regardless of scene limitations such fluctuating background, object scale, object positional interpretation, object introduction, object brightening, and so on.; machines endeavor to accomplish close human execution on object identification (Marak et al., 2009). Moreover, it is focused on that object location is very all the more trying for machines when contrasted with object acknowledgment. In object discovery, the object important to be distinguished can be situated in any district of a picture. While, in object acknowledgment, the objects important to be perceived is typically officially portioned; consequently, making the acknowledgment less difficult. So as to prevail in an errand, for example, object location, created vision models or frameworks ought to be equipped for adapting to the previously mentioned scene requirements.

One of the obvious choices is to consider an intelligent model such as artificial neural network (ANN). Artificial neural networks are popular for its capability to learn various tasks using collected examples or training data. More importantly, ANNs are capable of making intelligent decisions on tasks for which they are trained (Gouk and Blake, 2015). Some of the intelligent decision capabilities include tolerance to constraints such as object translation, object rotation, object scale, object illumination, and noise. Artificial neural networks have been successfully used in many important tasks such as face recognition, speaker identification, natural language processing, document segmentation, etc. In this project, a type of artificial neural network known as back propagation neural network (BPNN) is used to achieve the task of left ventricles detection in images obtained from Sunnybrook-cardiac database (Radau et al., 2009). The developed left ventricle detection system using back propagation neural network is found to be quite efficient and effective for the left ventricle detection task.

Many researches have been conducted on the segmentation and localization of ventricles. In (Tran., 2016) the authors have conducted a research on the use of Fully Convolutional Neural Network (FCN) for Cardiac Segmentation in Short-Axis MRI. This study has been focused on tackling the problem of automated left and right ventricle segmentation through the application of FCN, and were able to show that the FCN achieves state-of-the-art semantic segmentation in short-axis cardiac MRI acquired at multiple sites and from different scanners. The proposed FCN architecture used was efficiently trained end-to-end on a graphics processing unit (GPU) in a single learning stage to make inference at every pixel. This model was tested and it independently segments each image in milliseconds, therefore it could be employed in parallel on clusters of CPUs, GPUs, or both for scalable and accurate ventricle segmentation.

In another research (Zotti, 2017), authors trained a Convolutional Neural Network to perform semantic segmentation on images obtained from cardiac MRI scan in order to localize the left ventricle and influence the system to measure the capacity of the ventricle during the course of a heartbeat. Several pre- processing and data augmentation steps aimed at generalizing and preventing over fitting by use of image filters were done. Lastly, they were able to demonstrate the left ventricle as a single filled bordering region.

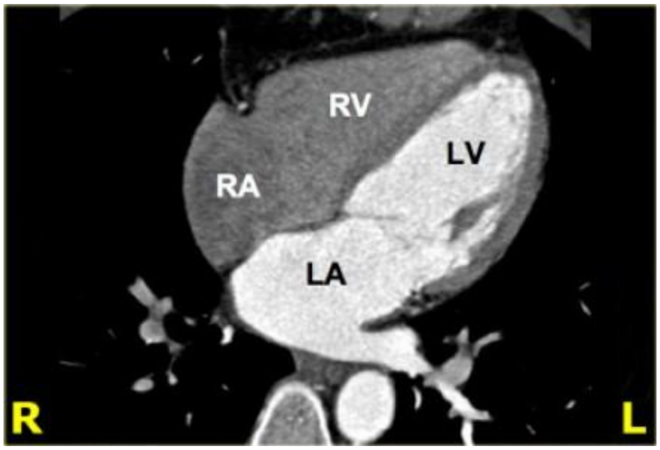

Figure 1.1: Heart Anatomy (Tran, 2016)

1.2 Significance of the Work

Medical object detection and localization is a must in the medicine field. Thus, there is a need for the use of machine learning techniques in detecting some organs and objects in medical images. Left ventricle is a critical object in the heart, which makes the researchers use different ways to localize it because its localization can yield many understanding and finding of diseases. The goal of this research work is to develop a vision system for the detection of left ventricles in images. In this work, we show a simple and new approach for detecting the left ventricle through a sliding window and a trained backpropagation neural network. It is important to note the scope and application of this project is more and broader than the detection of left ventricles only. The same idea and approach can be easily extended towards the detection of other medical objects in images.

1.3 Contributions of Research

1. Design an artificial vision system which can inspect presented images to determine if a left ventricle object is present in it.

2. The developed artificial vision system can perform competitively in the face of changes in left ventricle object characteristics such as scale, translation, rotation, illumination, noise, etc.

3. Tasked with the detection of left ventricle objects in highly complex and varying background, the developed artificial vision system can still perform with reasonable accuracy.

1.4 Aims of Thesis

The high efficiency obtained by using the machine networks in medicine motivated researchers to apply these machine models to localize medical objects in a similar and more accurate manner of the humans. Therefore, many studies have used machine networks to detect and localize different organs in medical images such as Left and right ventricles in the magnetic resonance images. In this work, we opt to design a machine learning approach for the localization of left ventricles in MR slices obtained from SunnyBrook database (Radau et al., 2009). The system is based on a machine learning network named backpropagation neural network (BPNN), which was trained to scan an MR slice to localize the left ventricle in magnetic resonance images.

1.5 Thesis Overview

The presented thesis is structured as follows:

Chapter one is an introduction of the work in addition to showing its significance and aims.

Chapter two is a literature review of the image processing tools used in detecting the left ventricles. Also, it shows a review of the works related to the left ventricle detection using neural networks.

Chapter three discusses the methodology of the presented research. Chapter three also presents the learning phase of the network that is trained on images to be capable to classify the left-ventricle and non-left ventricle images. In this chapter, the results of training are shown and discussed.

Chapter four presents the testing stage of the network where it is tested in images of heart in to detect the left ventricle in the images. In this chapter, many simulations are conducted in order to check the robustness of the system when images are noisy.

Finally, Chapter five shows the discussion and conclusion of the presented study.

CHAPTER 2
LITERATURE REVIEW

2.1 Overview

This chapter sufficiently gives insight into the different object detection approaches, ranging from traditional computer vision techniques to machine learning techniques using artificial neural network which can be seen as reformulations of the object detection tasks as learning problems. Also, an important aspect of artificial vision systems, image processing, are discussed.

2.2 Medical Object Detection

Technology is growing at an exponential rate, and so is the vast amount of data that need to be processed. However, for us to make efficient use of these large data, some important information about them is required e.g. control (Abiyev and Altunkaya, 2008; Abiyev, 2005). For example, in content-based media retrieval systems, it is the aim to extract some media information based on supplied specifications. What is interesting is that due to the vast amount of data that has to be filtered, manual processing or inspection is almost always infeasible. Hence, we rely on machines to efficiently perform this task. Some of these tedious and unconventional tasks that machines are tasked to perform include but not limited to face detection, face recognition, facial expression recognition, texture classification, etc. (Hsu et al., 2002; Zou and Yuan, 2012; Sandbach et al., 2012; Manthalkar et al., 2003). Many works have proposed machines or artificial vision systems to solving the aforementioned vision tasks (Sjrivastava and Tyagi, 2014; Crosier and Griffin, 2010; Qiao et al., 2010). What is common to all of these approaches is that some important features are firstly extracted, after which a classifier is used to determine if the presented features belong to an object of interest or not.

2.2.1 Template matching approach

In template matching approach, a template (representative example) of the object for detection is used to match the target image (Choi and Kim, 2002; Ullah and Kaneko, 2004). In order to realize this detection task, the image of the template should be smaller than the target image so that image search can be carried out. When searching the target with the template image (containing an example of the object to be detected), a metric is used to evaluate the correlation of the template and the captured region in the target image. Inasmuch as this approach is simple and straight forward, some problems which seriously impact on the performance of such an approach include large sensitivity to object scale, orientation, illumination, noise, etc (Oren et al., 1997)

Although, the sensitivity of the template matching approach to scale can be reduced by using object templates at different scales to search the target image, the problem is far from resolved. Also, the orientation of objects in the target image is another highly challenging problem for object detection. Also, this problem can be trivially solved by rotating templates through various angles during the search of target images. For illumination and noise constraints, the template matching approach performance drastically reduces. As discussed above that some template manipulation schemes can be used to improve the performance of template matching systems, one obvious downside to this is that required computational requirements increases significantly. Also, a detailed analysis of the template matching approach seems to reveal that it is more suited for object classification task rather than object detection tasks. This is obvious considering that in most object classification tasks, presented target images have constrained scenes or background. This contrasts with object detection tasks, in which most cases the backgrounds are unconstrained and therefore can assume any random scene; hence, providing too many possible scenes which can make finding an object in a target image using templates very hard.

2.2.2 Feature-based computer vision approach

Generally, in traditional computer vision approach for object detection, handcrafted features are extracted from target images using a bank of pre-defined filters. A bank of filters contains filters which are so designed extract specific information from target images. Some of the important features which are usually detected include edges, corners, blobs, ridges, object parts, etc. Common operations that are used to realize feature detectors in computer vision include delta function, Gaussian derivatives, Gabor filters coefficients (Camaniciu et al., 2000; Bay et al., 2008; Kyrki et al., 2004).

The detection of edges is used for finding horizon, object corners, tracking lines, and determining the shape of objects.

The detection of blobs in images can be used to determine if some group of connected pixels are related to one another. Blob detection finds very useful applications in image segmentation and the counting of objects in images.

Also, in some applications, pixel classification is carried out. The idea is to assign every pixel in an image to an object label. Although, pixel classification algorithms are usually computationally expensive, they have found usefulness in important tasks such as biomedical image segmentation, geo-satellite image segmentation, obstacle avoidance systems, etc. (Sudowe and Leibe, 2011; Lienhart et al., 2003).

Furthermore, the detected features can be used to build a rule based system in order to achieve some other higher level tasks such as identification, classification, etc. Also, in many researches, the features detected as concatenated to form several observations or training examples which can be used to train a simple classifier.

2.2.3 Sliding window machine learning approach

In such approach, a window of reasonable size, e.g. m×n, is played out a pursuit over the objective image (Sudowe and Leibe, 2011). Initial, a classifier is prepared on an accumulation of preparing tests spreading over the protest of enthusiasm for detection as one class and irregular questions as alternate class. Formally, tests having a place with the protest of enthusiasm for detection are alluded to as positive cases, while arbitrary question tests of no intrigue are alluded to as negative cases. For a solitary protest detection errand, the thought is to prepare a parallel classifier, which decides whether the exhibited question is 'sure' or 'negative'. The prepared classifier would then be able to be utilized to 'review' an objective image by examining it, beginning from the upper left corner. It is vital that the information measurement of the prepared classifier is by and large a small amount of the size or measurement of the objective image; thus, inspecting of target images can be accomplished.

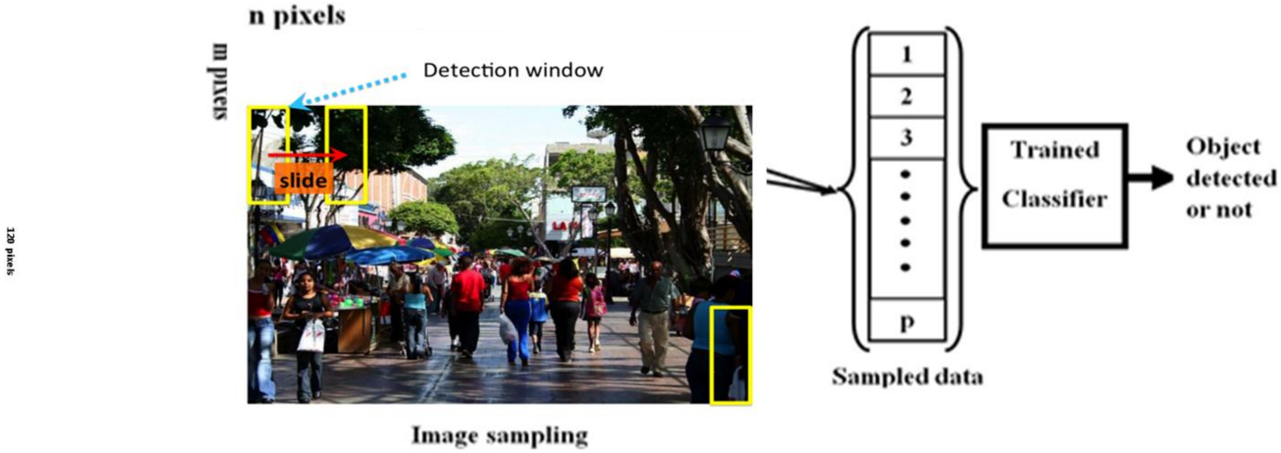

Figure 2.1: Example of the sliding window approach (Sudowe and Leibe, 2011)

From Figure 2.1, it is seen that a window of size m×n pixels is slid across the target image for object detection. Furthermore, it will be observed that there c patches (windows) sampled ‘column-wise’ along target image, and r patches (windows) sampled ‘row-wise’ along target image. Hence, it can be said that a total of r×c patches (windows) are sampled for the target image. The target image patches are concatenated to form a test set as shown in Figure 2. The test set is fed into the trained classifier which ‘looks’ at each patch or window of size m×n, and therefore determines if the object of interest is present or not. Also noteworthy is that p= r×c.

In as much as any suitable classifier that can reasonably handle data with high dimensionality can be used for the detection tasks, some critical concerns to be addressed are considered below.

1. A suitable classifier should have reasonably low training time. That is, the time that is required for the classifier to learn the presented training data. This is important because in

most real-life vision tasks, it is the aim to achieve object detection with very low latency. For example, in robotic based handling environments, response time is expected to be as short as possible.

2. A suitable classifier should have good generalization power on the object of interest for detection. That is, the classifier should be capable of identifying a large number of the object of interest that is not contained in the training set. Moderate variations in the object of interest for detection should not lead to a severe collapse in performance of the classifier. This attribute is extremely important since the classifier will be tasked with object detection in target images of different or very inconsistent backgrounds. This means that a suitable classifier should be capable of learning important features of the object of interest which are useful for the identification of objects belonging to the same category irrespective of image scene or background.

A portion of the classifiers that include discovered applications inside the setting of question detection incorporate deep neural network (DNN), convolutional neural network (CNN), decision trees (DT), and so forth (Ciresan et al., 2010; Turaga et al., 2010; Valstar et al., 2010). Considering the previously mentioned contemplations for choosing an appropriate classifier for question detection, the help vector machine, which is a most extreme edge classifier would be a conspicuous decision, however for the long preparing time when contrasted with backpropagation neural network and choice trees. In perspective of required preparing time, choice trees (DTs) ordinarily have the slightest preparing time when contrasted with the SVM and BPNN; notwithstanding, choice trees tend to rapidly overfit or 'retain' the preparation information (Ciresan et al., 2010). The result is with the end goal that the execution of choice trees on the test set (inconspicuous illustrations) isn't focused. The backpropagation neural network (BPNN) is by all accounts the unobtrusive exchange off between preparing time and speculation control; since the BPNN has a preparation time that is in the middle of that of the help vector machine and choice tree and a speculation execution that is superior to anything that of choice trees and focused with that of the help vector machine. Subsequently, in this undertaking work, the backpropagation neural network has been utilized as the classifier for the protest detection errand.

2.3 Related Works

Many researches have been conducted for the left ventricle detection, however, all previous researches aimed to completely segment the ventricle which in not lie our case here, in which we aim to detect the left ventricle as a square window without segmenting it. One research has been conducted on the use of Fully Convolutional Neural Network (FCN) for Cardiac Segmentation in Short-Axis MRI (Tran, 2016). In this study, it has been focused on tackling the problem of automated left and right ventricle segmentation through the application of FCN, and were able to show that the FCN achieves state-of-the-art semantic segmentation in short-axis cardiac MRI acquired at multiple sites and from different scanners. The proposed FCN architecture used was efficiently trained end-to-end on a graphics processing unit (GPU) in a single learning stage to make inference at every pixel. This model was tested and it independently segments each image in milliseconds, therefore it could be employed in parallel on clusters of CPUs, GPUs, or both for scalable and accurate ventricle segmentation.

Dataset used were obtained from Sunnybrook Cardiac Data, which comprises of cine MRI from 45 patients, having a mix of cardiac conditions. Expert manual segmentation contours for the endocardium, epicardium, and papillary muscles were provided for basal through apical slices at both end-diastole (ED) and end-systole (ES) phases.

Another set of data used in this study came from Left Ventricle Segmentation Challenge (LVSC). It comprises of 200 patients with coronary artery disease and myocardial infarction. The LVSC dataset comes with expert-guided semi-automated segmentation contours for the myocardium. In addition to LVSC data, Right Ventricle Segmentation Challenge dataset were also provided.

The main limitation of FCN model lies in its inability to segment cardiac objects in difficult slices of the heart, especially at the apex.

In another research conducted by (Narayan, 2014), they trained a Convolutional Neural Network to perform semantic segmentation on images obtained from cardiac MRI scan in order to localize the left ventricle and influence the system to measure the capacity of the ventricle during the course of a heartbeat. Several pre- processing and data augmentation steps aimed at generalizing and preventing over fitting by use of image filters were done. Lastly, they were able to demonstrate the left ventricle as a single filled bordering region.

This group employed the method of fully convolutional semantic segmentation model which combines pooling and upscaling layers to both start and end with the same resolution. They implemented the model in Theano/Lasagne and train the model on a NVIDIA GRID K540 GPU. They employed simple upscaling layers paired with convolutions as a rough approximation.

In order to get at this result, they obtained data from two sources, one was from the Sunnybrook Cardiac Data which consists of Cardiac MRI images for 45 patients, a few with healthy hearts and most with different heart conditions. For every patient, a subdivision of the images has ground-truth delineation outlined by expert cardiologists.

The second dataset comes from the Kaggle Second Annual Data Science Bowl. It consisted of Cardiac MRI images for 500 patients and end-systolic and end-diastolic left ventricle volumes for each patient.

All images from both sources were resized to either 192 x 256, 256 x 192, or 256 x 256, base on their original shape, and then the central 128 x 128 portion was taken. This was done due to discrepancies between the images from the first data set and the second data. Images from the second data set came from different hospitals, patients of different age and different health conditions.

This model was evaluated on two tasks. Firstly, they ran it on test set observations with ground-truth contours from the Sunnybrook data and compare the pixel-level predictions. Secondly, it was applied to the Kaggle data to compute left ventricle volumes for patients and measure its accuracy there.

2.4 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.

2.4.1 Image feature extraction and manipulation operations

Image processing is a critical part of pattern recognition and machine learning field. It offers different strategies to control image information, feature extraction, image upgrade, and image division. Image control procedures incorporate image testing for up-scaling or down-scaling, transformation to dark images, high contrast, and so on.

Feature extraction is a procedure in image processing, where a few attributes or parameters that portray an image are acquired. The features of intrigue generally fluctuate for various issues. By and large, these features are factual parameters that portray some essential properties of the images.

Feature extraction operations incorporate as edge detection, corners, focuses, and so on. These operations are extremely valuable in decreasing the measure of immaterial or repetitive data that are contained in images. Filters are extraordinary bits which have predefined pixel esteems to such an extent that it accomplishes the specific feature extraction of intrigue when connected to an image. Normal filters utilized as a part of feature extraction are the Sobel channel, Canny channel, Hough channel, and so on (Vairalkar and Nimbhorkar, 2012; Ding and Goshtasby, 2001; Barinova et al., 2012).

Image enhancement suffices in situations where there is a need to improve image quality, the image characteristics are manipulated such that an improved image is obtained on the considered image property. Some common image enhancement operations include contrast balance, histogram equalization, image denoising, image sharpening, etc (Al-Wadud et al., 2007; Starck et al., 2002).

Image segmentation includes the way toward attempting to isolate out a district or a few areas of an image of intrigue. This operation is extremely utilized as a part of restorative image processing, where a few locales of an image entire image are set apart out from the foundation. The checked or featured locale of intrigue is alluded to as the closer view. One normal and viable strategy utilized as a part of image segmentation is known as image thresholding.

Image thresholding can either be local or global. Local image thresholding requires that not the whole image is considered for the segmentation operation; while, global thresholding uses the whole image during segmentation.

Image thresholding can be achieved using the equation provided below.

$$g_{out}(u,v) = 0, \quad if\ g_{in}(u,v) < T \tag{2.1}$$

$$g_{out}(u,v) = 1, \quad if\ g_{in}(u,v) \geq T \tag{2.2}$$

where, $g_{in}(u,v)$ is the considered thresholded pixel, $g_{out}(u,v)$ is the result of the thresholding, and T is the pixel used for the thresholding operation (Bradley and Roth, 2007).

2.5 Artificial Neural Network (ANN)

In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper on how neurons might work. In order to describe how neurons in the brain might work, they modeled a simple neural network using electrical circuits (McCulloch and Pitts, 1943)

In 1949, Donald Hebb wrote The Organization of Behavior, a work which pointed out the fact that neural pathways are strengthened each time they are used, a concept fundamentally essential to the ways in which humans learn. If two nerves fire at the same time, he argued, the connection between them is enhanced (Hebb, 2005).

In 1959, Bernard Widrow and Marcian Hoff of Stanford developed models called "ADALINE" and "MADALINE." In a typical display of Stanford's love for acronymns, the names come from their use of Multiple ADAptive LINear Elements. ADALINE was developed to recognize binary patterns so that if it was reading streaming bits from a phone line, it could predict the next bit. MADALINE was the first neural network applied to a real world problem, using an adaptive filter that eliminates echoes on phone lines (Graupe, 2013). While the system is as ancient as air traffic control systems, like air traffic control systems, it is still in commercial use.

In 1962, Widrow & Hoff developed a learning procedure that examines the value before the weight adjusts it (i.e. 0 or 1) according to the rule:

$$\Delta W = (\text{pre-weight line value}) * (\text{error} / (\text{number of inputs})) \qquad (2.3)$$

where Δw is the weight change (Abdi et al., 1996).

It depends on the possibility that while one dynamic perceptron may have a major blunder, one can modify the weight esteems to convey it over the network, or if nothing else to nearby perceptrons. Applying this control still outcomes in a blunder if the line before the weight is 0, in spite of the fact that this will in the long run adjust itself. On the off chance that the blunder is monitored so every last bit of it is appropriated to the greater part of the weights then the mistake is dispensed with.

A artificial neural network is an arrangement of basic interconnected computational units called neurons or perceptrons; they are endeavors to mimic the structure and capacity of the cerebrum.

A neural network's capacity to perform calculations depends on the expectation that we can repeat a portion of the adaptability and energy of the human mind by simulated means (Zurada, 1992).

The neurons are associated by joins, and each connection has a numerical weight related with it. Weights are the fundamental methods for long haul memory in ANNs.

An Artificial Neural Network (ANN) is a scientific model that tries to mimic the structure and functionalities of natural neural networks (Krenker et al., 2011). These computational units are ordinarily alluded to as counterfeit neurons and they are the essential building squares of every single neural network. The natural neuron is appeared in Figure 2.2.

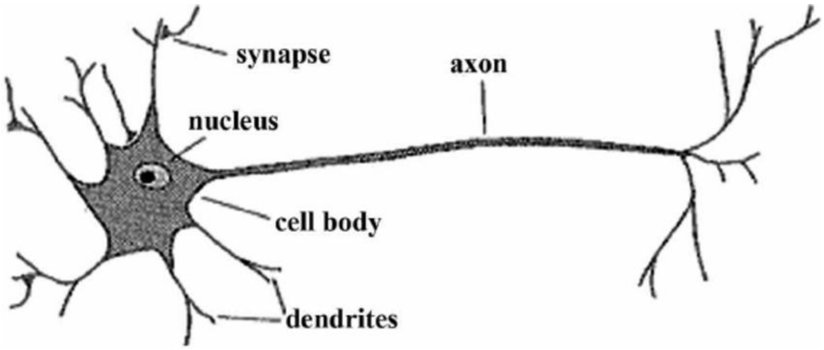

Figure 2.2: Biological neuron (Graupe, 2013)

Graupe, in his book, depicted fake neural networks as computational networks, which endeavor to recreate, in a gross way, the network of nerve cells (neurons) of the organic (human or creature) focal sensory system (Graupe, 2013)

The neurons get sources of info and deliver a yield which is a weighted entirety of the data sources known as the total potential (T.P), after which it is contrasted and the set edge. Neurons are terminated if the total potential surpasses the limit, and not fire if the total potential is not as much as the edge.

Artificial neural networks recreate the cerebrum in two center territories which are structure and capacities; neural networks have features which are at any rate generally proportional to the ones found in natural neurons. Manufactured neural networks are an endeavor at displaying the data processing abilities of sensory systems (Rojas, 2013).

The contributions to artificial neurons relates to the dendrites discovered organic neurons which get the jolts for processing, the hub compares to the cell body in natural neurons which is the place the calculation of data happens, weights relates to the synaptic weights found in natural neurons and fill in as memory to the neuron, the yield relates to the axons in natural neurons and it is the place the total potential is provided in the wake of processing.

The counterfeit neuron is demonstrated as follows, and the conditions communicating the connection between the data sources and the yield. The artificial neuron is appeared in Figure 2.3.

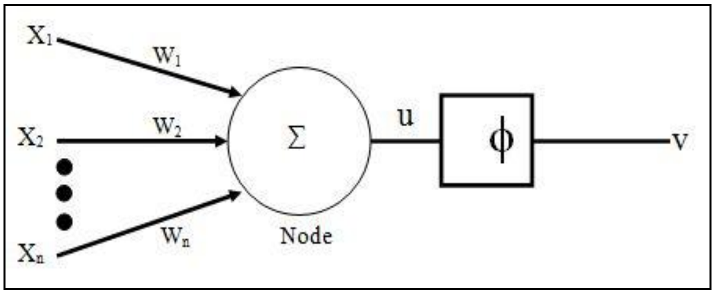

Figure 2.3: Artificial neuron (Fritzke, 1994)

$$u = \sum_{1}^{j} w_{ij} x_j \tag{2.4}$$

$$v = \Phi(u) \tag{2.5}$$

Where, I is the ith hub or neuron, j is the record of the data sources, wij is the weight interconnection from input j to neuron me, x1, x2, … .., xn, are the contributions to the neuron, w1, w2,… ., wn, are the comparing weights to the neurons, u is the aggregate potential, ϕ is the enactment which can be utilized to acquaint nonlinearity with the connection between the information sources and yield, and v is the yield of the network. Note, that different capacities are utilized as enactment capacities or exchange capacities; regular capacities in neural networks incorporate log-sigmoid, tan-sigmoid, Signum, straight, Gaussian and so on.

The single neuron has been discovered unequipped for processing only any capacity; subsequently the development of the multilayer neural networks, which have the ability to register any discretionary capacity or speak to any mind boggling relationship mapping the contributions to the yield diverse issues.

2.5.1 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 labelling.
- The labelling can be carried out even if the labels are only available for a small number of objects representatives of the desired classes (Fritzke, 1994).

2.4.2 Supervised learning rules

- Perceptron learning rule

There are several different models of supervised learning that have been implemented in artificial neural networks.

$$T.P = \sum_{i=1}^{m} w_{ji} x_i \tag{2.6}$$

$$\begin{aligned} &If\ T.P \geq \theta,\quad then\ y = 1 \\ &else\ T.P < \theta,\ then\ y = 0 \end{aligned} \tag{2.7}$$

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.

$$w_j = w_j(t) + \eta(d - y)x \tag{2.8}$$

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 (Freund and Schapire, 1999).

- Delta learning rule

An option yet related way to deal with the perceptron learning guideline is known as the delta run the show. While the perceptron preparing guideline depends on changing weights as per some portion of the contrast between the yield and focus on, the delta manage depends on the more broad thought of "angle plummet". For instance, consider the errand of preparing a solitary TLU with an arrangement of info patterns p, each with a coveted target yield tp. The worldwide mistake E is a component of the weights w. That is, as weights change, the blunder changes. The objective is to move in "weight space" down the incline of the blunder work regarding each weight. The span of the progression ought to be corresponding to the size of the slant. How is the slant computed? Utilizing math the slant might be communicated as the halfway subsidiary of the mistake as for the weight:

$$\Delta w_j = -\alpha \frac{\delta E}{\delta w_j} \tag{2.9}$$

Where, α is the learning rate

On the off chance that ep is the error delivered by a network processing a specific pattern p, at that point the worldwide error E is the mean error created over all the diverse patterns in the preparation set:

$$E = \frac{1}{N}\sum_{p=1}^{N} e^p \tag{2.10}$$

Where, N is the number of patterns in the training set.

The most straightforward method for deciding the pattern error ep is just the objective yield short the real yield:

$$e^p = t^p - y^p \tag{2.11}$$

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

In any case, the above condition has a few issues. To begin with the subtraction implies that the term might be either positive or negative instead of a basic size and may along these lines entangle facilitate counts. This issue is overseen by squaring the term:

$$e^p = (t^p - y^p)^2 \tag{2.12}$$

The second issue we experience is more unobtrusive. Keeping in mind the end goal to perform inclination drop esteems must be ceaseless. This can be cured by substituting initiation an instead of the yield y. Despite the fact that while doing this the objective ought to be precisely characterized - if the limit is set to 0 then one target ought to be set as positive and the other negative e.g. - 1 and 1.

$$e^{p} = {}^{1}\!/_{2}(t^{p} - y^{p})^{2} \tag{2.13}$$

Since E is the mean of all patterns one can't in fact figure dE/dwi until the point that the whole arrangement of patterns is accessible. Be that as it may, this is computationally serious, so de/dwi is typically performed exclusively with each preparation pattern as an estimate as demonstrated as follows.

$$\delta e^{p} \big/ \delta w_{i} = -(t^{p} - y^{p})x^{p} \tag{2.14}$$

CHAPTER 3
DATA COLLECTION & METHODOLOGY

3.1 Overview

This chapter describes the collection of data that is used in this work, methodology and techniques which are implemented to realize the aim of this research, and the simulation of the developed system to show its effectiveness.

3.2 Methodology

It is the plan to build up an artificial vision framework that can play out the undertaking of identifying left ventricle in MR slices. In this work, considering difficulties, for example, protest light, scale, interpretation, revolution, and so forth which make the recognition an intricate issue for such an open identification issue, we make plans to actualizing a wise framework which can to some degree generous adapt to the previously mentioned discovery limitations. Neural network, in particular, the back proliferation neural network (BPNN) has been utilized as a part of this work as the 'cerebrum' behind the recognition.

This exploration is accomplished in two stages. Initially is the left ventricle part acknowledgment stage via preparing a back propagation neural network (BPNN). The second stage is the recognition of left ventricle objects from pictures utilizing the back propagation neural network. The flowchart for the framework is appeared in Figure 3.1; and the two stages are quickly depicted beneath.

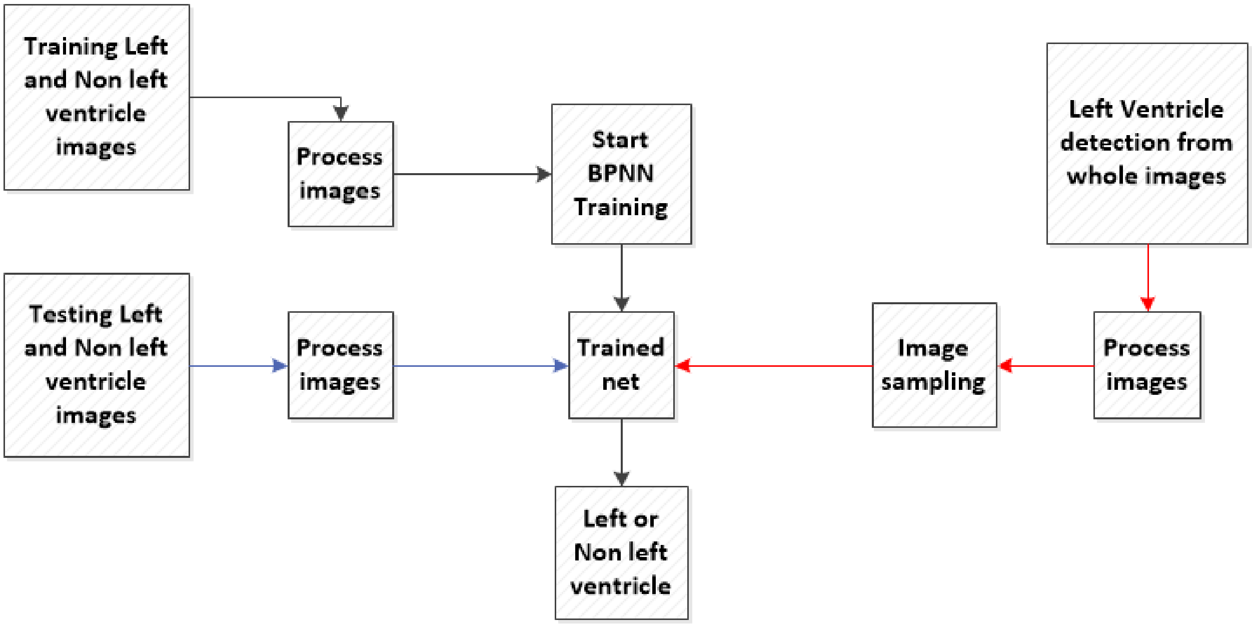
Training Left and Non left ventricle images
Process images
Start BPNN Training
Left Ventricle detection from whole images
Testing Left and Non left ventricle images
Process images
Trained net
Image sampling
Process images
Left or Non left ventricle

Figure 3.1: Flowchart for developed system

3.2.1 Left ventricle recognition

In this stage, a back propagation neural network is prepared to perceive left ventricle objects and non-objects. Keeping in mind the end goal to accomplish this paired characterization errand, training data is gathered to traverse both left ventricle objects and non-left ventricle objects. In this venture, all training and testing data are gathered from the web. Since, the real intrigue is to build up a framework that perceives left ventricle objects, pictures containing left ventricle objects are alluded to as positive illustrations or tests. On the other hand, pictures containing irregular non-left ventricle objects are alluded to as negative illustrations or tests. Note that exclusive divided left ventricle objects are gathered from the web as positive cases; while, no imperative is put on substance of the negative cases with the exception of that they don't contain left ventricle objects. Tests of positive and negative cases gathered from the web are appeared beneath in Figure 3.2.

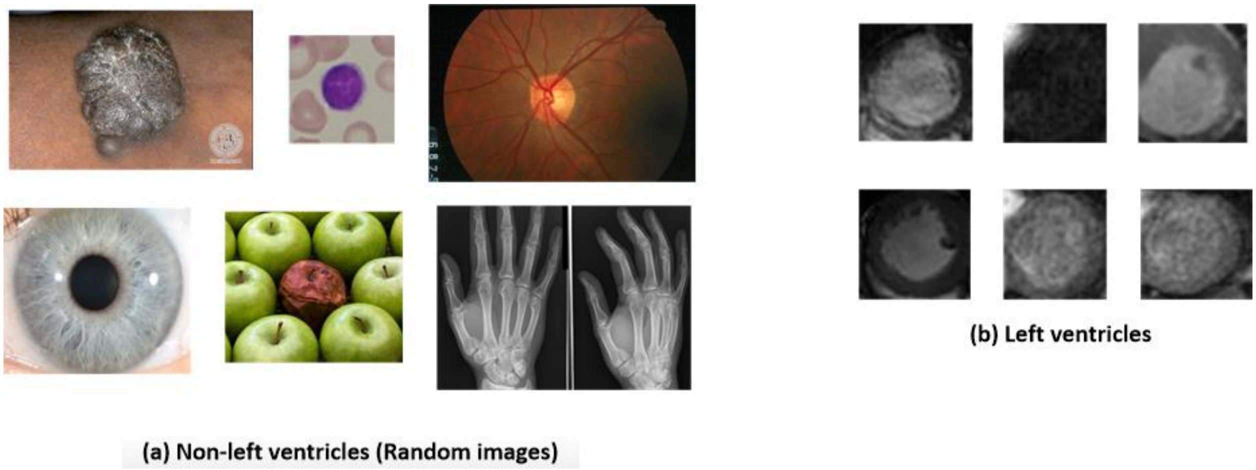

Figure 3.2: Samples for training images (Deng and Radau, 2009)

3.2.1 Image processing

The collected positive and negative examples are transformed from colour to grayscale images. This converts the three RGB colour channels to a single channel representing the brightness or intensities of pixels at different locations in the images. The transformation is achieved using a relation given in Equation 3.1 Samples of collected positive and negative examples transformed to grayscale are shown in Figure 3.3.

$$f'(x,y)=\frac{(R+G+B)}{3} \tag{3.1}$$

Where, f'(x,y) is the transformed pixel for the original RGB pixels.

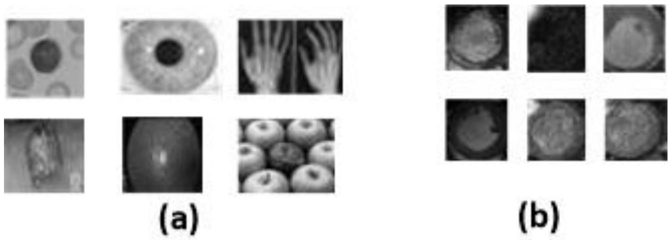

Figure 3.3: Grayscale samples

Since the images are collected on the internet, they are of different sizes. In order to make the images consistent, they are all resized to 40×40 pixels (1600 pixels).

3.2.2 Back propagation neural network (BPNN) design, training and testing

A back propagation neural network is trained on the collected samples spanning both positive and negative examples. For the positive examples (left ventricle objects), 50 samples are collected from the database; while, for the negative examples (non-left ventricle objects), 60 samples are randomly collected from the internet. The positive and negative examples frame the training and testing data for the outlined back propagation neural network (BPNN). All pictures are changed over to from shading to grayscale, and rescaled to 40×40 pixels (1600 pixels). The entire data is separated into training and testing data. The testing data permits the perception of execution of the prepared BPNN on inconspicuous or new data. The testing data allows the observation of performance of the trained BPNN on unseen or new data. It is very desirable that trained ANNs can perform well on unseen data. i.e. generalization. 50 left ventricle objects and 60 non-left ventricle objects are used for training; while, 45 left ventricle objects and 33 non-left ventricle objects are used for testing the trained BPNN. Hence, a total of 110 training images and 78 testing images. Table 3.5 shows the data used for training and testing the network. Note that the left ventricles are collected by cropping the full heart MRI images from the SunnyBrook database (Radau et al., 2009), while the non-left ventricle are obtained from the ImageNet (Deng et al., 2009).

Table 3.1: Dataset description

Data	Training	Testing
Left ventricles (Radau et al., 2009)	50	45
Non-left ventricles (Deng et al., 2009)	60	33
Total	110	78

(a) Input data and neurons

Considering that the training images are now 40×40 pixels, the designed BPNN has 1600 input neurons. Where, each input attribute or pixel is fed into one of the input neurons. Also, note that the input neurons are non-processing. i.e. they basically receive input pixels and supply them to the hidden layer neurons which are processing neurons.

(b) Hidden layer neurons

The hidden layer is where the extraction of input data features that allows the mapping of input data to corresponding target classes is achieved. Unlike the input layer neurons, the hidden layer neurons are processing. Also, each hidden layer neuron receives inputs from all the input layer neurons. In this work, several experiments are carried out to determine the suitable number of hidden layer neurons. Finally, the number of suitable hidden neurons was obtained as 70 during network training.

(c) Output layer coding

Considering that we aim to classify all images as left ventricle object or non-left ventricle object, the BPNN has two output neurons. The output of the BPNN is coded such that output neurons activations are as shown below.

- a left ventricle object: [1 0]
- a non-left ventricle object: [0 1]

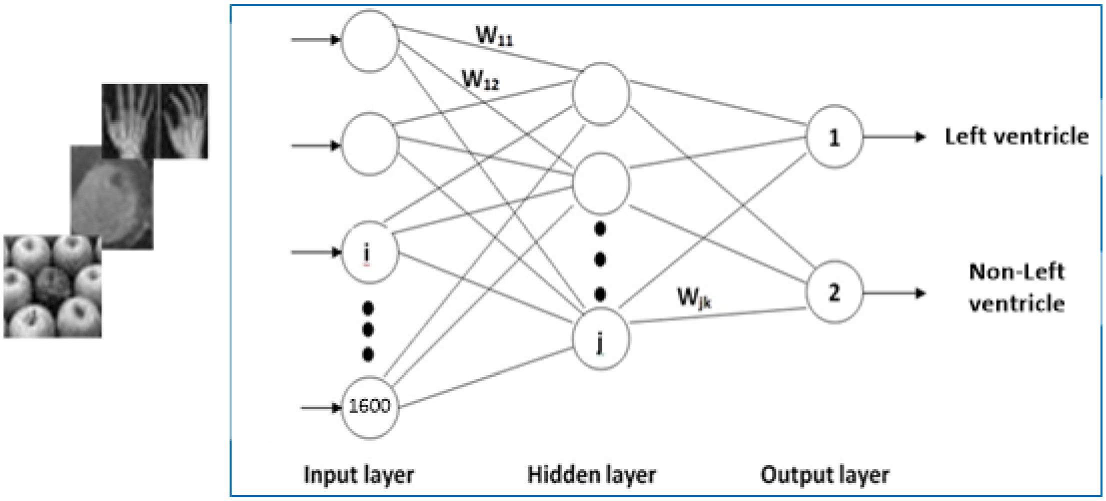

Figure 3.4: Designed back propagation neural network (BPNN) (Donoho, 2002)

Figure 3.4 shows the designed BPNN. The BPNN is trained on the processed images described above. The final training parameters are shown in Table 3.2.

Table 3.2: Final training parameters for BPNN

Number of training images	110
Number of input neurons	1600
Number of hidden neurons	70
Activation function	Log-Sigmoid
Learning rate	0.23
Momentum rate	0.70
MSE mean square error	0.0099
Maximum number of Epochs	3000
Training time	42 secs

Where, the Log-Sigmoid activation function allows neuron output in the range 0 to 1. From Table 3.1, it is seen that the BPNN achieve the required error of value 0.009 in 42 secs and with 3000 maximum epochs. The learning curve for the BPNN is shown in Figure 3.5.

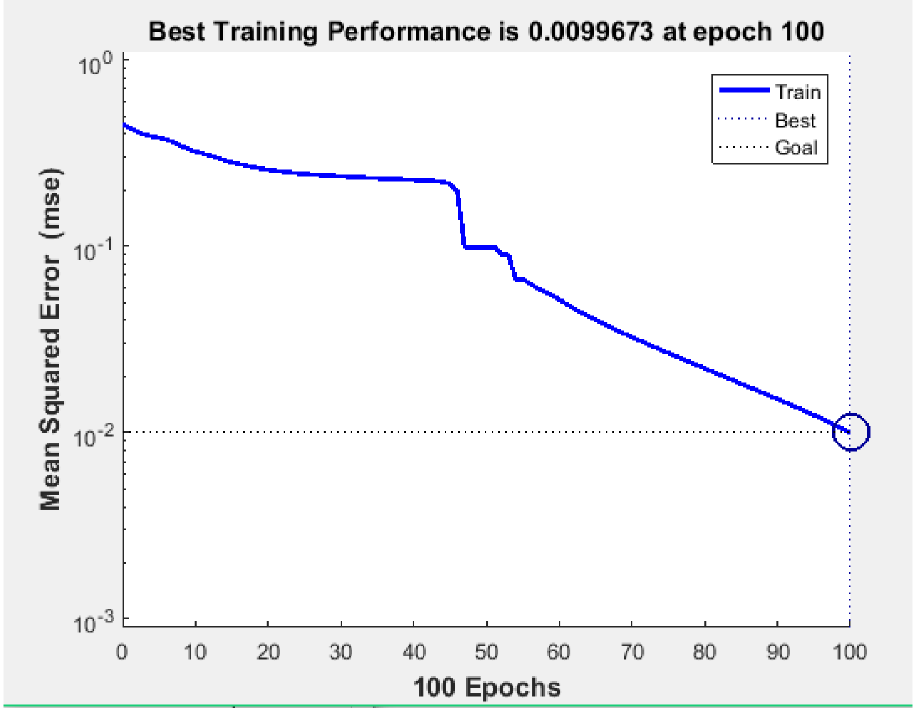

Figure 3.5: Error vs epochs curve for BPNN

The trained BPNN is then tested using the training and testing data. Table 3.3 shows the recognition rates of the BPNN on the training and testing data.

Table 3.3: Recognition rates for BPNN

Parameter	Training	Testing
Number of samples	110	78
Number of samples correctly classified	110	70
Recognition rate	100%	89.23%

It is seen in Table 2 that the BPNN achieved a recognition rate of 100% and 89.23% on the training and testing data, respectively. Note that a testing recognition rate of 89.23% is enough to show that the BPNN can generalize well on unseen data (images). i.e. classifying new images as left ventricle object or non-left ventricle object.

3.2.3 Training with cross validation

Cross-validation can be considered as a reliable way used in order to test the classification or the generalization power of a neural network while it is getting trained. Hence, in some cases and application cross-validation is a useful tool for creating a powerful, intelligent, and machine learning system. This is because a neural network needs a validation set in addition to training set and testing set, which helps in checking over the optimization.

In this work, the data are split first into train and test data. All the presented results are obtained using this technique of splitting. However, for more experiments, we used cross-validation by splitting the data into 70%, 20%, and 10% for training, testing, and validation, respectively. The performance of the network when cross-validation is used has slightly changed as seen in the following table.

Table 3.4: Training with cross-validation

Training sort	**Training** (70%)	**Validation** (10%)	**Testing** (20%)
Recognition rate	99.3%	86.2%	88.97%

As seen in the table, the network recognition rate during the testing has slightly decreased when cross-validation which is possibly due to the small training data used in training the network.

CHAPTER 4
SYSTEM PERFORMANCE

4.1 Left Ventricle Object Detection From Images

In this phase, the trained BPNN is used to detect left ventricle objects in images containing various objects, background, illumination, scale, etc. In order to detect left ventricle objects in new images, the new images are sampled in a non-overlapping fashion using a sliding window or mask. Firstly, all images in which left ventricle objects are to be detected are converted to grayscale and then rescaled to 120×120 pixels; this significantly reduces the required number of samplings and therefore computations. Note that the new size of images containing left ventricle object for detection is selected such that input field (40×40 pixels) of the earlier trained BPNN can fit in without falling off image edges.

It therefore follows that if the new images containing left ventricle object for detection is rescaled to 120×120 pixels, and a sliding window of size 40×40 pixels is used for non-overlapping sampling, 4 samplings are obtained in the x-pixel coordinate and 4 samplings are obtained in the y-pixel coordinate; this makes a total of 9 samplings for an image. Figure 4.1 shows the analogy of the sampling technique.

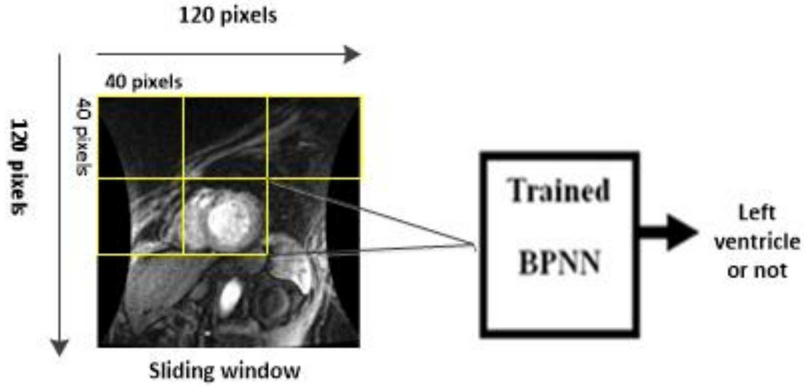

Figure 4.1: Sampling of image for left ventricle detection using trained BPNN

The sampling outcomes using a sliding window of size 40×40 pixels (1600 pixels) is supplied as the input of the trained BPNN as shown in Figure 4. It is expected that for windows containing a left ventricle object, the BPNN gives an output of [1 0]. i.e. as coded during the BPNN training. Also, it is expected that for windows not containing left ventricle objects, the trained BPNN gives an output of [0 1]. From the sampling approach described above, it will be observed that 9 samplings (patches) and therefore predictions are made for any target image. The BPNN output with closest match with the desired output for left ventricle object output, [1 0], is selected as containing a left ventricle object. i.e. with maximum activation value for neuron 1 in Figure 4.2. It is seen that to achieve the complete detection of left ventricle objects in images, both phases 1 and 2 are folded together as one module.

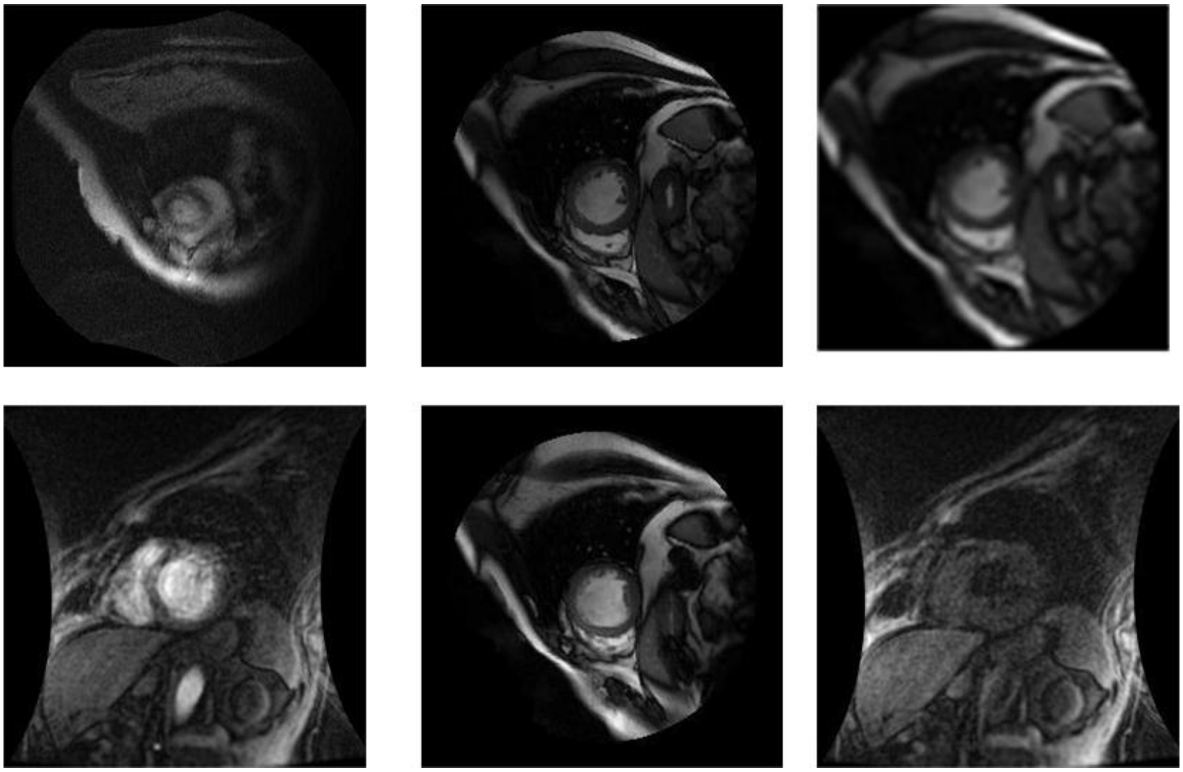

Figure 4.2: Sample of the images used for testing the trained network (Radau, 2009)

An example of left ventricle detection for the image shown in Figure 4.3 is shown in Figure 4.4 using the developed system. The detected left ventricle is highlighted in a rectangular bounding box.

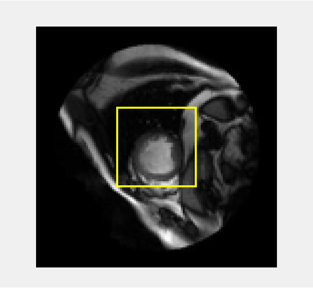

Figure 4.3: Detection outcome

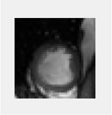

Figure 4.4: The extracted left ventricle

Also, samples of other target images for left ventricle detection using the developed system within this work are shown in Figure 4.5, 4.6 & 4.7. The detected left ventricle objects are highlighted as a rectangular bounding box.

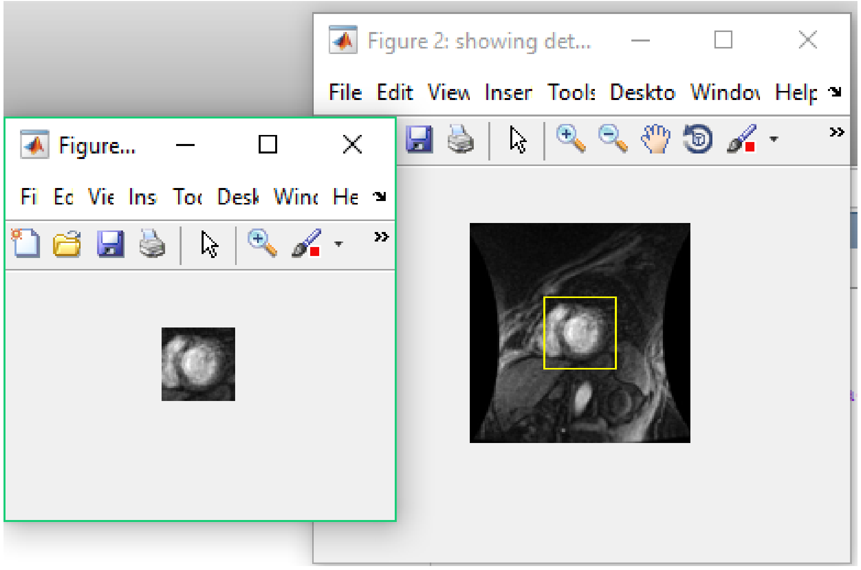
Figure 2: showing det...
File Edit View Inser Tools Deskto Windo Help
Figure...
Fi Ec Vie Ins Toc Desk Winc He

Figure 4.5: Detection outcome using the develop system

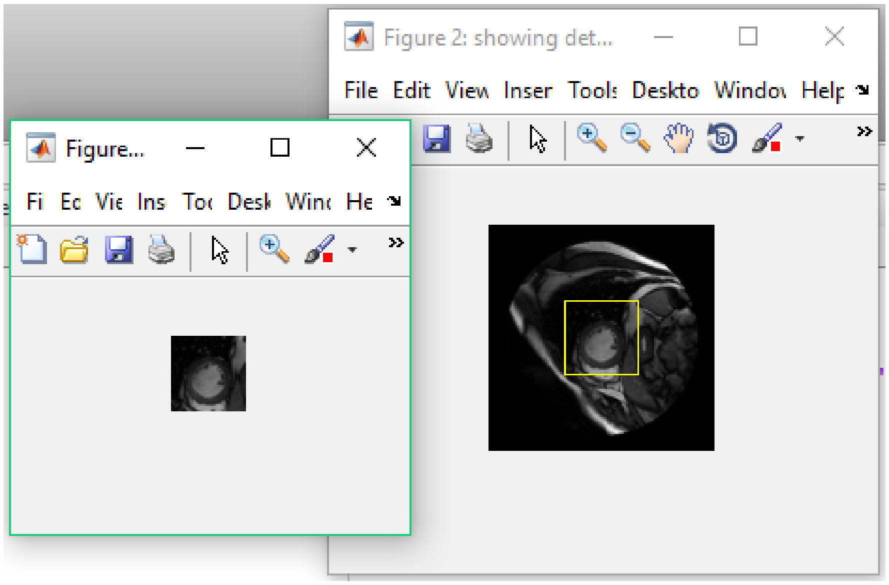
Figure 2: showing det...
File Edit View Inser Tools Deskto Windo Help
Figure...
Fi Ec Vie Ins Toc Desk Winc He

Figure 4.6: Detection outcome using the develop system

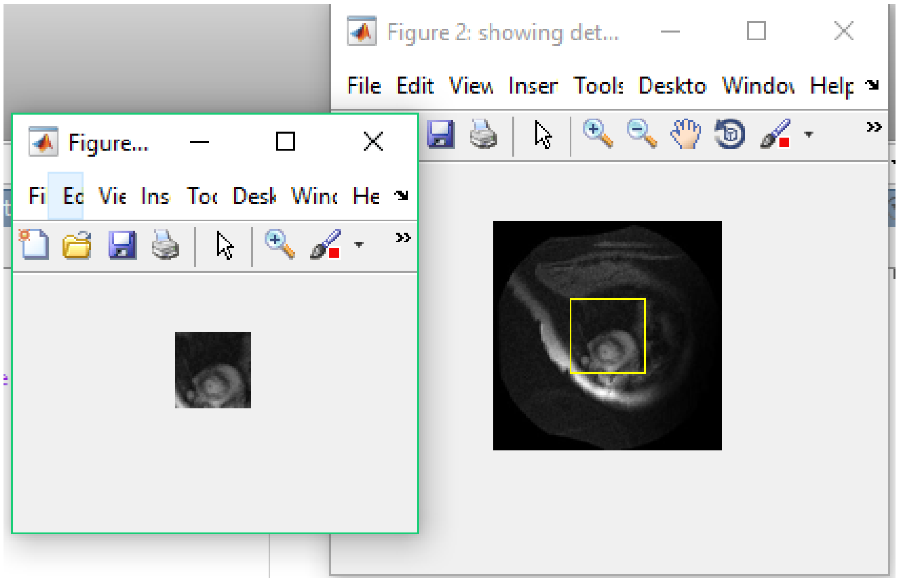
Figure 2: showing det...
File Edit View Inser Tools Deskto Windo Help
Figure...
Fi Ec Vie Ins Toc Desk Winc He

Figure 4.7: Detection outcome using the develop system

Also, some instances where the developed failed to achieve the correct detection of left ventricle object in images are shown in Figure 4.8.

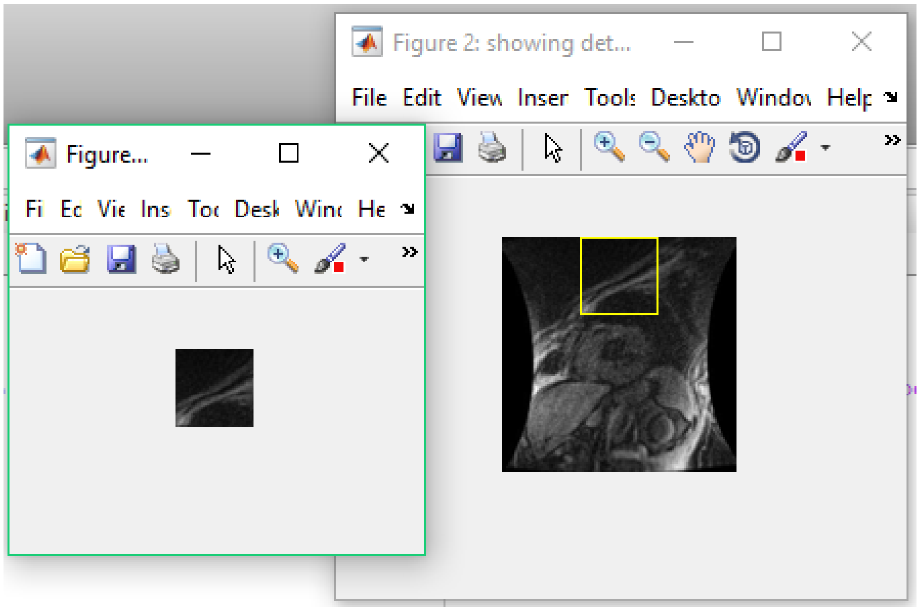
Figure 2: showing det...
File Edit View Inser Tools Deskto Windov Help
Figure...
Fi Ec Vie Ins Toc Desk Winc He

Figure 4.8: Wrong detection outcome using the develop system

4.2 System Evaluation

In order to show the effectiveness of the developed system, we perform experiments using noisy target images. The idea is to intentionally add noise to the target images for left ventricle detection; and then task the developed system to scan the noisy target images for left ventricle objects. To obtain noisy target images, salt and pepper noise at different noise levels are added to the original target images. In this work, salt and pepper noise of densities 3%, 5%, 10% are used. It is also the aim to observe at what noise level that the developed system begin to significantly begin to detect non-left ventricle objects in the target images. Samples of some obtained noisy target images and detection outcomes are shown in Figure 4.9, 4.10, & 4.11.

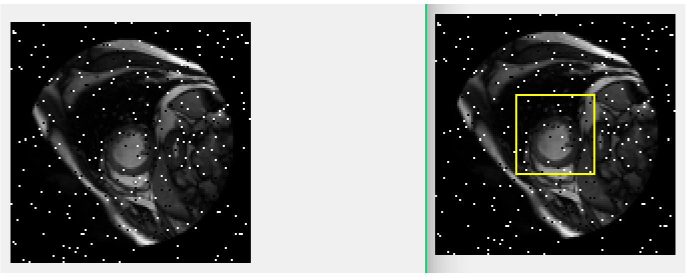

Figure 4.9: Detection outcome with 3% salt & pepper noisy target image using the develop system

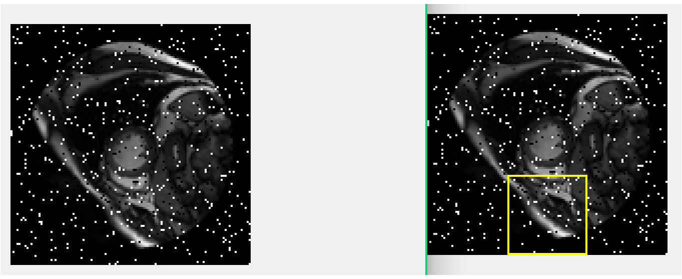

Figure 4.10: Wrong detection outcome with 6% salt & pepper noisy target image using the develop system

As seen in Figure 4.10 the trained network was not able to detect the left ventricle in the first run as it has 6% of salt and pepper noise. However, training the network again was enough to tune it and make it able to detect the left ventricle after two runs.

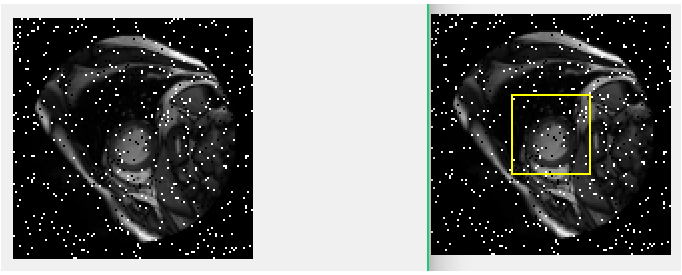

Figure 4.11: Correct detection outcome with 6% salt & pepper noisy target image after training the networks for two more runs

It is shown in Figures 4.9, 4.10, & 4.11.that the developed system can perform effectively the task of left ventricle detection even in the target images with up to 6% salt & pepper noise.

CHAPTER 5
DISCUSSION AND CONCLUSION

5.1 Discussion

Since artificial neural network weights are usually randomly initialized at the start of training, it therefore follows that trained BPNN is not always guaranteed to converge to the global minimum or good local minima. Consequently, the learning of left ventricle objects and non-left ventricle objects can be negatively affected; this therefore affects the detection phase, where the trained BPNN may wrongly predict a sampling window or patch as containing a left ventricle object. In order, to solve this problem, the MATLAB program written contains instructions to retrain the BPNN till a testing recognition (relating to BPNN generalization capability) of greater than 80% is obtained. This greatly reduces the BPNN's probability of wrongly predicting a sampling window (patch) as containing a left ventricle object. In this project, we have allowed for a maximum of 30 retraining schedules of the BPNN. Therefore, when the MATLAB script for the developed whole detection system is run, it is possible that the BPNN may be automatically retrained a couple of times before the detection task is then executed. In all, we found that for most images for detection, less than 7 retraining schedules are required.

Also, another challenge encountered is that even after the BPNN achieves a testing recognition rate of greater than 80%, it is still possible that sampling windows are wrongly classified; though, the probability of this happening is quite small. In this project, is found that when the BPNN achieves a testing recognition of greater 80%, a maximum of 3 retraining schedules is required to correctly detect a left ventricle object in the target image.

This work describes a highly challenging task in computer vision, object detection. We show that back propagation neural network (BPNN) can be employed to learn the robust recognition/classification of left ventricle objects and non-left ventricle objects as positive and negative training examples, respectively. The trained BPNN is then used in a non-overlapping sampling fashion to 'inspect' target images containing left ventricle objects for detection. The developed system is tested and found to be very effective in the detection of left ventricle objects in images containing other objects. Also important is that the developed system is intelligent such that image scene constraints such translation and scale only slightly affects the overall efficiency of the system.

5.2 Recommendations

Finally, a recommendations for the future enhanced work can be the usage of different classifier such as deep network including stacked auto-encoder, convolutional neural network etc. Moreover, other types of classifiers can be also used such as support vector machine (SVM). Those classifiers may result in better detection accuracy in particular, the deep network since they have shown a good efficiency in various detections fields. These networks can prevent the training of networks many times and also the overfitting obtained when the backpropagation neural network is used. Furthermore, the detection time taken for the network to localize the left ventricle may be shorter when deep networks are used.

5.3 Conclusion

The detection of objects in images is an important task required to make efficient and effective use of data. Also, it is interesting to note that vast image data that are now available to us can be leveraged on to achieve tasks such as face recognition, biometrics, security surveillance, etc. However, employing manual labour for such tasks is almost always infeasible, considering the volume of data that have to be processed. Conversely, machines or artificial systems can be employed to perform such somewhat unconventional tasks. Such systems are capable of processing thousands of images or video frames in reasonable time. Furthermore, considering the exponential rise in the capability and processing speed of emerging computing hardware, tasking machines to perform the aforementioned tasks become much more motivating. One of the most fundamental and important tasks in modern day multimedia processing is object detection, where it is the aim to develop artificial systems or machines which are capable of locating an object of interest in images. Although, such tasks are traditionally formulated as computer vision problems. In recent times, the problem of object detection are now being reformulated as that of a machine learning problem. The idea is to build a system that is capable of scanning images and using acquired experience (during training) to determine whether an object of interest is present somewhere in the images. It is noteworthy that the task of object is not one that is easily achieved in computer vision, as some critical scene constraints such as object scale, translation, rotation, illumination, noise, etc. make the task quite difficult.

In fact, for most real life object detection tasks, the environment or scene where the detection task is aimed usually constrained. Moreover, the performance of the artificial vision systems may significantly dwindle when deployed in unconstrained environments. Note that a constrained environment would mean that the developed system has been designed with some prior knowledge and scope of the environment in which the system will be deployed.

In this research, an artificial vision system for left ventricles object detection has been developed. It is important to note that the work in itself is broader than the detection of only left ventricle objects, since the same insight and approach presented within this work can be used to realize the detection of other objects. Also, considering the broadness of the scenes and environments in which the developed system will be deployed, we opt to reformulating the detection task as that of a machine learning problem. This allows some robustness to the aforementioned scene constraints which may render the developed system quite erroneous on the detection task. A back propagation neural network (BPNN) has been used as the learning system in this research. The BPNN is trained on samples of left ventricle and non-left ventricle (random) objects collected from the internet. For the detection of left ventricle objects in target images, a window size (40×40 pixels) corresponding to the size of the input to the BPNN is used to sample the target image in a non-overlapping fashion. The developed system is tested some randomly collected target images containing left ventricle objects without any scene constraints such as the scale, translation, illumination, orientation, etc. of left ventricle objects contained in the target images. The developed system is seen to perform quite well in detecting cub objects, including scenarios where left ventricle objects are even partially occluded.

Furthermore, to show the effectiveness and robustness of the developed system, it is tested on noisy target images. It is seen that the developed artificial vision system is capable of performing the left ventricle object detection task with up to 6% salt & pepper noise.

REFERENCES

Abdi, H., Valentin, D., Edelman, B., & O'Toole, A. J. (1996). A Widrow–Hoff learning rule for a generalization of the linear auto-associator. *Journal of Mathematical Psychology*, *40*(2), 175-182.

Abdullah-Al-Wadud, M., Kabir, M. H., Dewan, M. A. A., & Chae, O. (2007). A dynamic histogram equalization for image contrast enhancement. *IEEE Transactions on Consumer Electronics*, *53*(2).

Abiyev, R. H. (2005, July). Controller based of fuzzy wavelet neural network for control of technological processes. *In Proceeding of the International Conference on Computational Intelligence for Measurement Systems and Applications* (pp. 215-219). USA: Philadelphia.

Abiyev, R. H., & Altunkaya, K. (2008). Personal iris recognition using neural network. *International Journal of Security and Its Applications, 2*(2), 41-50.

Argyrou, M., Maintas, D., Tsoumpas, C., & Stiliaris, E. (2012, October). Tomographic image reconstruction based on artificial neural network (ANN) techniques. In *Proceeding of the Nuclear Science Symposium and Medical Imaging Conference* (pp. 3324-3327). Boston: Birkhäuser.

Barinova, O., Lempitsky, V., & Kholi, P. (2012). On detection of multiple object instances using hough transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(9), 1773-1784.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, *110*(3), 346-359.

Bradley, D., & Roth, G. (2007). Adaptive thresholding using the integral image. *Journal of Graphics Tools*, *12*(2), 13-21.

Choi, M. S., & Kim, W. Y. (2002). A novel two stage template matching method for rotation and illumination invariance. *Pattern Recognition*, *35*(1), 119-129.

Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, *22*(12), 3207-3220.

Comaniciu, D., Ramesh, V., & Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. *In Proceeding of the International Conference on Computer Vision and Pattern Recognition* (pp. 142-149). Canada: Cancouver.

Crosier, M., & Griffin, L. D. (2010). Using basic image features for texture classification. *International Journal of Computer Vision*, *88*(3), 447-460.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. *In Proceeding of the International Conference on Computer Vision and Pattern Recognition* (pp. 248-255). USA: Miami, Florida.

Ding, L., & Goshtasby, A. (2001). On the Canny edge detector. *Pattern Recognition*, *34*(3), 721-725.

Freund, Y., & Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine learning*, *37*(3), 277-296.

Fritzke, B. (1994). Growing cell structures—a self-organizing network for unsupervised and supervised learning. *Neural Networks*, *7*(9), 1441-1460.

Gouk, H. G. and Blake, A. M. Fast sliding window classification with convolutional neural networks. *In Proceedings of the 29th International Conference on Image and Vision Computing* (pp. 114–118). New Zealand: Hamilton.

Graupe, D. (2013). *Principles of artificial neural networks* (Vol. 7). World Scientific.

Hebb, D. O. (2005). *The organization of behavior: A neuropsychological theory*. Psychology Press.

Hsu, R. L., Abdel-Mottaleb, M., & Jain, A. K. (2002). Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(5), 696-706.

Huang, S., Liu, J., Lee, L., Venkatesh, S., Teo, L., Au, C., & Nowinski, W. (2009). Segmentation of the left ventricle from cine MR images using a comprehensive approach. *MIDAS J-Card MR Left Ventricle Segmentation Challenge*, *27*(5), 593-599.

Krenker, A., Bester, J., & Kos, A. (2011). Introduction to the artificial neural networks. In *Artificial Neural Networks-Methodological Advances and Biomedical Applications*. InTech.

Kumar, P., Gupta, D. K., Mishra, V. N., & Prasad, R. (2015). Comparison of support vector machine, artificial neural network, and spectral angle mapper algorithms for crop classification using LISS IV data. *International Journal of Remote Sensing*, *36*(6), 1604-1617.

Kyrki, V., Kamarainen, J. K., & Kälviäinen, H. (2004). Simple Gabor feature space for invariant object recognition. *Pattern Recognition Letters*, *25*(3), 311-318.

Manthalkar, R., Biswas, P. K., & Chatterji, B. N. (2003). Rotation invariant texture classification using even symmetric Gabor filters. *Pattern Recognition Letters*, *24*(12), 2061-2068.

Marak, L., Cousty, J., Najman, L., & Talbot, H. (2009, November). 4D Morphological segmentation and the MICCAI LV-segmentation grand challenge. In Proceedings of the *workshop on cardiac MR left ventricle segmentation challenge* (pp. 1-8). USA: Chicago.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115-133.

Myint, S. W., Gober, P., Brazel, A., Grossman-Clarke, S., & Weng, Q. (2011). Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. *Remote Sensing of Environment*, *115*(5), 1145-1161.

Narayan, T. Automated Left Ventricle Segmentation in Cardiac MRIs using Convolutional Neural Networks. *arXiv preprint arXiv:1705.08653*.

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., & Poggio, T. (1997, June). Pedestrian detection using wavelet templates. *In Proceeding of the 14th International Conference on Computer Vision and Pattern Recognition* (pp. 193-199). Canada: Torronto.

Oyedotun, O. K., Olaniyi, E. O., & Khashman, A. (2015). Deep learning in character recognition considering pattern invariance constraints. *International Journal of Intelligent Systems and Applications*, *7*(7), 1.

Qiao, L., Chen, S., & Tan, X. (2010). Sparsity preserving projections with applications to face recognition. *Pattern Recognition*, *43*(1), 331-341.

Radau, P., Lu, Y., Connelly, K., Paul, G., Dick, A., & Wright, G. (2009). Evaluation framework for algorithms segmenting short axis cardiac MRI. *The MIDAS Journal-Cardiac MR Left Ventricle Segmentation Challenge*, *49*(2).

Rojas, R. (2013). *Neural networks: a systematic introduction*. Springer Science & Business Media.

Sandbach, G., Zafeiriou, S., Pantic, M., & Yin, L. (2012). Static and dynamic 3D facial expression recognition: A comprehensive survey. *Image and Vision Computing*, *30*(10), 683-697.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85-117.

Shrivastava, N., & Tyagi, V. (2014). An effective scheme for image texture classification based on binary local structure pattern. *The Visual Computer*, *30*(11), 1223-1232.

Smeets, D., Loeckx, D., Stijnen, B., De Dobbelaer, B., Vandermeulen, D., & Suetens, P. (2010). Semi-automatic level set segmentation of liver tumors combining a spiralscanning technique with supervised fuzzy pixel classification. *Medical Image Analysis*, *14*(1), 13-20.

Starck, J. L., Candès, E. J., & Donoho, D. L. (2002). The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, *11*(6), 670-684.

Sudowe, P., & Leibe, B. (2011, September). Efficient use of geometric constraints for sliding-window object detection in video. *In Proceeding of the International Conference on Computer Vision Systems* (pp. 11-20). Germany: Berlin.

Tran, P. V. (2016). A fully convolutional neural network for cardiac segmentation in short-axis MRI. *arXiv preprint arXiv:1604.00494*.

Tran, P. V. (2016). A fully convolutional neural network for cardiac segmentation in short-axis MRI. *arXiv preprint arXiv:1604.00494*.

Turaga, S. C., Murray, J. F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., & Seung, H. S. (2010). Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, *22*(2), 511-538.

Ullah, F., & Kaneko, S. I. (2004). Using orientation codes for rotation-invariant template matching. *Pattern Recognition*, *37*(2), 201-209.

Vairalkar, M. K., & Nimbhorkar, S. U. (2012). Edge detection of images using Sobel operator. *International Journal of Emerging Technology and Advanced Engineering*, *2*(1), 291-293.

Valstar, M., Martinez, B., Binefa, X., & Pantic, M. (2010, June). Facial point detection using boosted regression and graph models. In Proceeding of the International Conference on *Computer Vision and Pattern Recognition (CVPR)* (pp. 2729-2736). Germany: Berlin.

Zotti, C., Luo, Z., Humbert, O., Lalande, A., & Jodoin, P. M. (2017). GridNet with automatic shape prior registration for automatic MRI cardiac segmentation. *arXiv preprint arXiv:1705.08943*.

Zou, W. W., & Yuen, P. C. (2012). Very low resolution face recognition problem. *IEEE Transactions on Image Processing*, *21*(1), 327-340.

Zurada, J. M. (1992). *Introduction to artificial neural systems* (Vol. 8). St. Paul: West.

APPENDICES

APPENDIX 1
IMAGE PROCESSING CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%Phase 1: Training BPNN to recognize LV objects

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

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

clear all

close all

%%%%%%%%%%%%%%%%% Training data for positives/LV

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

k=1;

eyed = strcat('C:\Users\TOSHIBA\Documents\MATLAB\B.Sc project on LV detection\ASc
    project\positives\', num2str(k), '.jpg');

a = imread(eyed);

b=mat2gray(a);

d=imresize(b,[40 40]);% pixel no=900

e1=reshape(d,[ ],1);% reshape 32 by 32 1 to 900 by 1 matrix

for k=2:35

eyed = strcat('C:\Users\TOSHIBA\Documents\MATLAB\B.Sc project on LV detection\ASc
    project\positives\', num2str(k), '.jpg');
```

```
a = imread(eyed);

b=mat2gray(a);

d=imresize(b,[40 40]);% pixel no=900

e2=reshape(d,[ ],1);% reshape 32 by 32 1 to 900 by 1 matrix

u=[e1 e2];% concantenation with k=1

e1=u;

end

A_LV=u;

t1=ones(1,35);

t2=zeros(1,35);

A_target=[t1;t2]; % target

LV_all=[A_LV;A_target];

% Training data for negatives/non-LV

 k=1;

eyed = strcat('C:\Users\TOSHIBA\Documents\MATLAB\B.Sc project on LV detection\ASc project\negatives\', num2str(k), '.jpg');

a = imread(eyed);

b=mat2gray(a);

d=imresize(b,[40 40]);% pixel no=900

e1=reshape(d,[ ],1);% reshape 32 by 32 1 to 900 by 1 matrix

for k=2:45
```

```
eyed = strcat('C:\Users\TOSHIBA\Documents\MATLAB\B.Sc project on LV detection\ASc project\negatives\', num2str(k), '.jpg');

a = imread(eyed);

b=mat2gray(a);

d=imresize(b,[40 40]);% pixel no=900

e2=reshape(d,[ ],1);% reshape 32 by 32 1 to 900 by 1 matrix

u1=[e1 e2];% concantenation with k=1

e1=u1;

end

A_random=u1;

t11=zeros(1,45);

t12=ones(1,45);

A1_target=[t11;t12]; % target

random_all=[A_random;A1_target];

%Concatenate all characters

hi= [LV_all, random_all];

%Shuffle training data

A1=hi(:,randperm(size(hi,2)));

% cut out training and target data

train_input=A1(1:1600,:);

train_target=A1(1601:end,:)
```

```
%%%%%%%%%%%%%%%% Testing data for positives/LVs

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

k=36;

eyed = strcat('C:\Users\TOSHIBA\Documents\MATLAB\B.Sc project on LV detection\ASc
          project\positives\', num2str(k), '.jpg');

a = imread(eyed);

b=mat2gray(a);

d=imresize(b,[40 40]);% pixel no=900

e1=reshape(d,[ ],1);% reshape 32 by 32 1 to 900 by 1 matrix

for k=37:39

eyed = strcat('C:\Users\TOSHIBA\Documents\MATLAB\B.Sc project on LV detection\ASc
          project\positives\', num2str(k), '.jpg');

a = imread(eyed);

b=mat2gray(a);

d=imresize(b,[40 40]);% pixel no=900

e2=reshape(d,[ ],1);% reshape 32 by 32 1 to 900 by 1 matrix

u=[e1 e2];% concantenation with k=1

e1=u;

end

A_LV=u;

t1=ones(1,4);
```

```
t2=zeros(1,4);

A_target=[t1;t2]; %

LV_all=[A_LV;A_target];

% Testing data for negatives/non-lv

 k=46;

eyed = strcat('C:\Users\TOSHIBA\Documents\MATLAB\B.Sc project on LV detection\ASc project\negatives\', num2str(k), '.jpg');

a = imread(eyed);

b=mat2gray(a);

d=imresize(b,[40 40]);% pixel no=900

e1=reshape(d,[ ],1);% reshape 32 by 32 1 to 900 by 1 matrix

for k=47:58

eyed = strcat('C:\Users\TOSHIBA\Documents\MATLAB\B.Sc project on LV detection\ASc project\negatives\', num2str(k), '.jpg');

a = imread(eyed);

b=mat2gray(a);

d=imresize(b,[40 40]);% pixel no=900

e2=reshape(d,[ ],1);% reshape 32 by 32 1 to 900 by 1 matrix

u1=[e1 e2];% concantenation with k=1

e1=u1;

end
```

```
A_random=u1;

t11=zeros(1,13);

t12=ones(1,13);

A1_target=[t11;t12]; %

random_all=[A_random;A1_target];

%Concatenate all characters

hi= [LV_all, random_all];

%Shuffle training data

A1=hi(:,randperm(size(hi,2)));

% cut out training and target data

test_input=A1(1:1600,:);

test_target=A1(1601:end,:);

%%%%%%%%%%%%%%%%%%%Training of back propagation neural network to classify LV objects(positive examples) and non-LV objects(negative examples)
```

APPENDIX 2
NEURAL NETWORKS CODE

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

for ii=1:40

% CREATING AND INITIATING THE NETWORK

net = newff(minmax(train_input),[100 2],{'logsig','logsig'},'traingdx');

% TRAINING THE NETWORK

net.trainParam.goal = 0.01; % Sum-squared error goal.

net.trainParam.lr = 0.17;  % Learning Rate.

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

net.trainParam.epochs =3000;% Maximum number of epochs to train.

net.trainParam.mc = 0.8 % Momentum Factor.

[net,tr] = train(net,train_input,train_target);

save('Abed_B.Sc
        project\Trained_BPNN_for_LV_detection.mat','net','train_input','train_target',
        'test_input', 'test_target')

%%%%%%RECOGNITION RATE OF TRAIN DATA

%target max indices

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

%dimensions of target matrix

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

```
%actual output matrix sim_net
sim_net=sim(net,train_input);
[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,
s=double(corrects*100/v); %let recognition rate be w
fprintf('train recognition rate is %d\n',s);
%%%%%%%%%RECOGNITION RATE OF TEST DATA
%target max indices
[M,I_t]=max(test_target);% row vector
%dimensions of target matrix
[u,v]=size(test_target);
%actual output matrix sim_net
sim_net=sim(net,test_input);
[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('test recognition rate is %d\n',w);


if w>=88                    %if recognition rate on test data is greater than 80%....exit, else retrain BPNN

break

end

end

%%%%%%save trained back propagation neural network for target image LV detection

%save('Abed_B.Sc project\Trained_BPNN_for_LV_detection.mat','net','train_input','train_target', 'test_input', 'test_target')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Phase 2: detection of LV object in target image using trained BPNN

%%%%%%%read in target image

img = strcat('C:\Users\TOSHIBA\Documents\MATLAB\B.Sc project on LV detection\Abed_B.Sc project\targetimage2', '.jpg');

sample= imread(img);
```

```
a1=imresize(sample,[120 120]);

a2=mat2gray(a1);

a2=mat2gray(a2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Add noise

J = imnoise(a2,'salt & pepper',0.06);

figure, imshow(J);

a2=J;

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Add noise

%find size of image (doc)

[r,c]=size(a2);

%declare mask size for sampling a2: (t1,t2)

t1=40; t2=40;

%no. of blocks to be sampled: (r1,c1)

r1=floor(r/t1);

c1=floor(c/t2);

m1; n=1;

%capturing blocks in image a2

for jj=1:(r1-1)

if jj==1

a3=a2(m:t1,n:t2);
```

```
a33=reshape(a3,[ ],1); %reshape a3 to column matrix

b1=a33;

b2=b1;

%shift mask to next non-overlapping regions in image by loops

c2=(c1-1);

for ii=1:c2       %sample masks after 1st block to the last on row

n=n+40; t2=t2+40;

a4=a2(m:t1,n:t2);

a33=reshape(a4,[ ],1); %reshape a3 to column matrix

b3=a33;

d=[b1 b3];

b1=d;

end

d1=d;   %save b1 & d in case

m=m+40; t1=t1+40;    %move mask down rows

n=1; t2=40;      %reset n & t2 for sampling new row

c1=floor(c/t2); %reset c1

end

a3=a2(m:t1,n:t2);

a33=reshape(a3,[ ],1); %reshape a3 to column matrix
```

```
f1=a33;
f2=f1;
%hift mask to next non-overlapping regions in image by loops
c2=(c1-1);
for ii=1:c2       %sample masks after 1st block to the last on row
n=n+40; t2=t2+40;
a4=a2(m:t1,n:t2);
a33=reshape(a4,[ ],1); %reshape a3 to column matrix
f3=a33;

e=[f1 f3];
f1=e;
end
e1=[d e];
d=e1;
m=m+40; t1=t1+40;   %move mask down rows
n=1; t2=40;      %reset n & t2 for sampling new row
c1=floor(c/t2); %reset c1
end
input=e1;
```

```
%%%%%load trained ANN for predicting patches containing LV object

load('Abed_B.Sc project\Trained_BPNN_for_LV_detection.mat');

out= sim(net, input);   %simulate trained ANN with target image


%%%%%%algorithm for filtering predicted patches containing LV object

out1=out(1, :);                        % obtain the first row of simulated ANN output which
        shows if patch contains LV object

[x, y]= max(out1);      %obtain the index of image patch with highest activation value as y


fprintf('predicted patch number is %d \n',y);

disp('extract the pixels coordinates for predicted patch from target image grid');

%%%%note that predicted patches range from 1 to 16 has shown in the grid below

   %%%%%%%extracting images patch/patches containing LV objects from target image

if y==1

p1=1; p2=40; q1=1; q2=40;

z =a2(p1:p2, q1:q2);

elseif y==2

p1=1; p2=40; q1=41; q2=80;

z =a2(p1:p2, q1:q2);


elseif y==3
```

```
p1=1; p2=40; q1=81; q2=120;
z =a2(p1:p2, q1:q2);

elseif y==4
p1=1; p2=40; q1=41; q2=80;
z =a2(p1:p2, q1:q2);

elseif y==5
p1=41; p2=80; q1=41; q2=80;
z =a2(p1:p2, q1:q2);

elseif y==6
p1=41; p2=80; q1=41; q2=80;
z =a2(p1:p2, q1:q2);

elseif y==7
p1=41; p2=80; q1=81; q2=120;
z =a2(p1:p2, q1:q2);

elseif y==8
```

```
p1=81; p2=120; q1=41; q2=80;

z =a2(p1:p2, q1:q2);


elseif y==9

p1=81; p2=120; q1=81; q2=120;

% z =a2(81:120, 1:40);

z =a2(p1:p2, q1:q2)

end

figure('Name', 'showing cropped detected LV object');

imshow(z);      %show detected object 'LV'

%%%%%%%%%%%%%%%%%%%%%marking out detected LV object with a rectanglar bounding box in target image

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

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

figure('Name', 'showing detected LV object');

imshow( a2 ); % show original image

hold all;  % hold image and plot rectangle on top

from_x=p1; from_y=q1; width= p2-p1; height= q2-q1;        %coordinates of LV object

rect= [from_y from_x height width];             % marking out location of LV object from target image

rectangle( 'Position', rect, 'EdgeColor','y');
```