

SEGMENTATION OF FEATURES USING NEURAL NETWORK WITH CARDIAC DATASET

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering
by

MANIGANDLA PAVAN KALYAN, k ajay chowdary
(38110297, 38110233)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH - 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
MANIGANDLA PAVAN KALYAN, k ajay chowdary (38110297,
38110233) who carried out the project entitled "**SEGMENTATION OF
FEATURES USING NEURAL NETWORK WITH CARDIAC DATASET**"
under my supervision from **November 2019** to **April 2020**.

Internal Guide

Dr.Suganiya, M.E, Ph.D.,

Head of the Department

Dr.L.Lakshmanan.,M.E.,Ph.D

Dr.S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

WE MANIGANDLA PAVAN KALYAN, k ajay
chowdary (38110297, 38110233)

hereby declare that the Project Report entitled **SEGMENTATIONOF FEATURES
USING NEURAL NETWORK WITH CARDIAC DATASET** done by
me under the guidance of **Dr.Suganiya, M.E., Ph.D.**, is submitted in partial
fulfillment of the requirements for the award of Bachelor of Engineering / Technology
degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D**, Dean, School of Computing , **Dr. S.Vigneshwari M.E., Ph.D.** and **Dr. L.Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.Suganiya M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Accurate segmentation of cardiac bi-ventricle (CBV) from magnetic resonance (MR) images has a great significance to analyze and evaluate the function of the cardiovascular system. However, the majority of cardiac MR images show that the similar intensity distribution in different regions, thus providing a little edge information. In this proposed system the input images are color images means then we convert to gray scale from the color images. The image features like color, weight, and depth and pixel information to apply before the classifier (neural network). The ROI (Region of interest) segmentation algorithm is used in order to detect segment the portion of defected areas. The back propagation neural network concept is used for training the image and testing the image with the help of weight estimating classifier.

TABLE OF CONTENTS

CHAPTER NO.	TOPIC	PAGE:NO
	ABSTRACT	iv
	LIST OF ABBREVIATIONS	viii
	LIST OF FIGURES	ix
1.	INTRODUCTION	01
1.1	IMAGE PROCESSING	02
1.2	TYPES OF DIGITAL IMAGE	02
1.3	COLOR MODEL	02
1.3.1	RGB COLOUR MODEL	02
1.3.2	CONVERTING COLOR TO GRAYSCALE	03
1.3.3	SINGLE CHANNEL COLOR IMAGES	04
1.3.4	HISTOGRAM EQUALIZATION	05
1.4	DIGITAL IMAGE PROCESSING	06
1.4.1	IMAGE REPRESENTATION	06
1.4.2	IMAGE ENHANCEMENT	07
1.4.3	IMAGE SEGMENTATION	07
1.5	FUNDAMENTAL IN DIGITAL IMAGE	08
1.5.1	IMAGE ACQUISITION	08
1.5.2	IMAGE PREPROCESSING	08
1.5.3	IMAGE SEGMENTATION	09

1.5.4	Image representation	09
1.5.5	Image recognition	09
2.	LITERATURE SURVEY	10
3.	AIM AND SCOPE	16
3.1	AIM	16
3.2	SCOPE	16
3.3	EXISTING SYSTEM	16
3.4	PROPOSED SYSTEM	16
3.5	MODULE DESCRIPTION	16
3.6	BLOCK DIAGRAM	17
3.7	SEGMENTATION	17
3.7.1	EDGE-BASED SEGMENTATION	18
3.7.2	REGION-BASED SEGMENTATION	18
3.8	ACTIVE CONTOURS	19
3.8.1	EDGE-BASED ACTIVE CONTOURS	19
3.8.2	REGION-BASED ACTIVE CONTOURS	20
3.9	NEURAL NETWORK	21
3.9.1	HISTORICAL BACKGROUND	21
3.9.2	BASICS OF NEURAL NETWORK	22
3.9.3	NEURAL VS CONVENTIONAL	23
3.9.4	NETWORK LAYERS	24

4.	SYSTEM REQUIREMENTS	25
4.1	HARDWARE REQUIREMENTS	25
4.2	SOFTWARE REQUIREMENTS	25
4.3	INTRODUCTION TO MATLAB	26
4.3.1	Tools and development	26
4.3.2	Mathematical function library	26
4.3.3	The language	26
4.3.4	Graphics	26
4.3.5	External interfaces	27
4.3.6	Image processing toolbox	27
4.3.7	Computer vision toolbox	27
4.3.8	Image Acquisition toolbox	29
5.	RESULTS AND DISCUSSION	30
5.1	RESULTS	30
5.2	DISCUSSION	32
6.	SUMMARY AND CONCLUSION	33
6.1	SUMMARY	33
6.2	CONCLUSION	33
	REFERENCES	34
	APPENDICES	35

LIST OF ABBREVIATIONS

RGB – Red Green Blue

CMYK – Cyan Magenta Yellow Black

LV – Left Ventricle

RV – Right Ventricle

ACDC – Accurate Coordinate Dataset Collection

LVC, RVC – Left and Right Ventricular Cavity

LVM – Left Ventricular Myocardium

MLP – Multi Layer Perceptron

CAD – Computer Assisted Diagnosis

ESV - End Systolic Volume

EDV – End Diastolic Volume

EF – Ejection Fraction

SV – Stroke Volume

VM – Ventricular Mass

ICC – Infraclass Correlation Coefficient

NCD – Non-Communicable Disease

ROI – Region Of Interest

CVD – Cardiovascular Disease

DALY – Disability Adjusted Life Year

ISBI – International Symposium on Biomedical Imaging

DIC – Differential Interference Contrast

CNN – Convolutional Neural Network

CBV – Cardiac Bi-Ventricle

MSE – Mean Square Error

LIST OF FIGURES

FIGURE	TITLE	PAGE:NO
1.1	RGB color model	03
1.2	Composition of RGB from 3 Grayscale images	04
1.3	Original image	05
1.4	Color-equalized image	05
1.5	Image representation	06
1.6	Fundamental steps in digital image processing	08
3.1	Segmentation and classification	17
3.2	Neural network	22
5.1	Preprocessing and Segmentation	30
5.2	Performance, Training State and Error Histogram graph	31

CHAPTER 1

INTRODUCTION

INTRODUCTION TO IMAGE PROCESSING

Image Processing is a method to enhance raw images received from cameras/sensors placed on satellites, space probes and aircrafts or pictures taken in normal day-to-day life for various applications. Various techniques have been developed in Image Processing during the last four to five decades. Most of the techniques are developed for enhancing images obtained from unmanned spacecrafts, space probes and military reconnaissance flights. Image Processing systems are becoming popular due to easy availability of powerful personnel computers, large size memory devices, graphics software's etc. Image processing is a physical process used to convert an image signal into a physical image. The image signal can be either digital or analog. The actual output itself can be an actual physical image or the characteristics of an image.

Image processing is photography. In this process, an image is captured or scans using a camera to create a digital or analog image. In order to produce a physical picture, the image is processed using the appropriate technology based on the input source type. In digital photography, the image is stored as a computer file. This file is translated using photographic software to generate an actual image. The colors, shading, and nuances are all captured at the time the photograph is taken the software translates this information into an image. When creating images using analog photography, the image is burned into a film using a chemical reaction triggered by controlled exposure to light. The image is processed in a darkroom, using special chemicals to create the actual image. This process is decreasing in popularity due to the opening of digital photography, which requires less effort and special training to product images. The field of digital imaging has created a whole range of new applications and tools that were previously impossible. Face recognition software, medical image processing and remote sensing are all possible due to the development of digital image processing. Specialized computer programs are used to enhance.

TYPES OF DIGITAL IMAGE

For photographic purposes, there are two important types of digital images: color and grayscale. Color images are made up of colored pixels while grayscale images are made of pixels in different shades of gray.

- **Grayscale Images:** A grayscale image is made up of pixels, each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different grays. Assuming 256 gray levels, each black and white pixel can be stored in a single byte (8 bits) of memory.
- **Color Images:** A color image is made up of pixels, each of which holds three numbers corresponding to the red, green and blue levels of the image at a particular location. Assuming 256 levels, each color pixel can be stored in three bytes (24 bits) of memory. Note that for images of the same size, a black & white version will use three times less memory than a color version.
- **Binary Images:** Binary images use only a single bit to represent each pixel. Since a bit can only exist in two states- ON or OFF; every pixel in a binary image must be one of two colors, usually black or white. This inability to represent intermediate shades of gray is what limits their usefulness in dealing with photographic images.

COLOR MODEL

RGB Color Model

A representation of additive color mixing. Projection of primary color lights on a screen shows secondary colors where two overlap; the combination of all three of red, green, and blue in appropriate intensities makes white. The RGB color model is an additive color model in which red, green, and blue light is added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.



Fig 1.1 RGB color model

The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography.

Converting Color to Grayscale

Conversion of a color image to grayscale is not unique; different weighting of the color channels effectively represents the effect of shooting black-and-white film with different-colored photographic filters on the cameras.

To convert any color to a grayscale representation of its luminance, first one must obtain the values of its red, green, and blue (RGB) primaries in linear intensity encoding, by gamma expansion. Then, add together 30% of the red value, 59% of the green value, and 11% of the blue value (these weights depend on the exact choice of the RGB primaries, but are typical). Regardless of the scale employed (0.0 to 1.0, 0 to 255, 0% to 100%, etc.) the resultant number is the desired linear luminance value; it typically needs to be gamma compressed to get back to a conventional grayscale representation.

This is not the method used to obtain the luma in the YUV and related color models, used in standard color TV and video systems as PAL and NTSC, as well as in the Lab color model. These systems directly compute a gamma-compressed luma as a linear combination of gamma-compressed primary intensities, rather than use linearization via gamma expansion and compression to convert a gray

intensity value to RGB, simply set all the three primary color components red, green and blue to the gray value, correcting to a different gamma if necessary.

Single Channel Color Images

Color images are often built of several stacked color channels, each of them representing value levels of the given channel. For example, RGB images are composed of three independent channels for red, green and blue primary color components. CMYK images have four channels for cyan, magenta, yellow and black ink plates, etc.

Here is an example of color channel splitting of a full RGB color image. The column at left shows the isolated color channels in natural colors, while at right there are their grayscale equivalences:

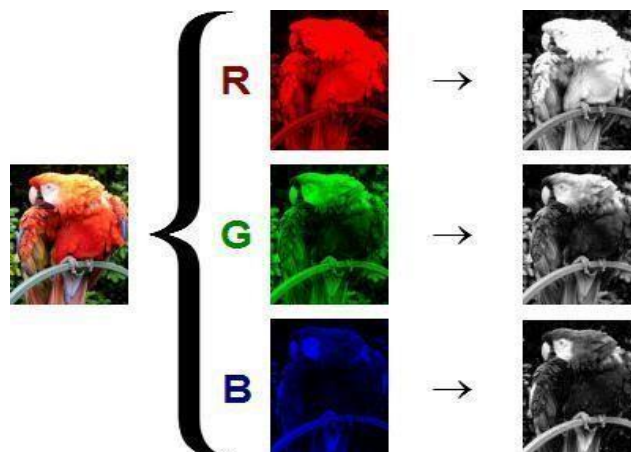


Fig 1.2 Composition of RGB from 3 Grayscale images

Histogram Equalization of a Color Image

Figure shows a color image with 768 x 512 pixels and 16 levels for each band. As a result of the color histogram equalization, Figure 3 shows more diversity in color than. Due to lack of space, the three-dimensional histogram is not depicted here. In fact, it is equalized uniformly. It took 412.6 seconds to equalize the histogram with nonzero values at 1501 colors out of 4096. Consecutive probabilities in the histogram would be further investigated. The results of the histogram equalization for two color images are displayed.

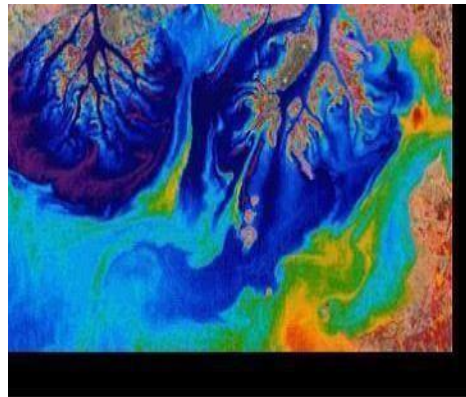


Fig 1.3. Original Image of CBV

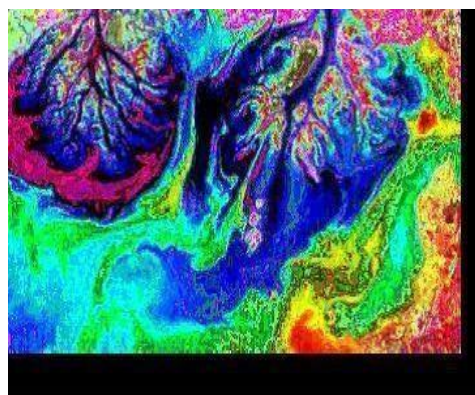


Fig 1.4. Color-equalized Image of CBV

DIGITAL IMAGE PROCESSING TECHNIQUES

The various Image Processing techniques are:

- Image representation
- Image preprocessing
- Image enhancement
- Image restoration
- Image analysis
- Image reconstruction
- Image data compression

Image Representation

An image defined in the "real world" is considered to be a function of two real variables, for example, $f(x,y)$ with f as the amplitude (e.g. brightness) of the image at the *real* coordinate position (x,y) . The effect of digitization is shown in Figure.

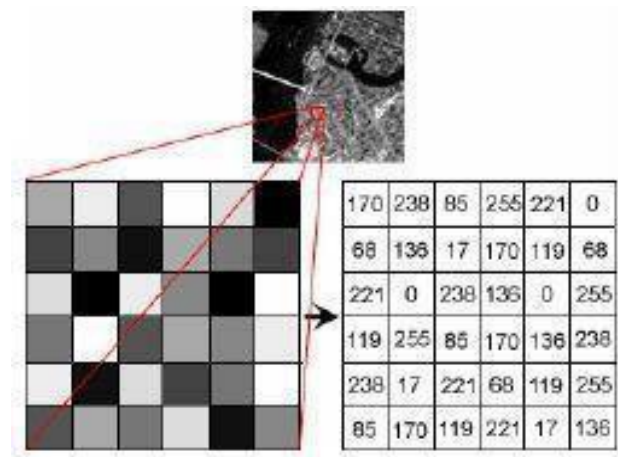


Fig.1.5 Image Representation

The 2D continuous image $f(x,y)$ is divided into N rows and M columns. The intersection of a row and a column is called as *pixel*. The value assigned to the integer coordinates $[m,n]$ with $\{m=0,1, 2,...,M-1\}$ and $\{n=0,1,2,...,N-1\}$ is $f[m,n]$. In fact, in most cases $f(x, y)$ which we might consider to be the physical signal that impinges on the face of a sensor. Typically an image file such as BMP, JPEG, TIFF etc., has some header and picture information. A header usually includes details like format identifier (typically first information), resolution, number of bits/pixel, compression type, etc.

Image Enhancement

Sometimes images obtained from satellites and conventional and digital cameras lack in contrast and brightness because of the limitations of imaging sub systems and illumination conditions while capturing image. Images may have different types of noise. In image enhancement, the goal is to accentuate certain image features for subsequent analysis or for image display. Examples include contrast and edge enhancement, pseudo-coloring, noise filtering, sharpening, and magnifying. Image enhancement is useful in feature extraction, image analysis and an image display. The enhancement process itself does not increase the inherent information content in the data. It simply emphasizes certain specified image characteristics. Enhancement algorithms are generally interactive and application-dependent.

Image Segmentation

Image segmentation is the process that subdivides an image into its constituent parts or objects. The level to which this subdivision is carried out depends on the problem being solved, i.e., the segmentation should stop when the objects of interest in an application have been isolated e.g., in autonomous air-to-ground target acquisition, suppose our interest lies in identifying vehicles on a road, the first step is to segment the road from the image and then to segment the contents of the road down to potential vehicles.

FUNDAMENTAL IN DIGITAL IMAGE PROCESSING

The various basic steps are as follows

- Image Acquisition
- Image preprocessing
- Image segmentation
- Image Representation and Description
- Image Recognition and Interpretation and knowledge base.

Image Acquisition

In the process, the first step in the process is image acquisition that is, to acquire a digital image. It highly requires an imaging sensor and the capability to digitize the signal produced by the sensor. The sensor could be a monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec.

The imaging sensor could also a line-scan camera that produces a single image line at a time. In this case, the object's motion past the line scanner produces two-dimensional image. If the output of the camera or other imaging sensor is not already in digital form, an analog-to-digital converter digitizes it. The nature of the sensor and image it produces are determined by the application.

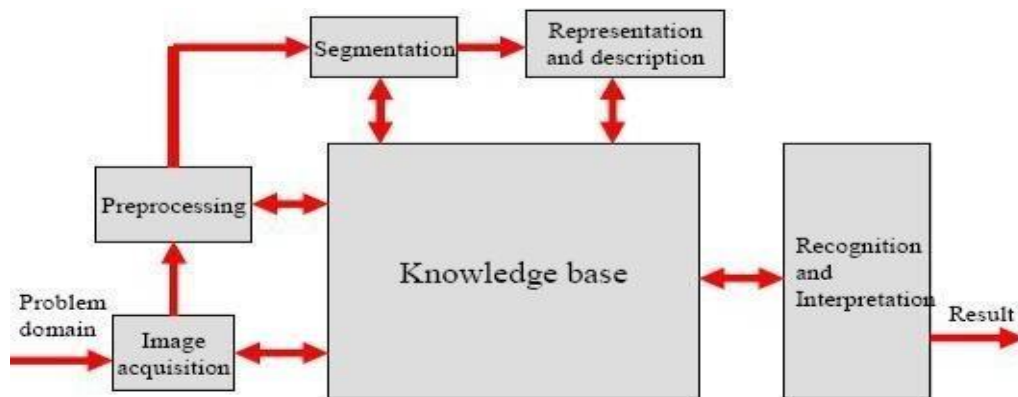


Figure 1.6 Fundamental steps in digital image processing

Image Preprocessing

The key function of preprocessing is to improve the image in ways that increase the chances for success of the other processes. Preprocessing typically deals with techniques for enhancing contrast, removing noise, and isolating regions whose texture indicate a likelihood of alphanumeric information.

- Image Enhancement.
- More suitable than original image for a specific application.
- Image Restoration.
- A process that attempts to reconstruct or recover an image that has been degraded using prior knowledge of the degradation concept.

Image Segmentation

The next stage deals with segmentation. Segmentation partitions an input image into its small constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. On the one hand, a rugged segmentation procedure brings the process a long way towards the successful solution of an imaging problem. On the other hand, weak or erratic segmentation algorithms almost guarantee eventual failure. In terms of character recognition, the key role of segmentation is to extract individual characters and words from the background.

Image representation and description

The output of the segmentation stage usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. In either case converting the data to a form suitable for computer processing is necessary.

The first decision that must be made, whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, however, these representations coexist. This situation occurs in character recognition applications, which often require algorithms based on boundary shape as well as skeletons and other internal properties.

Image recognition and interpretation

Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

For example, identifying a character as *c*, requires associating the descriptors for that character with the label *c*. Interpretation attempts to assign meaning to a set of labeled entities. For example, a string of five numbers are followed by a hyphen and four more numbers can be interpreted to be a ZIP code.

CHAPTER 2

LITERATURE SURVEY

Automatic and accurate segmentation of Left Ventricle (LV) and Right Ventricle (RV) in cine-MRI is required to analyze cardiac function and viability. We present a fully convolutional neural network to efficiently segment LV and RV as well as myocardium. The network is trained end-to-end from scratch. Average dice scores from five-fold cross-validation on the ACDC training dataset were 0.94, 0.89, and 0.88 for LV, RV, and myocardium. Experimental results show the robustness of the proposed architecture (Yeonggul Jang, Yoonmi Hong, et al,2018).

Cardiac magnetic resonance imaging improves on diagnosis of cardiovascular diseases by providing images at high spatiotemporal resolution. Manual evaluation of these time-series, however, is expensive and prone to biased and non-reproducible outcomes. In this paper, we present a method that addresses named limitations by integrating segmentation and disease classification into a fully automatic processing pipeline. We use an ensemble of U-Net inspired architectures for segmentation of cardiac structures such as the left and right ventricular cavity (LVC, RVC) and the left ventricular myocardium (LVM) on each time instance of the cardiac cycle. For the classification task, information is extracted from the segmented time-series in form of comprehensive features handcrafted to reflect diagnostic clinical procedures. Based on these features we train an ensemble of heavily regularized multilayer perceptrons (MLP) and a random forest classifier to predict the pathologic target class. We evaluated our method on the ACDC dataset (4 pathology groups, 1 healthy group) and achieve dice scores of 0.945 (LVC), 0.908 (RVC) and 0.905 (LVM) in a cross-validation over the training set (100 cases) and 0.950 (LVC), 0.923 (RVC) and 0.911 (LVM) on the test set (50 cases). We report a classification accuracy of 94% on a training set cross-validation and 92% on the test set. Our results underpin the potential of machine learning methods for accurate, fast and reproducible segmentation and computer-assisted diagnosis (CAD). (Fabian Isensee, Paul F. Jaeger, et al,2018). We have developed a method for automatic contour propagation in cine cardiac magnetic resonance images. The method consists of a new active contour model that tries to maintain a constant contour environment by matching gray values in profiles perpendicular to the contour.

We have developed a method for automatic contour propagation in cine cardiac magnetic resonance images. The method consists of a new active contour model that tries to maintain a constant contour environment by matching gray values in profiles perpendicular to the contour. Consequently, the contours should maintain a constant position with respect to neighboring anatomical structures, such that the resulting contours reflect the preferences of the user. This is particularly important in cine cardiac magnetic resonance images because local image features do not describe the desired contours near the papillary muscle. The accuracy of the propagation result is influenced by several parameters. Because the optimal setting of these parameters is application dependent, we describe how to use full factorial experiments to optimize the parameter setting. We have applied our method to cine cardiac magnetic resonance image sequences from the long axis two-chamber view, the long axis four-chamber view, and the short axis view. We performed our optimization procedure for each contour in each view. Next, we performed an extensive clinical validation of our method on 69 short axis data sets and 38 long axis data sets. In the optimal parameter setting, our propagation method proved to be fast, robust, and accurate. The resulting cardiac contours are positioned within the interobserver ranges of manual segmentation. Consequently, the resulting contours can be used to accurately determine physiological parameters such as stroke volume and ejection fraction. (Steven Lobregt, Marcel Breeuwer, et al, 2006).

Segmentation of cardiac magnetic resonance imaging is considered an important application in clinical practice. An automatic algorithm is proposed for segmentation of both endocardial and epicardial boundaries, in long-axis views. The data consisted of 126 patients, yielding 1008 traces. Estimated clinical parameters were highly correlated to gold standard measurements. The error between the automatic tracing and the gold standard was not significantly different than the error between two manual observers. In conclusions, a tool for segmenting the myocardial boundaries in the long-axis views is proposed, which works well, as demonstrated by the validation performed using a clinical dataset. (Yossi Tsadok, Yael Petrank, et al, 2013). In this paper, we present a framework to estimate local ventricular myocardium contractility using clinical MRI, a heart model and data assimilation. First, we build a generic anatomical model of the ventricles including muscle fiber orientations and anatomical subdivisions.

Then, this model is deformed to fit a clinical MRI, using a semi-automatic fuzzy segmentation, an affine registration method and a local deformable biomechanical model. An electromechanical model of the heart is then presented and simulated. Finally, a data assimilation procedure is described, and applied to this model. Data assimilation makes it possible to estimate local contractility from given displacements. Presented results on fitting to patient-specific anatomy and assimilation with simulated data are very promising. Current work on model calibration and estimation of patient parameters opens up possibilities to apply this framework in a clinical environment. (Sermesant, Moireau, et al,2006).

Background: Cardiac MRI derived biventricular mass and function parameters, such as end systolic volume (ESV), end-diastolic volume (EDV), ejection fraction (EF), stroke volume (SV), and ventricular mass (VM) are clinically well established. Image segmentation can be challenging and time-consuming, due to the complex anatomy of the human heart.

Objectives: This study introduces a deep learning approach allowing for fully-automated high quality segmentation of right (RV) and left ventricular (LV) endocardium and epicardium for extraction of cardiac function parameters.

Methods: A set consisting of 253 manually segmented cases has been used to train a deep neural network. Subsequently, the network has been evaluated on 4 different multicenter data sets with a total of over 1000 cases.

Results: For LV EF the intraclass correlation coefficient (ICC) is 98, 95, and 80 % (95 %), and for RV EF 96, and 87 % (80 %) on the respective data sets (human expert ICCs reported in parenthesis). The LV VM ICC is 95, and 94 % (84 %), and the RV VM ICC is 83, and 83 % (54 %). This study proposes a simple adjustment procedure, allowing for the adaptation to distinct segmentation philosophies. _-net exhibits state of-the-art performance in terms of dice coefficient.

Conclusions: Biventricular mass and function parameters can be determined reliably in high quality by applying a deep neural network for cardiac MRI segmentation, especially in the anatomically complex right ventricle. Adaption to individual

segmentation styles by applying a simple adjustment procedure is viable, allowing for the processing of novel data without time-consuming additional training. (Hinrich B Winther, Christian Hundt, et al,2017).

Convolution networks are powerful visual models that yield hierarchies of features. We show that convolution networks by themselves, trained end-to-end, pixels-to-pixels, improve on the previous best result in semantic segmentation. Our key insight is to build “fully convolution” networks that take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. We define and detail the space of fully convolution networks, explain their application to spatially dense prediction tasks, and draw connections to prior models. We adapt contemporary classification networks (Alex Net, the VGG net, and Google Net) into fully convolution networks and transfer their learned representations by fine-tuning to the segmentation task. We then define a skip architecture that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentations. Our fully convolution network achieves improved segmentation of PASCAL VOC (30% relative improvement to 67.2% mean IU on 2012), NYUDv2, SIFT Flow, and PASCAL-Context, while inference takes one tenth of a second for a typical image.(Evan Shelhamer, Jonathan Long, et al,2016).

Non-communicable diseases (NCDs) lead to substantial mortality and morbidity worldwide. The most common NCDs are cardiovascular diseases (CVD), diabetes, cancer and chronic respiratory diseases. With the rapid increase in NCD-related deaths in Asia Pacific countries, NCDs are now the major cause of deaths and disease burden in the region. NCDs hamper achievement of the Millennium Development Goals (MDG). People in the low socio-economic group are most affected by NCDs as they have poor access to policies, legislations, regulations and healthcare services meant to combat NCDs. This results in loss of productivity by a decreasing labor force with implications at the macroeconomic level. The 3 major NCDs in the Asia Pacific region are CVDs, cancer and diabetes due to the increasing loss of disability adjusted life years (DALYs). The 4 major behavioral risk factors for NCDs are: tobacco use, alcohol consumption, inadequate physical activity and unhealthy diet.

The underlying risk factors are urbanization, globalization, sedentary lifestyle, obesity and hypertension. Strategies to combat NCDs in the Asia Pacific region are as follows: population-based dietary salt reduction, health education, psychological interventions, i.e., cognitive behavioral therapy and motivational-interviewing, taxation and bans on tobacco-related advertisements, implementing smoke-free zones and surveillance by the World Health Organization. Control measures must focus on prevention and strengthening inter-sectorial collaboration. (Wah-Yun Low, Yew-Kong Lee, et al,2014).

There is large consent that successful training of deep net-works requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. We show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window convolution network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Using the same network trained on transmitted light microscopy images (phase contrast and DIC) we won the ISBI cell tracking challenge 2015 in these categories by a large margin. Moreover, the network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU. (Olaf Ronneberger, Philipp Fischer, and Thomas Brox, et al,2015).

Convolution Neural Networks (CNNs) have been recently employed to solve problems from both the computer vision and medical image analysis fields. Despite their popularity, most approaches are only able to process 2D images while most medical data used in clinical practice consists of 3D volumes. In this work we propose an approach to 3D image segmentation based on a volumetric, fully convolution, neural network. Our CNN is trained end-to-end on MRI volumes depicting prostate and learns to predict segmentation for the whole volume at once. We introduce a novel objective function that we optimize during training, based on Dice coefficient. In this way we can deal with situations where there is a strong imbalance between the number of foreground and background voxels. To cope with the limited number of

annotated volumes available for training, we augment the data applying random non-linear transformations and histogram matching. We show in our experimental evaluation that our approach achieves good performances on challenging test data while requiring only a fraction of the processing time needed by other previous methods. (Fausto Milletari, Nassir Navab, et al,2015).

CHAPTER 3

AIM AND SCOPE

AIM

The main aim of this project is to segment CBV (cardiac bi-ventricle) based on deep neural network.

SCOPE

Gray scale images have standard variation in intensity which provides greater contrast images. Makes it easier during segmentation. The use of grayscale images for CBV yields greater resolution and pixel information which provides wider scope for segmentation and classification.

Existing System

The resolution of the feature maps and spatial information after three max-pooling layers, need to be solved before our model training, so we add two corresponding up sampling layers to restore image resolution. However, the lost spatial information is difficult retrieved. To compensate for the loss of resolution caused by the pooling layer, the Cardiac-DeepLED introduces skip connections between its encoder and decoder. Skipping connections helps the decoder path to recover fine-grained information from the encoder path. The higher resolution information is passed by means of standard skip connections between the encoder and the decoder.

Proposed System

In this proposed system the input images are color images means then we convert to gray scale from the color images. The image features like color, weight, and depth and pixel information to apply before the classifier (neural network). Here we used the segmentation algorithm is used in order to segment the portion of defected areas. The neural network concept is used for training the image and testing the image with the help of weight estimating classifier.

Module Description

Module 1: Preprocessing

If the input images are color images means then we convert to gray scale from the color images. The median filter is applied in order to remove the noise from the image.

Module 2: Segmentation

The image features like color, weight, and depth and pixel information to apply before the classifier (neural network). Here we used the segmentation algorithm is used in order to segment the portion of defected areas.

Module 3: Classification

This module is used to establish the neural network concept is used for training the image and testing the image with the help of weight estimating classifier.

Block diagram

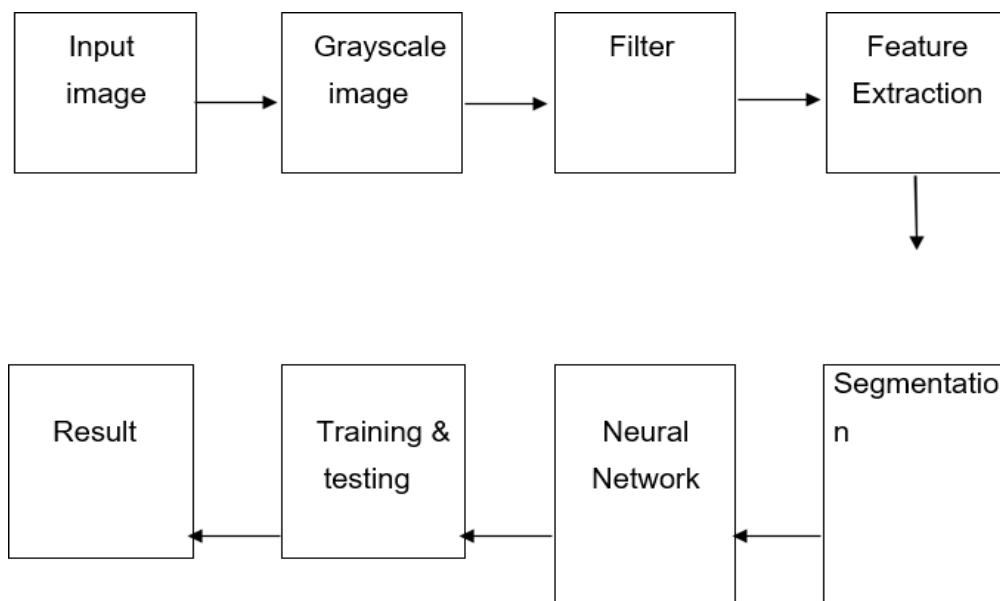


Fig.3.1. Segmentation and classification

Segmentation:

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in images. More precisely image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region is similar with respect to some characteristic or computed property, such as color, intensity, or texture. There are two major approaches in image segmentation: edge and region based. Edge based segmentation partitions an image based on discontinuities with sub-regions, while region-based segmentation does the similar function based on the uniformity of a desired property within a sub-region.

Edge-based Segmentation

Edge-based segmentation looks for discontinuities in the intensity of an image. It is more likely edge detection or boundary detection rather than the exact meaning of image segmentation. An edge can be defined as the border between two regions with relatively separate properties. The assumption of edge-based segmentation is that every sub-region in an image is sufficiently uniform so that the transition between two sub-regions can be determined on the basis of discontinuities alone. When this statement is not valid, region-based segmentation, discussed in the next section, regularly provides more reasonable segmentation outcome. Basically, the idea underlying most edge-detection techniques is the computation of a local derivative operator.

Edge detection by gradient operations usually works well only in the images with sharp intensity transitions and relatively low noise. Due to its sensitivity to noise, various smoothing operation is usually essential as preprocessing, and the smoothing effect consequently blurs the edge information. However, the computational cost is comparatively lower than other segmentation methods because the computation can be complete by a local filtering operation, i.e. convolution of an image with a kernel.

Region-based Segmentation:

Region-based segmentation looks for equality inside a sub-region, based on a desired property, e.g. intensity, color, and texture. Clustering techniques encountered in pattern classification literature have related objectives and can be applied for image segmentation. Region rising is a technique that merges pixels or small sub-regions into a bigger sub region. The simplest implementation of this approach is pixel aggregation which starts with a set of seed points and grows regions from these seeds by appending nearby pixels if they satisfy the given criteria. Additional criteria that use properties to raise the regions lead area growing into more sophisticated methods, e.g. region competition. Region competition merges neighboring sub-regions under criteria involving the equality of regions or sharpness of boundaries. Strong criteria tend to generate over segmented results, while weak criteria lean to produce poor segmentation outcome by over-merging the sub regions with blurry boundaries. An alternative of region rising is split-and-merge which partitions an image firstly into a set of arbitrary, disjointed sub-regions, and then combine and/or split the sub-regions in an attempt to satisfy the segmentation criteria.

Active Contours

The method of active contours has become quite popular for a range of applications, mainly image segmentation and motion tracking, through the last decade. This methodology is based upon the use of deformable contours which match to various object shapes and motions. This section provides a theoretical setting of active contours and an indication of existing active contour methods. There are two main approaches in active contours based on the mathematic implementation: snakes and level sets. Snakes explicitly shift predefined snake points based on an energy minimization method, while level set approaches move contours completely as a particular level of a function.

As image segmentation methods, there are two kinds of active contour models according to the force evolving the contours: edge- and region-based. Edge based active contours apply an edge detector, typically based on the image gradient, to locate the boundaries of sub-regions and to draw the contours to the detected boundaries. Region based active contours apply the statistical information of image intensity inside each subset instead of searching geometrical boundaries. Region-based approaches are also closely connected to the region-based segmentation.

Edge-based Active Contours

Edge-based active contours are strongly connected to the edge-based segmentation. Most edge based active contour models consist of two parts: the regularity part, which determines the form of contours, and the edge recognition part, which attracts the contour towards the boundaries. Edge-based active contour models have a little disadvantage compared to the region-based active contour models, discussed in the next section. Because of the constant term, edge based active contour models evolve the contour towards only one way, each inside or outside. Therefore, an primary contour must be placed completely inside or outside of ROI, and some level of a previous knowledge is still necessary. Also, edge-based active contours inherit a few disadvantages of the edge-based segmentation methods due to the parallel method used. Since both edge-based segmentation and edge-based active contours rely on the image gradient process, edge-based active contours may omit the blurry boundaries, and they are sensitive to local minima or noise as edge-based segmentation does. Gradient vector flow quick geodesic dynamic contours proposed by Paragios replaced the border detection (boundary attraction) word with gradient vector field that refers to a spatial diffusion of the boundary information and guides the propagation to the object boundaries from equally sides, to give extra freedom from the restriction of first contour position.

Region-based Active Contours

Most region-based active contour models consist of two parts: the regularity part, which determines the smooth form of contours, and the energy minimization part, which searches for equality of a preferred feature within a subset. A good characteristic of region-based active contours is that the first contours can be situated anyplace in the image as region-based segmentation relies on the global energy minimization rather than local energy minimization. Therefore, less previous knowledge is required than edge-based active contours. Although usual region-based active contours partition an image into several sub regions, those several regions belong to only two subsets: both the inside or the outside of contours. Chan and Vese proposed multi-phase active contour model which increases the number of subsets that active contours can locate simultaneously.

Multiple active contours evolve independently based on the piecewise-constant model or the piecewise-smooth model, and multiple subsets are defined by a set of disjoint combination of the level set functions. Due to the global energy minimization; region based active contours usually do not have any restriction on the placement of first contours. That is, region-based active contour can detect interior boundaries regardless of the position of initial contour. That is, region-based active contour can detect inner boundaries regardless of the position of initial contours. The use of pre-defined initial contours provides a method of independent segmentation. Also, they are less responsive to local minima or noise than edge-based active contours. However, due to the supposition of uniform image intensity, most methods are relevant only to images where each subset is stand for able by a simple expression, e.g. single Gaussian distribution or a constant. If a subset, i.e. class, consists of multiple distinguishing sub-classes, these methods would produce over-segmented or under-segmented results. We propose novel region-based active contour models which produce better results using multivariate mixture density functions.

Neural Network

A Neural Network (NN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. NNs, like people, learn by example. An NN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons.

Historical background

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras. Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm the

field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few researchers. These pioneers were able to develop convincing technology which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

The Basics of Neural Networks

Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted connections. The hidden layers then link to an 'output layer' where the answer is output as shown in the graphic below.

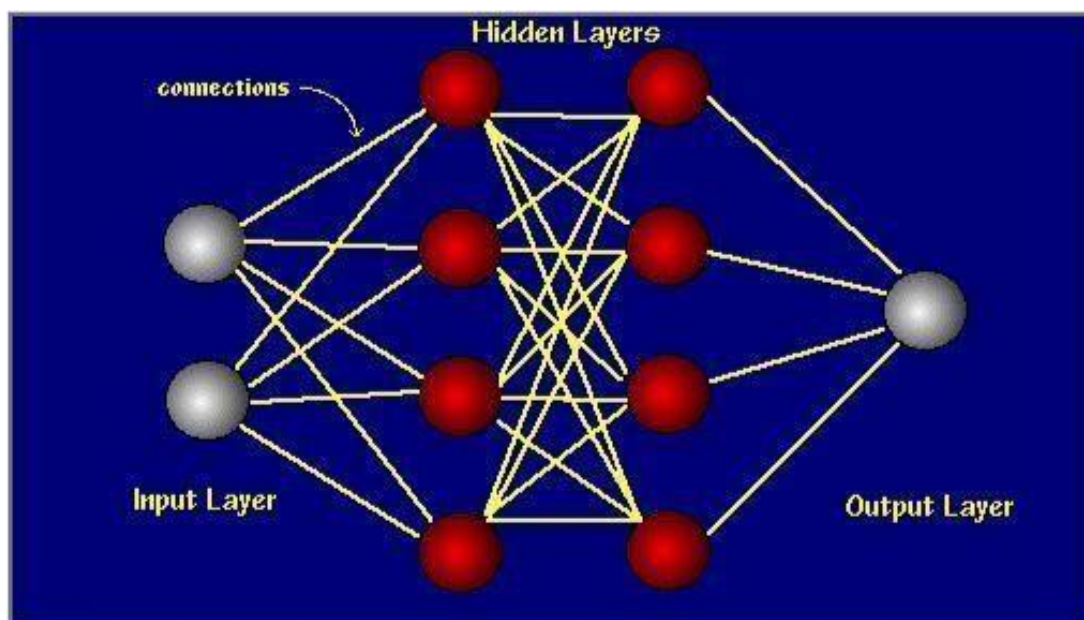


Fig.3.2. Neural network

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- Self-Organization: An NN can create its own organization or representation of the information it receives during learning time.
- Real Time Operation: NN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Neural networks versus conventional computers

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Solve the problem. That restricts the problem-solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more unless the specific steps that the computer needs to follow are known the computer cannot useful if they could do things that we don't exactly know how to do.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to be solved must be known and stated in small unambiguous instructions. These instructions are then converted to a high-level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

Network layers

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units. The activity of the input units represents the raw information that is fed into the network.

The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units. The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units. This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents. We also distinguish single-layer and multi-layer architectures. The single-layer organization, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multi-layer organizations. In multi-layer networks, units are often numbered by layer, instead of following a global numbering.

CHAPTER 4

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

- Hard Disk : 40GB and Above
- RAM : 512MB and Above
- Processor : Pentium III and Above

SOFTWARE REQUIREMENTS

- MATLAB R2018a
- Windows Operating System XP and Above

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the Mu PAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

The basic data structure in MATLAB is the array, an ordered set of real or complex elements. This object is naturally suited to the representation of images, real-valued ordered sets of color or intensity data.

MATLAB stores most images as two-dimensional arrays (i.e., matrices), in which each element of the matrix corresponds to a single pixel in the displayed image. For example, an image composed of 200 rows and 300 columns of different coloured dots would be stored in MATLAB as a 200-by-300 matrix. Some images, such as RGB, require a three-dimensional array, where the first plane in the third dimension represents the red pixel intensities, the second plane represents the green pixel intensities, and the third plane represents the blue pixel intensities.

This convention makes working with images in MATLAB similar to working with any other type of matrix data, and makes the full power of MATLAB available for image processing applications. For example, you can select a single pixel from an image matrix using normal matrix subscripting.

INTRODUCTION TO MATLAB R2018a:

The MATLAB system consists of these main parts:

Desktop Tools and Development Environment

This is the set of tools and facilities that help you use and become more productive with MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, a code analyzer and other reports, and browsers for viewing help, the workspace, files, and the search path.

Mathematical Function Library

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

The Language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

Graphics

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

External Interfaces

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), for calling MATLAB as a computational engine, and for reading and writing MAT-files.

Image Processing Toolbox

The Image Processing Toolbox is a collection of functions that extend the capability of the MATLAB numeric computing environment. The toolbox supports a wide range of image processing operations, including:

- Spatial image transformations
- Morphological operations
- Neighborhood and block operations
- Linear filtering and filter design
- Transforms
- Image analysis and enhancement
- Image registration
- De blurring
- Region of interest operations

Many of the toolbox functions are MATLAB M-files, a series of MATLAB statements that implement specialized image processing algorithms. You can extend the capabilities of the Image Processing Toolbox by writing your own M-files, or by using the toolbox in combination with other toolboxes, such as the Signal Processing Toolbox and the Wavelet Toolbox.

Computer Vision Toolbox

The field is of importance for such various applications as autonomous vehicles, navigating with the help of images, captured by a mounted camera, and high precision measurements using images, taken by calibrated cameras. In this paper, we will present a number of numerical routines, implemented in MATLAB, that are useful in a variety of computer vision applications. The collection of routines will be called the Computer Vision Toolbox. One of the main problems in Computer Vision is to calculate the 3D-structure of the scene and the motion of the camera from measurements in the images taken from different view-points in the scene.

This problem is called structure and motion, referring to the fact that both the structure of the scene and the motion of the camera are calculated from image measurements only. A number of different sub problems, arising from different knowledge of the intrinsic properties of the camera appear. Other important problems are to calculate the structure of the 3D-scene given the motion of the camera and to calculate the motion of the camera given the structure of the scene. These problems are somewhat simpler than the general structure and motion problem, but are nevertheless important for navigation and obstacle avoidance. A related problem is to calibrate a camera, i.e. calculate the focal distance, the principal point etc., from images of a known scene. This calibration may be used to make a more precise reconstruction. The problem of estimating structure and motion from image sequences are closely related to the field of closed range photogrammetric. In this field images are used to make precise measurements of three-dimensional objects from a number of images, often taken by calibrated cameras. The main difference is that in computer vision, the focus is not on accuracy, but instead on reliability and speed of calculation. In all these problems different kind of so-called features can be used. The simplest feature is points that are easily detected in the sequence, such as corners or painted marks. Other types of features are lines and curves, which are easier to detect and track but more difficult to use in structure and motion algorithms. In order to use our computer vision routines these features must be detected in advance and the correspondence between different images must be established. The main emphasis of our computer vision toolbox is to use these detected features to solve for structure and motion.

Image Acquisition Toolbox

The Image Acquisition Toolbox's is a collection of functions that extend the capability of the MATLAB numeric computing environment. The toolbox supports a wide range of image acquisition operations including

- Acquiring images through many types of image acquisition devices, from professional grade frame grabbers to USB-based Webcams
- Viewing a preview of the live video stream
- Triggering acquisitions (includes external hardware triggers)
- Configuring call back functions that execute when certain events occur
- Bringing the image data into the MATLAB workspace

Many of the toolbox functions are MATLAB M-files. You can extend the capabilities of the Image Acquisition Toolbox by writing your own M-files, or by using the toolbox in combination with other toolboxes, such as the Image Processing Toolbox and the Data Acquisition Toolbox. The toolbox also includes a Simulink interface called the Image Acquisition Block set. This block set extends Simulink with a block that lets you bring live video data into a model.

CHAPTER 5

RESULTS AND DISCUSSION

RESULTS

- We proposed an automatic segmentation network based on deep learning to segment CBV at the pixel-level.
- Hence, we have classified several heart diseases such as Coronary artery disease, Arrhythmia, Congenital heart disease, Myocardial infarction.

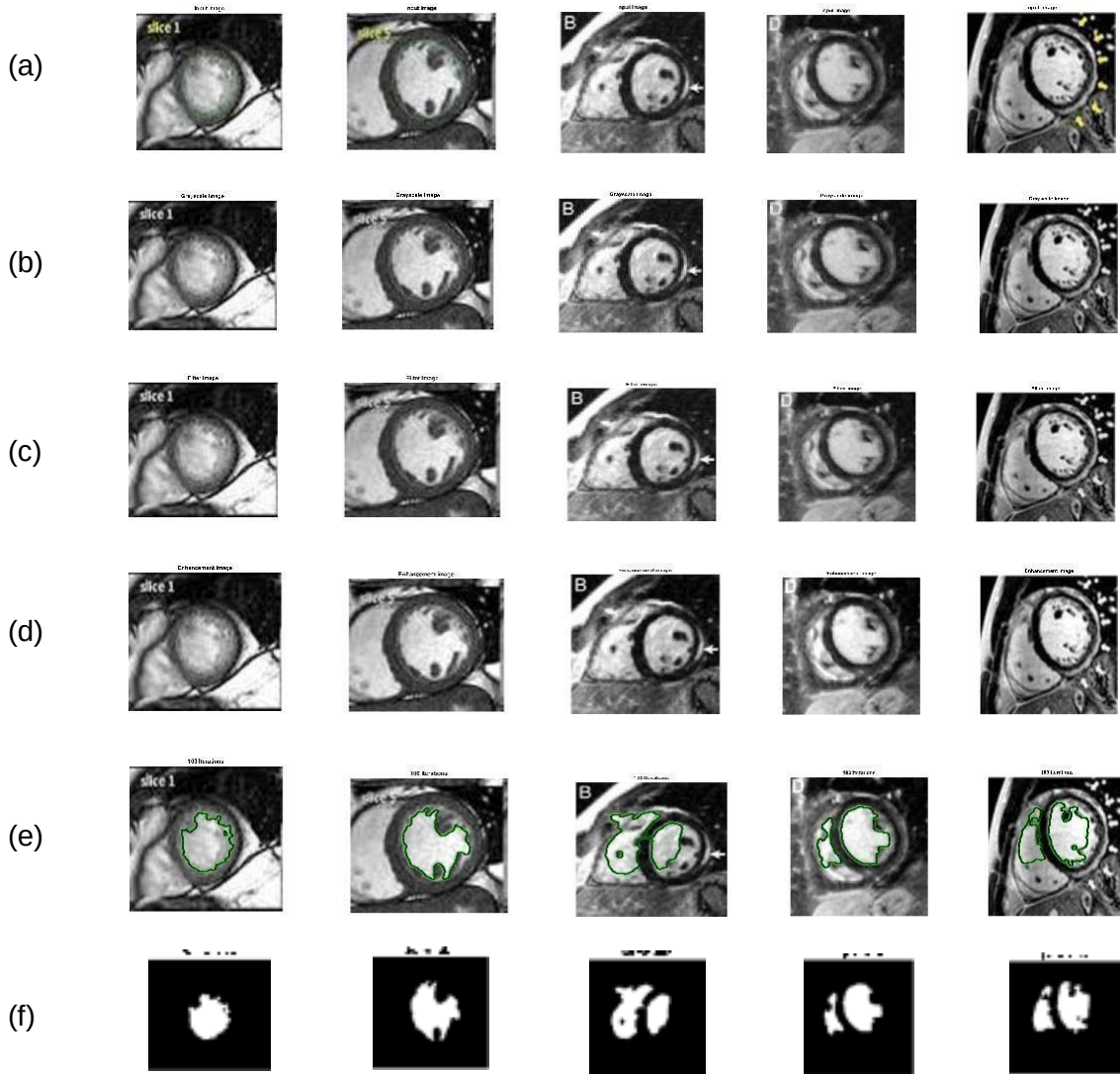


Fig.5.1. Preprocessing and segmentation. (a) Input image. (b) Grayscale conversion. (c) Filtered Image. (d) Image enhancement. (e) Region of Interest. (f) Segmented region

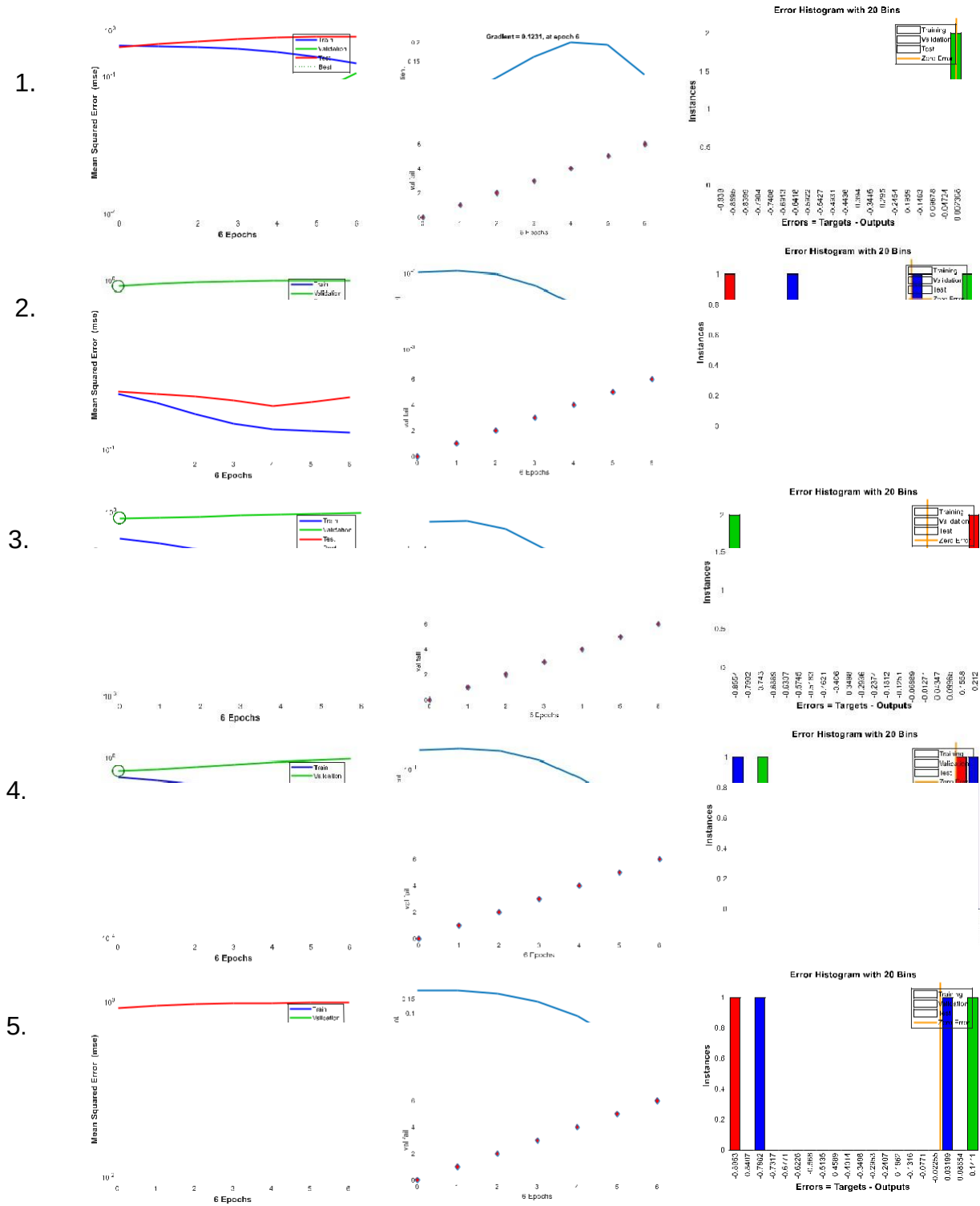


Fig.5.2. Performance plot, Training state plot and Error Histogram measurements of the tested images

1	Area	Centroid	Accuracy
2	3199	100.4711, 91.8275	87.0971
3	4319	104.6555, 84.1489	87.0608
4	3483	69.0818, 88.0818	87.0371
5	1144	53.6949, 100.1766	87.0443
6	1711	61.3647, 92.5272	87.0411
7	896	52.8817, 78.1685	87.0345
8	268	58.5299, 63.0299	76.1565
9	438	63.2717, 59.5639	87.0529
10	5319	102.4407, 97.4270	87.0337
11	5872	104.5324, 91.1286	87.0332

Fig.5.3. Feature Extraction from the input images showing area, centroid and accuracy values

- Feature extraction a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector.
- This approach is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as image matching and retrieval.
- In this case, required features such as area, centroid and accuracy levels are extracted from the MR images which is useful for classification purposes.

Performance Plot

- This plot basically shows error vs. epoch for the training, validation, and test performances of the training record TR returned by the function train.
- Performance plot shows you mean square error dynamics for all your datasets in logarithmic scale. Training MSE is always decreasing, so its validation and test MSE you should be interested in. Your plot shows a perfect training.
- Generally, the error reduces after more epochs of training, but might start to increase on the validation data set as the network starts overfitting the training data. In the setup, the training stops after six consecutive increases in validation error, and the best performance is taken from the epoch with the lowest validation error.

Training State Plot

- This plots the training state from a training record `tr` returned by `train`.
- Gradient is a value of back propagation on each iteration in logarithmic scale.
- The gradient value less than or close to zero means that you reached the bottom of the local minimum of your goal function.
- Validation fails are iterations when validation MSE increased its value. A lot of fails means overtraining. training automatically stops after 6 fails in a row.

Error Histogram Plot

- Error histogram is the histogram of the errors between target values and predicted values after training a feedforward neural network. As these error values indicates how predicted values are differing from the target values, hence these can be negative.
- Bins are the number of vertical bars you are observing on the graph. The total error range is divided into smaller bins.
- Y-axis represents the number of samples from your dataset, which lies in a particular bin. Zero error line corresponds to the zero-error value on the error axis (i.e. X-axis).

DISCUSSION

- The input images are converted to Gray scale images for efficient segmentation and classification.
- Image features like area, centroid, depth and pixel information obtained before the classifier.
- This segmentation and classification process are performed using active contour segmentation.
- Gray scale images have standard variation in intensity which provides greater contrast images. Easier during segmentation.
- Region Based segmentation algorithm is used to segment the required portion, which helps in acquiring for pixel information and better contrast.
- Back propagation network is used for training, testing and validation processes using BPN classifier.

CHAPTER 6

SUMMARY AND CONCLUSION

SUMMARY

- MATLAB 2018a is installed to train the MRI datasets and images for analysis.
- The pre-processing of input images for segmentation process.
- Training and testing of sample images which provides the accuracy, performance, gradient graph and error histogram.
- Training and testing of CBV MRI images having various heart diseases.

CONCLUSION

- The majority of cardiac MR images show that the similar intensity distribution in different regions, thus providing a little edge information.
- The resolution of the feature maps and spatial information after three max-pooling layers need to be solved before our model training, so we add two corresponding up sampling layers to restore image resolution.
- A collaborative formulation of segmentation is discussed by jointly integrating region and boundary information.
- Here we used the active contour segmentation algorithm is used in order to segment the portion of defected areas.
- We used to establish the back propagation neural network concept for training the image and testing the image with the help of weight estimating classifier.
- The result image will be compared with the dataset images and it will display types of disease name.

REFERENCES

1. Evan Shelhamer, Jonathan Long, Trevor Darrell, "Fully Convolutional Networks for Semantic Segmentation", May 2016.
2. Fabian Isensee, Paul F. Jaeger, Peter M. Full, Ivo Wolf, Sandy Engelhardt, Klaus H. Maier-Hein, "Automatic Cardiac Disease Assessment on cine-MRI via Time-Series Segmentation and Domain Specific Features", January 2018.
3. Fausto Milletari, Nassir Navab, Seyed-Ahmad Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation", October 2015.
4. Gilon Hautvast, Steven Lobregt, Marcel Breeuwer, Frans Gerritsen, "Automatic Contour Propagation in Cine Cardiac Magnetic Resonance Images", IEEE Transactions on Medical Imaging, vol. 25, no. 11, November 2006.
5. Hinrich B. Winther, Christian Hundt, Bertil Schmidt, Christoph Czernera, Johann Bauersachs, Frank Wacker, Jens Vogel-Claussen, "Deep Learning for Generalized Biventricular Cardiac Mass and Function Parameters", June 2017.
6. Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", May 2015.
7. Sermesant M, Moireau P, Camara O, Sainte-Marie J, Andriantsimiavona R, Cimrman R, Hill D.L.G, Chapelle D, Razavi R, "Cardiac function estimation from MRI using a heart model and data assimilation: Advances and difficulties", September, 2006.
8. Wah-Yun Low, Yew-Kong Lee, Alexander Lourdes Samy, "Non-communicable diseases in the Asia-Pacific region: Prevalence, risk factors and community-based prevention", December 2014.
9. Yeonggul Jang, Yoonmi Hong, Seongmin Ha, Sekeun Kim, and Hyuk-Jae Chang, "Automatic Segmentation of LV and RV in Cardiac MRI", January 2018.
10. Yossi Tsadoka, Yael Petranka, Sebastian Sarvari, Thor Edvardsen, Dan Adam, "Automatic segmentation of cardiac MRI cines validated for long axis views", September 2013.

APPENDICES

A. SOURCE CODE

```
clc;
clear all;
close all;

%% Preprocessing

[fn fp]=uigetfile({'*.jpg;*.jpeg;*.png','All video Files';...
    '*. *','All Files' },'mytitle',...
    'C:\Work\Twin_A');

InputImage=[fp fn];
I=imread(InputImage);
figure,imshow(I);
title('Input image');

%% Grayscale image

I1=rgb2gray(I);
figure,imshow(I1);
title('Grayscale image');

%% Filter image

I3=medfilt2(I1);
figure,imshow(I3);
title('Filter image');

%% Enhancement image

I4=imadjust(I3);
figure,imshow(I4);
```

```

title('Enhancement image');

%% segmentation

m = zeros(size(I4,1),size(I4,2));
m(111:222,123:234) = 1;

I = imresize(I4,.5);
m = imresize(m,.5);

% figure,imshow(I),title('Input Image');

% figure,imshow(m),title('Initialization');

figure,title('Segmentation'); seg =

region_seg(I, m, 100);

figure,imshow(seg),title('Segmentation');

stats=
regionprops(seg,'area','Centroid','Perimeter','MajorAxisLength','MinorAxisLength','
FilledArea','FilledImage');

CC = bwconncomp(seg);

area=stats.Area;

Centroid=stats.Centroid;

Perimeter=stats.Perimeter;

MajorAxisLength=stats.MajorAxisLength;

```

```

MinorAxisLength=stats.MinorAxisLength;

%% back propogation

hiddenLayerSize = 4;
net = patternnet(hiddenLayerSize);

net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};

net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

net.trainFcn = 'traincgb'; % conjugate gradient backPropagation

net.performFcn = 'mse'; % Mean squared error

net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression', 'plotfit'};

inputs=[1 2 3 4];
targets=[0 1 1 0];

[net,tr] = trainrp(net,inputs,targets);
outputs = net(inputs);
ktry=net(targets); % Just trying gsubtract

errors = gsubtract(ktry,outputs);
targets=ktry;
performance = perform(net,targets,outputs)

```

```

trainTargets = targets .* tr.trainMask{1}; valTargets =
targets .* tr.valMask{1}; testTargets = targets .*
tr.testMask{1}; trainPerformance =
perform(net,trainTargets,outputs); valPerformance =
perform(net,valTargets,outputs); testPerformance =
perform(net,testTargets,outputs);

```

```

view(net)

```

```

if area<500
    msgbox('Congenital heart disease');
elseif(area >500)& (area<2500)
    msgbox('Arrhythmia');
elseif (area >2500)& (area<4500)
    msgbox('Coronary artery disease');
elseif (area >4500)
    msgbox('Myocardial infarction')
end

```

```

value1=seg;

```

```

numOfClasses = size(value1,1);
totalSamples = sum(sum(value1));

```

```

[TP,TN,FP,FN,accuracy,sensitivity,specificity,precision,f_score]=
deal(zeros(numOfClasses,1));

```

```

FN1=length(seg);TN1=length(seg);
TP=length(seg);
[m,n]=size(seg);
FP=abs(TP-(m-6*n-6));
FN=abs(FN1-TP);
TN=abs(TN1-FP);

```

```

for class = 1:numOfClasses
accuracy = TP + TN / totalSamples;
sensitivity = TP / (TP + FN);
specificity = TN / (FP + TN);
precision = TP / (TP + FP);
f_score = 2*TP/(2*TP + FP + FN);
score=5;
end

field2 = "accuracy"; value2 = accuracy;
field3 = "sensitivity"; value3 = sensitivity;
field4 = "specificity"; value4 = specificity;
field5 = "precision"; value5 = precision;
field6 = "recall"; value6 = sensitivity;
field7 = "Fscore"; value7 = f_score;
disp("Accuracy");
Accuracy=mean(value2/2.3);
disp(Accuracy);
% Specificity=mean(value4*100);
% disp("Specificity");
% disp(Specificity);
% Sensitivity=mean(value2/score);
% disp("Sensitivity");
% disp(Sensitivity);

```

B. SCREENSHOTS

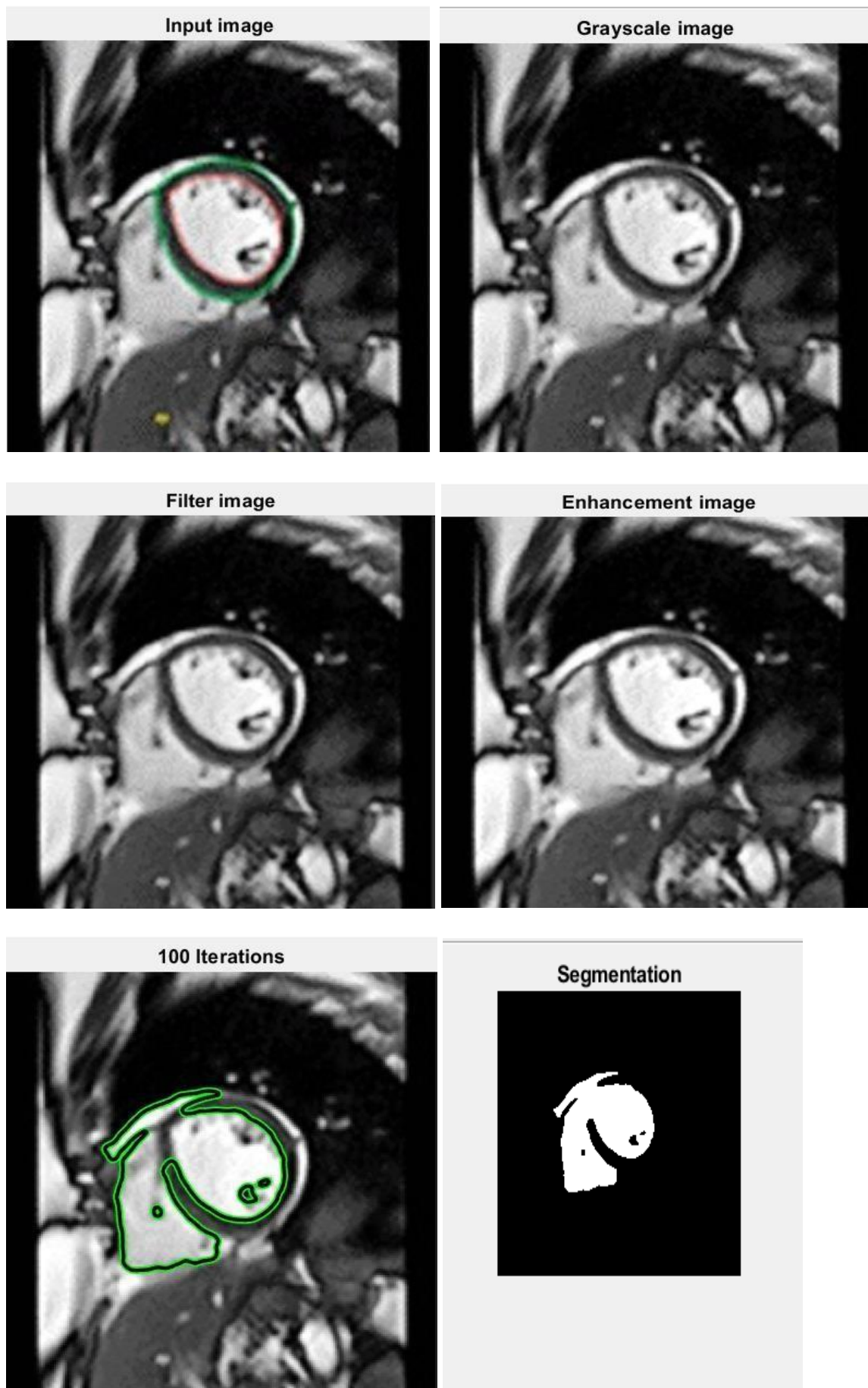


Fig B.1 Initial Process

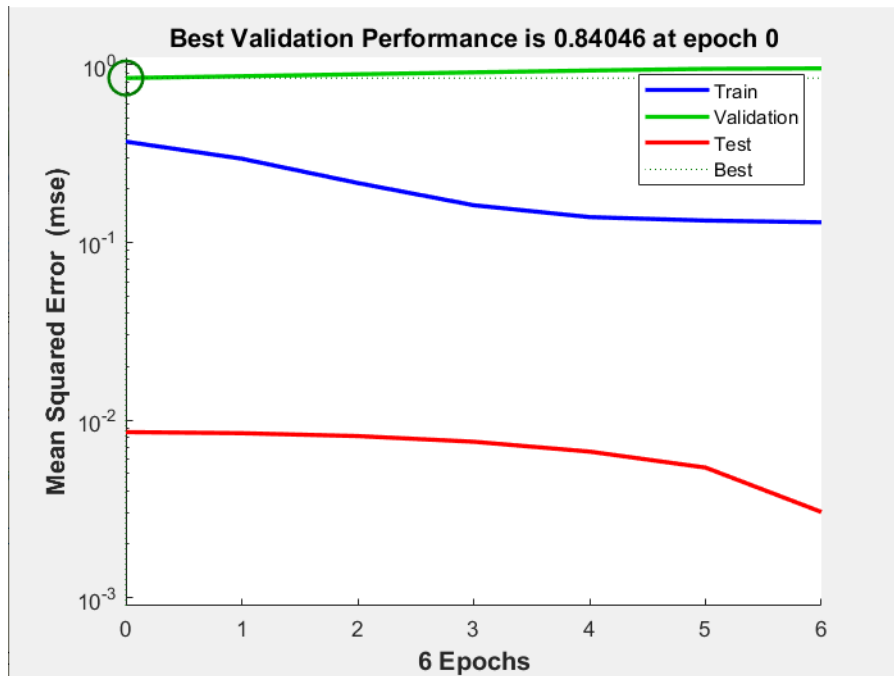


Fig B.2 Performance graph

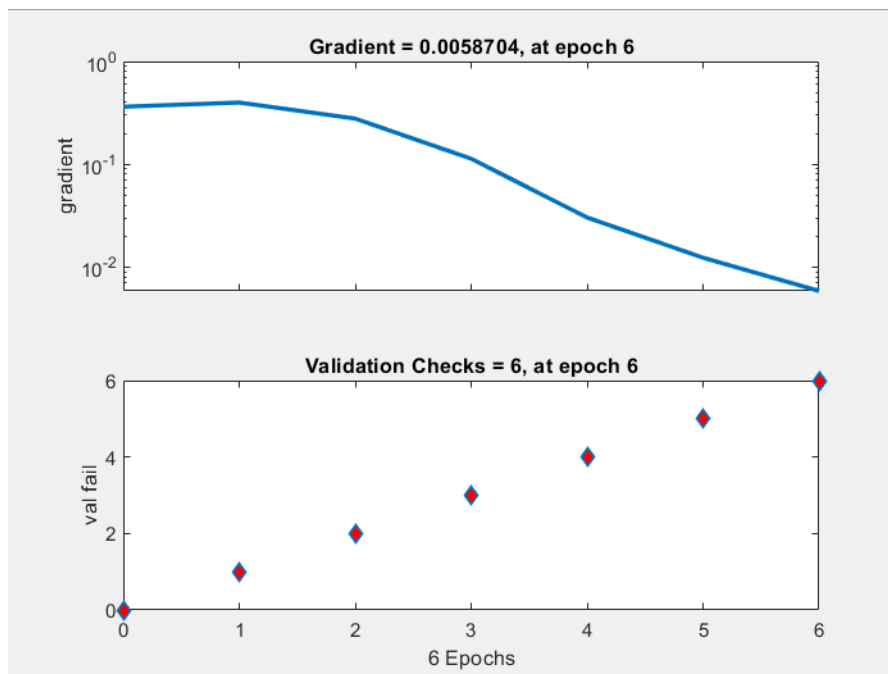


Fig B.3 Training state

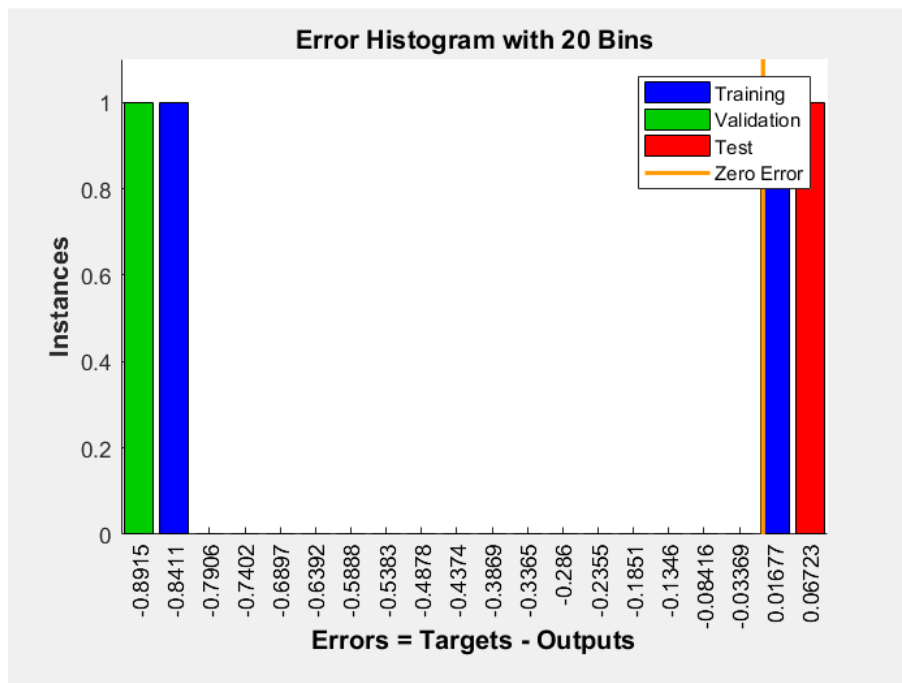


Fig B.4 Error Histogram

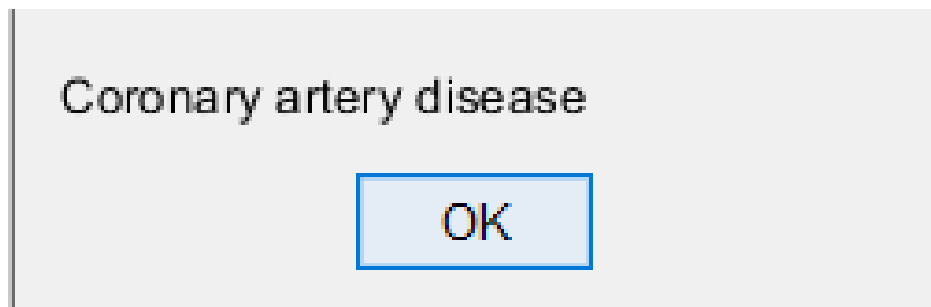


Fig B.5 Result