AUTOMATIC SEGMENTATION OF CRISTAE MEMBRANES IN 3D
ELECTRON MICROSCOPY TOMOGRAPHY IMAGES USING ARTIFICIAL
NEURAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MERİH ALPHAN KARADENİZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE
IN
MEDICAL INFORMATICS

SEPTEMBER 2016

AUTOMATIC SEGMENTATION OF CRISTAE MEMBRANES IN 3D
ELECTRON MICROSCOPY TOMOGRAPHY IMAGES USING ARTIFICIAL
NEURAL NETWORKS

Submitted by MERİH ALPHAN KARADENİZ in partial fulfillment of the
requirements for the degree of **Master of Science in The Department of Medical
Informatics Middle East Technical University** by,

Prof. Dr. Nazife Baykal
Director, **Graduate School of Informatics**   _____

Assist. Prof. Dr. Yeşim Aydın Son
Head of Department, **Medical
Informatics**   _____

Prof. Dr. Ünal Erkan Mumcuoğlu
Supervisor, **Medical Informatics**   _____

Examining Committee Members:

Assoc. Prof. Dr. Tolga Esat Özkurt
Medical Informatics, Middle East
Technical University   _____

Prof. Dr. Ünal Erkan Mumcuoğlu
Medical Informatics, Middle East
Technical University   _____

Assist. Prof. Dr. Aybar Can Acar
Medical Informatics, Middle East
Technical University   _____

Assoc. Prof. Dr. Uğur Murat Leloğlu
Geodetic and Geographic Information
Systems, Middle East Technical
University   _____

Assoc. Prof. Dr. Hadi Hakan MARAŞ
Computer Engineering, Çankaya University   _____

Date: 01.09.2016

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 wok.

Name, Last name :   Merih Alphan Karadeniz

Signature   :   _____

ABSTRACT

AUTOMATIC SEGMENTATION OF CRISTAE MEMBRANES IN 3D
ELECTRON MICROSCOPY TOMOGRAPHY IMAGES USING ARTIFICIAL
NEURAL NETWORKS

Karadeniz, Merih Alphan

MSc. Department of Health Informatics

Supervisor: Prof. Dr. Ünal Erkan Mumcuoğlu

September 2016, 99 pages

Electron Microscopy Tomography (EMT) technique produces 3D images of cells comprising hundreds of slices of high resolution frames. Segmentation of membranes in these images are necessary in order to reveal the relations between the structural components of the cell and its behaviour. The physical shape of the crista which is a membrane of the mitochondria has been hypostatized for being an early indicator for many diseases or mitochondrial dysfunctions. Automatic segmentation of cristae in EMT images are necessary since it needs a huge human effort to manually segment these membranes. In this study, a method for automatic and robust segmentation of the crista membrane in mitochondria is proposed. The method incorporates a pre-processing stage in which a bilateral image smoothing is applied for noise removal while preserving the crista membrane boundaries. The cristae membranes are first detected by an artificial neural network (ANN) trained on cropped mitochondria images from three different data sets. When a portion of the membrane boundary is almost or totally invisible, ANN may produce disconnected segmentation. In order to overcome this issue and increase the final performance by means of detecting the barely invisible membrane boundaries and decreasing false alarms, a boundary growing method called '*directional growing*' is proposed. The method is tested with examples from four different data sets and numerical and visual analysis of the results are conducted.

Keywords: Cristae Segmentation, Artificial Neural Networks, Electron Microscopy

ÖZ

ÜÇ BOYUTLU ELEKTRON MİKROSKOPİ TOMOGRAFİ
GÖRÜNTÜLERİNDEKİ KRİSTA MEMBRANLARININ YAPAY SİNİR
AĞLARI KULLANILARAK BÖLÜTLENMESİ

Karadeniz, Merih Alphan

Yüksek Lisans, Sağlık Bilişimi Bölümü

Tez Yöneticisi: Prof. Dr. Ünal Erkan Mumcuoğlu

Eylül 2016, 99 sayfa

Elektron Mikroskop Tomografi (EMT) tekniği yüzlerce kesit yüksek çözünürlükte karelerden oluşan görüntüler üretir. Hücrenin yapısal bileşenleri ile hücrenin davranışı arasında ilişkinin ortaya çıkarılabilmesi için bu görüntülerdeki membranların bölütlemesi önem arz etmektedir. Mitokondrinin bir membranı olan kristanın fiziksel şeklinin birçok hastalık ve mitokondri fonksiyon bozukluğu için bir erken uyarıcı olduğu hipotezi ortaya atılmıştır. EMT görüntülerindeki kristaların manuel bölütlemesi çok fazla miktarda insan gücü gerektirdiğinden bunların otomatik bölütlemesi çok önem arz etmektedir. Bu çalışmada, mitokondri içindeki kristaların otomatik bölütlemesini sağlayacak bir yöntem önerilmektedir. Yöntem, görüntülerdeki gürültünün giderilmesi için bilateral görüntü filtrelemenin uygulandığı bir ön işleme aşaması içermektedir. Krista membranları üç farklı veri setinden kırpılarak elde edilmiş mitokondri görüntüleri ile eğitilmiş bir yapay sinir ağı (YSA) ile tespit edilmektedir. Eğer bir kirsta membranın kenarları neredeyse veya tamamen görünmez ise YSA kopuk bir krista bölütlemesi yapabilir. Bu sorunu ortadan kaldırmak ve az görünen kenarları tespit ederek ve yanlış tespitleri azaltarak son performansı artırmak için 'yönsel büyütme' adı verilen bir yöntem önerilmiştir. Yöntem dört farklı veri setinde test edilmiş ve sayısal ve görsel analizler gerçekleştirilmiştir.

Anahtar Sözcükler: Otomatik Krista Bölütlemesi, Yapay Sinir Ağları, Elektron Mikroskopi

To My Mother

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

xii

CHAPTER 1

INTRODUCTION

Mitochondria is known as the energy source of the cell. This energy production is achieved by means of ATP synthesizes which is done by the use of dietary calories (Zick, Rabl, & Reichert, 2009). The force behind the production of ATP from ADP is the constitution of an electrochemical gradient on inner mitochondrial membrane by which the cristae is formed (Zick, Rabl, & Reichert, 2009).

Although mitochondria synthesizes 95% of cellular metabolic energy, recent studies like (McBride, Neuspiel, & Wasiak, 2006), (Zick, Rabl, & Reichert, 2009) and (Scheffler, 2008) revealed that it takes critical roles on controlling various metabolic functions such as regulation of cellular life and death (Mumcuoglu, et al., 2012) and cell degeneration-regeneration (Taşel F. S., PHD Proposal, 2012).

Mitochondria have double membrane structure where the outer membrane forms external shape and the inner membrane forms cristae by means of invaginations as seen in Figure 1. Since the detailed very first observation of mitochondria under electron microscope by Prof. George Palade and Prof. Fritiof Sjöstrand (they did not work together but both made outstanding contributions to science of the mitochondrion), many unknowns about the function of mitochondria has been revealed or at least hypothesized. In these studies many evidence showed that there may be a link between mitochondrial function and its physical structure (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016).
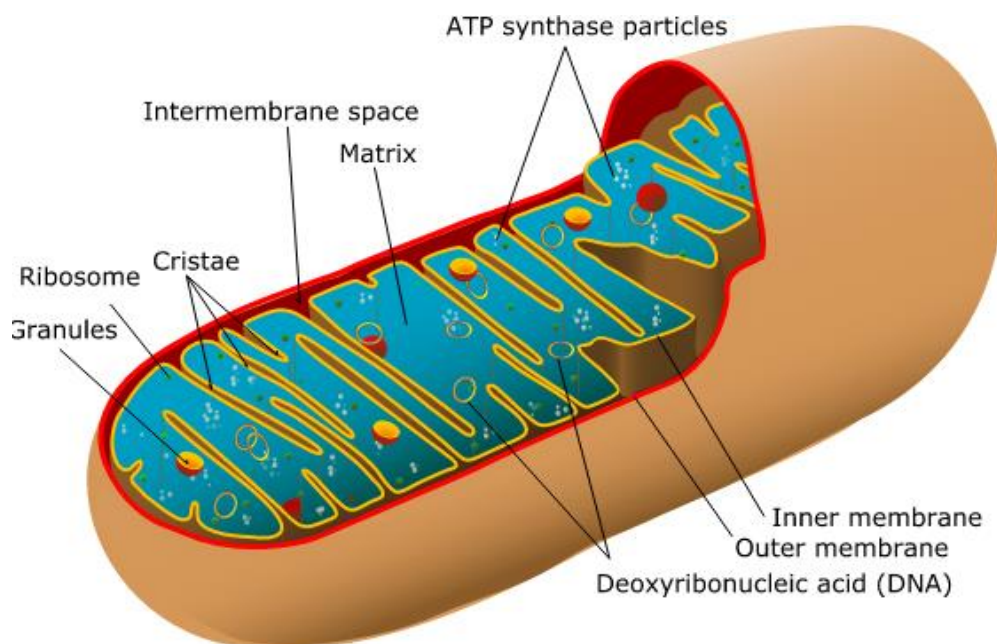
Figure 1. Diagram of the Mitochondrion. (Image taken from: https://simple.wikipedia.org/wiki/Mitochondria. Last seen at 25.07.2016)

Although electron microscopy is a promising technique in order to reveal the relations between the physical structure and functional behaviour of the mitochondria, there still exists obstacles to overcome because electron microscopy produces 3D stack images consisting of hundreds of frames which makes the morphological analysis of mitochondria very difficult. In order to overcome this issue, manual segmentation tools such as IMOD (Kremer, Mastronarde, & McIntosh, 1996) and Amira are proposed. Once the manual segmentation is obtained, these tools allow to model and analyse the 3D structure of electron microscopy images in 3D computer aided environment. Analysing the 3D models of EM images in computer environment is certainly required because only then it is possible to measure many features of mitochondria in massive numbers of images. As well as segmentation with these tools still requires a huge human effort, the produced 3D models may include many faulty results due to human error because the manual marking of so large stacks of images requires highly concentrated specialists with dedicated attention. These problems directed scientist to develop automatic algorithms aiming to produce 3D models of EM images consisting segmented mitochondria boundaries. With these systems, hundreds of 3D EM images consisting of hundreds of frames in each would be automatically segmented into required regions and so that it would be possible to test the hypothesis proposed by many scientist about the relation between many disease and shape of the mitochondrion.

Many studies (which are explained in the reminder of this chapter) hypothesized that the physical shape of cristae is also an important indicator for many diseases or mitochondrial dysfunctions. Problems arising when analysing hundreds of stacks of EM images are valid for cristae also. Analysing the morphology of cristae by means of manual methods is a more difficult task than analysing the mitochondria since there exist many times more cristae than mitochondria in each image and each of the cristae may be formed in various different shapes.

Although there exist many studies for the automatic segmentation of mitochondria in the literature, this is not true for cristae. There exists only a limited numbers of studies that address the automatic segmentation of cristae. In the study by Narasimha et al. (Rajesh, Ouyan, Gray, McLaughlin, & Subramaniam, 2009) a texton-based joint classification and segmentation algorithm that deals with the whole cell image is proposed. The data set they used in their study was obtained by scanning electron microscopy (SEM) so the resolution was low and the boundaries of the cristae in the images could be seen as a single thick line. In another study by Bazan et al. (Bazan, Miller, & Blomgren, 2009) a level set algorithm subsequent to a novel noise removal pre-processing is applied. Although promising results are achieved, they tested their algorithm with a single and very clear image (where all membrane boundaries were clearly visible) obtained by electron microscopy tomography (EMT) technique. In a work by Sanchez et al. (Martinez-Sanchez, Garcia, & Fernandez, 2011), membrane segmentation including mitochondria and cristae electron tomography images is aimed. They proposed a generic algorithm that can be used for segmentation of all membranes in the cell including Golgi apparatus, mitochondria and its inner structure. Although they showed that their algorithm performs well on various data sets, the single tomogram image they used in order to present their result for mitochondria and cristae segmentation was of very low resolution so the cristae seemed as a thick and single line so that their method can be seen more of a mitochondria boundary detector

instead. Methods developed for mitochondria outer membrane detection cannot be effectively used for the segmentation of cristae because it differs from mitochondria in two points: (i) cristae can be in various forms of shape and size and (ii) there exists many disconnected cristae membranes that shall automatically be connected and segmented. This study aims to propose a method which aims the automatic and robust segmentation of only the cristae membrane in mitochondria. Since this study is concerned on the segmentation of only the cristae membranes, it is expected that the proposed method will overperform the antecedents which are not specialized in cristae membranes.

1.1. Biological Background

When mitochondrion was first identified under light microscope, it was seen as a bacteria living in the cell which was indeed far ahead from its time (Scheffler, 2008). Today its relation to prokaryotes is recognized, but it is clearly known that they cannot reproduce independently outside the cell (Scheffler, 2008). After the use of electron microscopy the mitochondria research boosted rapidly. Electron microscopy let the researchers to visually observe the intercellular structures of the mitochondria so that it induced novel understanding about the morphology and functionality of the mitochondria. Sample slices of mitochondria from various types of cells can be seen in Figure 2 & Figure 3.
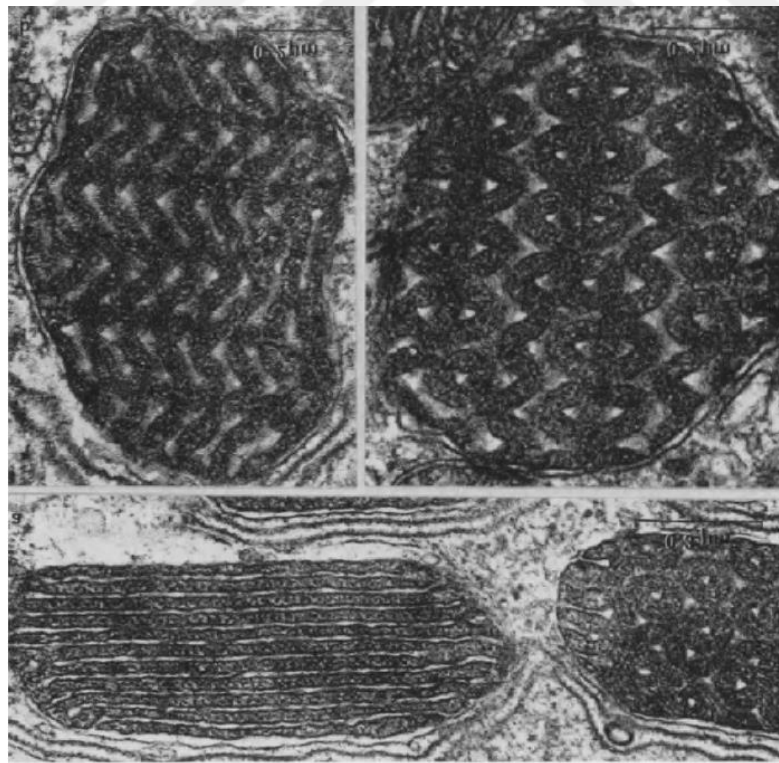
Figure 2. Scanning electron microscopy images of mitochondria *(Munn, 2014, p. 25)*
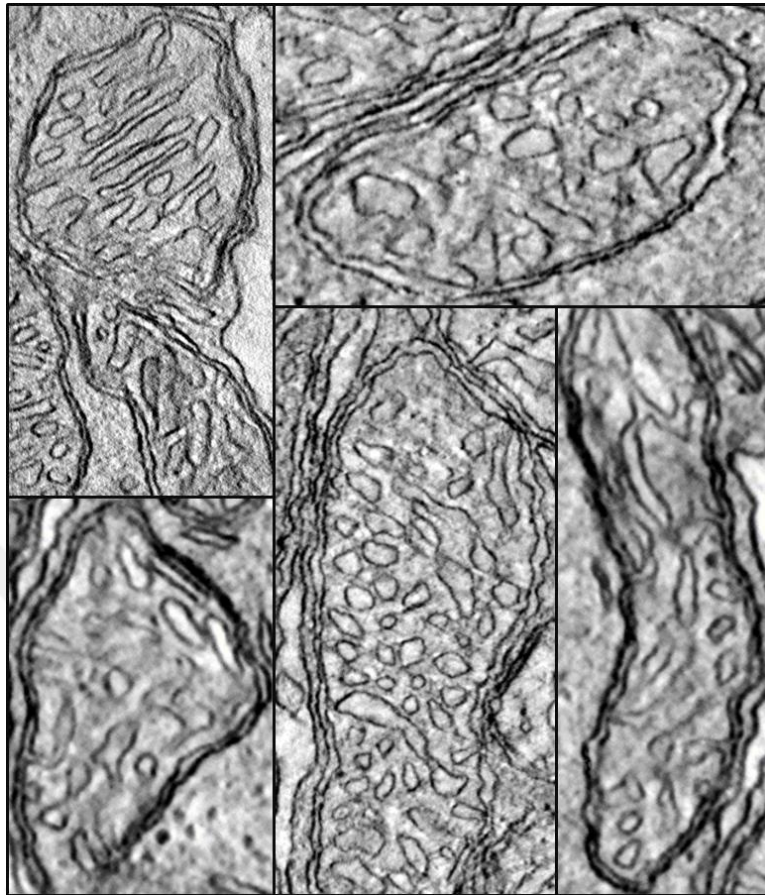
Figure 3. Transmission electron microscopy images of mitochondria. (Image are taken from the data set used in this study)

Palade unveiled the ultrastructure of mitochondrial membranes in 1953 (Palade, 1953) by the use of thin sectioning electron microscopy. After this study, the morphology of the inner mitochondria could be observed in detail and only then it was discovered that the inner membrane of the mitochondria forms convolutions. These convolutions were proposed to provide an increase for the capacity of oxidative phosphorylation (Zick, Rabl, & Reichert, 2009).

In order to express the morphological organization of the intercellular membrane of the mitochondria, Palade proposed "Baffle model" as seen in Figure 4 and suggested that the inner membrane of the mitochondria is folded into a new component which was named as "cristae mitochondriales" (Palade, 1953). Baffle model implies that "the mitochondrial inner membrane is convoluted in a baffle-like manner with broad openings towards the intracristal space regard the cristae as invaginations of the inner membrane with rather broad openings" (Zick, Rabl, & Reichert, 2009).

Another model - which is called 'septa model'- for the morphological organization of inner membranes is proposed by (Sjöstrand, 1953). This model (see Figure 4) implies that the septa increases the total surface area of the inner membrane (Zick, Rabl, & Reichert, 2009).

In 1966, after the investigation of thin serial section images obtained by means of transmission electron microscopy, a new model that identifies the presence of the

structures called 'cristae junctions' is proposed in (Daems & Wisse, 1966). This model (see Figure 4) could not achieve strong acceptance until electron microscopy tomography (EMT) method is used for imaging the ultrastructure of the mitochondria (Zick, Rabl, & Reichert, 2009). In (Manella, Marco, Penczek, Bernard, & Frank, 1994) EMT is used to visualize the rat liver mitochondria and they showed that the cristae is connected to inner boundary of the mitochondria with narrow tubular structures which was named as pediculi cristae in the work by Daems at al. (Daems & Wisse, 1966).
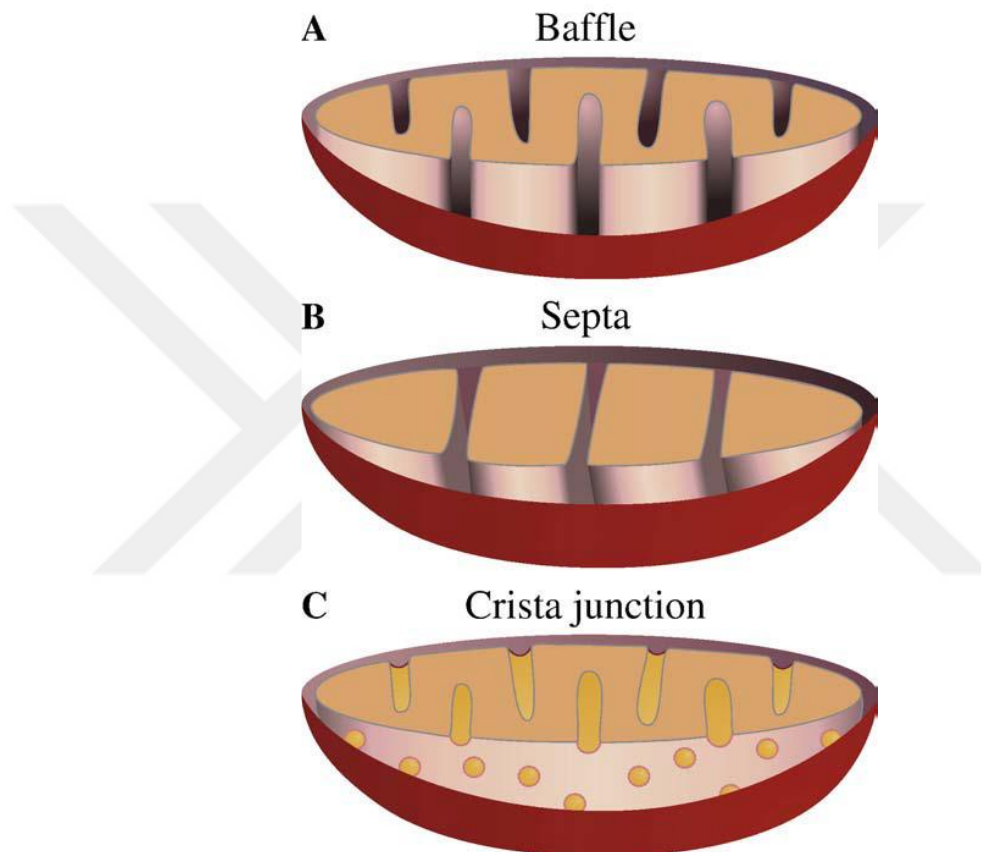
Figure 4. Proposed Models for Inside of the Mitochondria *(Zick, Rabl, & Reichert, 2009)*

All the proposed models indeed explain the same general basic pattern consisting of an outer boundary membrane (OBM) which is also limiting the size of the mitochondria, an inner boundary membrane (IBM), just peripheral to outer membrane, which surrounds the matrix of the mitochondria, and different sized membranous structures which are named as cristae mithochondriales (Munn, 2014). The OBM and the IBM form the double layer envelope (Rabl, 2009) of the mitochondria as seen in Figure 5.

Cristae in mitochondria can appear either free or attached to the inner membrane with the structures called Cristae junctions (Manella, Marco, Penczek, Bernard, & Frank, 1994) (Frey, Renken, & Perkins, 2002). Cristae junctions are small narrow tubular structural openings that are attached to IBM and exhibit sizes ranging from 12 to 40 nm (Zick, Rabl, & Reichert, 2009).

The space inside the cristae is called intracristal space while the space between the outer and the inner boundary membranes is called peripheral space (Munn, 2014).

The structural explanation of the mitochondria can be seen in Figure 5.
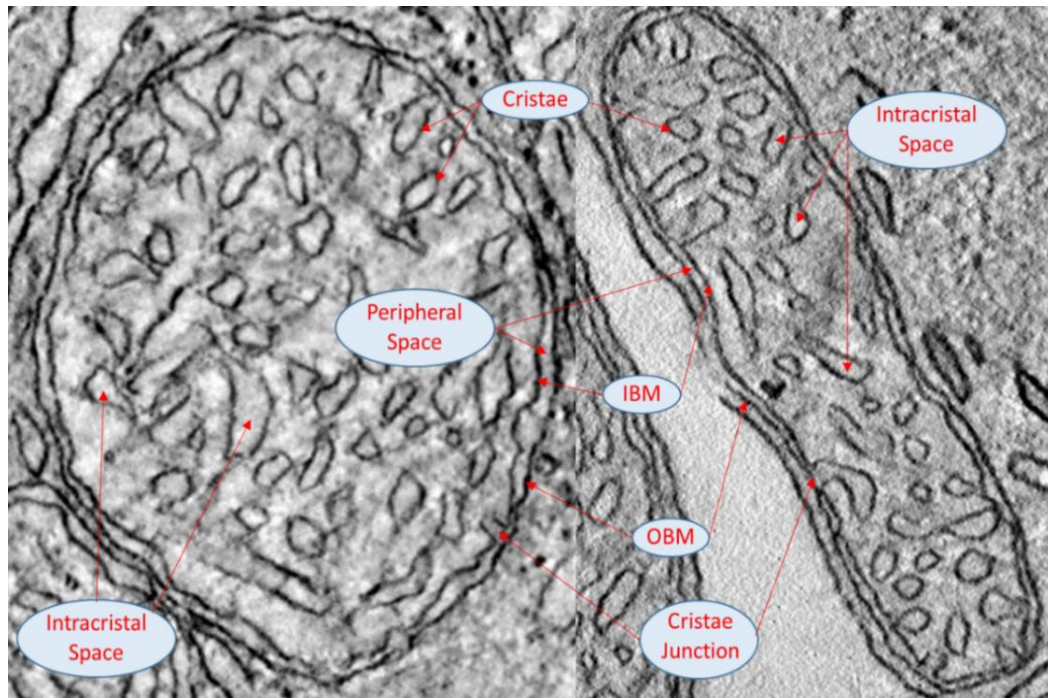
Figure 5. Structural components of the Mitochondria.

1.2. Mitochondrion and Cristae and Their Relation to Many Disease

The correlation of structural status and the function of the mitochondria attract much interest as the relation between the mitochondria and degenerative disorders such as Alzheimer's and Parkinson's disease are revealed (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016).

In the work by Gabor et al. (Gabor & Kunz, 2013), many evidence that unveil the connection between the functionally deceived mitochondria and neurodegenerative disorders have been studied. They also mentioned the quality control role of the mitochondria in neurons:

In addition to the quality control on the protein level, recent evidence suggests also the operation of a quality control on organelle level. The prerequisite of a functional organelle level quality control is that neurons must be able to distinguish between ''intact'' and ''damaged'' organelles. This distinction is apparently based on the potential of the mitochondrial inner membrane and the rate of generation of reactive oxygen species (ROS) (Gabor & Kunz, 2013).

It was shown that mutations in mitochondrial DNA causes early ageing in mice (Trifunovi, et al., 2004). This result deduces persuasive evidence that the loss of mitochondrial functionality shall be considered as a causative factor instead predecessor in aging of mammals (Zick, Rabl, & Reichert, 2009).

6

It has been known that structural alterations in the intercellular mitochondrial structures are associated to several diseases in humans (Zick, Rabl, & Reichert, 2009). One important example is the structural alterations observed on Barth syndrome patients. The size of the mitochondria of lymphoblasts of these patients enlarges while surface of the cristae considerably shrinks (Acehan, Xu, Strokes, & Schlame, 2007).

Another disease in which mitochondrial morphology alterations are observed is the Alzheimer's disease. In Alzheimer's disease; mitochondria appear swollen (Zick, Rabl, & Reichert, 2009) and the cristae membranes are arranged in parallel stacks or in concentric positions (Baloyannis, 2006). Structural alteration of mitochondria is also observed in Parkinson's disease (Trimmer, 2000).

In another study it is observed that several functional changes occur in cancer cells and this reveal the evidence that the mitochondria may be responsible in tumour formation by means of "*reactive oxygen species (ROS), decreased oxidative phosphorylation, and a corresponding increase in glycolysis*" (Verschoor, et al., 2013).

Consequently the functional significance of the mitochondrial structure is still an open issue.

1.3. Motivation: Importance of Automatic Cristae Membrane Segmentation

Imaging the intercellular structures of mitochondria by the help of electron microscopy has boosted the knowledge about the mitochondria and revealed many unknowns for the internal structure of mitochondria. Although electron microscopy imaging techniques allow many novel insights for many details of the mitochondria and its structure, it is still a challenging task to reveal the underlying mechanisms that drive many diseases inside mitochondria. With the use of electron microscopy imaging techniques, many studies attempt to reveal the relation between morphological properties (shape, size, fragmentation, elongation etc.) of the mitochondria, cristae, and cristae junctions and many disease like Barth syndrome (Acehan, Xu, Strokes, & Schlame, 2007) (Brandner, et al., 2005), various types of cancer (Han, et al., 2006) (Exner, et al., 2007), Parkinson's disease (Exner, et al., 2007), Wolf-Hirschhorn syndrome (Dimmer, et al., 2008), Autosomal dominant optic atrophy (Alexander, et al., 2000) etc.

In most of these studies mentioned above, a massive amount of human effort were required in order to reveal the proposed underlying mechanisms for related diseases. In electron microscopy imaging it is desired to obtain the image resolution as high as possible to let the specialist investigate as much detail as possible. But since mitochondria has a large shape when compared to its intercellular structures, it is a difficult task if not possible to make simultaneous analysis of large volumes of 3D mitochondria images. This brings forth the need of massive labour dependent and time consuming effort for marking the membranes in images before the analysis of the scientists.

The automatic segmentation of the membranes of mitochondria can resolve this problem and let the scientist simultaneously review and analyse the structure of mitochondria with the help of 3D computer aided modelling as well as mine data to

infer statistical results. Hopefully this will help the scientist to enlighten many hidden morphological mechanisms that are directly related to disease as Alzheimer's, Cancer etc…

1.4. Current Electron Microscopy Imaging Techniques

The term "tomography" is widely used encompassing many different methods but it literally means visualization of slices used to reconstruct the interior of an object from its projections (Frank, 1992).

The novel and modern insights into the intercellular structure of mitochondria could only be understood after the use of electron microscopy in 1950s and by then the mitochondrial research is boosted (Zick, Rabl, & Reichert, 2009).

In electron microscopy a beam of electrons are used to visualize the specimen that is methodically prepared before the visualization process. Electron microscope has greater resolving power than light microscope, allowing the specialists to observe the very fine details of micro seized pre-prepared specimens (https://www.jic.ac.uk/microscopy/intro_EM.html). Basically there are two types of electron microscopy: Transmission Electron Microscope and Scanning Electron Microscope.

In TEM technique the electron beam is transmitted through an ultra-thin specimen and it interacts with specimen as it passes through the specimen. The result of the interaction of the electrons transmitted through the specimen is used to compute and reconstruct the slice image of the specimen as seen in Figure 6. The transmitted electrons are magnified and focused onto an imaging device, such as a fluorescent screen, a photographic film, or a charged-coupled device (CCD) sensor (Khan, 2012).
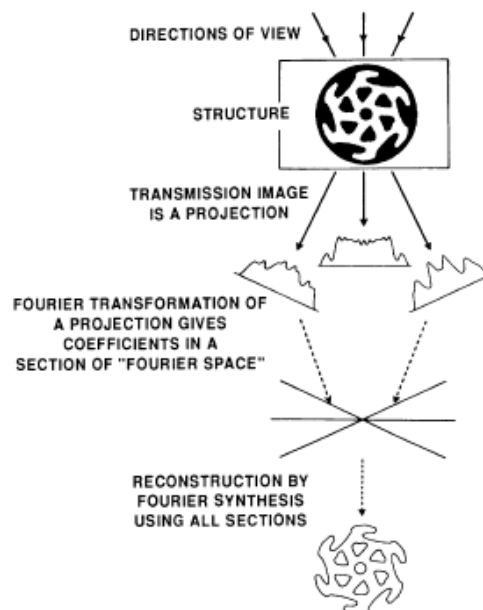
Figure 6. Principle of two dimensional reconstruction in tomography *(Frank, 1992)*.

EM tomography allows the three-dimensional reconstruction of volumes of rather thick sections (∼0.5 μm). For this a series of electron micrographs is recorded while the sample is tilted over a wide range of angles (Zick, Rabl, & Reichert, 2009).

In TEM, an electron gun produces the electron beam which is directed by lenses and electric field. The image is formed once the electron beam passes through the specimen. The projected electron intensity is measured by a detector. The intensity on the image is proportional to the amount of electron absorption by the tissue. The process is repeated for different tilt angles of the electron beams (Taşel F. S., PHD Proposal, 2012). Finally the total absorption of each tilt angle is presented in a sonogram-like data as seen in Figure 6. The images are then reconstructed by means of back projection techniques.

"The projection angles are physically limited to a typical range of ±70°. The angular separation of each consecutive projection is typically 1°-2°. These limitations cause some reconstruction artefacts such that particles may appear elongated in the direction normal to the surface of the specimen. Another limitation is the thickness of the specimen that must allow sufficient penetration. The 3D image obtained by this technique is typically large in two dimensions but thin in the third dimension. Therefore, a single mitochondrion may not be fully visible in one dimension." (Taşel F. S., PHD Proposal, 2012).
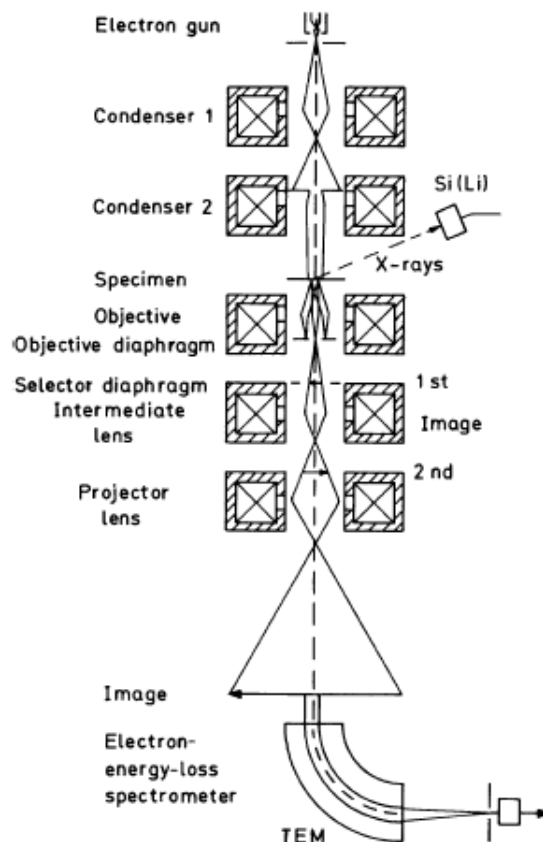
Figure 7. Schematic ray path for a transmission electron microscope equipped for additional x-ray and electron energy-loss spectroscopy *(Kohl & Reimer, 2008)* .

"In SEM technique, the surface of specimen is visualized in 2D (see Figure 9). The electron beam is focused on the specimen and deflected by the electric field which is generated by the scanning coils. The electrons hit the surface of specimen and scatter. The intensity of electrons back-scattered from the surface is measured by a detector. In order to obtain a 2D image, electron beams are deflected onto different locations to scan the whole surface. Imaging a 3D volume can be achieved by utilizing a special type of SEM technique which is called Serial Block-Face Scanning Electron Microscopy (SBFSEM). In SBFSEM technique, a stack of 2D images are obtained by cutting ultra-thin sections from the surface using a diamond knife and then imaging by SEM (see Figure ?). This technique supplies 3D volumetric data formed by a series of 2D images each associated to a single slice. On the contrary to TEM tomography, the slice thickness is higher than the lateral resolution. But, there is no limitation for the number of slices" (Taşel F. S., PHD Proposal, 2012).
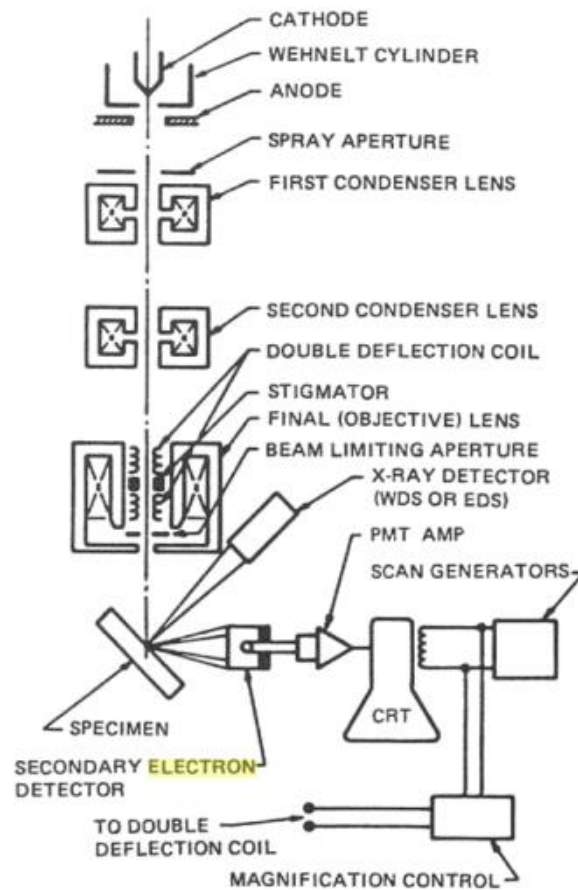
Figure 8. Schematic diagram of the parts of the electron column *(Lyman, Newburry, Goldstein, Williams, & Roming, 2012)*.
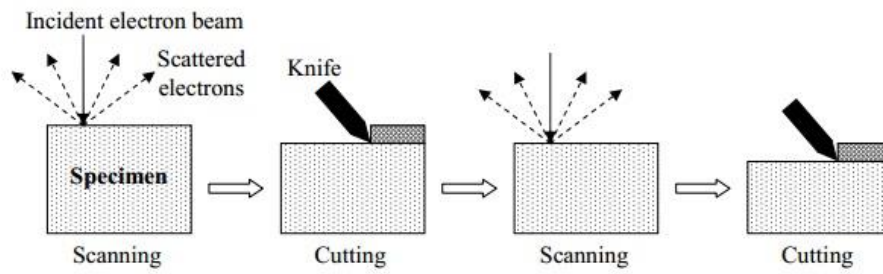
Figure 9. Scanning and cutting phases in SBFSEM *(Taşel F. S., PHD Proposal, 2012)*.

On the contrary to TEM imaging, SEM produces images with three-dimensional appearance of the specimen (see Figure 10) since this technique has the capability of imaging with large depth of field and shadow relief effect of the secondary and backscattered electron contrast (Glodstein, et al., 2012).
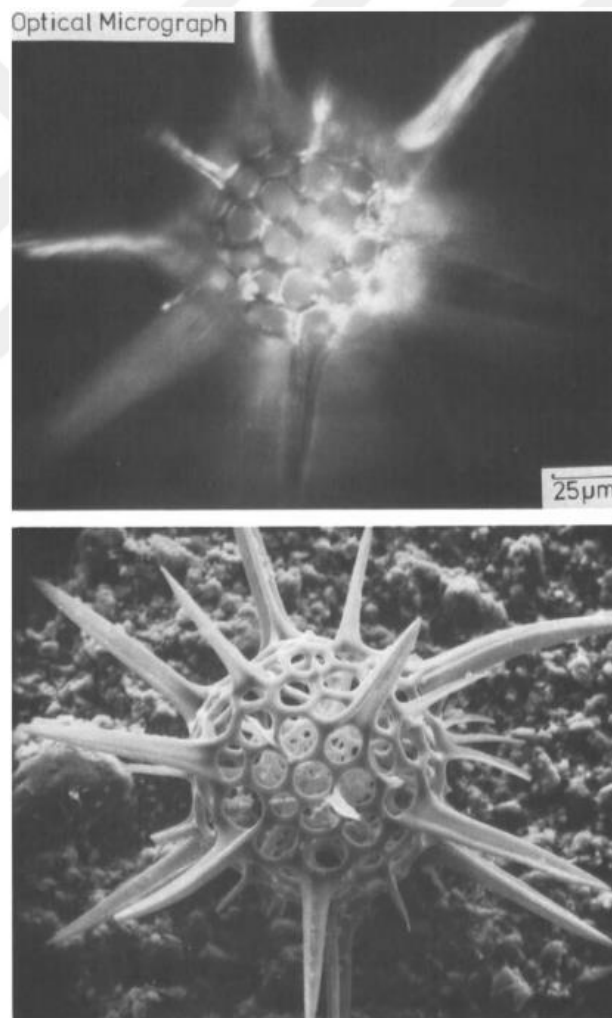
Figure 10. Skeleton of small marine organism (the radiolarian Trochodiscus longispinus).The upper image shows the skeleton viewed with a light microscope and the lower image shows the same skeleton viewed with SEM *(Glodstein, et al., 2012)*.

1.5. Aim and Scope

As mentioned in the previous chapters, analysing the EM images in 3D computer environment is critical in order to enlighten the unknowns lying behind the mechanism of the shape of the mitochondria and the cristae and infer some results that allow the early detection of many disease.

The automatic segmentation of mitochondria in EM images has been studied many times and satisfying results are obtained. For instance the detection of mitochondria's inner and outer boundary has successfully been accomplished by Mumcuoglu et al. (Mumcuoglu, et al., 2012) and Tasel et al. (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016). But there exists a small number of studies aiming at the segmentation of cristae.

In this study it is aimed to develop a robust and automatic segmentation method for cristae membranes which may be fully or partially seen inside the mitochondria. The proposed method will be automatic because there will be no human intervention in any step of the method. The method proposed can be used for many different types of cells since the development and the testing will be done for cristae with different types and shapes. Also the discrimination of the cristae from other intercellular structures will be ensured so that the false positive rate can be minimised.

1.6. Contributions

This study attempts to accomplish the automatic segmentation of cristae by using the power of ANNs and proposing a method to increase the performance obtained from the trained ANN. In order to achieve a robust solution, two problems shall be overcome one of which is that the cristae can be in various sizes and shapes. This problem is overcome with the ability of neural networks to model a very wide range of non-linear variation in the data. The second problem is that many portions of crista membrane boundary may be disconnected or barely visible due to limitations in imaging technique and equipment. Since artificial neural networks may underperform in these regions, a method that attempts to connect these disconnected boundaries by means of directional growing is developed.

The proposed method is applied to electron microscopy tomography images and the results and the performance are presented.

CHAPTER 2

RELATED WORK

Introduction

In this chapter basic image segmentation methods are roughly explained. These methods are either directly used or benefitted from in the pipeline of the proposed algorithm. In the second section of this chapter, artificial neural networks are explained in detail. It is important to understand every detail of the networks and the derivation of backpropagation algorithm in order to build an effective architecture that can perform well on the data to be used. And finally a limited portion of the literature search done during this study is presented in the final section.

2.1. Image Segmentation

"Image segmentation is the task of finding groups of pixels that go together" (Szeliski, 2010). The aim in segmentation is to distinguish an image into one or more labels depending on one or more features of related regions (Rogowska, 2000). From point of statistics, segmentation problem is referred as cluster analysis and is a widely studied area with hundreds of different algorithms (Szeliski, 2010). From the very beginning of the computer vision science; segmentation problems have been studied widely (Brice & Fennema, 1970) (Salotti, 2001) (Rosenfeld & Davis, 1979). Historically, the development process of segmentation algorithms can be divided into two stages. In the early stage, divisive and agglomerative algorithms, which are based on region splitting and merging techniques, were widely used; but more recently, algorithms often depending on the optimization of a global criterion has been used (Szeliski, 2010, p. 237). Another approach for categorizing the segmentation algorithms is based on the basic properties of intensity values by means of discontinuity and similarity. Discontinuity category is based on the detection of the sudden changes in the images. In the similarity approach, the image is segmented depending on a predetermined feature set (Gonzalez & Woods, 2002, p. 690). Indeed, segmentation techniques can be categorized in many ways depending on the point of view (Rogowska, 2000). If the categorization is done based on the human interaction, the techniques can be divided as; manual, semi-automatic, and automatic (Shareef, Wang, & Yagel, 1999). Although there exists a vast amount of different techniques in the literature, important segmentation methods include; thresholding (basic, adaptive, Otsu's), edge detection operators (Sobel, Prewitt, Roberts, Robinson, Kirsh, Frei-Chen, Laplacian, Laplacian of Gaussian etc.), other edge detection methods like Marr and Hildreth (Marr & Hildreth, 1980) and Canny (Canny, 1986), Snakes (Kass, Witkin, & Terzopoulos, 1988), Active Shape Models (Cootes, Taylor, Cooper, & Graham, 1995), level sets (Cremers, Rousson, & Deriche, 2007), region splitting and merging (Felzenszwalb & Huttenlocher, Efficient graph-based image segmentation, 2004), mean shift (Comaniciu & Meer, 2002) and mode finding (Bishop, 2006), normalized cuts (Malik, Belongie, Leung, & Shi, 2001), graph cuts (Boykov & Funka-Lea, 2006).

2.1.1. Thresholding

There exists many thresholding methods based on different features extracted from the image such as; images histogram, mean intensity value, standard deviation of intensity values or local gradient. But amongst all different thresholding methods global thresholding is the most intuitive approach (Rogowska, 2000).

In global thresholding it is assumed that the image has a bimodal histogram (see Figure 11) so that the desired object can be extracted from the image by means of comparing the intensity values with a threshold value (Rogowska, 2000).
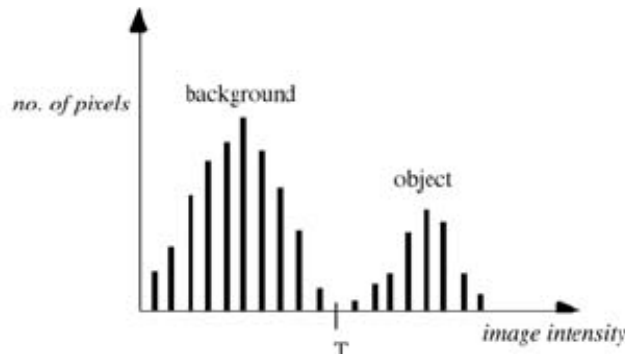
Figure 11. An example for bimodal histogram *(Rogowska, 2000)*.

The thresholding operation which produces a binary image is defined as

$$I(x,y) = \begin{cases} 1, & if\ I(x,y) > T \\ 0, & \text{otherwise} \end{cases} \qquad (1)$$

Where I is the image to be threshold, $I(x,y)$ is the intensity value at location (x,y), and T is the threshold value.

2.1.2. Edge Detection

Edge detection is the most frequently used approach in for segmenting the images based on locals intensity changes (Gonzalez & Woods, 2002). Local intensity changes are determined by the gradient of the pixel by considering the neighbourhood of related pixel. The first order derivative approximation of the image intensity changes are called first gradient while second order derivative approximation yields to second gradient. The direct application of continuous differentiation is not possible on discrete nature of digital image so that the gradient of an image is computed by means of differencing approach (Rogowska, 2000). There exist many different approximations for the calculation of gradient of an image. In order to obtain the gradient of an image the partial derivatives at each pixel shall be computed (Gonzalez & Woods, 2002). A basic approximation of the partial derivative of a pixel over its neighbourhood is given by;

14

$$g_x = \frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y) \qquad (2)$$

and

$$g_y = \frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y) \qquad (3)$$

But for computing the edge direction, it is more common to use symmetric masks about the centre point for which the gradient is being computed. These masks reveal more information regarding the direction of the edge because they allow to compute the data on opposite sides of the pixel that is being considered (Gonzalez & Woods, 2002). Some common masks used for edge detection can be seen in Figure 12.
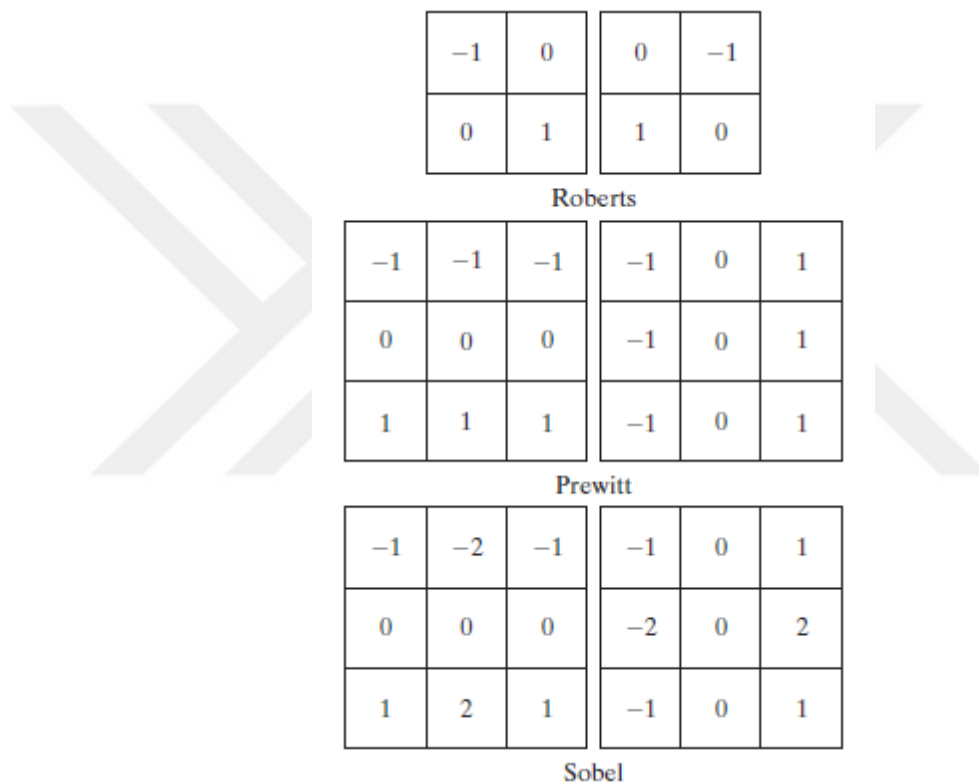
Figure 12. Commonly used mask for computing the gradient at a point.

The 3x3 masks shown in Figure 12 above involve convolutional calculations by means of summation of weighted pixel intensities with the constants in masks. There exist various gradient operators used for edge detection. Some of these are; Sobel, Roberts, Prewitt, Robinson, Krisch, Frei-Chen.

The above methods are simply filters the image by means of masking operators. These filters operate on the image without any provisions being made for the characteristics of edges and noise (Gonzalez & Woods, 2002). But the noise content in the image highly effect the performance of gradient based edge detection methods. In order to overcome noise issue and improve the performance of gradient based methods, methods such as (Marr & Hildreth, 1980) and (Canny, 1986) are proposed. These methods take into account factors such as image noise and the nature of edges themselves.

2.1.3. Active Contours

Every object in the natural environment have curves corresponding to their boundaries. Detecting and locating these curves play such an important role on detection or recognition of desired objects. Active contours are used to locate such boundary curves in images. While there exist different active contour methods, all methods iteratively move towards their final solution under the combination of image and optional user-guidance forces.

One of the most famous active contour method is called snakes (Kass, Witkin, & Terzopoulos, 1988). In snakes method a two dimensional spline curve is iteratively moves towards some defined image features (Szeliski, 2010).

2.1.4. Region Based - Region Splitting and Merging

Region based algorithms attempt to segment the image by directly finding the regions of interest (Gonzalez & Woods, 2002). One of the two main approaches in region based segmentation is region growing while the second is split and merge approach.

In the region growing approach a set of seed points are selected and neighbouring pixels of these seed points that have predefined properties are added to the region (Gonzalez & Woods, 2002).

Split and merge algorithms work in two different types; by recursively splitting the whole image into pieces based on region statistics or merging pixels and regions together in a hierarchical manner (Szeliski, 2010). Also there are methods that use the advantage of both approaches (Horowitz & Pavlidis, 1976). An image is divided into a set of arbitrary, disjoint regions and then merge or split the regions that satisfy the predefined conditions.

2.1.5. Image Segmentation in Medical Imaging

In medical imaging, segmentation is widely used in labelling the images depending on the anatomical features such as blood vessels (Hoover, Kouznetsova, & Goldbaum, 2000) (Soares, Leandro, Cesar, Jelinek, & Cree, 2006) (Kobashi, Kamiura, Hata, & Miyawaki, 2001), (Frangi A. , Niessen, Vincken, & Viergever, 1998), internal organs like liver (Masomi, Behrad, Pourmina, & Roosta, 2012) or ventricular (Paragios, 2003), muscles (Kwok, 2004), and bones (Sebastian, Tek, Crisco, & Kimia, 2003) (Suzuki, Abe, MacMahon, & Doi, 2006). It is also used for classifying the pathological regions such as lesion, tumour boundary detection (Zhu & Yan, 1997), detection of mammographic calcifications (Shen, Rangayyan, & Desautels, 1993), or tissue deformities in medical images (Rogowska, 2000).

With the use of 3D imaging techniques, segmentation has gained more importance for the construction of 3D computer modal of the scanned volume. Computer Aided Analysis (CAA) of these medical images is very important since modern 3D imaging modalities (such as Computer Tomography, Magnetic Resonance Imaging, and Electron Microscopy etc.) have reached a very high quality and resolution levels so

that an extensively large volume of data is produced. Examining and analysing this massive data is a very hard task, if not possible, for human specialist.

Some important studies that show the diverse applications of segmentation in 3D Electron Microscopy image analysis are; computerized detection of mitochondria on electron microscope images (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016), (Mumcuoglu, et al., 2012), (Lucchi, Smith, Achanta, Knott, & Fua, 2012), (Seyedhosseini, Ellisman, & Tasdizen, 2013, April), the automation of reconstruction of neural tissue in serial section transmission electron micrographs (Mishchenko, 2009), segmentation of neuronal membranes in electron microscope images (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012) (Jurrus, et al., 2010).
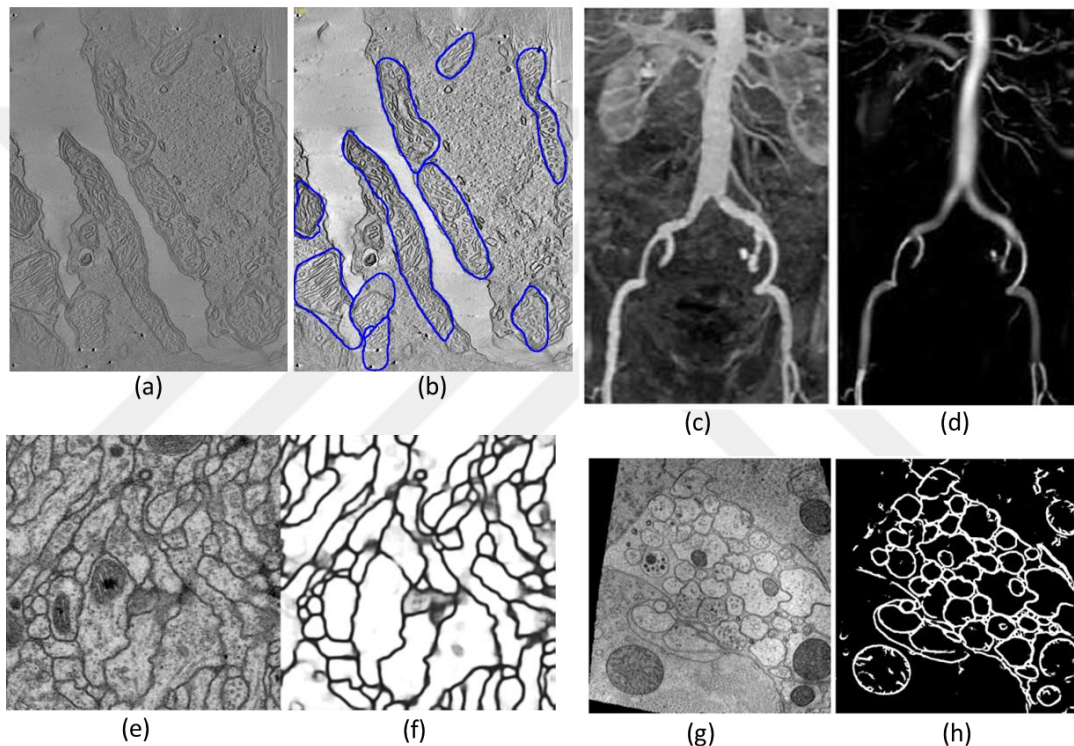
Figure 13. Various segmentation results of medical images from state-of-the art studies. (a) and (b) show the original and the resultant image of mitochondria detection in *(Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016)*. (c) and (d) show of the vessel enhancement method developed in *(Frangi A. , Niessen, Vincken, & Viergever, 1998)* for aortoiliac MRA. The neuronal membrane segmentation applications; (e) and (f) with deep neural networks *(Ciresan, Giusti, Gambardella, & Schmidhuber, 2012)* and (g) and (h) with serial neural networks *(Jurrus, et al., 2010)*.

Consequently; in medical image analysis, segmentation play critical role including for various measurements, visualization of multi-dimensional images, and computer aided diagnosis.

2.2. Supervised Learning and Artificial Neural Networks

The origin of the motivation for developing neural systems goes down to the effort of McCulloch and Pitts (McCulloch & Pitts, 1943) and Hebb (Hebb, 1949) for modelling the brain. Their approach includes independent units that process the inputs and return

17

a simple function of the total input. They proposed the most prevailing paradigm in neuro-science that the power of the brain lie in the connections of neurons instead the power of individual neurons (Dunne, 2007, p. 1). On the path that (McCulloch & Pitts, 1943) and (Hebb, 1949) opened, neural networks evolved to the machines that model the problems and tasks just as the brain does but by the use of electronic components and software on digital computers instead of biological metabolisms (Haykin, 2009, p. 2).

Multilayer neural networks are most widely used for function approximation and pattern classification because they are very powerful tools for modelling very complex nonlinear problems (Krawczak, 2013, p. V). Especially for the last two decades, they have been used in many fields mostly for the problems of pattern recognition. Although many different types and architectures (for different kinds of problems) have been proposed (Bengio, Courville, & Vincent, 2013), feed-forward network architectures such as the multi-layer perceptron and the radial basis function network became the most popular ones amongst others (Bishop, 2006, p. preface). These are very general frameworks that maps several input variables to several output variables by means of many parameters which are adjusted depending on the problem itself. This adjusting process is called *training* (Bishop, 2006).

Simon Haykin (Haykin, 2009, p. 2) defined a neural network as; "*a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use*".

Neural networks are composed of many information processing elements called neurons. A neuron model many of which is used for designing various types of artificial neural networks can be seen in Figure 14.
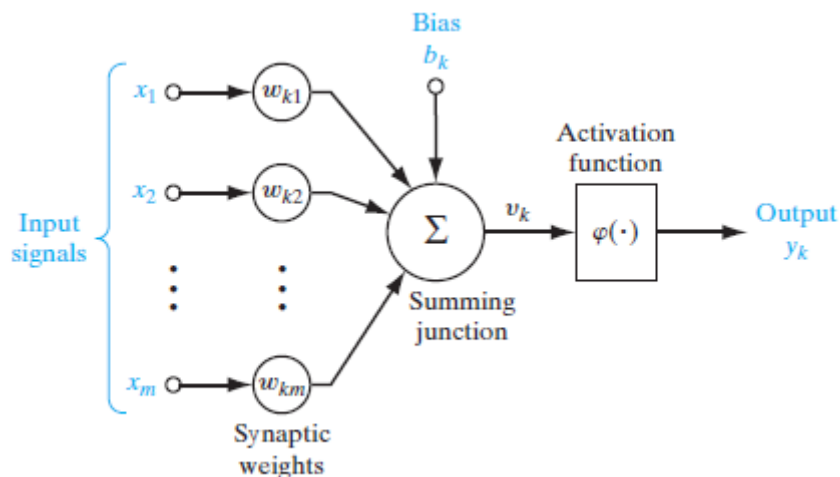
Figure 14. The neuron model for the kth neuron in an imaginary network. (Image taken from *(Haykin, 2009)*)

There exist three basic components in a neuron model. First, there are linking elements or synapses which connect the input signals to the summing component of the neuron. Each connection in a neuron model has a coefficient which determines the strength of its own (Haykin, 2009, p. 10) since the related input signal is multiplied by this coefficient so that the contribution of the input signal is determined. If the weight of

the link is very small then it means that the related input has a small contribution to the output of that neuron. The multiplied input signals are summed in an adding unit which is the second basic component in the neuron model proposed in Figure 14. The third main component is the activation function also called the squashing function which is used for limiting the amplitude of the neuron (Haykin, 2009, p. 10). There also exist a bias input component to the summing component. Bias increase or decrease the total input of the activation function depending on its sign (Haykin, 2009, p. 11). The neuron model is mathematically expressed as;

$$v_k = \sum_{j=1}^{m} w_{kj}x_j + b_k \qquad (4)$$

and

$$y_k = \varphi(v_k) \qquad (5)$$

where j is the index number of the input signals $x_1, x_2, ..., x_m$ and $w_{k1}, w_{k2}, ..., w_{km}$ are the synaptic weights of input signals respectively. It shall be noticed that the weights $w_{k1}, w_{k2}, ..., w_{km}$ are belong to k^{th} neuron as "k" denotes the index of the neuron (there exist only one neuron in the Figure 14 above which is the kth one). b_k is the bias and $\varphi(v_k)$ is the activation function whose output produces the final output, y_k, of the neuron. The bias b_k may be viewed as the fixed coefficient for an affine transformation (see Figure 15) to the output $u_k = \sum_{j=1}^{m} w_{kj}x_j$ in the linear combiner (summing component of the neuron).
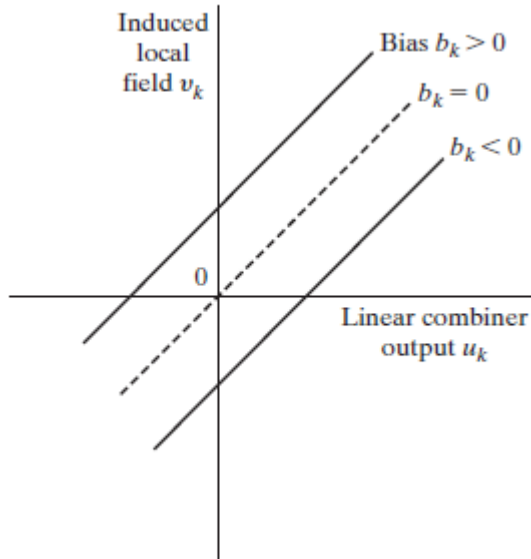
Figure 15. The transformation applied to the output of linear combiner, u_k, so that graph of v_k versus u_k does not pass through the origin.

The linear combinations of the input signals are transformed by activation functions, denoted as $\varphi(v_k)$ in Equation 6. There exist many different types of activation functions including; step function which is also known as Heaviside function in

engineering community, ramp function, sigmoid function, and Gaussian function. Step function is defined as in Equation 6 and graphed as in Figure 16.

$$\varphi(v) = \begin{cases} 1 & if\ v \geq 0 \\ 0 & if\ v < 0 \end{cases} \qquad (6)$$

A neuron that use threshold function in its activation layer is referred as McCulloch-Pitts model due to their cutting-study accomplished in 1943 (McCulloch & Pitts, 1943).
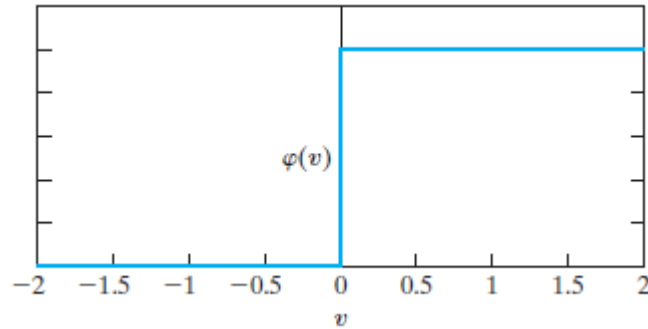
Figure 16. Heaviside (threshold) function (Image taken from *(Haykin, 2009)*)

Another commonly used activation function is the sigmoid function. The graph of the sigmoid function is in the form of "S" as it is an increasing function exhibiting an elegant balance between linear and nonlinear behaviour (Haykin, 2009). A typical example for the sigmoid function is the logistic function which is defined as;

$$\varphi(v) = \frac{1}{1 + e^{(-av)}} \qquad (7)$$

The graph of the sigmoid function can be seen in Figure 17. In Figure 17 the behaviour of the function as a is increased can be seen.
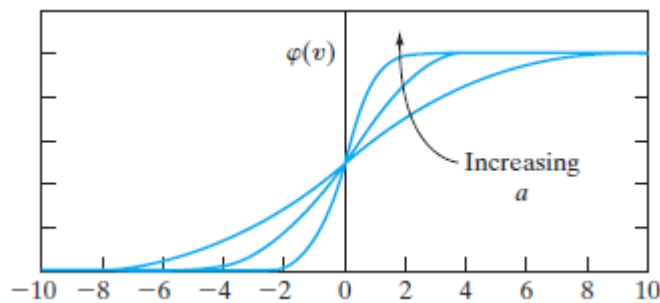
Figure 17. The sigmoid function as 'a' is increased (Image taken from *(Haykin, 2009)*).

2.2.1. Feedforward Neural Networks

Feedforward neural networks are commonly used multilayer neural network frameworks modelling the linear or non-linear mappings between input and output variables (Bishop, 2006). Feedforward multilayer neural networks are capable of

approximating any function with sigmoid or linear output unit and "squashing" activation functions in the hidden layers on closed and bounded sets (Dunne, 2007, p. 37). For $f: R^p \rightarrow R^q$, for a closed, bounded $K \subset R^p$ and $\varepsilon > 0$; there exist a e^* that

$$\|f(x) - e^*(x)\| < \varepsilon, \forall x \in K \qquad (8)$$

where $f(x)$ is the real output of function aimed to be modelled, and $e^*(x)$ is the output of the function. This result is presented by (Hornik, Stinchcombe, & White, 1989) and (Funahashi, 1989) independent of each other.

In a feedforward network, the input signals projects directly to the next layer of the network (which is the output layer in a single layer network and the next hidden layer in a multilayer network). Each signal passes through the network only towards the forward direction without activating any backward signals. Multilayer feedforward networks have hidden layers between the input and the output layers. Hidden layers mediate between the external input and the network output so that the network can extract higher-order statistics from its input (Haykin, 2009, p. 22). Because of the increase in synaptic connections and neural interactions' dimension, the network gain more global perspective (Churchland & Sejnowski, 1992) despite its local connectivity.

The hidden nodes in a neural networks plays an important role during the learning process such as deducing the features that enables the network to use the extracted features that characterize the training data. They transform the input data to a new space so that the discrimination between the patterns can be accomplished more easily (Haykin, 2009, p. 126).

For the learning process of a neural network the back-propagation algorithm is one of the most commonly used algorithm. The backpropagation algorithm proceeds in two stages: the forward and the backward stage. In the forward stage, the input signal is passed through each neuron in the next layer by means of being multiplied with the corresponding weights of the neurons until the output layer. An example of the forward run calculation for one hidden neural network as in Figure 18 are shown in the equations below.

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix} * \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} v_{h1} \\ v_{h2} \end{bmatrix}$$

$$\rho_1(v_{h1}) = y_{h1}$$
$$\rho_1(v_{h1}) = y_{h2} \qquad (9)$$
$$\begin{bmatrix} w_{o1} & w_{o2} \end{bmatrix} * \begin{bmatrix} y_{h1} \\ y_{h2} \end{bmatrix} = \begin{bmatrix} v_o \end{bmatrix}$$
$$\rho_3(v_o) = y_o$$

In the beginning of the second stage, the output obtained after the forward run is compared with the ground truth of the related pattern which is used as the input vector in the first stage and an error signal is obtained. This signal is then passed through the network in the backward direction and the weight adjustments are accomplished in this stage as to minimize the calculated error which is defined as;

$$e_j(n) = d_j(n) - y_j(n) \qquad (10)$$

where $d_j(n)$ is the desired response (ground truth) of the jth output neuron in the network when n[th] pattern is used as input.
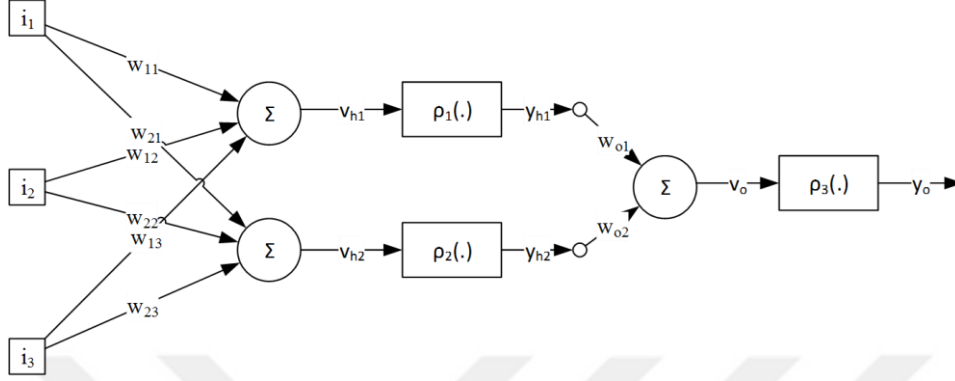
Figure 18. A multilayer neural network with 3 input elements, one hidden layer with 2 nodes, and one output node in the output layer.

The adjustment in each neuron is made by the help of gradient descent algorithm which is used to seek a direction through which the weights are updated so that the value of the error is reduced (Haykin, 2009, p. 131). The correction value in the ith input of the jth neuron is defined as;

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} \tag{11}$$

$\mathcal{E}(n)$ is the total instantaneous error energy of the whole network and η is the learning rate parameter. The Equation (11) depicts that the weight w_{ji} is updated with a rate which is proportional to the partial derivative of $\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$. Translation of the delta rule expressed in Equation (11) is that; once the direction of the change for the total error with respect to that weight is found, then make the adjustment on that weight through the opposite direction so that the total error shall reduce. The magnitude of the adjustment to be made is determined by the learning rate, η.

According to the chain rule of calculus the gradient can be expressed as

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{12}$$

This derivative calculated in Equation (12) determines the direction of search for the weight to be adjusted in the in the weight space (Haykin, 2009). Notations in Equation (12) are shown in Figure 19.

The total instantaneous error energy is defined as (Haykin, 2009)

$$\partial \mathcal{E}(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \tag{13}$$

where $e_j(n)$ is defined in Equation (10) and C is the neurons in the output layer. Differentiating both sides of the equation above gives:

$$\frac{\partial \mathcal{E}(n)}{\partial e_j(n)} = e_j(n) \qquad (14)$$

From Equation (10);

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \qquad (15)$$

Since $y_j(n) = \rho_j(v_j(n))$

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \rho_j'(v_j(n)) \qquad (16)$$

And finally

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \qquad (17)$$

Since $v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n)$ where m is the number of input applied to the jth neuron and $i = 0$ stands for the bias.

So the correction value for the weight w_{ji} yields

$$\Delta w_{ji}(n) = \eta e_j(n) \rho_j'(v_j(n)) y_i(n) \qquad (18)$$
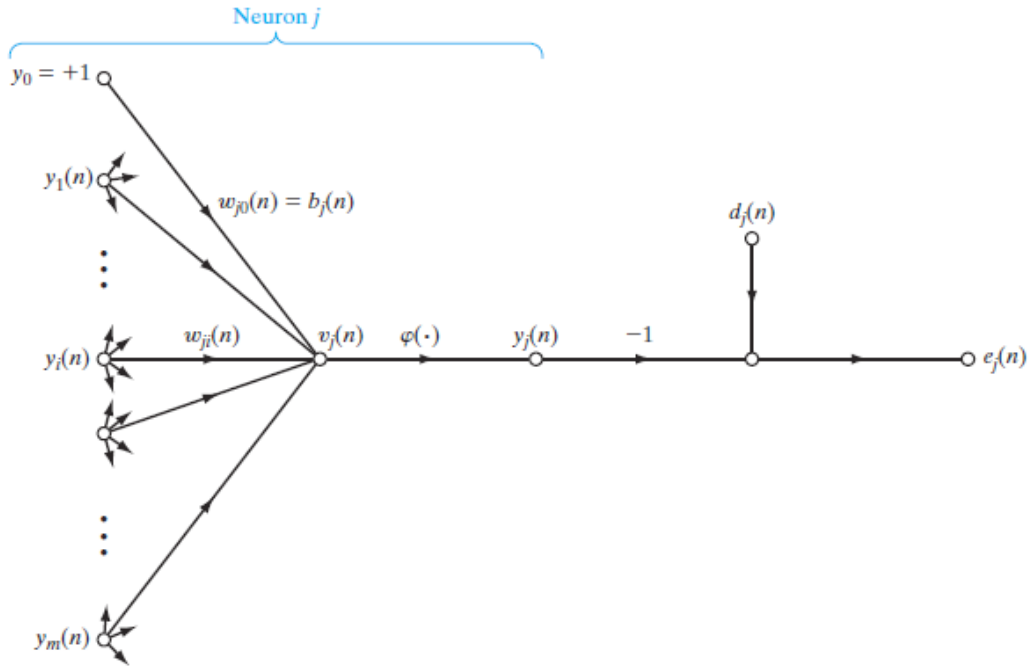
Figure 19. The flow in neuron j.(Image taken from *(Haykin, 2009)*)

The correction value in Equation (18) can be defined in terms of local gradient as in Equation (19) below.

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \qquad (19)$$

Local gradient $\delta_j(n)$ is defined by

$$\delta_j(n) = \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n)\, \rho_j'\big(v_j(n)\big) \tag{20}$$

Since there is no desired response for the neurons in the hidden layer, the error signal can be calculated in terms of the errors of the neurons that are connected to that neuron which is in the hidden layer (Haykin, 2009). The flow at the hidden layer can be seen in Figure 20 below. The local gradient can be redefined as

$$\delta_j(n) = \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = \frac{\partial y_j(n)}{\partial v_j(n)} \sum_k \frac{\partial \mathcal{E}(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \tag{21}$$

Where

$$\frac{\partial \mathcal{E}(n)}{\partial v_k(n)} = \delta_k(n) \ and \ \frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \tag{22}$$

Finally the equation for the local gradient in the hidden layer yields to;

$$\delta_j(n) = \rho_k'(v_k(n)) \sum_k \delta_k(n) w_{kj}(n) \tag{23}$$

And the correction value is defined as;

$$\Delta w_{ji}(n) = \eta \rho_k'\big(v_k(n)\big) \sum_k \delta_k(n) w_{kj}(n)\, y_i(n) \tag{24}$$
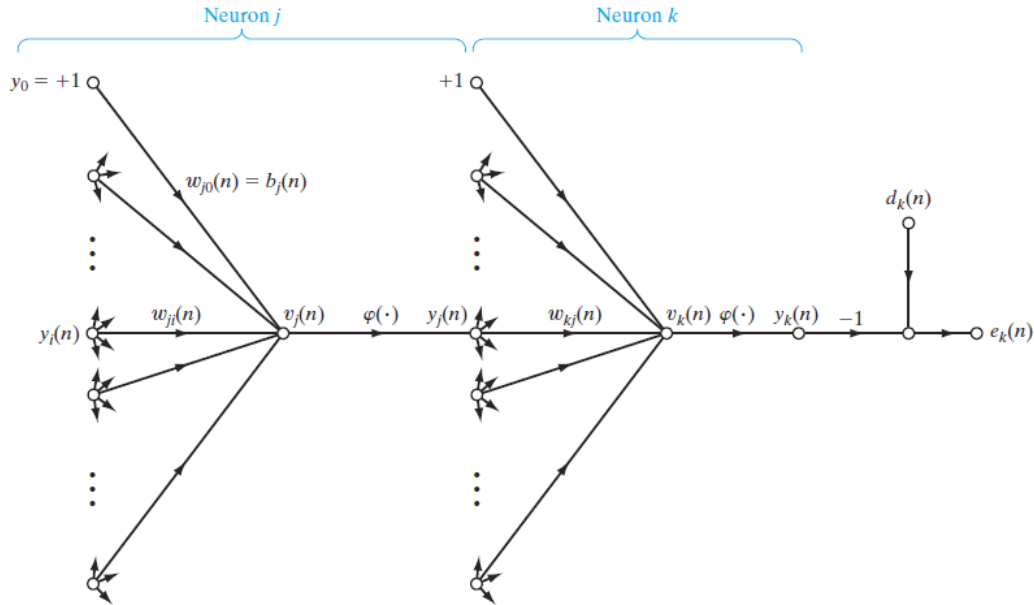
Figure 20. Flow of from hidden neuron j to output neuron k. (Image taken from *(Haykin, 2009)*)

2.2.2. Neural Networks used for EM Image Segmentation

In the paper by Ciresan et al. (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012) neuronal structures in stacks of electron microscopy images are attempt to be automatically segmented by means of a deep neural network which is used for labelling of each pixel. Their network uses the image intensities as the input extracted from a pre-defined input window centred on the pixel to be classified. As training set they used TEM image consisting of 30 images with a resolution of 512x512. All pixels belonging to the neuronal membranes are used as positive examples, and the same number of negative examples are randomly selected from remaining non-membrane pixels. This yield to 3 million patterns in total to be used in training where both labels (membrane and non-membrane) are equally represented to the network during training. They also mirrored and/or rotated by ±90° each pattern at the beginning of each epoch. The input data is manipulated by means of foveation and non-uniform sampling which makes the input windows carrying more information. Since different architectures of networks performs different for different parts of images, they average the calibrated outputs of the neural networks trained. An output from their method can be seen in Figure 21.
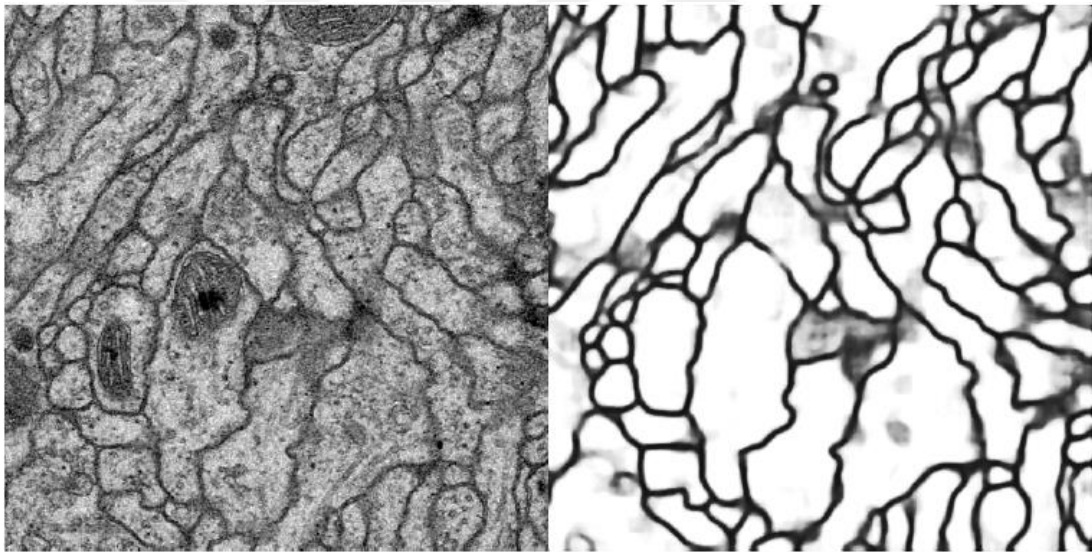
Figure 21. The original image and corresponding output of the proposed method in Ciresan et al.
(Ciresan, Giusti, Gambardella, & Schmidhuber, 2012)

In another study (Jurrus, et al., 2013) 3D images obtained by the use of Serial-section Transmission Electron Microscopy (ssTEM) and Serial Block Face Scanning Electron Microscopy (SSBFSEM) techniques are used in order to segment neuron boundaries in these image. They proposed a serial neural network architecture and trained the network with a 2D input stencil. The Figure 22 shows the diagram for the serial neural network they trained.
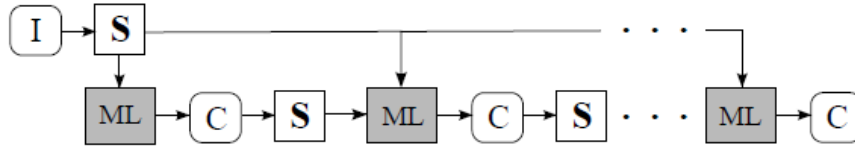
Figure 22. Serial neural network architecture proposed in Jurrus et al. *(Jurrus, et al., 2013)*. I is the input image, S is the intensity values of the image I extracted with the help of input stencil. Ml is the classifier and C is the output of the classifier in each step.

The trained serial neural network is then used to segment 3 subsequent images from 3D stacks of images. A sequential section serial network is trained with the inputs obtained from a 3D input stencil which spans 3 images. The images used for training sequential section serial network are the resultant image from serial classifiers from the first stage. The diagram demonstrating the framework can be seen in Figure 23.
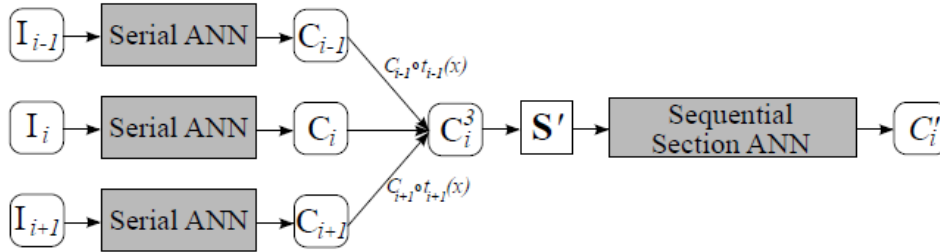
Figure 23. The final flowchart of serial neural networks trained in Jurrus et.al. *(Jurrus, et al., 2013)*.

In the study by Middleton and Damper (Middleton & Damper, 2004) the binary classification results of a trained neural network are used as external energy function for a snake in order to segment the lungs from multiple MR slices. A multilayer perceptron is trained with error backpropagation algorithm. As the input values to the network, the normalised intensity values of the image within a 7x7 input window are used for the network which consists of output layer with one node, one hidden layer with 30 nodes and input layer with 49 input elements. They used a fixed number of 1000 epoch as the stopping criteria of the training. This approach is not suggested since it does not guarantee the convergence of the error performance to the global minimum. The fixed number of epoch approach may also lead to overfitting. Since the multilayer neural network's resultant image consists of disconnected boundaries, they used an active contour model as the post-processing step in order to obtain closed lung boundaries.

2.3. Current Research on Automatic Mitochondria and Cristae Segmentation

In the work by Mumcuoglu et al. (Mumcuoglu, et al., 2012) it is aimed to detect and segment the mitochondria on electron microscope images including partially seen boundaries. Four different data sets provided by "National Center for Microscopy and Imaging Research, USA" were used for their experiments. The proposed method includes two phases. In the first stage the detection of mitochondria boundary is done

depending on the two properties of the mitochondria: mitochondria possess an elliptical shape and the boundary of the mitochondria contains a double membrane geometry. Some important geometrical properties of the mitochondria that they considered are; mitochondria can be global or elongated, the diameter (smallest cross-section) of the mitochondria is usually between 0.25μm and 2μm, the length (largest cross-section) can be up to 20μm or even longer, the membrane thickness is between 4nm and 6nm for different cell types in conventional tomograms, and the gap between the double membrane (inner and outer membrane) structure in healthy cells is almost uniform and differ from 10nm to 36nm. The proposed algorithm basically possess two phases: the detection and the segmentation phases. In the first stage of the detection phase; noise reduction, contrast enhancement and normalization is done. Once the pre-processing in the first stage is accomplished, a double ridge energy function is computed for each pixel prior to connected component analysis and morphological operations in which it is aimed to determine the probable double membrane regions. Found probable double membrane regions are used in an ellipse detection operation so that many ellipses in different sizes and orientations are detected around the double membrane boundaries. At the end of the detection phase, merge and split methods are used to obtain the mitochondria detections from many detected ellipses. At the detection phase of the algorithm, the approximate boundary positions of the mitochondria are obtained. For a complete and robust segmentation, the algorithm first refines the initial detected boundaries by means of active contours in the first stage of the segmentation phase. Still the results of active contours method may produce boundaries that are misaligned with the detected mitochondria. In order to overcome this issue, a modified live-wire graph search algorithm is used in order to finalize the segmentation. An automatic seed point selection algorithm is used for selection of robust seed points to be used by the live-wire algorithm. Although satisfying results are obtained, the proposed method is dependent to the successful removal of cristae in order to locate the peripheral mitochondrial membranes (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016).
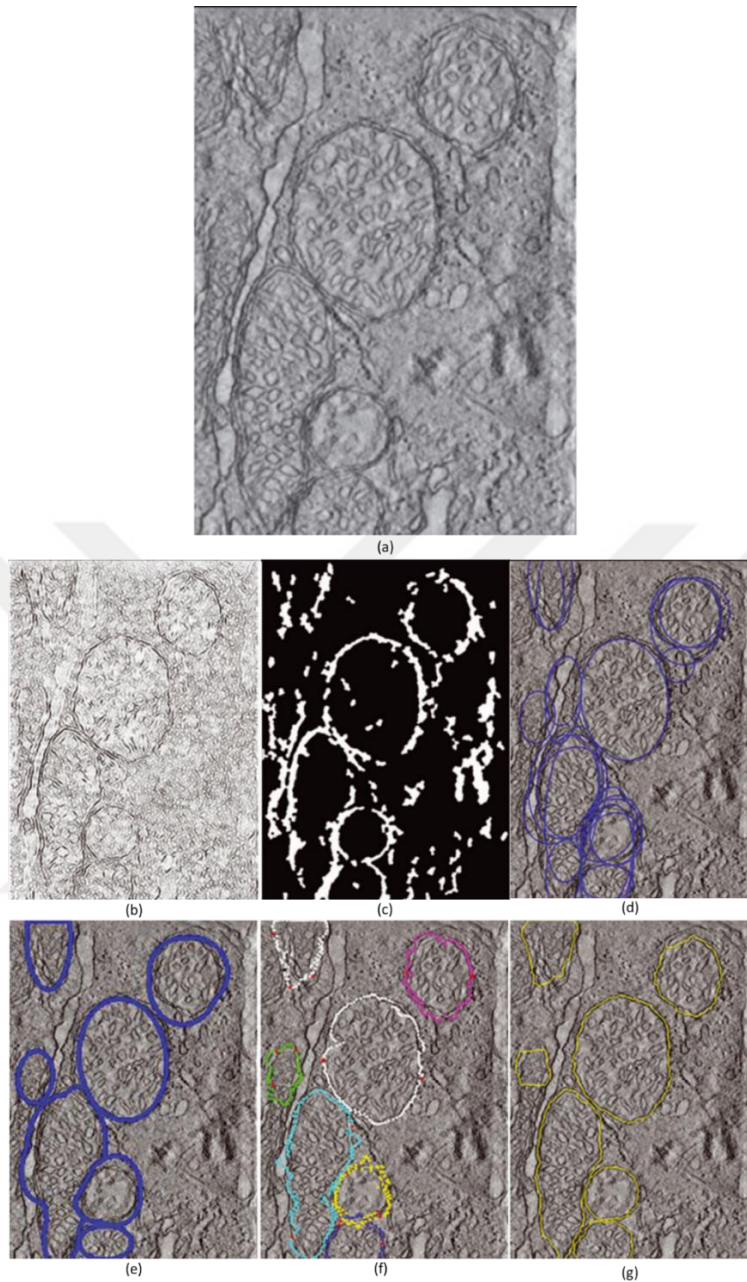
Figure 24. Results of the method in Mumcuoglu et al. (a) shows the original tested image; (b) shows the double ridge energy image where the darker pixels points the double ridge locations; (c) the image on which the thresholding morphological filtering and connected component analysis applied; (d) first detection of ellipses; (e) ellipses that are merged from (d); (f) the results of active contour algorithm and the automatically selected seed points (red points); (g) the results of live-wire contour tracking algorithm. *(Mumcuoglu, et al., 2012)*

In a study by Taşel et al. (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016), robust segmentation of mitochondria boundary membrane is aimed as their previous attempt in (Mumcuoglu, et al., 2012).

In the proposed method, a ridge detection algorithm is applied following a pre-processing which includes auto-contrast adjustment, resampling, and smoothing. In ridge detection step, membrane-like structures are obtained by a constructed Hessian

28

based ridge detector. They constructed the detector based on the observation that membranes possess a bright-dark-bright transition profile. The eigenvalues of the Hessian Matrix are used in order to define the ridge energy of each pixel. The Hessian matrix is defined as:

$$H = \begin{bmatrix} G_{xx} & G_{xy} \\ G_{yx} & G_{yy} \end{bmatrix} \tag{25}$$

$G_{xx}, G_{xx}, G_{xx}, G_{xx}$ are the second order Gaussian derivatives of the intensity values of each pixel in the image. The eigenvalues are interpreted as in Figure 25. According to this interpretation; a valley-like structure is observed when $\lambda_1 \gg |\lambda_2|$ and a saddle point in the valley is obtained when $\lambda_1 \gg 0$ and $\lambda_2 \ll 0$ on which the point belongs to a weak point in the membrane. Also they mentioned that the point corresponds to a dark blob when $\lambda_1 \approx \lambda_2 \gg 0$. Consequently, they define the ridge energy as:

$$r(\lambda_1, \lambda_2) = \begin{cases} \lambda_1 - \lambda_2 & if \ \lambda_1 > 0 \ and \ \lambda_2 > 0 \\ \lambda_1 & if \ \lambda_1 > 0 \ and \ \lambda_2 < 0 \\ 0 & otherwise \end{cases} \tag{26}$$
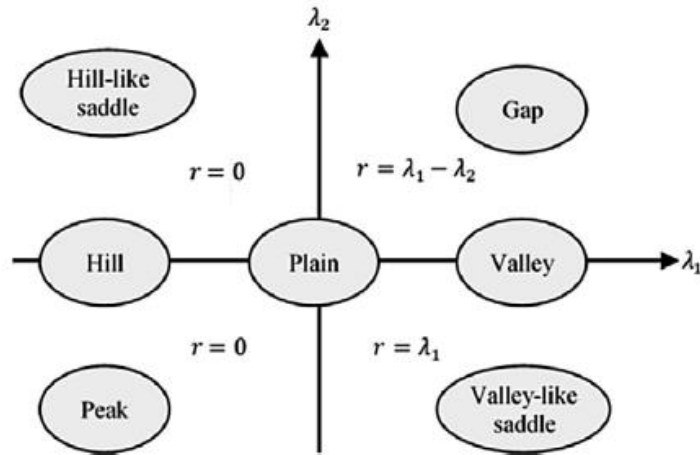
Figure 25. Topographic shapes that eigenvalues of Hessian matrix points *(Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016).*

Once the ridge energy image is obtained by the formula in Equation (26), an energy mapping algorithm is applied to ridge energy image since the classification of mitochondria cannot be accomplished only with the ridge energy image. The energy map is constructed on the observation that peripheral membranes have low curvature and are longer structures. A parabolic arc model which utilizes the ridge energy maps is then adopted and used with curve fitting and filtering algorithm in order to obtain the membrane-like patterns. Due to weak membrane regions the ridge detection may be inaccurate. At these circumstances curve fitting algorithm fails. In order to overcome this issue another step snake-based shape extraction is implemented. At the final stage a validator mechanism is built for separating the possible outliers. The validator function depends on the facts that a mitochondrion has an enclosing membrane and cristae inside. Mitochondria segmentation results of the proposed method can be seen in Figure 26.
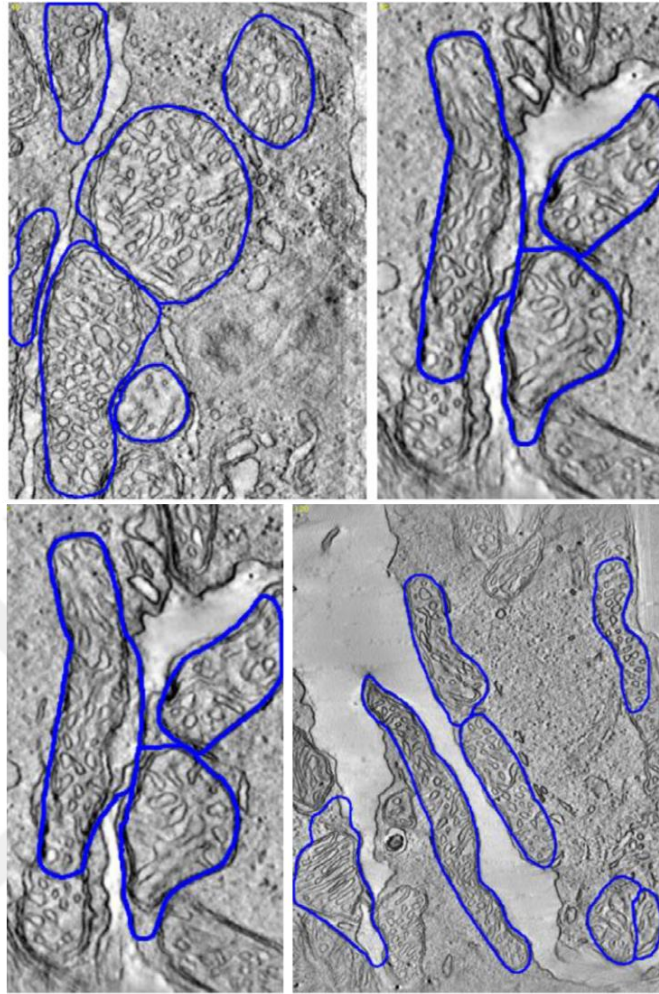
Figure 26. Results of the method in Tasel et al. Mitochondria segmentation results *(Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016)*.

In another study Seyedhosseini et al. (Seyedhosseini, Ellisman, & Tasdizen, 2013, April) attempt to segment the mitochondria in electron microscopy images used the advantage of algebraic curves. They extract textural features and shape information from images and used the power of algebraic curves for boundary detection of mitochondria in cluttered electron microscopy images. The method proposed include four stages. In the first stage of the proposed method, a rotation and scale invariant different degrees of polynomial curve fitting operation is applied on an $n \, x \, n$ image patch. In the second stage a relatively complex set of features are extracted from the inside of the fitted curve on image patches. The features extracted from the inside region of the curve include; entropy of the pixel intensities, skewness, kurtosis, variance, Hu's invariant moments, and mean. These features correspond to the textural information of the patch. Also some additional features that correspond to the shape of the curve are extracted. These include; the average intensity value of pixels on the fitted curve, Hu's invariant moments of the curve, and ratio of the number of pixels in the inner area to the curve length. In Figure 27 below, input patches, fitted polynomials and the selected fitted curve can be seen.
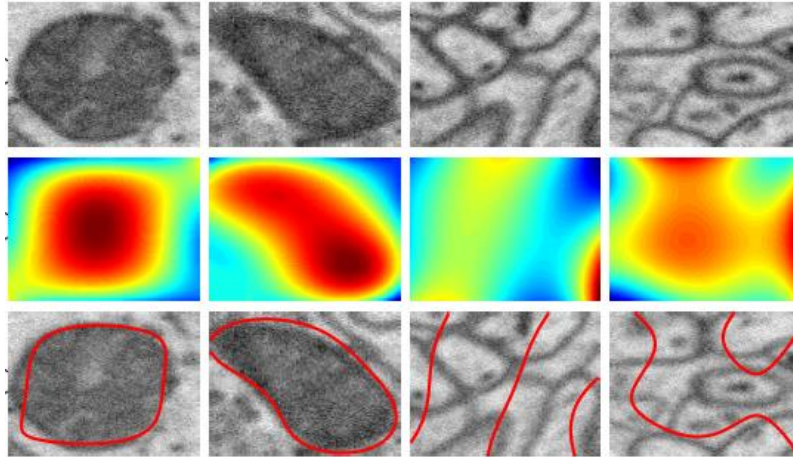
Figure 27. The fitted polynomials of the method in Seyedhosseini et al. Top left two images are two patches belonging to a mitochondrion and top right two are non-mitochondrion patches. Fitted forth degree polynomials to the related patches (same column upper images) can be seen in the middle row. The fitted curves are seen at the bottom row. *(Seyedhosseini, Ellisman, & Tasdizen, 2013, April)*

A binary random forest classifier is trained by using the features extracted. The trained random forest is used to label the pixels weather they belong to a mitochondria or not. At the final stage of the method; is the patch that is passed through the random forest is classified as positive then all the pixels that lay on the fitted curve are marked as mitochondria.
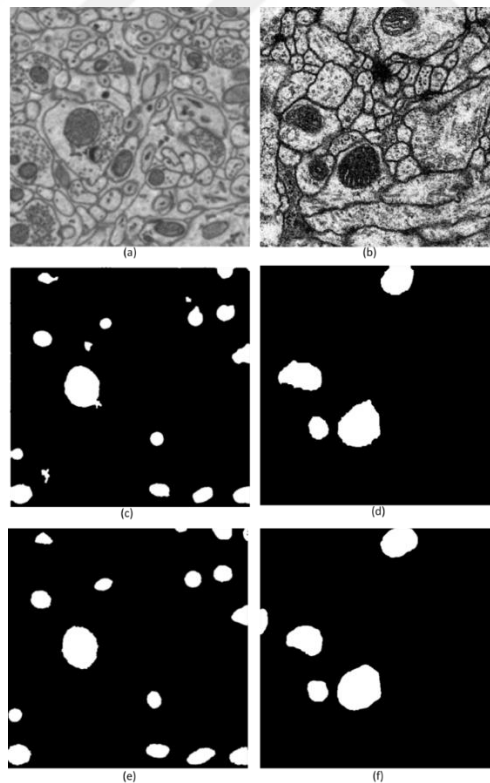
Figure 28. The fitted polynomials of the method in Seyedhosseini et al. (a) and (b) show two different test EM images. (c) and (d) are the segmentation results of the proposed method while (e) and (f) are the related ground truths of the images in (a) and (b). *(Seyedhosseini, Ellisman, & Tasdizen, 2013, April)*

31

In the work by Sanchez and Fernandez (Martinez-Sanchez, Garcia, & Fernandez, 2011) a segmentation method targeted at membrane detection is proposed. Their method which is based on local differential structure and a Gaussian-like membrane model which defines the natural limits of compartments within biological specimens. Their method include two main stages. In the first stage they produce an output map that describes how well a point in the image belongs to a boundary. Then by using this output map the final segmentation is obtained. They proposed that the intensity variation across a membrane section shows the characteristics of a Gaussian function. So they expressed the density variation across a membrane as:

$$I(r) = \frac{D_0}{\sqrt{2\pi}\sigma_0} e^{\frac{r^2}{2\sigma_0{}^2}} \qquad (27)$$

They did the inference in Equation (27) depending on the observation that the density along the normal direction to the curve of the membrane progressively decreases as the distance to the centring element of the membrane increases. In order to determine the curve direction, the eigenvectors of the Hessian matrix of the image is used. Once the direction is determined, they calculated the Hessian matrix of the membrane model (which is calculated in the first stage) with direction of the curve and then a membrane strength is defined by using the result of the gauge defined in Equation (28) below.

$$R = \begin{cases} |\lambda_1| - \sqrt{\lambda_2 \lambda_3} & if\ \lambda_1 < 0 \\ \lambda_1 & if\ \lambda_1 \geq 0 \end{cases} \qquad (28)$$

(a) (b)

(c) (d)
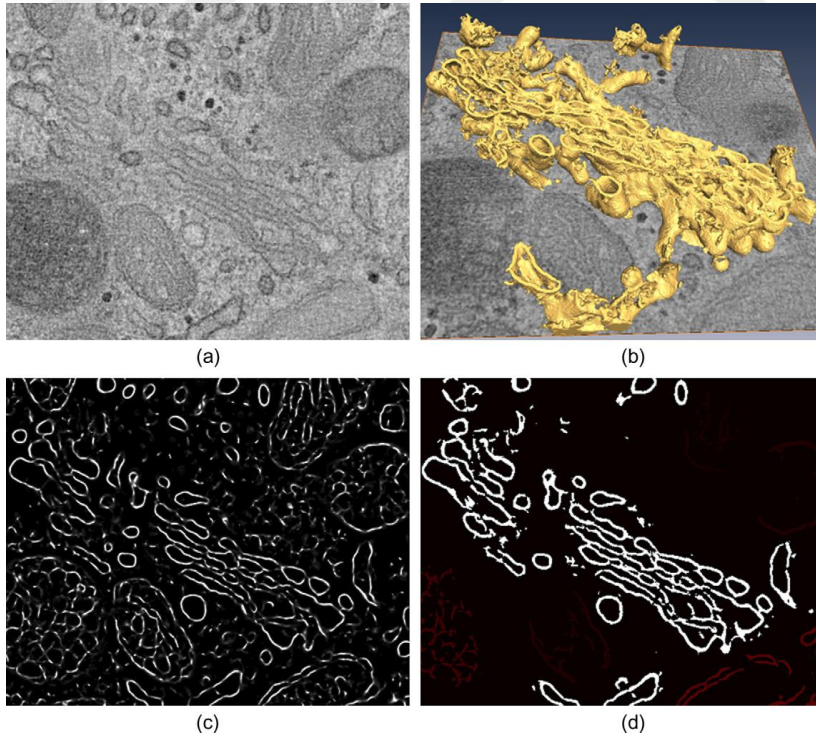
Figure 29. Results of the method in Sanchez et al. The original EMT image is seen in (a). The 3D view of the membrane which are segmented by the proposed method is seen in (b). (c) and (d) are the membrane strength image and the result of hysteresis thresholding respectively. *(Martinez-Sanchez, Garcia, & Fernandez, 2011)*

32

Subsequent to membrane strength detection by using the gauge (R) obtained in Equation (28) a hysteresis thresholding is applied. For finalization of the segmentation, the size (i.e. the number of voxels of the component) of detected membranes are calculated and only the one that are larger than a pre-set threshold are selected. In Figure 29 above, the result of their method can be seen.

CHAPTER 3

PROPOSED WORK / METHODS

3.1. Data

As mentioned before the aim of this study is the automatic segmentation of cristae boundaries on TEM images. Since this study aims to achieve the segmentation of cristae membranes without any dependency on the cell type, the data used shall be collected from various different cell types so that the generalization could be achieved in the targeted scope.

A neural network shall have enough number of nodes and hidden layers to approximate complex functions. If the number of nodes are too small then the network may not have enough representative power to model the data (Lai, 2015). A neural network aiming at modelling relatively simple problems requires smaller number of parameters (smaller degree of freedom) which requires smaller data sets for training (Lai, 2015). The same suggestion is valid from the opposite side: if the problem to be approximated is relatively complex than the size of the neural network, then the number of parameters to be trained are increased accordingly, bigger datasets are required in order to ensure that the generalization is achieved. Since the segmentation of the cristae membranes can be seen as a multidimensional and highly non-linear classification problem, huge numbers of patterns representing the whole input and output space are required.

3.1.1. Data Sets

The dataset used in this study is obtained from the Cell Centered Database (CCDB). The CCDB is a web accessible database for high resolution data from light and electron microscopy and supported by "National Center for Microscopy and Imaging Research, USA" and also is a part of the Cell Image Library.

Four dataset including various sized cristae membranes and show diversity in image contrast are selected. Each dataset differs in resolution of the images and size of the voxel (see Table 1).

Table 1. Properties of datasets used in the experiments. Table taken from (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016)

Dataset name	Image size (width × height)	Number of slices	Voxel size (nm) ($X \times Y \times Z$)
6_22.sub	1960 × 2560	91	1.1 × 1.1 × 1.1
bclpb-d.sub	720 × 878	61	2.4 × 2.4 × 2.4
cone.sub	736 × 1010	97	2.4 × 2.4 × 2.4
gap18_sub	350 × 600	54	2.2 × 2.2 × 2.2

Cone.sub set which obtained by using a conventional back-weighted EMT technique is formed from a mouse retina-cone cell (Taşel F. S., PHD Proposal, 2012). The other

sets belong to a mouse retina-rod cell and are collected with a slow-scan CCD camera (Taşel F. S., PHD Proposal, 2012).

Electron microscopy tomography image examples from each dataset can be seen in Figure 30.
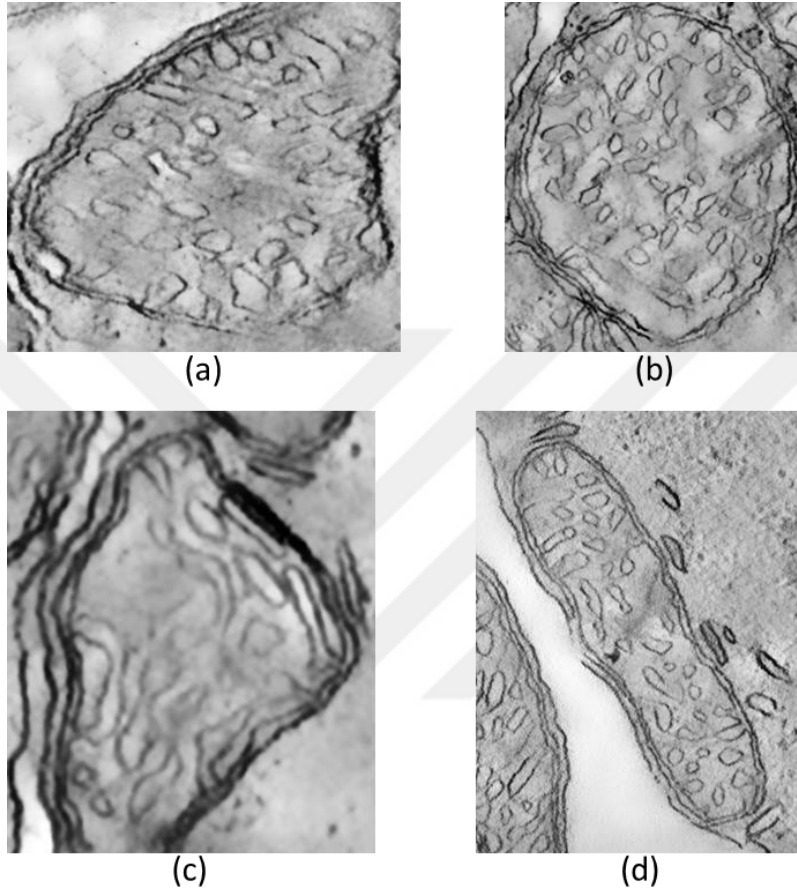
Figure 30. Mitochondria examples from different datasets used. (a) from bclpb-d.sub, (b) from cone_sub.sub, (c) from gap18_sub, (d) from 6_22.sub

3.1.2. Ground Truth

In this study the ground truth for training and testing the neural networks and evaluating the results is obtained by manually marking the data set which is provided by the National Centre for Microscopy and Imaging Research, University of California, San Diego, CA, USA. Since manual marking of TEM images is a highly labour dependent effort due to large number of images and pixels, vast amount of this thesis study was spent for manually marking the cristae membranes. Although it is a common approach to obtain the ground truth by averaging the markings of many specialist, in this study markings are done only by the writer of this thesis due limited means of specialists so that ground truth may contain erroneous markings since the writer of this thesis is not competent in the cell morphology science. Also it is important to note that marking such a big ground truth database requires an outstanding labour dependent and time consuming effort.

It shall be mentioned that Sub_6.22 data set which is one of the four data sets used in this study were marked by the National Centre for Microscopy and Imaging Research, University of California, San Diego, CA, USA. However the markings were roughly done such as misaligned markings on cristae membrane boundaries (see Figure 31), they provided an initial truth for fine tuning the Sub_6.22 data set and a directive example while marking the other data sets such as cone_sub, bclpb-d, and gap 18.
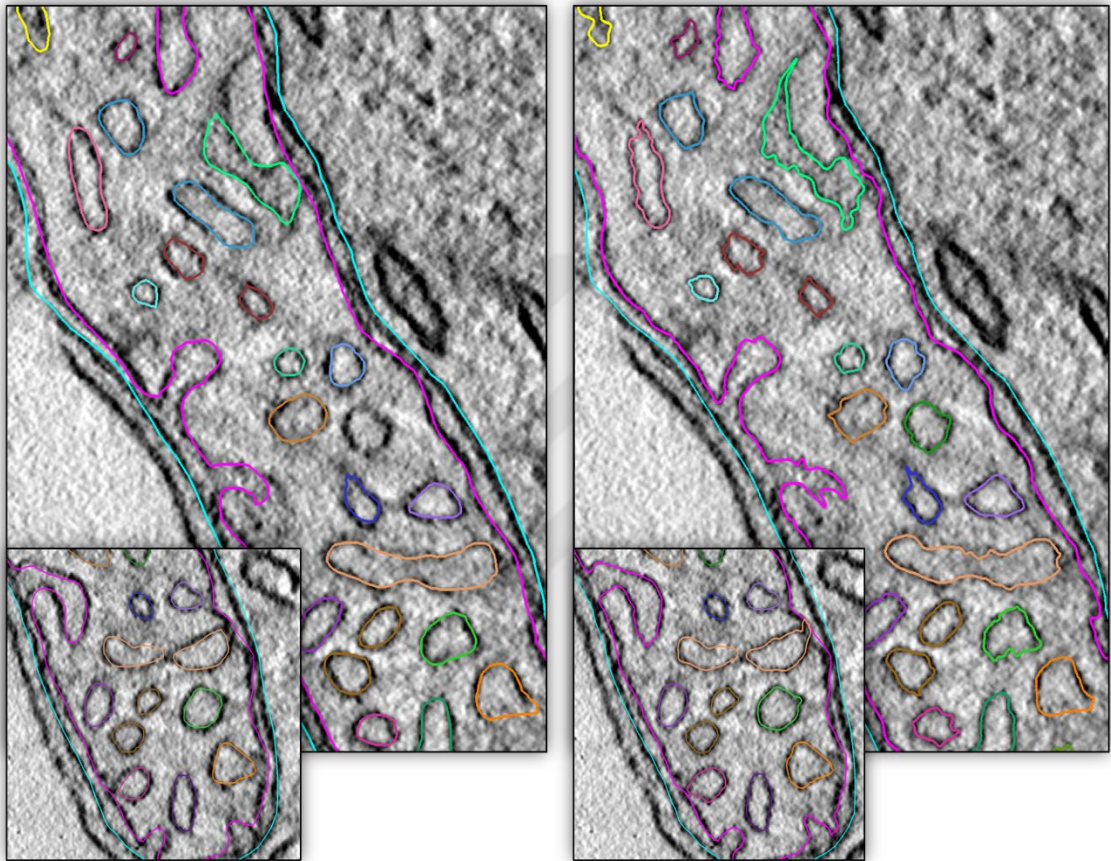
Figure 31. Misaligned markings of the ground truth provided by the National Centre for Microscopy and Imaging Research on the left hand side of the figure. Retraced examples for a better ground truth for 6.22_sub data set.

Because the scope of this study only covers to segment the cristae membranes, the mitochondria boundaries are not marked on data sets bclpb-d, cone_sub, and gap 18 as well as not refined on data set sub_6.22.

Boundaries of cristae membranes are carefully traced by using the IMOD tool (Kremer, Mastronarde, & McIntosh, 1996). The thickness of the boundaries are set to 3 as it is observed that this size fits best to the thickness of cristae boundaries. Although the thickness of the boundaries may vary through the periphery of a single cristae, marking is done with a fixed thickness size because the 3D mode software do not allow to draw a boundary with variable thickness size. In Table 2 below, the numbers for the ground truth marking study can be seen.

Table 2. Summary of the effort spent for manual marking of ground truth.

Data Set No	Data Set Name	# of inner mitochondria marked	# of slices	ROI size	total # of cristae marked / slice	total # of pixels marked	# used slices for training	# used slices for testing
1	Cone_sub	1	6	428x497	~43	90000	2	1
2	6.22_sub	1	8	371x496	~39	88800	2	1
3	bclpb-d_sub	2	2	393x350	~34	18000	1	1
4	gap18_sub	2	2	161x241	~19	8000	0	1

3.1.3. Preparation of the Data for Training and Testing

It is a known fact that the machine learning algorithms perform different depending on the selection of the features that represent the data (Yoshua, Courville, & Vincent, 2013). This requires a labour intensive effort and human intervention and shows the inability of the algorithm to obtain the discriminative information from the row data and make the suitable inferences (Yoshua, Courville, & Vincent, 2013). In order to design systems that are not directly dependent on the data and are easy to apply on data obtained on different conditions, it may be the best way to use the very low level data without applying any feature extraction. From the point of this approach, the input space shall be formed with the intensity values of the pixels in the image since they are the very raw data of an image.

In order to achieve a neural network that can identify all the various patterns in the target space from the given input space data, the training set shall contain sufficient numbers of all possible types of input patterns for all different types of output labels. Multilayer Neural Networks cannot extrapolate beyond the range of the training set so that the training data shall span all the range of the input space (Beale, Hagan, & Demuth, 2014).

There are two different approaches in the literature in order to guarantee that the input patterns in the training set cover all the range of the input space when dealing with images. In the first approach, where the aim is usually to classify the whole image, all the pixels in the image is used for the training of the neural network. In the second approach only a region in the image is considered for the classification purpose and intrinsically the number of the input space is dramatically decreased when compared to the first approach. The neural networks trained with this approach has the ability to classify only the related region or the pixel instead of the whole image so that this approach is more widely used when the aim to segment a region or a part of an image.

If the data to be classified is pixels in the image then the input vector to the network may be formed of features extracted from the related pixel or directly the intensity values of the related pixel and its neighbourhood pixels. One can also use the intensity values of the pixel to be classified and its neighbouring pixels in addition to an extracted set of features as inputs to the neural network. In (Mishchenko, 2009) in addition to 3x3 neighbourhood of the related pixel, Gaussian Smoothed Hessian

eigenvalues are used. In studies like (Suzuki, Horiba, Sugie, & Nanki, 2004), (Middleton & Damper, 2004), (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012) a square input window covering the intensity values of neighbouring pixels is transformed to input vector and used as input for the neural networks.
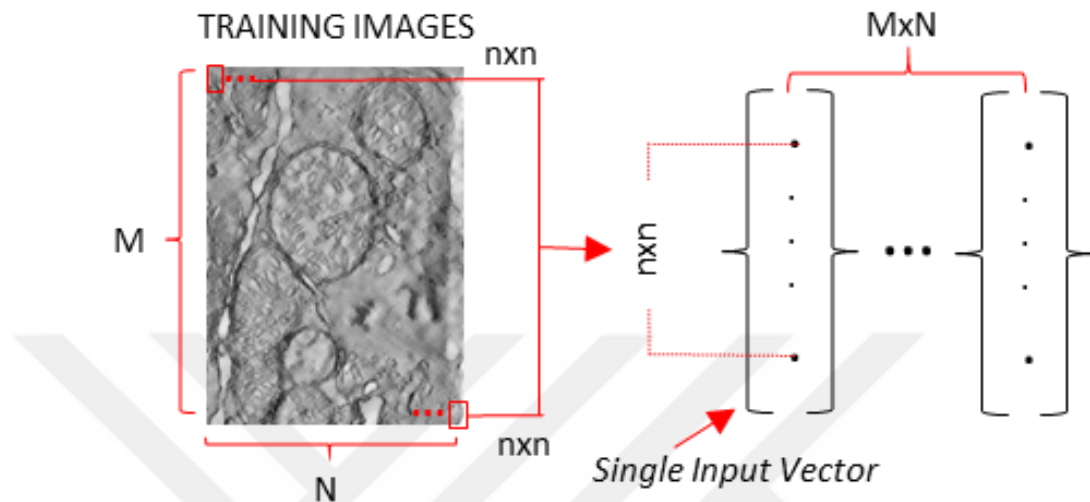
Figure 32. Input Windows and the whole training patterns for an image.

Depending on the classification task, the shape of the input window can be modified. Using a square input window with size of nxn is a common approach. An illustration that explains the procedure of obtaining the training pattern with square input window can be seen in Figure 32. In studies such as (Jurrus, et al., 2010), (Jurrus, et al., 2013) an input stencil window is used, as can be seen in Figure 33, in order to keep the input vector as small as possible while still keeping the generalization performance in hand. This type of neighbourhood sampling technique is used when a large area is needed for the segmentation of related pixel. Different input windows depending on the type of the classification purpose, resolution of the images, and size of the training set has been used with many different artificial neural network models and architectures in the literature. The main objective for selecting different input window sizes and shapes is to optimize the number of input layers in the neural network. Since the training process of the neural networks usually includes iterative error minimization, the size of the input space shall be kept minimum being formed of limited numbers of carefully selected input elements. This input window shall provide enough contextual information about the pattern of the pixel to be classified. Of course this contextual information approach for training neural networks is useful only when the local image features are sufficient for the segmentation purposes.
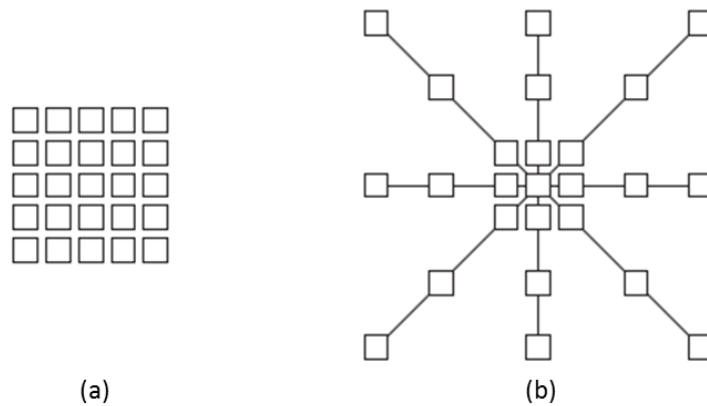
Figure 33. Input stencil versus square path. (a); a square input window with 25 input features (which are the intensity values of the pixel). (b); an example of stencil input window which contains 25 input features as (a) but cover a larger neighbourhood area around the centre pixel. (Image taken from *(Jurrus, et al., 2010)*.

As it is a common and convenient approach when working with high resolution image data, the input vector for training the neural networks is limited to a considerably small numbers of pixels covering the neighbourhood of size w (window size of the input vector) of the centre pixel to be classified. Otherwise the use of all pixels of the image in a single input would be impractical due to the huge computation power and massive numbers of training images needed. For instance, a 512x512 image has a total of 262144 pixels. If we tend to train the network with an input vector of size 262144 then we would have a very large NN to train which gives 262144 weights plus 1 bias for each neuron in the first hidden layer. This consequence implies that a very large training set and huge computational resources would be needed. In practice such an approach is impractical. Because a large w (it is the whole image when w equals the width of the image) results in much bigger networks, which take longer to train and, at least in theory, require larger amounts of training data to retain their generalization ability (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012). In addition to computational power problem; the increased input window size can cause modelling the noise. In massive training of ANNs, the size of the input window determines the number of features used for modelling the desired function. The feature size shall be big enough to let the ANN model the function but more than this will tend to memorize the data as mentioned in (Lawrance, Giles, & Tsoi, 1996). This problem is more commonly known as overfitting.

In this study segmentation of cristae boundaries by means of ANN will be accomplished by means of massive training. Massive training with many input windows are introduced in many works like; (Suzuki, Amato III, Li, Sone, & Doi, Massive training artificial neural network (MTANN) for reduction of false positives in computerized detection of lung nodules in low-dose computed tomography, 2003), (Middleton & Damper, 2004), (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012), (Jurrus, et al., 2010) and many others.

In massive training hundreds of thousands of input windows are used to represent all the useful patterns in a data set in order to let the network model the desired function. Each input window, in which enough information to be able to segment the cristae

40

boundaries shall be contained, represents a small region of the whole image. The information contained by the input window is determined by the size of the input window. The more the size is increased the more information is carried out. The view of the cristae boundaries inside different sized windows can be seen in Figure 34.
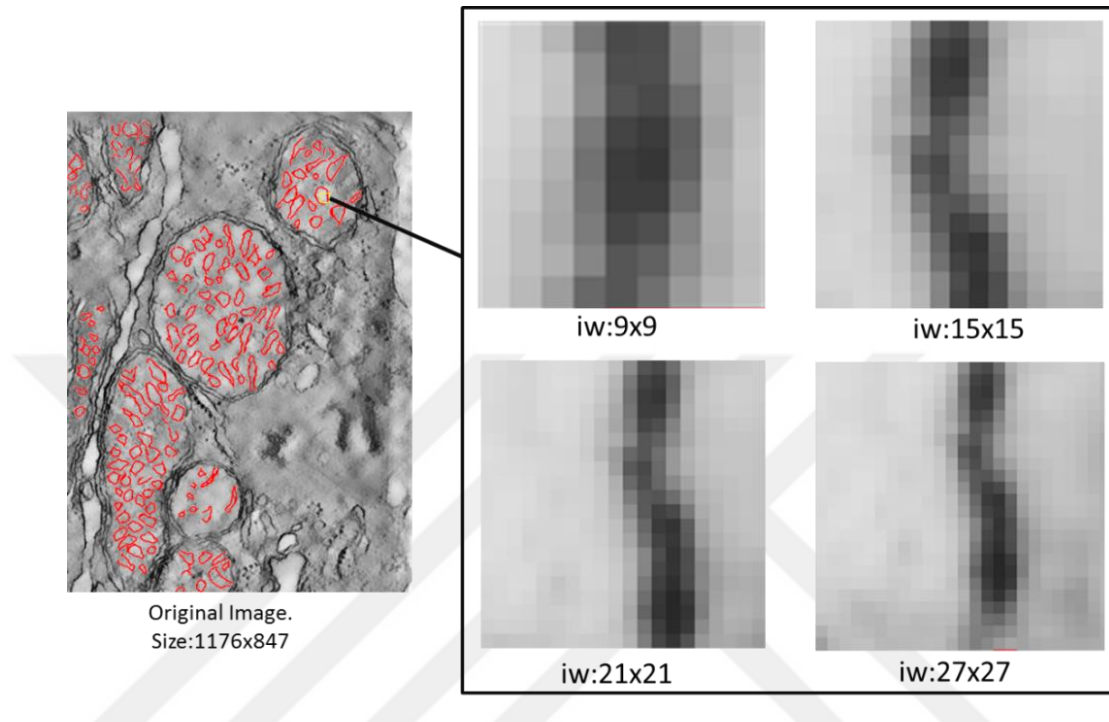
Figure 34. Images from 'cone_sub' date set with different size of input windows.

In Figure 34, one can observe that although it seems possible to identify the boundary even with the smallest input window, it is impossible to differentiate whether it is mitochondria and cristae boundary. In order to differentiate whether the boundary belongs to cristae or not, the input window shall cover all the cristae. This is possible only by increasing the size of the input window. In (Mumcuoglu, et al., 2012) the mitochondria boundary is effectively segmented. Since the aim of this study is to only segment the cristae boundaries, the problem arose from the difficulty of discriminating the mitochondria and the cristae boundaries can be ignored.

But the size of the input window still effects the performance of the trained neural networks when the intercellular structures in mitochondrion except cristae membranes taken into consideration. This problem is described in Figure 35. In Figure 35-(a) a small portion of a section from Sub_6.22 data set is seen. On the left hand side, a crista boundary is seen with different size input windows while on the right views of different input windows belonging to a non-membrane area can be seen. Figure 35 below reveals that, it is impossible to distinguish the cristae membrane area from non-membrane area when the input window size is 9x9. As the input window size increases it becomes clear that the images c, e and g belong to non-membrane area in the mitochondria.
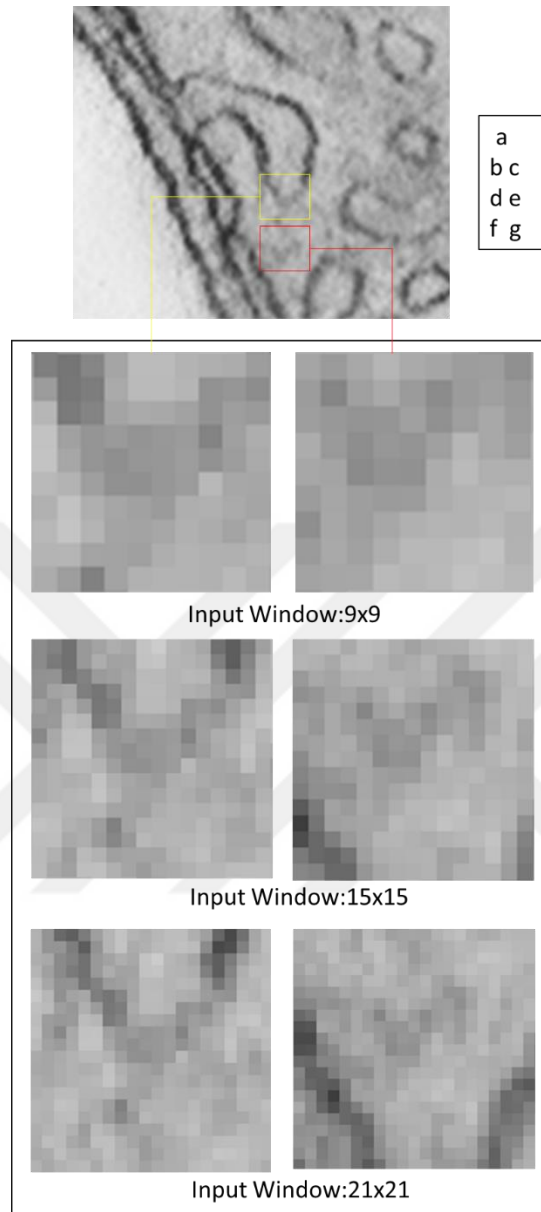
Figure 35. The views of membrane (on the left) and non-membrane (on the right) areas with different size input windows.

Since the scope of this study is to segment the cristae boundaries inside the mitochondria, only the area remaining within the mitochondria inner membranes shall be used for training and testing. Because the slices in the data set includes many other cell structures like nucleus, endoplasmic reticulum etc. besides mitochondria, the slices to be used for training and testing are cropped as seen in Figure 36.
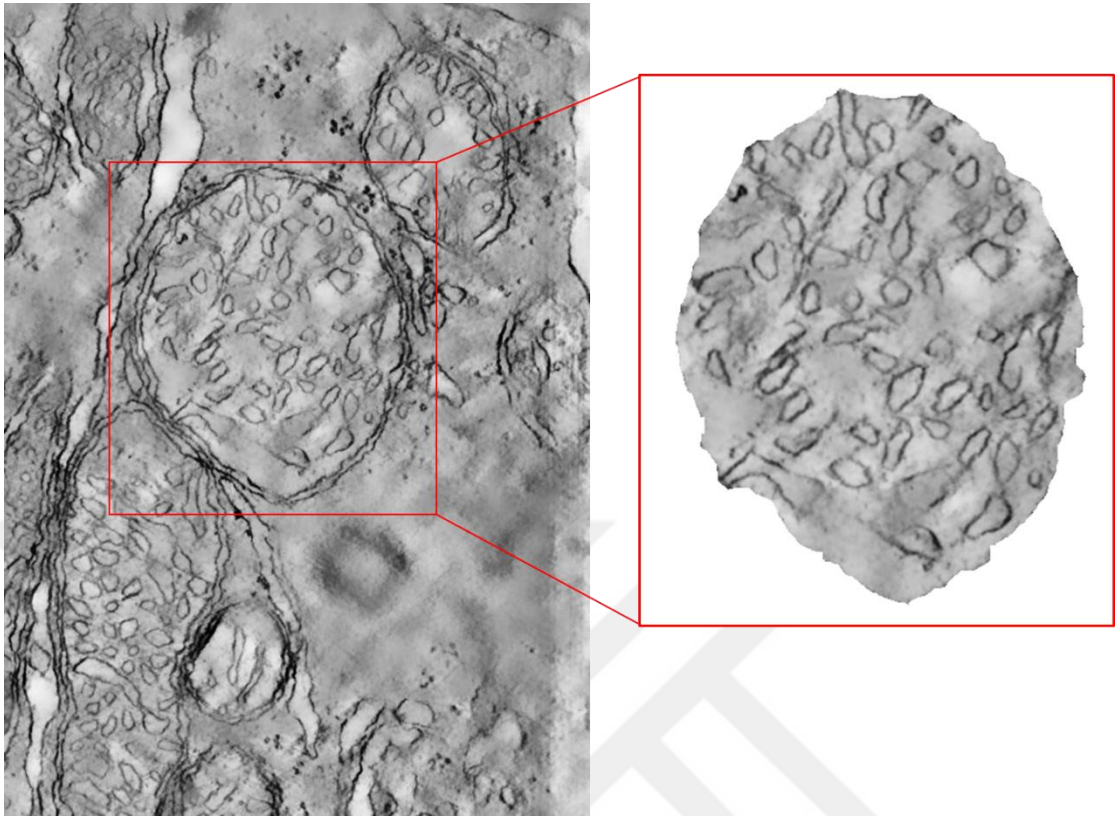
Figure 36. Cropped image of the mitochondria. Each slice to be used in training of the networks are cropped and cleaned.

There exists a huge number of negative examples when compared to positive ones since the positive examples in a slice are only the membrane boundaries. If all the patterns in the image is used, the network will see many times more negative examples than positives during the training. This turns out that the trained network will be prone to classify the massive number of pixels as negative so that the false negative rate will be unacceptably high. In order to overcome this issue, the ratio between the number of positive and negative examples are arranged by removing the determined numbers of randomly selected negative examples from the training set.

The pixels falling into the membrane – non membrane transition area are difficult to label since they represent the features between membrane and non-membrane structure. These pixels may decrease the training performance of the networks. In order to determine the effect of this phenomenon different training sets both including and excluding (without the pixels adjacent to membrane boundaries) the pixels in this transition zone are prepared and used for training the networks. The networks trained with the sets excluding these pixels are reported in "Results" Chapter.

3.1.4. Pre-processing

EMT images may contain some artefacts including; extremely dark or bright intensity values due to the imaging technique, limitation of the equipment, or weak or very strong membrane edges (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016). In addition to the artefacts of the EMT, the technique may produce noisy images so that

noise removal is recommended when working with electron microscopy data (Bazan, Miller, & Blomgren, 2009) (Mumcuoglu, et al., 2012) (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016). These characteristics on the input image may cause undesirable or insufficient findings on the output segmentation results.

As mentioned before the method proposed in this study aims to operate on different data sets which are obtained from different types of cells. This turns out that each data set has different pixel size which originates the problem of parameter tuning since the features used for segmentation of the membranes are not invariant to the scale and size of the viewed and analysed mitochondria in the image.

 In order to overcome the possible effects of these issues on the results, the pre-processed images obtained with the application of the procedure proposed in the study by Taşel et al. (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016) is used. The original and the pre-processed image can be seen in Figure 37. The pre-processed images are provided by Dr. Serdar Taşel. At the first stage of the proposed method auto-contrast adjustment is applied on the data set. Direct normalization cannot be applied since it can cause substantial degradation in image contrast due to extreme changes in image intensity levels which may arise due to gold markers utilized or X-rays and other flaws in CCD sensors (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016). Auto-contrast adjustment assumes that there exist extreme points in the intensity histogram so that the algorithm can re-normalize the intensity distribution to a certain range (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016). Subsequent to auto-contrast adjustment, images are resampled so that all the thickness of cristae membrane in different data sets are set to a fixed pixel size. The typical mitochondria membrane thickness ranges from 4nm to 6nm (Rog, et al., 2009) and the resampling is achieved by means of interpolating the image to 2 nm pixel size which prevents any data loss for the mitochondria membrane during resampling. In the end of the pre-processing, the noise removal operation is applied by means of bilateral smoothing technique (Tomasi & Manduchi, 1998, January) which preserves edge while removing the noise in the image. This technique is also useful for removing the non-membrane-like structures while conserving the membranes (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016). In Figure 37, it is seen that the application of bilateral smoothing makes image clearer while preserving the membranes with very limited data loss.
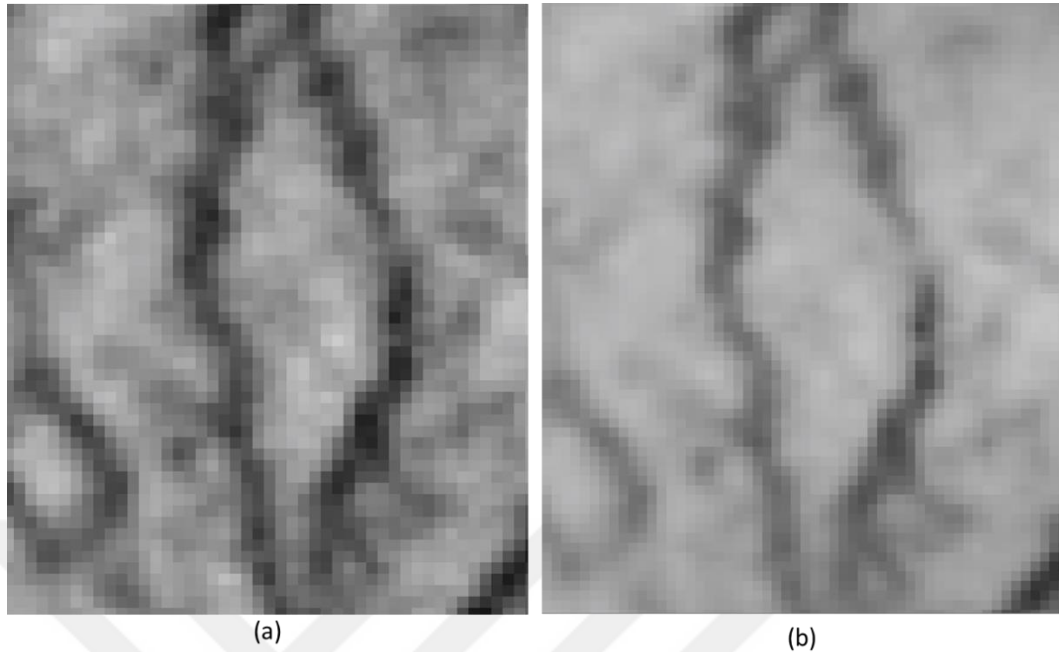
Figure 37. The pre-processed image processed by the method proposed in *(Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016)*

3.2. Segmentation in Two-Dimensions

In many studies (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016) (Mumcuoglu, et al., 2012) (Martinez-Sanchez, Garcia, & Fernandez, 2011) it has been attempted to segment the inner and the outer membrane of mitochondria. Although the mitochondria boundaries have successfully segmented in these studies, there exists limited number of studies for the problem of segmentation of the cristae membranes.

In this study we aim to segment only the inside region of the mitochondria excluding the boundary membranes. These include the cristae and other intercellular structures inside mitochondria.

As a simple approach we first tested 'Canny' edge detection method (Canny, 1986) which is known to be one of the best methods for edge detection with connectivity. The Figure below shows the results of the 'Canny' method. As seen from the Figure that although the method is capable of detecting many of the edges, still it shows considerably low performance on detecting weak boundary edges and false positive rate.

Figure 38. Result of Canny edge detection method.

The proposed method for the segmentation of cristae membrane from the inner mitochondria images includes two main stages after the pre-processing step is accomplished. In the first stage, the image transformed to the input vectors obtained as described in Section 3.1.3 is used for training a neural network. The output of the network produces the probability values whether the centring pixel of the input window from which the input vector is obtained belongs to a cristae membrane or not. In the following stage the result of the neural networks are refined by means of a special method named as "Directional Hessian Ridge Growing". The flowchart of the proposed method can be seen in Figure 39.
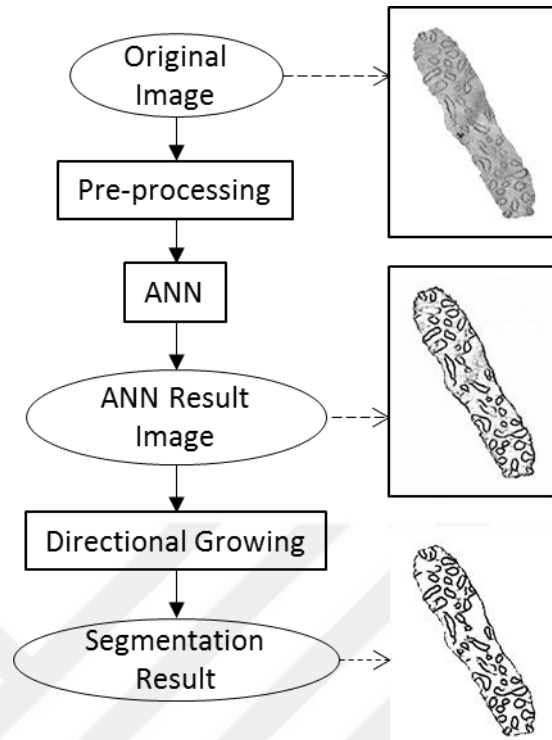
Figure 39. Flowchart of the proposed method.

Although the segmentation of cristae in hundreds of slices of EMT images has a critical importance in order to reveal the role of the cristae in many disease within its morphology as mentioned Section 1, there exist a limited number of studies in the literature for the automatic segmentation of the cristae membrane in the mitochondria. In this study, only the interior region of the mitochondria is considered as the analysing data leaning on the fact that the segmentation of boundary membrane of the mitochondria has successfully been accomplished by the studies like (Mumcuoglu, et al., 2012) (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016). So the images on which the proposed method applied are the images that includes only the region that is bounded by the inner boundary membrane of the mitochondria where the inner boundary is excluded. The remaining section of the images are cropped out.

3.2.1. Segmentation Using Classical Feed-Forward Neural Networks

In order to start to train the neural network once the data is ready, an architecture that let the network to represent all the input patterns in the data while not overfitting the data shall be selected. The parameters that define the architecture of the network are; the number of hidden layers, the size of the input layer, the number of hidden nodes in each hidden layer, and the size of the output layer. A fully connected feedforward network example can be seen in Figure 40. The size of the input layer is defined by the size of the input vectors as seen in Figure 40. This turns out that the selection of input window stencil which is described in Section 3.1.3 defines the size of the input layer. We aim to decide whether a pixel belongs to a membrane boundary or not, so that there exist only one node in the output layer.

47
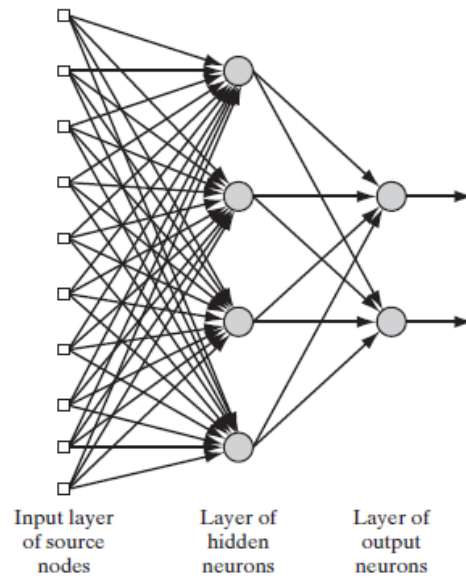
Figure 40. Illustration of a feedforward neural network. (Image taken from *(Haykin, 2009)*)

The remaining parameters in order to define the architecture of the network are the number of hidden layers and hidden nodes in the hidden layers. The determination of the number of hidden layers and the number of hidden nodes in the hidden layers is not a straightforward task. Even though there exist many proposed rule-of-thumbs, it a very common approach to determine them by trial and error. Starting with a very small network may result networks that do not have enough power to represent all useful patterns in the input. This phenomenon is known as high bias. On the contrary, if a very large network is used, then the network may be trained to model the noise and overfit the data. This phenomenon is known as high variance (Lai, 2015). As a common approach many different networks with varying numbers of hidden layers and hidden nodes are trained.

Another issue that characterizes the performance of the networks is the choice of the activation functions in the neurons. For pattern recognition problems sigmoid activation functions are often used (Beale, Hagan, & Demuth, 2014). Since we want to obtain outputs falling between 0 and 1, sigmoid activation function is used in both the hidden and the output layers.

3.2.2. Training

The training phase of a neural network can be defined as the process of estimating the values of adjustable parameters which define the characteristics of the classification model depending on the data set (Bishop, 2006, p. 5).

In the training phase of the neural network, the whole data set is randomly divided into three subsets: training set, validation set, and the test set. The training set which is %70 of the whole set is used to update the weights and biases as well as computing the gradient of the error function. The validation set which is %15 of the whole set is used to decide the moment that the network begins to model the noise in the training set

48

which is called overfitting. And finally the test set which is %15 of the whole set is used to represent the present performance of the current trained network and compare with different networks. The receiver operating characteristic (ROC) curves and the confusion matrix which are useful tools in order to measure the performance of the network and select the best one are formed with the test data.

Before the division of the training set into three different parts, a normalization operation is applied to the input vectors, which comprises the intensity values of input window or patch, so that the input values are mapped in the range between -1 and 1 with the formula defined in Equation (29).

$$y = \frac{(y_{max} - y_{min}) * (x - x_{min})}{(x_{max} - x_{min})} + y_{min} \qquad (29)$$

Where y is the normalized intensity value, $y_{max} = 1$, $y_{min} = -1$, x_{min} and x_{max} are the minimum and maximum intensity values in the image respectively.

The reason for the normalization operation on the input patterns lay behind the saturation behaviour of the sigmoid function. The sigmoid functions used in the activation of each neuron are saturated by approaching the limit values of the functions when the net output of the neuron is greater than a specific value. For instance, let the sigmoid function used defined as;

$$y = \frac{1}{1 + e^{-x}} \qquad (30)$$

Where $f(x)$ is the output of a neuron and x is the sum of all inputs multiplied by the weights plus the bias. If the summed output of the neuron is 4 (four) then the output of the activation function approach to the upper limit of the function (From the Equation (30) for $x = 4$, $f(x) = 0.9820$). This means that the gradient will become too small so that the progress of the learning will be limited. In order to avoid the phenomenon explained below either the weights shall be extremely very small or the input shall be normalized in a relatively small range.

Before starting the training, the parameters to be adjusted which are weights and bias shall be initialized. As it is a very common approach the initialization is made by means of randomly selected small values. The selection of random values for weights and bias guarantees the avoidance of the problems due to symmetries in the network and the selection of small values provides that the sigmoid activation functions are not driven into the saturation regions (Bishop, 2006, p. 261).

As the initial weights directly define the course of the training (minimum error rate achieved, training time, total iterations) and the performance of the resultant network, it is a common approach to train the network for many times and to select the best network by comparing their performances by the use of the data set which is not used for training and reserved for independent testing (Bishop, 2006).

Perrone and Cooper (Perrone & Cooper, 1993) proposed a method which may result networks with improved performance as combining these trained networks to form a committee. In this study the best performing network is selected amongst some trained networks. It shall be noticed that selecting a single network amongst many trained ones

has an disadvantage along with wasting the effort paid for training the remaining networks: the network which has the best performance on one single selected test set may not be the best performing network for new data sets because the test set selected possess random components due to the noise (Bishop, 2006, p. 364). In our study, each network (with a different architecture) is trained for 10 times. Once each training is completed with one of the stopping criteria defined below, the trained network is tested on a completely different image (using its ground truth). Input vectors whose sizes depend on the shape of the input stencil are obtained for each centring pixel and its neighbourhood from the test image. Then the input patterns are passed through the network and the results are compared with the ground truth image by the help of metrics like true positive rate (TPR) which is also called recall, true negative rate (FPR) which is also called specificity, precision and a kind of effectivity metric defined in (Middleton & Damper, 2004). These metrics are defined in Equation (31). Although many metrics are computed, TPR metric is used for the selection of the best neural network from 10 trained networks.

$$
\begin{aligned}
TPR(recall) &= \frac{\sum positive\ examples\ that\ are\ correctly\ labeled}{\sum positive\ examples} \\
TNR(\text{specificity}) &= \frac{\sum negative\ examples\ that\ are\ correctly\ labeled}{\sum negative\ examples} \\
Precision &= \frac{\sum positive\ examples\ that\ are\ correctly\ labeled}{\sum examples\ labeled\ as\ positive} \\
Effectiveness &= 1 - \frac{Precision * TPR}{(1 - \alpha) * Precision + \alpha * TPR}
\end{aligned}
\tag{31}
$$

In the training set there is an orders of magnitude higher number of negative examples as compared with the positive examples since only cristae boundaries are considered as positives. This naturally causes an important issue that TPR, FPR and precision measures cannot effectively represent the performance of the networks trained. The denominators of FPR and TPR metric are significantly different and consequently tenderness of these two measures against changes are not comparable. Thus a small change in the false positive error rate is relatively more important than a comparable change in the false negative error rate, leading to difficulties in interpretation (Middleton & Damper, 2004).

For the learning of the parameters in the network, the Levenberg-Marquardt [(Levenberg, 1944) (Marquardt, 1963)] optimization algorithm which is also known as damped-least-squares is used instead of standard gradient descent algorithm. Levenberg-Marquardt method is a compromise of the Newton's method and gradient descent methods (Haykin, 2009) as described in Section 2.2.

It has not been proven that the backpropagation algorithm can converge, likewise no golden standard stopping criteria for its operation has been proposed (Haykin, 2009, p. 139). But there exist some criteria that are used for the termination of training process such as checking the rate of change of the average squared error per epoch. In this approach, if the derivative of the average squared error with respect to epoch number is sufficiently small, then the training is stopped (Haykin, 2009, p. 139). This approach may cause to premature stopping if the error function temporarily stick

around a local minima (Bishop, 2006, p. 262). Also the error value itself can be used as the training is terminated when it falls below a specified threshold value. But this time the specified threshold may never be reached so that some additional limits like processing time or maximum iterations may be required (Bishop, 2006, p. 262).

In this study, in addition to "maximum iterations reached" and "gradient value of error function" checks for the termination, validation stop criterion is used. As the training progress, the validation set is passed through the present trained network in order to validate the model on another data which is different from one used for parameter adjustment (Haykin, 2009, p. 171). During the training process the training set error decreases as well as the validation set error. With the use of validation stop criterion, the training is stopped when the validation set error starts to increase which is the moment the network starts to overfit the training data. In order to be sure that overfitting is started the validation error shall be increasing simultaneously in each new iteration. If the validation error increases for 6 consecutive epochs which is empirically determined in this study then the training is stopped and the network (weights and biases) is saved at the minimum of the validation set error which is 6 epochs before. This is called the validation stop criteria in the neural networks literature. The graphical explanation of the training and validation error formation can be seen in 41. From Figure 41 it can be observed that although the training error decreases throughout the process, the error computed with the validation set data which is not used for updating the weights starts to increase at moment where the network starts to lose its generalization ability.
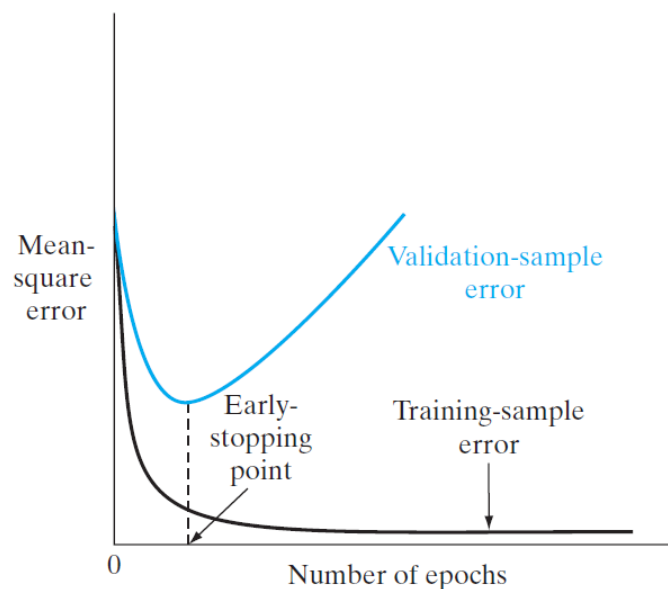
Figure 41. The error formation during training process of a neural network. The dashed line shows the moment that the overfitting begins. (Image taken from *(Haykin, 2009, p. 174)*)

Although there are studies in the literature which trains the networks with a fixed number of epochs, in our study, fixed epoch numbers approach is not employed because if the neural network converges the desired performance much far before the pre-set number of epochs then the network will start to model the noise in the data set because the training is not stopped. By the use of decreasing validation performance

51

check approach, training is stopped just when the neural network starts to memorize the data.

As mentioned before, in the weight adjustment stage of the backpropagation algorithm the error function is used as a cost function to be minimized. There exists two main approaches in the literature for training of the synaptic weights during this stage of the backpropagation algorithm. In the first approach, the weights are updated just after each pattern is shown to the network. This type of learning is called online learning.

In the second approach the adjustments of the weights and bias are performed after all the patterns in the data set (which is called one epoch of the training (Haykin, 2009, p. 127)) are shown to the network. This type of learning is called batch learning (Bishop, 2006, p. 146). In this approach the learning is progressed through an epoch by epoch basis. Since the adjustments of the weights are done after each epoch is completed, the error function is defined as the average error function, $\mathcal{E}_{av}$, as defined in Equation (32).

$$\mathcal{E}_{av}(N) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{E}(n) \tag{32}$$

where N is the number of patterns in the training set and $\mathcal{E}(n)$ is the error of nth pattern.

Due to its advantages of accurate estimation of the gradient vector and parallelization of the learning process (Haykin, 2009, p. 128), the batch learning approach is used in this study.

The learning rate parameter in the learning algorithms controls the level of the adjustment applied to the weight at each iteration when standard gradient descent algorithm is used. If it is selected too small then the minimization of the error function will be slightly slow while the training may not converge to a minimum error, for the opposite case (if it is selected too large) divergent oscillations may occur (Bishop, 2006). The Levenberg-Marquardt algorithm is a composed version of the gradient descent and Gauss-Newton algorithm, no learning rate is used in steps of the iteration.

The flowchart of the training process applied for the one hidden layered networks can be seen in Figure 42. The neural network toolbox of the MATLAB® R2016a software is used since it offers wide range of access and control to the parameter of the networks including the ones explained above.
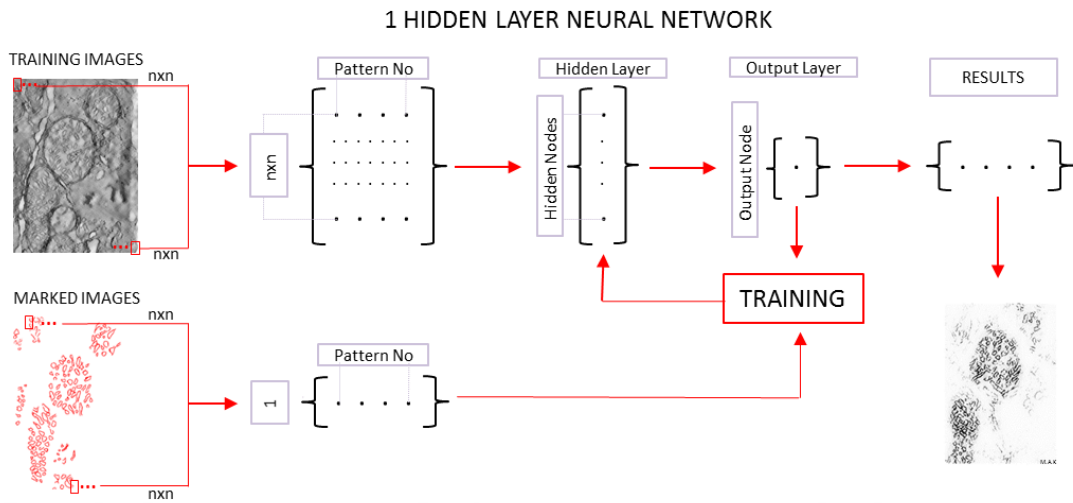
Figure 42. Flowchart of the training process. Each pattern is comprised of the centring pixel with its neighbourhood. The pairing target values of the patterns form a matrix with size of 1xPattern_No.

3.3. Segmentation in Three-Dimensions

Although three dimensional electron microscopy datasets are used in this study, the useful information that lays in the third direction (the thickness of the cell) of the images are not benefitted in the first section of the study.

Membrane boundaries in the data set may appear disconnected when looking at a single slice of image. But one may observe that these disconnected boundaries are connected when the following sections are analysed. While it is impossible to specify the boundary on only one section of the image, it may be easier when the following sections in the z direction of the 3D image are considered because the boundary of the membrane usually becomes apparent in the neighbouring sections. This phenomenon can be observed in Figure 43.
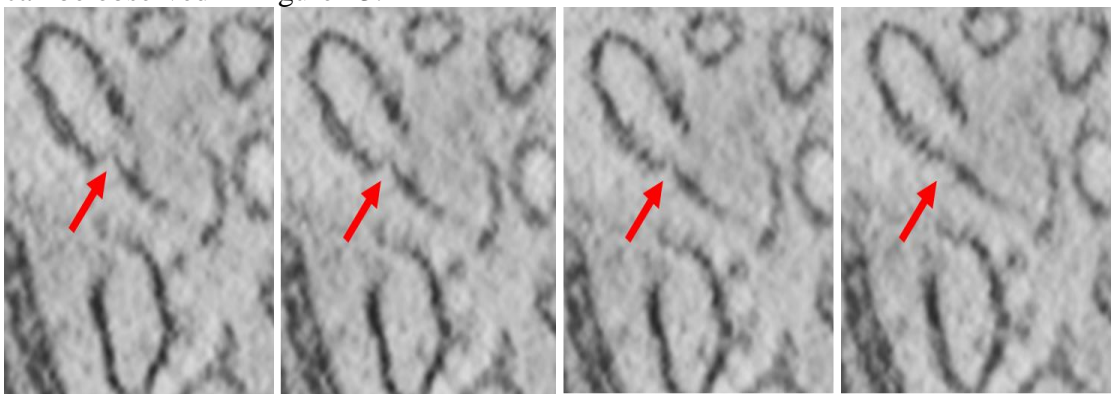
Figure 43. Appearing boundaries in 3D sections. In the very left image the indicated region of the boundary is disconnected while it becomes apparent in following sections.

In addition to the phenomenon indicated in Figure 43, there also exist many boundary-like structures that do not belong to a membrane in the image. These structures usually appear and disappear one or two sections. When only one slice of image is

53

investigated, these structures may be classified as a part of boundary of a crista. The information that these structures are not continuous through the thickness of the cell can only be ingested to the network with the use of 3D input patterns so that these deceptive artefacts can be truly classified. In Figure 44 below a disappearing non-membrane structure can be seen. On the very left image, the indicated structure can be classified as a part of the crista. But the following section reveal that it is a not.
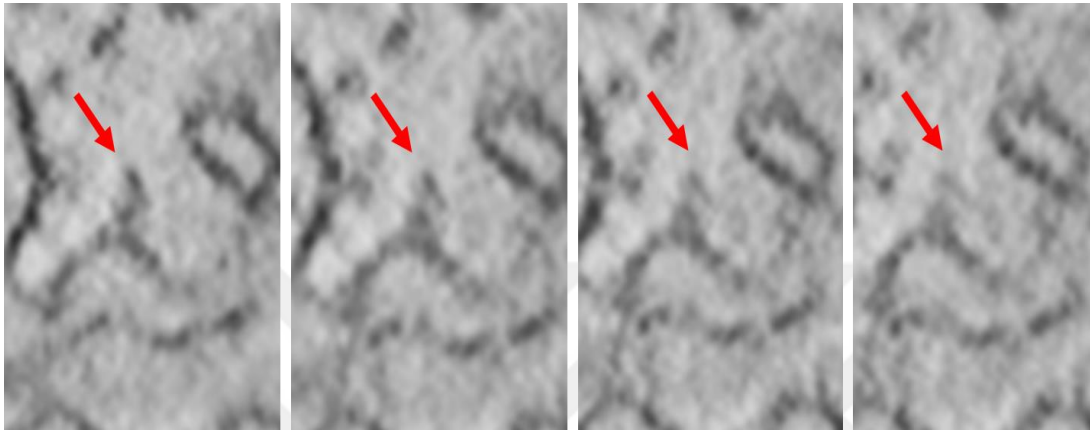
Figure 44. Discrimination of non-membrane structures in 3D sections. For many cases non-membrane structures can only be truly classified by the observation of the thickness of the cell.

Since the aim lays beneath the 3D extension of the method is to use the information which is hidden in the neighbouring images, simply, the input stencil or window used in the 2D training section is enlarged to 3D by means of adding pixel intensities of neighbouring pixels of the centring pixel in the following images through z direction. By this the training is accomplished including the 3D information so that the performance of the algorithm is enhanced. An input stencil example with the depth of three can be seen in Figure 45-b. On the left side (Figure 45-a) an input stencil used for extracting input features from a single 2D image can be seen while on the right side the same input stencil is tripled through z axis. With the use of this 3D input stencil, the intensity values of the neighbouring pixels including the centring image and its neighbouring pixel can be obtained.
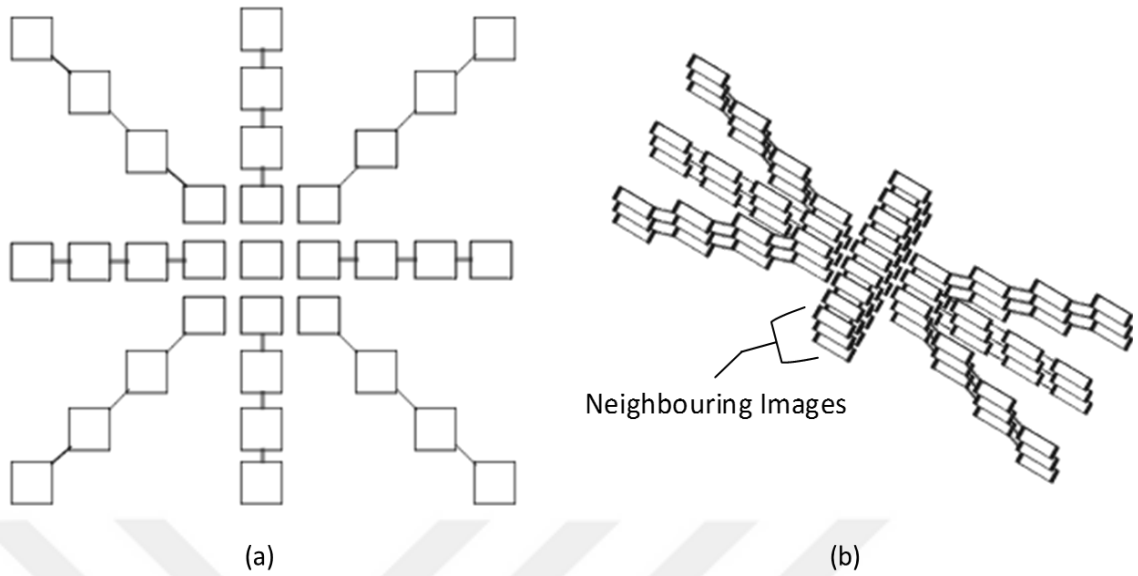
Figure 45. Examples of 2D and 3D input stencils. (a) shows the standard input stencil used for 2D training (on one section). (b) Shows the same input stencil but extended to 3D so that the training section thickness becomes three.

3.4. Directional Hessian Ridge Growing Method

In the first stage of the proposed method, vast majority of cristae boundaries are segmented except areas where the gradient of the membrane is relatively small (which means the intensity value of the boundaries are almost same as the non-membrane areas) when compared to neighbouring cell fluid region of related boundary. In these zones, the neural networks cannot produce significantly different intensity values to discriminate the membrane from darker cell fluid regions. In these areas the segmentation cannot be accomplished by means of direct thresholding methods (of the output stage) and consequently disconnected boundaries are observed in the results.
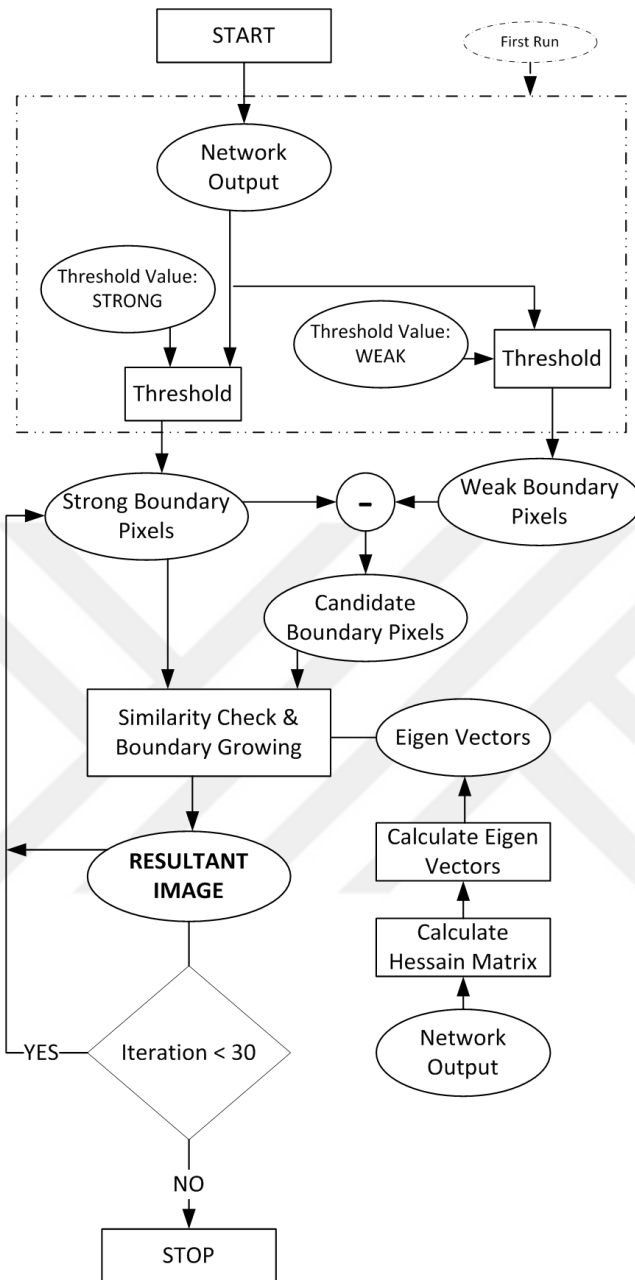
Figure 46. The flowchart of the iterative directional hessian growing method

In order to enhance the performance of the results obtained from neural networks, a method which grows the previously segmented membranes and attempts to connect the disconnected boundaries is proposed. The proposed method is inspired by the Canny (Canny, 1986) method in which edge growing through a hysteresis thresholding is applied. The proposed method used the advantage of hysteresis thresholding in order to determine the real and candidate ridges separately just as Canny did. The flowchart of the method can be seen in Figure 46 above. Below the examples from the sub outputs of the algorithm which are numbered in Figure 46 above can be seen in Figure 47-b-c-d. Image in Figure 47-b form the base for the resultant image. Pixels in Figure 47-d are added to resultant image if they are located in the search window are of the pixels in b and satisfy the rules explained in the remainder of this section.
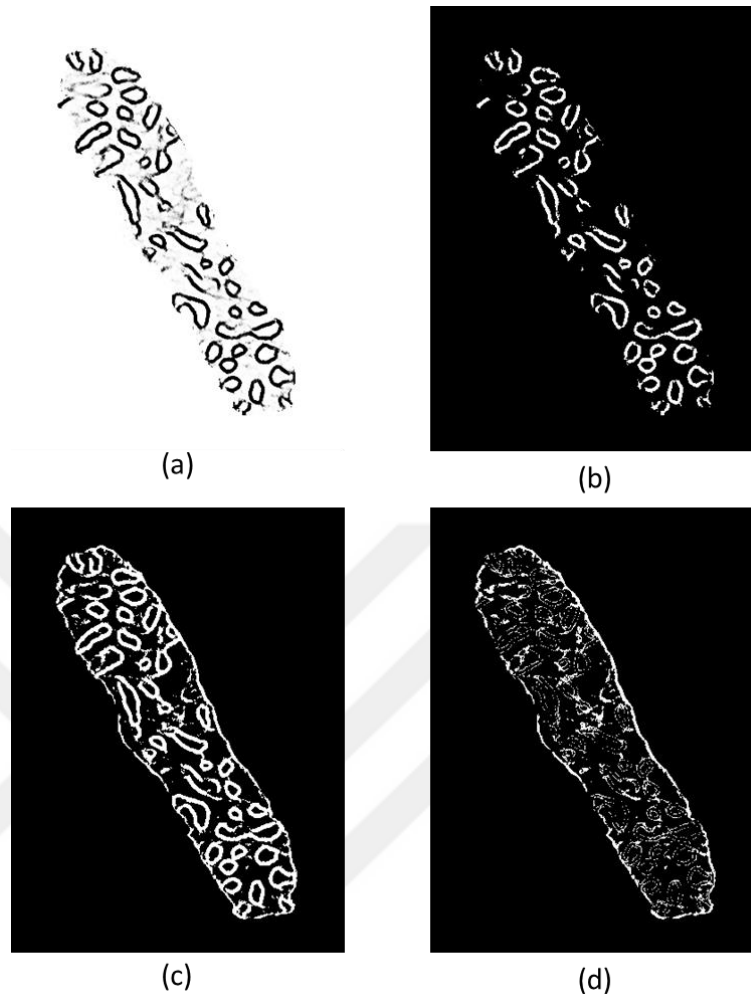
Figure 47. Sub-outputs of the directional hessian growing method. (a) is the output image of the network, (b) is the strong membrane boundary image that forms the base for the growing operation, (c) is the weak boundary pixels that are used to obtain the candidate pixels in (d).

The proposed method takes advantage of two basic assumptions about the membranes that were successfully segmented in the previous stage depending on the fact that the intensity values shall remain similar through the membrane connectivity direction and increase through both orthogonal (opposite to each other) directions:

1. However the intensity value produced by the network is almost similar or lower when compared to many non-membrane areas in the cell, if the analysed pixel is part of a cristae membrane, still, it shall have a smaller intensity (where a %100 possibility of belonging a membrane boundary is represented with an intensity value of 0) value than the vast majority of the non-membrane areas and possesses visually observable valley-like intensity transition zones.

2. If the observed pixel in a weak valley-like transition zone is a membrane structure, then the curvature direction of the zone around that pixel shall be similar to the direction of the curvature of neighbouring strongly classified as membrane structures which fall into strong valley-like transition zones.

In order to check the first assumption, a double thresholding operation is applied so that the continuous result of the network is divided into two binary images of which one contains the pixels that strongly belong to a membrane structure. The pixels belonging to strong boundaries are classified as membrane boundary pixels at this stage, and this decision is not changed in the following stages of the algorithm. This is accomplished by means of applying a basic thresholding as in Equation (33).

$$I_{strong}(x,y) = \begin{cases} 1, & I(x,y) < Threshold_{Low} \\ 0, & \text{otherwise} \end{cases} \quad (33)$$

In Equation (33) I_{strong} is a binary image where only the real or most likely membrane structures exists. In order to determine the optimal $Threshold_{Low}$ value; various different threshold values are applied to resultant images of networks and an optimal threshold value which maximizes the TPR while minimizing the FPR is empirically determined.

The second binary image obtained in this stage (double thresholding) contains the weak boundary pixels that were eliminated in the first thresholding stage but still are candidates to be a part of a membrane boundary. This image is used in connecting the disconnected membrane structures. In order to obtain the weak candidate membrane pixels, a higher threshold (for selecting both the strong and the weak pixels) is applied:

$$I_{all}(x,y) = \begin{cases} 1, & I(x,y) < Threshold_{High} \\ 0, & \text{otherwise} \end{cases} \quad (34)$$

where;

$$I_{all}(x,y) = I_{strong}(x,y) + I_{weak}(x,y) \quad (35)$$

In Equation (35), $I_{weak}(x,y)$ is the image that contains weak candidate membrane pixels. In order to discriminate the weak membrane structures from non-membrane structures, the second assumption is assured by using the information that the eigenvectors of the Hessian Matrix provide. In this method; each pixel marked as 1 in the I_{weak} binary image is marked as membrane if both of these two conditions are satisfied;

1. There exists at least one $I_{strong}(i,j) = 1$ within a neighbouring area defined by the input window with size nxn.
2. The similarity metric (S) of the eigenvector of $I_{weak}(x,y)$ and $I_{strong}(i,j)$ is greater than *similarity threshold* (which is set empirically) for at least one of the $I_{strong}(i,j)$ in the nxn neighbouring window around the weak pixel location ($I_{weak}(x,y)$).

The similarity metric in the second condition is defined as:

$$similarity\ (S) = \frac{e_{strong}{}' * e_{weak}}{|e_{strong}| * |e_{weak}|} \qquad (36)$$

where; e_{strong} is the eigenvector that corresponds to the greatest eigenvalue of the Hessian matrix (H) which is shown in Equation (37).

$$H = \begin{vmatrix} G_{xx} & G_{xy} \\ G_{yx} & G_{yy} \end{vmatrix} \qquad (37)$$

The members of the Hessian matrix shown in Equation (37) are the second order partial derivatives of I(i, j) in the image where i and j are the row and column of the image pixel. The gradient of the pixel I(i, j) through directions x and y are calculated as in Equation (38);

$$G_x(i,j) = I(i, j + 1) - I(i, j - 1)$$
$$G_y(i,j) = I(i + 1, j) - I(i - 1, j) \qquad (38)$$

The second order derivatives are calculated as in Equation (39):

$$G_{xx}(i,j) = G_x(i, j + 1) - G_x(i, j - 1)$$
$$G_{yy}(i,j) = G_y(i + 1, j) - G_y(i - 1, j) \qquad (39)$$

And the diagonal elements G_{xy} and G_{yx} is calculated as in Equation (40).

$$G_{xy}(i,j) = G_{yx}(i,j) = G_x(i + 1, j) - G_x(i - 1, j) \qquad (40)$$

The process explained above is iterated until the desired performance is achieved as explained in Figure 46 above.

CHAPTER 4

RESULTS

Introduction

Although the data obtained from the Cell Centered Database (CCDB) is a huge data set including total of 249 slices (for "cone, sub22, and bclpb-d" sets), only a small number of slices could be used because of the restrictions rising from the limited power of processor and memory used in this study. Two slices from the "6_22-sub" data set, two slices from the "cone_sub" data set, and one slice from "bclpb-d" data set is selected and used for training of the neural networks. The trained networks are tested with slices that are not used during the training phase. The 2D networks are trained only with one single slice of training image while the 3D networks are trained with stacks of neighbouring slices through z direction (thickness direction), where the segmentation is done on the centring slice.

The image to be tested is transformed into input vectors which are formed from the intensity values of the pixels obtained with the input stencil window, and each input window is paired with the corresponding target value which is the *ground truth*. In order to evaluate the performance of the networks, metrics described in "Section 3.2.2 Training" are calculated with the result produced by related network. The numbers of true positives, false positive, true negatives, and false negatives are computed by comparing the resultant image and the related ground truth. Ground truth image is a binary image which is composed of positive (membrane boundaries) and negative (everything else in the image) labels while the row output of the network is continuous since the activation function in the output layer produces values between 0 and 1 where a value near to 0 corresponds to negative classification while 1 corresponds to positive. In order to obtain the exact binary classification, a threshold operation is applied to the output image. Although the balance between the number of positive and negative examples used in the training may affect the selection of the threshold value used in this step as discussed in the work by Middleton (Middleton & Damper, 2004), this is ignored in this study and a threshold value of 0.5 is used for many of the networks trained.

4.1. Tests for Selection of Neural Network Parameters

As described in section "3.1.3. Preparation of the Data for Training and Testing", the balance between the negative and positive number of training patterns are arranged so that the trained networks do not tend to produce too much false negatives. In the work by Ciresan et al. (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012) same amount of positive examples are randomly selected from all negative examples while in another work (Jurrus, et al., 2013) the number of randomly selected negative examples are two times more than the positive ones. When the aim is to segment the cristae membrane boundaries in electron microscopy images, using more negative examples makes sense since there exists relatively much more negative examples that represent

the non-membrane pixels. In order to observe the effect of the ratio to be used during the training, Figure 48 can be viewed where a specific network is trained with different training sets which are composed of; (i) the same number of negative and positive examples, (ii) the number of negative examples are three times more than the positive ones, (iii) and all the negative and positive patterns in the images without decreasing the number of negatives. In the case (i), the total number of examples are 2x48550; in the case (ii) the number of negative examples is 145650 (3x48550); and in the final case the number of negative examples is 263430 while the positives are 48550. The width of the network used here is 1 which means that the segmentation is done in 2D. It is seen from the Figure that the best true positive rate can be obtained with equal number of positive and negative training examples are used (Figure 48-a). But as true positive rate increases false positive rate is also increased and as a result the minimum false positive rate is obtained in Figure 48-a.

(a) (b) (c)
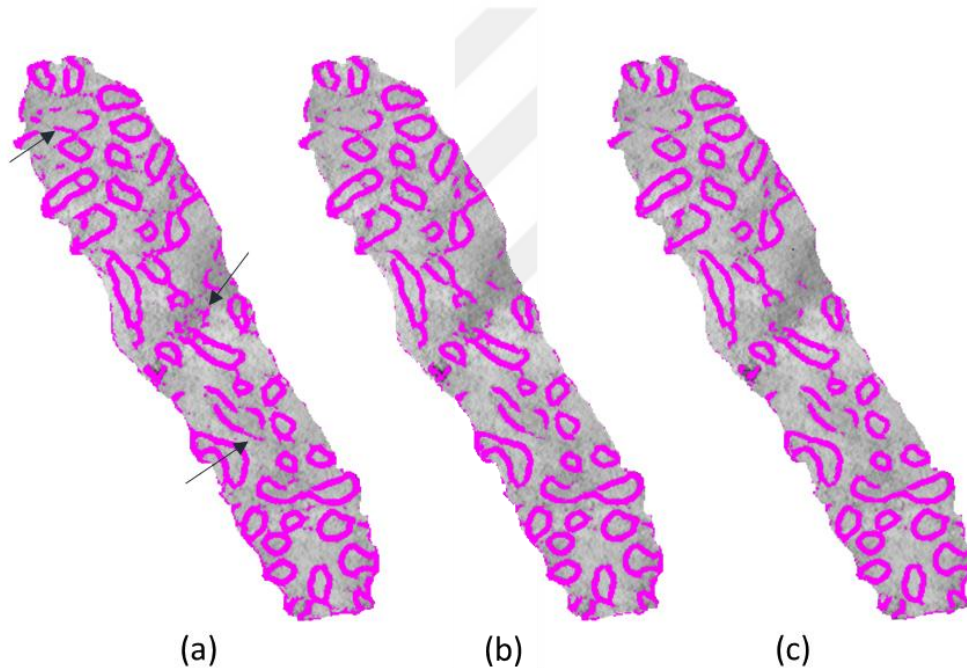
Figure 48. Results from the networks trained with different numbers of negative training examples. Remarkable differences are pointed with black arrows. Positive examples = 48550 in all cases. The number of negative patterns are; 48550 in (a), 145650 in (b), 263430 (c). (Network properties: Input Stencil Size =15, Centring Square Patch = 9, Width = 1 (2D network); Hidden nodes = 55, Pixels adjacent to positives are not used, balance ratio of one, three, and no ratio is used for training set in row one, two, and three respectively.)

The performance metrics can be seen in Table 3. As described above, TPR decreases as the number of negative conditioned examples in the training set increases. Although effectivity metric has approached the highest value where all the negative examples are used, it will not be used during the preparation of training sets in the remainder of this study because the TPR is too low to be acceptable while the aim of this study is to segment the cristae membrane boundaries.

Table 3. Performance metrics of the networks trained with different numbers of negative training examples for the comparison of the effect of the number of negative conditioned examples versus positive ones. (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Width = 1 (2D network); Hidden nodes = 55, Pixels adjacent to positives are not used, balance ratio of one, three, and no ratio is used for training set in row one, two, and three respectively.

	TPR	TNR	precision	effectivity
NN1-iw1-15-HN-55-W1-B1-WOA	85%	98%	76%	80%
NN1-iw1-15-HN-55-W1-B3-WOA	79%	99%	83%	81%
NN1-iw1-15-HN-55-W1-Ball-WOA	77%	99%	86%	82%

In section "3.2.2.1 Segmentation Using Classical Feed-Forward Neural Networks", it was described that there are no gold standard methods in the literature for determining the number of hidden layers and nodes while designing the architecture of the network. In order to observe the effect of the number of hidden layers, two networks trained under the same conditions (i.e: input nodes, activation functions, training set). The only difference is that one of them has one hidden layer while the other one has two hidden layers. The resultant images of networks can be seen in Figure 49. It shall be noted that the initial weights in the beginning of two training were not the same but this can be ignored because each training is done for 10 times so that the effect of the weight initialization is discarded.

(a)   (b)
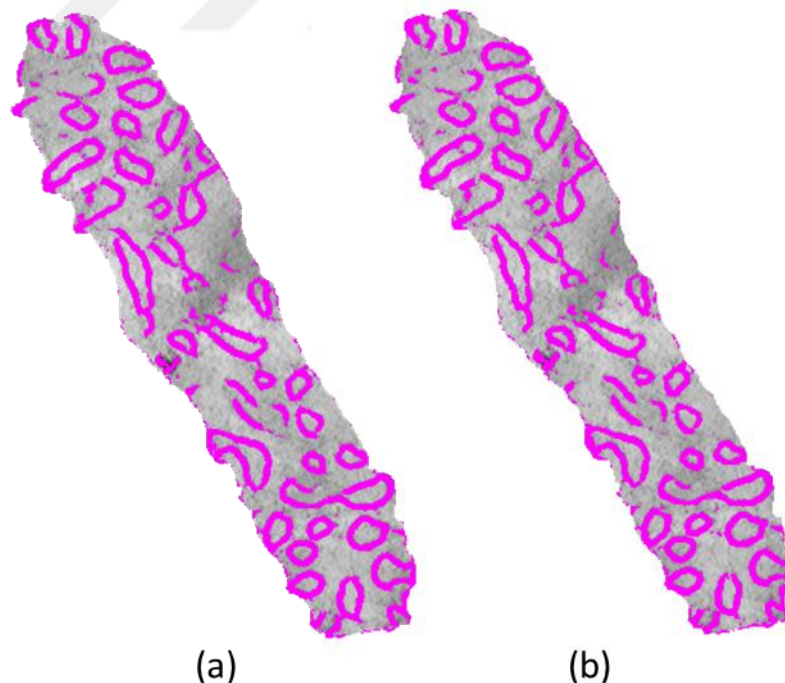
Figure 49. The comparison of the result of one and two hidden layer networks. (a: one hidden layer, b: two hidden layer). (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Width = 1 (2D network); Hidden nodes in the first hidden layer= 55, Hidden nodes in the second layer = 10, Pixels adjacent to positives are not used, balance ratio of one used for training set)

From the Table 4 below, it can be seen that no significant performance change on TPR and TNR occurs while the effectivity performance is decreased with additional hidden layers.

Table 4. Performance metrics of one and two hidden networks. The first one (in first row) is trained with one hidden layer while the second with two hidden layers. (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Width = 1 (2D network); Hidden nodes in the first hidden layer= 55, Hidden nodes in the second layer = 10, Pixels adjacent to positives are not used, balance ratio of one used for training set)

	TPR	TNR	precision	effectivity
NN1-iw1-15-HN-55-W1-B2-WOA	79%	99%	83%	81%
NN1-iw1-15-HN1-55-HN2-10-W1-B2-WOA	79%	99%	82%	80%

In Figure 50, two networks whose number of nodes in the hidden layer are different from each other can be seen. From the results it can be seen that the increase in the number of hidden nodes in hidden layer does not significantly affect the TNR and effectivity performance of the network but increase the TPR metric for %0.5.

(a)   (b)
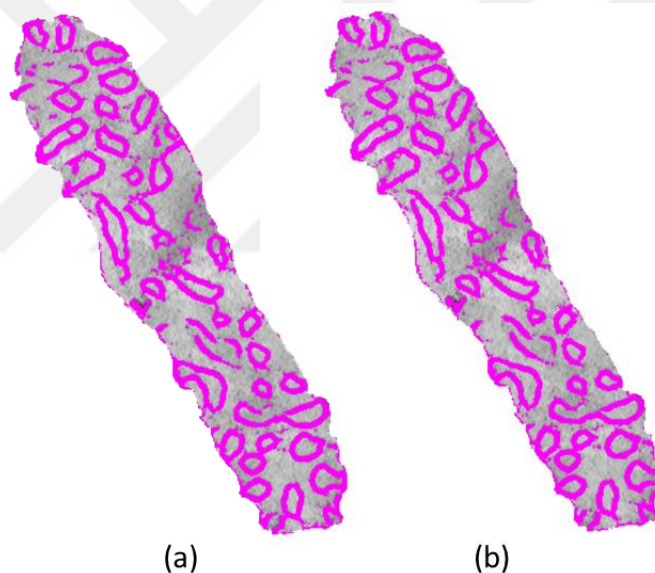
Figure 50. The comparison of the result of identical networks except whose number of nodes in the hidden layer is different. The number of nodes is 55 in (a) and 20 in (b). (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Width = 1 (2D network); Pixels adjacent to positives are not used, balance ratio of one used for training set)

The performance metrics in Table 5 verifies the results seen in Figure 50 above.

Table 5. Performance metrics of the networks trained with 55 and 20 number of nodes in the hidden layer in the first and the second row respectively. All other parameters of the networks and the training set are identical. (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Width = 1 (2D network); Pixels adjacent to positives are not used, balance ratio of one used for training set)

	TPR	TNR	precision	effectivity
NN1-iw1-15-HN-55-W1-B1-WOA	85%	98%	76%	80%
NN1-iw1-15-HN-20-W1-B1-WOA	84%	98%	75%	80%

In Figure 51 the effect of not using the negative patterns (non-membrane) that are adjacent to the positive examples (cristae membrane) during the training can be seen. It is seen that the method used in Figure 51-b during preparation of the training set effected the performance of the training so that the network produced thicker membrane boundaries when compared Figure 51-a. This result make sense because in the training set prepared with the method in Figure 51-b, there are no patterns that can teach the network not to mark the negative patterns that are adjacent to the membrane boundaries and as a result these pixels are marked as membrane boundaries which yields to higher true positive rates. But the number of true negatives are decreased with 'without adjacent pixels' approach.
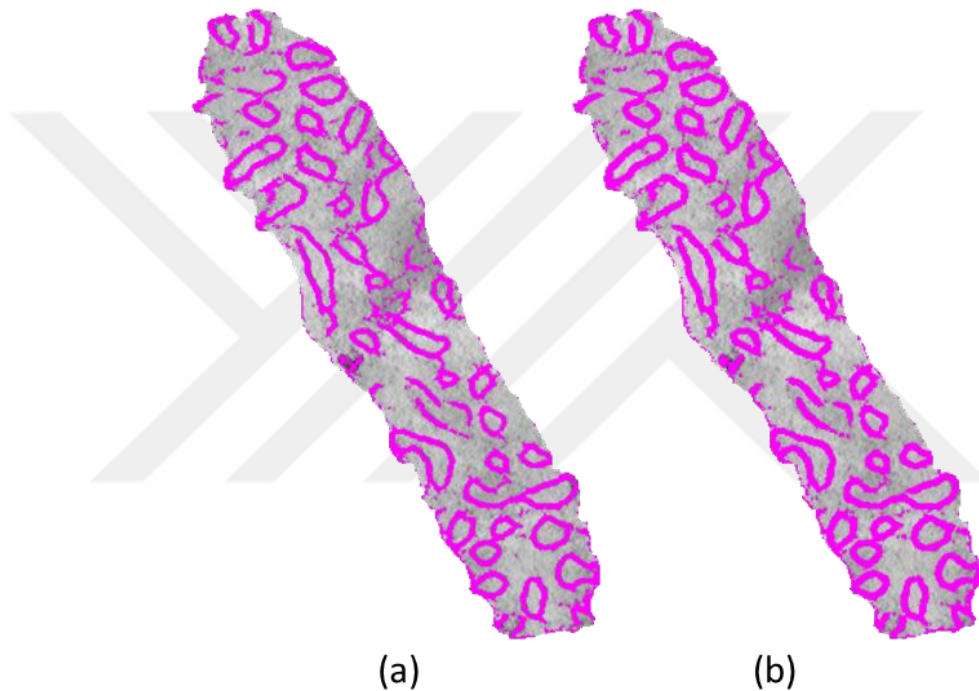
Figure 51. Results for the comparison of training without the adjacent pixels to positives. In (a) the result of the network trained with the examples including the negative ones that are adjacent to the positive ones. In (b) the patterns mentioned in (a) are not included in training set of the network. (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Hidden nodes in the first hidden layer= 55, Width = 1 (2D network))

The effect of producing thicker membrane boundaries can be seen in Table 6 that the true negative rate (TNR) is decreased for %1. The TPR increased dramatically so that the effectivity metric is increased for %4 although the precision metric has been decreased due to %4 decrease in TNR.

Table 6. Performance metrics of the networks trained with the use of "with adjacent pixels" versus "without adjacent pixels (WOA)" approaches explained above. (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Hidden nodes in hidden layer= 20, Width = 1 (2D network))

	TPR	TNR	precision	effectivity
NN1-iw1-15-HN-20-W1-B1	73%	99%	79%	76%
NN1-iw1-15-HN-20-W1-B1-WOA	84%	98%	75%	80%

Another important issue is to select the size of the input window which indeed defines the number of elements in the input layer of the network trained. In Figure 52 a basic comparison of two networks one of which is trained with a relatively smaller size of input window than the other one.

(a)   (b)

```
[1 0 0 0 0 0 0 0 0 1;          [1 0 0 0 0 0 0 1 0 0 0 0 0 0 1;
 0 1 0 0 0 0 0 0 1 0;           0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
 0 0 1 1 1 1 1 1 0 0;           0 0 1 0 0 0 0 1 0 0 0 0 1 0 0;
 0 0 1 1 1 1 1 1 0 0;           0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 1 1 1 1 1 1 1 1 1 1;           0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 0 0 1 1 1 1 1 1 0 0;           0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 0 0 1 1 1 1 1 1 0 0;           1 0 1 1 1 1 1 1 1 1 1 1 1 0 1;
 0 1 0 0 1 0 0 0 1 0;           0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 1 0 0 0 1 0 0 0 0 1].          0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
                                0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
                                0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
                                0 0 1 0 0 0 0 1 0 0 0 0 1 0 0;
                                0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
                                1 0 0 0 0 0 0 1 0 0 0 0 0 0 1].
```

(c)   (d)

Figure 52. The result of the networks trained in order to reveal the effect of the input window size. The size of the input layer is 41 in (a) and 97 in (b). (a) and (b) are the resultant images and (c) and (d) are related input windows used to obtain the input vectors. (The network properties: Hidden nodes in hidden layer= 20, Width = 1 (2D network), Pixels adjacent to positives are not used)

It seems there is no significant difference between the network trained with the smaller seized window and the other one. This result is can be observed with the metrics seen in Table 7. This result reveal that the input window covering around 9 pixels contains enough information for segmenting the membrane boundaries.

Table 7. Performance metrics of the networks trained with the use of small and large input stencil windows where input stencil size is 9 and centring square patch is 5 in the first row and input stencil size is 15 and centring square patch is 9 in the second row (The network properties: Hidden nodes in hidden layer= 20, Width = 1 (2D network), Pixels adjacent to positives are not used)

	TPR	TNR	precision	effectivity
NN1-iw1-9-HN-20-W1-B1-WOA-CP5	84%	98%	78%	81%
NN1-iw1-15-HN-20-W1-B1-WOA-CP9	84%	98%	75%	80%

As described in section "3.1.3. Preparation of the Data for Training and Testing", two networks that have the same architecture are trained with the training sets that were prepared with two different input window shapes (square and stencil patch) in order to observe the effect on the classification performance. Almost no change is observed in the performance except little changes as marked with black arrows in Figure 53 although additional 16 input features are used with the use of input stencil window during the training phase.

(a)   (b)

```
[1 1 1 1 1 1 1 1 1;
 1 1 1 1 1 1 1 1 1;
 1 1 1 1 1 1 1 1 1;
 1 1 1 1 1 1 1 1 1;
 1 1 1 1 1 1 1 1 1;
 1 1 1 1 1 1 1 1 1;
 1 1 1 1 1 1 1 1 1;
 1 1 1 1 1 1 1 1 1;
 1 1 1 1 1 1 1 1 1]
```

(c)

```
[1 0 0 0 0 0 0 1 0 0 0 0 0 0 1;
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0;
 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 1 0 1 1 1 1 1 1 1 1 1 1 0 1;
 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0;
 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0;
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1].
```

(d)

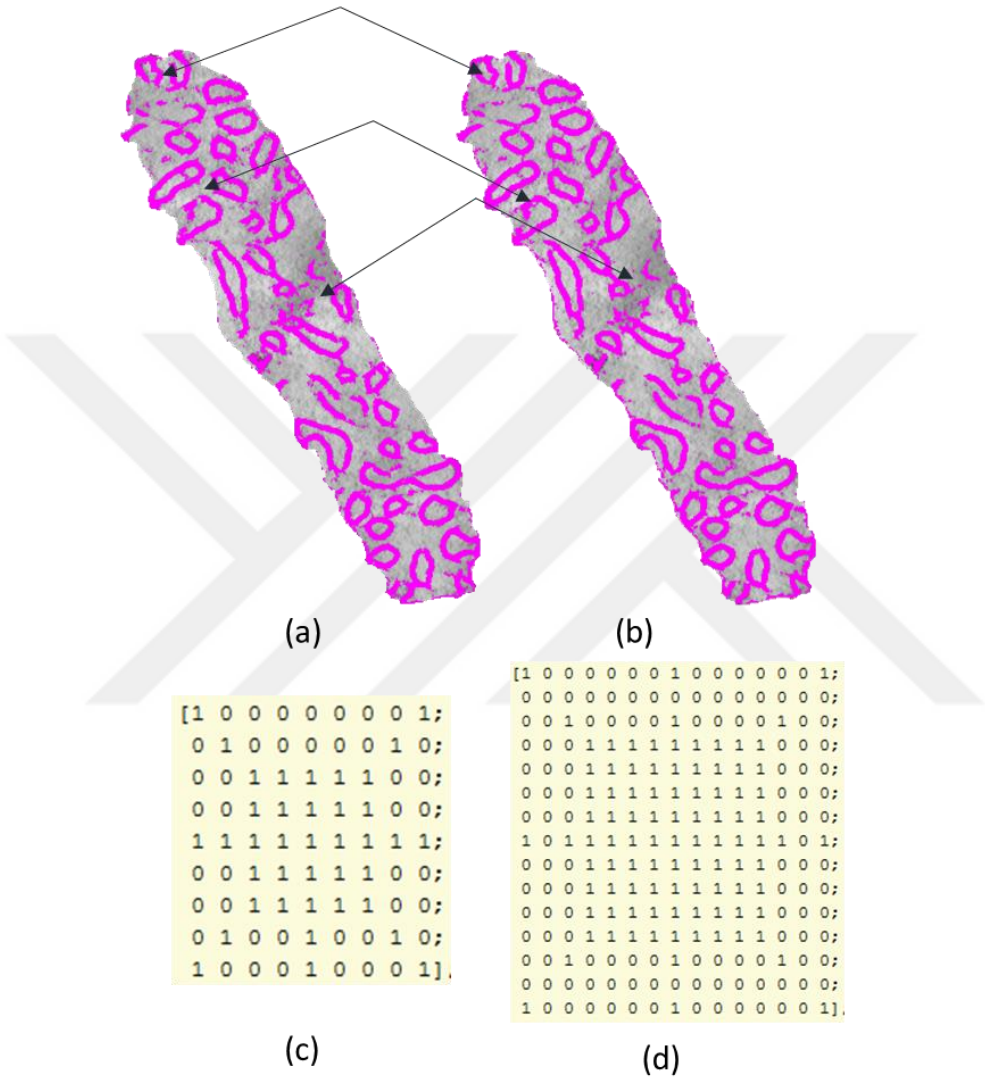Figure 53. The result of the networks trained in order to determine the effect of stencil window approach. The size of the input layer is 81 in (a) and 97 in (b). (a) and (b) are the resultant images and (c) and (d) are related input windows used to obtain the input vectors. (The network properties: Hidden nodes in hidden layer= 20, Width = 1 (2D network), Pixels adjacent to positives are not used)

The situation observed in Figure 53 above can be viewed in Table 8. The test results show that stencil window approach do not increase the performance of the network.

Table 8. Performance metrics of the networks trained with the use of stencil and square window approaches where input stencil and centring square patch size is 9 in the first row and input stencil size is 15 and centring square patch is 9 in the second row. (The network properties: Hidden nodes in hidden layer= 20, Width = 1 (2D network), Pixels adjacent to positives are not used)

	TPR	TNR	precision	effectivity
NN1-iw1-9-HN-20-W1-B1-WOA-CP9	84%	98%	78%	81%
NN1-iw1-15-HN-20-W1-B1-WOA-CP9	84%	98%	75%	80%

Electron Microscopy Tomography imaging provides 3D volumetric data that contain many information that would make the performance of the cristae membrane segmentation better as mentioned in section "3.3. Segmentation in Three-Dimensions". In Figure 54, two identical networks one of which trained with a training set obtained from single 2D slices while the other one is trained with volumetric data with a width of 5 slices. Networks trained with 3D data are better at segmenting disconnected membrane boundaries as well as not segmenting darker boundary-like non-membrane regions as seen from the figure below.
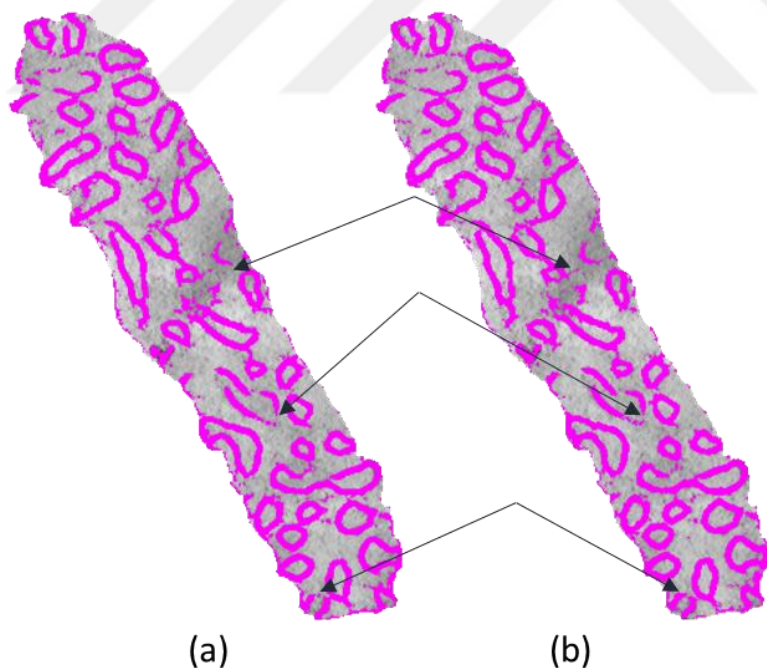
Figure 54.The comparison of 2D versus 3D networks' results. (a) is a network trained with 2D single slices. (b) is a network trained with 5 slices of volumetric 3D data. (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Hidden nodes = 55, Pixels adjacent to positives are not used, balance ratio of one is used).

The performance metrics of the networks trained above verifies the inferences made from the resultant images. In Table 9, it is seen that both the precision and the effectivity metrics are increased. Also the TNR is increased for %1. It shall be noted

that a very small change in TNR corresponds to a very big change in the performance of the network when compared to TPR since there exist almost 5-6 times more negative patterns than positive ones in a test image. With the use of advantage of 3D volumetric data, all four performance metric have reached the best performance.

Table 9. Performance metrics of the networks trained with 2D versus 3D training data. (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Hidden nodes = 20, Pixels adjacent to positives are not used, balance ratio of one is used).

	TPR	TNR	precision	effectivity
NN1-iw1-15-HN-20-W1-B1-WOA-CP9	84%	98%	75%	80%
NN1-iw1-15-HN-20-W5-B1-WOA-CP9	84%	99%	81%	82%

4.2. Quantitative Evaluation of Results

There exist various segmentation techniques and proposed methods in the literature as reviewed in sections "2.1 Image Segmentation" and "2.3. Current Research on Automatic Mitochondria and Cristae Segmentation". But segmentation of EMT images by means of neural networks gained importance because they produce promising results in the studies like (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012), (Jurrus, et al., 2013), and (Jain, et al., 2007, October).

In the previous section, the effort made in order to select the architecture and the parameters of the network to be trained is explained. Once the architecture is selected and the parameters are determined, the data set to be used for training is prepared as described in section "3.1.3. Preparation of the Data for Training and Testing". On the training set preparation phase, a challenging problem arose: Determination of the balance ratio between the negative and positive samples in the training set. It is a common approach to train the networks with same numbers of positive and negative examples (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012), (Jurrus, et al., 2013). But since each class is not equally represented in the testing data, the networks trained with this approach tend to overestimate the membrane probability (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012) resulting with low TNR while maximizing the TPR. As there exist much more numbers of negative pixels in the images aimed to be segmented, equal representation of two classes in the training set approach yields to a decrease in the effectivity metric due to the decrease in the TNR metric. The TNR performance can be reached to an acceptable value with increasing the numbers of negative examples in the training set. But this time the TPR performance decreases as FNR is increased. In Table 10, the numbers for the classification results of two networks where the one in the first row is trained with equally representation of both classes. The second network is trained with three times more number of negative examples than positives. As seen, with the use of three times more number of negative examples than positives in the training set, a number of 880 true positive classifications are sacrificed in order to obtain a number of 1052 more true negative classifications.

Table 10. Results of two networks trained with different number of negative examples. The first row; equal representation of two classes and the second row; 3 times more representation of negative examples in the training set. (Note that a threshold value of 0.5 is used.) (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Hidden nodes = 20, Width = 5 (3D network), Pixels adjacent to positives are not used).

	TP	TN	FP	FN
NN1-iw1-15-HN-20-W5-B1-WOA-CP9	9320	170737	2148	1811
NN1-iw1-15-HN-20-W5-B3-WOA-CP9	8440	171789	1096	2691

The threshold value applied (0.5 (127 in [0-255] range) in the Table 10 above) can be arranged as a solution to increase the number of true positives in "NN1-iw1-15-HN-20-W5-B3-WOA-CP9". If a value of 199 is used for thresholding then the classification table for the second network (NN1-iw1-15-HN-20-W5-B3-WOA-CP9) becomes as seen in Table 11. It is seen that the adjustment of threshold value is not a solution to this problem since the network produces more false positives although the number of true positives are equalized.

Table 11. The effect of thresholding operation. Classification results of the network in Table 10 above when threshold value is set to 199. (The network properties: Input Stencil Size =15, Centring Square Patch = 9, Hidden nodes = 20, Width = 5 (3D network), Pixels adjacent to positives are not used).

	TP	TN	FP	FN
NN1-iw1-15-HN-20-W5-B3-WOA-CP9	9318	170711	2174	1813

In Figure 55, the resultant images of the networks in Table 10 above can be seen. It is seen that the cristae boundaries that relatively easy to discriminate in the original image in Figure 55-a can be strongly segmented in both cases. But the problem arises for the ones that are represented with a very weak gradient (almost invisible) or seem totally obscure. In EMT images this may occur in some portion of membrane boundaries. These portion of membranes are also produced as weak boundaries in the output continuous images of the network. Detection of disconnected boundaries that usually are represented with weak values is a very challenging problem. This phenomenon can be observed in Figure 55–(c) (red arrows). In Figure 55-c, the result of the network trained with the set in which there exist three times more number of negative examples than the number of positives can be seen. The red arrows in Figure 55-c show the boundaries where the network could not produce strong enough results when compared to boundaries in Figure 55-b in which the network is trained with equal representation of both classes. The boundaries are classified with stronger intensities (red arrows) in Figure 55-b. But this time many pixels that belong to non-boundary regions are represented with more probability for belonging to membrane boundary regions (shown with blue arrows in Figure 55).
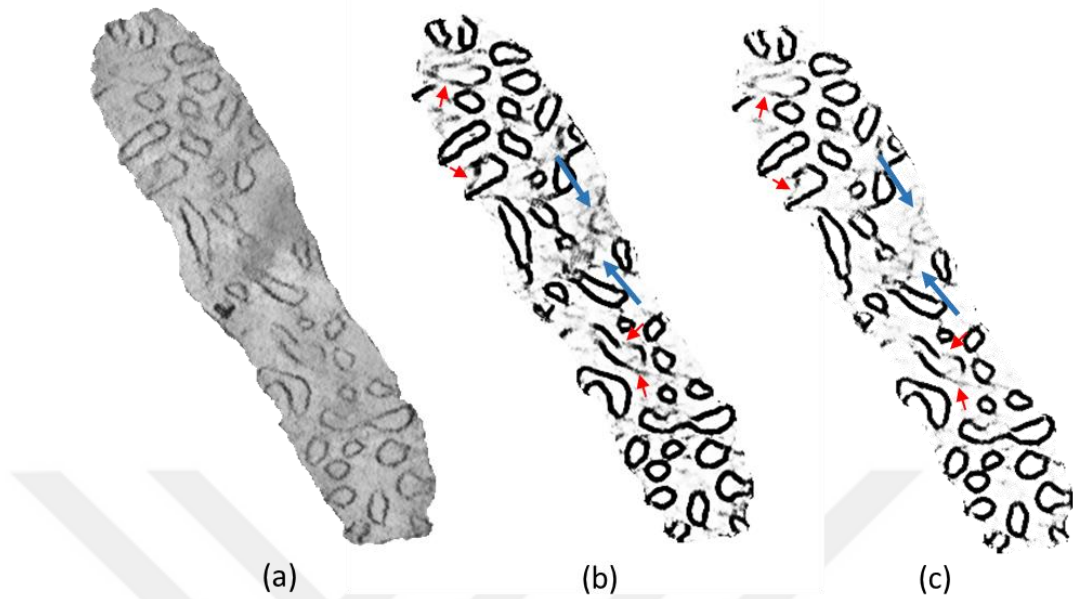
Figure 55.The comparison of result images of two networks. (a) the original image to be segmented, (b) the result of network trained with equal representation of two classes in the training set and (c) the result of network trained with three times more representation of negatives in the training set.(The network in b and c are NN1-iw1-15-HN-20-W5-B1-WOA-CP9 and NN1-iw1-15-HN-20-W5-B3-WOA-CP9 respectively where NN1-iw1-15-HN-20-W5-B1-WOA-CP9 represents the network properties: Input Stencil Size =15, Centring Square Patch = 9, Hidden nodes = 20, Width = 5 (3D network), Pixels adjacent to positives are not used, balance ratio of one is used and NN1-iw1-15-HN-20-W5-B3-WOA-CP9 represent the same network except balance ratio of three is used.)

As seen in Figure 55 above, the networks produce continuous outputs since the activation function in the output layer is a continuous differentiable sigmoid function. It is a common approach to apply thresholding operation to the outputs of the networks in order to obtain binary classification results of the networks. While the determination of the threshold value can be done by means of different adaptive methods or polynomial post-processor functions (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012), in this study empirically determined threshold values used in order to obtain binary results. The threshold application is shown in Eq (*) below.

$$I(x,y) = \begin{cases} 1, & I(x,y) > T \\ 0, & \text{otherwise} \end{cases} \qquad (?)$$

where T is the threshold value and $I(x,y) = 1$ represents the negative (non-boundary) classifications. Since a value of 0 (zero) is an absolute boundary (%100 positive) with zero probability of belonging to non-boundary region in the continuous output images, the number of positives increase as the threshold value T is increased. As a result the selected threshold value determines the balance between TPR and TNR performance of the network. In Figure 56 below, the binary results obtained with the application of different thresholds can be seen where the threshold value is increased from left to right. As the threshold value is increased, much more number of pixels fall behind the threshold value so that the number of positive marked pixels increase. In Figure 56-a, some weak or disconnected boundaries are marked as negative while in Figure 56-c

many of them are marked as positive. But as the weak or disconnected boundaries marked, the number of false positives are increased.

(a)   (b)   (c)
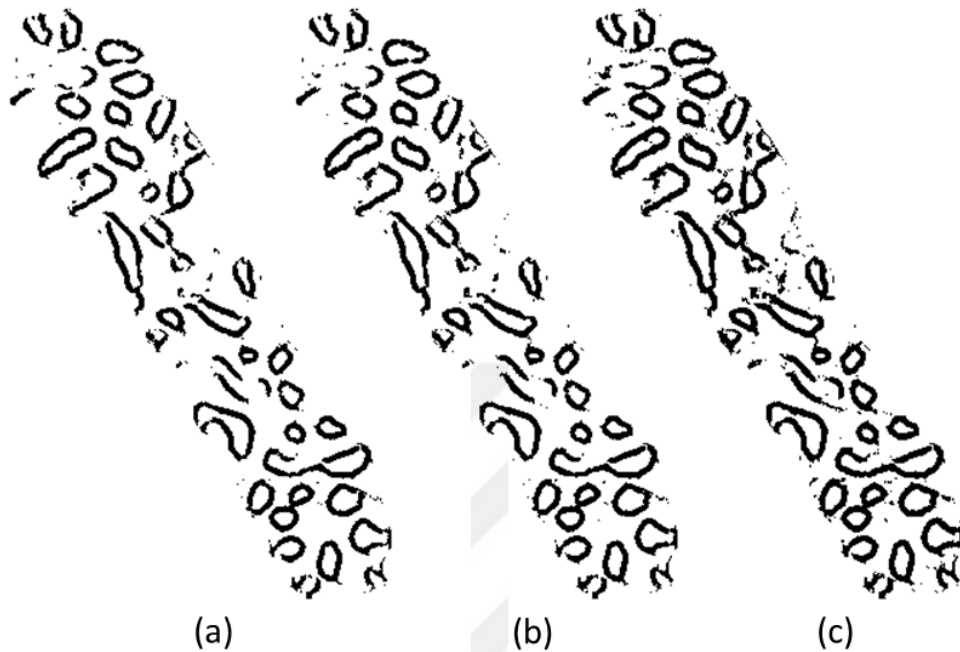
Figure 56. The effect of threshold operation on the performance of detecting weak or disconnected boundaries. The threshold values used are 100, 127, and 195 respectively. (The network (NN1-iw1-15-HN-20-W5-B3-WOA-CP9) properties: Input Stencil Size =15, Centring Square Patch = 9, Hidden nodes = 20, Width = 5 (3D network), Pixels adjacent to positives are not used, balance ratio of three is used)

It shall be noted that the weak boundaries mention above are the membranes that are almost invisible on the original image. The 3D neural networks trained are capable of detecting the semi-weak membrane boundaries as seen in Figure 57 below.

Figure 57. The neural networks can successfully detect semi-weak boundaries.

A special approach for the post-processing phase as a substitute of threshold operation is proposed in order to overcome the problems raised above when thresholding is used. With this method, it is aimed to increase the number of true positives while keeping the increase in the number of false positives relatively low. The method as explained in "Section 3.4 Directional Hessian Ridge Growing" iteratively enlarges the strong boundaries through the ridge direction so that the weak boundaries that cannot be detected with basic threshold operation can be segmented.

As the method needs the selection of parameters like two threshold values for hysteresis thresholding, similarity metric threshold, input window size, and number of iterations; many experiments are done and the values for related parameters are determined empirically.

In Figure 58 below, a comparison of the proposed method can be seen. In (b), the binary image in which thresholding is applied to the network's output image in (a). As one can observe from the comparison between the ground truth image in (d) and (b), many boundaries cannot be detected and a couple of cristae seem to be disconnected.

The application of directional growing method almost connected the big gaps on the disconnected regions of cristae.
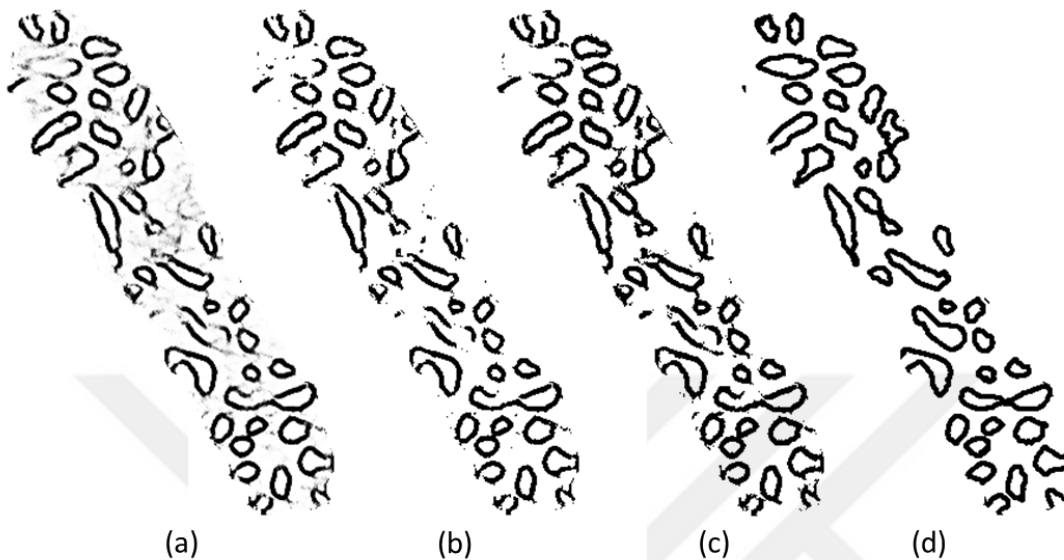
Figure 58. The comparison of the performance of thresholding and directional growing methods. (a); the original output of the network, NN1-iw1-15-HN-20-W5-B3-WOA-CP9. (b); the thresholded image with value, 127. (c); the resultant image after directional growing method applied with the parameters; thresholds for hysteresis thresholding: 40 and 212, input window size: 5, similarity metric threshold: 0.828, and number of iterations: 30. (d) is the ground truth image.

In Figure 59 below benefits of the method can be observed much clearly that the small gaps between disconnected cristae boundaries are filled and the cristae which are disconnected with big gaps are almost connected.

While the gaps between the disconnected cristae boundaries are filled with this method, still the method can produce false alarms in the areas where the output of the network fall behind the threshold with which the strong boundaries are selected. The method may cause to grow these regions.
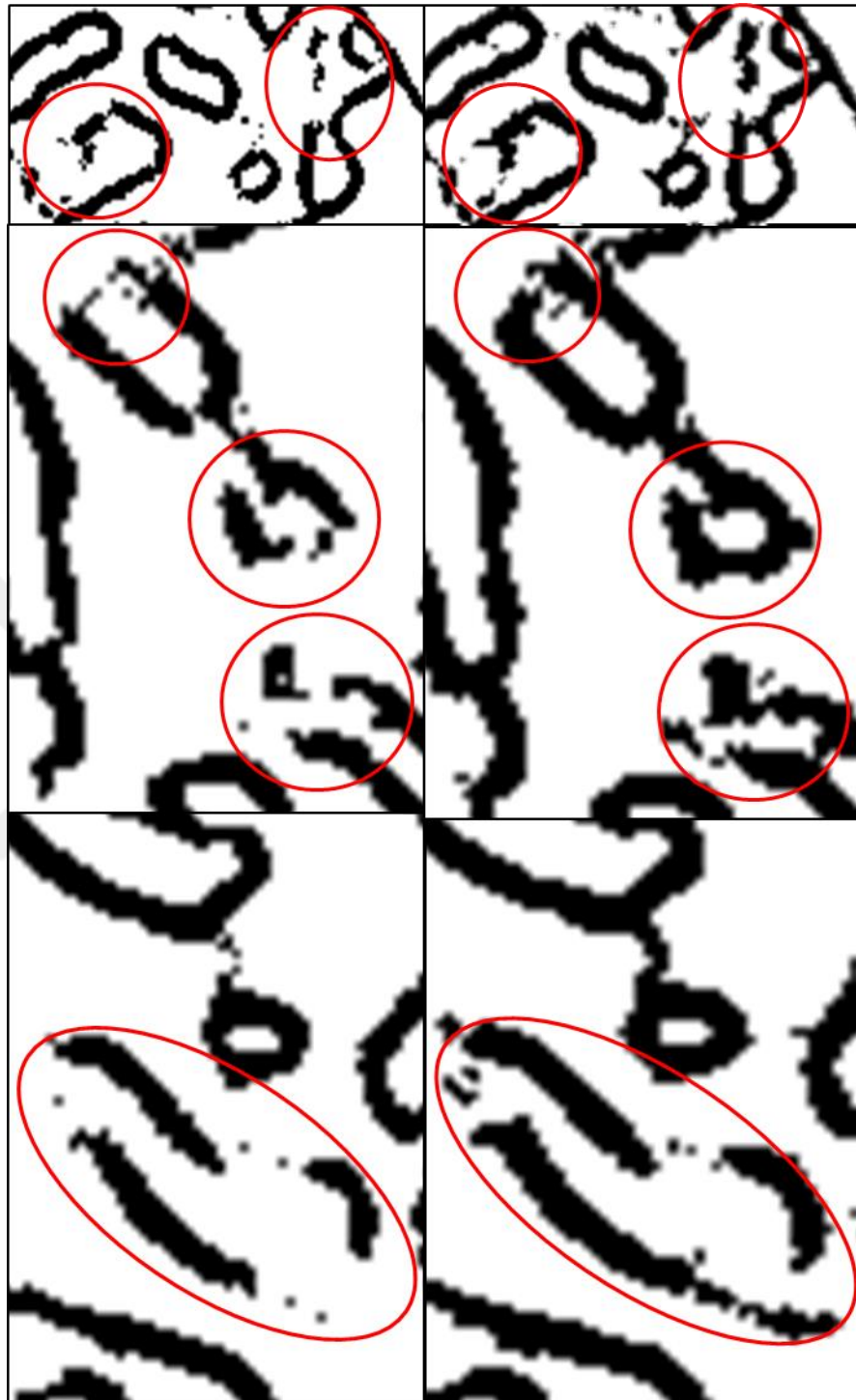
Figure 59. Detail view of the results shown in Figure 58. On the left side the results of thresholding method is seen. On right side it is seen that the boundaries are connected or at least big gaps are closed with the proposed method. (Results from NN1-iw1-15-HN-20-W5-B3-WOA-CP9 network is used).

One way to make a fair comparison between the thresholding operation and the directional growing, the threshold value can be arranged so that the number of true positives in the thresholded image become equal to the number of true positives in the directional growing result. The threshold value that produces almost the same number

of true positives is determined empirically (th=199). When parameter of both methods are set to produce the same number of TPs, it is clearly seen that the directional growing method over performs the thresholding method as seen in Table 12 and Figure 59.

Table 12. The comparison of directional growing method versus basic thresholding. (NN1-iw1-15-HN-20-W5-B3-WOA-CP9 network is used for both of the results) (Parameters of directional growing method: thresholds for hysteresis thresholding: 40 (strong boundaries) and 210 (weak boundaries), input window size: 5, similarity metric threshold: 0.84, and number of iterations: 30.)

	TP	TN	FP	FN
NN1-iw1-15-HN-20-W5-B3-WOA-CP9 Threshold value = 199	9318	170711	2174	1813
DirGrow-thH-212-thL-40-iw-5-sim-0.828	9323	170883	2002	1808

In Figure 60, the first two images are obtained by means of threshold operation (threshold values 199 and 127 respectively) on the network's output and the image at the very right side of the figure is obtained by means of directional growing method. Figure 60-(a) and (c) has the same number of true positives (~10340), but (a) produced a much more noisy and dirty result which includes significantly much more false positives. The coloured boundaries seen intermittently on cristae membranes are the ground truth of the image and shows the regions that could not be detected.

(a)　　　　　(b)　　　　　(c)
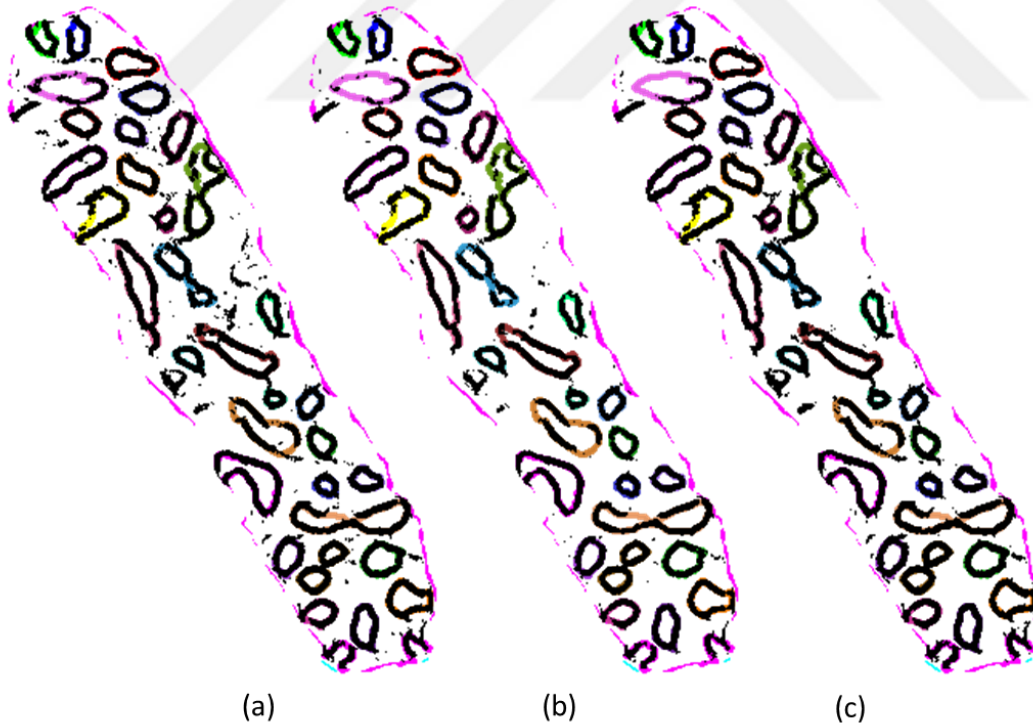
Figure 60. A fair comparison between threshold (images in a & b) and directional growing (image in c) methods where the coloured boundaries are the ground truth markings and black boundaries are the produced segmentations. (Threshold values 199 & 127 respectively. NN1-iw1-15-HN-20-W5-B3-WOA-CP9 network is used for all of the results. Parameters of directional growing method: thresholds for hysteresis thresholding: 40 (strong boundaries) and 212 (weak boundaries), input window size: 5, similarity metric threshold: 0.828, and number of iterations: 30.)

As mentioned in the beginning of this chapter, equal representation of two classes in the training set maximizes the effectivity metric defined in the previous chapters. But it was also shown that the TNR is decreased while TPR is maximized. Figure 61 can be viewed in order to compare the result of best performing neural network (by which the effectivity metric is maximized) with the results obtain from direction growing method. It is seen from the figure that the results of directional growing method (which is shown in Figure 61-b) are better from the results (which is shown in Figure 61-a) obtained with the best performing network (NN1-iw1-15-HN-20-W5-B1-WOA-CP9) especially for connecting the disconnected membranes (regions pointed with red arrows). Also as seen in the regions covered with red sketches, the directional growing method produces much 'clearer' results and increase the TNR.

(a)   (b)
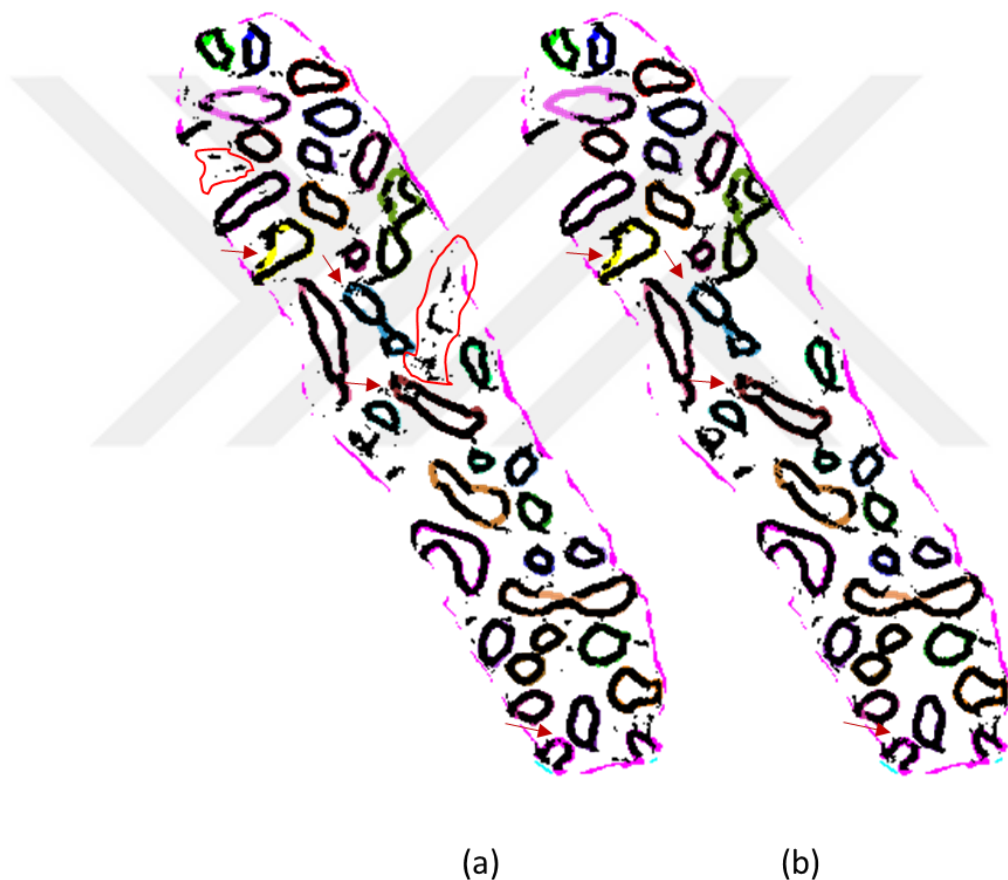
Figure 61. The comparison of the results of the best performing network (a) and directional growing method (b) where the coloured boundaries are the ground truth markings and black boundaries are the produced segmentations. (The network in (a) is NN1-iw1-15-HN-20-W5-B1-WOA-CP9, the threshold value used is 127. The network in (b) is NN1-iw1-15-HN-20-W5-B3-WOA-CP9 and parameters of directional growing method are; thresholds for hysteresis thresholding: 40 (strong boundaries) and 210 (weak boundaries), input window size: 5, similarity metric threshold: 0.828, and number of iterations: 30

As Figure 61 above reveals the benefits of directional growing method, from Table 13 below, it is seen that the directional growing method increases the effectivity and precision metric with only a value of %1. If Figure 61 is investigated in detail, one can see that the boundaries produced by directional growing method are thicker than the

networks output since the regional growing occurs also on the edges of detected boundaries. This increases the number of false positives and approximates the performance of two methods.

Table 13. Binary classification results of the best performing network and directional growing method.

	TP	FP	TPR	TNR	Precision	Effectivity
NN1-iw1-15-HN-20-W5-B1-WOA-CP9	9320	2148	84%	99%	81%	82%
DirGrow-thH-212-thL-40-iw-5-sim-0.828	9323	2002	84%	99%	82%	83%

In Figure 62 below the results of the algorithms applied to an example from the cone.sub data set can be seen. It is seen that while the result of the neural network produces satisfactory results for segmenting the cristae membrane, it also produces many false positives. The directional growing method decreases the number of false positives and produces much more 'cleaner' resultant image.

(a)　　　　　　　　　　(b)

(c)　　　　　　　　　　(d)

Figure 62. Results from cone data set. (a) is the original image (from cone data set and obtained by means of cropping around mitochondria inner membrane) that is used as input to the trained network which produced the result in (c). (b) is the ground truth image and (d) is the result of directional growing method which used the grayscale version of the network result seen in (c).

In Figure 63 below the results of the algorithms applied to an example from the bclpb-d.sub data set can be seen. As many disconnected cristae membrane boundaries can be seen in Figure 63-c and d, neither the neural network nor the directional growing method could produce satisfactory results. This result is somehow understandable since much portion of the cristae in the image is invisible. So one can conclude that there is not enough information for a satisfying segmentation performance. But still the directional growing method eliminated many false positives produced by neural network.

(a)

(b)

(c)

(d)
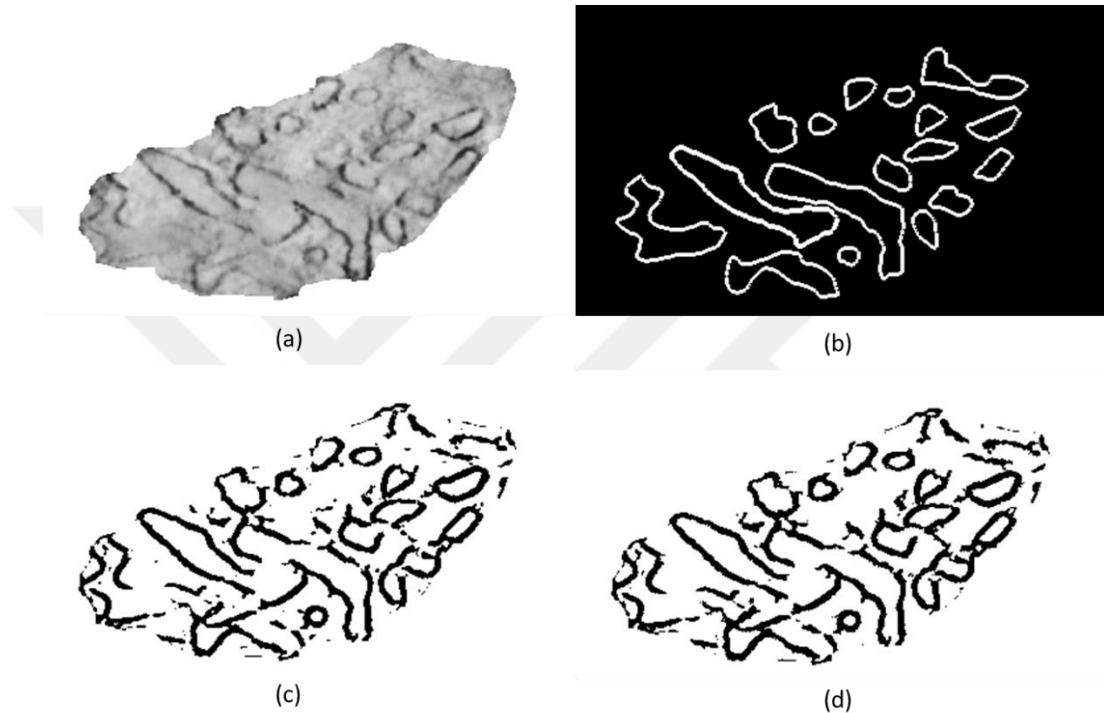
Figure 63. Results from bclpb-d data set. (a) is the original image (from bclpb-d data set and obtained by means of cropping around mitochondria inner membrane) that is used as input to the trained network which produced the result in (c). (b) is the ground truth image and (c) is the result of directional growing method which used the grayscale version of the network result seen in (c).

In Figure 64-a below, an example from gap-18 data set in which most of the cristae boundaries are almost invisible can be seen. When Figure 64-d and Figure 64-c is compared, one observe that directional growing method over perform the neural network in this example. Many of cristae membrane boundaries are segmented after the grayscale result of the neural network is processed with directional growing method. It shall also be noted that the no slices from the data set that this example in Figure 64 belongs to is used during training.
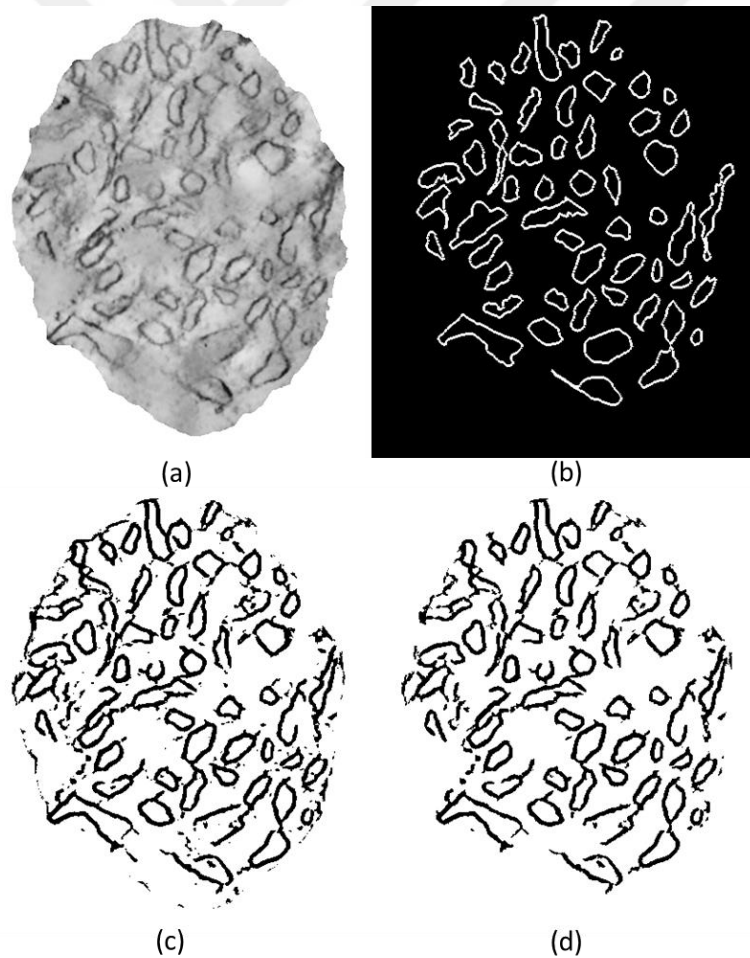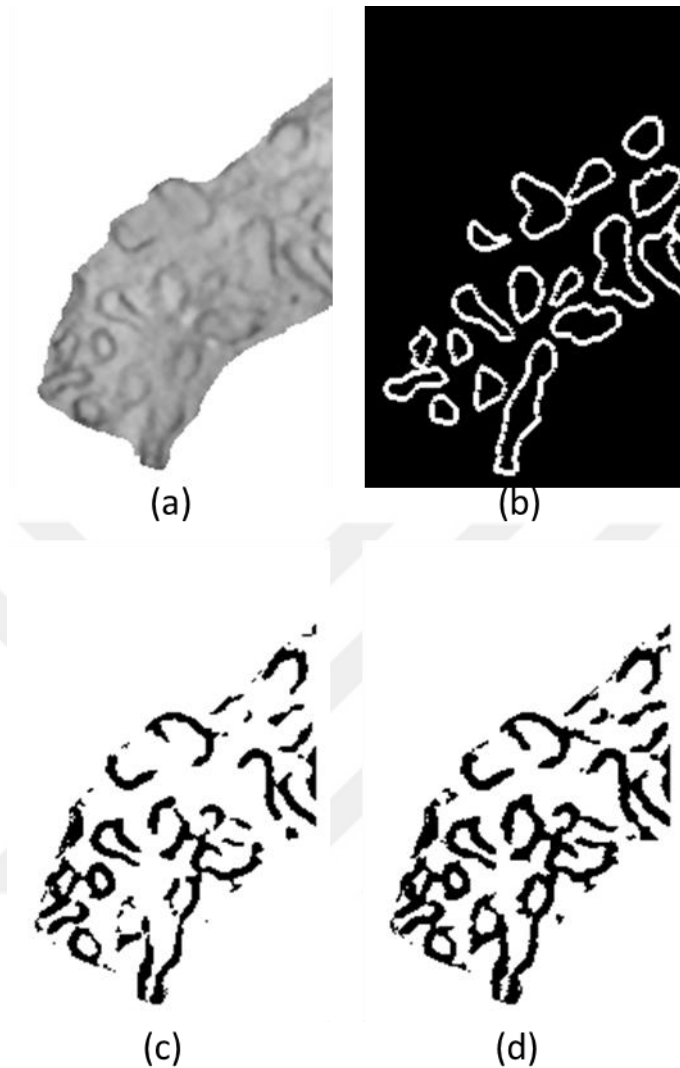
Figure 64. Results from gap-18 data set (a) is the original image (from gap-18 data set and obtained by means of cropping around mitochondria inner membrane) that is used as input to the trained network which produced the result in (c). (b) is the ground truth image and (c) is the result of directional growing method which used the grayscale version of the network result seen in (c).

4.3. Qualitative Evaluation of Results

We presented our results to Dr Guy Perkins in order to have an evaluation for the quality of results. Mr. Perkins is a known authority on the structure of the mitochondria and the cristae and researcher at National Center for Microscopy and Imaging Research. His comments on our study is as follows:

"The manual segmentation is very, very good! The ANN auto-segmentation is also very good, both 2D and 3D, with the 3D being slightly better than the 2D. In fact, in once sense, the auto-segmentation is more accurate than the manual segmentation because it shows only the membranes that are contrasted, i.e., have significant signal-to-noise. As you know, the human brain wants to complete, i.e., close the contour of a boundary curve. Another way to say it is to "fill in the gaps." We do this because we

80

know from many years of electron microscope imaging of biological membranes, and in particular mitochondrial membranes, that they should be "closed" structures in order for them to function properly as specialized compartments housing enzymes and metabolites that must interact with each other, or be in close proximity to cascading events, and be kept from diffusing too far away from their interaction partners. However, human error may introduce inaccuracies in filling in the gaps or closing the contour of mitochondrial membranes that have such low contrast that they are essentially invisible. Furthermore, even when the membranes are well-contrasted, human error may not trace the membrane exactly in the centre, i.e., the hand may stray when tracing. For these reasons, when given the opportunity, I would choose to display the auto-segmented versions in papers, videos and conference presentations in addition to any manual segmentation that may be used to distinguish different cristae. At times, it will be necessary to manually edit the auto-segmented version to close the contours in order to use modelling software, such as mcell, that require closed contours.

Because a mitochondrion may have anywhere from a few to more than 100 cristae, segmenting the cristae is by far the most time-consuming step in manual segmentation of mitochondria. Other programs exist that with varying degrees of accuracy and semi-automated segmentation can segment the rather topologically simple outer mitochondrial membrane, but your programs are the only ones that have had success in segmenting the much more topologically complex cristae membranes. Thus, your effort will be essential to accelerate our efforts in studying the structure/function relationships of cristae and their macromolecular components."

4.4. Computational Cost

One of the most important issues when training neural networks with massive number of training patterns is the huge magnitude of the computation power required. In this study a machine with Intel® Core ™ i7-3720QM CPU @ 2.60GHz processor and 16 Gigabyte of Random Access Memory is used. Computation power directly effects the time of the training and testing of the neural networks. The whole process (including 10 consecutive trainings) for one single network architecture takes approximately 35-40 minutes for a 3D neural network. The process includes the generation of the training set, training the network, testing with a sample image, and saving the results. In Table 14 below, the time spent for training and testing of one single 3D network can be seen. It is seen that most of the time is spent for generation of the training patterns. Also it is seen that the results can be obtained within a trice with the trained network.

Table 14. Time stamps during training of a single network. The first row is for a 3D network while the second row is for 2D. (All units are in seconds)

	Generate Training Set	Train	Test
NN1-iw1-15-HN-20-W5-B1-WOA-CP9	739,2	78,8	1,5
NN1-iw1-15-HN-20-W1-B1-WOA-CP9	294,3	22,3	0,7

CHAPTER 5

CONCLUSION, DISCUSSION & FUTURE WORK

A method with the aim of obtaining a robust detection of cristae membranes is proposed in this study. The method mainly comprise of two stages; a neural network that is aimed to segment the membrane boundaries, and a special post-processing method that increases the performance of basic thresholding applied to the continuous result of the network.

The ground truth needed for training of the network is also generated by means of manually marking the EMT images by tracing the cristae membrane boundaries. This took an important portion of the time spent for this study since labelling the EMT images needs a huge human labour dependent effort. Also the ground truth, which is modifiable and ready for fine tuning by a specialist (with IMOD software), generated in this study can be used for future studies which we see this as one of the remarkable contributions of this study.

In order to obtain the parameter set of the network, many networks are trained with different parameter sets. The results are demonstrated and interpreted and the best performing amongst all is selected. Also it is showed that using the advantageous 3D nature of EMT images in the training set increases the performance of the networks.

The restrictions of classical thresholding approach in the post-processing stage which is used for obtaining the binary classification results of neural networks is indicated and a special method to be used instead of basic thresholding is proposed. The method is called 'directional hessian growing' and showed to be performing better than the basic thresholding method.

This study showed that a single hidden layer network with a carefully selected set of parameters can robustly segment the cristae membranes even if they are weakly represented in the EMT images. But a single network itself is not powerful enough to obtain a continuous membrane boundary and detect the disconnected boundaries that are almost invisible. For this, an additional post-processing method, 'Directional Growing', is applied. Although the method was successful in connecting the boundaries for the vast majority of cristae, still there exists exceptional cases that the cristae boundary is disconnected. In order to obtain fully connected boundaries for each cristae active contour methods may be applied after directional growing stage of the algorithm.

In order to obtain the best performance from the 'directional growing' method, for each data set a manual parameter adjustment is required. This makes the algorithm semi-automatic for the human interference needed at this stage. An algorithm that automatizes this process depending on the characteristics of the data set to be used may be proposed.

A vast amount of the ground truth used in this study is not manually marked by a specialist on the morphology of mitochondria and cristae. So, the ground truth used for the training of the networks and for testing the methods proposed may be including

false markings. The performance of the networks trained may increase if the ground truth is fine-tuned by a specialist on the morphology of mitochondria and cristae.

Some features that help describe whether the pixel under consideration belongs to a corner, edge, ridge etc. can be obtained from the Hessian matrix. There exists many interpretations of these features in medical image segmentation literature like (Taşel, Mumcuoglu, Hassanpour, & Perkins, 2016), (Sener, Mumcuoglu, & Hamcam, 2016), (Frangi A. , Niessen, Hoogeveen, Van Walsum, & Viergever, 1999), (Koller, Gerig, Szekely, & Dettwiler, 1995), and (Lesage, Angelini, Bloch, & Funka-Lea, 2009). Using these features may have a positive effect especially for eliminating the pixels that pass the strong boundary test. Eigenvalues of the Hessian matrix which are widely used in the literature can be used to determine whether the selected eigenvector actually represents the principal direction of the boundary curvature or not. If the ratio between the eigenvalues of the selected and ignored vectors are close to one then this means that no dominant direction exits where the pixels in these regions do not belong to a membrane boundary.

Some studies (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012) which use neural networks in the medical image analysis literature use the advantage of isotropic nature of the data in order to increase the number of training patterns. By means of rotating each input window, the number of positive and negative patterns to be used in the training can be increased so that the training performance of the networks may be increased. This approach is much more beneficial when the data has three directions since there exist one more axis to rotate the input windows.

Another approach for increasing the performance of the networks with the use of same data is to exaggerate the patterns to be used by means of foveation and non-uniform sampling. These techniques allow to have an information covering a large area while keeping the input window size in the input layer of the network smaller. Applying these techniques to the input patterns while training of the networks may increase the performance of the networks.

The images used in this study are resampled so that the all of the pixels are set to a fixed size of 2 nm. But resampling is not applied through third direction so that the voxel size of 3D images differs for three different data sets used during training. Resampling the images in 3D and setting all of the voxels in the images to fixed size may increase the performance of the networks trained.

It is shown that averaging or combining many networks trained with exactly the same training set increases the performance of the network (Perrone & Cooper, 1993) (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012). Since the best performing network amongst ten consecutive trainings is selected and used in this study, combining these ten networks may increase the performance of the network.

Activation function in the output layer of neural networks trained in this study is a sigmoid function which has limits at 0 and 1 so that although the output of the function rapidly saturates to values very close 0 or 1, it can never exactly reach to them. This may result oscillation of the calculated error in the backpropagation algorithm since the ground truth values are 0 or 1. If the ground truth values are set to values that are very close to 0 and 1, but not exactly them, the training performance may be increased.

Convolutional neural networks (CNN) may perform better than many methods known in the literature for image classification tasks. This is also valid for medical image segmentation tasks. Using CNN instead of feed forward neural networks may increase the performance of the method proposed in this study especially for weak boundary regions. But in order to train a CNN, a much larger training set and computational power would be required.

Segmentation with sequentially trained neural networks usually yields to better performing networks (Jurrus, et al., 2010). A serial neural network architecture may decrease the number of false positives in the output of the networks so that the performance of the directional growing for connecting the disconnected boundaries may be increased.

REFERENCES

Acehan, D., Xu, Y., Strokes, D. L., & Schlame, M. (2007). Comparison of lymphoblast mitochondria from normal subjects and patients with Barth syndrome using electron microscopic tomography. *Laboratory investigation, 87(1)*, 40-48.

Alexander, C., Votruba, M., Pesch, U. E., Thiselton, D. L., Mayer, S., Moore, A., . . . Bhattacharya, S. S. (2000). OPA1, encoding a dynamin-related GTPase, is mutated in autosomal dominant optic atrophy linked to chromosome 3q28. *Nature genetics, 26(2)*, 211-215.

Baloyannis, S. J. (2006). Mitochondrial alterations in Alzheimer's disease. *Journal of Alzheimer's disease 9(2)*, 119-126.

Bazan, C., Miller, M., & Blomgren, P. (2009). Structure enhancement diffusion and contour extraction for electron tomography of mitochondria. *Journal of structural biology, 166(2)*, 144-155.

Beale, M. H., Hagan, M. T., & Demuth, H. B. (2014). *Neural Network Toolbox™ User's Guide." R2014a ed 2014*.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(8)*, 1798-1828.

Bernsen, J. (1986). Dynamic thresholding of grey-level images. *International conference on pattern recognition*, 1251-1255.

Bishop, C. M. (2006). Pattern Recognition. In *Machine Learning*.

Boykov, Y., & Funka-Lea, G. (2006). Graph cuts and efficient ND image segmentation. *International journal of computer vision, 70(2)*, 109-131.

Brandner, K., Mick, D. U., Frazier, A. E., Taylor, R. D., Meisinger, C., & Rehling, P. (2005). Taz1, an outer mitochondrial membrane protein, affects stability and assembly of inner membrane protein complexes: implications for Barth Syndrome. *Molecular biology of the cell, 16(11)*, 5202-5214.

Brice, C. R., & Fennema, C. L. (1970). Scene analysis using regions. *Artificial intelligence, 1(3)*, 205-226.

Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, (6)*, 679-698.

Chow, C., & Kaneko T. (1972). Automatic boundary detection of the left ventricle from cineangiograms. *Computers and biomedical research, 5(4)*, 388-410.

Churchland, P., & Sejnowski, T. (1992). *The computational brain.* Cambridge: MIT Press.

Ciresan, D., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems* (pp. 2843-2851).

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(5*, 603-619.

Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1995). Active shape models-their training and application. *Computer vision and image understanding, 61(1)*, 38-59.

Cremers, D., Rousson, M., & Deriche, R. (2007). A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International journal of computer vision, 72(2)*, 195-215.

Daems, W. T., & Wisse, E. (1966). Shape and attachment of the cristae mitochondriales in mouse hepatic cell mitochondria. *Journal of ultrastructure research, 16(1)*, 123-140.

Dimmer, K. S., Navoni, F., Casarin, A., Trevisson, E., Endele, S., Winterpacht, A., . . . Scorrano, L. (2008). LETM1, deleted inWolf Hirschhorn syndrome is required for normal mitochondrial morphology and cellular viability. *Human molecular genetics, 17(2)*, 201-214.

Dunne, R. A. (2007). *A statistical approach to neural networks for pattern recognition (Vol. 702).* John Wiley & Sons.

Exner, N., Treske, B., Paquet, D., Holmström, K., Scihieslig, C., Gispert, S., . . . Krüger, R. (2007). Loss-of-function of human PINK1 results in mitochondrial pathology and can be rescued by parkin. *The Journal of Neuroscience, 27(45)*, 12413-12418.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision, 59(2)*, 167-181.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision, 59(2)*, 167-181.

Frangi, A., Niessen, W. J., Vincken, K. L., & Viergever, M. A. (1998). Multiscale vessel enhancement filtering. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI'98* (pp. 130-137). Berlin Heidelberg: Springer.

Frangi, A., Niessen, W., Hoogeveen, R., Van Walsum, T., & Viergever, M. (1999). Model-based quantitation of 3-D magnetic resonance angiographic images. *IEEE Transactions on medical imaging, 18(10)*, 946-956.

Frank, J. (1992). *Electron tomography.* New York: Londan: Plenum.

Frey, T. G., Renken, C. W., & Perkins, G. A. (2002). Insight into mitochondrial structure and function from electron tomography. *Biochimica et Biophysica Acta (BBA)-Bioenergetics 1555.1*, 196-203.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural networks, 2(3*, 183-192.

Gabor, Z., & Kunz, W. S. (2013). Mitochondrial involvement in neurodegenerative diseases. *IUBMB life, 65(3)*, 263-272.

Glodstein, J., Newbury, D., Echlin, P., Joy, D., Roming, A., & Lyman, C. (2012). *Scanning electron microscopy and X-ray microanalysis: a text for biologists, materials scientists, and geologists.* Springer Science & Business Media.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing.*

Han, S. M., Lee, T. H., Mun, J., Kim, M. J., Kritikou, E. A., Lee, S.-J., . . . Koo, H.-S. (2006). Deleted in cancer 1 (DICE1) is an essential protein controlling the topology of the inner mitochondrial membrane in C. elegans. *Development 133*, 3597–3606.

Haykin, S. (2009). *Neural Networks and Learning Machines.* Pearson.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological approach.* John Wiley & Sons.

Hoover, A., Kouznetsova, V., & Goldbaum, M. (2000). Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *Medical Imaging, IEEE Transactions on, 19(3)*, 203-210.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks, 2(5)*, 359-366.

Horowitz, S., & Pavlidis, T. (1976). Picture segmentation by a tree traversal algorithm. *Journal of the ACM (JACM), 23(2)*, 368-388.

https://www.jic.ac.uk/microscopy/intro_EM.html. (n.d.). *Last seen in 06.05.2016*.

Isard, M., & Blake, A. (1998). Condensation—conditional density propagation for visual tracking. *International journal of computer vision, 29(1)*, 5-28.

Jain, V., Joseph, M., Roth, F., Turaga, S., Zhigulin, V., Briggman, K., & Helmstaedter, M. (2007, October). Supervised learning of image restoration with convolutional networks. *IEEE 11th International Conference on Computer Vision.* IEEE.

Jurrus, E., Paiva, A. R., Watanabe, S., Anderson, J. R., Jones, B. W., Whitaker, R. T., . . . Tasdizen, T. (2010). Detection of neuron membranes in electron microscopy images using a serial neural network architecture. *Medical image analysis, 14(6)*, 770-783.

Jurrus, E., Watanbe, S., Guily, R. J., Paiva, A. R., Ellisman, M. H., Jorgensen, E. M., & Tasdizen, T. (2013). Semi-automated neuron boundary detection and nonbranching process segmentation in electron microscopy images. *Neuroinformatics, 11(1)*, 5-29.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International journal of computer vision, 1(4)*, 321-331.

Khan, M. (2012). *The Transimission Electron Microscope.* Rijeka, Crotia: InTech.

Kobashi, S., Kamiura, N., Hata, Y., & Miyawaki, F. (2001). Volume-quantization-based neural network approach to 3D MR angiography image segmentation. *Image and Vision Computing, 19(4)*, 185-193.

Kohl, H., & Reimer, L. (2008). *Transmission Electron Microscopy: Physics of Image Formation.* New York: Springer-Verlag.

Koller, T., Gerig, G., Szekely, G., & Dettwiler, D. (1995). Multiscale detection of curvilinear structures in 2-D and 3-D image data. *Fifth International Conference on Computer Vision* (pp. 864-869). IEEE.

Krawczak, M. (2013). *Multilayer Neural Networks.* Springer.

Kremer, J. R., Mastronarde, D. N., & McIntosh, R. J. (1996). Computer visualization of three-dimensional image data using IMOD. *Journal of structural biology, 116(1)*, 71-76.

Kwok, S. M. (2004). Automatic pectoral muscle segmentation on mediolateral oblique view mammograms. *Medical Imaging, IEEE Transactions on, 23(9)*, 1129-1140.

Lai, M. (2015). Deep Learning for Medical Image Segmentation. *arXiv preprint arXiv:1505.02000.*

Lawrance, S., Giles, L. C., & Tsoi, C. A. (1996). *What size neural network gives optimal generalization? Convergence properties of backpropagation.* Institute for Advanced Computer Studies, University of Maryland.

Lesage, D., Angelini, E., Bloch, I., & Funka-Lea, G. (2009). A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical image analysis, 13(6)*, 819-845.

Levenberg, K. (1944). A method for the solution of certain non–linear problems in least squares. *Quarterly Applied Matematics (2)*, 164-168.

Lucchi, A., Smith, K., Achanta, R., Knott, G., & Fua, P. (2012). Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features. *Medical Imaging, IEEE Transactions on, 31(2)*, 474-486.

Lyman, C., Newburry, D., Goldstein, J., Williams, D., & Roming, A. (2012). *Scanning Electron Microscopy, X-Ray Microanalysis, and Analytical Electron.* Springer Science & Business Media.

Malik, J., Belongie, S., Leung, T., & Shi, J. (2001). Contour and texture analysis for image segmentation. *International journal of computer vision, 43(1)*, 7-27.

Manella, C. A., Marco, M., Penczek, P., Bernard, D., & Frank, J. (1994). The internal compartmentation of rat-liver mitochondria: Tomographic study using the high-voltage transmission electron microscope. *Microscopy research and technique, 27(4)*, 278-283.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics, 11(2)*, 431-441.

Marr, D., & Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London B: Biological Sciences, 207(1167)*, 187-217.

Martinez-Sanchez, A., Garcia, I., & Fernandez, J.-J. (2011). A differential structure approach to membrane segmentation. *Journal of structural biology, 175(3)*, 372-383.

Masomi, H., Behrad, A., Pourmina, M. A., & Roosta, A. (2012). Automatic liver segmentation in MRI images using an iterative watershed algorithm and artificial neural network. *Biomedical Signal Processing and Control, 7(5)*, 429-437.

McBride, H. M., Neuspiel, M., & Wasiak, S. (2006). Mitochondria: more than just a powerhouse. *Current Biology, 16(14)*, R551-R560.

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.

Middleton, I., & Damper, R. I. (2004). Segmentation of magnetic resonance images using a combination of neural networks and active contour models. *Medical engineering & physics, 26(1)*, 71-86.

Mishchenko, Y. (2009). Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs. *Journal of neuroscience methods, 176(2)*, 276-289.

Mortensen, E., & Barret, W. (1995). Intelligent scissors for image composition. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (pp. 191-198). ACM.

Mumcuoglu, E. Ü., Hassanpour, R., Taşel, S. F., Perkins, G., Martone, M. E., & Gurcan, M. N. (2012). Computerized detection and segmentation of mitochondria on electron microscope images. *Journal of microscopy 246.3*, 248-265.

Munn, E. A. (2014). *The structure of mitochondria.* Academic Press.

Nakagawa, Y., & Rosenfeld, A. (1979). Some experiments on variable thresholding. *Pattern recognition, 11(3)*, 191-204.

Ohlander, R., Price, K., & Reddy, D. (1978). Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing, 8(3)*, 313-333.

Palade, G. E. (1953). An electron misroscope study of the mitochondrial structure. *Journal of Histochemistry & Cytochemistry 1.4*, 188-21.

Paragios, N. (2003). A level set approach for shape-driven segmentation and tracking of the left ventricle. *Medical Imaging, IEEE Transactions on, 22(6)*, 773-776.

Perrone, M., & Cooper, L. (1993). When networks disagree: Ensemble methods for hybrid neural networks. *BROWN UNIV PROVIDENCE RI INST FOR BRAIN AND NEURAL SYSTEMS.*, No. TR-61.

Rabl, R. (2009). Formation of cristae and crista junctions in mitochondria depends on antagonism between Fcj1 and Su e/g. *The Journal of cell biology*, 1047-1063.

Rajesh, N., Ouyan, H., Gray, A., McLaughlin, S., & Subramaniam, S. (2009). Automatic joint classification and segmentation of whole cell 3D images. *Pattern Recognition, 42(6)*, 1067-1079.

Rog, T., Martinez-Seara, H., Munck, N., Oresic, M., Karttunne, M., & Vattulainen, I. (2009). Role of cardiolipins in the inner mitochondrial membrane: insight gained through atom-scale simulations. *The Journal of Physical Chemistry B, 113(11)*, 3413-3422.

Rogowska, J. (2000). Overview and fundamentals of medical image segmentation. In *Handbook of medical imaging* (pp. 69-85). Academic Press, Inc.

Rosenfeld, A., & Davis, L. S. (1979). Image segmentation and image models. *Proceedings of the IEEE, 67(5)*, 764-772.

Salotti, M. (2001). An efficient algorithm for the optimal polygonal approximation of digitized curves. *Pattern Recognition Letters, 22(2)*, 215-221.

Scheffler, I. (2008). Basic molecular biology of mitochondrial replication. In *Drug-Induced Mitochondrial Dysfunction* (pp. 37-70). Wiley.

Sebastian, T. B., Tek, H., Crisco, J. J., & Kimia, B. B. (2003). Segmentation of carpal bones from CT images using skeletally coupled deformable models. *Medical Image Analysis, 7(1)*, 21-45.

Sener, E., Mumcuoglu, E., & Hamcam, S. (2016). Bayesian segmentation of human facial tissue using 3D MR-CT information fusion, resolution enhancement and partial volume modelling. *Computer methods and programs in biomedicine, 124*, 31-44.

Seyedhosseini, M., Ellisman, M. H., & Tasdizen, T. (2013, April). Segmentation of mitochondria in electron microscopy images using algebraic curves. In *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium* (pp. 860-863). IEEE.

Shareef, N., Wang, D. L., & Yagel, R. (1999). Segmentation of medical images using LEGION. *Medical Imaging, IEEE Transactions on, 18(1)*, 74-91.

Shen, L., Rangayyan, R. M., & Desautels, J. L. (1993). Detection and classification of mammographic calcifications. *International Journal of Pattern Recognition and Artificial Intelligence, 7(06)*, 1403-1416.

Sjöstrand, F. S. (1953). Electron microscopy of mitochondria and cytoplasmic double membranes: ultra-structure of rod-shaped mitochondria. *Nature 171*, 30-31.

Soares, J. V., Leandro, J. J., Cesar, J. R., Jelinek, H. F., & Cree, M. J. (2006). Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification. *Medical Imaging, IEEE Transactions on, 25(9)*, 1214-1222.

Suzuki, K., Abe, H., MacMahon, H., & Doi, K. (2006). Image-processing technique for suppressing ribs in chest radiographs by means of massive training artificial neural network. *IEEE Trans. Med. Imaging, 25(4)*, 406-416.

Suzuki, K., Amato III, S. G., Li, F., Sone, S., & Doi, K. (2003). Massive training artificial neural network (MTANN) for reduction of false positives in computerized detection of lung nodules in low-dose computed tomography. *Medical Physics, 30(7)*, 1602-1617.

Suzuki, K., Amato III, S. G., Li, F., Sone, S., & Doi, K. (2003). Massive training artificial neural network (MTANN) for reduction of false positives in computerized detection of lung nodules in low-dose computed tomography. *Medical Physics, 30(7)*, 1602-1617.

Suzuki, K., Horiba, I., Sugie, N., & Nanki, M. (2004). Extraction of Left Ventricular Contours From Left. *Medical Imaging, IEEE Transactions on, 23(3)*, 330-339.

Szeliski, R. (2010). *Computer vision: algorithms and applications.* Springer Science & Business Media.

Şener, E. (2012). *AUTOMATIC BAYESIAN SEGMENTATION OF HUMAN FACIAL TISSUE USING 3D MR-CT FUSION BY INCORPORATING MODELS OF MEASUREMENT BLURRING, NOISE AND PARTIAL VOLUME.* Thesis Submitted tı Informatics Institute, METU.

Şener, E., Mumcuoğlu, E. Ü., & Hamcan, S. (2016). Bayesian segmentation of human facial tissue using 3D MR-CT information fusion, resolution enhancement and partial volume modelling. *Computer methods and programs in biomedicine, 124*, 31-44.

Taşel, F. S. (2012). *Detection and Segmentation of Mitochondria From Electron Microscope Tomography Images.*

Taşel, F. S. (2012). PHD Proposal. In *Detection and segmentation of mitochondria from electron microscope tomography images.*

Taşel, S. F., Mumcuoglu, E. Ü., Hassanpour, R., & Perkins, G. (2016). A validated active contour method driven by parabolic arc model for detection and segmentation of mitochondria. *Journal of structural biology.*

Tomasi, C., & Manduchi, R. (1998, January). Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision* (pp. 839-846). IEEE.

Trifunovi, A., Wredenberg, A., Falkenberg, M., Spelbrink, J. N., Rovio, A., Bruder, C. E., & Törnell, J. (2004). Premature ageing in mice expressing defective mitochondrial DNA polymerase. *Nature, 429(6990)*, 417-423.

Trimmer, P. A. (2000). Abnormal mitochondrial morphology in sporadic Parkinson's and Alzheimer's disease cybrid cell lines. *Experimental neurology, 162(1)*, 37-50.

Verschoor, M. L., Ungard, R., Harbottle, A., Jakupciak, J. P., Parr, R. L., & Singh, G. (2013). Mitochondria and cancer: past, present, and future. *BioMed research international*.

Weszka, J. S. (1978). A survey of threshold selection techniques. *Computer Graphics and Image Processing, 7(2)*, 259-265.

Yoshua, B., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(8)*, 1798-1828.

Zhu, Y., & Yan, H. (1997). Computerized tumor boundary detection using a Hopfield neural network. *Medical Imaging, IEEE Transactions on, 16(1)*, 55-67.

Zick, M., Rabl, R., & Reichert, A. S. (2009). Cristae formation—linking ultrastructure and function of mitochondria. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research, 1793(1)*, 5-19.

APPENDIX A

A Preliminary Work: Segmentation on the Whole Image Including both Interior and Exterior of Mitochondria

In the very early stages of this thesis study; it is aimed to segment the cristae from the whole EMT images without cropping a region of interest area (which is the interior region of a mitochondrion). This problem has its own problems in addition to the problems when only segmenting the interior region of the mitochondria. One of them is that almost all boundary membranes of the cell has very similar properties (i.e: the width, the intensity variation, and first or second order gradients through curvature direction etc.). An example of a whole EMT image can be seen below.
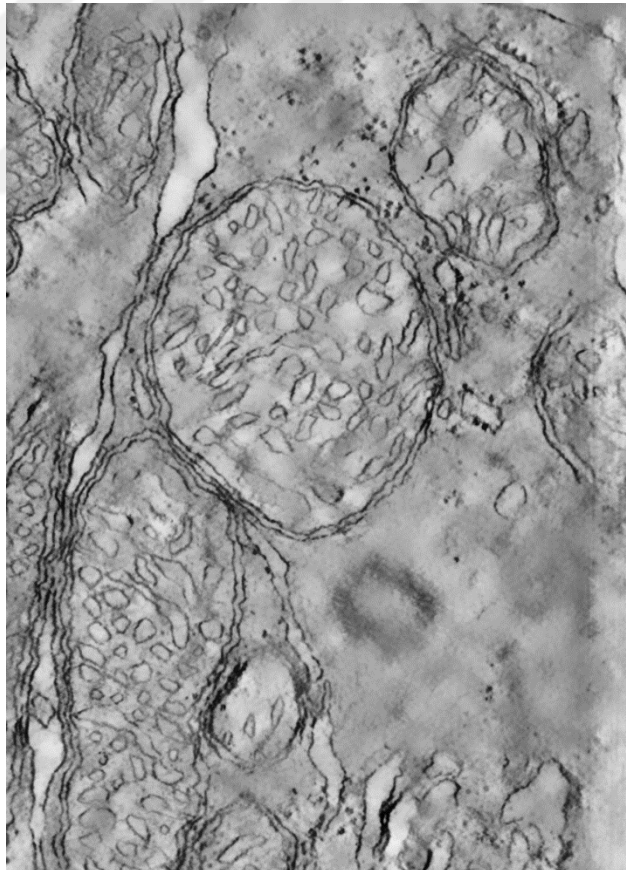
In order to obtain a neural network that can segment cristae boundaries while not segmenting the mitochondria or endoplasmic reticulum boundaries a classical feed forward neural network architecture is used. Both one and two hidden layered

networks were not successful enough for discriminating the mitochondria boundaries from cristae membrane boundaries. Grayscale results of one hidden layered networks can be seen in the figure below.

iw: 27 - hn: 100   iw: 21 - hn: 100

Grayscale results of two hidden layered networks can be seen in the figure below.

iw: 27 − hn1: 55 − hn2: 30   iw: 21 − hn1: 55 − hn2: 30

As seen from the figures above, both networks trained with a classical approach are prone to produce too many false positives. In order to overcome this issue a serial neural network architecture is proposed. In this architecture the basic idea is to force the first neural network to learn more low level abstractions on the data. Low level abstractions help the first network to extract the edges and clean the small membranes in the image and transform the image to a more easily understandable form so that the second network can learn more high level features like double ridges, connected curves etc. Consequently the second network can distinguish the cristae boundaries from i.e. mitochondria boundary.

In this training method each network uses the inputs produced by the previous network. Serial neural networks can over-perform the classical feedforward neural networks because they use the deduced context produced by the previous network so that in each step the detection accuracy is improved (Jurrus, et al., 2013). The first network uses the intensity values of the input stencil just as the classical neural networks, but the remaining networks in the series uses the input vectors extracted from the output of the previous network.

In this preliminary study, a two stage serial network is trained. By the help of the two stage serial network, the false positives produces by the first network is reduced to an acceptable ratio in the second neural network. The flowchart of the two stage serial network during the training and testing can be seen in Figure below.
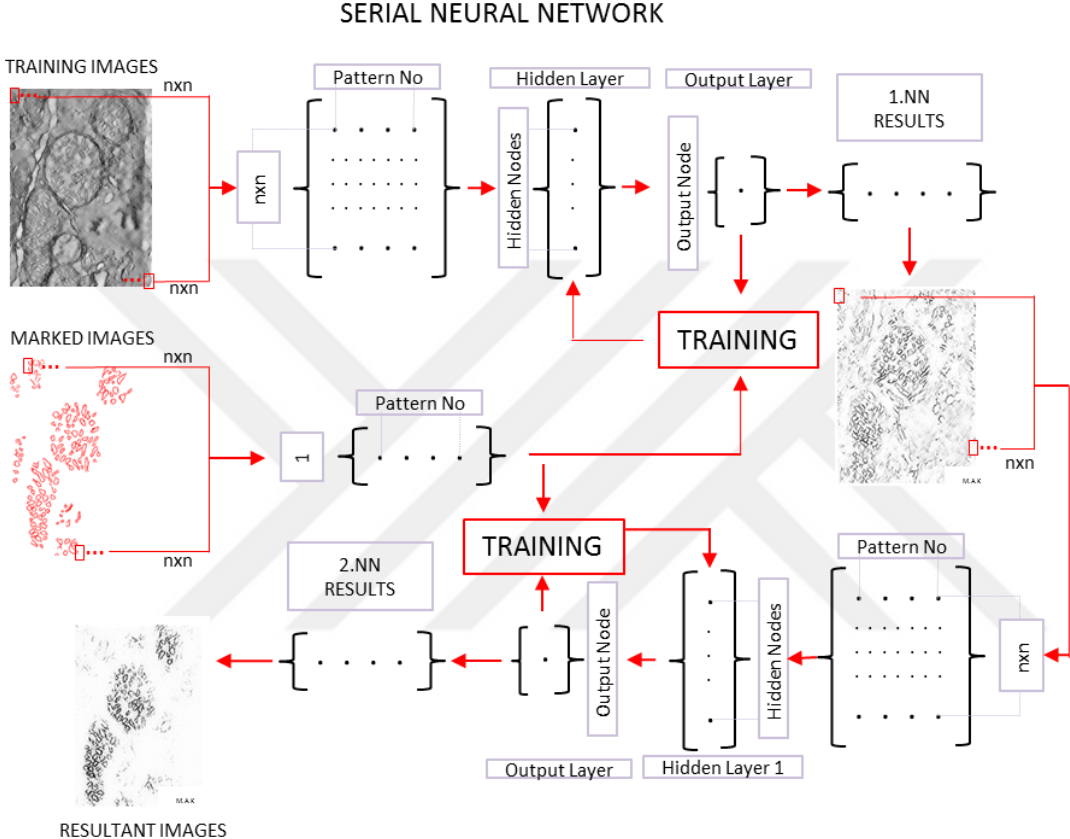
Grayscale results of two different serial neural networks can be seen in the figure below. It can be seen that membranes that are not cristae throughout the whole cell are discriminated from cristae and not marked.

1st NN's input window size: 15 & Hidden Nodes: 55
AND 2nd NN's Hidden Nodes: 120

2nd NN input window: 21

2nd NN input window: 27