SEGMENTATION OF MEDIAN NERVE BY SIGNAL PROCESSING AND ARTIFICIAL INTELLIGENCE ON ULTRASOUND IMAGES

KUENZANG THINLEY

A Thesis Submitted to the Graduate School of Naresuan University
in Partial Fulfillment of the Requirements
for the Master of Engineering in Electrical Engineering
2021

SEGMENTATION OF MEDIAN NERVE BY SIGNAL PROCESSING AND ARTIFICIAL INTELLIGENCE ON ULTRASOUND IMAGES

KUENZANG THINLEY

A Thesis Submitted to the Graduate School of Naresuan University
in Partial Fulfillment of the Requirements
for the Master of Engineering in Electrical Engineering
2021

Thesis entitled "Segmentation of Median Nerve by Signal Processing and Artificial Intelligence on Ultrasound Images"

By KUENZANG THINLEY

has been approved by the Graduate School as partial fulfillment of the requirements for the Master of Engineering in Electrical Engineering of Naresuan University

Oral Defense Committee

.... Chair
(Professor Kosin Chamnongthai, Ph.D.)

.... Advisor
(Associate Professor Surachet Kanprachar, Ph.D.)

.... Co Advisor
(Settha Tangkawanit, Ph.D.)

.... Internal Examiner
(Chairat Pinthong, Ph.D.)

Approved

....
(Associate Professor Krongkarn Chootip, Ph.D.)

Dean of the Graduate School

Title	SEGMENTATION OF MEDIAN NERVE BY SIGNAL PROCESSING AND ARTIFICIAL INTELLIGENCE ON ULTRASOUND IMAGES
Author	KUENZANG THINLEY
Advisor	Associate Professor Surachet Kanprachar, Ph.D.
Co-Advisor	Settha Tangkawanit, Ph.D.
Academic Paper	M.Eng. Thesis in Electrical Engineering, Naresuan University, 2021
Keywords	U-Net, Median Nerve, Carpal Tunnel Syndrome (CTS), Signal Processing, Convolutional Neural Network

ABSTRACT

In medical sciences, visualizing the internal dynamics and structures of the human body throughout bodily function is critical, and ultrasound imaging (US) is one of the most extensively utilized medical imaging technologies. Carpal tunnel syndrome (CTS) is a kind of peripheral neuropathy, a frequently occurring disease in the wrist that affects many people. When the median nerve is compressed within the carpal tunnel, it produces a variety of nerve function problems that manifest as CTS. In this study, automatic segmentation of median nerve using signal processing and convolutional neural network (CNN)-based methods were studied.

In signal processing, mathematical morphology, edge detection, and contouring are employed, while in convolutional neural network (CNN), U-Net is used. The performance of signal processing techniques was engineered by concentrating on structural or kernel alterations for the signal processing method. The base U-Net, U-Net with pre-processed data, and U-Net with augmented and pre-processed data with batch norm layer are the three architectures evaluated in deep learning. The dice score, accuracy, Jaccard Similarity coefficient, recall, precision, and F1 score are all used to compare the results.

The signal processing technique observed a significant correlation of cross-correlation coefficient (CSA) between the ground truth (GT) and the segmented image with a close resemblance of over 90% with a correlation coefficient of 0.962

when tested on the 35 images. However, the model has estimated the CSA of the median nerve as normal in several situations, even when the expert or sonographer evaluated it as abnormal. The process of feature extraction, however, is where this technique's shortcoming lies. It took more time and processing to manually modify the kernel's weight and iterate numerous times to segment the median nerve. Furthermore, the approach was not reliable and favored certain feature images over others.

The U-Net model trained with pre-processed data, and augmented data with batch norm layer surpassed the two other models and achieves amazing results in median nerve segmentation. When evaluated on test datasets, an accuracy of 99.8% was achieved, which is 14.1 % higher than approach one (base U-Net) and 4.4% higher than approach two (U-Net with pre-processed data). The method was also quite successful in finding the median nerve, with a dice similarity coefficient (DSC) of 0.899, which was much higher than the other two approaches. This shows that when deep learning is given additional training data and the input data is cleaned, the outcomes are more accurate. This implies that data pre-processing and data augmentation are important not just for cleaning data and expanding the number of datasets, but also for improving accuracy. This demonstrates that this model could be used as a screening tool in clinical practice to expedite the identification, diagnosis, and assessment of CTS.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deepest appreciation to His Majesty the King of Bhutan and Naresuan University in Thailand for bestowing me with a scholarship to pursue my Master's studies. I would also like to express my thanks to the Electronics and Communication Department, College of Science and Technology, the Royal University of Bhutan for choosing me for the Master's program.

I would also like to acknowledge my advisor, Assoc. Prof. Dr. Surachet Kanprachar, and co-advisor, Dr. Settha Tangkawanit, for their guidance and support throughout the whole of my master's degree journey. I appreciate their assistance and many thought-provoking deliberations concerning my research, especially when it appears that there is no ultimate option. Dr. Hemwarun Deema, Specialist, Department Head, Rehabilitation Department, Fort Somdej Phra Naresuan Maharaj Hospital, Phitsanulok, Thailand, also deserves special thanks for providing the study with the data (ultrasound images). I am also grateful to my thesis committee members for going through and providing me the insightful and encouraging feedback.

My appreciation also extends to Mr. Jaruwat Patmanee, a lab mate who inspired me with a great deal of direction and inspiration during my study, and to all the persons who directly or indirectly assisted me in completing my study.

Finally, I want to gratitude to my family, particularly my parents, who believed in me, have always supported me, and never failed to remind me that I can be a better person.

KUENZANG THINLEY

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

1.1 Background

Our hands provide us with a great deal of assistance. Many of our daily tasks, such as writing, driving, cooking, and so on are made possible by our hands. They play a crucial role in who we are and how we view ourselves. The human hand is made up of palm, wrist, and fingers which consist of 27 bones, 27 joints, 34 muscles, over 100 ligaments, and tendons, and numerous blood arteries and nerves. The wrist has a complicated mechanism with many articulations and soft tissue components that keep our hands stable. Wrist functions include giving the hand flexibility and strength, transmitting forces from the hand, and moving the hand back and forth and side to side, and it inhouse many nerves and soft tissues. Hand nerves send electrical impulses from the brain to the muscles of the forearm and hand, allowing them to move. They also transport touch, pain, and temperature sensations from the hands to the brain. The ulnar nerve, radial nerve, and median nerve are the three major nerves of the hand and wrist which carry out different activities to keep our hand functioning. Each of the three nerves begins at the shoulder and travels down the arm to the hand. There are sensory and motor components to each of these nerves. Figure 1 shows the overall anatomy of the human wrist.

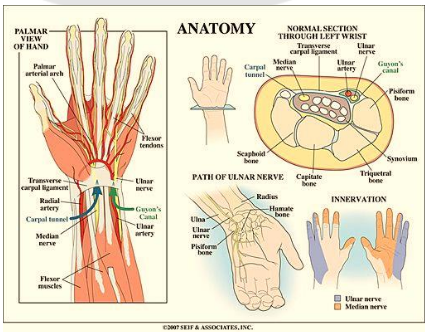


Figure 1 Anatomy of the Wrist in the Human Hand

Source: Image from (Mark E. Pruzansky)

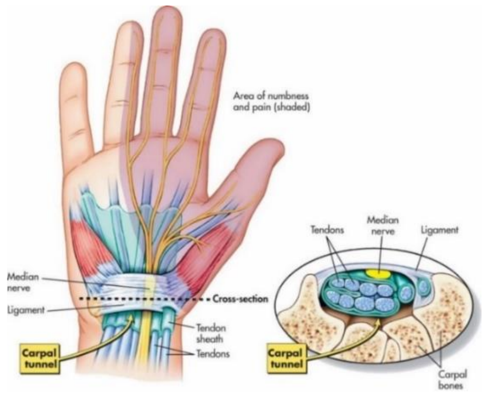


Figure 2 Anatomy of Carpal Tunnel in the Wrist

Source: Image from (Reed, 2005)

Carpal tunnel syndrome (CTS) is one of the frequently occurring diseases in the wrist. Carpal tunnel syndrome (CTS) is a kind of peripheral neuropathy that affects many people. When the median nerve is compressed within the carpal tunnel, it causes numerous abnormalities in nerve function that appear as CTS. CTS symptoms include numbness, weakness, and discomfort in the hand and wrist (Ferry et al., 1998). It also reduces blood supply to the nerve due to compression, adding to the physiological alterations that lead to nerve dysfunction (Topp & Boyd, 2006). CTS results in thickening of transverse carpal ligament, fibrotic alterations in the sub-synovial connective tissues, and reduces the carpal tunnel space (Yang et al., 2021). Symptoms are more common in the morning following a night's sleep or at night, with localization in the hand's extremities, however in more severe cases, discomfort spreads to the forearm, arm, and shoulder (Padua et al., 2016).

The carpal tunnel houses the median nerve (MN) as well as the tendons that bend the fingers (Figure 2). The MN gives motor innervation to the hand and wrist and also supplies sensation to the thumb, second &third finger, (although not the little finger) (Newington et al., 2015). The CTS condition affects 99 out of every 100,000 persons in the general population. Globally the prevalence rates range from 7% to 19% (Newington et al., 2015). According to the Royal College of Surgeons Commissioning Guide 2017, the prevalence of Carpal Tunnel Syndrome is expected to be between 7 % and 16 % in the UK, and incidence rates in the United States are about 5% (Dale et al., 2013). Although CTSs are common to all age groups, people older than 40 are the most common age group who are likely to suffer from CTS while, women account for about 65-75% of cases (Duncan & Kakinoki, 2017). Experts believe this is because women's carpal tunnels are smaller in comparison to a man. Women are also 3 to 5 times more likely than men to get Rheumatoid Arthritis, with the fact that women go through pregnancy (Kazantzidou et al., 2021). It also affects the working population, with a focus on manual labor, and also affects occupations that need sophisticated hands movements, such as musicians and dentists (Burton et al., 2018). Carpal tunnel syndrome is typically caused by a combination of events that increase pressure on the median nerve and tendons in the carpal tunnel, rather than a problem with the nerve itself. Swelling produced by a sprain or fracture to the wrist; an overactive pituitary gland; an underactive thyroid gland; All these conditions, as well as rheumatoid arthritis, are significant causes.

The primary and well-established examination techniques for the diagnosis of CTS are physical examination and electrophysiological nerve conduction studies (Ibrahim et al., 2012) and are considered the gold standard. However, electrophysiology-based diagnosis has been linked to a high prevalence of false-positive and false-negative findings due to the presence of other neurologic diseases, such as diabetic neuropathy (Gazioglu et al., 2011; Stevens, 1997), and also produces discomfort to the patients and the results are often ambiguous and inconclusive. Hence, a novel physical assessment procedure of the CTS, such as the manual tactile test and the pinch-holding-up activity evaluation, is created and presented, allowing for a further improvement in the sensitivity of clinical examination among a well-established clinical test.

Lately, the high-frequency ultrasound (US) imaging technique was extensively used and adapted in various medical fields to assess and diagnose the internal health of our human body. One such application is tissue characterization, i.e. assessment of musculoskeletal and neuromuscular diseases (Chang et al., 2017; Chang et al., 2018), as well as peripheral nerve pathologies (Im Suk et al., 2013). Ultrasound imaging has a sensitivity and specificity of up to 94 % and 98%, respectively (Duncan et al., 1999). Changes in the geometry of the median nerve and adjacent tissues, such as tendons, have been quantified using ultrasound imaging (Filius et al., 2015; Mhoon et al., 2012; Mohammadi et al., 2010; Moran et al., 2009).

The mean nerve cross-sectional area (CSA) of the median nerve (MNA) is the most often used parameter to measure CTS, with cut-off values diagnostic for MN pathology defined from 9 mm^2 to 14 mm^2 in various analytical settings (Möller et al., 2018; Seror, 2008; A. Torres-Costoso et al., 2018). However, the value of 9 mm^2 for MNA is the most trusted cut-off value for MN pathology during CTS as shown by NCS investigations (Tai et al., 2012).

Due to the great variety and complexity of medical images, as well as the fact that they are frequently contaminated by noise, automatic segmentation of medical images is a difficult undertaking. Over the previous two decades, the research community has made significant progress, with several methods for medical image segmentation. Some of these techniques include thresholding, clustering, edge, and region-based segmentation. Some of the reliable methods have been included in commercial software. They are, however, usually restricted to certain segmenting tasks like the segmentation of bones. Designing automated partitioning methods for more complicated organs is still a challenge. One such example is in the wrist, localizing and segmentation of median nerve in ultrasound images. Hence, the precise identification of these tissues is frequently the foundation for accurate diagnosis. Deep learning approaches based on various types of deep artificial neural networks have recently been effective and widely utilized in a variety of common computer vision applications, including image identification, classification, and segmentation. This work will concentrate on the segmentation of the median nerve in ultrasound images using signal processing and artificial intelligence. The results from these studies could be used as one of the parameters for the faster diagnosis of CTS by the doctor.

1.2 Problem Statement

One significant hindrance of ultrasonography in the diagnosis and detection of the morphology of median nerve is the sonographer's freedom to place a hand-held probe on the wrist (Kaymak et al., 2008). The type of transducer used, and its bandwidth also affect the proper diagnosis. As a result, even when probe alignment is consistent, interpretation of the data may differ since the results rely not only on objective nerve planimetry but also on interpreting nerve brightness as displayed on the screen in the US B-mode. The interpretation by the "human sight," on the other hand, leaves a significant range of subjectivity and is too slow to locate and detect the median nerve. As a result, there is a significant demand for a reliable measure that is fast and that can assist CTS diagnosis to replace the image's subjectivity and relativism.

Despite numerous good results in earlier attempts to characterize, segment, and detect median nerve from US-based imaging, one drawback of US nerve imaging techniques was, it has significant variability, both across patients and in scanning parameters (Byra et al., 2020), and also tracking median nerve areas of interest with ultrasonic imaging is difficult (Horng et al., 2020) and challenging as it required a very experienced and skilled operator to detect the median nerve from the US images due to the close closeness of many small bones and soft-tissue structures, as well as small articulations.

There had been an effort to develop an automatic detection of the median nerve in ultrasound images using a contour detection framework (Wang et al., 2015), however, the method was not robust and more biased to certain feature images. The reference contour remains a guiding factor in contour calculation and any faulty reference contour may result in segmenting the false area.

1.3 Purpose of the Study

Recently, the breakthrough and development in deep learning have revolutionized the field of computer vision in particular segmentation of the medical image, and are now a widely adopted technique in image processing tasks. Many breakthrough approaches have been developed over the years to address the diverse obstacles of traditional approaches in image processing using deep learning, and fresh research continues to lead to the creation of more creative and inventive solutions. A fundamental benefit of deep learning is its ability to quickly analyze vast and complex quantities of data and extract important image characteristics from the original data. It is made up of many processing levels. The main objective of these layers is to learn and extract all the image's fine details at multiple levels. As a result, deep learning has emerged as a must-have tool in the realm of image analysis.

One of the most prevalent and frequently used deep learning techniques is the convolutional neural network (CNN). Through a succession of approaches, CNN has been able to minimize the computer vision problem with a huge quantity of data and eventually train it. Some of its applications are automatic brain tumor segmentation (Khan et al., 2021; Menze et al., 2014; Pereira et al., 2016; Ranjbarzadeh et al., 2021), detection of breast cancer (Benhammou et al., 2020; Bhogal et al., 2021; Bychkov et al., 2021; Eskreis-Winkler et al., 2021; Gamble et al., 2021; Saber et al., 2021), liver (Chen et al., 2020; Hectors et al., 2021; Kiani et al., 2020), and so on. Recently it is also used in the early detection of Covid-19 (Dansana et al., 2020; Jia et al., 2021; Kassania et al., 2021; Mohammad-Rahimi et al., 2021).

One deep learning approach that has shown remarkable performance gains on a variety of medical image segmentation tasks, establishing a new state-of-the-art is U-net (Ronneberger et al., 2015). The use of U-Net in medical imaging has increased dramatically since its introduction in 2015. Researchers are now adopting new methodologies or blending other imaging modalities into the U-net model to make it more robust and to bring numerous improvements to U-net architecture. The U-Net's ingenious idea of using skip connection achieves high accuracy from a relatively small dataset. The main purpose of the study is the segmentation and localization of the median nerve in an ultrasound image using a U-Net model with the following objectives:

1. Train CNN model (U-Net) to segment the median nerve on Ultrasound images.
2. Apply preprocessing technique using signal processing technique to clean the input image and increase the accuracy of the model prediction.
3. Apply data augmentation technique to enhance the data by artificially adding diversity and variation to existing data.

1.4 Expected Outcome

This research has the expected outcome as follows:

1. Wrist ultrasound image datasets.
2. Able to localize median nerve in ultrasound images.

3. Able to segment median nerve in ultrasound images.

1.5 Significant of Study

This study will bring some uplift in the field of medical imaging especially in the segmentation of median nerve in ultrasound images to diagnose CTS. Some of them are:

- Precise segmentation and location of the median nerve in ultrasound images.
- Faster segmentation of the median nerve in the wrist by replacing the traditional method of manual segmentation with the use of deep learning.
- Provide robustness to the median nerve segmentation in ultrasound images.
- This study will benefit the clinical physicians or doctors to diagnose CTS at a faster rate. It will also reduce the dependency on skilled sonographers to detect median nerves in ultrasound images.

1.6 Scope of Study

The scope of the study is:

1. **Datasets**: For this study since there are no free available datasets online hence secondary data (ultrasound images) is used. The secondary data is collected from the Fort Somdej Phra Naresuan Maharaj Hospital, Phitsanulok, Thailand. However, no physical experiments on the patient or individual were involved in the process of the collection of data. We collected the ultrasound images of hand wrists which are already been experimented with in previous years and stored them in the database of the ultrasound machine. Moreover, this study only focuses on the segmentation of the distal median nerve type in the hand wrist, hence we collected only ultrasound images of this type. The secondary data (stored ultrasound image) does not contain any personal and medical information of the patient and those which contained some information were kept protected, confidential, and secret. The experiments were carryout by a trained specialist (doctor) who has 11 years of experience in the field of medicine and works in the rehabilitation clinic/department in Fort Somdej Phra Naresuan Maharaj Hospital, Phitsanulok, Thailand, and has a degree in doctor of medicine M.D with specialization in the rehabilitation medicine.

Before the collection of data, proper permission is asked from the higher management authorities of the hospital for the access of data for research.

2. **Study area**: Segmentation of median nerve in ultrasound images using convolutional neural network (CNN) a subfield of artificial intelligence (AI) and signal processing technique. This study adapted the supervised type learning method. In this type of learning each input data to the network is labeled and tagged with the desired output value, allowing the system to determine how the output will be when the input is received. To function well and to get reliable and accurate output, a supervised learning model sometimes necessitates the size of a large dataset from ground truth observations. These larger datasets which include a more historical example to learn from, allow the algorithms to include multiple cases and generate a model that can manage them.
3. **CNN model**: This study adopted the U-Net architecture (Ronneberger et al., 2015) for the segmentation task. It includes the training and validation results from pre-processing, use of activation, and cost function.
4. **Result verification and evaluation:** The accuracy, Jaccard similarity coefficient, Recall, Precision, F1 score, and Dice similarity coefficient (DSC) are used as performance measures in the assessment. The result is verified and compared with the ground truth images from the hospital which are manually segmented by a sonographer.

CHAPTER II

LITERATURE REVIEW AND RELATED THEORY

2.1 Introduction

Recently deep learning has been progressively adopted in a wide range of computational fields, consistently giving remarkable output on a range of challenging tasks. It outputs equal or even outperformed human performance in numerous situations. The ability to handle a large amount of complex data is the prime benefit of deep learning. It extended its popularity in numerous disciplines such as natural language processing, medicine, robotics, and more. Therefore, in the first sub-section of Chapter II, the holistic introduction of neural & DL is presented to understand the fundamentals of ML, and a comprehensive survey of the most recent trends and development in DL is outlined. Subsequently, it details the architecture layers and important terminology of convolutional neural network (CNN) and some of the neural network regularization methods. The second sub-section of Chapter II will present a review of some of the recent works focusing on image segmentation problems.

2.2 Basics of Artificial Neural Network

Artificial neurons are replicas of the biological neurons system (Figure 3). The roles and specialties of each component of a biological neuron are unique. Beginning with the neuron's center, the cell body or soma. The cell body contains genetic information and serves as the neuron's major source of energy.

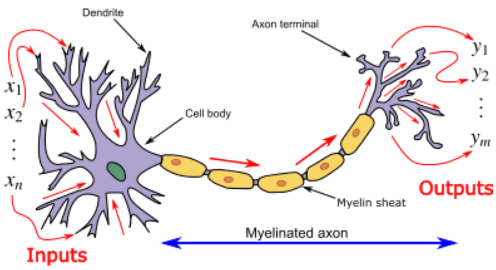


Figure 3 The Biological Neuron

Source: Wikipedia

Dendrites are the tree-like structure that branches out from the cell body and surrounds it. Its primary function is to receive signals from other neurons' axons via a biological process known as neurotransmission. The signal is subsequently processed and delivered to the next neurons through an axon, which is a long, tail-like structure. Many axons are surrounded and insulated by a fatty material called the myelin sheath, which helps to speed up the transmission process.

One of the most significant components of our nervous system is interconnected neurons. Action potentials are used by neurons to send and receive messages from and to one other. In a nutshell, it's the change in a neuron's electric potential induced by ion movement within the cell. This technique allows a sequence of neurons to send synapses, allowing us to do simple tasks like moving our arms, smelling objects, and so on.

Artificial neurons, like biological neurons, are linked to one another and stimulate one another via these connections. As the name implies, an artificial neural network is a network made up of artificial neuron units with connections between them. Perception is the first kind of artificial algorithm that was invented in 1958 by Frank Rosenblatt (Rosenblatt, 1958). It calculates the output by using the function in Equation (2.2) and its pictorial representation in Figure 4. Where u_N, w_N, θ, and X represents the number of inputs, weights corresponding to each input, bias, and output respectively. Multiplication and addition are the most typical mathematical operations that will be implemented for a normal neuron. The first step is to multiply each of the inputs by the weights assigned to them, as shown below:

$$(u_1 \times w_1), \ldots\ldots, (u_{N-1} \times w_{N-1}), (u_N \times w_N) \qquad (2.1)$$

After getting the products of each input and the accompanying weight, the sum of these products is calculated and added with a bias θ.

$$a = f \ \sum_{j=1}^{N} u_j w_j + \theta \qquad (2.2)$$

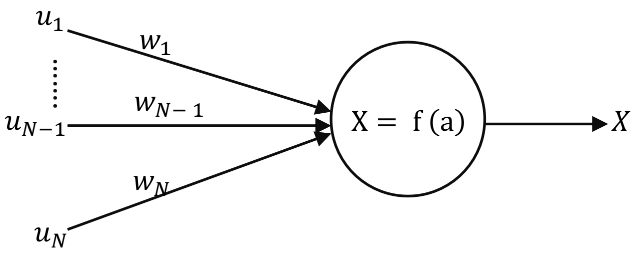


Figure 4 Artificial Neuron (Perception)

The network is referred to as a feedforward network if the connections between neurons are exclusively in one direction. In other words, a neuron cannot be linked to another neuron that is closer to the input. In feedforward neural networks, connections between neurons in the same layer or nonconsecutive layers are not permitted. For a multilayer feedforward network, the input-output relation can be deduced by Equation (2.3), where x_i represents the output of the i^{th} neuron.

$$x_i = f \times \left(\sum_{j=1}^{N} u_j w_{ji} + \theta \right) \tag{2.3}$$

The input-output relationship is written in matrix form to make the computation easier, as illustrated in Equation (2.4), where X, W, U are output vector, weight matrix whose first row is equal to the bias vector θ and input vector respectively.

$$X = f \times \left(W^T U \right) \tag{2.4}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}, \quad U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}, \quad W = \begin{bmatrix} w_{01} & w_{01} & \dots & w_{0M} \\ w_{11} & w_{12} & \dots & w_{1M} \\ \dots & \dots & \dots & \dots \\ w_{N1} & w_{N2} & \dots & w_{NM} \end{bmatrix}, \quad \theta = \begin{bmatrix} w_{01} \\ w_{02} \\ \vdots \\ w_{0M} \end{bmatrix}$$

2.3 Optimization of Neural Network

The principle of optimization of the neural network is to reduce the losses or errors and make the network perform the task more accurately by updating the weights of the neurons. This process is done with the idea of backpropagation. Backpropagation by (Rumelhart, Hinton, & Williams, 1986) revitalized the concept of AI and became more popular. The fundamental goal of backpropagation is to minimize the cost function by modifying the weights of the model. The weights are updated until the difference between the actual and calculated output is minimum or equal.

Assume the network contains u as an input, x as an output, and y as the goal output (or the ground truth). So that $x = y$, the network's weights should be updated. To accomplish this, a cost function like mean square error must first be used. The mean square error may be calculated using Equation (2.5), where N is the number of samples.

$$e = \frac{1}{N}\sum_{j=1}^{N}\left(y_j - x_j\right)^2 \tag{2.5}$$

Equation (2.5) can be minimized using a variety of optimization techniques. F or example, Newton's approach has adapted the second-order derivatives to calculate the roots of a cost function (Baydar, 2018). When compared to approaches such as gradient descent (Kiefer & Wolfowitz, 1952; Robbins & Monro, 1951), which employ first-order derivatives, it greatly increases the computing complexity. Because of this, most neural networks are trained via gradient descent. As a result, only gradient descent will be used in this thesis, which will be thoroughly discussed.

2.3.1 Gradient Descent

Gradient Descent is a method that iteratively finds a (local) minimum by utilizing the function's gradient. The current point's gradient is determined at each iteration. A step is taken in proportion to the gradient's negative. To make it simpler to understand, imagine the function as a valley, and each iteration's points as a ball rolling down the valley. When the ball is on the valley's right wall, it will roll to the left at a rate proportionate to the slope until it attains the local minima. For the left wall, it is just the opposite, as shown in Figure 5.

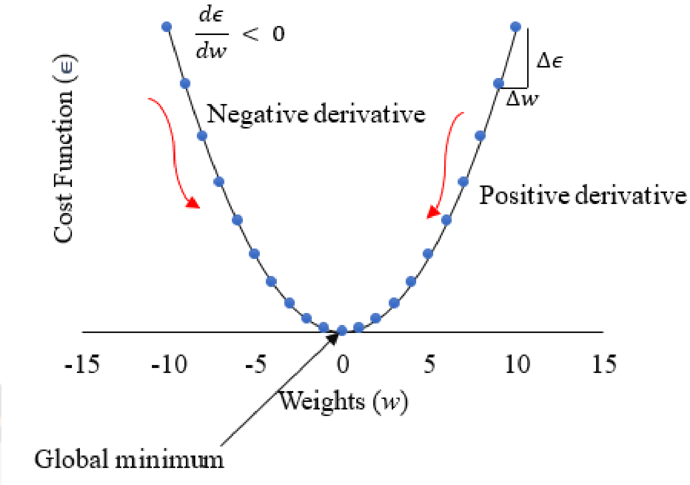


Figure 5 Gradient Descent

When a single neuron is examined, the update of weights at iteration i may be expressed as in Equation (2.6), assuming MSE as a cost function.

$$w(i+1) = w(i) + \Delta w(i) \tag{2.6}$$

$w(i+1)$, $w(i)$ & $\Delta w(i)$ is the new updated weight at i^{th} iteration, initial wight at $(i-1)$ iteration and change in weight at i^{th} iteration respectively. The weight change can be expressed by Equation (2.7).

$$\Delta w(i) = -\mu \nabla e\ (w(i)) \tag{2.7}$$

Where μ is the learning rate. It regulates how much weight is updated and how quickly the weights are updated and e () is the cost function with weight $w(i)$. The $\nabla e(w(i))$ is deduced by Equation (2.8).

$$\nabla e(w(i)) = \frac{de}{d\, w(i)} \tag{2.8}$$

$$x_j = u_j w_j(i) \tag{2.9}$$

Substituting Equation (2.5) in Equation (2.8)

$$\nabla e(w(i)) = \frac{\partial\left(\frac{1}{N} \times \sum_{j=1}^{N} (y_j - x_j)^2\right)}{\partial\, w(i)} \tag{2.10}$$

Substituting Equation (2.9) in Equation (2.10)

$$\nabla e(w(i)) = \frac{\partial\left(\frac{1}{N} \times \sum_{j=1}^{N} (y_j - u_j w_j(i))^2\right)}{\partial\, w(i)} \tag{2.11}$$

$$\nabla e(w(i)) = -\frac{2}{N} \times \sum_{j=1}^{N} (y_j - x_j)\, u_j \tag{2.12}$$

These are steps involved in the weight updating approach using gradient descent. Some of the variants of gradient descent are Stochastic Gradient Descent (SGD) and mini-batch stochastic gradient descent (MBSGD).

Stochastic Gradient Descent (SGD): Bottou (2010) introduces the simple technique to update the parameter. Rather than performing the gradient computation on the entire set it randomly samples the training data in each epoch to training. This approach is both more efficient and memory-efficient and significantly quicker than BGD for large-sized training datasets. However, it is regularly updated, and it makes incredibly noisy steps towards the solution, causing the convergence behavior to become exceedingly unstable. The stochastic gradient has the disadvantage of being readily stuck in a local minimum. Furthermore, because weights are updated by the reaction of a single sample rather than the responses of the entire sample, inconsistent outcomes might be achieved.

Mini-batch Gradient Descent: In this technique, the training samples are divided into multiple mini-batches, each mini-batch may be thought of as a small collection of samples with no overlaps (Hinton et al., 2012). The small batch's sample count is predetermined. This approach aims to combine gradient descent with a single sample and complete gradient descent with a single sample. The training period is greater than single-sample gradient descent, but it can avoid local minimums. As a result, it is the most extensively employed of the three methods. The batch size is determined by the hardware's capability, the size of the dataset, and the outcomes of the experiments Although gradients were extensively adopted as the optimizer for much deep learning, however, they are often considered sluggish and do not ensure a smooth and rapid convergence for the following reasons:

- Choosing a suitable learning rate in SGD is critical and tricky. A low learning rate causes delayed convergence, whereas a high learning rate impedes convergence and causes the loss function to oscillate around the minimum. As a result, the learning rate is now considered a configurable hyper-parameter.
- In the case of high-dimensional parameters, SGD uses the same learning rate for each dimension since each dimension contributes to the total cost distinctly.
- Another SDG difficulty is avoiding becoming stuck in global minimum when the gradient is vanishingly small.

To solve the challenges, numerous optimizers have been developed, such as learning rate annealing approaches or allocating a different learning rate to each dimension of the parameters. Another approach is to use the cost function's second-order derivatives to guide and accelerate the gradient descent. The expense of computation, however, is exorbitant. As a result, the most effective solutions are aiming to approximate the second-order data. SGD with momentum, AdaGrad, Adadelta, RMSProp, and Adam are some of the most often used SGD variations presented.

Momentum: It improves both the accuracy and training speed of Equation (2.7) by adding the calculated gradient from the previous training steps by a factor δ. Where δ is the momentum factor. Equation (2.13) (Qian, 1999; Rumelhart, Hinton, & McClelland, 1986) gives the mathematical representation of the momentum gradient. The value of the momentum factor ranges between 0 and 1, as a result, the step size of the weight update grows in the direction of the bare minimum to reduce inaccuracy.

As the momentum value falls below a certain threshold, the model loses its capacity to escape the local bare minimum. In contrast, when the momentum components value increases, the model acquires the capacity to converge faster. If a large momentum factor is employed in conjunction with LR, the model may miss the global bare minimum by crossing over it.

$$\Delta w(i) = \delta \Delta w(i-1) - \mu \nabla e\left(w(i)\right) \tag{2.13}$$

Adaptive Momentum Estimation (Adam): Adam (adaptive moment estimation) (Kingma & Ba, 2014) is a stochastic optimization technique that only utilizes first-order gradients and takes relatively low resources. The method calculates unique adaptive learning rates for various variables by predicting the first and second moments of the gradients. The methodology blends the features of two optimizers i.e., AdaGrad (Duchi et al., 2011), which performs well with sparse gradients, and RMSProp (Tieleman & Hinton, 2012), which performs well in non-stationary and online settings.

The critical point in machine learning is to optimize the network by updating the weights of neurons by reducing the learning rate with the iteration. This is handled by Adam on his own. It also gives each learned parameter a separate learning rate. For updates, Adam also uses a moving average of the first and second momentums. Another useful aspect of Adam is that it does a bias adjustment before updating the parameters. This eliminates huge step sizes and, as a result, divergence (Baydar, 2018; Duchi et al., 2011). Mathematically Adam can be calculated by Equation (2.14).

$$\Delta w(i+1) = \Delta w(i-1) - \mu \frac{m_1}{\sqrt{v_1} + e} \tag{2.14}$$

Where m_i and v_i are the compute bias-corrected first moment estimates and compute bias-corrected second raw moment respectively, and can be calculated by the following Equation (2.15) and (2.16):

$$m_i = \frac{m_i}{1-\beta_1} \tag{2.15}$$

$$v_i = \frac{v_i}{1-\beta_2} \tag{2.16}$$

Here m_i & v_i represents the update biased first-moment estimate and update biased second-row moment estimate respectively.

2.4 Backpropagation

The backpropagation algorithm is perhaps the most basic component of a neural network. It was initially introduced in the 1960s, and Rumelhart, Hinton, and Williams popularized it nearly 30 years later (Rumelhart, Hinton, & Williams, 1986). Through a mechanism called chain rule, the algorithm is utilized to successfully train a neural network (LeCun et al., 1998; Rumelhart, Hinton, & McClelland, 1986). The backpropagation on a deep neural network is accomplished by using the chain rule to determine the change in output x_k with respect to weight of j^{th} input of i^{th} neuron in L^{th} layer $w_{L,j,i}$.

Using Equation (2.4), we may describe a single output x_k in terms of the weights in the final layer and the outputs of the preceding layer, as illustrated in Equation (2.17).

$$x_k = f\left(W^T_{(L+1),k}\ x_L\right) \tag{2.17}$$

Where $f\left(\ \right)$ represents the activation function. The output of a neuron in the hidden layer may also be represented as shown in Equation (2.18)

$$x_{L,i} = f\left(W^T_{L,i}\ u\right) \tag{2.18}$$

To calculate the gradient decent of $w_{L,j,i}$ $\dfrac{\partial e}{\partial w_{L,j,i}}$ is required. Where ϵ is the cost function say MSE, defined in Equation (2.5). Thus, $\dfrac{\partial e}{\partial w_{L,j,i}}$ can be calculated by performing the chain rule (Equation (2.21)) operation.

$$\frac{\partial e}{\partial w_{L,j,i}} = \frac{\partial e}{\partial x_k}\frac{\partial x_k}{\partial w_{L,j,i}} \tag{2.19}$$

$$\frac{\partial x_k}{\partial w_{L,j,i}} = \frac{\partial x_k}{\partial x_{L,j,i}}\frac{\partial x_{L,j,i}}{\partial w_{L,j,i}} \tag{2.20}$$

$$\frac{\partial e}{\partial w_{L,j,i}} = \frac{\partial e}{\partial x_k}\frac{\partial x_k}{\partial w_{L,j,i}}\frac{\partial x_{L,j,i}}{\partial w_{L,j,i}} \tag{2.21}$$

Which, using Equations (2.17) and (2.18), becomes Equation (2.22)

$$\frac{\partial e}{\partial w_{L,j,i}} = -\frac{2}{M}\times\left(t_k - x_k\right)\, f'\left(W_{(L+1),k}^{T}\, x_L\right)(w_{(L+1),i,k})\, f'(W_{L,u}^{T}\, u)u_j \tag{2.22}$$

When the weight is updated through the process of back propagation, $w_{L,j,i}$ is affected by all the weight's $w_{(L+1),j,k}$ positions. Where k spans from 1 to M. Hence, Equation (2.23) can be derived by adding these partial derivatives.

$$\frac{\partial e}{\partial w_{L,j,i}} = -\frac{2}{M}\times\sum_{k=1}^{M}\left(t_k - x_k\right)\, f'\left(W_{(L+1),k}^{T}\, x_L\right)(w_{(L+1),i,k})\, f'(W_{L,u}^{T}\, u)(u_j) \tag{2.23}$$

If the gradient of the activation function $f()$ is straightforward and simple, the solution to Equation (2.23) becomes simple and less difficult.

2.5 Deep Learning (DL)

Deep learning is a subfield of machine learning that emulates human brain behavior. DL is a data-hungry machine that requires a large amount of data to operate to produce a precise and accurate output. Deep learning algorithms conduct experiments with a predefined logical framework to reach comparable findings to man. It achieves this by adopting a multi-layered neuron called a neural network, with each of the layers functions a unique interpretation of the data that the network is fed (LeCun et al., 2015).

A neural network is made up of many neurons where each neuron can be thought of as a discrete single processor performing a unique function. Hence, the name deep learning was coined after the concatenation of multiple hidden layers to form the network as shown in Figure 6. DL has become an exceedingly popular type of ML approach in recent years as a result of the massive development and progress in the area of big data (Najafabadi et al., 2015).

Deep learning technology has had an impact on nearly every scientific subject. The usage of DL has already disrupted and revolutionized most sectors and businesses. The world's largest technological and economy-focused companies are competing to improve DL and recently it has surpassed human performance on image classification tasks based on ImageNet datasets (Figure 7) (Alzubaidi et al., 2021).

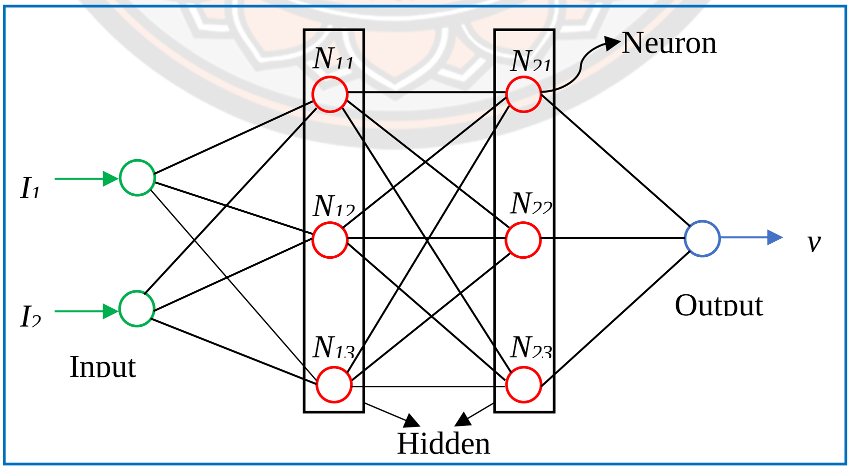


Figure 6 General Structure of the Deep Neural Network

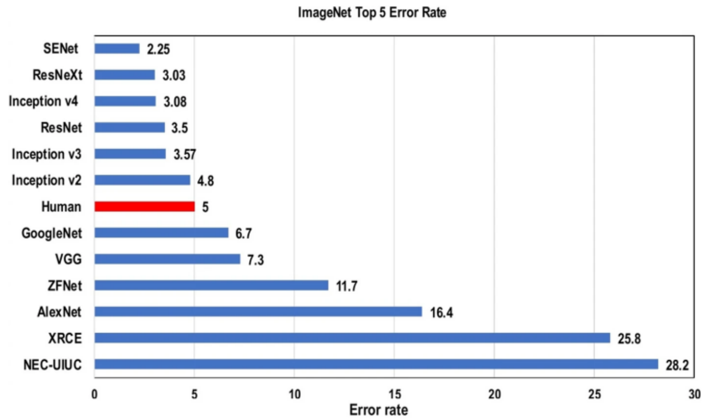


Figure 7 Comparison of Image Classification on ImageNet Datasets Using Different Deep Learning Architecture and Humans

Source: (Alzubaidi et al., 2021)

Some of the reasons why DL has outperformed the many machines learning are listed below:

- Robustness, and Universal
- Scalability
- Generalization & reproducibility

Deep learning is classified into three types namely supervised, unsupervised, and semi-supervised. Furthermore, reinforcement learning (RL), also often referred to as DRL (deep reinforcement learning), is another type of DL that is generally characterized as a partially supervised (and sometimes unsupervised) approach.

Supervised is the most prominent type of learning. In this type of learning each input data to the network is labeled and tagged with the desired output value, allowing the system to determine how the output will be when the input is received. To function well and to get a reliable and accurate output, a supervised learning model sometimes necessitates the size of a large dataset from ground truth observations. These larger datasets which include a more historical example to learn from, allow the algorithms to include multiple cases and generate a model that can manage them. The downside of supervised learning is it suffers from overstraining decision boundary when training sets lacks samples that belong in a class.

In unsupervised learning the network is fed with datasets without any labels, the model discovers the inherent structure and extracts useful hidden features and information from datasets. It allows users to perform more complex tasks with minimal human intervention. However, detailed validation needs to be put by the data analyst to make it more ascertain the accurate information. The unsupervised learning model is generally deployed to perform a task such as clustering, association, and dimensionality reduction. Unsupervised learning's key drawbacks are its inability to give precise data sorting information and it is computationally complex.

In semi-supervised learning, the input data to the network are semi-labeled datasets. One of the benefits of this method is that it reduces the amount of labeled data required. However, the disadvantage of this type of learning is it may produce inaccurate information or decisions due to the irrelevant input features contained in the training data. One of the most well-known applications of this type of learning is the text documents classifier (Alzubaidi et al., 2021).

2.6 Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is the dominant and frequently used algorithm inspired by the combination of DL with image processing. It has resulted in various successes and advancements in the development of computer vision tasks, such as image classification, segmentation, feature extraction, and pattern recognition, to mention a few. The fundamental advantage of CNN over other neural networks is the ability to automatically detect important features without the need for human intervention (Gu et al., 2018).

From its inception to its current state, the convolutional neural network has passed through stages of theoretical, experimental development, large-scale application, and in-depth research (Liu et al., 2021). Like a traditional neural network, the structure of CNN was inspired by neurons in human and animal brains. In particular, the complicated succession of cells forms the visual cortex of a cat's brain (Hubel & Wiesel, 1962). They concluded through biological research that the multilayer receptive field is used to transmit visual information from the retina to the brain. Based on the concept of receptive fields, Fukushima and Miyake (1982) presented a neurocognitive machine and it's regarded as the first implementation of CNN.

However, the development of CNN was not at an exponential rate and was in a dormant state until 2012 when Krizhevsky et al. (2012) introduced AlexNet. This was the first CNN architecture that brought success in image classification from ImageNet datasets. This brought a key steppingstone for the researcher to consider CNN as an un-droppable tool for computer vision applications. The state-of-art review of CNN and its future direction is presented in (Alzubaidi et al., 2021). The basic schematic architecture of CNN is shown in Figure 8 and its building blocks in Figure 9**.**

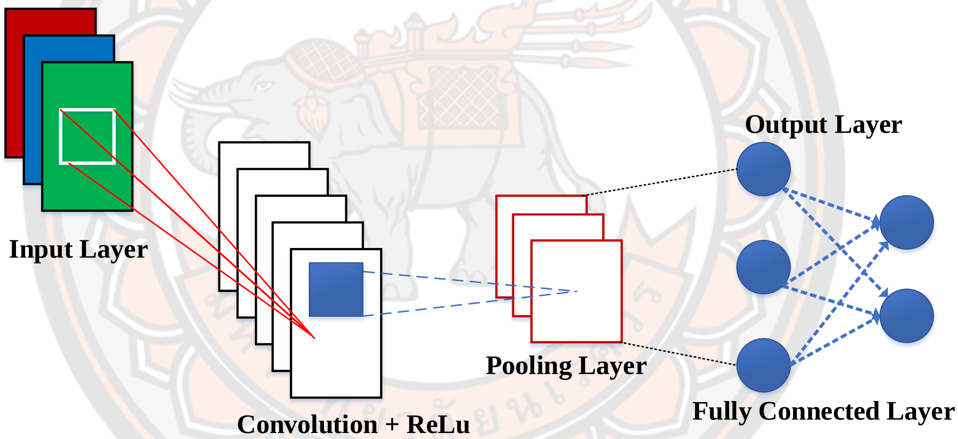


Figure 8 Basic Architecture of CNN

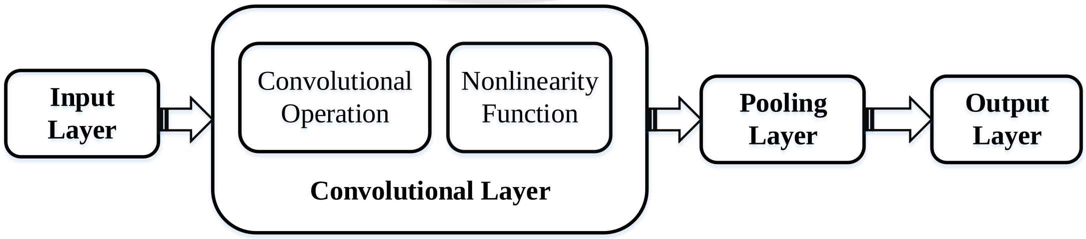


Figure 9 Basic Building Blocks of CNN

Source: Goodfellow et al. (2016)

The basic block of CNN includes the input layer, the convolutional layer, and the output layer. The function of each block will be explained in the following section. The convolutional layer is the fundamental component of a CNN. Convolutional layers generally comprise many consecutive processes, such as convolution, nonlinearity activation, and pooling, as shown in Figure 9. CNNs can view increasingly greater areas of the picture by stacking convolutional layers, and extracting higher-order, more complicated information as the depth increases.

2.6.1 Different Layers of CNN

Input Layer

The input layer contains all the CNN's input data. It usually represents the image's pixel matrix in a neural network for image processing. Depending on the types of images the number of channels will be either 3 (RGB) or 1(gray).

Convolutional Layer

The convolutional layer is the brain of the CNN network, consisting of multiple layers of neurons. These neurons are the filters, also known as convolutional kernels, responsible for learning or extracting the many local information during the training phase. It conducts the dot operation by convolving the kernels with the input picture to produce the output feature map.

Convolution is a mathematical process that determines the degree of overlay between two functions of a real-valued input. Let $g(t)$ and $f(t)$ be the two functions convolve. If $s(t)$ is the output generated by the convolution of two functions, then mathematically it can be defined as the integral of the product of the two functions after one is shifted by factor τ over the other function (Bracewell & Bracewell, 1986). It can be expressed as:

$$s(t) = (f * g)(t) = \int f(\tau) g(t-\tau) d\tau \tag{2.24}$$

However, the commutative property of convolution (Damelin & Miller Jr, 2012), on the other hand, is not retained in CNN, and the convolution process has a new meaning. The first argument $f(t)$ to the convolution is frequently referred to as the input, the second argument $g(t)$ as the kernel/ filter, and the output $s(t)$ as the feature map in CNN nomenclature.

In CNN, convolution is more of a cross-correlation operation (Bracewell & Bracewell, 1986), as illustrated in Equation (2.25), which is a similarity measure of two series as a function of one's displacement relative to the other, also known as a sliding dot product or sliding inner-product. Cross-correlation is implemented in a lot of machine learning libraries, including deep learning libraries, but it is called convolution.

$$s(t)=(f*g)(t)=\sum_{\tau=-\infty}^{\tau=\infty} f(t+\tau)g(\tau) \tag{2.25}$$

Figure 10 shows an illustration of 2D convolution. Let I and K be the two functions namely the input image and the kernel and $I*K$ is the convolution output. The input gray image I having 6×6 dimensions is convolved with a kernel K with dimensions 3×3 resulting in an output feature map of 4×4 dimensions. The green area is called the local receptive field. A dot operation is performed between the I and K by sliding or shifting the kernel by the stride of 1 (the τ value) from left to right and top to bottom and generates an output feature map. Thus, the 2D convolution may be expanded to N-D convolution based on the dimensions of both the input and the kernel. where N is the kernel's spatial dimensions.

As depicted in Figures 11 (a) and (b), CNN employs several kernels in each layer to learn the individual traits and features of that layer, such as aligned edges, corners, or blobs. Furthermore, the most fundamental distinction between CNN convolution filters (kernels) and kernels employed in basic computer vision tasks is how kernel weights are changed during network optimization. In basic computer vision tasks, the kernel weights are hand-crafted, but in CNN, the weights are automatically updated by optimizing the entire network on training data using the objective function as a guide (Chen, 2019).

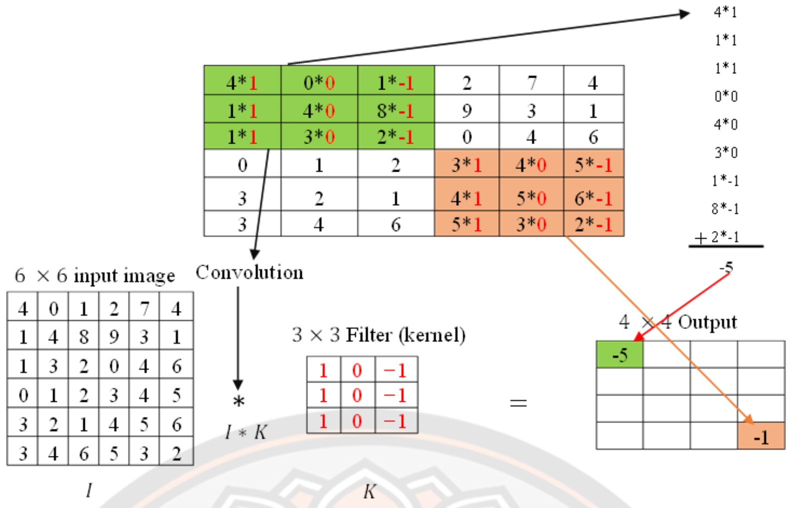
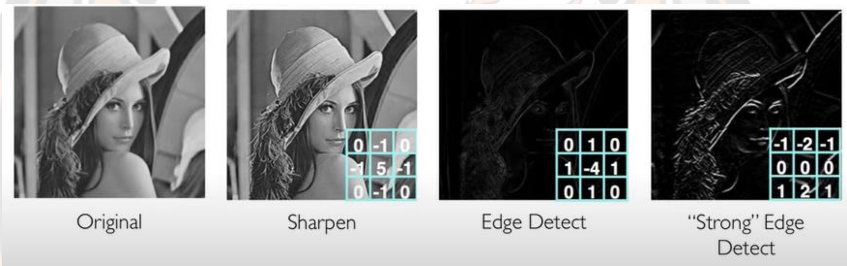
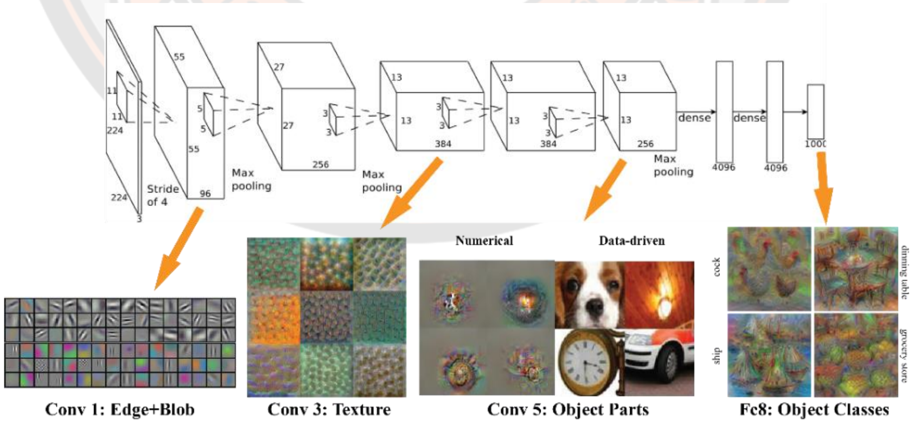


Figure 10 2D Convolution Operation



(a)



(b)

Figure 11 Multiple Kernels used in Each Layer to Extract Different Local Features

Source: Wei et al. (2017)

Some of the challenges with convolution are

- Loss of corner information: One disadvantage is that pixels from the image's corners are only used in a few outputs, but pixels from the central area contribute more, resulting in data loss from the corners of our original image.
- Shrinking output: One of the most difficult aspects of convolving is that if we conduct convolutional operations in numerous layers, the image size will continue to decrease. If the deep neural network has 100 hidden layers and we conduct convolution operations on each layer, the image size will drop after each convolutional layer resulting in the shrinking output.

To alleviate the aforementioned problem of convolution, padding was used. Padding is the process of adding additional pixels to an input image. It makes the output picture the same size as the original image. Padding is classified into two types: valid padding and the same padding. When the same padding is applied, the output image has the same size as the input image, whereas valid padding is the same as no padding.

Suppose if K is the filter size, S is the stride, I is the picture's size, and P is the amount of padding we need, then the output image size can find as in Figure 12 and mathematically can be calculated using the Equation (2.26).

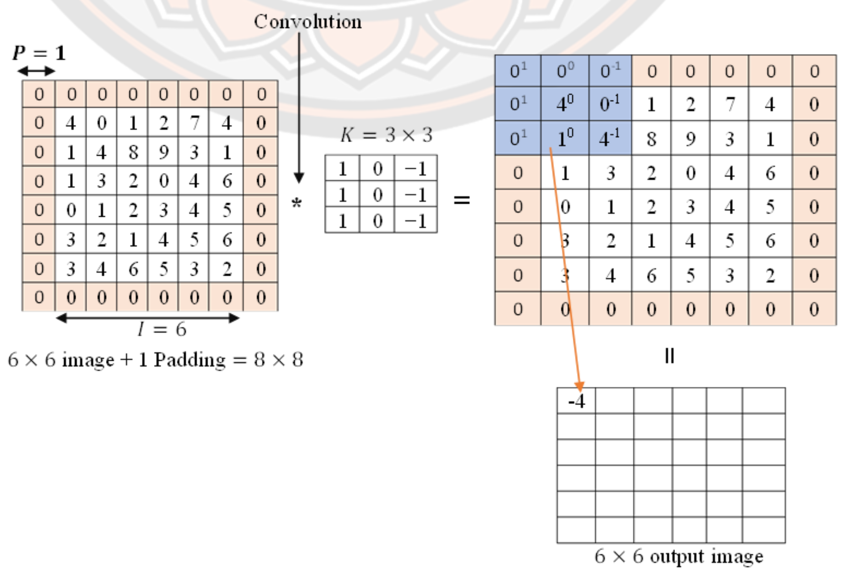


Figure 12 Padding Operation

$$\text{Output size} = \frac{I - K + 2P}{S} + 1 \qquad (2.26)$$

We can see that by setting zero padding to 1(when the condition is set to the same), we were able to keep the original image's size. When $P = 0$ i.e., when padding is valid, signifies no padding at.

Pooling Layer: The primary function of the pooling layer is to provide translational invariance by subsampling the feature maps. This method reduces the size of the feature map by maintaining the important characteristic information at each stage of the pooling layer. Even though the overall number of parameters in a network decrease after every convolution, we still need to minimize the number of parameters and computation time in the network by further condensing the spatial size of the representation. This is done by the pooling layer, which speeds up the calculation while also highlighting some of the key characteristics of the input image.

The filter size and stride are the two main hyperparameters that determine the step of downsizing. The value of stride and kernel size are assigned similarly to the convolution operation. Some of the pooling strategies are max pooling, tree pooling, gated pooling, average pooling, global average pooling (GAP) min pooling, and global max pooling. The max, min, and GAP pooling methods are the most well-known and widely used pooling algorithms. Figure 13 shows the example for max pooling.

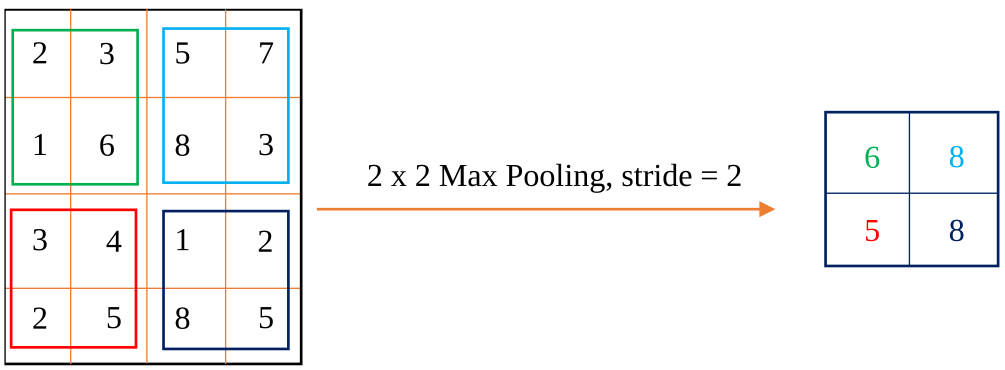


Figure 13 Max Pooling Operation

Activation Function: It decides whether or not a neuron activates depending on the weighted sum of input. It also maps the input and output. In other words, an activation function acts as a gate, ensuring that an incoming value is greater than a threshold value or not. It allows the network to learn powerful operations by providing non-linearities to the network and should be able to distinguish, which is a crucial characteristic because it allows the network to be trained using error back-propagation. On the other hand, if there is no activations function then the network becomes a linear operation and can no longer perform a complex task such as image segmentation. Some of the extensively adopted activation functions are explained in the succeeding paragraphs:

A. **Sigmoid**: Its non-linearity is its major benefit over other steps and linear functions. The function has an *S* (Figure 14) form and ranges from 0 to 1. In certain publications, it is also known as the logistic or squashing function. The sigmoid function is utilized for probability-based output in the DNN's output layers. Sharp damp gradients during backpropagation, gradient saturation, sluggish convergence, and non-zero-centered output are some of its key shortcomings, allowing gradient updates to propagate in multiple directions and its function is given by Equation (2.27).

$$f(x) = \frac{1}{1+e^{-x}} \tag{2.27}$$

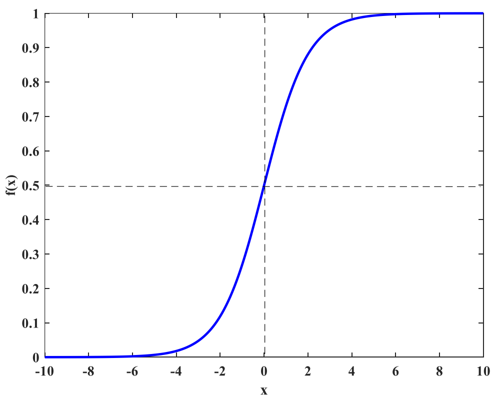


Figure 14 Sigmoid Activation Function

B. **Tanh Function**: Unlike the sigmoid function, it is a zero-centered function, and its value lies between -1 to 1 (Figure 15). Because this function is zero-centered, it is easier to simulate inputs with significantly negative, neutral, or positive values. They are predominantly used in RNN for speech recognition and natural language processing and their function is given by Equation (2.28).

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.28}$$

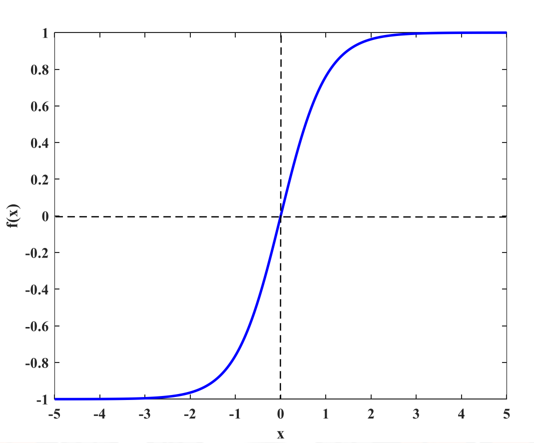


Figure 15 Tanh Function

C. **Rectifier Liner Function (ReLu)**: It is the most extensively used activation function. When compared to the Sigmoid and Tanh, it outperforms them in terms of performance and generalization. ReLU (Figure 16) provides faster calculation because it does not contain exponentials or divisions terms, thus reducing the computation time. However, it also easily gets overfit sometimes referred to as 'drying ReLu' which is one of the key drawbacks. These overfitting issues are reduced by using techniques such as dropouts. Its function is given by Equation (2.29): Some of the variances of ReLu are:

- Leaky ReLU
- Parametric ReLu
- Randomized ReLU

$$f(x) = \max(0, x) \tag{2.29}$$

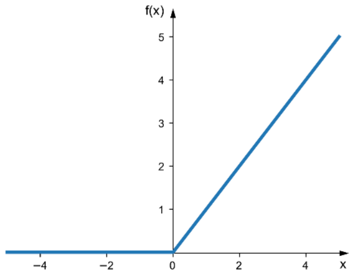


Figure 16 Rectifier Linear Function

- **Leaky ReLu**: to solve the problem of the drying ReLu, leaky ReLU introduces (as shown in Figure 17) the extra term "*m*", is which is called a leaky factor, and its value is very small such as 0.01. This function is mathematically given by Equation (2.30):

$$f(x) = \begin{cases} mx, & if\ x < 0 \\ x, & x \geq 0 \end{cases} \tag{2.30}$$

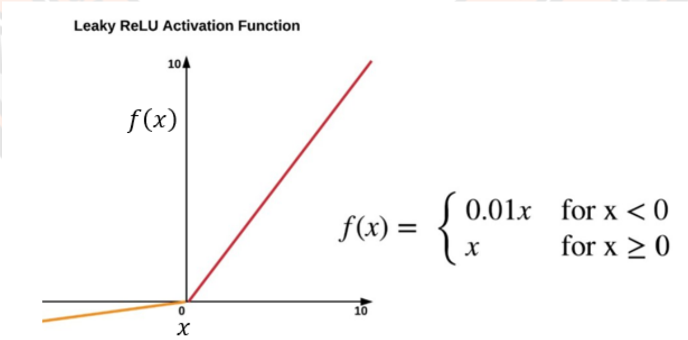


Figure 17 Leaky ReLu Function

Fully Connected Layer: It is the last layer of CNN that connects the preceding layers of neurons with the current layer. The last convolutional layer served as an input to the FC layer and takes a shape form of a vector, which is created by flattening the feature maps.

Loss Function / Cost Function: The lost function calculates and predicts the error between the predicted output and the labeled or ground truth value. The disparity between the actual and expected output is revealed by this inaccuracy. The CNN learning procedure will then be used to optimize it. Some of the commonly used loss functions are discussed as follows:

- **Euclidean Loss Function**: It is sometimes also referred to as mean square error and is widely used in regression problems. The estimated Euclidean loss is expressed mathematically as in Equation (2.31). where $y, \widehat{y}, N$ are ground truth, predicted value, and numbers of sample data respectively.

$$H\left(y, y\right) = \frac{1}{2N}\sum_{i=1}^{N}\left(y - \hat{y}\right)^2 \tag{2.31}$$

- **Hinge Loss Function**: This function is frequently used to solve problems involving binary categorization. This issue is related to maximum-margin classification; it is particularly relevant for SVMs. Hinge Loss function can be calculated by given Equation (2.32) where m is generally set to 1.

$$H\left(y, y\right) = \sum_{i=1}^{N} max\left(0, m - \left(2y - 1\right)\hat{y}\right) \tag{2.32}$$

- **Softmax or Cross-Entropy Loss:** It is also referred to as a log loss function and is used frequently to evaluate the performance of CNN. Furthermore, it is frequently used to replace the square error loss function in multi-class classification tasks. The probability of the output class is given by Equation (2.33): where N denotes the total number of neurons in the output layer and e^{a_i} is the non-normalized output from the preceding layers.

$$P_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e_k^a}, \quad p \in \left[0,1\right] \tag{2.33}$$

$$H\left(y, y\right) = -\sum_{i=1}^{N} y_i \, log\left(\hat{y}\right) \tag{2.34}$$

Equation (2.34), calculates the final cross-entropy loss of the model.

2.6.2 Regularization to CNN

CNN performs well on the training data but when the network is fed in with unseen or test data the model is unable to predict the expected output, hence this phenomenon is referred to as an overfitting problem. Some of the techniques that are used to mitigate such issues are discussed below:

a) **Batch Normalization**: The primary notion behind batch normalizing is motivated by the conventional normalization for input training data. It solves the internal covariate shift problem by normalizing the activations of CNN in each intermediate layer. Particularly, during learning, a Z-score normalization is performed on the output of the preceding layer by subtracting the batch mean and dividing it by the batch standard deviation. As a result of its blocking, etc. on anomalous activations, it permits every layer of a network to train a little more autonomously than other levels and allows for significantly greater learning rates. Batch normalization is also considered a type of regularization since it introduces some noise in each layer. When used in conjunction with dropout, extra care should be taken to prevent losing excessively relevant information during training.

b) **Dropouts**: This strategy was pioneered by (Srivastava et al., 2014). This is a common method of generalization. During each training session, neurons are dropped at random. As a consequence, the feature selection resources are distributed equally across the entire network of neurons, the system is made to study many unique properties throughout backpropagation and forward propagation, and the discarded neuron is left unused. Some of the applications of dropouts in segmentation and classification may be found in (Huang et al., 2017) and (Jégou et al., 2017).

2.7 Related Work on Medical Image Segmentation

For decades, technological innovation has aided mankind, and it continues to improve daily. Machine learning algorithms are capable of learning from data without the need for direct human intervention. The learning increases naturally as a result of the encounters and leads to improved decisions. Emulation of bodily sensory reactions such as hearing, speech, and vision leads to the development of various algorithms (Khan et al., 2019). The computer vision field is one of the most important applications of machine learning. Image segmentation, image categorization, object identification, and object detection are only a few of its applications.

Image segmentation is a well-known subject in computer vision research and has recently become a hot topic in the field of image comprehension. Image segmentation is the process of dividing an image into numerous disconnected regions based on characteristics such as grayscale, color, spatial texture, and geometric forms. As a result, these characteristic exhibits consistency or resemblance in the same region, but a distinct contrast between locations. Medical image segmentation is considered a semantic segmentation. At the moment, there are increasing numbers of image segmentation research disciplines, such as satellite image segmentation, medical image segmentation, and autonomous driving (Ess et al., 2009; Geiger et al., 2012). The image segmentation research is developing at a faster rate resulting in better and more accurate results. However, there is no universal segmentation algorithm that is applicable for all images for diverse segmentation scenarios. Hence, in this section, some of the recent work on medical image segmentation using a convolutional neural network will be reviewed.

2.7.1 CNN Based Segmentation of Median Nerve in CTS

Large image datasets like ImageNet, and high-performance computing platforms, such as GPUs, have made it possible to train incredibly deep CNNs in recent years. CNN-based methods have recently become more popular, and their effectiveness in segmenting the median nerve has been proven. The majority of the techniques developed thus far are based on CNNs, with a few minor deviations. The techniques might differ in terms of dimensions of the input, the kernel size, model depth, and connection between the layers. Although other CNN network designs have been used for image segmentation, such as FCN-8s (Long et al., 2015), SegNet (Badrinarayanan et al., 2017), Mask R-CNN (He et al., 2017), DeepLab (Chen et al., 2017), DeepLb1 (Chen et al., 2014), and GAN (Luc et al., 2016) and DeepMedic (Kamnitsas et al., 2017), most medical image segmentation algorithms, including median nerve segmentation, use a U-Net (Ronneberger et al., 2015) architecture. U-Net transfers contextual information to upsampling layers by concatenating lower layer output to higher layers, resulting in more feature channels.

Kakade and Dumbali (2018) studied the identification of brachial plexus nerve in ultrasound imaging using the U-Net design. They preprocessed all images in their article using linear Gabor binary patterns, which were then fed into the U-Net for segmentation. Thus, they obtained an average Dice measure of 0.669, indicating that using U-Net to directly segment the median nerve is ineffective.

According to Hafiane et al. (2017), Conventional CNNs were not robust enough to recognize nerve regions, therefore the researchers used a CNN with spatial and temporal consistency to enhance localization. The CNN model consists of a pooling layer, convolutional layer, and fully connected layers which identify the area of interest in the nerve and utilize the PGVF technique to designate the median nerve's areas. They obtained a Dice and Hausdorff metrics average of 0.85 and 10.72, respectively. MaskTrack (Chen et al., 2017; Perazzi et al., 2017) is a recursive neural network architecture that can successfully segment items in a sequence image and appropriately find areas of the median nerve. MaskTrack coarsens prior predictions in each step to maintain approximate position and shape information before combining the current picture with former predictions as an input image for segmenting targets.

Horng et al. (2020) proposed a CNN model for the localization and segmentation of median nerve called DeepNerve. Four deep learning models were evaluated based on the original U-Net: lightweight U-Net, U-Net + MaskTrack, ConvLSTM + U-Net + Mask- Track, and DeepNerve.To effectively locate and segment the median nerve, the design is based on the U-Net model and incorporates MaskTrack and convolutional long-term short-term memory (ConvLSTM) and it was discovered that it produced the best median nerve segmentation among the four models. The proposed model performed well in the experiments, with Dice measurement, accuracy, recall, and F-score values of 0.8975, 0.8912, 0.9119, and 0.9015, respectively.

Festen et al. (2021) deployed a U-Net model for fully automated segmentation of median nerve in CT. This method achieved a dice coefficient of 0.88 for finger flexion. Yang et al. (2021) utilized the MaskTrack technique to improve the segmented findings after using a modified DeepLab3+ CNN method to segment the three major components of the carpal tunnel from MRI images and obtained a 0.805 ADSC for the median nerve.

CHAPTER III

METHODOLOGY

3.1 Introduction

In Chapter III, the detailed structure of the study approach adopted in this research study is presented. The chapter is divided into five sections. The second and third sections of the chapter go through the data acquisition and pre-processing strategy in detail. It describes how datasets are gathered for research purposes and how they are processed for analysis. The data pre-processing technique includes image resizing/ROI selection, image denoising, and image enhancement. Sections four and five of the chapter cover the two methods used to segment the median nerve in ultrasound scans. Both methods use the same data acquisition and pre-processing stages. In this work, the signal processing method, and the artificial intelligence (AI) method were used to conduct median nerve segmentation in ultrasonography.

3.2 Data Acquisition (Datasets)

All ultrasound scans were obtained as a secondary image from Fort Somdej Phra Naresuan Maharaj Hospital Phitsanulok, Thailand. The data gathering procedure did not include any patient or individual physical trials. All the ultrasound images of hand wrists collected for this study have been tested in past years and kept in the ultrasound machine's database. The images of the hand wrist were obtained from the ultrasound machine's database between the years 2020 and mid of 2021.

Furthermore, because the study is limited to segmenting the distal median nerve type in the hand and wrist, only ultrasound images of this kind are obtained. The secondary data (stored ultrasound images) does not contain any of the patient's personal or medical information, and those that did were kept protected, confidential, and secret. A total of 70 ultrasound images of the distal type of the hand wrist ultrasound were collected. Each image slice is 864 × 648 pixels in size in JPEG (Joint Photographic Experts Group) format.

The ground truth is obtained by manually annotating the median nerve's boundary on ultrasound images with GIMPS 2.10.28 (GNU Image Manipulation Program), a free and open-source raster graphic program, which is marked or annotated by an expert sonographer.

All the examination was examined using SONIMAGE HS1, Konica Minolta, a portable ultrasound machine. The experiments were carried out by a qualified expert (doctor) with 11 years of experience in medicine and a specialization in rehabilitation medicine who works in the rehabilitation clinic/department at Fort Somdej Phra Naresuan Maharaj Hospital in Phitsanulok, Thailand.

Before collecting data, authorization was obtained from the hospital's higher management authorities for access to data for research purposes and was informed of the study's objectives by completing a consent form by the researcher. The study was reviewed and approved by the institutional review board (IRB) of the Naresuan University, Phitsanulok, Thailand (IRB No. P3-0192/2564 and COA No. 007/2022).

3.3 Data Pre-processing

The acquisition of medical images differs from the acquisition of natural images. The technical characteristics utilized to create a medical image in each modality [ultrasound, CT, MRI] are directly associated with distinct technical parameters in each modality (Masoudi et al., 2021). The information gathered (datasets) is frequently disorganized and comes from a variety of sources. There are generally small differences in image quality, resolution, and field of view when data are collected from the various patients. They must be cleaned before being further processed. Pre-processing is a commonly used step to minimize the complexity and clean the input data. It not only cleans the data but also improves the accuracy of the applied algorithm (Masoudi et al., 2021). Table 1 presents the effect of pre-processing in increasing the accuracy of the deep learning model in image classification and segmentation problems. It is evident from the table, that the accuracy of the model trained with pre-processed data outperformed the model trained without pre-processing for medical image classification and segmentation.

Pre-processing plays a critical role when dealing with noisy, inconsistent, or incomplete data. The purpose of this phase is to prepare input data for further analysis i.e., transform the raw data into machine-understandable formats, making it easier to interpret and process computationally.

Figure 18 shows the overall data pre-processing steps that are adopted in this study. The pre-processing steps include image resizing or region of interest selection (ROI), image contrast enhancement, and image denoising. The details of each of these techniques are discussed in the following sub-section. All of these techniques are implemented using Python 3.8 on OpenCV 4.5.5 (Open-Source Computer Vision Library) as the main library function. The ground truth is achieved by conducting manual area tracing on the median nerve using GIMPS 2.10.28 (GNU Image Manipulation Program) a free and open-source raster graphic tool on the manually marked ultrasound image by the expert sonographer. When the tracing is completed, the selected region is highlighted and changed to a binary image (foreground as white and background as black). The annotation of data (ground truth) is only applicable to the AI method.

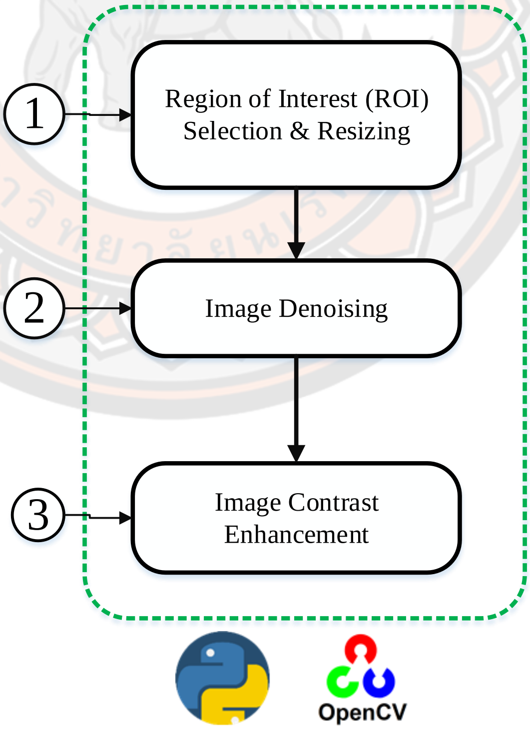


Figure 18 Data Pre-processing Steps

Table 1 Effects of Pre-processing Technique on Medical Image Segmentation and Classification Task in Terms of Accuracy

Task	Image Type	Criteria	With pre-processing (%)	No pre-processing
Classification	MRI	Accuracy	**73.3**	68.74
	CT	Accuracy	**82.28**	77.72
Segmentation	MRI	Mean Abs. Err.	**2.73**	47.64
		Dice	**98.64**	81.74
	CT	Mean Abs. Err.	**3.68**	19.99
		Dice	**98.25**	95.25

Source: Table from (Masoudi et al., 2021)

3.3.1 ROI Selection and Resizing

This process rescales the input data before feeding it into the learning system. It also includes the selection of ROI to reduce the complexity and unrequired area. A Region of Interest (ROI) in medical imaging is a segment of an image that provides critical diagnostic information. Since ROI is utilized as a representative of the image, all subsequent calculations and diagnoses are dependent on it, hence choosing a suitable image region as ROI is critical. Furthermore, using an ROI to extract features from a subset of an image rather than the entire image will minimize the computing time required.

A square ROI size of 432 ×432 pixels from the center of the 864 × 648 pixels original image was chosen as the center of the image is the most credible and provides an optimal spot and also avoids any distorting effects in ultrasound wave patterns as shown in Figure 19. The information on the sides of ultrasound images is not very dependable due to the scattering of ultrasound echoes and associated constructive/destructive interference. From Figure 19, it is seen that from the original input image with a size of 864 × 648 pixels, there is an area where most of the pixels are in black (lower portion of the image) while in some area the values of the pixels seems to be randomly combined pixels ranging between black and white (upper portion of the image). From Information Theory point of view, the upper portion of the image contains more information than the lower portion. Thus, in the ROI selection process, the upper part of the image should be selected, and this is shown in Figure 19 (b). Thus, using an ROI to extract features from a subset of an image rather than the entire image will minimize the computing time and power required.

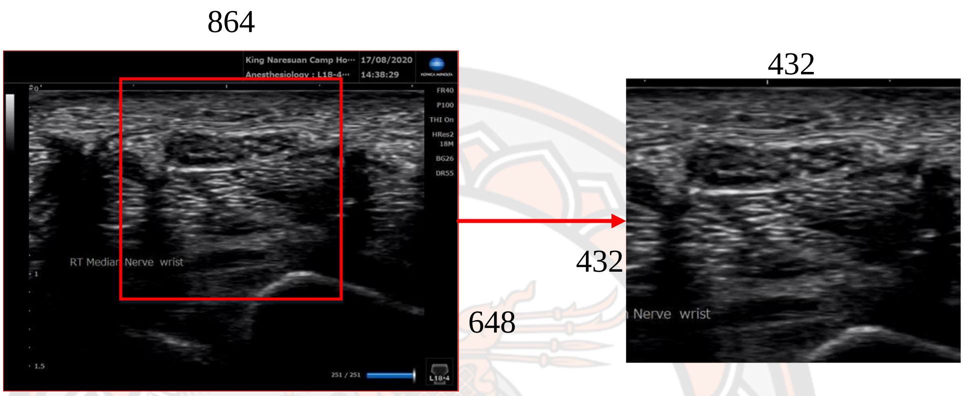


Figure 19 Image Resizing and Selection of ROI

3.3.2 Denoising of Ultrasound Image

Noise is an undesirable signal that occurs during the transmission process or the data acquisition process. During the data acquisitions from the patient due to the patient movements, scanner failure, technical expert of the sonographer, inadequate resolution, and no proper contact or air gap between the scanner prob and patient leads to disruption in the quality of collected data and potentially the data are associated with noise which makes the data interpretation difficult. Some of the common types of noise are Gaussian noise, salt and paper noise, Poisson noise, and speckle noise (Bharati et al., 2021). In the medical image in particular ultrasound images, speckle noise is the predominant noise that deteriorates the quality of the image (Michailovich & Tannenbaum, 2006). Speckle noise is a granular noise that affects and degrades the image quality resulting in the degradation of fine details of edges and other important textures hence adding complication and harder for the physician to distinguish between the fault and normal conditions.

Some of the traditional filtering methods that are widely used for the removal of speckle-noise are discussed in (Gu et al., 2014; Gupta et al., 2018; Michailovich & Tannenbaum, 2006; Nadeem et al., 2019). Lately, some deep learning techniques are also studied for denoising the speckle noise (Ilesanmi & Ilesanmi, 2021; Karaoğlu et al., 2021; Kaur et al., 2018; Shen et al., 2017; Tian et al., 2018). In this work, a bilateral filter is adopted for filtering and smoothing the ultrasound image.

A bilateral filter is a non-linear, edge-preserving filter that smooths the noise. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on the Euclidean distance of pixels but also on the radiometric differences (e.g., range differences, such as color intensity, depth distance, etc.). A bilateral filter not only removes the noise but also preserves sharp edges. For medical images, edges play an important role in the identification of diseases from other associate structures.

The bilateral filter is firstly presented by Tomasi and Manduchi (1998). The concept of the bilateral filter was also presented in Smith and Brady (1997) as the SUSAN filter and (Yaroslavsky, 1985) as the neighborhood filter. Mathematically the bilateral filter is defined by

$$BF\left[I\right]_P = \frac{1}{W_P}\sum_{x_i=S} I\left(x_i\right)G_r\left(\| I(x_i)-I(x)\|\right)G_s\left(x_i-x\right) \tag{3.1}$$

Where Wp is a normalization factor

$$W_p = \sum_{x_i=S} G_r\left(\| I(x_i)-I(x)\|\right)G_s\left(x_i-x\right) \tag{3.2}$$

Where $BF\left[I\right]_P$ is the filtered image, $I(x_i)$ is the original ultrasound input image, x is the current pixel coordinate of the image to be filtered, s defines the window cantered for x, G_s and G_r are the kernel range for smoothing images in intensities and spatial kernel for smoothing different coordinates.

The input median nerve ultrasound image was filtered using kernels value $\left(S, G_s, G_r\right) \rightarrow \left(9,18,85\right)$ as the optimal values. The value of G_s and G_r lower than 18 have a fewer effect and greater than 85 make the image smoother and more blurred in some edges of the median nerve structure. Furthermore, the kernel value or window S less than 9 makes less impact on noise removal.

Figure 20 shows the comparison of different filters in removing the spackle noise from the ultrasound image. Figures 20 (a) and (b) represent the original image and the corresponding threshold image. The threshold image (Figure 20 (b)) shows that along with the major median nerve structure, there are tiny segments of noise speckle that surround them, making the analysis difficult. To minimize noise, Gaussian (Figure 20 (c)) and Median (Figure 20 (d)) filters are used. Although this filter reduces noise by a certain proportion, it also fades the median nerve's edges, which is crucial for analysis. In comparison to other filters such as the Gaussian filter (Figure 20 (c)) and the Median filter (Figure 20 (d)), the bilateral filter not only reduces noise but also keeps and preserves the fine borders of the median nerve structure (Figure 20 (e)). This demonstrates the advantage of bilateral in reducing speckle-noise while preserving fine edges as compared to other filters (Gaussian & median).

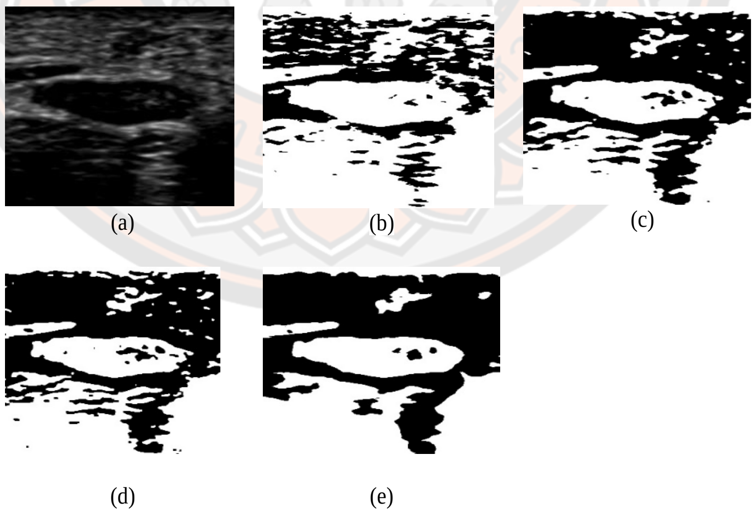


Figure 20 Image Filtering (a) Original image (b) Threshold Input Image (c) Gaussian Filter (d) Median Filter (e) Bilateral Filter

3.3.3 Image Enhancement

Contrast enhancement is a crucial part of image processing for both human and machine vision. It is commonly used in medical image processing, as well as speech recognition, texture synthesis, and a variety of video processing applications as a pre-processing step. Due to the intrinsic properties of medical images, such as poor contrast, speckle noise, signal dropouts, and complicated anatomical structures, medical image analysis is a particularly tough challenge to solve. If the input image to the learning system is of low quality and contrast, further processing operations such as image segmentation, feature extraction, and image classification will fail. As a result, before further processing and analysis, it is critical to improve the contrast of such images.

The performance of pre-processing procedures such as image enhancement will have a direct influence on any subsequent processing. As a result, high-performance image-enhancing algorithms can improve total system performance dramatically. Histogram equalization (HE) is a frequently used technique for increasing visual contrast. Its core concept is to map gray levels using the probability distribution of the input gray levels. Suppose an input image $g(x, y)$ composed of discrete gray levels in the dynamic range of $[0, L-1]$, the transformation mapping $C(r_k)$ of HE is given by Equation (3.3):

$$s_k = C(r_k) = \sum_{i=0}^{k} P(r_i) = \sum_{i=0}^{k} \frac{n_i}{n} \tag{3.3}$$

where $0 \le s_k \le 1$ and $k = 0,1,2,...,L-1$. n_i represents the number of pixels having a gray level r_i, n is the total number of pixels in the input image, and $P(r_i)$ represents the Probability Density Function (PDF) of the input gray level r_i. Based on the PDF, the Cumulative Density Function (CDF) is defined as $C(r_k)$.

The mapping in (Equation (3.3)) is called Global Histogram Equalization (GHE) or Histogram Linearization. This approach uses the image histogram to spread out the gray levels in an image and reassigns the brightness value of pixels. The GHE approach works best when the original image has little contrast. GHE remaps the gray levels in this situation so that contrast stretching is limited in some dominating grey levels with larger image histogram components, while significant contrast loss occurs in others (i.e., it gives to either too dark or too bright regions in the image). If the image has certain gray levels with high frequencies, it will dominate the low-frequency gray levels.

In this work, we have adopted the contrast limited adaptive histogram (CLAHE) to perform the contrast enhancement of ultrasound images. The contrast limited adaptive histogram (CLAHE) method is another technique to enhance the contrast of the image which overcomes the disadvantage of GHE. It is a hybrid of HE and adaptive histogram equalization, in which the histogram is adaptively equalized in blocks with a predetermined clip limit. CLAHE does not function on the entire image; instead, it operates on tiny areas of the image known as tiles. It works the same way on each tile as regular Histogram Equalization (HE) (Ahmed & Nordin, 2011; Faten A. Dawood, December-2018). The ultrasound input image is clipped with a clip limit of 8 and a tile gride size of (9×9). The tile grides size greater than (9×9) produces more imbalance contrast in an image i.e., it makes the image either too white or dark.

Figure 21 shows the effects of GHE and CLAHE on image enhancement. From Figure 21 (b) it is observed that the GHE equalization stretches the contrast of an image to either too dark or too bright region. Figure 21 (c) depicts the effects of CLACHE equalization in which the contrast of the image is stretched adaptively. Figure 22 shows the histogram of the original and the CLAHE equalized images. As shown in the diagram the histogram of the original image (Figure 22(a)) is more skewed towards 0 to 155 compared to the CLAHE equalized image (Figure 22(b)) whose histogram intensity is almost uniformly distributed across 0 to 255.

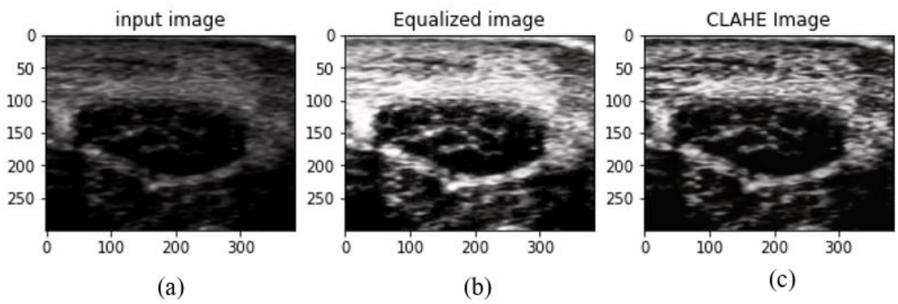


Figure 21 (a) Original Image (b) Global Histogram Equalization (c) CLAHE Equalization

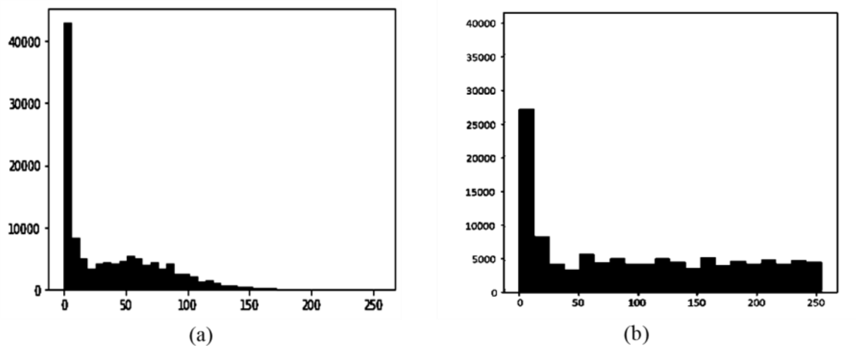


Figure 22 (a) Histogram of Original Image (b) Histogram of CLAHE Equalized Image

3.4 Method 1: Signal Processing Method

This section discusses the segmentation framework to detect the median nerve from the ultrasound images using the signal processing techniques framework.

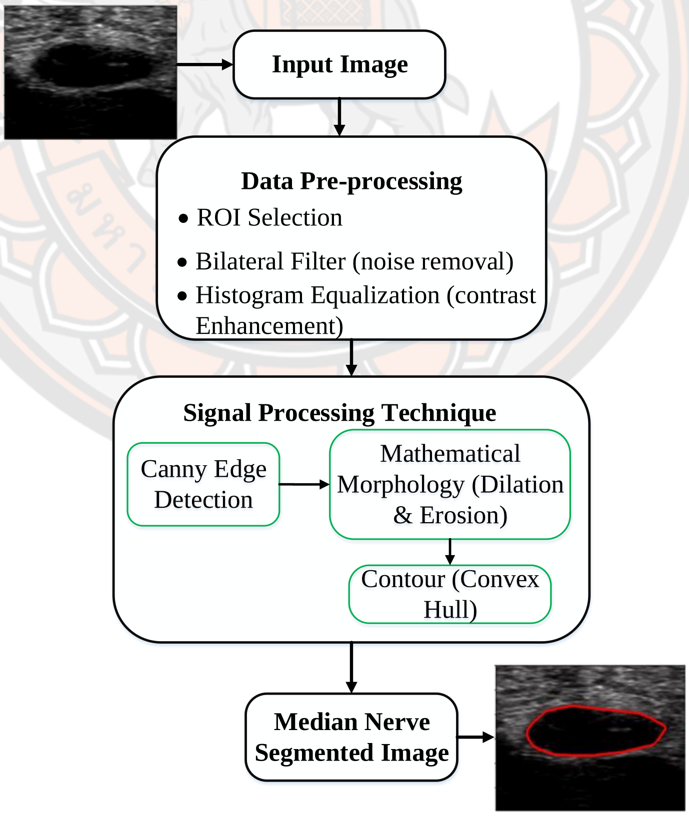


Figure 23 System Block Diagram of Signal Processing Method

Figure 23 shows the overall methodology adopted for the signal processing method. Overall, there are two stages involved in the segmentation of median nerve on ultrasound images namely the pre-processing and the feature extraction part. The pre-processing stage includes the ROI selection, image denoising, image contrast enhancement, and thresholding. The details of these techniques are discussed in section 3.3. The feature extraction stage includes edge detection, morphology operation, and contouring.

After the pre-processing stage, the canny operator is utilized as the edge detection method to determine the structure of the median nerve in an ultrasound image to divide the region. This edge identification approach helps to simplify image analysis by dramatically lowering the quantity of data that must be processed while keeping important structural information about median nerve borders.

Once the edge has been detected, a mathematical morphological method called dilation is used to extend the edge such that it encompasses the anatomical region of interest. To ensure that the growth process does not extend beyond the interest region, an appropriate dilation value is computed by repetitive iteration. Any undesired growth regions are removed as a precautionary approach. As a result, dilation will reunite the disjointed dotted edge lines, particularly those that make up the median area shape. The erosion will then eliminate the unnecessary edge line that connects the specified region.

After the median nerve region's edge has been produced as a closed rounded line, the close operation is performed to close the rounded boundary. The region area is filled to create a region of interest (ROI). The ROI is significant since it will be used as the image mask for the other detection task shortly after. The removal operation is performed to separate each ROI from its adjacent region, which may contain other undesirable areas, particularly those tiny portions next to the median nerve area, resulting in a single object (shown in Figure 24). The convex hull is used to define the contour to join some small disjoints in the segmented median nerve boundaries.

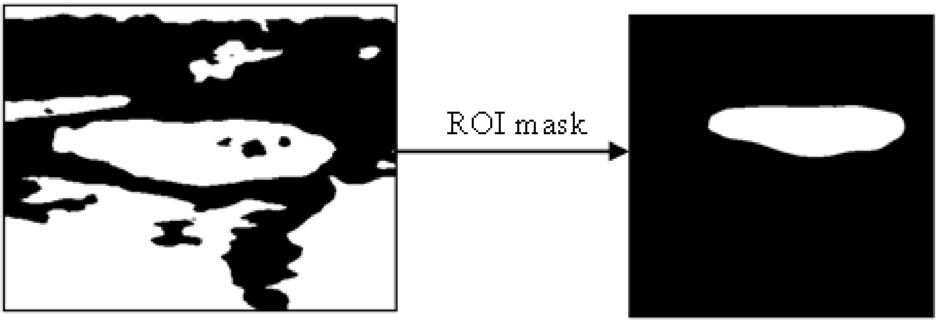


Figure 24 ROI Mask after Morphology Operation

Finally, a Jaccard similarity measurement test is performed to see the similarity between the segmented median nerve and the ground truth. Furthermore, the cross-sectional area (CSA) of the segmented median nerve is calculated to check the presence of CTS or not and compared with the value calculated by the expert sonographer.

3.5 Method 2: Artificial Intelligence (AI) Method

Figure 25 portrays the median nerve segmentation framework adopted in this study. The framework is divided into two phases: the training phase and the median nerve segmentation phase. Three distinct U-Net implementations were trained during the training phase. All three implementations are based on the original four-layer U-Net. The original input image from the median nerve datasets is fed to the base U-Net, which is responsible for the extraction of median nerve features and performing segmentation.

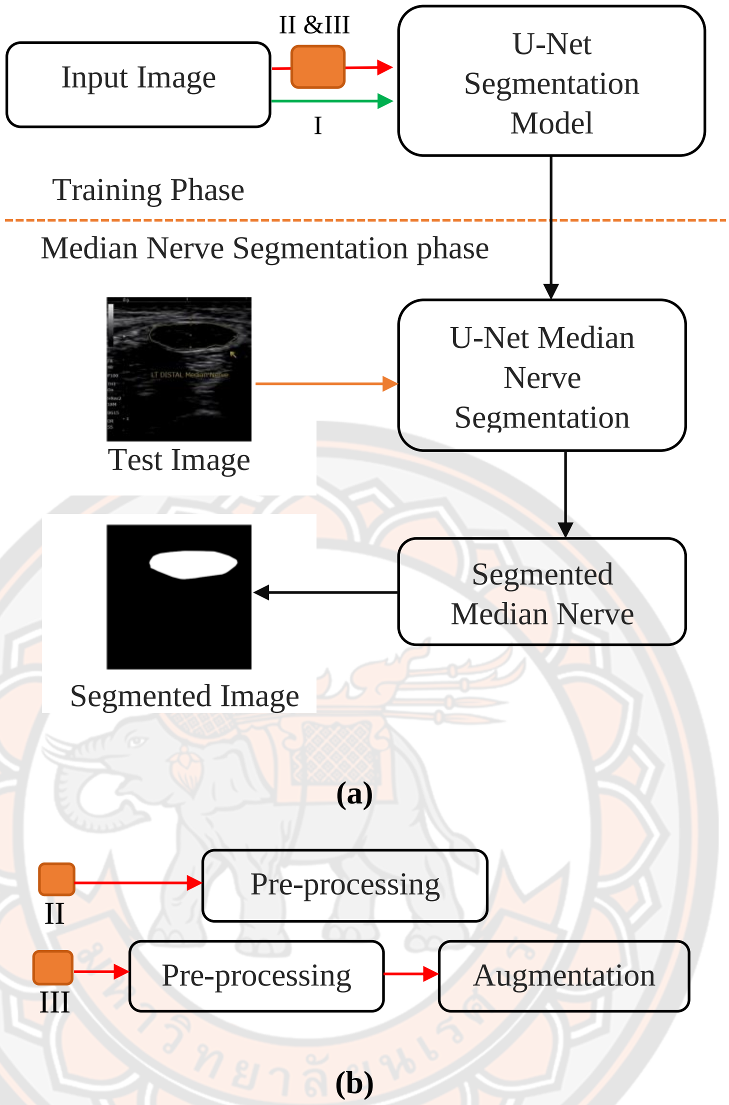


Figure 25 Block Diagram for AI Method

In approach I, the input image is fed directly to the segmentation model (represented by the green arrow in Figure 25 (a)). No data modification was made to the input image. The image is resized into 432×432 pixels. However, in approaches II and III the input image is not fed directly into the segmentation model. To make the segmentation work easier, it is transmitted via the brown box (Figure 25 (a)).

In approach II the input image is pre-processed before feeding into the segmentation model (Figure 25 (b)). Pre-processing is applied to remove the noise associated with the input image and enhance the contrast of the image thus minimizing the complexity of the segmentation process. The detail of data pre-processing is explained in section 3.3.

In approach III, in addition to pre-processing the input test data is augmented (the detail of the data augmentation will be discussed in the sub-section 3.5.1) before being fed into the segmentation model to generate enough training data as shown in Figure 25 (b). Furthermore, as a slight change to the base U-Net, a batch normalization layer (BN) was introduced at the end of each convolutional layer for quicker convergence and to increase the training accuracy.

In the median nerve segmentation phase, the test data are experimented on in the trained model and the performance of each model is evaluated. It outputs the binary image (0&1) as the segmented image. The white (1) pixel represents the median nerve structure and the black (0) pixel denotes the surrounding structure.

3.5.1 Data Augmentation

Data augmentation is the process of adding diversification and variance to the existing data by scaling or rotation or applying other transformations. This allows the CNN to expose to a wide variation of data so that network will be able to recognize data of any shape and form during the testing phase. Hence, letting the network learn invariance to such deformations without having to look at the annotated image corpus for these modifications.

This technique also increases the size of datasets, particularly for medical images. It is exceedingly difficult to develop large medical image databases because of the rarity of diseases, patient privacy, the requirement for medical personnel to label images, and the expense and human labor necessary to undertake medical imaging procedures. Hence, augmentation assists in artificially incorporating the realistic deformations caused by the variation in tissue, thus enhancing medical data. In this study, we adopted geometric transformations such as rotation, shearing, flipping, zooming, and translation to perform the data augmentation.

Figure 26 depicts one sample of the augmented image after different geometrical transformations have been applied. While executing the augmentation, careful observation was made to ensure that the augmented image resembled the true realistic deformations. The first row represents the input image, and the second row represents the corresponding mask in the input image.

Figure 26 (a) denotes one of the inputs of original sample images. Figure 26 (b) is the horizontally filliped image. We applied horizontal flipping since vertical flipping does not give a practical representation.

Figure 26 (c) represents the zoomed augmented image. The input image is zoomed in with the zooming factor of 0.04. When the zooming factor is larger than 0.04, in some input datasets, a piece of the median nerve structure is destroyed or truncated. Hence, an optimum value of 0.04 is chosen as the zooming factor.

Figure 26 (d) is the rotated clockwise and anticlockwise image. The rotation augmentation is performed by rotating the ultrasound image clockwise or anticlockwise by restricting the angle θ between $-20 \le \theta \le 20$ degrees. The input image was rotated every one-degree angle clockwise and anticlockwise. The boundary of space generated on the image's boundaries because of the rotation is filled with mode "constant," i.e., with binary pixel value 0. When the image is rotated beyond $-20 \le \theta \le 20$ degrees, it no longer resembles the particle nature of the image.

Figure 26 (e) represents the sheared augmented image. The image is sheared left and right by shearing factors $0.2, 0.09, -0.25,$ and -0.20. The empty area created on the image's edges because of sharing is filled with mode "constant," i.e., the binary pixel value 0.

Figure 26 (f) represents the horizontal flipped-translated image. The image is first horizontal flipped and then the combination of left and up translation is applied.

Lastly, Figure 26 (h) portrays the translated image. The translation augmentation shifts the image up, down, left, or right. The image is translated left, right up, and down by applying translation matrix $\left[[1,0,-25],[0,1,-50]\right]$, $\left[[1,0,5],[0,1,150]\right]$, $\left[[1,0,10],[0,1,80]\right]$, and $\left[[1,0,-50],[0,1,-20]\right]$ respectively. The translation matrix value is carefully chosen so that it does not over-translate the image. Similarly, the empty area created on the image's edges because of translation is filled with mode "constant," i.e., binary pixel value 0. Similarly, the mask of each corresponding image is also augmented with the same augmentation technique and augmentation factor.

Thus, by employing these transformations, the datasets are multiplied by 56, resulting in 4200 number images. It not only enhances the data but also artificially added a wide range of different variations of images in datasets.

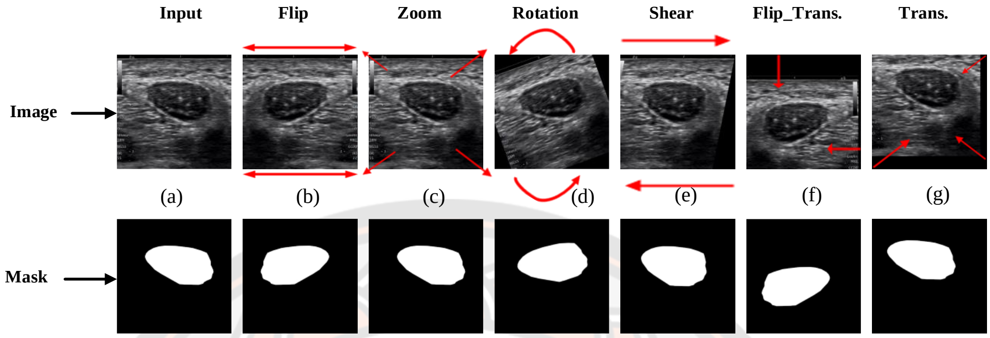


Figure 26 Augmented Ultrasound Images

3.5.2 Model Selection for Segmentation

In this study, we adopted the U-Net (Ronneberger et al., 2015) model, a convolutional neural network (CNN) architecture to perform the median nerve segmentation. The U-Net network is made up of the encoding path, decoding path, and bottleneck connection which connect the encoding and decoding path. Each path offers four levels of resolution.

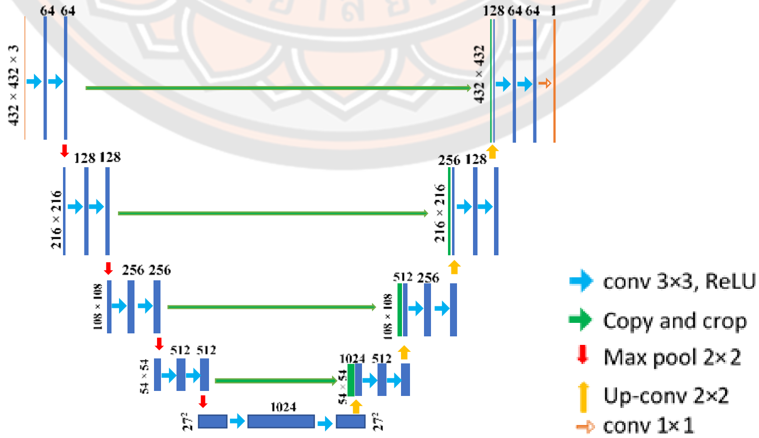
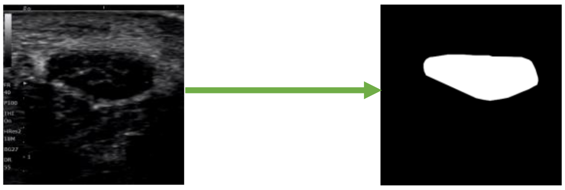


Figure 27 Network Work Architecture of U-Net

Source: Image adapted from Ronneberger et al. (2015).



Input image (3x432x432)

Output image (1x432x432)

Figure 28 Input Image to Model and Expected Output Image

Architecture

Figure 27 shows the U-Net architecture implemented to perform the median nerve segmentation. The model is fed with an RGB image of size 3×432×432 pixels and outputs a black and white image of size 1×432×432 pixels (Figure 28). The white pixel represents the segmented median nerve, and the black pixel denotes the non-median nerve structure.

The 11 layers at the left part of the U-Net represent the encoding path. The path is divided into four blocks. Each block consists of two repeated convolution layers followed by a max-pooling layer. Each repeated convolution layer uses a 3×3 kernel filter followed by a ReLu activation layer. The ReLu activation was used as it can train the model significantly quicker while also optimizing the weights compared to other activation functions (details of different activation functions in section 2.6.1 of Chapter II). To reduce dimensionally and the risk of information loss during the downsampling process, we employed a maximum pooling layer with a 2×2 kernel and stride of 2. In the encoding path, the spatial information is reduced from 432×432 to 27×27, while the feature information increased from 64 to 1024.

The bottom 3 layer is called a skips connection and connects the encoding and decoding path. The 13 layers at the right part of the U-Net depict the decoding path. Similar to the encoding path, there are two successive convolution layers, copy and cropping (concatenating) from each block of encoding and the up-convolution layer. Each layer in the decoding path starts with a 2×2 up-convolution layer with a stride of 2, then two 3×3 convolution layers. After each convolution, a ReLu layer is added. Due to the loss of boundary pixels in every convolution, cropping is applied. The crop and copying restore each spatial information lost during the down convolution by concatenating the output of the up-convolution layer with the high-resolution feature information from the encoding blocks at the same level. The output layer of the U-Net is modeled with 1×1 FC convolution to connect the preceding layers and convert the 64-feature vector into 1 class. The network comprises a total of 27 convolutional layers.

3.5.3 Training and Evaluation of Network

All the experiments were performed on a Google Colab Pro cloud service using TensorFlow 2.7.0. and Keras 2.3.1 framework in python3. Google Colab Pro provides a 24-hour cloud service to perform machine learning training. The model is trained for 200 epochs with batch size 16. Dice loss is used as a cost function, Adam as an optimizer, and the learning rate of 0.000001(10^{-6}). The details of each term i.e., learning rate, cost function, and Adam optimizer are discussed in Chapter II. The model has a total parameter of 31,055,279 of which 31,043,521 are the trainable parameters.

We trained three different U-Net models namely, the base U-Net, U-Net with pre-processed data, and U-Net with pre-processed & augmented data, and batch norm layer, and their accuracy and other evaluation indices are compared. After performing the training for all sets the best model with high evaluation indices results is saved for the testing purpose for unseen data to perform the final median nerve segmentation.

All experiments have been performed with 10-fold cross-validation to validate the proposed method for automatic segmentation of the median nerve in ultrasound images. It is allocated that in each experiment, 90% of the data is utilized for training and the remaining 10% is used for testing. The model performance result is the average result after a 10-fold cross-validation.

3.5.3.1 Evaluation Metrics

To evaluate the model, we used the confusion matrix (shown in Figure 29) to describe the performance of the model on the test data in which the true values are known. The confusion matrix includes accuracy, precision, and F1-score which will be discussed in detail in the following sub-section. Some of the terminologies used in defining the confusion matrix are TN (True Negative): Correctly rejected prediction for a certain class, FP (False Positive): Incorrectly identified prediction for a certain class, and FN (False Negative): Incorrectly rejected prediction for a certain class.

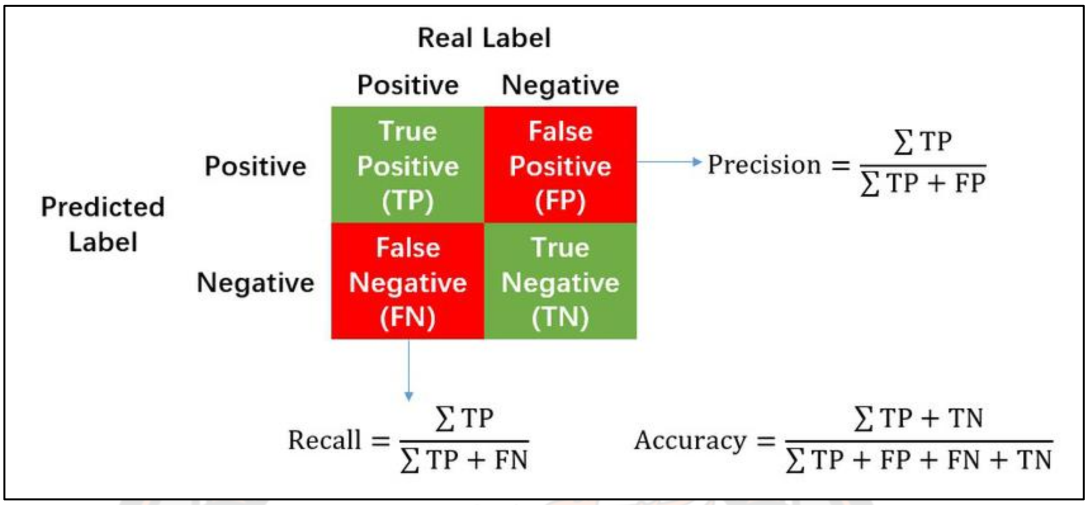


Figure 29 Confusion Matrix

1. Accuracy

Accuracy is a matrix that indicates how well the network executes across all the classes in general. It is effective when all classes are of similar significance. It is calculated as the ratio between the number of correct predictions ($\sum(TP+TN)$) to the total number of predictions ($\sum(TP+FP+TN+FN)$) and is mathematically represented by Equation (3.4).

$$\text{Accuracy } (A) = \frac{\sum(TP+TN)}{\sum(TP+FP+TN+FN)} \tag{3.4}$$

2. Jaccard similarity coefficient

The Jaccard similarity index (also known as the Jaccard similarity coefficient) compares members of two sets to determine which are identical and which are distinctive. It is a metric that compares the similarity of two sets of data on a scale of 0 to 1. The closer the number is near one, the more closely connected the two objects are, and vice versa. Despite its ease of use, it is prone to small sample sizes and may produce inaccurate results, particularly with extremely small samples or data sets with missing observations.

Mathematically the Jaccard similarity is deduced using Equation (3.5), where $J(A,B)$, are the Jaccard similarity between A and B, the predicted (segmented) and ground truth image respectively. Figure 30 is the pectoral representation of $J(A,B)$.

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|} \tag{3.5}$$

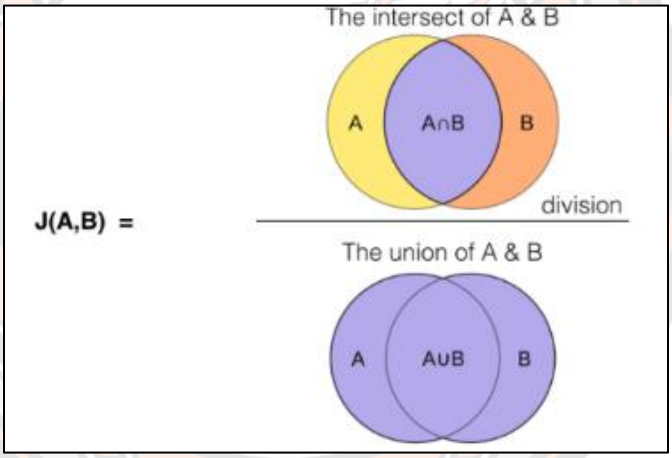


Figure 30 Jaccard Coefficient

3. Recall

The recall is calculated as the ratio between the number of positive samples correctly classified as positive $(\sum TP)$ to the total number of positive samples $(\sum(TP+FN))$, and its mathematical representation is given by Equation (3.6). The recall score evaluates the model's potential to recognize positive samples. The greater the recall, the increasing the percentage of positive samples identified. It only concerns how positive samples are labeled. It is unaffected by how the negative samples are labeled.

$$\text{Recall}\left(R\right)=\frac{\sum TP}{\sum\left(\ TP+FN\right)} \tag{3.6}$$

4. Precision

Precision is the ratio of true positives $(\sum TP)$ to the total of the true positives and false positives $(\sum\left(\ TP+FP\right))$ (Equation (3.7)). Precision examines how many false positives were introduced into the mix. When there are no false positives (FPs), the model has an accuracy of 100 percent. The more FPs that are added to the mix, the worse the precision will appear. The purpose of precision is to correctly categorize all positive samples as positive while avoiding incorrectly classifying a negative sample as positive.

$$\text{Precision}\left(P\right)=\frac{\sum TP}{\sum\left(\ TP+FP\right)} \tag{3.7}$$

5. F1 score

The F1 score is calculated as the harmonic mean of precision and recall (Equation (3.8)). This is used to rate performance statistically. The F1 score ranges from 0 to 9, with 0 being the lowest and 9 being the highest. Since we will be using an unbalanced number of data, finding simple accuracy may not completely ascertain the model performance. Hence, the F1-Score will be used to find the model accuracy which takes care of unbalanced data.

$$F1=2\times\frac{\text{Recall }(R)\times\text{Precision }(P)}{\text{Precision }(P)+\text{Recall }(R)} \tag{3.8}$$

In this chapter, the detailed methodology for the median nerve segmentation in ultrasound images is outlined. It includes the signal processing method and AI method. In the AI method, three approaches are presented based on U-Net. The results of each method in segmentation of median nerve in ultrasound images are presented in Chapter IV.

6. Dice Similarity Coefficient (DSC)

The DSC is evaluated by comparing the ground truths to the automated segmentation outputs using the spatial overlap percentage between two binary images as a metric. The score might range from 0 to 1, with 0 indicating no overlap and 1 indicating a perfect match.

$$DSC = \frac{2 \times |X_a \cap X_b|}{|X_a| + |X_b|} \tag{3.9}$$

X_a represents the ground truth manually annotated by the expert and X_b denote the predicted or automatically segmented region by the segmentation model.

CHAPTER IV

RESULT AND DISCUSSION

4.1 Introduction

The overall results of the adopted strategy for segmenting the median nerve in ultrasound images are discussed in this chapter. The outcomes include training and validation findings based on the measurement matrices discussed in Chapter II. Examples of the segmentation outputs image are also presented for viewing along with the numerical findings.

The results of signal processing techniques are discussed in the second section of the chapter. The outcomes of deep learning with various approaches are shown in section three of the chapter. In the sub-section 4.3.4 of section three, the findings of the suggested approach are compared with the results described in Chapter II of the literature.

4.2 Signal Processing Method

4.2.1 Experimental Setup

The experimental setup is presented in Table 2 with values. The test images are obtained from the hospital as a secondary image. The segmentation of the median nerve was implemented on the OpenCV image processing library using open-source python language. Each of the 35 test images has a varied image quality, and contrast, and has a different structure or form of the median nerve.

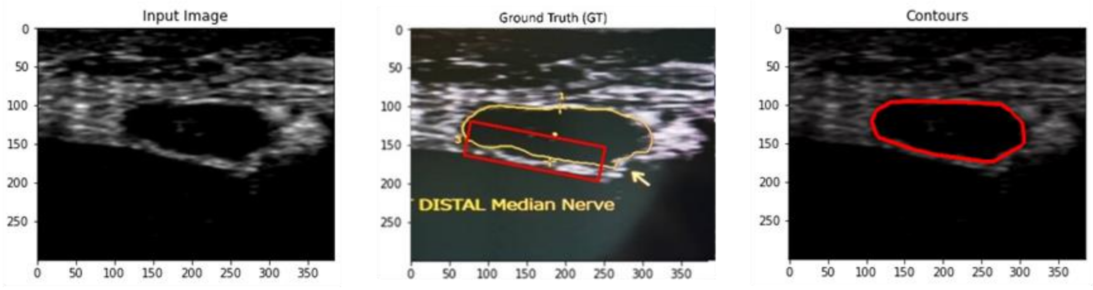
Table 2 Experimental Setup Environment

System	Name	Specs/version
Hardware	Personal Laptop	Lenovo, 16GB RAM, Intel Core i7-9750H CPU @ 2.60GHz - 5.00GHz, GPU @ NVIDIA Quadro T2000
Programming Language	Python	3.7
Editor	VS Code	1.58
Test Image	Ultrasound image	35 test images with a size of 864 × 648 pixels
Library	Open CV	4.5.4

4.2.2 Experiment Result

A qualitative evaluation was carried out by visually assessing how well the segmented findings fit the genuine median nerve border. Figure 31 shows a comparison of expert-segmented median nerves with the proposed study's median nerve segmentation.

Figure 31(a) depicts the initial wrist ultrasound scan, which reveals structural information about the median nerve. Figure 31(b) depicts an image segmented manually by an expert sonographer, whereas Figure 31(c) depicts an image segmented using the proposed signal processing technique. In contrast to the manually segmented image (Figure 31(b)), the proposed method's segmented output (Figure 31(c)) displays the contour outline placed onto the true median nerve boundary. Figure 31(b) shows that a piece of the true median nerve boundary has been left unsegmented, as indicated by the little red rectangle. As can be observed, the proposed segmented image has a higher level of precision and accuracy in median nerve segmentation.



(a) (b) (c)

Figure 31 Median Nerve Segmented (a) Original Image, (b) Median Nerve Segmented by A Sonographer, and (c) Median Nerve Segmented by the Proposed Algorithm

In addition to the qualitative evaluation, the accuracy of the model was further confirmed by comparing the automatic findings to the ground truth which was manually segmented by the sonographer. Equations (3.5) to (3.8) indices were used to measure the performance of the proposed method.

The algorithm is tested on 35 ultrasound images and obtained a precision, recall, F1-score, and Jaccard similarity of 0.87,0.93, 0.76, and 0.93 respectively, which shows that there is a higher degree of overlap between the segmented and ground truth image. The ground truth is achieved by conducting manual area tracing on the median nerve using GIMPS 2.10.28 (GNU Image Manipulation Program) a free and open-source raster graphic tool on the manually marked ultrasound image by the expert sonographer. When the tracing is completed, the selected region is highlighted and changed to a binary image (foreground as white and background as black).

Figures 32 (a) to (h) illustrate some of the median nerve segmentation findings utilizing the suggested technique. The image without a contour is the test image or input image, while the image marked with a red contour is the corresponding segmented image from the proposed method. The real boundary of the median nerve is precisely segmented, as can be seen in the result images. Figure 32 (a) and (b) represent the segmentation result when the input image is highly clean, with a defined border between the actual median nerve structure and the surrounding. Figures 32 (c) to (f) show the segmentation result from the second situation when the contrast of an input image is not equally distributed and the true border between the genuine median nerve is noisy. It also depicts the situation when the structure of the median is of different forms. Figures 32 (g) and (h) show the segmentation result from the proposed system in the worst-case situation when the input ultrasound image cannot discriminate between the median nerve structure and its surroundings, and there is no discernible border between the structure of the median nerve and its surroundings. The resilience and the robustness of the suggested model in segmenting the median nerve efficiently in ultrasound images despite higher variability in the input image are demonstrated by these different case outcomes.

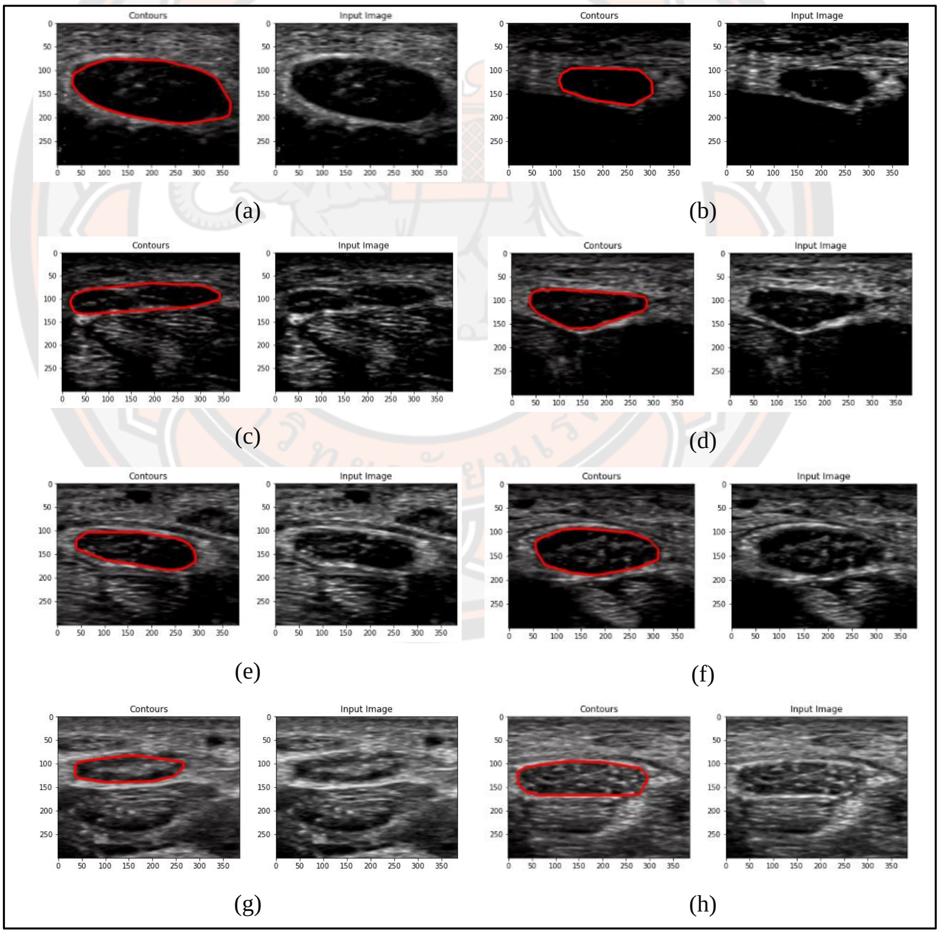


Figure 32 Segmentation Result of Median Nerve Ultrasound Images Using Signal Processing

4.2.3 Discussion

To further validate the method, the cross-sectional area (CSA) of the segmented median nerve was determined and compared to the CSA of ground truth (GT). The median nerve's cross-sectional area (CSA) is the most often utilized metric to assess and quantify the CTS, with diagnostic cut-off values ranging from 9 mm^2 to 14 mm^2 in various analytical settings (Ana Torres-Costoso et al., 2018). According to nerve conduction studies (NCS) research, the value of 9 mm^2 for MNA is the most reliable cut-off value for MN pathology during CTS.

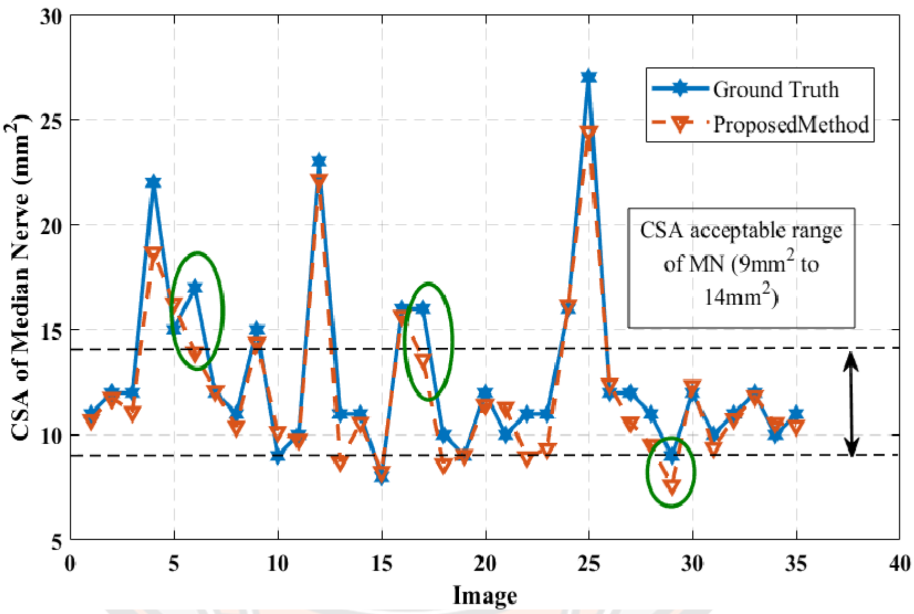


Figure 33 Comparison of MN Cross-Sectional Area (CSA) Between the Original Image (Ground Truth) & Segmented Image (Algorithm)

Figure 33 compares the cross-sectional area (CSA) of the median nerve between the original image or ground truth (GT) and the segmented median nerve from the proposed algorithm. The blue solid line represents the CSA value calculated by the expert and the brown dash line illustrates the CSA value of the median nerve calculated using the proposed algorithm. It is evident from the figure that there is a significant correlation of CSA between the original image and the segmented image with a close resemblance of over 90%. This promising result justifies that, the proposed algorithm could be used in clinical practice to aid in diagnosing the CTS in ultrasound images. The average and standard deviation from these two cases are determined and shown in Table 3.

Table 3 shows the average CSA value of ground truth and the mean value determined using the proposed method. The ground truth and proposed methods yielded average CSA values of 12.8 mm^2 ± 4.384 mm^2 and 11.95 mm^2 ± 3.909 mm^2, respectively. The table shows that the average CSA value acquired using the proposed method is nearly identical to the value determined by the professional sonographer. Furthermore, the standard deviation determined from the proposed method is also equivalent to the ground truth. This demonstrates that the model performs in the same way as the ground truth. It indicates that the proposed approach is more uniform and resilient in the segmentation of the median nerve. It also concludes that the proposed model is more stable and effective. As a result, it can be concluded that suitable data pre-processing introduced before the feature extraction model produces a more accurate result in the segmentation of median nerve in ultrasound images. Additionally, it does not demand the sonographer to manually detect and identify the median nerve to diagnose carpal tunnel syndrome (CTS), which would require a lot of experience and time. As a result, the proposed approach could be employed to segment the median nerve in an ultrasound machine to diagnose CTS.

However, the model has estimated the CSA of the median nerve as false positive (represented by the green circle marked in Figure 33) in some situations, even when the expert or sonographer evaluated it as true positive. Image numbers 6 and 17 in Figure 33 are classed as normal (i.e., the 13.9 mm^2 and 13.8 mm^2 are CSA of the segmented image, and 17.32 mm^2 and 16 mm^2 are the CSA calculated by the expert sonographer, respectively). Additionally, for image number 29, from the proposed algorithm, it has been determined as abnormal with the CSA value of 9.2 mm^2 while this has been assigned as a normal case by the expert. This may be the outcome of excessive erosion and dilatation during the morphological procedure. These issues could be further analyzed and could be solved by using a more precise techniques like artificial intelligence and deep learning.

Table 3 The Mean and Standard Deviation of Cross-Sectional Area (CSA) Value from Original Image (or Ground Truth) and Proposed Algorithm

Model	CSA (Mean ± SD) in mm^2
Original Image (Ground Truth)	12.8 ± 4.384
Proposed Algorithm	11.95 ± 3.909

To show the degree of similarities in greater detail, statistical analysis is performed using Microsoft Excel (Microsoft 365). The correlation coefficient (also known as Pearson's product-moment "*r*") is used to determine the strength of the CSA's linear relationship with the ground truth and segmented image. In the examination of 35 test data, the CSA values from ground truth and the segmented image had a correlation coefficient of 0.962. These results suggest that the two CSAs (calculated and ground truth) have a stronger association and again confirm that the proposed algorithm is a promising method for segmenting the median nerve.

To further present the usefulness of the proposed algorithm, some key performance indicators obtained from the proposed algorithm and work done previously are compared as shown in Table 4.

Table 4 Comparison of Proposed Algorithm with other Method based on Signal Processing Algorithm

Model	Precision (%)	Recall (%)	F-1 Score (%)	Jaccard Similarity (%)
Wang's Method	85	91	75	-
Proposed Algorithm	87	93	76	93

Table 4 shows the comparison of the proposed algorithm with the work done by Wang based on the signal processing techniques to detect the median nerve in ultrasound images. In Wang’s work, a greedy active contour-based detection framework was used to detect the median nerve in ultrasound images. When tested on the test image, it is obtained a mean precision, recall, and F-1 score values of 85%, 91%, and 75%, respectively. The reference contour remains a factor in contour computation and contour segmentation may produce an incorrect result due to a defective contour in the reference image. The proposed algorithm obtained a mean precision, recall, and F-1 score value of 87%, 93%, and 76%, respectively; which are 2%, 2%, and 1% higher compared to Wang’s method. Note that there is no information about Jaccard Similarity provided by Wang’s method. The obtained improvement is from the pre-processing and additional signal processing techniques used in the proposed algorithm. This demonstrates that the proposed method is more dependable and consistent in median nerve segmentation.

This signal processing approach has been proven to be effective in segmenting the median nerve in ultrasound images to compute its properties. In practice, the methodology can aid in the development of a viable tool for detecting CTS in ultrasound pictures. This method may also be used to predict the outcome of CTS therapy. The process of feature extraction, however, is where this technique's shortcoming lies. It took more time and processing to manually modify the kernel's weight and iterate numerous times in the morphological operation to segment the median nerve. Furthermore, the approach was not reliable and favored certain feature images over others. There is a considerable risk the model may not partition or segment the median nerve correctly. It was also discovered that only 35 of the 70 ultrasound images could be segmented using this method. As a result, further segmentation of median nerve in ultrasound images is done using the AI method to make it more robust, biased-free, and quicker.

4.3 Artificial Intelligence Method

4.3.1 Effect of Different Learning Rates on Training Model

Learning rate is a hyper-parameter that governs how much to update the network or model weights concerning the loss gradient. It also determines how faster to update the weights during the training process to learn the features. Choosing a correct learning rate is difficult, the lower the value, the slower the descent. While adopting a low learning rate may be a good idea for ensuring that the model does not miss any local minima, it may also mean that it will take a long time to converge particularly if it gets trapped on a plateau region, while a value that is too high may result in learning a sub-optimal set of weights too quickly or an unstable training process. We exercised the hyper parametrization of the learning rate from 0.001 (10^{-3}) to 0.000001(10^{-6}) to see the effect on the training model. The result of each learning rate based on the loss curve is discussed in Figure 34.

Figures 34 (a) to (d) show the loss curve of different learning rates during the training process. Figure 34 (a) shows the loss curve for the learning rate of 0.001 (10^{-3}). The loss started to decrease (loss = 0.03) till the 9 epochs, however, instead of decreasing further the training and validation loss suddenly started to rise high (loss =1) after the 9th epoch which indicates that the model is suffering from underfitting. Figure 34 (b) and (c) show the loss curve of the 0.0001 (10^{-4}) and 0.00001 (10^{-5}) learning rate respectively. Compared to 0.001 (10^{-3}), it converges the loss more toward an optimal point (i.e., 0.102864 & 0.3528 of training and validation loss respectively for 0.0001 (10^{-4}) and 0.10405 & 0.20505 of training and validation loss respectively for 0.00001 (10^{-5})), however, there is a large gap between the validation and training loss which depicts that model is still experiencing the overfitting problem.

Figure 34 (d) shows the loss curve of the 0.000001 (10^{-6}) learning rate. Compared to 0.001 (10^{-3}), 0.0001 (10^{-4}), and 0.00001 (10^{-5}), it regularizes the training more smoothly with a negligible discrepancy between the validation and training cost function and learns more features from the image hence, solving the problem of the model overfitting. However, it took a long time compared to other learning rates to complete the training. The training time increases as the learning rate decreases. Table 5 shows the different times taken to train the model with different learning rates.

Table 5 Training Time for Different Learning Rates

Learning Rate	Training Time (hrs.)	Training Loss	Validation Loss
10^{-3}	1	1	1
10^{-4}	10	0.102864	0.3528
10^{-5}	14	0.10405	0.20505
10^{-6}	20	0.015379	0.022968

From Table 5 it is observed that the training time increased as the learning rate of the model used is smaller and vice versa. The model took around 1 hour to train with a 10^{-3} learning rate and 10 hours for 10^{-4}. The model required 14, and 20 hours to train with learning rates of, 10^{-5}, and 10^{-6}, respectively. It can be deduced that the smaller learning rate takes a long time to train the model and learns more features during the training process.

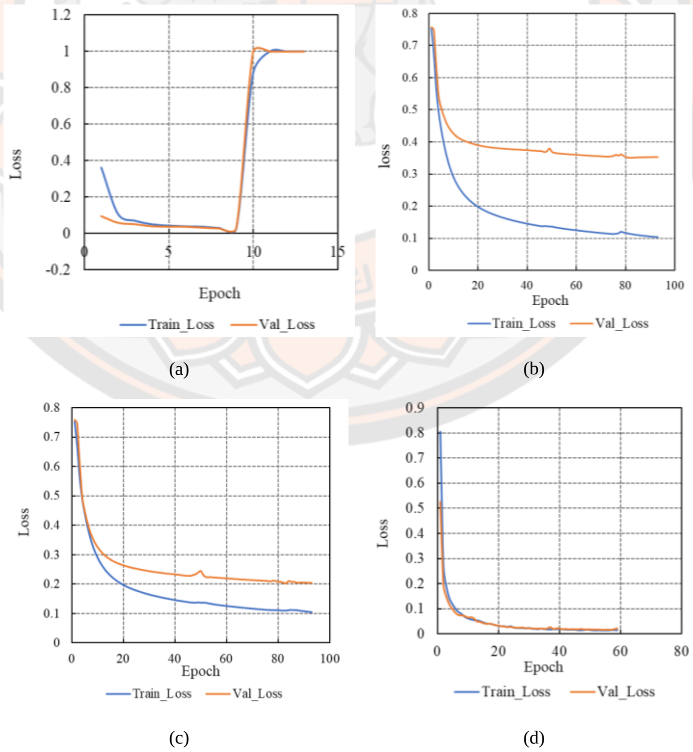


Figure 34 Loss Curve and Training Curve with Different Learning Rates for Model Training: (a) Loss curve for 10^{-3}, (b) Loss curve for 10^{-4}, (c) Loss curve for 10^{-5}, and (d) Loss curve for 10^{-6}

4.3.2 Training and Test Result of Three Different Approaches

In this study, we used three distinct sets of approaches to training the model i.e., the base U-Net, U-Net with pre-processed data, and U-Net with pre-processed, augmented data, and batch norm layer, and their accuracy and other evaluation indices are compared. The findings of each of the distinct sets are discussed in depth in the following paragraphs.

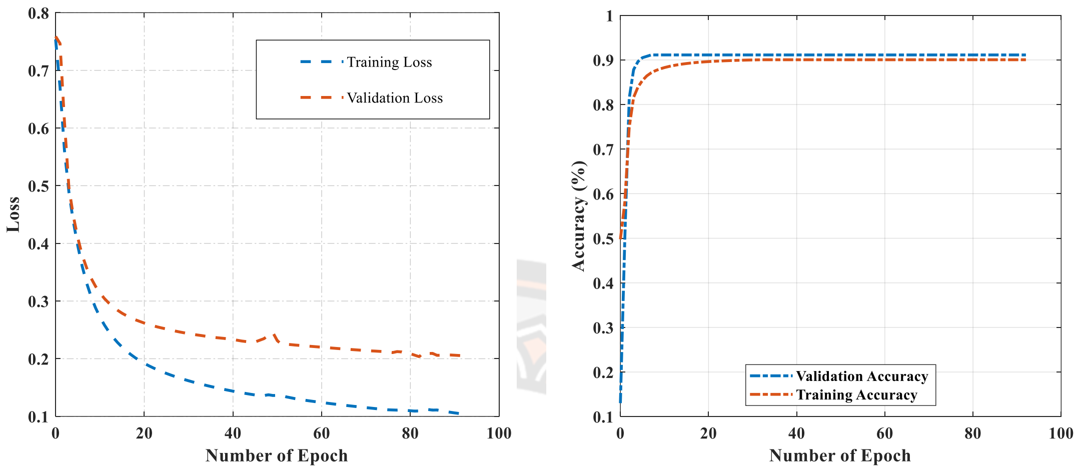
4.3.2.1 Base U-Net

In the first approach, the basic U-Net, or the original model constructed by (Ronneberger et al., 2015), was trained on the median nerve datasets. Adam is used as the optimizer with loss function as dice loss. The model was trained for 200 epochs with a batch size of 16 and a learning rate of 10^{-6}. The datasets are split into training and test data in a 90:10 ratio.

Training Result

Figures 35 (a) and (b) show the training loss curve and accuracy curve of the training process. The training loss represents how well the model fits the training data, whereas the validation loss represents how well the model fits new data. The training stopped to learn after the 91st epoch because of the early stopping. Early stopping is a technique that allows programmers to define an arbitrary number of training epochs and then stop training whenever the model's performance on the validation dataset stops learning. Figure 35 (a) shows that the model converges quickly over the first 10 epochs, with a very negligible difference between the validation and training losses. However, after the tenth epoch, the training loss decreases adequately, but the validation loss decreases slowly, resulting in a wide difference between them which shows the model is overfitting the data. This might be due to the fewer training data to learn the feature. The validation and training loss both dropped from 0.75884 to 0.20505 and 0.75357 to 0.104053, respectively.

Figure 35 (b) represents the accuracy curve during the training process. It depicts how well the model can categorize two images during training on the training and validation datasets. (A validation dataset is a sample of data held back from model training that is used to measure model competence while the model is being trained.) It is evident from the figure, that the model can classify the training data with an average validation and training accuracy of 90.2% and 88.8%, respectively, with a minor difference between them.



(a) (b)

Figure 35 The Loss Curve and Accuracy Curve (a) Training and Validation Loss Curve (b) Training and Validation Accuracy Curve

Experiment Result (Test Result)

To evaluate localization, accuracy Jaccard Similarity, recall, precision, and F1 score were used as performance indicators, while segmentation was assessed using the Dice coefficient measurement metric (the details are discussed in subsection 3.6.3 of Chapter III). Table 6 tabulates the model performance indices of the test results on the test datasets for base U-Net. The U-Net achieved an average accuracy, Jaccard Similarity, recall, precision, F1, and Dice similarity coefficient (DSC) of 0.857 ± 0.0961, 0.541 ± 0.20622, 0.765 ± 0.21889, 0.622 ± 0.2064, 0.662 ± 0.2026, and 0.540 ± 0.21738 respectively. Although the Jaccard similarity result is lower than that of the signal processing approach (Jaccard similarity = 0.93), the U-Net can exercise the biased problem of the signal processing method. It was able to locate the median nerve of different structures; however, the segmentation of the median is not very perfect as the ground truth.

Table 6 Test result of Base U-Net Model

Model	Accuracy	Jaccard Similarity	Recall	Precision	F1 Score	DSC
U-Net	0.857 ± 0.0961	0.541 ± 0.2062	0.765 ± 0.2188	0.622 ± 0.2064	0.662 ± 0.2026	0.540 ± 0.21783

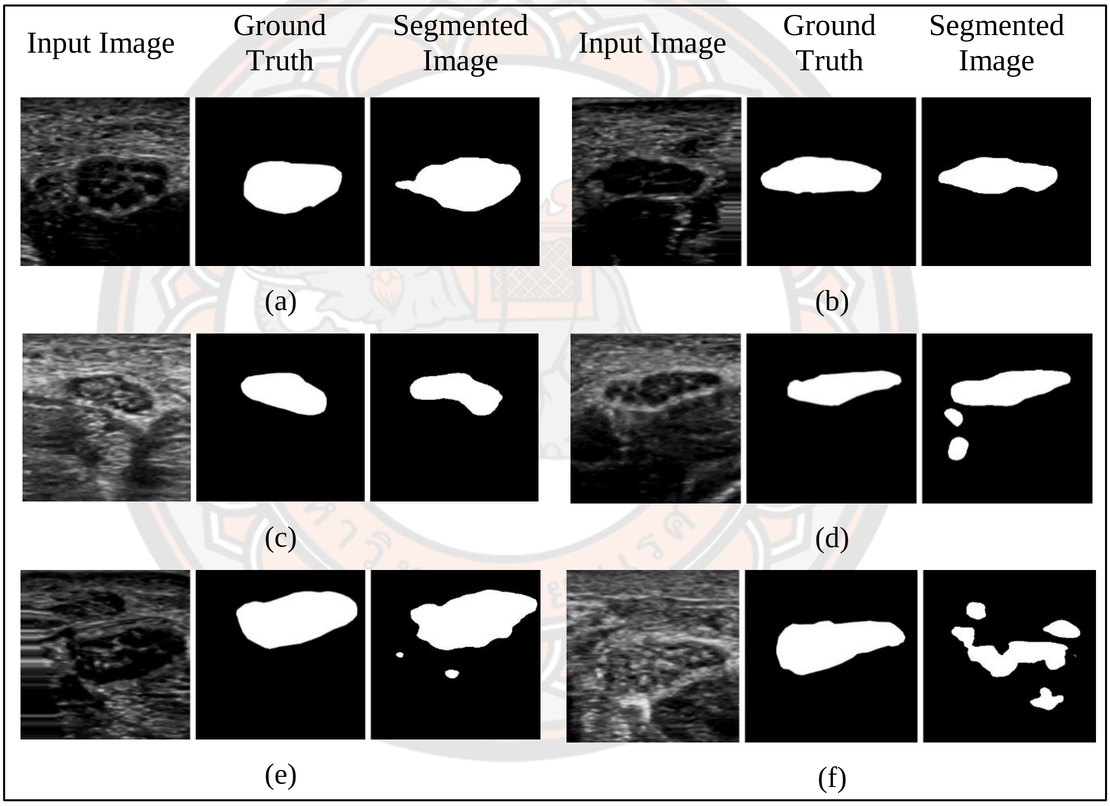


Figure 36 Segmentation Result of Median Nerve in Ultrasound Images from base U-Net Model

The segmentation results from the base U-Net architecture are enumerated in Figures 36 (a) to (f). It is evident from the segmented result; that the model is not able to perform the segmentation of the median nerve perfectly. Figures 36 (a) to (c) depict that the model has segmented the median nerve very differently when compared to the ground truth. Figures 36 (d) and (e) portray the second case scenario, in which the model segmented various minor structures in addition to the genuine median nerve structure, and the segmented output does not match the ground truth image perfectly. Figure 36 (f) illustrates the worst-case scenario, in which the model fails to locate the median nerve structure and the segmented image becomes utterly deformed. From this result, it is observed that the U-Net struggles to predict the perfect structure and edges of the median nerve well since the input datasets contain the noise. In addition, as compared to the training accuracy, the model accuracy for segmenting the median nerve on test datasets is relatively poor.

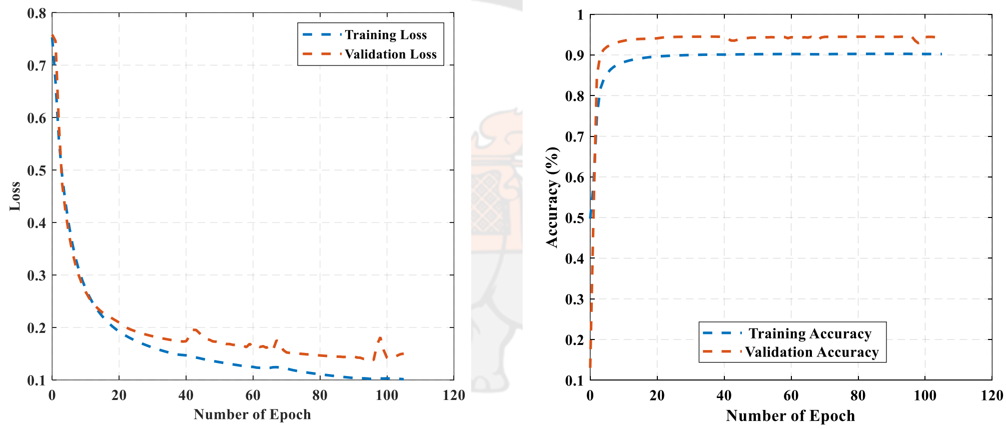
4.3.2.2 U-Net with Data Pre-processing

To improve the accuracy and raise the concerns in approach one, data pre-processing is adapted to clean or filter the noise in approach two. When dealing with noisy, inconsistent, or missing data, pre-processing is essential, and it is typically employed to conduct procedures that reduce the data's complexity. It not only cleans the data but also improves the algorithm's accuracy (Masoudi et al., 2021). Image Enhancement (intensity modification), ROI selection, and image denoising are some of the data pre-processing techniques used in this study (the details of each of these techniques are discussed in sub-section 3.3 of Chapter III). The model was trained for 200 epochs with a batch size of 16 and a learning rate of 10^{-6}. Adam is used as the optimizer with loss function as dice loss and the train-test split ratio of 90:10.

Training Result

Figures 37 (a) and (b) show the training loss curve and accuracy curve of the training process. The training stopped to learn after the 109^{th} epoch because of the early stopping. The model converges swiftly throughout the first 17^{th} epochs, with just a small discrepancy between the validation and training losses (Figure 37 (a)). The training loss drops sufficiently after the 17^{th} epoch, however, the validation loss updates weights in tiny stages, thus it declines slowly with some oscillation in between. As a result, after the 40^{th} epoch, there is a slight difference between training and validation loss. However, when compared to approach one, the difference is small. This might result in an adequate balance of train and validation data and data continues to be overfitted by the model. Both the validation and training losses decreased from 0.7581 to 0.1480 and 0.7502 to 0.1002, respectively. When compared to approach one, the cost functions for validation and training dice loss dropped by 0.057 and 0.0038, respectively.

The training accuracy curve during the training procedure is depicted in Figure 37 (b). The model can classify the training data with an average validation and training accuracy of 94.1% and 90.1%, respectively. When compared to approach one, the disparity between train and validation accuracy is greater. However, there is a 1.3% and 3.9% improvement in training and validation accuracy, respectively. It shows how the data pre-processing approach evaluated the model's ability to classify and segment images during training on both the training and validation datasets.



(a) (b)

Figure 37 The Loss Curve and Accuracy Curve for the U-Net Model with Pre-processed Data (a) Training and Validation Loss Curve (b) Training and Validation Accuracy Curve.

Experiment Result (Test Result)

The model performance indices of the test results on the test datasets for approach two are presented in Table 7. The accuracy, Jaccard similarity, recall, precision, F1, and Dice similarity coefficients (DSC) of 0.954 ± 0.0242, 0.718 ± 0.1009, 0.888 ± 0.1079, 0.762 ± 0.1216, 0.814 ± 0.1063, and 0.820 ± 0.1078 are obtained, respectively. Overall, there is a 9.7% gain in model accuracy when compared to approach one.

Table 7 Test Result of U-Net with Data Pre-processing

Model	Accuracy	Jaccard Similarity	Recall	Precision	F1 Score	DSC
U-Net + Preprocessed data	0.954 ± 0.0242	0.718 ± 0.1009	0.888 ± 0.1079	0.762 ± 0.1216	0.814 ± 0.1063	0.820 ± 0.1078

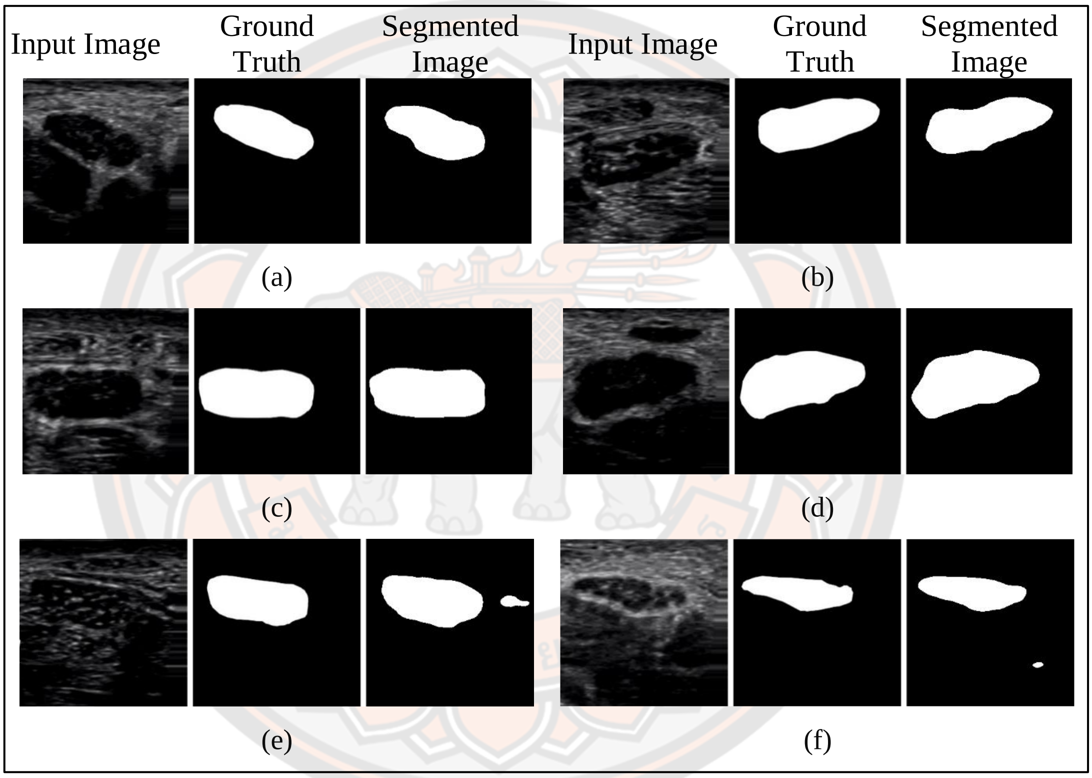


Figure 38 Segmentation Test Result of Median Nerve in Ultrasound Images using U-Net model and data pre-processing

Figures 38 (a) to (f) show the segmented test results from U-Net with a data pre-processing approach. The model can conduct the median nerve segmentation more accurately than approach one, as seen by the segmented outcome. Figures 38 (a) to (d) show the model has segmented the median nerve in a way that is remarkably similar to the ground truth. However, the segmented image is still far from flawless, and in some cases, the model has over-segmented beyond the nerve border (Figure 38 (a)) and left other areas unsegmented (Figure 38 (b)). Figures 38 (e) and (f) show that, in addition to the true median nerve structure, the model still segmented certain small structures. It is clear from this finding, the small number of input datasets, that the U-Net struggles to anticipate the ideal structure and edges of the median nerve.

4.3.2.3 U-Net with Data Augmentation and Data Pre-processing

One of the most prevalent issues with approaches one and two is that the model is unable to generalize, resulting in overfitting the training data. Generalizability is the performance difference of a model when evaluated on previously observed data (training data) vs data it has never seen before (test data). It is evident from Figure 35 (a) and Figure 37 (a), that approaches one and two are experiencing overfitting issues i.e., the large gap between the training and validation cost function. Models with poor generalizability overfit the training data. The validation error must continue to decrease in conjunction with the training error to develop usable deep learning models with zero gaps between them. Transfer learning, pre-training the model, dropouts, batch normalization, and data augmentation are some of the techniques used to solve this problem (Shorten & Khoshgoftaar, 2019). A comprehensive list of deep learning regularization algorithms has been published by Kukačka et al. (2017). Data augmentation and batch normalization are two approaches used in this method to address the concerns of approaches one and two.

Data augmentation is the practice of enhancing variety and variation in existing data by applying various geometrical transformations to it. This is done to extend the datasets and expose the CNN to a diverse variety of data during the testing or classic discriminative phase, allowing the network to detect data in any shape or form. The augmented data will reflect a broader range of potential data points. Data Augmentation is comparable to dreaming or imagining. Based on their previous experiences, humans create various situations. Imagination aids us in gaining a deeper knowledge of the world around us. Due to the rarity of illnesses, patient privacy, the need for medical professionals to label patients, and the cost and manual labor required to perform medical imaging operations, it is extremely difficult to build large medical image databases. As a result, augmentation aids in including a wider variety artificially and improving the medical data for deep learning training. Geometrical transformations such as rotation, horizontal flipping, sheering, zooming, and translation are used in this study (the details of each are discussed in sub-section 3.5.1 of Chapter III). Thus, by employing these transformations, the datasets are multiplied by 56, resulting in 4200 number images. It not only enhances the data but also artificially added a wide range of different variations of images in datasets.

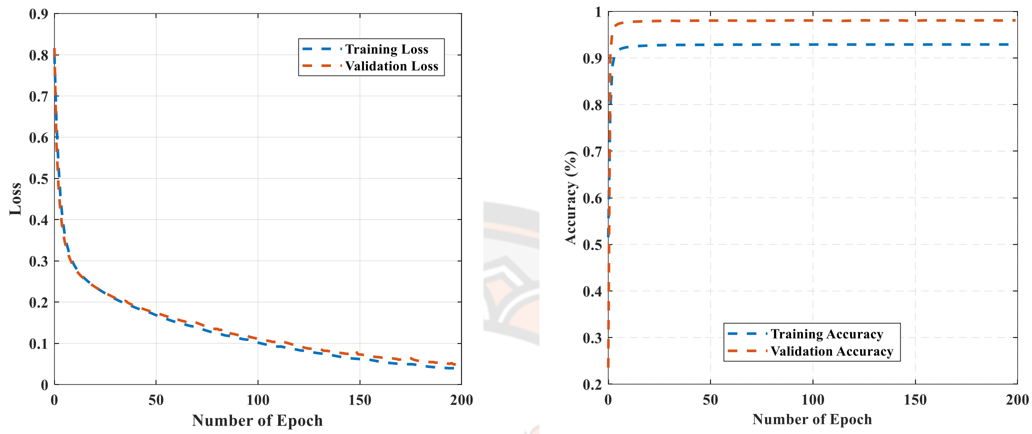
The inclusion of the batch normalization (BN) (Ioffe & Szegedy, 2015) layer in the network was one improvement made to the original U-Net in this study to regularize, speed up, and boost the accuracy of the training. Batch normalization (BN) is a method for normalizing activations in deep neural networks' intermediary layers. BN has been a popular deep learning approach due to its ability to enhance accuracy and speed up training. When training deep networks, BN enhances convergence and generalization by allowing for a high learning rate and avoiding overfitting. It coordinates the updates of various layers in a model by standardizing the activations of each input variable in every mini-batch, such as the activations of a node from a previous layer. The benefit of adopting batch normalization, data augmentation, and data pre-processing are discussed in the following result.

Training Result

Figures 39 (a) and (b) show the training loss curve and accuracy curve of the training process. Adam is used as the optimizer with loss function as dice loss. The model was trained for 200 epochs with a batch size of 16 and a learning rate of 10^{-6}. The datasets are split into training and test data in a 90:10 ratio.

It is apparent from Figure 39 (a), that the model converges swiftly throughout the training process with a negligible discrepancy between the validation and training cost function. In comparison to approaches one and two, the model generalizes and converges faster. This undercuts the effectiveness of the adopted methodology in resolving the issue in approaches one and two. The validation and training losses, respectively, fell from 0.8163 to 0.047 and 0.8006 to 0.039.

The accuracy curve during the training procedure is depicted in Figure 39 (b). The model can classify the training data with an average validation and training accuracy of 97.5% and 92.5%, respectively. When compared to approach two, the disparity between train and validation accuracy is greater. However, there is a 2.4% and 3.4% improvement in training and validation accuracy, respectively. It demonstrates the influence of the combined approaches such as data augmentation, data pre-processing, and batch normalization to identify and segment images in the training and validation dataset.



(a) (b)

Figure 39 The Loss Curve and Accuracy Curve for Modified U-Net Model (Batch Normalization Layer) with Pre-processed Data and Data Augmentation (a) Training and Validation Loss Curve (b) Training and Validation Accuracy Curve

Experiment Result (Test Result)

Table 8 shows the model performance indices for the test results on the test datasets for the third approach. The accuracy, Jaccard similarity, recall, precision, F1, and Dice similarity coefficients (DSC) are correspondingly 0.998 ± 0.0032, 0.891 ± 0.0990, 0.989 ± 0.0591, 0.898 ± 0.0893, 0.941 ± 0.0632, and 0.899 ± 0.0990. When compared to approach two the combined procedures (data pre-processing, data augmentation, and batch normalization) resulted in a 4.4% increase in model accuracy and 14.1% when compared to approach one. Concurrently there is also an increase in the other measuring indices. It can be also observed that the standard deviation (SD) of each index is much smaller than approaches one and two. This ascertains that approach three is more consistent in segmenting the median nerve in ultrasound images.

Table 8 Test Result of Modified U-Net with Data Pre-processing and Data Augmentation

Model	Accuracy	Jaccard Similarity	Recall	Precision	F1 Score	DSC
U-Net + Preprocessed data +Data Augmentation + Batch Norm	0.998 ± 0.0032	0.89 ± 0.0990	0.989 ± 0.0591	0.898 ± 0.0893	0.94 ± 0.0632	0.89 ± 0.0990

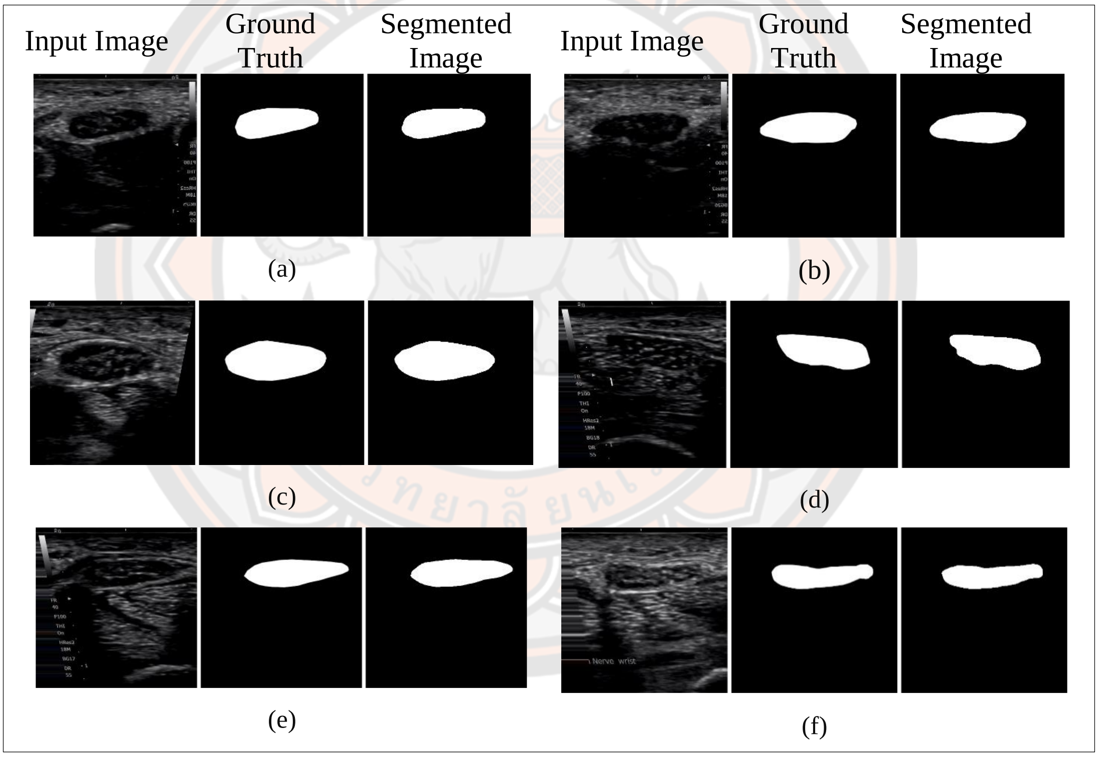


Figure 40 Segmentation Result of Median Nerve in Ultrasound Images using U-Net Model with Pre-processing, Augmentation, and Batch Norm Layer

Figures 40 (a) to (f) show the segmentation results from U-Net trained with data pre-processing and data augmentation with batch norm layer. It is evident from Figure 40, that the segmented median nerve overlaps the ground truth. Figure 40 (a) shows the model has segmented the median nerve in a way that is remarkably like the ground truth when the input image has a clear boundary between the edges of the median nerve and its surrounding. Figure 40 (b) and (c) illustrate the segmentation result when there is no distinct boundary between the border of the median and its surrounding. For example, in Figure 40 (b), there is no clear line separating the median nerve from its surroundings at the bottom half of the image. Figures 40 (d), (e), and (f) show the segmentation result of the proposed system in the worst-case scenario, where the input ultrasound image is unable to distinguish between the median nerve structure and its surroundings, and there is no discernible border between the median nerve structure and its surroundings. Furthermore, all kinds of the median nerve are unique from one another and have varying degrees of intensity. In all these scenarios, the model was still able to segment the median nerve.

From these results, it is apparent, that the resilience and the robustness of the suggested model in segmenting the median nerve efficiently in ultrasound images despite higher variability in the input image are demonstrated by these different case outcomes. The combined method of data pre-processing, data augmentation, and inclusion of batch normalization layer in the U-Net has addressed the problems expressed by the signal processing method, as well as those raised by approaches one and two.

4.3.3 Comparison of Experimental Results of 3 Approaches

Table 9 and Figure 41 provide the overall summary findings of the measurement indices and cost function curve of the three approaches presented in the previous part, as well as a comparison between them. It is evident from Table 9, that the U-Net model trained with pre-processed data, augmented data, and the inclusion of a batch norm layer outperforms the two previous models and produces remarkable median nerve segmentation. When tested on test datasets, the accuracy was 99.8 % $\pm$ 0.0032, which is 14.1% and 4.4% greater than approach one and approach two, respectively. A Jaccard similarity coefficient of 0.891 was obtained, which is 0.039 lower than the signal processing approach but 0.173 and 0.35 higher than approaches two and one respectively. The approach was also extremely effective in locating the median nerve, with a DSC of 0.899 $\pm$ 0.0990, which was significantly higher than the other two methods. This demonstrates that when deep learning is given more training data and input data is cleansed, it gives more accurate results. Moreover, the standard deviation of all the performance indexes of approach three is comparatively lower than approaches one and two which signifies that the model is more reliable.

The highest Dice measuring parameter consistently showed that approach three segmentation worked best of all approaches. The suggested approach provides an end-to-end mechanism for segmenting the median nerve that requires no user involvement and successfully locates and segments portions of the median nerve without the requirement for an initial localization operation.

Table 9 Comparison of Different Measuring Indices of the Three Different Methods

Model	Accuracy	Jaccard Similarity	Recall	Precision	F1 Score	DSC
U-Net	0.857 ± 0.0961	0.541 ± 0.2062	0.765 ± 0.2188	0.622 ± 0.2064	0.662 ± 0.2026	0.540 ± 0.21783
U-Net + Pre-processed data	0.954 ± 0.0242	0.718 ± 0.1009	0.888 ± 0.1079	0.762 ± 0.1216	0.814 ± 0.1063	0.820 ± 0.1078
U-Net + Pre-processed data +Data Augmentation + Batch Norm	0.998 ± 0.0032	0.89 ± 0.0990	0.989 ± 0.0591	0.898 ± 0.0893	0.94 ± 0.0632	0.89 ± 0.0990

The cost functions of the three methods are compared in Figure 41. The U-Net model trained with pre-processed data augmented data, and the addition of a batch norm layer model converges quickly during the training process, with a small disagreement between the validation and training cost functions which is denoted by the red curve and continues to learn the feature throughout the training process. The model generalizes and converges quicker than approaches I and II thus, solving the problem of overfitting that was experienced in approaches one and two.

The training and validation loss of the base U-Net approach (I) and the U-Net with data pre-processing (approach II) decreases smoothly with a small difference until the 20th epochs, which are represented by orange and green curves, respectively. The training loss, on the other hand, lowers significantly after the 20th epoch, and the validation loss updates weights in modest steps, so it declines slowly for both approaches. As a result, the discrepancy between training and validation loss widened, resulting in an overfitting issue.

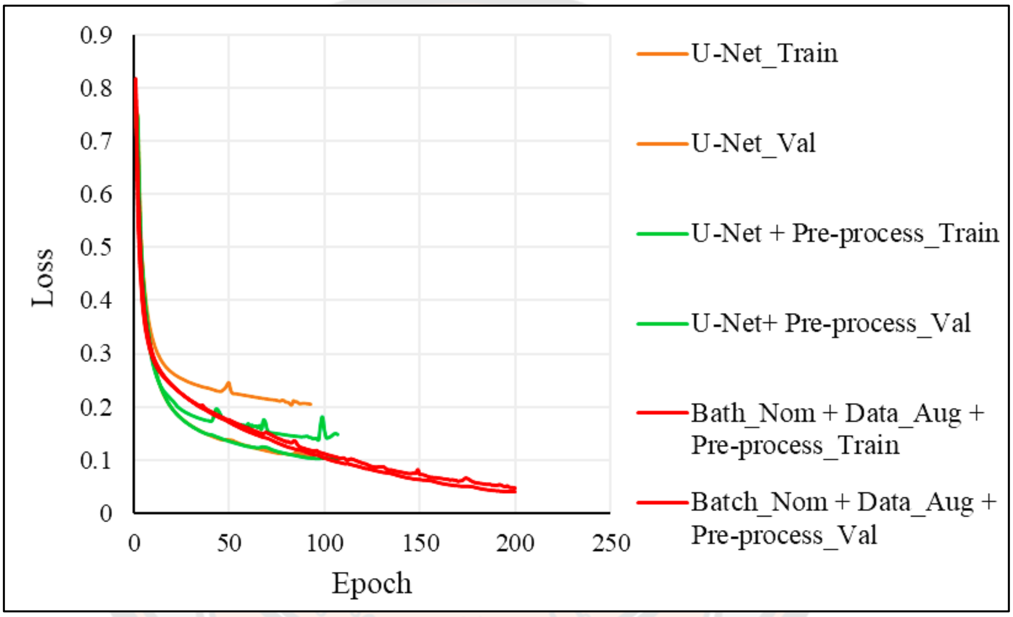


Figure 41 Comparison of the Loss Curve of Three Methods

4.3.4 Comparison of the Proposed Method with Methods in the Literature

i. Comparison of Proposed Method with Convectional Active Contour Method

Table 10 illustrates the comparison of DSC achieved by some convectional active contour methods such Chan-Vese (CV) method (Chan & Vese, 2001), Distance regularized level set method (DRLS) (Hadjerci et al., 2016; Li et al., 2010), Probabilistic gradient vector flow method (PGVF) (Hafiane et al., 2014), Localization + CV (Hadjerci et al., 2016), Localization + DRSL (Hadjerci et al., 2016), and Localization + PGVA (Hadjerci et al., 2016) to the proposed method. Traditional image analysis algorithms for segmenting the median nerve frequently use the active contour model as a foundation. Segmentation issues have arisen as a result of the reliance on the reference or starting contour. It is evident from the table, the DSE attained by deep learning (proposed technique) surpassed the best standard contour approach, Localization + PGVA (Hadjerci et al., 2016), which produced satisfactory segmentation results with an average DSE of 0.81 $\pm$ 0.10, which is 0.089 fewer than deep learning. This indicates that when compared to the conventional technique (active contour), deep learning solutions are more accurate, resilient, bias-free, and more effective in locating regions of the median nerve and directly segmenting them without the need for an initial localization procedure. Furthermore, unlike the active contour approach, it is more robust in that it can segment the median nerve with varying morphology.

Table 10 Comparison of DSC for Convectional Active Contour Techniques and the Proposed Approach for Segmenting the Median Nerve

Model	**DSC**
Chan-Vese (CV) method (Chan & Vese, 2001)	0.09 ± 0.13
Distance regularized level set method (DRLS) (Hadjerci et al., 2016; Li et al., 2010)	0.13 ± 0.03
Probabilistic gradient vector flow method (PGVF) (Hafiane et al., 2014)	0.75 ± 0.15
Localization + CV (Hadjerci et al., 2016)	0.69 ± 0.11
Localization + DRSL (Hadjerci et al., 2016)	0.71 ± 0.18
Localization + PGVA (Hadjerci et al., 2016)	0.81 ± 0.10
U-Net + Pre-processed data +Data Augmentation + Batch Norm	**0.899 ± 0.099**

ii. Comparison of Proposed Method with other Deep Learning Methods

Table 11 lists the performance indices of the various methods used to segment the median nerve, which is compared to the proposed approach. The method includes Lightweight U-Net (Horng et al., 2020), U-Net + Mask Track (Horng et al., 2020), ConvLSTM + U-Net + Mask Track (Horng et al., 2020), and DeepNerve (Horng et al., 2020) which integrates the characteristics of Mask- Track, and ConvLSTM based on the lightweight version of U-Net. The ConvLSTM layer is placed at the bottom of the U-Net +Mask Track to reduce the computation time.

Table 11 Comparison of the Performance Indices of Proposed Method with other Deep Learning Models

Model	Accuracy	Recall	Precision	F1 Score
Lightweight U-Net (Horng et al., 2020)	0.993 ± 0.003	0.702 ± 0.2618	0.772 ± 0.2441	0.735 ± 0.2387
U-Net + Mask Track (Horng et al., 2020)	0.993 ± 0.0048	0.897 ± 0.1345	0.7668 ± 0.1520	0.810 ± 0.1391
ConvLSTM + U-Net + Mask Track (Horng et al., 2020)	0.995 ± 0.0037	0.789 ± 0.2053	0.792 ± 0.2324	0.790 ± 0.2214
DeepNerve (Horng et al., 2020)	0.997 ± 0.0007	0.912 ± 0.0438	0.891 ± 0.047	0.901 ± 0.2214
U-Net + Pre-processed data +Data Augmentation + Batch Norm	**0.998 ± 0.0032**	**0.989 ± 0.0591**	**0.898 ± 0.0893**	**0.94 ± 0.0632**

It is evident from the table, the proposed method (U-Net with data pre-processing, data augmentation, and batch norm) outperformed the best deep learning approach, DeepNerve (Horng et al., 2020) which combines the features of MaskTrack and convolutional long short term memory (MaskTrack + ConvLSTM) and produced agreeable segmentation results with average accuracy, recall, precision, and F1 scores of 0.997 ± 0.0007, 0.912 ± 0.0438, 0.891 ± 0.047, and 0.901 ± 0.2214 which were fewer than proposed method.

The proposed obtained higher performance indices such as accuracy, recall, precision, F1score by 0.001, 0.077, 0.003, and 0.04 respectively compared to DeepNerve. Moreover, the standard deviation (SD) of the proposed method is comparable to the SD values of DeepNerve and some SD values are less compared to DeepnNerve. It demonstrates that the suggested technique is equally consistent and robust in median nerve segmentation and that the model is more reliable. This also implies that data pre-processing and data augmentation are important not just for cleaning data and expanding the number of datasets, but also for improving accuracy.

One of the most common challenges with the aforementioned methodology is that the algorithms underperform when subjected to noisy, inconsistent, low contrast input images and when the images are of diverse morphological variations. In addition, incorporating numerous deep feature extraction techniques into a single algorithm to perform segmentation necessitates a large amount of computing power and a longer calculation time. This bottleneck caused the major roadblock for the deep learning-based algorithm to rapidly deploy in clinical care. Thus, the proposed method which adopts the data pre-processing and data augmentation before classifying, or feature processing along with the addition of a batch normalization layer after every convolutional layer alleviates the problems and improves the accuracy of targets identification.

In Table 12, DSC results generated by the deep learning method such as Four-layer U-Net (Horng et al., 2020), Lightweight U-Net (Horng et al., 2020), Original U-Net with PCA (Principal component analysis) transformation (Kakade & Dumbali, 2018), Spatiotemporal consistency-based U-Net localization + PGVF (Probabilistic gradient vector flow method) (Hafiane et al., 2017), and DeepNerve (Horng et al., 2020) which are originally discussed in the paper (Horng et al., 2020) are compared with the results generated by the proposed method. The method obtained average DSC value of 0.6497 ± 0.19509, 0.7183 ± 0.2462, 0.68828, 0.85 ± 0.15 and 0.897 ± 0.025 respectively.

Table 12 Comparison of DSC of Proposed Method with other Deep Learning Models

Model	DSC
Four-layer U-Net (Horng et al., 2020)	0.6497 ± 0.19509
Lightweight U-Net (Horng et al., 2020)	0.7183 ± 0.2462
Original U-Net with PCA (Principal component analysis) transformation (Kakade & Dumbali, 2018)	0.68828
Spatiotemporal consistency-based U-Net localization + PGVF (Probabilistic gradient vector flow method) (Hafiane et al., 2017)	0.85 ± 0.15
U-Net (Festen et al., 2021)	0.88
DeepNerve (Horng et al., 2020)	0.897 ± 0.025
U-Net + Pre-processed data +Data Augmentation + Batch Norm	**0.899 ± 0.0990**

When performing automated segmentation of the median nerve in the carpal tunnel, U-Net (Festen et al., 2021) obtained a DSC value of 0.88. Except for DeepNerve, it outperformed all of the methods provided in Horng et al. (2020) by a significant margin. The table shows that the proposed approach outperformed DeepNerve and U-Net (Festen et al., 2021) by 0.002 and 0.019, respectively, in terms of DSC value. Furthermore, the suggested method's standard deviation (SD) is smaller than all the method-presented SD values except for DeepNerve. It demonstrates that the suggested method is more consistent and robust in median nerve segmentation. It also concludes that the proposed model is more stable and reliable. The greatest Dice similarity coefficient value indicated that the proposed method could detect and segment the median nerve in ultrasound images. Therefore, the suggested approach may be beneficial in the diagnosis of ultrasonic carpal tunnel syndrome (CTS).

It is evident from all of these compressions; that the proposed method outperformed all the presented methods with a higher performances index and lower

SD. As a result, it can be concluded that suitable data pre-processing and data augmentation, introduced before the feature extraction model produces a more accurate result in the segmentation of median nerve in ultrasound images. Furthermore, it does not demand combining many deep feature extraction approaches into a single algorithm to accomplish segmentation, which would require a lot of processing power and time. As a result, the proposed approach could be employed to segment the median nerve in an ultrasound machine to diagnose CTS.

In this chapter, the results of the proposed method adopted to perform the median nerve segmentation in ultrasound scans are discussed. Furthermore, the result of the proposed method is compared with other methods presented in the literature and found that the proposed method was more accurate and consistent in the segmentation of median nerve in ultrasound images.

CHAPTER V

CONCLUSION

Image segmentation, a subset of computer vision, is a widely adopted tool in a range of applications, including medical imaging, object identification, and self-driving. In this study, a signal processing and deep learning technique are employed to perform the segmentation of the median nerve in ultrasound images to diagnose Carpal tunnel syndrome (CTS). CTS is a kind of peripheral neuropathy, a frequently occurring disease in the wrist that affects many people.

The first part of the thesis presented a method that employed a signal processing model to segment the median nerve from ultrasound images to permit automated segmentation of structural features of the median nerve. Pre-processing is applied to reduce the associated noise to boost the sensitivity and increase the accuracy of the model in segmenting the median nerve in ultrasound images. The signal processing methods such as mathematical morphology, edge detection, and contouring employed to segment and identify the median nerve in ultrasound images produced a Jaccard similarity of 0.93. Furthermore, a remarkable 0.962 correlation of Cross-Sectional Area (CSA) between the ground truth and the segmented image is observed. However, manually modifying the kernel's weight and iterating several times in the morphological operation to segment the median nerve took more processing time. In addition, the method proved unreliable, favoring certain feature images over others. It was also observed that only 35 of the 70 ultrasound images could be segmented using this procedure. As a solution, an alternate approach utilizing a deep learning approach was implemented to make it more resilient and faster.

The second approach presented in this thesis is the segmentation of the median nerve using the U-Net (Ronneberger et al., 2015) model, which is a deep learning algorithm that addresses the issues that are faced in the signal processing method. We initially hyper-parametrized the learning rate from 10^{-3} to 10^{-6} and discovered that 10^{-6} produces the best training results when compared to other learning rates, however, it was observed that 10^{-6} takes maximum time to train the model. The U-Net model was trained in three approaches: base U-Net, U-Net with pre-processed data, and U-Net with batch normalization layer, pre-processed and data augmented data. The U-Net model trained with pre-processed data, and enhanced data with batch norm layer surpassed the two other models and achieves amazing results in median nerve segmentation. When evaluated on test datasets, an accuracy of 99.8% was achieved, which is 14.1 % higher than approach one (U-Net with original data) and 4.4% higher than approach two (U-Net with pre-processed data). The Jaccard similarity coefficient was found to be 0.8901, which is 0.039 lower than the signal processing method but 0.172 and 0.349 higher than approaches two and one. Approach three was also quite successful in finding the median nerve, with a DSC of 0.899, which was much higher than the other two approaches. This shows that when deep learning is given additional training data and the input data is cleaned, the outcomes are more accurate. The proposed method (U-Net with data pre-processing, data augmentation, and batch norm) outperformed the best deep learning approach, DeepNerve (Horng et al., 2020) which combines the features of MaskTrack and convolutional long short term memory (MaskTrack + ConvLSTM). This implies that data pre-processing and data augmentation are important not just for cleaning data and expanding the number of datasets, but also for improving accuracy. This demonstrates that this model could be used as an initial screening tool in clinical practice to expedite the identification, diagnosis, and assessment of CTS.

Limitation of Study

Some of the limitations of the study are listed below:

1. This study solely considers the segmentation of the distal median type of the human hand wrist and ignores the prox median type.
2. The datasets utilized in this study were only obtained from one hospital and evaluated by one expert.
3. Because the datasets published in the literature are not openly accessible, the model could only be evaluated on the test datasets created in-house.

Challenges

1. Limited Data size: The amount of data accessible is the most significant constraint in attaining study objectives. Ideally, this study would result in a network that is spatially invariant and capable of detecting and segmenting the median nerve in ultrasound pictures. However, there is currently insufficient data available at various spatial resolutions.
2. Limited Annotated Data: Obtaining annotated medical data is very difficult due to the confidentiality of medical data.
3. Limited Data Source: No freely available data cannot accessed the data easily.

Future Work

1. Increase the datasets to make the model more spatial invariant.
2. Incorporate more median nerve types by collecting the ultrasound images from different hospitals performed with different experts and ultrasound machines.
3. Perform the post-processing to further increase the accuracy of the model.
4. Experiment with several models and compare the results to the U-Net model.
5. Another way is to pre-train the network using large datasets such as ImageNet and then use the transfer learning approach to enhance segmentation accuracy and processing performance.

REFERENCES

REFERENCES

Ahmed, H. S., & Nordin, M. J. (2011). Improving diagnostic viewing of medical images using enhancement algorithms. *Journal of Computer Science*, *7*(12), 1831.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big data*, *8*(1), 1-74.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481-2495.

Baydar, B. (2018). *Convolutional neural network based brain MRI segmentation* Middle East Technical University].

Benhammou, Y., Achchab, B., Herrera, F., & Tabik, S. (2020). BreakHis based breast cancer automatic diagnosis using deep learning: Taxonomy, survey and insights. *Neurocomputing*, *375*, 9-24.

Bharati, S., Khan, T. Z., Podder, P., & Hung, N. Q. (2021). A comparative analysis of image denoising problem: noise models, denoising filters and applications. In *Cognitive Internet of Medical Things for Smart Healthcare* (pp. 49-66). Springer.

Bhogal, R. K., Suchit, P. D., & Naresh, C. (2021). Breast Cancer Detection Using Deep Learning. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI),

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010* (pp. 177-186). Springer.

Bracewell, R. N., & Bracewell, R. N. (1986). *The Fourier transform and its applications* (Vol. 31999). McGraw-Hill New York.

Burton, C. L., Chen, Y., Chesterton, L. S., & van der Windt, D. A. (2018). Trends in the prevalence, incidence and surgical management of carpal tunnel syndrome between 1993 and 2013: an observational analysis of UK primary care records. *BMJ open*, *8*(6), e020166.

Bychkov, D., Linder, N., Tiulpin, A., Kücükel, H., Lundin, M., Nordling, S., Sihto, H., Isola, J., Lehtimäki, T., & Kellokumpu-Lehtinen, P.-L. (2021). Deep learning identifies morphological features in breast cancer predictive of cancer ERBB2 status and trastuzumab treatment efficacy. *Scientific Reports*, *11*(1), 1-10.

Byra, M., Hentzen, E., Du, J., Andre, M., Chang, E. Y., & Shah, S. (2020). Assessing the performance of morphologic and echogenic features in median nerve ultrasound for carpal tunnel syndrome diagnosis. *Journal of Ultrasound in Medicine*, *39*(6), 1165-1174.

Chan, T. F., & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on image processing*, *10*(2), 266-277.

Chang, K.-V., Wu, W.-T., Han, D.-S., & Özçakar, L. (2017). Static and dynamic shoulder imaging to predict initial effectiveness and recurrence after ultrasound-guided subacromial corticosteroid injections. *Archives of physical medicine and rehabilitation*, *98*(10), 1984-1994.

Chang, K.-V., Wu, W.-T., Huang, K.-C., Jan, W. H., & Han, D.-S. (2018). Limb muscle quality and quantity in elderly adults with dynapenia but not sarcopenia: an ultrasound imaging study. *Experimental gerontology*, *108*, 54-61.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), 834-848.

Chen, M., Zhang, B., Topatana, W., Cao, J., Zhu, H., Juengpanich, S., Mao, Q., Yu, H., & Cai, X. (2020). Classification and mutation prediction based on histopathology H&E images in liver cancer using deep learning. *NPJ precision oncology*, *4*(1), 1-7.

Chen, Y. (2019). *Deep Learning based 3D Image Segmentation Methods and Applications*. Ohio University.

Dale, A. M., Harris-Adamson, C., Rempel, D., Gerr, F., Hegmann, K., Silverstein, B., Burt, S., Garg, A., Kapellusch, J., & Merlino, L. (2013). Prevalence and incidence of carpal tunnel syndrome in US working populations: pooled analysis of six prospective studies. *Scandinavian journal of work, environment & health*, *39*(5), 495.

Damelin, S. B., & Miller Jr, W. (2012). *The mathematics of signal processing*. Cambridge University Press.

Dansana, D., Kumar, R., Bhattacharjee, A., Hemanth, D. J., Gupta, D., Khanna, A., & Castillo, O. (2020). Early diagnosis of COVID-19-affected patients based on X-ray and computed tomography images using deep learning algorithm. *Soft Computing*, 1-9.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, *12*(7).

Duncan, I., Sullivan, P., & Lomas, F. (1999). Sonography in the diagnosis of carpal tunnel syndrome. *AJR. American journal of roentgenology*, *173*(3), 681-684.

Duncan, S. F., & Kakinoki, R. (2017). Carpal Tunnel Syndrome and Related Median Neuropathies. *Carpal Tunnel Syndrome and Related Median Neuropathies*.

Eskreis-Winkler, S., Onishi, N., Pinker, K., Reiner, J. S., Kaplan, J., Morris, E. A., & Sutton, E. J. (2021). Using Deep Learning to Improve Nonsystematic Viewing of Breast Cancer on MRI. *Journal of Breast Imaging*, *3*(2), 201-207.

Ess, A., Müller, T., Grabner, H., & Van Gool, L. (2009). Segmentation-Based Urban Traffic Scene Understanding. BMVC,

Faten A. Dawood, Z. M. A. (December-2018). The Importance of Contrast Enhancement in Medical Images Analysis and Diagnosis. *International Journal of Engineering Research & Technology (IJERT)*, *7*(12).

Ferry, S., Pritchard, T., Keenan, J., Croft, P., & Silman, A. (1998). Estimating the prevalence of delayed median nerve conduction in the general population. *British journal of rheumatology*, *37*(6), 630-635.

Festen, R. T., Schrier, V. J., & Amadio, P. C. (2021). Automated Segmentation of the Median Nerve in the Carpal Tunnel using U-Net. *Ultrasound in Medicine & Biology*, *47*(7), 1964-1969.

Filius, A., Scheltens, M., Bosch, H. G., Van Doorn, P. A., Stam, H. J., Hovius, S. E., Amadio, P. C., & Selles, R. W. (2015). Multidimensional ultrasound imaging of the wrist: Changes of shape and displacement of the median nerve and tendons in carpal tunnel syndrome. *Journal of Orthopaedic Research*, *33*(9), 1332-1340.

Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267-285). Springer.

Gamble, P., Jaroensri, R., Wang, H., Tan, F., Moran, M., Brown, T., Flament-Auvigne, I., Rakha, E. A., Toss, M., & Dabbs, D. J. (2021). Determining breast cancer biomarker status and associated morphological features using deep learning. *Communications Medicine*, *1*(1), 1-12.

Gazioglu, S., Boz, C., & Cakmak, V. A. (2011). Electrodiagnosis of carpal tunnel syndrome in patients with diabetic polyneuropathy. *Clinical Neurophysiology*, *122*(7), 1463-1469.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. 2012 IEEE conference on computer vision and pattern recognition,

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., & Cai, J. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354-377.

Gu, S., Zhang, L., Zuo, W., & Feng, X. (2014). Weighted nuclear norm minimization with application to image denoising. Proceedings of the IEEE conference on computer vision and pattern recognition,

Gupta, M., Taneja, H., & Chand, L. (2018). Performance enhancement and analysis of filters in ultrasound image denoising. *Procedia computer science*, *132*, 643-652.

Hadjerci, O., Hafiane, A., Morette, N., Novales, C., Vieyres, P., & Delbos, A. (2016). Assistive system based on nerve detection and needle navigation in ultrasound images for regional anesthesia. *Expert Systems with Applications*, *61*, 64-77.

Hafiane, A., Vieyres, P., & Delbos, A. (2014). Phase-based probabilistic active contour for nerve detection in ultrasound images for regional anesthesia. *Computers in biology and medicine*, *52*, 88-95.

Hafiane, A., Vieyres, P., & Delbos, A. (2017). Deep learning with spatiotemporal consistency for nerve segmentation in ultrasound images. *arXiv preprint arXiv:1706.05870*.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. Proceedings of the IEEE international conference on computer vision,

Hectors, S. J., Kennedy, P., Huang, K.-H., Stocker, D., Carbonell, G., Greenspan, H., Friedman, S., & Taouli, B. (2021). Fully automated prediction of liver fibrosis using deep learning analysis of gadoxetic acid–enhanced MRI. *European Radiology*, *31*(6), 3805-3814.

Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, *14*(8), 2.

Horng, M.-H., Yang, C.-W., Sun, Y.-N., & Yang, T.-H. (2020). Deepnerve: A new convolutional neural network for the localization and segmentation of the median nerve in ultrasound image sequences. *Ultrasound in Medicine & Biology*, *46*(9), 2439-2452. https://doi.org/https://doi.org/10.1016/j.ultrasmedbio.2020.03.017

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition,

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, *160*(1), 106-154.

Ibrahim, I., Khan, W., Goddard, N., & Smitham, P. (2012). Suppl 1: carpal tunnel syndrome: a review of the recent literature. *The open orthopaedics journal*, *6*, 69.

Ilesanmi, A. E., & Ilesanmi, T. O. (2021). Methods for image denoising using convolutional neural network: a review. *Complex & Intelligent Systems*, 1-20.

Im Suk, J., Walker, F. O., & Cartwright, M. S. (2013). Ultrasonography of peripheral nerves. *Current neurology and neuroscience reports*, *13*(2), 328.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning,

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition workshops,

Jia, G., Lam, H.-K., & Xu, Y. (2021). Classification of COVID-19 chest X-Ray and CT images using a type of dynamic CNN modification method. *Computers in biology and medicine*, *134*, 104425.

Kakade, A., & Dumbali, J. (2018). Identification of nerve in ultrasound images using u-net architecture. 2018 International Conference on Communication information and Computing Technology (ICCICT),

Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., & Glocker, B. (2017). Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical image analysis*, *36*, 61-78.

Karaoğlu, O., Bilge, H. Ş., & Uluer, İ. (2021). Removal of speckle noises from ultrasound images using five different deep learning networks. *Engineering Science and Technology, an International Journal*.

Kassania, S. H., Kassanib, P. H., Wesolowskic, M. J., Schneidera, K. A., & Detersa, R. (2021). Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: a machine learning based approach. *Biocybernetics and Biomedical Engineering*, *41*(3), 867-879.

Kaur, P., Singh, G., & Kaur, P. (2018). A review of denoising medical images using machine learning approaches. *Current medical imaging*, *14*(5), 675-685.

Kaymak, B., Özçakar, L., Çetin, A., Çetin, M. C., Akıncı, A., & Hasçelik, Z. (2008). A comparison of the benefits of sonography and electrophysiologic measurements as predictors of symptom severity and functional status in patients with carpal tunnel syndrome. *Archives of physical medicine and rehabilitation*, *89*(4), 743-748.

Kazantzidou, V., Lytras, D., Kottaras, A., Iakovidis, P., Kottaras, I., & Chatziprodromidou, I. P. (2021). The efficacy of manual techniques in the treatment of carpal tunnel syndrome symptoms: A narrative review. *Int. J Orthop Sci*, *7*(2), 423-427.

Khan, A. R., Khan, S., Harouni, M., Abbasi, R., Iqbal, S., & Mehmood, Z. (2021). Brain tumor segmentation using K-means clustering and deep learning with synthetic data augmentation for classification. *Microscopy Research and Technique*.

Kiani, A., Uyumazturk, B., Rajpurkar, P., Wang, A., Gao, R., Jones, E., Yu, Y., Langlotz, C. P., Ball, R. L., & Montine, T. J. (2020). Impact of a deep learning assistant on the histopathologic classification of liver cancer. *NPJ digital medicine*, *3*(1), 1-8.

Kiefer, J., & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The annals of mathematical statistics*, 462-466.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, 1097-1105.

Kukačka, J., Golkov, V., & Cremers, D. (2017). Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature (2015). *May; 521 (7553): 436 10.1038/nature14539*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Li, C., Xu, C., Gui, C., & Fox, M. D. (2010). Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on image processing*, *19*(12), 3243-3254.

Liu, X., Song, L., Liu, S., & Zhang, Y. (2021). A review of deep-learning-based medical image segmentation methods. *Sustainability*, *13*(3), 1224.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition,

Luc, P., Couprie, C., Chintala, S., & Verbeek, J. (2016). Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*.

Mark E. Pruzansky, J. S. P. *HandSport Surgery Institute*. Retrieved 11 December from https://handsurgeonsnyc.com/patient-education/wrist-anatomy/

Masoudi, S., Harmon, S. A., Mehralivand, S., Walker, S. M., Raviprakash, H., Bagci, U., Choyke, P. L., & Turkbey, B. (2021). Quick guide on radiology image pre-processing for deep learning applications in prostate cancer research. *Journal of Medical Imaging*, *8*(1), 010901.

Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., & Wiest, R. (2014). The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE transactions on medical imaging*, *34*(10), 1993-2024.

Mhoon, J. T., Juel, V. C., & Hobson-Webb, L. D. (2012). Median nerve ultrasound as a screening tool in carpal tunnel syndrome: correlation of cross-sectional area measures with electrodiagnostic abnormality. *Muscle & nerve*, *46*(6), 861-870.

Michailovich, O. V., & Tannenbaum, A. (2006). Despeckling of medical ultrasound images. *ieee transactions on ultrasonics, ferroelectrics, and frequency control*, *53*(1), 64-78.

Mohammad-Rahimi, H., Nadimi, M., Ghalyanchi-Langeroudi, A., Taheri, M., & Ghafouri-Fard, S. (2021). Application of machine learning in diagnosis of COVID-19 through X-ray and CT images: a scoping review. *Frontiers in cardiovascular medicine*, *8*, 185.

Mohammadi, A., Afshar, A., Etemadi, A., Masoudi, S., & Baghizadeh, A. (2010). Diagnostic value of cross-sectional area of median nerve in grading severity of carpal tunnel syndrome.

Möller, I., Miguel, M., Bong, D. A., Zaottini, F., & Martinoli, C. (2018). The peripheral nerves: update on ultrasound and magnetic resonance imaging. *Clin Exp Rheumatol*, *36*(Suppl 114), 145-158.

Moran, L., Perez, M., Esteban, A., Bellon, J., Arranz, B., & del Cerro, M. (2009). Sonographic measurement of cross-sectional area of the median nerve in the diagnosis of carpal tunnel syndrome: correlation with nerve conduction studies. *Journal of clinical ultrasound*, *37*(3), 125-131.

Nadeem, M., Hussain, A., & Munir, A. (2019). Fuzzy logic based computational model for speckle noise removal in ultrasound images. *Multimedia Tools and Applications*, *78*(13), 18531-18548.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of big data*, *2*(1), 1-21.

Newington, L., Harris, E. C., & Walker-Bone, K. (2015). Carpal tunnel syndrome and work. *Best practice & research Clinical rheumatology*, *29*(3), 440-453.

Padua, L., Coraci, D., Erra, C., Pazzaglia, C., Paolasso, I., Loreti, C., Caliandro, P., & Hobson-Webb, L. D. (2016). Carpal tunnel syndrome: clinical features, diagnosis, and management. *The Lancet Neurology*, *15*(12), 1273-1284.

Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., & Sorkine-Hornung, A. (2017). Learning video object segmentation from static images. Proceedings of the IEEE conference on computer vision and pattern recognition,

Pereira, S., Pinto, A., Alves, V., & Silva, C. A. (2016). Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE transactions on medical imaging*, *35*(5), 1240-1251.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural networks*, *12*(1), 145-151.

Ranjbarzadeh, R., Kasgari, A. B., Ghoushchi, S. J., Anari, S., Naseri, M., & Bendechache, M. (2021). Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images. *Scientific Reports, 11*(1), 1-17.

Reed, P. (2005). Sample topic: Carpal tunnel syndrome. *The medical disability advisor: Workplace guidelines for disability duration: Reed Group*.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400-407.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention,

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A general framework for parallel distributed processing. *Parallel distributed processing: Explorations in the microstructure of cognition*, *1*(45-76), 26.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

Saber, A., Sakr, M., Abo-Seida, O. M., Keshk, A., & Chen, H. (2021). A Novel Deep-Learning Model for Automatic Detection and Classification of Breast Cancer Using the Transfer-Learning Technique. *IEEE Access*, *9*, 71194-71209.

Seror, P. (2008). Sonography and electrodiagnosis in carpal tunnel syndrome diagnosis, an analysis of the literature. *European journal of radiology*, *67*(1), 146-152.

Shen, D., Wu, G., & Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual review of biomedical engineering*, *19*, 221-248.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 1-48.

Smith, S. M., & Brady, J. M. (1997). SUSAN—a new approach to low level image processing. *International journal of computer vision*, *23*(1), 45-78.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

Stevens, J. C. (1997). AAEM minimonograph# 26: the electrodiagnosis of carpal tunnel syndrome. *Muscle & Nerve: Official Journal of the American Association of Electrodiagnostic Medicine*, *20*(12), 1477-1486.

Tai, T.-W., Wu, C.-Y., Su, F.-C., Chern, T.-C., & Jou, I.-M. (2012). Ultrasonography for diagnosing carpal tunnel syndrome: a meta-analysis of diagnostic test accuracy. *Ultrasound in Medicine & Biology*, *38*(7), 1121-1128.

Tian, C., Xu, Y., Fei, L., & Yan, K. (2018). Deep learning for image denoising: a survey. International Conference on Genetic and Evolutionary Computing,

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, *4*(2), 26-31.

Tomasi, C., & Manduchi, R. (1998). Bilateral filtering for gray and color images. Sixth international conference on computer vision (IEEE Cat. No. 98CH36271),

Topp, K. S., & Boyd, B. S. (2006). Structure and biomechanics of peripheral nerves: nerve responses to physical stresses and implications for physical therapist practice. *Physical therapy*, *86*(1), 92-109.

Torres-Costoso, A., Martínez-Vizcaíno, V., Álvarez-Bueno, C., Ferri-Morales, A., & Cavero-Redondo, I. (2018). Accuracy of Ultrasonography for the Diagnosis of Carpal Tunnel Syndrome: A Systematic Review and Meta-Analysis. *Arch Phys Med Rehabil*, *99*(4), 758-765.e710. https://doi.org/10.1016/j.apmr.2017.08.489

Torres-Costoso, A., Martínez-Vizcaíno, V., Álvarez-Bueno, C., Ferri-Morales, A., & Cavero-Redondo, I. (2018). Accuracy of ultrasonography for the diagnosis of carpal tunnel syndrome: a systematic review and meta-analysis. *Archives of physical medicine and rehabilitation*, *99*(4), 758-765. e710.

Wang, Y.-W., Chen, C.-J., Huang, S.-F., & Horng, Y.-S. (2015). Segmentation of Median Nerve by Greedy Active Contour Detection Framework on Strain Ultrasound Images. *J. Inf. Hiding Multim. Signal Process.*, *6*(2), 371-378.

Wei, D., Zhou, B., Torralba, A., & Freeman, W. T. (2017). mneuron: A matlab plugin to visualize neurons from deep models. *Massachusetts Institute of Technology*.

Yang, T.-H., Yang, C.-W., Sun, Y.-N., & Horng, M.-H. (2021). A Fully-Automatic Segmentation of the Carpal Tunnel from Magnetic Resonance Images Based on the Convolutional Neural Network-Based Approach. *Journal of Medical and Biological Engineering*, 1-16.

Yaroslavsky, L. P. (1985). Correcting Imaging Systems. In *Digital Picture Processing* (pp. 186-210). Springer.