RAM ROBOTICS AND MECHATRONICS

Automatic Segmentation and 3D Reconstruction of Liver and Tumor

G. (Girindra) Wardhana

MSc Report

Committee:
Dr. Ir. M. Abayazid
H. Naghibi Beidokhti, MSc
Prof.dr.ir. C.H. Slump
Dr. C. Brune
Dr. B. Sirmaçek

August 2018

029RAM2018
Robotics and Mechatronics
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE.

MIRA CTIT
BIOMEDICAL TECHNOLOGY AND TECHNICAL MEDICINE

Summary

In liver-related symptoms, image segmentation is an important step to initiate biomechanical computational modeling, robotic surgeries and help clinicians in visualizing the anatomy of the patient's body as a crucial basis for surgery planning. Various methods of image segmentation have been developed especially for liver treatments. Previously, manual segmentation techniques are applied to obtain a precise liver segmentation. However, this method is time-consuming, laborious, and very subjective where the result varies depending on the operator. Meanwhile, employing an automatic method is also very challenging. The effects of different contrast agent and different acquisition technique contribute to a large variety of liver and tumor intensity.

A deep convolutional neural network is proposed to perform automatic liver and tumor segmentation from CT images. To train and test the network, the dataset that provided by Liver Tumor Segmentation (LiTS) Challenge is employed in this project. The network structure utilizes the encoder and decoder structure from the SegNet with some modification to improve its performance. Several tests have been conducted to examine the network segmentation performance, including the preparation on the dataset, the variation of network configuration and the evaluation of segmentation result using manual segmentation.

The test result reveals that different preparation techniques affect the segmentation accuracy. At the same time, utilizing class balance in the network is very crucial, where the network without class balance ignores the tumor and only recognize the background and liver area from the image. The evaluation of the network has been done by participating in the LiTS Challenge and conducting manual segmentation experiment. In the LiTS challenge, the proposed network outperforms other states of the art methods for detecting the tumor and subsequently get the first rank on the online LiTS leaderboard in Recall category. Meanwhile, the proposed network also shows an impressive performance in the manual segmentation experiment. The segmentation from the network exceeds the manual segmentation result in term of segmentation accuracy and processing time. Moreover, more than 90% participants are very satisfied with the liver and tumor segmentation that obtained from automatic method.

Further work of this project should be focused on improving the segmentation performance. Implementing liver detection before doing the segmentation can reduce the processing time significantly. Furthermore, implementing post-processing such as tumor detection is necessary to reduce the false positive in the segmentation result. This topic could also be expanded to other topics, such as tumor classification and educational application that can be used to train the technical medicine student for segmenting the liver and tumor.

Acknowledgements

I would like to give my biggest thanks to my parents, my sister and my brother that always support me spiritually and be there for me throughout my life.

I would like to express my gratitude to Dr. Ir. M. Abayazid (Momen) for the opportunities that were given to me to conduct research at the RaM department. He has been there providing his encouragement, invaluable guidance, and constructive advice during the process of researching and writing this thesis. My grateful thanks are also for H. Naghibi Beidokhti MSc (Hamid), Dr. B. Sirmacek (Beril) and Dr. M. Shurrab (Mohammed) for the patience, guidance, and feedback throughout my thesis work. I would also like to thank Prof. Dr. Ir. C.H. Slump (Kees) and Dr. C. Brune (Christoph) for becoming the committee member of mine.

I have great pleasure in acknowledging my gratitude to all member in RaM department for having interesting talks, working together before the deadline and providing a productive and friendly environment during the research. My thanks also for Geert Jan from CTIT computing Lab for assistance and support to set up the cluster computer for processing the thesis data. I would also like to thank all of the volunteers that participated in the experiment. Thanks to them I was able to collect the data and complete my research in time.

Furthermore, a special acknowledgment to *Lembaga Pengelolaan Dana Pendidikan* (LPDP) Scholarship from Ministry of Finance of Indonesia for giving me the opportunity and providing the funding for study in the University of Twente.

It would be inappropriate If I did not mention Nuzul Hesty and all of my friends, especially my housemate in “DBL 19”, Mas Zul, Mbak Nden, Imran, Teh Cui, Kak Manda, and Eva, for their unfailing support and continuous encouragement that help me reach this stage in my life. They entertain me when I have difficult moments and ensure me always to have a good time. This accomplishment would not have been possible without these people around.

Girindra Wardhana
Enschede, Thursday 30 August 2018

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Context

Liver cancer was among the leading cause of cancer death globally where 854 thousand new case and 810 thousand deaths [1]. Prevention and treatment of liver disease are urgent and become a hot topic for research. One of the prevention could be like an early diagnosis. The sooner the lesion can be detected, the higher the chance the patient can survive from liver disease. In recent years, clinicians utilize medical imaging to provide a clear picture of the possible lesion inside the patient body. Computed Tomography (CT) is the common imaging technology, not only for detection but also for diagnosis and follow up the lesion. Medical imaging is also necessary for liver diagnosis and liver transplantation. From the image, information such as size, shape and the exact location are required which could be obtained by doing a segmentation.

Image segmentation is a process to group object that has a similar attribute from the background to simplify image representation. In medical term, image segmentation help in separating the object from other organ or tissue and make it easier to analyze and give diagnoses [2]. Figure 1.1 shows liver segmentation strategies [3], which are manual, contour optimization, semi-automated and fully automated.

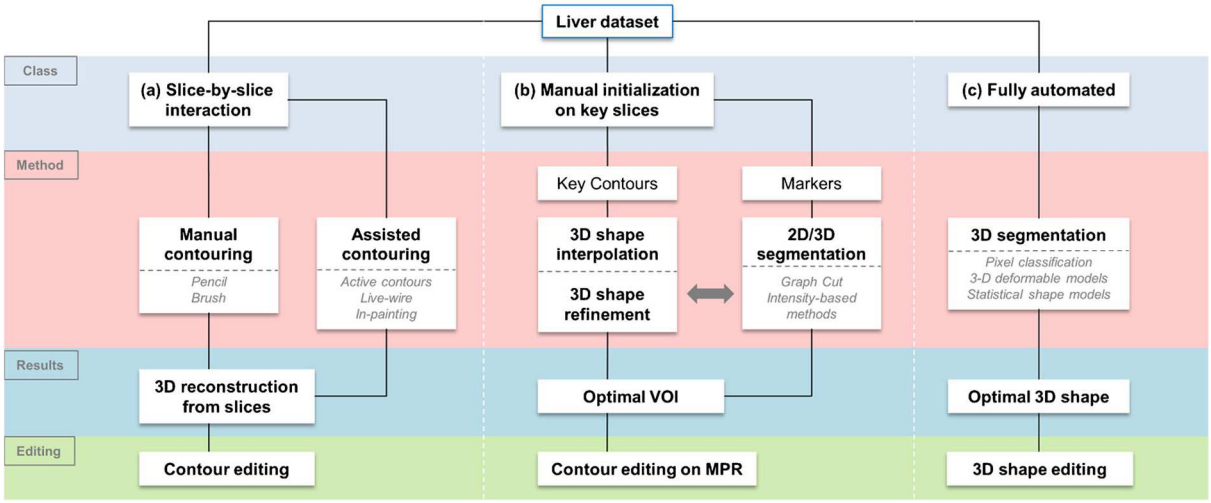


Figure 1.1 Liver Segmentation Strategy [3]

Manual liver segmentation depends heavily with the clinician to perform segmentation. In this method, the liver boundary was selected by contouring the pixel along the edge or painting the liver region in each slice. After all slices have been processed, the data is post-processed to shape the liver volume. From this process, precise liver shape and volume can be obtained. However, due to the increasing technology of X-ray imaging, higher data can be produced with providing higher resolution [4]. Even though this method is still used regularly by the radiologist, the method requires a long processing time, laborious and subjective.

The need for accurate and efficient tumor delineation leads to the development of contour optimization methods. In contour optimization, the clinician is helped by assisted contouring or assisted painting where the user needs to draw a rough border of the liver, and later the algorithm will help to find the border. However, these methods are still prone to subjectivity due to the need to receive input from the user. Development of semiautomatic and automatic segmentation method is one of the ways to address the issues that occurs in the previous methods. The more automatic step introduced in the segmentation process, the more time could be saved, and the more precise the segmentation is due to reducing human

subjectivity[5]. Some popular techniques are worked based on region growing [6] [7], level set [8] [9] and a statistical shape model [10] [11] [12].

1.2. Problem Statement

Despite the fact that automatic method can provide fast segmentation and prone to subjectivity error, the performance remains relatively poor. In the liver case, segmenting liver from contrast-enhanced CT volume is a very challenging task because of different contrast agent and different acquisition technique. These contribute to a large variety of liver intensity. Moreover, low contrast between liver and neighbor organs make the liver boundaries fuzzy and hard to detect. Automatic methods such as threshold and region growing which rely on the intensity information are prone to detect other organs as the liver area because of no shape control. On the other hand, utilizing shape information to separate the liver from other organs is a challenging task. The highly varied liver shapes and sizes among different individuals are among the factors that need to be considered.

Compared to the liver case, tumor segmentation is even more challenging. Even though the liver shape has various size and shape, but its location can be predicted on the upper right side of the abdomen. Meanwhile, liver tumors have, not only various size and shape but also the location and numbers varied within one patient. This makes segmenting method that employs prior knowledge of organs locations will have poor results. Furthermore, some tumors do not have clear boundaries which limit the performance of the automatic segmentation method that using intensity information.

Both the issue on liver and tumor are interesting to be solved. Many methods have been developed recently to improve the robustness and the performance of the automatic method for liver and tumor segmentation [13] [14] [15].

1.3. Related Work

In a recent competition that organized in conjunction with MICCAI 2017 and ISBI 2017, participants are challenged to develop an automatic segmentation algorithm for liver and tumor in contrast-enhanced abdominal CT scan. Most of the participant employed deep neural networks to obtain liver and tumor segmentation. Chlebus [16] presented Convolutional Neural Network (CNN) based on 2D U-Net network. Two models were proposed to segment liver and segment tumor. Random Forest classifier was employed to reduce false positives among tumor candidate. Lei Bi [17] proposed cascaded ResNet to overcome layer limitation to obtain more discriminative features. As additional, multiscale approach was implemented to derive more information in stage and pixel resolution for discovering the intrinsic correlation between small scale and large scale object

Other researches show that including 3D information can improve the segmentation result. Han [18], who was a winner of the first round of the competition, proposed deep convolutional neural network architecture combining long-range connection of U-Net and short-range residual connection of ResNet. Two models were developed, where the first model was used to segment the liver region as an input for the second model to detect and segment tumor. Both models worked in 2.5D where five adjacent slices as input were used during training and produce the segmentation of the center slice. The purpose was typically to keep computation efficiently using 2D slices and to provide 3D context information to the network. A different method that proposed by Li [19] mentions that employing stacked slice is not enough to cover the spatial information of the third dimension. Hybrid dense units are proposed which consist of 2D dense units to extract intra slice features and 3D dense units to exploit interslice context from a volumetric image. Later, intra slice and interslice information are combined and optimized through a hybrid feature fusion (HFF) layer to achieve a high-performance network for liver and tumor segmentation.

1.4. Research Approach

The primary goal of this project is to develop an automatic method for detecting and segmenting the liver and its lesion from CT Scan Image. A method based on Deep Convolutional Neural Network (DCNN) is developed to perform a pixel-wise classification, where the neural network will predict the pixel label from the image to obtain the liver and lesion region. Ultimately, this result will be used to reconstruct the liver and the tumor 3D shape model to provide volume, size and location information which can be utilized by a clinician to give clinical diagnoses and surgery planning.

The primary goal of this project is split into three parts which formulated as research questions as follows:

1. How is the preparation of the dataset contributed to the segmentation accuracy?
    - Arrangement of image slices (single slice or multiple slices)
    - Adjustment of image contrasts

2. What is the best network architecture and setup that gives the best performance to the liver and tumor segmentation result?
    - Influence of class balancing in the data set
    - Effect of different network structures

3. How well do the Network perform compared to other segmentation results?
    - Participation in the Liver Tumor Segmentation Challenge leaderboard
    - Comparison with the manual segmentation

1.5. Report Outline

In Chapter 2, information regarding the neural network and its component are explained, as well as several network structures for image segmentation. The network configuration in this project is presented in chapter 3. Chapter 4 presents the explanation of the experiment setup, while chapter 5 presents the experiment result and discussion for answering the research questions. In the end, chapter 6 cover the conclusion and the suggestion for future work

2. Background

In medical practice, image segmentation has an important role in helping clinicians in visualizing the anatomy of the patient's body as a crucial basis for surgery planning. Various methods have been developed for medical image segmentation, especially for liver treatment. Recently, deep learning method based on Convolutional Neural Network (CNN) has shown promising results in segmenting the liver and tumor area, where the method achieves high accuracy result without too dependent on the hand-crafted features from the user. In this chapter, background information related to neural network and convolution neural network is explained in section 2.1 and 2.2, while examples of deep learning application in several medical cases are discussed in section 2.3.

2.1. Introduction to Artificial Neural Networks

A neural network is a brain-inspired system that tries to imitate the way of people learn. Like people who learn from the example, the neural network enables the machine to learn from the data given. This section gives a brief explanation about the background of a neural network and its structure.

2.1.1. Neuron Model

A neural network is a type of machine learning method that tries to employ the mechanism of a brain in its method. While the brain is based on the connection of neuron as the computational unit, the neural network is based on the collection of nodes that correspond to the neuron in the brain. An illustration of a biological neuron can be seen in figure 2.1 [20].

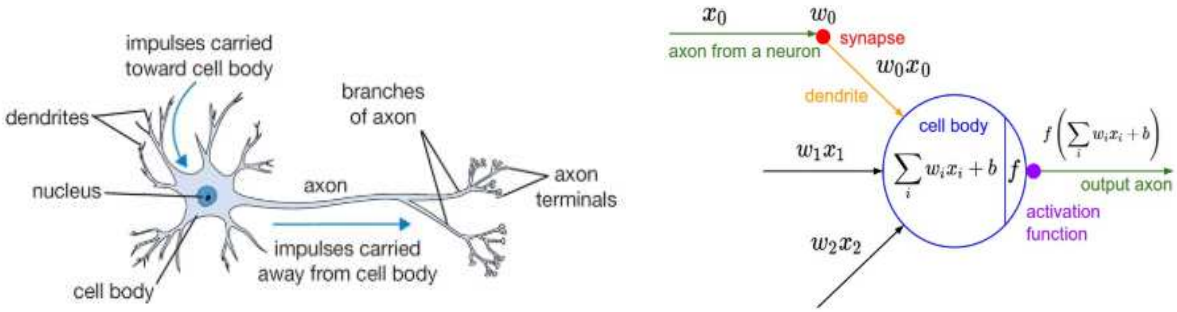


Figure 2.1 Illustration of (left) Biological Neuron and (right) Mathematical Model of Neuron [20]

From the biological point of view, a neuron consists of three main components, dendrites, cell body, and axon. The signal that comes to the neuron is received by the dendrite and gathered in the cell body. Then, the signal output from the neuron will be transmitted using the axon that connected via a synapse to the dendrites of other neurons. Synapses are characterized by the strength or weight that control the influences of one neuron to another. When a signal transmitted from the axon, the signal will be multiplied by the weight of the synapse in the dendrite. All signals that received by neuron's dendrites will be collected to the cell body and get summed. Only if the total sum can reach a certain value, then a neuron can send output through its axon. This biological mechanism, especially the weighting and the sending output behavior, is being imitated by the computer artificial neural network in its method.

From the computer machine point of view, the information of the neural network is stored in the term of weight and bias. Like a neuron in the brain, the input signal in the neural network is also multiplied by the weight and collected at the node. This weighted signal then gets summed and calculated by using the equation 2.1.

$$v = \sum w_i x_i + b \tag{2.1}$$

where the weight sum *v*, the weight *w*, the input signal *x,* the node *i,* and the bias *b*.

In Figure 2.1, the mathematical model of how the signal is processed in the neuron is also given. As mention earlier, there is a certain threshold that needs to be fulfilled by the final sum in order to send output. This behavior can be modeled using a non-linear function which called an activation function. There are many types of activation functions that available in the neural network. More detail explanation about activation functions will be given in the section 2.2.1.

2.1.2. Node Structure in Neural Network

Commonly, a neural network is made up of three main layers: the input layer, the output layer, and the hidden layer. The input layer is the layer where the input signal enters the net. Since this layer is the first layer, no weight sum is calculated. For the output of the network, the outcome will be transmitted by the output layer. The layer between the input layer and the output layer are called the hidden layer. This name is given because this layer is not accessible from the outside of the network.

Many developments have been done in the structure of a neural network, from a simple network structure into a complex structure. If the network is composed of an input and an output layer, it is called a single layer neural network. Meanwhile, if there are one or more hidden layers included, it becomes a multi-layer neural network. In a multi-layer structure, it can be differentiated into more detail according to the number of the hidden layer. If it consists of one hidden layer, then it can be classified as a Shallow Network. However, if it has two or more hidden layer, it becomes a Deep Neural Network. Figure 2.2 shows the example of these network structures [21].

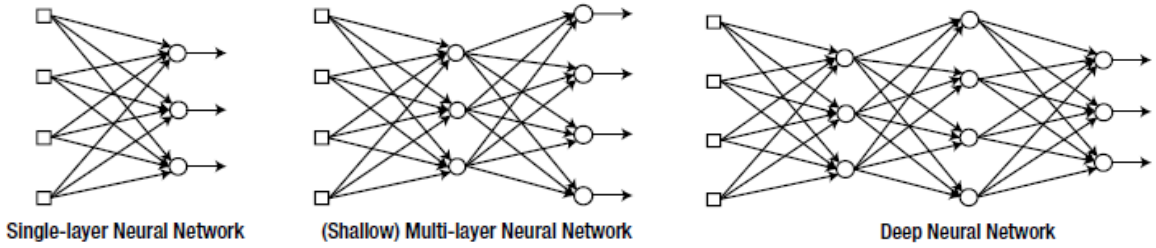


Figure 2.2 Neural Network Structures [21]

2.1.3. Challenges in the Neural Network Development

Even though the neural network can achieve a better performance compare to the other types of machine learning. There are still some challenges that make the implementation of the neural network not easy. The need for high computation power is one of the challenges. The more complex the structure, the more weights of nodes need to be computed. Therefore, more processing and memory resource are required to support this. Moreover, it is still hard to prove the optimum network structure mathematically because of the complexity of the network. Most of the determination is made based on trial and error which make to conclude the performance of one method to other methods need the empirical comparison [22]. Furthermore, the computation in the network involves the numerical value for the problem. The difference method for translating the problem into a numerical value will give a different result from the network [23]. Therefore, it is necessary to know and implement the same data processing to obtain the same result from a network.

2.2. Convolutional Neural Networks

Recently, Convolutional Neural Network (CNN) has shown an astounding performance in addressing the problem in the computer vision area. It was started from Krizhevsky achievement in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 by introducing AlexNet [24]. After that, many CNN structures have been developed, such as VGG Net [25], GoogleNet [26], and ResNet[27]. VGG Net introduced the implementation of stacking convolution layer with a small receptive field to ease the training process. GoogleNet came up with the inception module, proposing a new method of stacking the convolution layer and ResNet employ a very deep structure (152 layers) and suggest the skip connection to overcome the vanishing gradient that occurs in the deeper layer.

2.2.1. Layer in Convolutional Neural Networks

The main difference between a convolutional neural network and other methods in addressing the computer vision problem is its ability to extract the features from the image automatically and included as a part of the training process. In this section, several types of layers that commonly used in the CNN will be explained [21].

INPUT LAYER

The specification of the input image is given in the input layer. It is described in term of a 3D matrix that contains the image size information, such as width, height and the number of channels. The number of channel value is varied depending on the image type. In the grayscale image, the channel has a value of 1. While in the RGB image, its value is 3, where it represents the red channel, green channel and Blue channel. Furthermore, a greater number of channels can be found in the multispectral images that have multiple channel color.

CONVOLUTION LAYER

Convolution layer is a type of learnable layer where it has a weight that obtained during the training process. However, instead of normal weight, the weight in the convolution layer is represented as a digital filter which will be used to perform the convolution operation. Convolution operation is applied to the input image to obtain the new image which called a feature map. Feature map has a different result depending on the filter that applied to the image. From this map, the unique feature from the image can be obtained.

Several parameters need to be set up in the convolution layer. Those are the number of filters, filter size, stride, and padding. The number of filters determines the number of image features that will be extracted. The more the filter number, the more features can be obtained and the better the performance of the network to recognize the pattern in the images.

Stride represent how large the filter windows slide over the input matrix. The larger the stride, the smaller the feature maps are produced. Additionally, zero padding can be added to give zeros around the border. By doing this, the filter can slide around the bordering element. Therefore, the size of the feature maps can be controlled. The final size of the feature maps is calculated using equation 2.2.

$$W' = \frac{W - F + 2P}{S} + 1 \tag{2.2}$$

Where *W'* indicate the feature maps size, *W* input image size, *F* filter width, *P* padding size and *S* stride size.

ACTIVATION LAYER

The activation function is a nonlinear mathematical function that determines the node to produce an output or not. The type of the activation function is set in the activation layer. At the beginning of the neural network development, Sigmoid and Hyperbolic Tangent are the most popular function that are used. The behavior of those function can be seen in figure 2.3 [28].

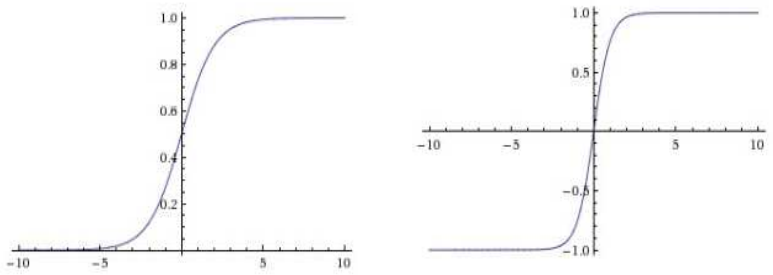


Figure 2.3 Activation Function (left) Sigmoid and (right) Hyperbolic Tangent [28]

Sigmoid function processes the input and gives result in the range of 0 and 1. The output is calculated using the equation 2.3 where a large negative number will have a value of 0 and a large positive number will become 1. Since the value is bound to a range of 0 and 1, Sigmoid has an advantage over the linear function where its output can reach infinite value. However, when the Sigmoid output saturates at either 0 or 1, the gradient becomes very small, which rising the vanishing gradient problem. In this situation, the network will become very slow, and the learning process will be hard to continue. Another drawback of this function is the output that is not zero centered. It causes the zig-zag movement of the gradient while updating the weight.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

To overcome the last problem in the Sigmoid function, the hyperbolic tangent function is proposed as the activation function which formulated in the equation 2.4. Instead of 0 and 1, the hyperbolic tangent has a result from -1 to 1. However, the problem related to the gradient vanished is still existed in this activation function.

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.4}$$

The effort to improve the performance of the activation function is continued. In recent years, Rectified Linear Unit (ReLU) function shows impressive performance. It has been used in the AlexNet structure and found that it can accelerate the convergence of the gradient during training compare to the Sigmoid or Tanh function by a factor of 6 [24]. In this function, the output is thresholded to zero, which can be formulated using equation 2.5. Not only simple to implement, but this function is also keeping the computation cost low, which is important when the network structure becomes complex.

$$f(x) = \max(0, x) \tag{2.5}$$

Despite its popularity, there is a serious issue that can occur in the ReLU function, where it can die and stop updating the neuron weight in some sort of condition. To fix this issue, the LeakyReLU function is developed. Instead of giving a constant zero to the negative input, the input will be multiplied with a small α that equal to 0.01 or 0.001. The behavior of ReLu and LeakyReLU is shown in figure 2.4.

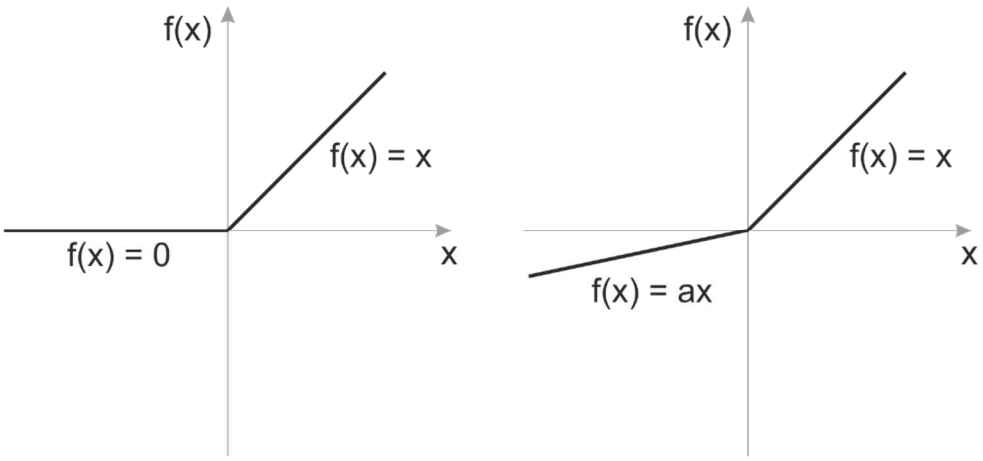


Figure 2.4 Activation Function (left) ReLU and (right) Leaky ReLU

The final suggestion to choose the proper activation function is applied the ReLU function in the network as stated in [28]. If during the training, most of the unit in the network seems died, the activation function should be change into LeakyReLU. It is not recommended to choose the Sigmoid and Tanh. Not only outdated, but also the result is worse compare to other activation functions.

POOLING LAYER

Pooling layer or known as downsampling layer is the layer that is used to reduce the spatial size of the feature map. This layer is important since by reducing the spatial size, the number of features that can be extracted from the image can be increased while keeping the computation manageable. Besides, reducing the feature dimension will also reduce the parameter number which decreases the chance of overfitting in the network. Furthermore, since the output of pooling layer is taken from the value of the local neighborhood, the pooling layer make the output more robust due to transformation in the input image. The parameters that need to be set in this layer are quite similar with the parameters in the convolution layer, consisting of window size, stride, and padding.

There are three types of pooling layer, which are Max Pooling, Average Pooling, and Sum Pooling. In the max pooling, a small window is taken and slide over the feature maps to take the largest element inside the windows filter. The movement of the windows is adjusted by the stride value in the setting. Meanwhile, the average values are used in the average pooling and the sum of the value in the sum pooling.

2.2.2. Learning Rate

The output of the network is depended on the weight value of the nodes. The parameter that controls the adjustment of the weight with respect to the loss gradient is called the learning rate. The lower the learning rate, the smaller the update made into the weight value, which is good to obtain more precise value. However, it can cause a slow convergence to the network. While keeping a higher learning rate can make the updating the weight fast, but the risk to not achieve the convergence is also high. With the intention to improve the performance of the learning rate, some functions have been developed to give a schedule for decreasing its value along to the iteration number. It means that in the beginning, the learning rate has a larger value and cause a bigger change to the parameter. The later the iteration, the learning rate becomes smaller and can perform a fine tuning to the network weight. The examples of the solver function are Stochastic Gradient Descent with Momentum (SGDM), RMSProp and Adam Optimizer.

2.2.3. Hyperparameter Tuning

During designing the network, there are many choices that need to be adjusted regarding the network architecture, such as convolutional kernel size, non-linear activation function, number of layers, learning rate, optimizer, and batch size. All these parameters are known as hyperparameter that depend highly on the size and type of the training dataset. As an example, the setting of batch size can determine the time required for training the network. The larger the batch size, the faster the network training can be finished. However, it means the network need a larger memory space as the size of batch size growing. It is similar to the determination of the learning rate, the higher the learning rate, the faster the network can be convergence, but it also harder to get the optimum result. All these settings need to be set properly to get the desired outcome. However, no specific rule can be followed while setting the hyperparameter. It seems that the parameter setting is based on the experience rather than theoretical.

2.3. Deep Learning in Medical Imaging

It is true that deep learning shows a potential result and demonstrated performance that outperforms other methods based on handcrafted features. Since the development of neural network become more advanced and significant, some object recognition tasks are developed to be automatic and capable of exceeding the human accuracy [29]. Application of deep learning in medical imaging are predicted to replace the human position for making a diagnosis, predicting disease, prescribing medicine and guiding in treatment [30]. Some applications in medical image are in skin cancer, diabetic retinopathy, histological analysis and gastrointestinal problem.

Skin cancer is the uncontrolled growth of skin cell, mostly appears in the area of the body that receive high sun exposure, such as the head, face, neck, lower arm and lower leg [31]. It is a challenging task to perform an automatic diagnose due to variability in the appearance of the skin lesion. Esteva et al. [32] trained a single DCNN to classify skin cancer, achieving performance comparable to dermatologists and showing a potential development to implement the neural network to a mobile device that provides low-cost diagnostic care. On another case, Murphree and Ngufor [33] implemented transfer learning to develop a deep neural network for Melanoma Detection in Skin lesion classification. They used Inception v3 network that initially pre-trained using ImageNet case for feature extractor.

Diabetic Retinopathy (DR) is an eye disease that occurs to the retina due to diabetes mellitus and can lead to the eye blindness. It is difficult and time-consuming to detect DR due to unavailability of equipment and expertise. Several fundus databases are available online to encourage research in this area. Development of deep learning model to perform automatic detection of DR shown optimized and better accuracy. Kathirvel [34] developed DCNN using dropout layer for classification of fundus on the popular databases such as STARE, DRIVE, and Kaggle fundus and reported to have accuracy to 94-96%. Haloi [35] implemented five layers CNN with drop-out mechanism for detection a novel microaneurysm(MA) of early-stage DR on Retinopathy Online Challenge and Massidor datasets and claimed to get sensitivity, specificity, and accuracy up to 97%, 96% and 96% respectively. Yang [36] propose automatic diabetic retinopathy analysis based on two stages of deep convolution Neural network using Kaggle competition database. The first function is to identify the location and type of lesion and the second function to give severity grade of DR, normal, mild, moderate or severe.

Histological analysis is the study of cell, group of cell and tissue using a microscope [37]. When the disease has infected to cellular and tissue level, it is possible to detect the characteristic and features using microscopic image and stain. Quinn [38] evaluated the performance of DCNN to diagnosis three different microscopy tasks, malaria in thick blood smears, tuberculosis in sputum samples and intestinal parasite eggs in stool samples. In addition to malaria case, Dong [39] examined automatic identification of malaria-infected cell. The dataset of malaria-infected red blood cells and non-infected cell are created using slide images of thin blood stains and labeled by four pathologists. The study is done by comparing three convolutional network structures, LeNet, AlexNet, and GoogleNet, to do the classification task on the dataset.

Gastrointestinal disease refers to the disease that occurs to the organs involved in food digestion, nutrient absorption, and waste product excretion. Employing image processing in diagnosing and analyzing the disease can help the doctors to make the decision faster for treatment efficiently and accurately. Jia [40] employed DCNN for detecting bleeding in gastrointestinal using 10.000 wireless capsules endoscopy images. This method achieves F1 score up to 0.9955. Wimmer [41] implement CNN for computer-assisted diagnosis of celiac disease based on endoscopic images of the duodenum. The method result is compared with other popular methods such as Local Binary Pattern, Local Ternary pattern and Improved Fisher Vector, showing the classification rated achieves a better result up to 97% than other methods.

The result that has been achieved by deep learning encourage big research organization to develop and apply deep learning in the medical area. Deep learning is not only applied for diagnosis but also can be implemented to help surgery planning and to guard the biopsy robot during the surgery process. Moreover, it can be employed as an education tool for the new doctor to help them to perform classification, detection, and segmentation from the medical image.

3. Network Configuration

This chapter explains the preparation needed for developing the automatic segmentation method. The preparation includes: the data set in section 3.1, the network architecture in section 3.2 and the evaluation step in section 3.3.

3.1. Data Set

The dataset in this project are taken from Liver and Tumor Segmentation (LiTS) challenge that organized in conjunction with Medical Image Computing and Computer Assisted Intervention (MICCAI) 2017 and IEEE International Symposium on Biomedical Imaging (ISBI) 2017. The datasets contain contrast-enhanced abdominal CT scans from different clinical sites around the world [42]. There are two parts of the dataset given, training data and test data. The training data, which consist of data and segmentation, contains 130 CT scans, while the test data has 70 CT scan excluding the segmentation. Since the data are collected from various clinical sites, there is a considerable variation of spatial resolution in the CT image. The axial slice has the same size of 512×512 in all patients, but the number of slices is varied between 42 to 1026 slices. Moreover, the in-plane resolution is range from 0.60 to 0.98 mm, and the slice spacing (axial direction) is from 0.45 to 5.0 mm as presented in figure 3.1.

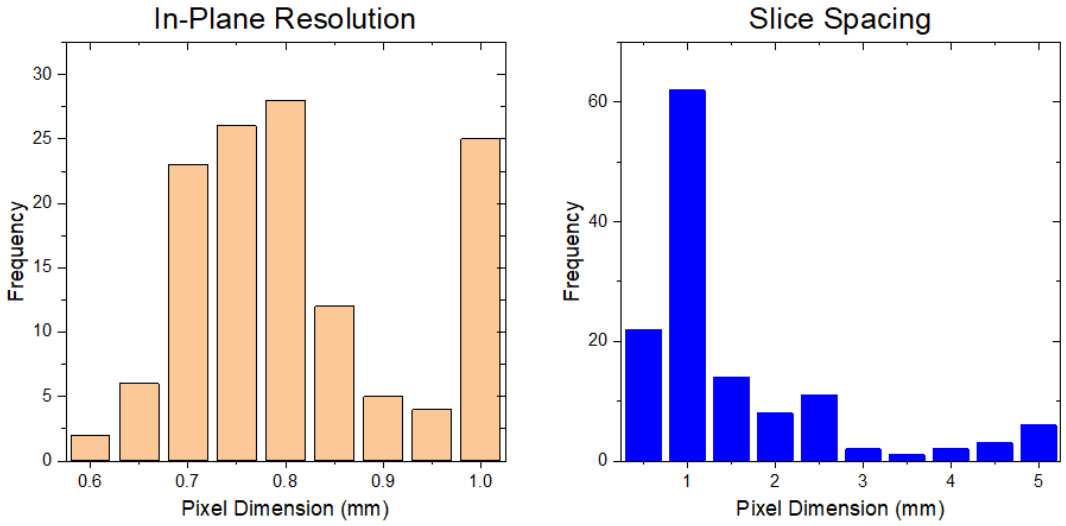


Figure 3.1 In-Plane Resolution and Slice Spacing from Training Dataset

3.1.1. Data Set Partitioning

In the training set, one patient data consists of the CT image and the segmentation. Both have the same slice number where the CT image contains the scan of the patient abdominal part, and the segmentation represents the liver and tumor region in the image. Since the segmentation will be performed in 2D, the CT image and the segmentation data need to be extracted from their 3D Volume. The CT image data is extracted into 2D slice image and stored in MAT file format. Instead of using a standard image file type like PNG or JPG, MAT-file type is chosen due to its capability to store multiple dimensions matrix in a file. It allows the extracted slice image to be stacked with other slices which form a 3D dimension matrix. Meanwhile, the segmentation data is extracted and stored in PNG file type. It is due to the slice image from segmentation can be treated as a grayscale image with the intensity value representing the class label. A standard image file type is sufficient to handle the matrix data from the segmentation

image. After the extraction process finished, the training dataset is divided into two parts, train data and test data. Detail explanation regarding the data set partitioning is shown in figure 3.2.

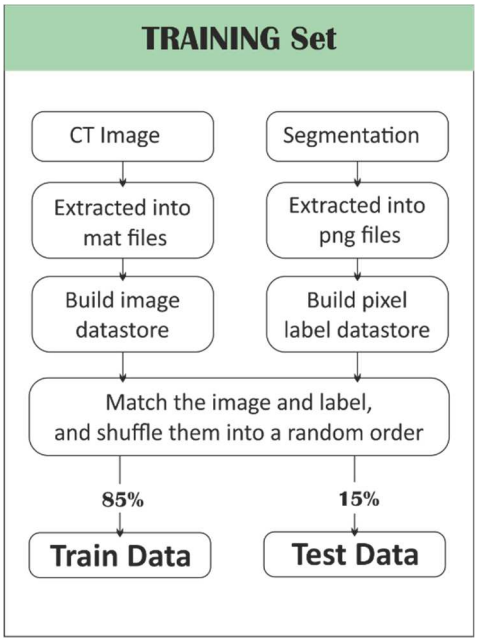


Figure 3.2 Partition on Training Set

3.1.2. Prepare Data for Network Input

The raw data set is constructed as 3D slices image, forming the volume of the patient body. Before these data can be fed into the network, data need to be prepared first. Due to the limitation of the memory, instead of using 3D data, the data is proceeded in a 2D manner. Not only reduce the computation needed, but also let the network design to be deeper and wider.

The image intensity in the CT Scan data is measured in the Hounsfield unit. Hounsfield unit (HU) is a scale that created by Sir Godfrey Hounsfield and becomes standard for CT image to represent the density of the object [43]. The value is based on the linear transformation, where for radiodensity of water has 0 HU and radiodensity of air has -1000 HU at standard temperature and pressure condition. The intensity of some organs in the HU scale can be seen in figure 3.3.

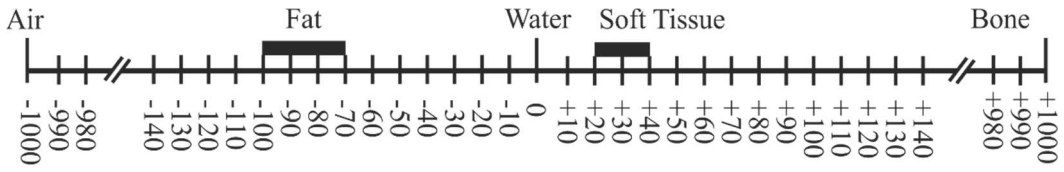


Figure 3.3 Various Organ Intensity in HU

Even though using the same scaling unit, the images still need to be calibrated due to the differences of machines and procedures. The images are calibrated by using a linear calibration function expressed in the equation 3.1, with "*a"* multiplication factor and "*b*" addition factor. These factors can be found in the information header of the CT Image.

$$HU = a.HU_o + b \tag{3.1}$$

After the calibration process is finished, the image intensity has a value in the range of -1000 to 1000. Windowing the image on a specific range can reduce the level of complexity and also increase the contrast in the image [17], [44], [14] and [45]. In this project, the image intensity is truncated into the range of -250 to 250 to focus on the fat tissue and soft tissue areas. After reducing the intensity range, the intensity value is normalized into a grayscale unit of 0 to 255.

In the case of the segmentation files, the image file uses a different scale unit. The image intensity only has three different values that represent the label of the image. Background label is signed by the value of 0, while the liver label has value 1 and the tumor label with 2. There is no preprocessing step needed in the segmentation file except to extract the 3D slices into a 2D slice and store them as PNG files.

3.2. Model Architecture

Neural Network model that is used to perform automatic liver and tumor segmentation is developed using Neural Network Toolbox from Matlab 2018a. The network design is inspired from the SegNet encoder and decoder network structure by Badrinarayanan [46] and modified with some improvement. The arrangement of the network layer is shown in figure 3.4.

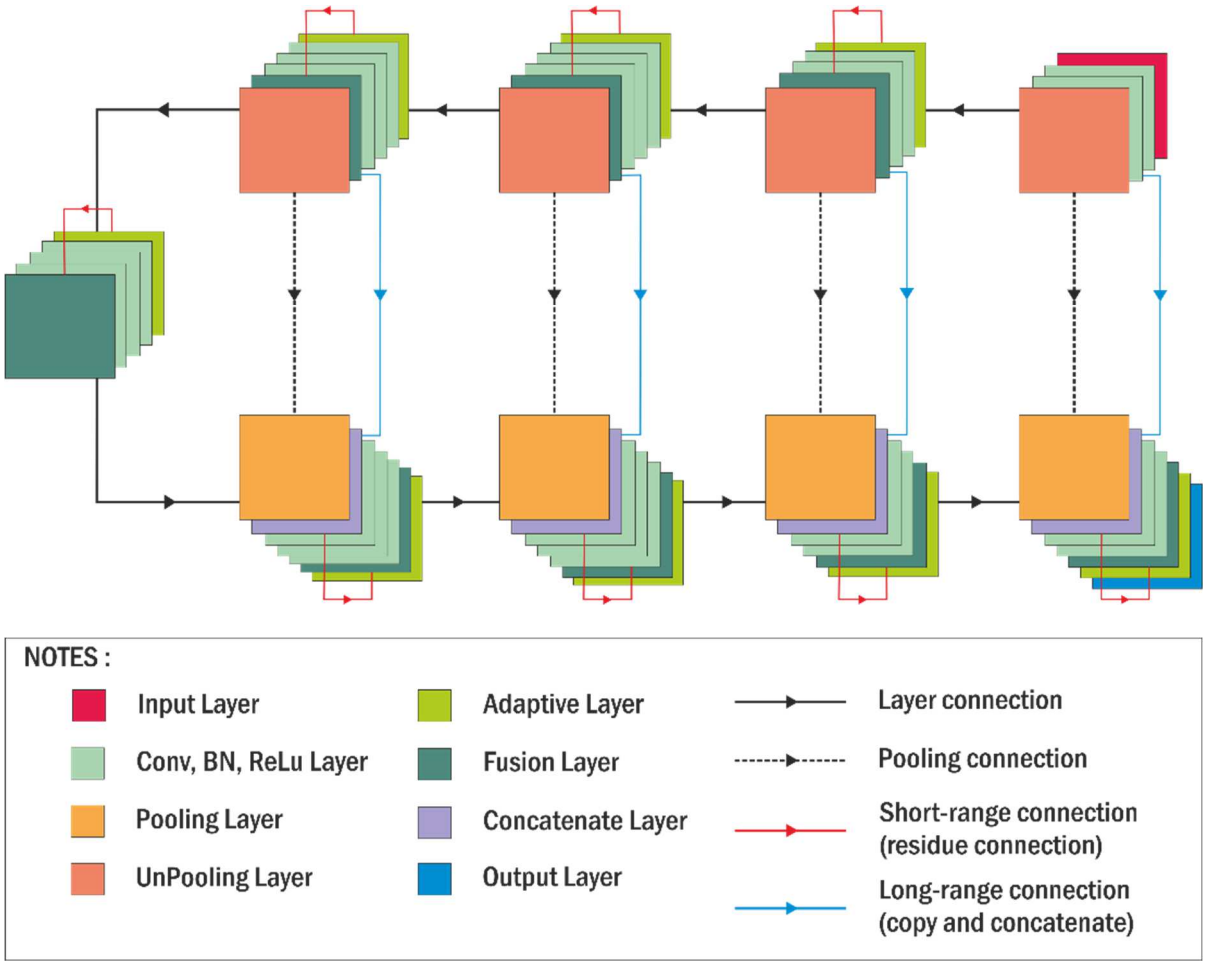


Figure 3.4 Network Model Architecture for Liver and Tumor Segmentation

Two main modifications are implemented in the basic SegNet network. The first modification is utilized the long-range connection that adopts from U-Net [47]. With this connection, the decoder can recover the spatial dimension details better. The input for this connection is taken from the output layer before the pooling layer in the encoder part. Then, it will be connected using the concatenate layer that stacked the input with another input from the un-Pooling layer. The second improvement is the implementation of the short-range or short-skip connection that usually found in the ResNet [27]. By using this connection, the result of the stacking layer will still preserve the information from the input. Therefore, it reduces the chance of gradient vanish in the deeper network and help the deeper network training easier to optimize. The input of this connection is taken from the adaptive layer in the design network and will be combined with the fusion layer that performs addition operations with the final output of the last stacked layer.

The full output size from the feature maps in every layer is shown in table 3.1. The number in the output size, e.g., 256×256×64, represented the spatial dimension and the number of channels from feature maps. The further explanation regarding the layer specification will be explained, Convolution layer in section 3.2.1, Batch Normalization and Activation Function in section 3.2.2 and the hyperparameter setting in chapter 3.2.3.

Table 3-1 Feature Maps Size in Each Layer

Layer	Output Size	Layer	Output Size
Input	256×256×3	dec_unpool_4	32×32×512
enc_conv 1_1 (3×3)	256×256×64	dec_con_4 (Concatenation)	32×32×1024
enc_conv 1_2 (3×3)	256×256×64	dec_conv_4_3 (3×3)	32×32×512
enc_pool_1	128×128×64	dec_conv_4_2 (3×3)	32×32×512
enc_conv_adapt_2 (1×1)	128×128×128	dec_conv_4_1 (3×3)	32×32×512
enc_conv 2_1 (3×3)	128×128×128	dec_conv_adapt_4 (1×1)	32×32×256
enc_conv_2_2 (3×3)	128×128×128	dec_unpool_3	64×64×256
enc_pool_2	64×64×128	dec_con_3 (Concatenation)	64×64×512
enc_conv_adapt_3 (1×1)	64×64×256	dec_conv_3_3 (3×3)	64×64×256
enc_conv_3_1 (3×3)	64×64×256	dec_conv_3_2 (3×3)	64×64×256
enc_conv_3_2 (3×3)	64×64×256	dec_conv_3_1 (3×3)	64×64×256
enc_conv_3_3 (3×3)	64×64×256	dec_conv_adapt_3 (1×1)	64×64×128
enc_pool_3	32×32×256	dec_unpool_2	128×128×128
enc_conv_adapt_4 (1×1)	32×32×512	dec_con_2 (Concatenation)	128×128×256
enc_conv_4_1 (3×3)	32×32×512	dec_conv_2_2 (3×3)	128×128×128
enc_conv_4_2 (3×3)	32×32×512	dec_conv_2_1 (3×3)	128×128×128
enc_conv_4_3 (3×3)	32×32×512	dec_conv_adapt_2 (1×1)	128×128×64
enc_pool_4	16×16×512	dec_unpool_1	256×256×64
enc_conv_adapt_5 (1×1)	16×16×512	dec_con_1 (Concatenation)	256×256×128
enc_conv_5_1 (3×3)	16×16×512	dec_conv_1_3 (3×3)	256×256×64
enc_conv_5_2 (3×3)	16×16×512	dec_conv_1_2 (3×3)	256×256×64
enc_conv_5_3 (3×3)	16×16×512	dec_conv_adapt_1 (1×1)	256×256×3
		dec_conv_1_1 (3×3)	256×256×3
		Output	256×256×1

3.2.1. Convolutional Layer

Two different convolution layer sizes are used in this network design. Those are convolution layer that has 3×3 kernel size and convolution layer with 1×1 kernel size. An illustration of these convolution layers are shown in figure 3.5 [48].

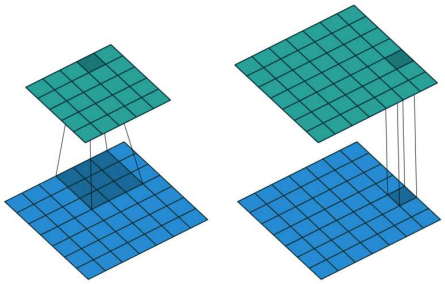


Figure 3.5 Convolution Layer with (left) 3x3 kernel size and (right) 1x1 kernel size [48]

The convolution layers with 3×3 kernel size are used in all encoder parts, where there are two stacking convolution layers in the first two part and the other parts have three stacking convolution layers. These convolution layers have the same configuration such as the kernel size of 3×3, stride of 1, and padding the border to get the same output size as an input. The only different set is the number of filters. The number of filters is varied from a small number in the earlier part and get larger along with the deeper the location of the part in the network. As an example, the convolution layer has 64 filters in the first part, while there are 512 filters in the last encoder part.

Another type of convolution layer is the one with a kernel size of 1×1. This convolution layer is called an adaptive layer and used in all encoder part except in the first part. The function of this layer is to transform the feature from the previous part into the same feature size as the current part. For instance, if the output image from the pooling layer in the first encoder part has a size of 256×256×64, by applying the convolution layer of 1×1 with filter number of 128 will produce an image with size 256×256×128.

3.2.2. Batch Normalization and Activation Layer

Batch Normalization method is introduced by Ioffe [49] in order to perform a normalization with mean and variance from the hidden unit values. With this method, higher learning rate can be employed, and the faster training process can be achieved. Using the Batch Normalization, the same accuracy can be achieved with 14 times fewer training steps compared to the original model. Furthermore, Batch Normalization has regularization effect that is similar to drop out that can reduce the overfitting.

The activation function for the network is chosen based on the experiment result that has been done by Schilling [50]. In this experiment, the effect of Batch Normalization on the deep convolutional neural network are investigated using several datasets where several combinations of Batch Normalization layer and activation function layer are tested. The result shows that the combination of Batch Normalization with ReLU activation function match or outperform the result from other combinations on all model and dataset. Based on this result, for every convolution layer in the network design, it will be followed by Batch Normalization layer and ReLu activation layer.

3.2.3. Hyperparameters

Hyperparameter setting in this design consists of the setting for the network layer and the training options for the network. Beside initial learning rate and solver function, other parameters that need to be set up in the training options are dropout rate, dropout factor, epoch, and batch size. Dropout rate and dropout factor are used to reduce the learning rate gradually, where the dropout rate controls the reduction period, and the dropout factor manages the reduction factor. Epoch and batch size parameters are related to the iteration during the network training. Epoch describes the number of iterations over the data set during the training process, and batch size defines the number of training examples for updating weight process. Due to the memory constraint from the system, batch size in the training options has two different values that depend on the resolution of the image dataset. Table 3.2 shows the complete hyperparameter setting for the network.

Table 3-2 Hyperparameter Setting

Network Layer	Value	Training Options	Value
Activation Function	ReLU	Initial Learning Rate	0.001
Adaptive Convolution		Optimizer	SGDM
Padding	same'	Momentum	0.9
Kernel size	1×1	Dropout rate	5
Stride	1	Dropout factor	0.1
Convolution		Epoch	20
Padding	same'	Batch Size	
Kernel size	3×3	*Input 256×256*	8
Stride	1	*Input 512×512*	1
Pooling Layer			
Type	MaxPooling		
Stride	2		
Size	2×2		

3.3. Evaluation Metrics and Method

The performance of the training network will be measured using a confusion matrix which comparing the ground truth segmentation (actual value) with test segmentation (predicted value). If the actual values are predicted correctly, then it will be count as True Positive (TP) for positive values and True Negative (TN) for negative values. However, false prediction on the actual values will be scored as False Positive (FP) or False Negative (FN). Figure 3.6 shows an example of the confusion matrix. Using this matrix, further evaluation will be computed.

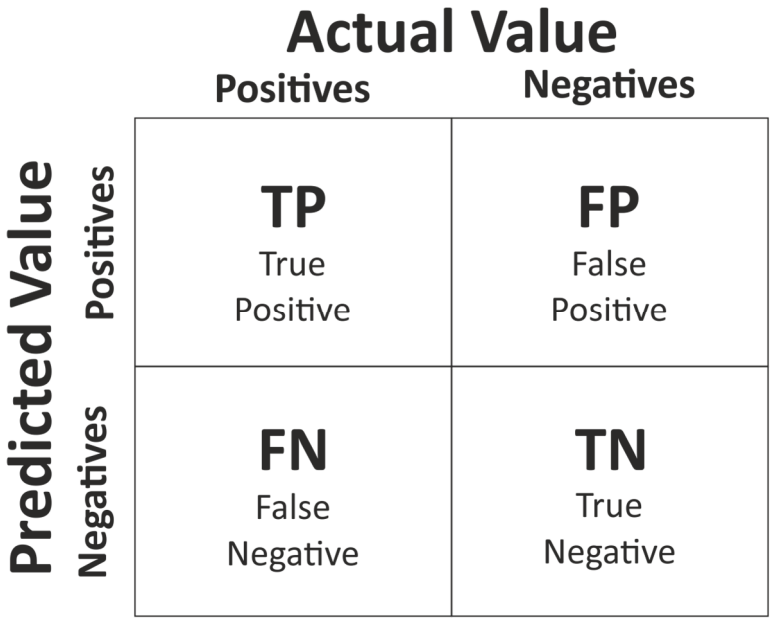


Figure 3.6 An Example of Confusion Matrix

3.3.1. Spatial Overlap Based Metric

In medical imaging, the most used evaluation metric is Dice score. The function has a value in a range between 0 and 1, where 0 indicate no match found between the predicted value and actual value and 1 means prediction and ground truth has an exact value. Dice is calculated using the equation 3.2, with S_A^1 positives value from the ground truth and S_P^1 positives value from predicted.

$$Dice = \frac{2\left|S_A^1 \cap S_P^1\right|}{|S_A^1| + |S_P^1|} = \frac{2TP}{2TP + FP + FN} \tag{3.2}$$

Jaccard Index is also popular evaluation metric that defined by the intersection over the union. Jaccard Index is quite similar to Dice score where both of them have a range between 0 and 1. and can be related to using equation 3.3. Since Jaccard and Dice measure the same aspect in the system, only one of them is usually chosen as a validation metric [51].

$$Jaccard = \frac{\left|S_A^1 \cap S_P^1\right|}{|S_A^1 \cup S_P^1|} = \frac{TP}{TP + FP + FN} = \frac{Dice}{2 - Dice} \tag{3.3}$$

Another measurement that can be used is Recall and Precision. Recall or known as True Positive Rate measure the fraction of positive value in the ground truth that identified as positive also in the prediction. Meanwhile, Precision or Positive Predictive Value measures the fraction of true positive in the whole prediction result. Both measurements are defined in the equation 3.4 and 3.5.

$$Recall = TPR = \frac{TP}{TP + FN} \tag{3.4}$$

$$Precision = PPV = \frac{TP}{TP + FP} \tag{3.5}$$

3.3.2. Evaluation Steps

The performance of the algorithm will be evaluated in several steps. In the first step, the algorithm will be validated using the training data from the competition. There are 130 patient data that provided for the training purpose. From these data, 20 patients will be separated from the training dataset and will be used as a test set for evaluating the network model. For the second steps, an experiment will be held that is followed by ten technical physicians and an international radiologist. The participant will be asked to perform a separate manual segmentation on several patient data set. Later, the automatic result from the network model will be compared with these manual segmentations. As the last step, the network will be employed to perform segmentation on all the test set that provided by the LiTS Challenge. There are 70 patients that need to be segmented. The result will be collected and submitted to the LiTS website. The evaluation result will be evaluated and compared with other algorithms from other participants which shown in the leaderboard online at the LiTS website.

4. Experiment Setup

Developing network for segmenting the liver and tumor automatically from the CT image is the primary aim of this project. During the network development, various configurations, such as preparation in the training data and configuration in the network structure, were used for training and testing the network to obtain the best segmentation result. Besides, manual segmentation experiment is conducted to evaluate the network performance by the result of the participant with technical medicine and clinical background. All experiment setups are explained in more detail in this chapter.

4.1. Training and Testing Workflow

Various networks will be trained and tested to investigate the influence of different configurations on the segmentation result. To facilitate the training and testing process, workflows for network training and testing have been designed. Training process workflow and testing process workflow are discussed in section 4.1.1 and section 4.1.2 respectively.

4.1.1. Training Workflow

Figure 4.1 shows the workflow of the training process. In the beginning, the training data will be loaded to build the training databases, included a label and a image database. For the label database, the 3D patient segmentation will be extracted into 2D slice data and will be stored in PNG image format. Meanwhile, the 3D patient data for image database need to be calibrated first and then extracted and stored in MAT file format. After that, extra processes, such as slice arrangement and contrast enhancement, are implemented in the image data before storing them into 2D slice image. These extra processes are discussed further in section 4.2.

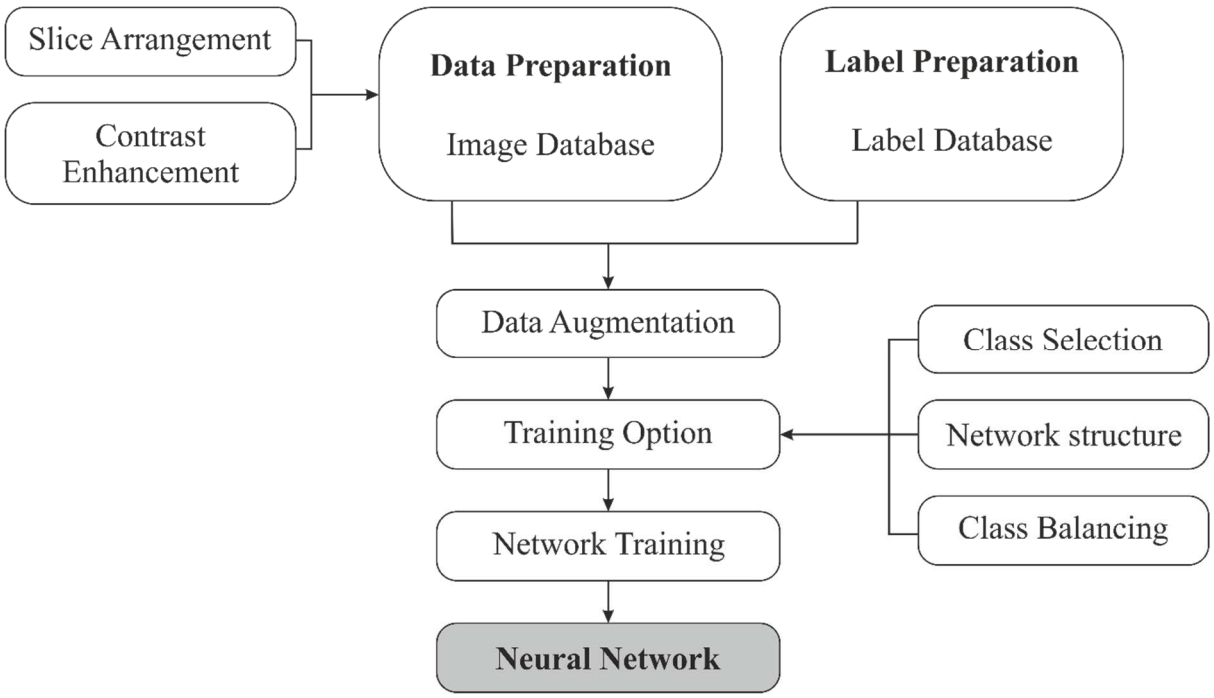


Figure 4.1 Workflow for Training a Network

Data augmentation is the next step after preparing the image and label databases. It is needed to increase the number of the training data. In this step, a minor alteration like reflection, rotation, scales, and transitions are randomly applied to the training dataset to create augmented data. The more the training data, the better the parameter in the network tuned. Therefore, the network can become more robust and become more independent to the variation of the object in the image.

Another configuration that needs to be set up before training the network is the training option. Some hyperparameters such as layer specification, learning rate, and data batch size are set to be the same for all network with the configuration as mention in section 3.3.3. However, other parameters like class balancing and network structure are set with several configurations to inspect the effect of those parameters on the segmentation result. A setting like class selection has different value depending on the case that will be examined. As an example, the parameter of class selection is set to recognize only background and liver class in the data preparation experiment, while tumor class is included in the network configuration experiment.

4.1.2. Testing Workflow

It is necessary to test the network after being trained. In general, the testing procedure has followed the workflow as shown in figure 4.2.

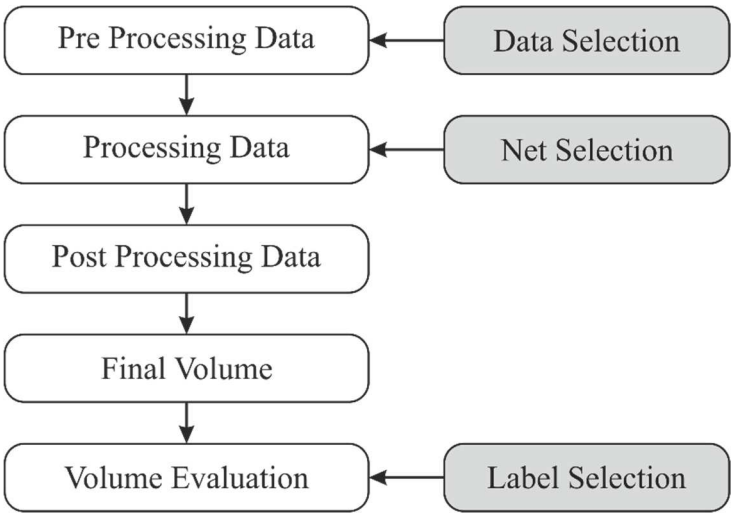


Figure 4.2 Workflow for Testing a Network

Grey box in the workflow indicates some initial steps that must be completed before starting the test. The first step is data selection, where another dataset that was excluded from the training dataset is chosen. In this experiment, the testing dataset consists of 20 image patient data including their segmentation map. The other steps are Net selection to select the network that will be tested, and Label selection to decide the class that will be included during the experiment. Liver and tumor classes are combined to be liver class in liver segmentation case, while they will be treated as different classes in the tumor segmentation case.

Testing data that have been selected must be treated similarly as the training data. In this process, the image intensity will be calibrated using a calibration function in section 3.1.2. Then, the image intensity will be truncated, and the image resolution is adapted to the image training resolution. In addition, the same setup in data preparation experiments such as slice arrangement or the contrast enhancement

should be implemented in the testing data. The segmentation process can be started after all the pre-processing steps are finished.

The segmentation is done in a slice by slice manner, where each slice will be segmented individually, and the result will be combined later with other segmentation to form a 3D segmentation. The segmentation results from this process are mapped with the categorical value like background, liver, and tumor. To evaluate the result with the ground truth, it is required to compare the results using the same data type. In that case, the categorical values are translated into numerical value where the background is represented by 0, liver with 1 and tumor with 2.

Some post-processing functions are implemented to reduce the noise in the final segmentation volume. First, false positive segmentations are removed by selecting the largest 3D connected component from the volume. Afterward, the volume is smoothened by performing some morphological operations such as erosion and dilation, to the segmentation result. After this process, the final volume is compared with the ground truth to obtain the dice score and Jaccard index which will be used to evaluate the network performance.

4.2. Dataset Preparation

The general purpose of the experiment in this part is to investigate the influence of the dataset preparation to the segmentation result. Two different preparations will be examined. The first preparation is about the slice arrangement in the data set and the second is about the contrast level of the image.

Data that provided by the LiTS Challenge will be used as a dataset in this experiment, where training data is consist of 109 patients and test data with 20 patients. The resolution of all image data will be reduced by half size. It means that the original image resolution of 512×512 will become a new image with 256×256 size. The size reduction is employed to ensure that the process has enough memory and to allow a higher batch data can be used during training.

4.2.1. Slice Arrangement

In image segmentation case, Convolutional Neural Network (CNN) recently shows an outstanding result compare to other segmentation methods. However, the CNN method can give a different result when employed in a medical area where most of the images consist of a volumetric image. Features that extracted from 2D convolutions is not enough to cover the spatial information in the third dimension. While implementing 3D convolution encounters several issues such as high computational cost and high memory usage. To tackle this problem, a model of 2.5D approach is favored where the 2D image is stacked on top of each other to build a small volume. By using this way, 2D slice image can be extracted to obtain the intra slice features while the stacked slices provide enough information in the third dimension.

From some research that develops the automatic segmentation method for liver and tumor, the slice for the dataset is arranged in a different manner, where [52] used three stacked slices and [18] used five stacked slices. However, no comparison is made in those studies regarding the influence of the number of stacked slices to the segmentation result. In this experiment, the purpose is to inspect the slice arrangement factor related to the network performance. Slice arrangement means how the slice is organized in the dataset. There are five different variations in the dataset, from 1 slice, 3 slices, 5 slices, 7 slices and 9 slices with the segmentation map corresponding the center slice of the stack. The illustration of these datasets can be seen in figure 4.3.

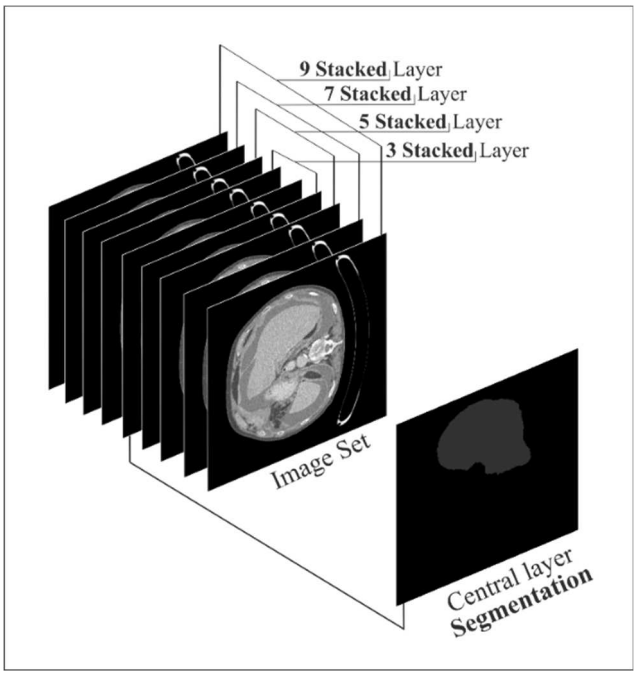


Figure 4.3 Slice Stacking Illustration

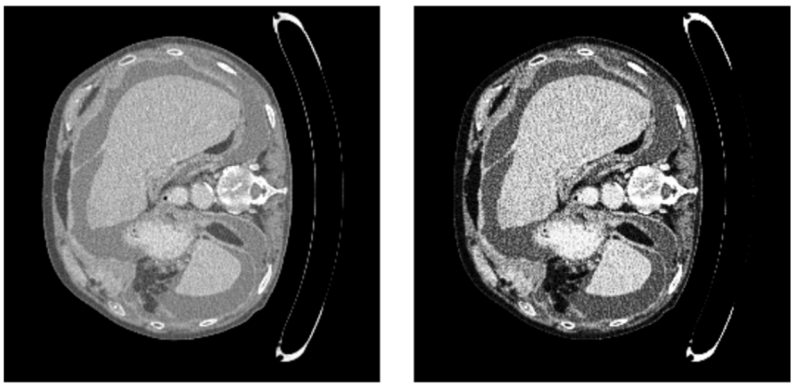
4.2.2. Image Contrast

Windowing the image intensity based on the HU scale has been done during the preparation of dataset. The aims are to reduce the complexity and to increase the contrast in the image. However, the image contrast can be further enhanced by using different methods, such as Histogram equalization, gamma correction, and bilateral filtering.

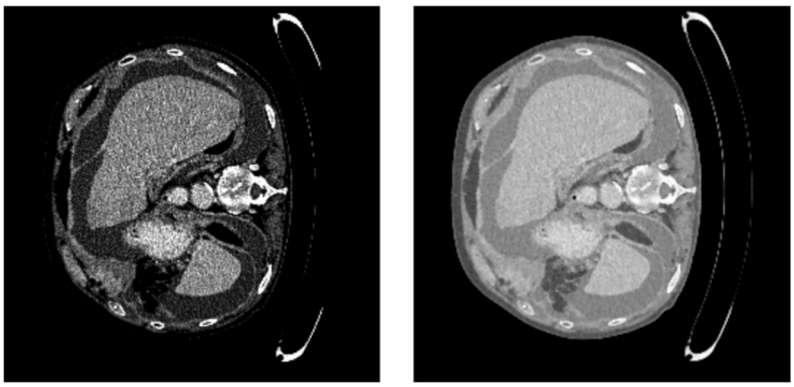
Histogram equalization improves the image contrast by distributing the image intensity using the histogram data. It maps the most frequently intensity value into a new intensity distribution, so the values are spread over equally. In some cases, it can give a better view of the scientific image, like satellite and x-ray images. Another contrast enhancement method is Gamma correction. This method applies a nonlinear operation to the image based on the gamma value (γ). The operation can be expressed using equation 4.1, where V_{in} input image, A constant value, and V_{out} output image.

$$V_{out} = AV_{in}^{\gamma} \tag{4.1}$$

Before implementing this method in the image, the new intensity range is chosen based on the observed objects, which are liver and tumor in this case. In the range of 0 to 1, the new image intensity is set from 0.4 to 0.9. This new range is selected to reduce the darker pixel effect while increasing the contrast from an object that has brighter pixel. After that, the gamma correction is done to the image using a gamma value of 1.4. For the last enhancement method, bilateral filtering using a different approach to increase the image contrast. Instead of working in the pixel intensity distribution, this method enhances the contrast by reducing the noise from the image. Therefore, the object becomes smoother and easier to be segmented. The preview of various contrast enhancement methods is presented in figure 4.4.



a. Original Image *b. Histogram Equalization*



c. Gamma Correction *d. Bilateral Filtering*

Figure 4.4 Contrast Enhancement Method

4.3. Experiment on Network Setup

Unlike in the previous experimental setup, the networks that are trained in this part are expected to be able to segment liver region including the tumor region. The dataset still contains the same dataset as the previous one, except for the additional processing such as the number of stacking slice and image enhancement technique that will follow the result from the preceding experiment.

Two different upgrades will be applied and examined for their impact on the final segmentation in this section. The first upgrade is related to the class balancing, as an effect of adding tumor class in the segmentation result, while the second upgrade is concerned about the layer expansion on the network design.

4.3.1. Class Balancing

For training a network, a balanced data in all class label is preferred to counter the effect of the dominant class in the segmentation result. However, in some cases, it is very hard to get an equal amount of data for each class. For instance, the number of tumor data in this experiment is always smaller compared to the liver data or the background data. Figure 4.5 shows the normalized frequency of background, liver and tumor pixel in the training dataset.

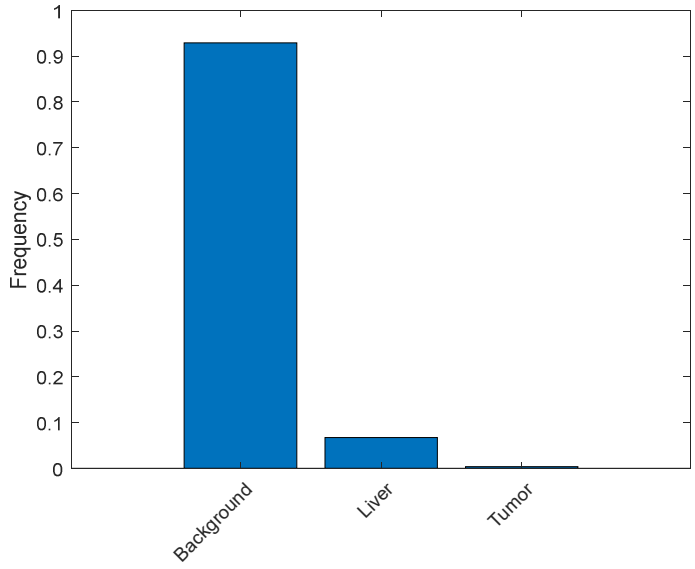


Figure 4.5 Pixel Class Distribution

Several methods have been discussed by López et al. [53] to overcome the unbalanced data. The first method is data resampling, where the training data is modified to produced more balanced data. In this project, undersampling technique has been applied in training set, where majority class (background) is eliminated from the training set by selecting only the slice that contains liver and tumor for training the network. The second method is applying the class weight that adjusting the cost of the class error. The lower the presentation of the class, the higher the class weight it has. Because of a higher-class weight, the under presented class error will be considered more costly compared to the other classes which will give more influence during weight adjustment in the training process. To compute the class weight in the image segmentation case, the total pixel number in each class is needed. After that, class weight is calculated using equation 4.2.

$$classWeight = \frac{total\ pixel\ number}{total\ pixel\ in\ that\ class} \tag{4.2}$$

Although the implementation of class weight may help to improve the segmentation result, sometimes leaving the network setup as it is can still give a proper result (i.e. natural distribution on the data set). Due to these reasons, the experiment will be performed with the aim to investigate the effect of class weight in the liver and tumor dataset.

4.3.2. Network Architecture Comparison

An upgrade has been given to improve the architecture of the basic network. In this design, the network architecture is expanded from one into two encoder and decoder components as can be seen in figure 4.6. It is based on research by Wang et al. [54] that found an advanced method to increase the number of layers in the U-Net structure. Instead of adding more layers in the convolution layer of each encoder and decoder part, a dense connection between encoder and decoder can achieve improved accuracy with the fewer number of parameters.

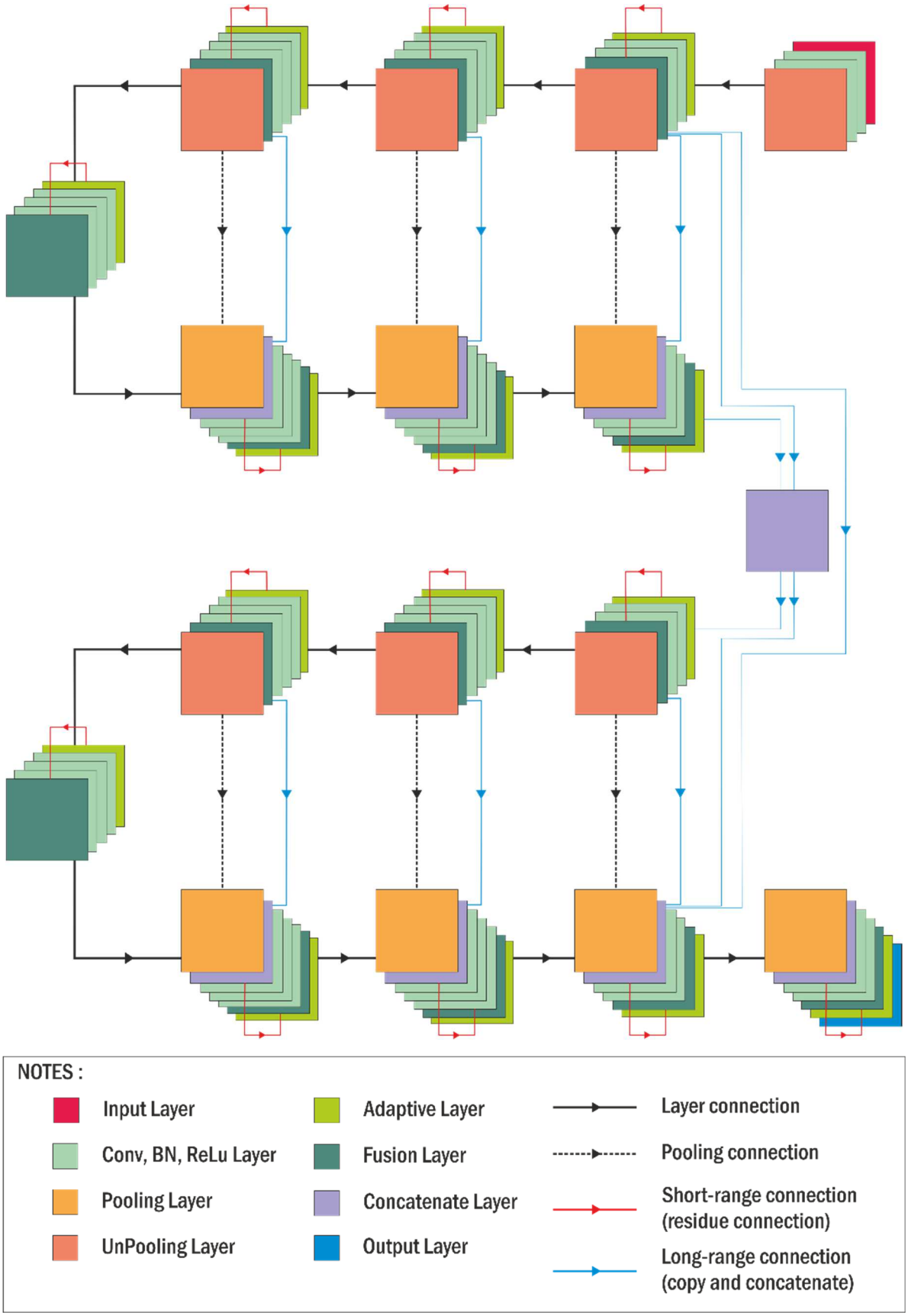


Figure 4.6 Expanded Version of Encoder and Decoder Network Model Architecture

To evaluate this upgrade, the new design is employed to train the network for segmenting the liver and tumor. In the end, the comparison will be made in term of the segmentation result between the basic network and the upgraded version.

4.4. Segmentation from Technician

To give an additional insight regarding the performance of the automatic method, an experiment has been done and followed by subjects with a clinical background (an international radiologist and ten technical physicians). In this experiment, the participants perform three activities. First, the participant will be asked to fill a participant consent as included in the Appendix 1. After that, the participant will perform a manual segmentation of the liver and tumor from 15 different patient slices. Then, the participant will use the application that employed the automatic segmentation algorithm to segment the patient data. The user guide for performing manual and automatic segmentation have been prepared and included in the Appendix 2 and Appendix 3. In the end, a survey that collects the user opinion regarding the performance of the automatic segmentation is answered by the participant. The example of the survey is given in Appendix 4. The results of this survey are employed to measure the performance of the algorithm.

In addition to the survey results, a manual segmentation that obtained from the first experiment will be used to measure the algorithm performance. The measurement is performed by comparing the manual segmentation and the automatic segmentation results with the ground truth provided by the LiTS. After that, the Dice score and Recall between manual segmentation and automatic segmentation will be computed.

5. Experiment Result and Discussion

This chapter shows results and discussion regarding the network experiments using various configuration, including dataset preparation and network configuration, and the manual segmentation experiments. Mostly, training of the network model in this experiment has been done using a single NVIDIA GeForce GTX 1070 GPU with 8 GB memory. However, for training a network that required larger memory, such as the expanded version network (section 4.3.2) using the dataset with image resolutions of 512×512, the training process could not be performed due to insufficient memory error. To solve this issue, a cluster that provided by CTIT computing lab has been employed for the training process. In this cluster, a high-end computer partition that powered by 80 CPUs, 512GB RAM and two 16GB Tesla P100 is selected to ensure that all network configurations can be trained.

5.1. Dataset Preparation

The purpose of the dataset preparation is to produce a data set that can provide better information to the network on how the object should be separated from the image. In this section, two preparations are examined, the slice arrangement and the image contrast.

5.1.1. Slice Arrangement

Slice arrangement is a part of dataset preparation where the slice in the dataset is arranged in a stacking manner. The center slice that corresponds to the ground truth is combined with the slice from its top and its bottom, produce a small 3D image from stacked 2D slices. In this section, five different configurations are tested as shown in figure 5.1.

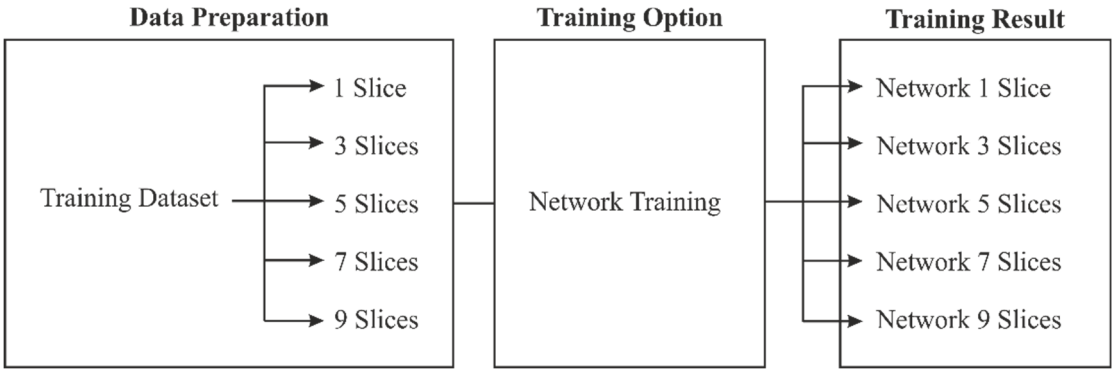


Figure 5.1 Slice Arrangement Experiment Workflow

Not all slices from the training dataset are included for training the network. Only slices that contain liver and tumor are selected which gives 2925 slices in total. Several adjustments are made in the training options, such as setting mini batch size to 8 and total epoch to 20. Using this setting, training a network takes 365 iterations per epoch, where becomes 7300 iterations for the whole process. Another adjustment is related to the learning rate setting. Initial learning rate value is set to 10^{-3} which drop periodically by the factor of 0.1 after 5 epochs. Stochastic Gradient Descent with Momentum is chosen as solver function with the momentum of 0.9. The network structure that described in section 3.2 are employed for training each slice arrangement. Total time needed for training one network is 200±12 minutes.

After finishing the training process, each network is evaluated using 20 image test set. In this process, liver regions are segmented by following the process that described in figure 5.2. The result from each slice then combined and stacked to form a 3D image.

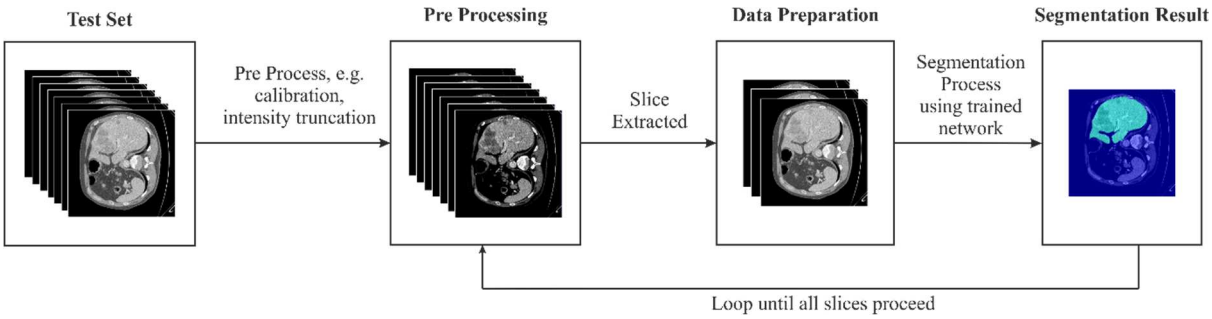


Figure 5.2 Evaluation Process Workflow

The network performance is measured by comparing the Dice score from all the data test set. Complete result of this measurement is shown in figure 5.3

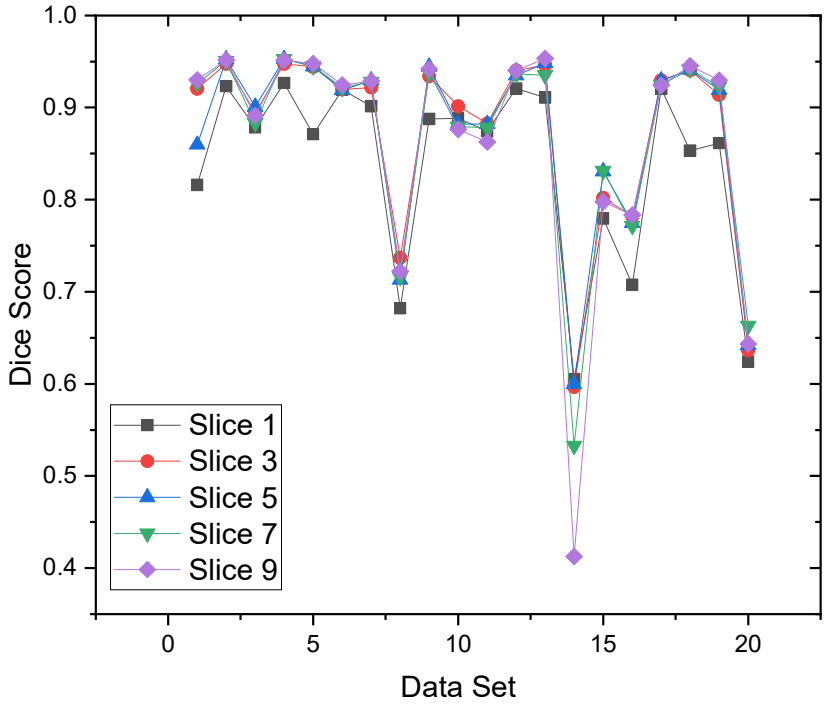


Figure 5.3 Dice Score Result from Various Networks That Trained Using A Single Slice and Multiple Slices of 3,5,7 and 9 Slices.

There are variations from the dice score result obtained. Based on the graph trend, the result from the network with one slice is inferior compared to other networks that adopt stacked slice. In average, network with one slice obtains the dice score of 83.6%, while networks with 3, 5, 7, and 9 slices get 86%, 86,5%, 86,3% and 85.3% respectively. A conclusion based on the results, more stacked slices are not always given an increment to the network performance. It is true that the dice score from 1 slice to 3 slices and from 3 slices to 5 slices increases. However, the scores stop increasing and decrease gradually by adding extra slices due to the utilization of 2D convolutional in the network layer. Even though more 3D volume context information is provided from more stacking slice, the network keeps

processing the image slice by slice. In the hyperspectral image, the shape of the object remains the same along the layers. However, in the CT scan images which built from a consecutive 2D slice image, if the stacking slice number is too much, the object structure from the top and bottom layer will have a significant difference shape which will distort the shape information and reduce the performance of the network in recognizing the object.

5.1.2. Image Contrast

One of the challenges when working with medical images is the unclear border among the organs. It is hard to segment the image when no distinct border separates the object. To address this issue, increasing the contrast in the image can be a solution. This will make the intensity difference higher so that the border becomes clearer.

In this section, the data preparation focusses on the methods for increasing the image contrast from the dataset. Three methods with different approaches are implemented in this experiment, histogram equalization, gamma correction and bilateral filtering. Histogram equalization increases the image contrast by adjusting the image intensities based on the histogram information. Meanwhile, gamma correction works based on the gamma value. A gamma value less than 1 will map the image toward a brighter area while a gamma value more than 1 will map image toward a darker area. Furthermore, bilateral filtering enhances the contrast by reducing the noise from the image. It is done by replacing the intensity of each pixel with the weighted average intensity that obtained from the Gaussian distribution of neighborhood pixel.

Each method has been applied during the dataset preparation to produce several training datasets. These data are used to train the networks which later will be compared to decide the best method for increasing the segmentation result. Training and evaluation process in this experiment are similar to the process that has been done in the slice arrangement experiment. Figure 5.4 shows the segmentation result on 20 image patient data using several contrast enhancement methods.

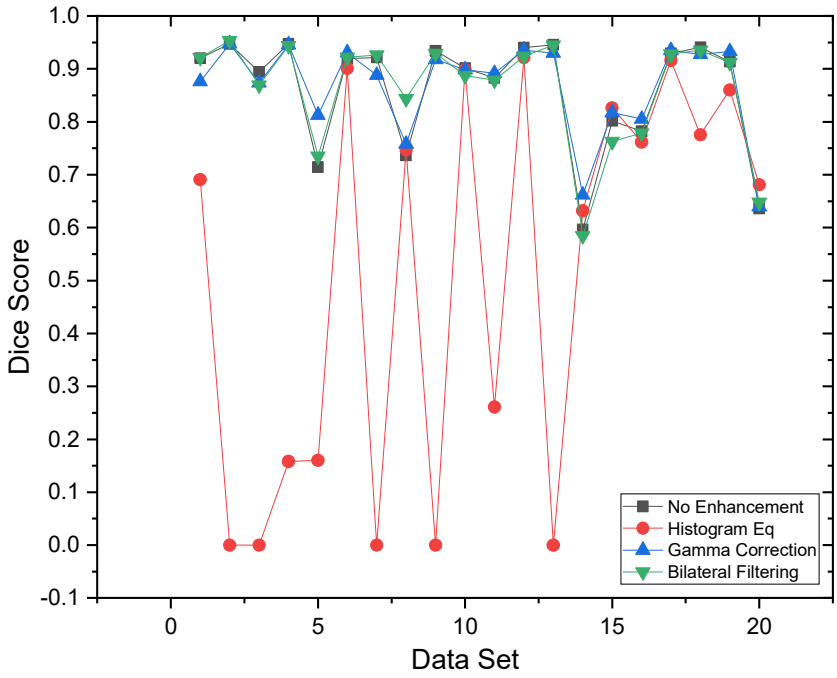


Figure 5.4 Dice Score Result from Various Network That Trained Using the Dataset with and without Contrast Enhancement Techniques

Figure 5.4 shows that employing histogram equalization to increase the image enhancement contributes to a lower Dice score result than other techniques. Histogram Equalization enhances the contrast by making the distribution pixel intensity more equal based on the image histogram. However, this method relies on the histogram data, which are varied along the image. As a result, different treatment will be given to each slice of the image. It causes different intensity distribution to an organ that has volume spread over several slices. In the convolutional neural network, the feature of the object is extracted using a convolution operation. Features are described as a relation among local pixel neighborhood. If the intensity of an object is varied in every slice, then it will be hard to determine specific features that represent the object in the image. Therefore, network with histogram equalization method has a poor segmentation result with average dice score of 50.9%.

On the other hand, applying the gamma correction and bilateral filtering technique to the image increase the dice score of liver slightly. In the original image that applied no contrast enhancement, the average dice score is 86.04%, while in the network with image adjustment and bilateral filtering become 86.62% and 86.14% respectively. The implementation of these techniques gives an opposite result compared to the histogram technique. Instead of using histogram data that vary over the slice, both techniques increase the object contrast using the pixel intensity information. Although the image changed due to the contrast enhancement effect, the pixel intensity ratio remains the same which cause an object can still be recognized as the same object after the enhancement method applied. Despite the minor improvement to the liver dice score, the utilization of these methods proves that the contrast enhancement technique can increase the network performance in segmenting an object.

A further experiment has been done based on the result achieved by the gamma correction and bilateral filtering. Since both methods work in different domains where gamma correction is used for increasing the image contrast and bilateral filtering is applied for reducing the image noise, implementing both methods in the dataset will not affect other results and supposed to increase the dice score higher. As expected, training the network using both techniques give an increment to the dice score to 87.77%, which exceed the result from other configurations.

5.2. Network Configuration

Tuning the hyperparameter is needed to obtain the desired network. One of the hyperparameter components that could be adjusted is the network configuration. In this section, two factors are examined to inspect their effect on the liver and tumor segmentation. First is the factor of unbalanced class in the dataset, where the pixel number from each class, background, liver, and tumor, has a significant difference. The second factor is the factor of changing the layer number and the layer architecture in the network design. During the experiment, networks are trained using the same training option as mention in chapter 5.1. In addition, tumor class is included as a parameter that is measured in the evaluation process.

5.2.1. Class Balancing

Pre-processing step has been done to reduce the unbalance effect, where only the slice with background, liver, and tumor are employed during training. However, the unbalanced class still found in the dataset with a significant difference among the classes. The representation of class frequency distribution is a background class with 92,91%, liver class 6.72% and tumor class 0.37%. To examine the unbalance class influence, there are two different networks tested, network with Class Weight (CW) and without Class Weight. The class weight value is calculated using the equation 4.2, which gives the result 1.10, 12,08 and 98,83 for background, liver and tumor class respectively.

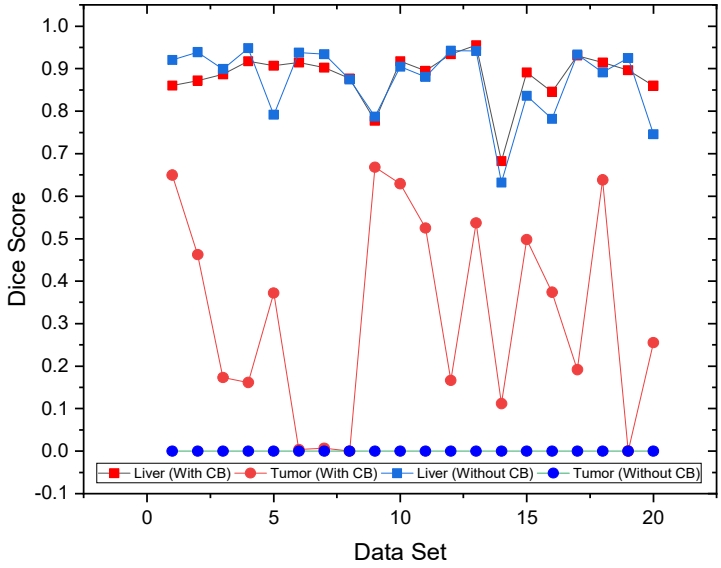
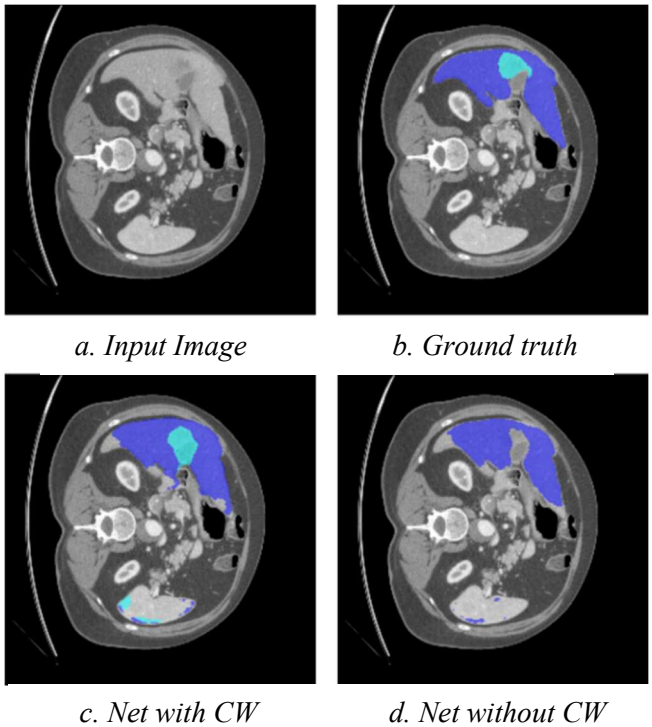


Figure 5.5 Comparison of Liver and Tumor Dice Score Result from Networks with and without Class Balance

Testing has been done on both networks, and the result can be seen in figure 5.5. From this result, it shows that the liver area can be identified and segmented correctly using both networks, achieving high dice score around 87% for both. However, the liver dice score contradicts with the tumor dice score. In the network without class balance, none of the tumor areas can be identified with zero scores on all dataset. When the pixel number difference is too high among the classes, network prefer to maximize the weight accuracy and reduce the error loss during training by focusing on the larger class and ignoring the smaller class. Figure 5.6 gives an example of the segmentation result from both networks.



a. Input Image *b. Ground truth*

c. Net with CW *d. Net without CW*

Figure 5.6 Segmentation Result with and without Class Balance

5.2.2. Network Architecture Comparison

Another network has been developed to see the influence of adding layers and changing the architecture into the segmentation result. The new network structure can be seen in section 4.3.2. To differentiate both networks in this experiment, the first network design is called Net01 and the second network is Net02. Both networks are trained by implementing all results that have been obtained from the previous experiment, such as utilization of stacked slice, application of image enhancement technique and employment of class weight in the network. In this experiment, both networks will be tested using the 20 dataset that taken from training data. Afterward, the Dice score from each dataset will be measured and used for making a comparison.

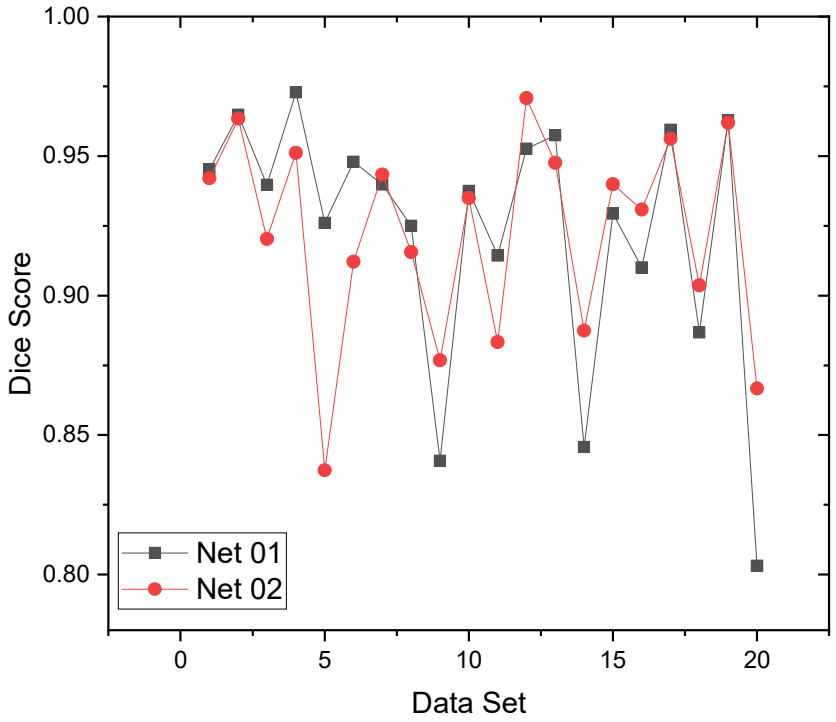
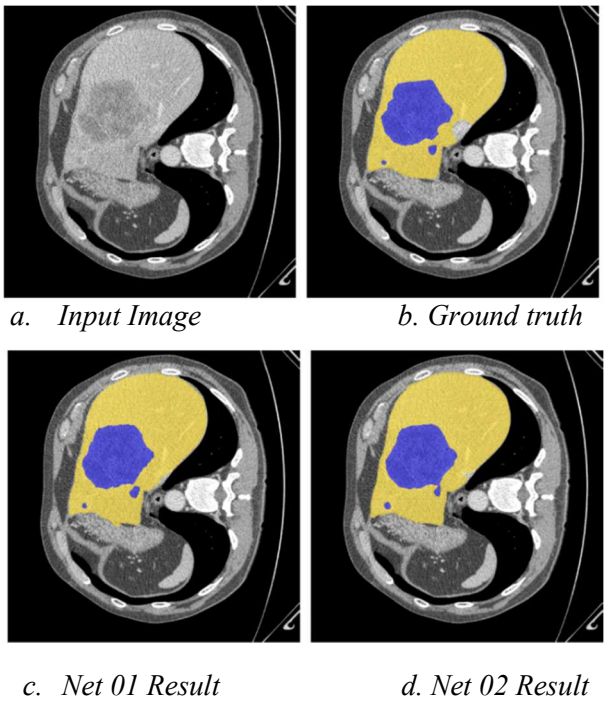


Figure 5.7 Comparison of Liver Segmentation Dice Score from Net01 and Net02

The dice score of liver segmentation is presented in the charts in figure 5.7. From this chart, both dice score from Net 01 and Net 02 show fluctuated result in the whole dataset. The variation on the result can be understood since the ground truth data is obtained from different clinicians. As a result, different perspectives on how the manual segmentation is made could happen which affect the ground truth data. Even though the result fluctuates, both networks show a similar trend, where the increment or decrement of the dice score in Net01 will be followed with the same movement in Net02. This explains why the average dice score from both networks are slightly different, with 92.3% and 92.2% respectively. With dice score higher than 92%, it can be said that both networks can produce remarkable liver segmentation. The comparison of liver segmentation among ground truth, Net01 and Net02 is shown in figure 5.8.



a. Input Image *b. Ground truth*

c. Net 01 Result *d. Net 02 Result*

Figure 5.8 Liver Segmentation Comparison

In figure 5.9, tumor segmentation from Net01 and Net02 are displayed. A different segmentation result is obtained in the evaluation of tumor segmentation compared to the result of liver segmentation. While all the score of liver segmentation is higher than 80%, there are only two results that have the score higher than 80% in the tumor segmentation. Almost all the results are below than 80% with some data that get zero scores in both networks. It can be explained, since segmenting the tumor is harder than segmenting the liver. There are many variations among the dataset, such as the number, size, location, and shape of the tumor which make more features and process needed to segment it. Despite the score variation in the result, Net01 show a higher score in the majority of the dataset than Net02. With an average score of 54.5%, It is proven that Net01 can give better performance in segmenting the tumor compared to Net02 with an average score of 49.2%.

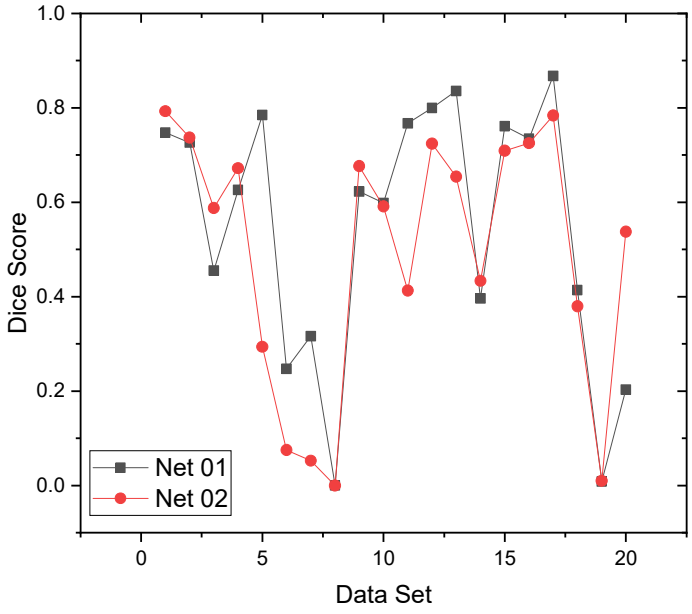
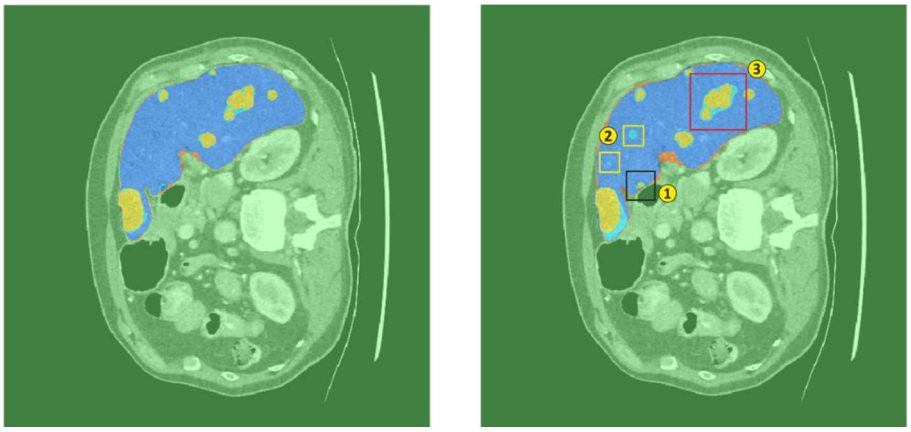


Figure 5.9 Comparison of Tumor Segmentation Dice Score from Net01 and Net02

An illustration of the segmentation result from Net01 and Net02 is given in figure 5.10. In figure 5.10(a), Net01 can detect and segment the tumor with very similar result to the ground truth. Meanwhile Net02 is more sensitive to the tumor where more tumors are found in Net02 segmentation. The tumor sensitivity is higher in Net02 due to the increment of the layer numbers in Net02 architecture. It gives more features and better recognition to the tumor than the Net01. Case 01 in the figure 5.10(b) shows an example where tumors that are missing from Net01 can be found and segmented in Net02. However, the increment in sensitivity does not always mean better performance for Net02. In case 02 that indicated by the yellow box, some tumors are misidentified by the network. In addition, it is found that Net 02 tend to exaggerate the tumor area as shown in case 3 with a red box. Therefore, high sensitivity helps Net02 to identify tumor better, but also increase the rate of the false positive tumor which in the end contribute to the lower dice score for the Net02 compare to Net 01.



a. *Net 01 Segmentation* *b. Net 02 Segmentation*

Figure 5.10 Net01 and Ne02 Segmentation

5.2.3. LiTS Challenge Leaderboard

Another test has been done to evaluate the performance of both network architecture that has been developed in section 5.2.2. The test is performed by participating in the LiTS competition which a part of ISBI 2017 and MICCAI 2017. In this test, the test dataset contains 70 patients from the LiTS competition which will be segmented using both networks. All segmentation results are stored in NIFTI files type and submitted to the LiTS website. The evaluation will be done on the server and the final score will be displayed in the online leaderboard. The original resolution, 512×512, is used during the process to maintain the tumor number and appearance in the patient data, where there is a possibility that tumors disappear if the image size reduced.

Table 5-1 Net01 and Net02 Segmentation Result on LiTS Test Dataset

Network Type	LIVER		LESION			
	Dice per Case	Jaccard	Dice per Case	Jaccard	Precision at 50% overlap	Recall at 50% overlap
Net01 - Encoder Decoder Network	0.914	0.846	0.562	0.619	0.102	0.348
Net02 - Densely Encoder Decoder Network	0.911	0.839	0.501	0.566	0.078	**0.465**

The evaluation score for both networks is shown in table 5.1. The whole result is quite similar to the result from the previous experiment in chapter 5.2.2. As examples, the liver dice score for both networks are slightly different and the dice score for lesion is higher in the Net01 than Net02. Although the performance of the networks is still low in term of the Dice score, Net02 successfully get the first rank in the online leaderboard for its tumor detection performance with achieving score 0.465. The leaderboard screenshot has been taken on 6 August 2018 and display in figure 5.11.

Lesion

#	User	Entries	Date of Last Entry	Dice per case ▲	Dice global ▲	VOE ▲	RVD ▲	ASSD ▲	MSD ▲	RMSD ▲	Precision at 50% overlap ▲	Recall at 50% overlap ▲	Precision at >0% overlap ▲	Recall at >0% overlap ▲
1	giriWardhana	4	07/07/18	0.5010 (29)	0.6530 (37)	0.434 (29)	-0.208 (5)	1.342 (25)	6.971 (17)	1.906 (23)	0.078 (34)	0.465 (1)		
2	mahendrakhened	10	11/17/17	0.5560 (25)	0.7240 (28)	0.435 (30)	7.179 (39)	1.460 (31)	7.914 (30)	2.108 (31)	0.083 (33)	0.446 (2)		
3	summer	10	05/08/18	0.6310 (12)	0.7860 (13)	0.400 (16)	-0.181 (9)	1.184 (12)	6.367 (8)	1.697 (11)	0.130 (25)	0.434 (3)		
4	MEDDIIR	7	12/29/17	0.6580 (10)	0.8190 (4)	0.380 (11)	-0.129 (14)	1.133 (8)	6.323 (7)	1.647 (8)	0.264 (17)	0.430 (4)		
5	mabc	7	05/16/18	0.6650 (5)	0.7720 (17)	0.408 (22)	-0.232 (4)	1.319 (22)	7.035 (18)	1.866 (21)	0.538 (1)	0.414 (5)		
6	mercury825	10	05/15/18	0.6630 (6)	0.7920 (9)	0.389 (13)	-0.173 (10)	1.190 (14)	6.498 (12)	1.708 (13)	0.230 (19)	0.413 (6)		
7	SurpathMedical	2	05/30/18	0.6600 (8)	0.8350 (1)	0.360 (5)	-0.057 (27)	1.023 (5)	6.409 (9)	1.548 (5)	0.336 (10)	0.410 (7)		
8	Yong	13	07/12/18	0.6610 (7)	0.8020 (8)	0.375 (10)	-0.007 (31)	0.986 (3)	6.493 (11)	1.534 (4)	0.143 (23)	0.406 (8)		
8	cxyjy	15	07/13/18	0.5920 (18)	0.7640 (21)	0.407 (21)	-0.106 (17)	1.209 (16)	7.385 (22)	1.809 (19)	0.093 (32)	0.406 (8)		
8	peiqi	11	08/07/18	0.6270 (13)	0.7840 (14)	0.402 (18)	-0.194 (7)	1.234 (18)	6.876 (14)	1.777 (18)	0.244 (18)	0.406 (8)		

Figure 5.11 LiTS Online Leaderboard

5.3. Manual Segmentation Experiment

The experiment has been done in the University of Twente, starting from 16 July to 6 August 2018. There were 11 participants who contributed to perform manual liver and tumor segmentation on giving images and share their opinion regarding the performance of the automatic method. In this section, three different parts are analyzed, including the development of Automatic Segmentation Graphical User Interface (GUI), the comparison between automatic segmentation and manual segmentation, and user evaluation of the automatic method performance.

5.3.1. GUI for Automatic Segmentation Method

To facilitate the user to employ the automatic method, the graphical user interface has been developed using Matlab GUI Development Environment (GUIDE). The GUI result is shown in figure 5.12, where the green number indicated the main functions and the orange letter show the supported part.

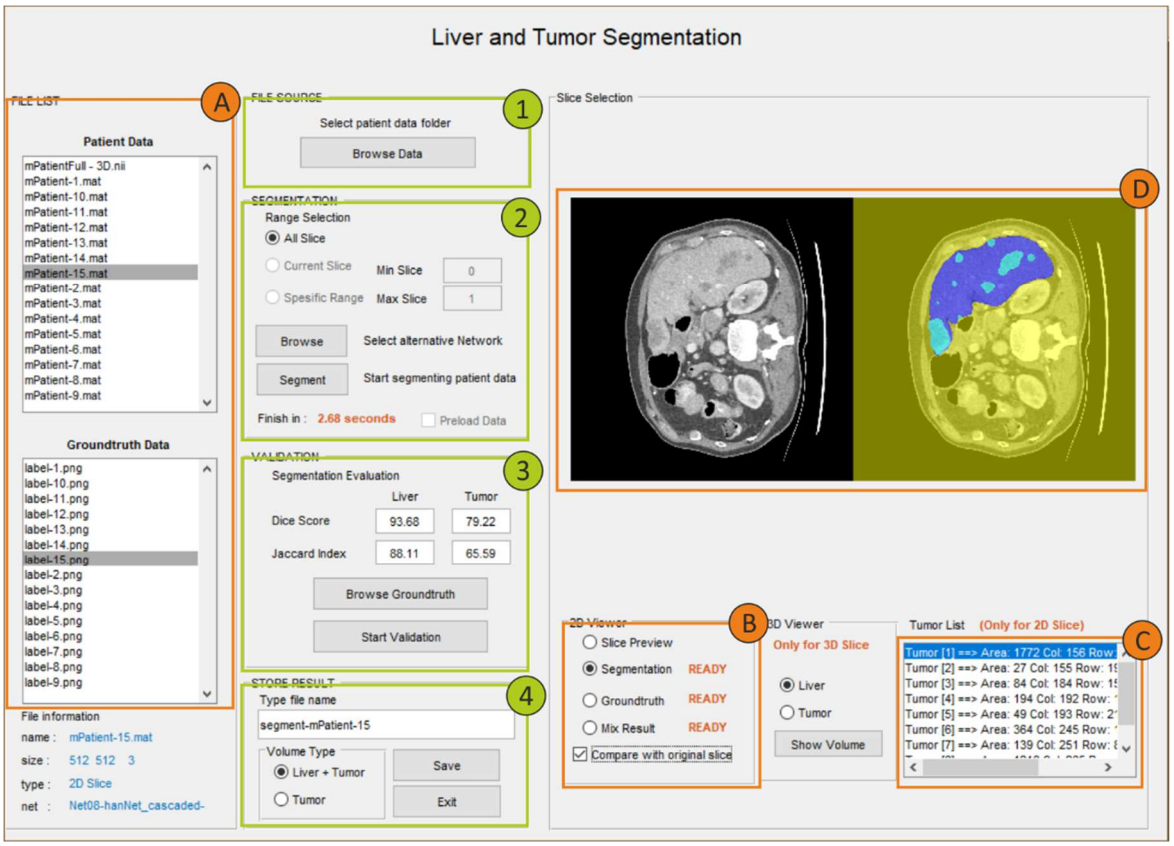


Figure 5.12 Graphical User Interface for Automatic Liver and Tumor Segmentation

The first function, marked by green number 1, is used to load the input images into the application. Two types of images, the 2D slice and 3D slice, are supported. A 2D slice can be a type of PNG or MAT-files and a 3D slice is a NIFTI file, a popular file types for the CT Scan image. After selecting the folder location, the list of supported files will be displayed in the Patient Data list which marked with the orange letter A.

Performing the segmentation of the liver and tumor is the main task of this program. The function that related to the segmentation process is put at the region of green number 2. The first feature in this function is the range selection. Selecting the range is not necessary if the input image is a 2D file type, since the segmentation will be done directly to the image slice or the center of the slice on the stacked slices. However, it is very useful when working with a 3D file type where sometimes the image contains hundreds of slices for one patient data. With this feature, the user can set the segmentation range, such as work for all slices, specific range or only in the current active slice. After setting the slice range, the segmentation can be started by pressing the Segment button. The processing time will be displayed under the Segment button. The segmentation result can be viewed by changing the 2D viewer into the Segmentation in the box of Orange B. Furthermore, the list of tumors that found in the segmentation result will be listed in the tumor list in a box of Orange C. In this list, other information that related to tumor such as location, area, longest diameter is included. Another important feature is the capability for the user to select the network model for the segmentation process. This feature is added to facilitate the improvement of the application performance in the future by providing a better network model.

The validation process is supported by the application, where the function is shown in the green box 3. To perform validation, the ground truth of the input image is required. User can provide this by selecting the directory where the ground truth can be found, and the list of ground truth will be updated in the Ground truth data at box Orange A. The ground truth data can be viewed by selecting it from the list and change the 2D viewer parameter at box Orange B into ‘Ground truth’. Additionally, the user can mix the result from the segmentation and the ground truth to provide a better view of the segmentation result by changing the parameter in box B into “Mix Result”. After the application recognize that the ground truth is already selected, another feature will be enabled which allow the user to compute the Dice score and Jaccard index. These two metrics are used to measure the similarity between the segmentation result and the provided ground truth.

The last function in this application is a function to store the segmentation result into files. The liver and tumor class have a pixel intensity of 1 and 2 while the background class has a value of 0. For a 2D slice, the segmentation will be saved into a grayscale image with PNG type. On the other hand, 3D slice segmentation will have two different outputs, the NIFTI file and the STL file. User can decide to store the volume of liver or tumor. The example result of liver and tumor segmentation volume are shown in figure 5.13.

Figure 5.13 Segmentation Result of (left) Liver Volume and (right) Tumor Volume

5.3.2. Comparison with Manual Segmentation

During the experiment, participants are asked to perform manual segmentation on liver and tumor region from 15 image slices. The experiment duration is varied among the participants, where it took from 30 to 50 minutes to segment all the images. The result from each participant and the automatic method later will be compared with the ground truth that is provided by the LiTS challenge. There are two evaluation metrics that are used, Dice score and Recall. The comparison result is shown in figure 5.14.

The first chart displays the evaluation of liver segmentation base on the dice score. All the manual results get very high score that more than 89%. It represents that all participant can recognize the liver inside the given image and give a good segmentation according to the ground truth data. This statement is also supported by the fact that all the manual liver segmentation from the participant achieve liver recall score more than 93%. Almost all the liver segmentation covers the whole liver region correctly. However, it can be seen from the chart too that the automatic method segmentation outperforms all the participant scores with a significant difference. Automatic method segmentation that utilizes Net01 and Net02 obtain score up to 95.3% and 94.6% while the manual segmentation only reaches a maximum score of 92.5%.

Not only in the liver case, but also in tumor case, tumor dice and tumor recall show a similar trend. In the tumor dice chart, an automatic method that using Net01 lead the scoreboard with result 78.4%, where the highest score from the manual segmentation gets 74.1%, 1% higher than Net02 with 73.1%. It is due to the ability of the automatic method to detect the tumor that has a quite similar intensity with the background, while it is almost impossible to detect by the human observation without extra processing. Moreover, the automatic method can distinguish the border shape better in blurry tumor case. Therefore, with the difference from these two methods up to 4%, it can be concluded that the automatic method can perform tumor segmentation better than the manual segmentation.

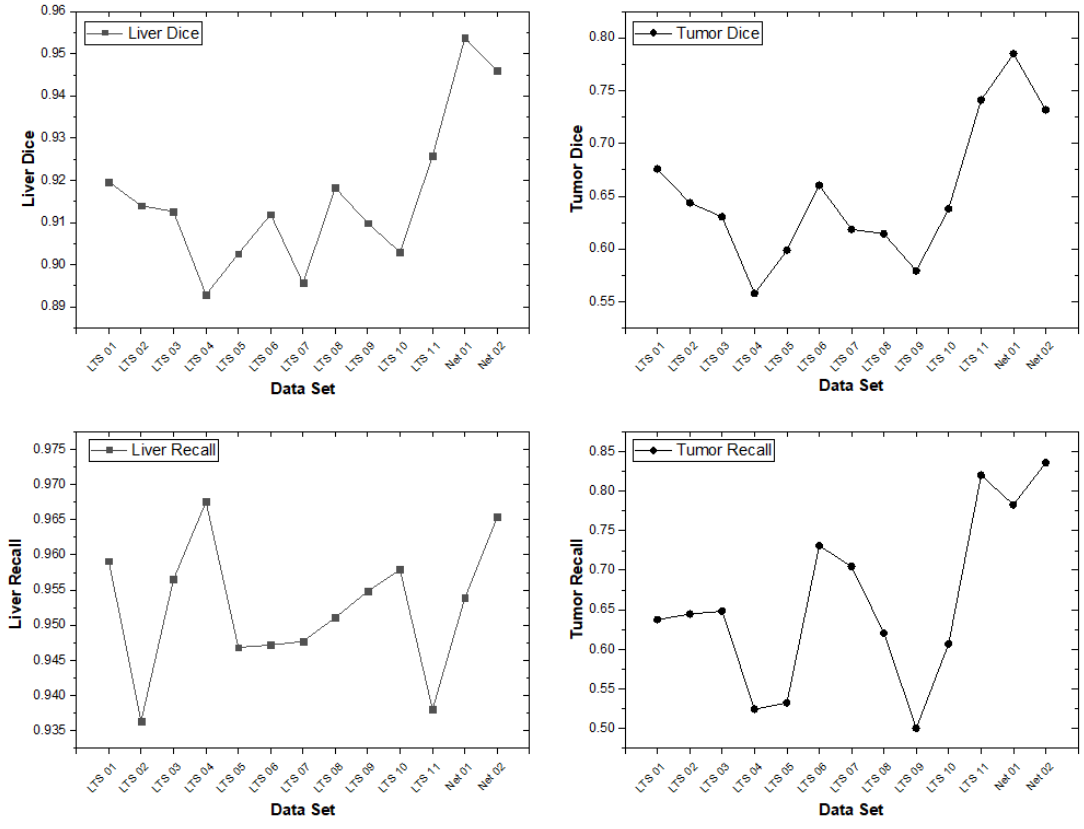


Figure 5.14 Dice Score and Recall Result from Manual and Automatic Segmentation

5.3.3. User Evaluation of Automatic Method Performance

The third part of the experiment is to evaluate the method performance based on the user experience. This is done by asking the participant to fill out a survey that is given to them after they finish doing manual and automatic segmentation. There are three aspects that asked in the survey. Those are related to the user satisfaction of the automatic software result, the information that essential to be included in the application, and the automatic method prospect in the medical area.

The degree of satisfaction is evaluated in several different categories as shown in figure 5.15. The user satisfaction is interpreted using the evaluation metric, such as very poor, poor, satisfied, good, and excellent. At segmentation category, most of the participant found that the automatic method can produce a good liver segmentation. While segmenting the tumor, the proposed method does a very satisfying performance. It can be seen by 18% participant mark the tumor segmentation with an excellent score. It is due to the ability of the automatic method to find the tumor that has a similar intensity to the background within the image, while most of the participant misses segmenting that tumor.

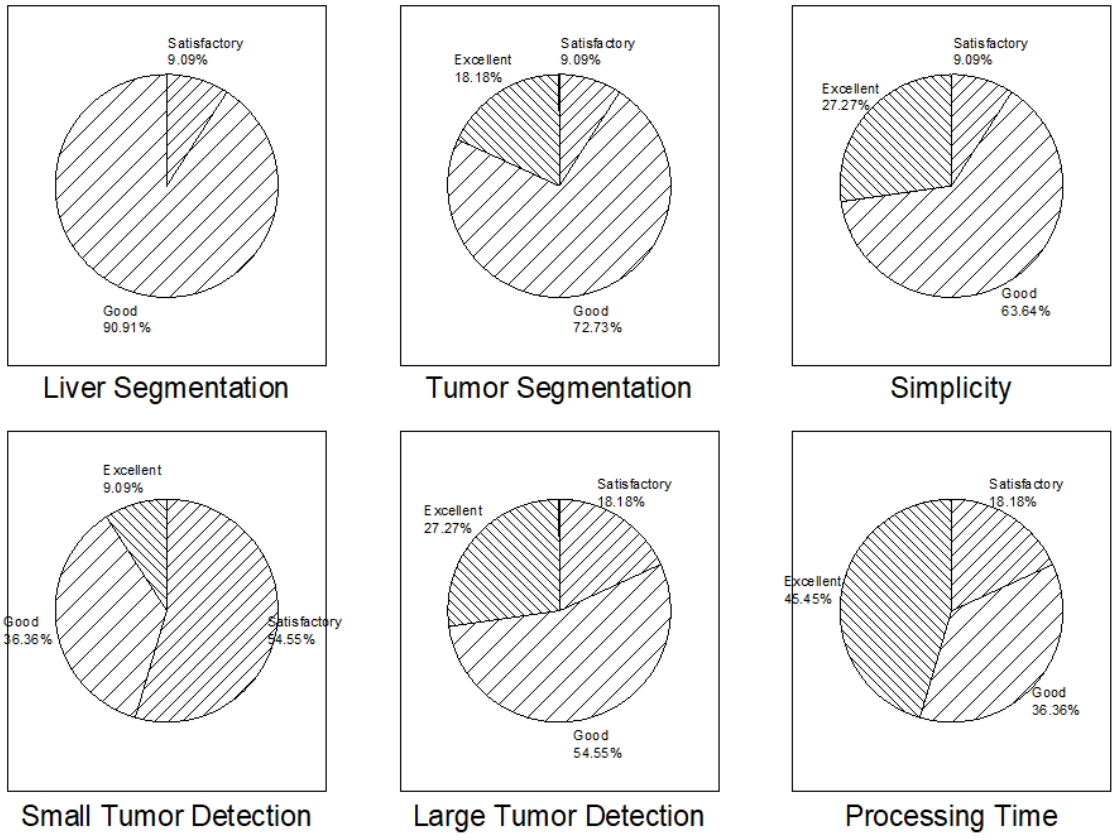


Figure 5.15 Degree of Satisfaction from Automatic Segmentation Performance

In the detection category, the participants are asked to rate the capability of the automatic method to detect the tumor. Tumors are separated into two groups, a small tumor that has a diameter less than 10 mm and the rest as a big tumor. According to the participant experience, most of them are very satisfied with the automatic segmentation performance when detecting the big tumor. However, in the case of a small tumor, the majority of the participant think the detection rate should be improved more due to the different perception among the participant on how to detect a small tumor. Figure 5.16 shows that some of the participants tend to detect a group of small tumors into one big tumor, while the other keep separating them.

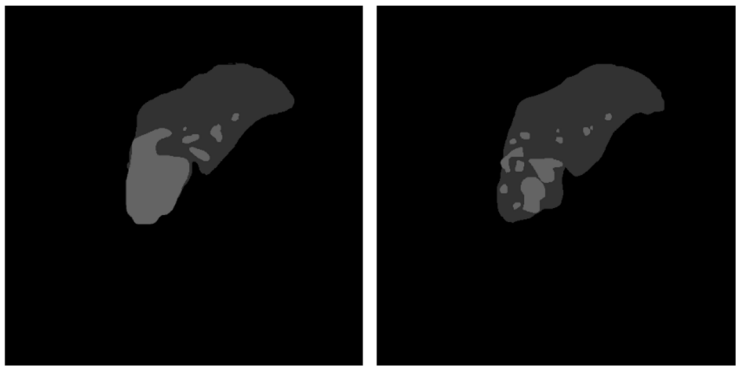


Figure 5.16 Two Different Perspective on How to Segment Small Tumor in the Manual Segmentation

When it comes to the simplicity, the majority of the users agree that the segmentation using the software is very simple. After selecting the image, the liver and tumor segmentation can be obtained immediately by pressing the segmentation button. There are no additional settings that need to be adjusted or selected by the user to perform the segmentation. It is also one of the advantages of using the neural network when developing the automatic method. Regarding the processing time, most of the participant are very satisfied with the method performance. While the manual segmentation requires around 2-3 minutes to separate the liver and tumor from the background, the automatic method can give the full segmentation in less than 3 seconds.

The next aspect is the kind of information needed by the medical staff from the segmentation result. Here, several kinds of information are listed as they also mentioned in the application. From this list, the participants are asked to rate, the information that they think useful for them. The result is displayed in the chart in figure 5.17. From this figure, more than 80% of participants agree that Tumor Diameter, Tumor Location needs to be included in the segmentation process. It can be explained since these are high-quality information that can provide useful insight for the clinician regarding the tumor in the patient body. Other information that beneficial and has rate higher than 50% are tumor volume and liver volume. Even though some of the participants also vote for time processing, the number of tumors and part of the liver where tumors are found, but it is not giving significance impact for clinician purpose.

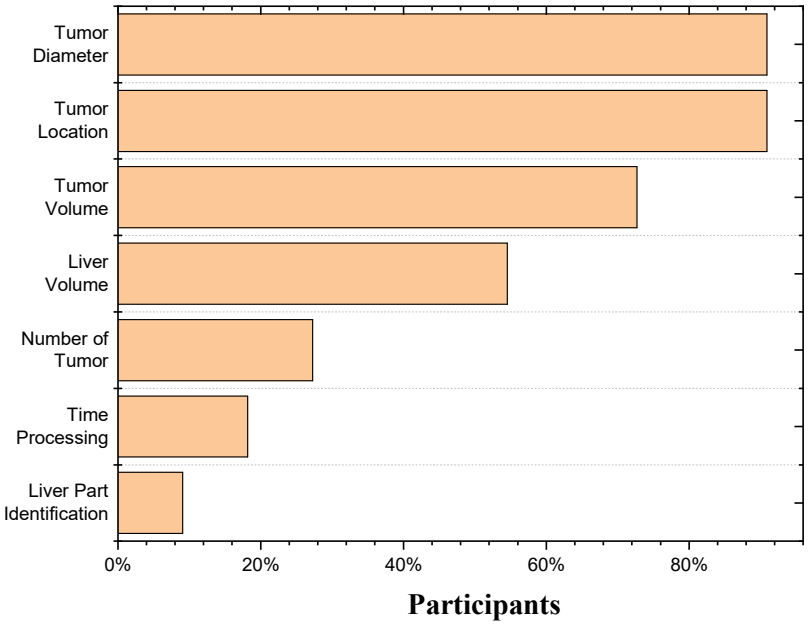


Figure 5.17 Relevant Information in the Segmentation Result

Another important aspect that needs to be discussed in this survey is how good is the prospect of this method for the clinician. There are two main categories, the application clarity and the impression from the participant. The application clarity is discussed since the application will be used along the practical work. It is essential to have an application that has clear instructions with a user-friendly interface. The application has been developed as simple as possible, to facilitate the participant for performing a segmentation. As can be seen in figure 5.18, the first statement that says the instruction is very clear for segmenting the liver and tumor receive a good result, where 27% of participants are strongly agree with the statement. Moreover, more than half of the participant has a high level of agreement regarding the user interface that very friendly. To start the segmentation process, user only need to click one segmentation button without the need to adjust other settings or parameters.

Although 9% of participants think that the application is unhelpful for the clinician, the rest found the result of the application can be useful for helping the clinician work. However, even though all previous statements receive a good response, but it does not mean that the application implementation will be easy. The number of participants that agree with the statement that says the application is easy to implement have the same number with participants that disagree with this statement. There are still many things that need to be considered before implementing the application in the real practice. In this case, most of the participant suggest that radiologists or people who mainly work with the CT image should be asked for their opinion. In addition, changing the flow of medical work by implementing the application is quite challenging. Providing a training for the clinician to adapt and learn how to use the application is one of the challenges that need to be addressed.

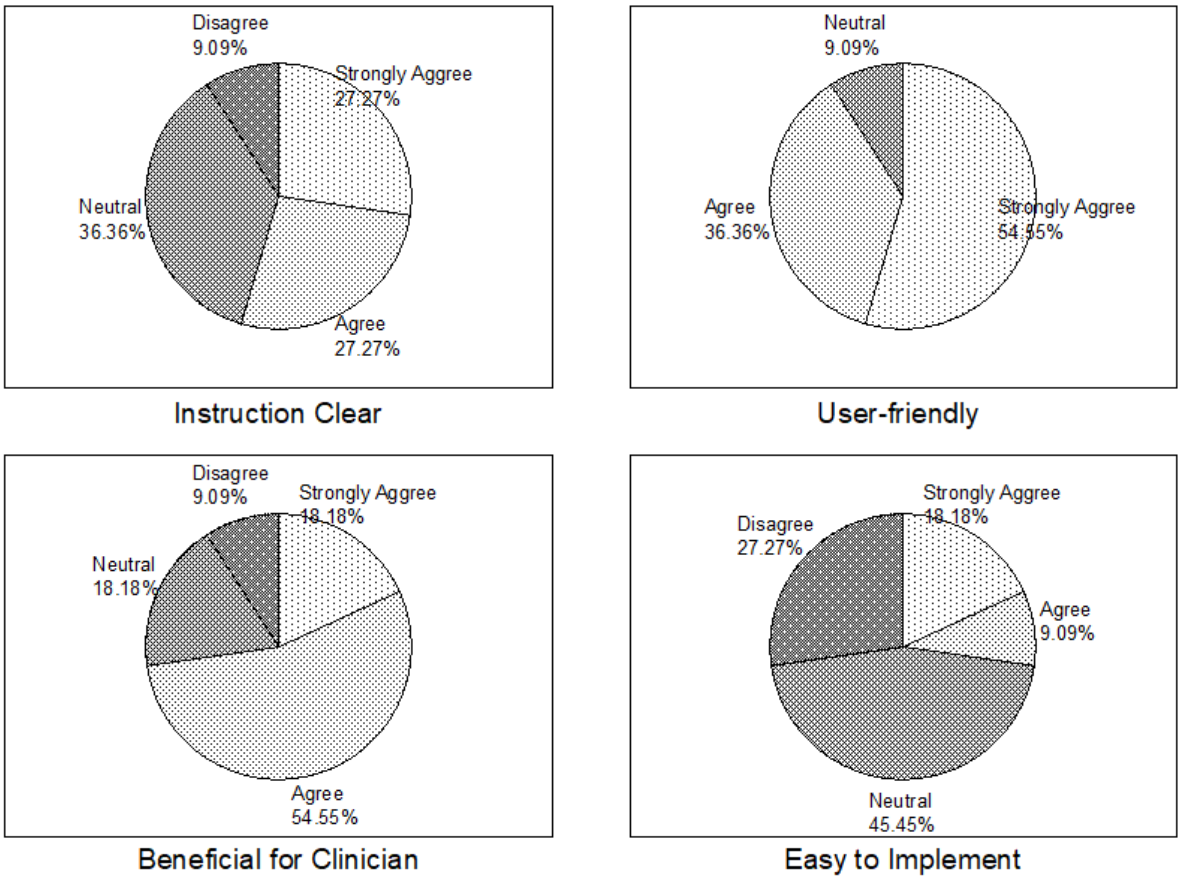


Figure 5.18 Degree of Agreement on Several Categories Related to Automatic Segmentation Prospect

6. Conclusion and Future Work

6.1. Conclusion

The automatic method for segmenting liver and tumor has been successfully developed using a deep convolutional neural network. The network utilizes the encoder and decoder structure from the SegNet with some modifications to improve its performance. To examine the performance improvement, several tests have been conducted related to the preparation on the dataset, the variation of configuration for training and the evaluation using manual segmentation.

How is the preparation of the dataset contributed to the segmentation accuracy?

In the data preparation test, two different factors are investigated, the slice arrangement and the image contrast. The slice arrangement defines the number of slices that are used for training the network. Instead of using a single slice, the stacked slice is introduced to provide 3D context from volume image. Using the dice metric, the network that trained using stacked slice achieve a higher score than the network with a single slice. However, there is a maximum number of slices that can be added. After passing this limit, the dice score stops showing an increasing trend and start decreasing gradually. The second factor that needs to be investigated is the influence of image contrast on the segmentation result. An opposite result has been observed by comparing two different enhancement techniques that based on pixel intensity and histogram data. The method that works based on the pixel intensity information shows a slight improvement to the dice score, while method based on image histogram lead to a sharp drop in the dice score value.

What is the best network architecture and setup that gives the best performance to the liver and tumor segmentation result? And how well do the Network perform compared to other segmentation results?

Different options for training the network are tested to check their influence on the segmentation result. The first option is the utilization of class balance in the output layer of the network. Class balance is used to compensate the effect of unbalance number of pixels among the segmentation class. The test result reveals that network without class balance ignore the tumor from the image and only recognize the background and liver area. At the same time, applying the class balance helps the network in recognizing the whole class in the image, so the liver and tumor can be segmented. The next test is the effect of changing the network structure and increasing the number of layers. The original network, called Net01 and the modified network, Net02, are registered to join the Liver and Tumor Segmentation (LiTS) Challenge. From the final score, it is found that the modification that has been made in Net02 increase the sensitivity of the network to find the tumor on the image. As a result, Net02 outperform other state of the art methods for detecting the tumor and get the first rank on the online LiTS leaderboard in Tumor Recall category.

Beside the result from the LiTS challenge, additional evaluation has been done by conducting an experiment that followed by 11 participants (an international radiologist and ten technical physicians). In this experiment, the participants are asked to perform manual and automatic segmentation. At the end of the experiment, the participants are requested to fill out a survey related to the performance evaluation of the automatic method. By comparing both segmentation result with the ground truth data, automatic segmentation method exceeds the manual segmentation result in term of segmentation accuracy and processing time. In addition, participants are very satisfied with the liver and tumor segmentation from the automatic method, while they also agree that the segmentation process using the automatic method are very simple and clear. However, to implement this method to the medical workflow is not an easy task. There are some considerable improvements that must be fulfilled to realize this. For instance, it is essential for the automatic method to get accuracy as high as possible and thus not miss or over-segmenting the lesion.

6.2. Recommendations

The development of an automatic method for segmenting liver and tumor is beneficial for clinicians. By using this method, the segmentation can be done faster while achieving a high accuracy result. However, further work is needed to improve the method performance. In term of speed, the implementation of liver detection can be an option to reduce the processing time during segmentation. Instead of working on all slices, the method should focus on the slices that include liver. Moreover, applying the parallel computation for the segmentation can increase the processing speed significantly. In term of accuracy, the post-processing for detecting the tumor can be developed to reduce the false positive tumor. Some tumor filtering techniques are shown in these studies, where Chlebus et al. [16] developed tumor filter based on the tumor shape information and Christ et al. [55] employed 3D conditional random field to refine the tumor segmentation. Another improvement that can be done is to develop a cascaded network where liver and tumor are segmented separately as seen in [18].

Further research on this topic could also be expanded to other topics, such as tumor classification and educational application that can be used to train technical medicine students for segmenting the liver and tumor. It can become an interesting topic and give a useful contribution to the medical world.

Bibliography

[1] T. Akinyemiju *et al.*, "The Burden of Primary Liver Cancer and Underlying Etiologies From 1990 to 2015 at the Global, Regional, and National Level," *JAMA Oncol.*, vol. 98121, no. 12, pp. 1683–1691, 2017.

[2] X. H. G. Tsechpenakis, "Medical Image Segmentation," *Adv. Mater. Res.*, no. i, pp. 1–35, 2013.

[3] A. Gotra *et al.*, "Liver segmentation: indications, techniques and future directions," *Insights Imaging*, vol. 8, no. 4, pp. 377–392, 2017.

[4] O. Ecabert *et al.*, "Automatic model-based segmentation of the heart in CT images," *IEEE Trans. Med. Imaging*, vol. 27, no. 9, pp. 1189–1202, 2008.

[5] L. Fernández-de-manuel *et al.*, "3D Liver Segmentation in Preoperative CT Images using a Level- Sets Active Surface Method," pp. 3625–3628, 2009.

[6] L. Ruskó, G. Bekes, G. Nemeth, and M. Fidrich, "Fully automatic liver segmentation for contrast-enhanced CT images," *MICCAI Wshp. 3D Segmentation Clin. A Gd. Chall.*, vol. 2, no. 7, pp. 1–8, 2007.

[7] R. Susomboon, "A hybrid approach for liver segmentation," *... Segmentation Clin. ...*, vol. i, pp. 151–160, 2007.

[8] D. Furukawa, A. Shimizu, and H. Kobatake, "Automatic Liver Segmentation Method based on Maximum A Posterior Probability Estimation and Level Set Method," *MICCAI - Work. 3D Segmentation Clin. a Gand Chall.*, pp. 117–124, 2007.

[9] Y. Chi, P. M. M. Cashman, F. Bello, and R. I. Kitney, "A Discussion on the Evaluation of A New Automatic Liver Volume Segmentation Method for Specified CT Image Datasets," *Heart*, pp. 167–175, 2007.

[10] D. Kainmüller, T. Lange, and H. Lamecker, "Shape constrained automatic segmentation of the liver based on a heuristic intensity model," *MICCAI Work. 3D Segmentation Clin. A Gd. Chall.*, pp. 109–16, 2007.

[11] D. Seghers *et al.*, "Landmark based liver segmentation using local shape and local intensity models," *Proc. Work. 10th Int. Conf. MICCAI, Work. 3D Segmentation Clin. A Gd. Chall.*, pp. 135–142, 2007.

[12] T. Heimann, H. Meinzer, and I. Wolf, "A Statistical Deformable Model for the Segmentation of Liver CT Volumes Using Extended Training Data," *Proc. MICCAI Work. 3-D Segmentat. Clin. A Gd. Challenge,* p. 161–166., 2007.

[13] Y. Yuan, "Hierarchical Convolutional-Deconvolutional Neural Networks for Automatic Liver and Tumor Segmentation," vol. i, pp. 3–6, 2017.

[14] P. F. Christ *et al.*, "Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks," pp. 1–20, 2017.

[15] S. Rafiei, E. Nasr-esfahani, and S. M. R. Soroushmehr, "LIVER SEGMENTATION IN CT IMAGES USING THREE DIMENSIONAL TO TWO DIMENSIONAL FULLY CONVOLUTIONAL NETWORK," pp. 1–5.

[16] G. Chlebus, A. Schenk, J. H. Moltz, H. K. Hahn, and H. Meine, "Deep learning based automatic liver tumor segmentation in CT with shape-based post-processing," no. Midl 2018, pp. 1–9.

[17] L. Bi, J. Kim, A. Kumar, and D. Feng, "Automatic Liver Lesion Detection using Cascaded Deep Residual Networks," 2017.

[18] X. Han, "Automatic Liver Lesion Segmentation Using A Deep Convolutional Neural Network Method," 2017.

[19] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P. A. Heng, "H-DenseUNet: Hybrid Densely Connected UNet for Liver and Liver Tumor Segmentation from CT Volumes," no. 1, pp. 1–10, 2017.

[20] A. Karpathy, "Convolutional Neural Networks for Visual Recognition," 2015. [Online]. Available: http://cs231n.github.io/convolutional-networks/. [Accessed: 16-Jul-2018].

[21] P. Kim, *MATLAB Deep Learning*. 2017.

[22] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," *J. Clin. Epidemiol.*, vol. 49, no. 11, pp. 1225–1231, 1996.

[23] M. M. Mijwel, "Artificial Neural Networks Advantages and Disadvantages," no. March, 2018.

[24] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst. 25*, pp. 1–9, 2012.

[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," pp. 1–14, 2014.

[26] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June-2015, pp. 1–9, 2015.

[27] K. He, "Deep Residual Learning for Image Recognition."

[28] A. Karpathy, "Neural Networks for Visual Recognition," 2015. [Online]. Available: http://cs231n.github.io/neural-networks-1/. [Accessed: 17-Jul-2018].

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 1026–1034, 2015.

[30] M. I. Razzak, S. Naz, and A. Zaib, "Deep Learning for Medical Image Processing: Overview, Challenges and Future," pp. 1–30, 2017.

[31] Cancer Council NSW, *Understanding Skin Cancer, A guide for people with cancer, their families and friends*, February 2. Sydney: SOS Print + Media Group, 2016.

[32] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[33] D. H. Murphree and C. Ngufor, "Transfer Learning for Melanoma Detection: Participation in ISIC 2017 Skin Lesion Classification Challenge," *arXiv*, pp. 0–2, 2017.

[34] Chandrakumar T and R Kathirvel, "Classifying Diabetic Retinopathy using Deep Learning Architecture," *Int. J. Eng. Res.*, vol. V5, no. 06, pp. 19–25, 2016.

[35] M. Haloi, "Improved Microaneurysm Detection using Deep Neural Networks," 2015.

[36] Y. Yang, T. Li, W. Li, H. Wu, W. Fan, and W. Zhang, "Lesion detection and Grading of Diabetic Retinopathy via Two-stages Deep Convolutional Neural Networks," pp. 1–8, 2017.

[37] M. B. Cotter and M. Loda, "Introduction to Histology," 2017.

[38] J. A. Quinn, R. Nakasi, P. K. B. Mugagga, P. Byanyima, W. Lubega, and A. Andama, "Deep

Convolutional Neural Networks for Microscopy-Based Point of Care Diagnostics," pp. 1–12, 2016.

[39] Y. Dong *et al.*, "Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells," *2017 IEEE EMBS Int. Conf. Biomed. Heal. Informatics*, pp. 101–104, 2017.

[40] X. Jia and M. Q.-H. Meng, "A deep convolutional neural network for bleeding detection in Wireless Capsule Endoscopy images," *2016 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pp. 639–642, 2016.

[41] G. B. Wimmer, S. Hegenbart, A. Vecsei, and A. Uhl, "Convolutional Neural Network Architectures for the Automated Diagnosis of Celiac Disease," *Int. Work. Comput. Robot. Endosc.*, vol. 10170, pp. 104–113, 2017.

[42] P. Christ, "LiTS - Liver Tumor Segmentation Challenge," 2017. [Online]. Available: https://competitions.codalab.org/competitions/17094. [Accessed: 02-Apr-2918].

[43] G. Hounsfield, "Computed Medical Imaging Nobel Lecture," *J. Radiol.*, vol. 61, no. 7, pp. 459–468, 1980.

[44] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang, "Deep Learning Markov Random Field for Semantic Segmentation," vol. 1, pp. 415–423, 2016.

[45] G. Chlebus, H. Meine, J. H. Moltz, and A. Schenk, "Neural Network-Based Automatic Liver Tumor Segmentation With Random Forest-Based Candidate Filtering," pp. 5–8, 2017.

[46] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[47] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," pp. 1–8.

[48] A. Prakash, "One by One [1 x 1] Convolution - counter-intuitively useful," 2016. [Online]. Available: https://iamaaditya.github.io/2016/03/one-by-one-convolution/. [Accessed: 20-Jul-2018].

[49] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.

[50] F. Schilling, "The Effect of Batch Normalization on Deep Convolutional Neural Networks," p. 113, 2016.

[51] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool," *BMC Med. Imaging*, vol. 15, no. 1, 2015.

[52] E. Vorontsov, A. Tang, C. Pal, and S. Kadoury, "Liver lesion segmentation informed by joint liver segmentation," 2017.

[53] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci. (Ny).*, vol. 250, pp. 113–141, 2013.

[54] Y. Wang, "DENSELY CONNECTED DECONVOLUTIONAL NETWORK FOR SEMANTIC SEGMENTATION Hanqing Lu National Laboratory of Pattern Recognition , Institute of Automation , Chinese Academy of Sciences , Beijing , China ; University of Chinese Academy of Sciences , Beijing , Chi."

[55] P. Christ *et al.*, "Automatic Liver and Lesion Segmentation in CT using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields," 2016.

Appendix: Manual Segmentation Experiment

Appendix 1. Participant Consent Form

CONSENT TO PARTICIPATE IN RESEARCH

Automatic Segmentation and 3D Reconstruction of Liver and Tumor

You are being invited to participate in a research study about liver and tumor segmentation. This research project is being conducted by Girindra Wardhana from the University of Twente as part of his thesis project.

Your participation in this study is entirely voluntary. You should read the information below and ask questions about anything you do not understand, before deciding whether or not to participate. You are being asked to participate in this study because you are a clinician or student who studies in the medical area.

- **PURPOSE OF THE STUDY**

The objectives of this research project are to make comparison among several manual segmentation and to measure the robustness of the segmentation result from proposed automatic segmentation method and the complexity of this method implementation in the medical area. We hope to use what we learn from the study to improve the performance of liver and tumor segmentation, so it will help clinicians doing their task.

- **PROCEDURES**

This experiment will take around 15-20 minutes. During this study, you will be asked to do the following things:

1. We will ask you to segment liver and tumor manually from 15 different slices of CT Scan image.
2. In the next session, you will be asked to segment liver and tumor using program that we developed.
3. At the end, we will ask you to fill out a questionnaire related to the performance of the program.

- **POTENTIAL RISKS AND DISCOMFORTS**

We expect that any risks, discomforts, or inconveniences will be minor, and we believe that they are not likely to happen. If discomforts become a problem, you may discontinue your participation.

- **POTENTIAL BENEFITS TO SUBJECTS AND/OR TO SOCIETY**

It is not likely that you will benefit directly from participation in this study, but the research should help us learn how to improve the process in medical image segmentation area.

This study does not include procedures that will improve your physical disability or general health.

- **COMPENSATION FOR PARTICIPATION**

You will not receive any payment or other compensation for participation in this study. There is also no cost to you for participation.

- **CONFIDENTIALITY**

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission or as required by law. Confidentiality will be maintained using a code number to let Girindra know who you are. We will not use your name in any of the information we get from this study or in any of the research reports. When the study is finished, we will destroy the list that shows which code number goes with your name.

Information that can identify you individually will not be released to anyone outside the study. Girindra will, however, use the information collected in his thesis and other publications. We also may use any information that we get from this study in any way we think is best for publication or education. Any information we use for publication will not identify you individually.

In case of an emergency, injury, or illness that occurs during this study, I hereby authorize the release of any health information to allow for medical care and treatment of my condition.

- **PARTICIPATION AND WITHDRAWAL**

You can choose whether or not to be in this study. If you volunteer to be in this study, you may withdraw at any time without consequences of any kind. You may also refuse to answer any questions you do not want to answer. There is no penalty if you withdraw from the study and you will not lose any benefits to which you are otherwise entitled. The investigator may withdraw you from this research if your physician tells us that continued participation may injure your health.

- **IDENTIFICATION OF INVESTIGATORS**

If you have any questions or concerns about the research, please feel free to contact
Girindra Wardhana at (31) 6307-98258 or at girindrawardhana@student.utwente.nl.

- **RIGHTS OF RESEARCH SUBJECTS**

If you have any concerns about your rights in this study, please contact
Dr.Ir. Momen Abayazid at +31 53 489 8980 or email m.abayazid@utwente.nl.

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

_____

Printed Name of Subject

_____ _____

Signature of Subject Date

Appendix 2. User Guide Manual Segmentation

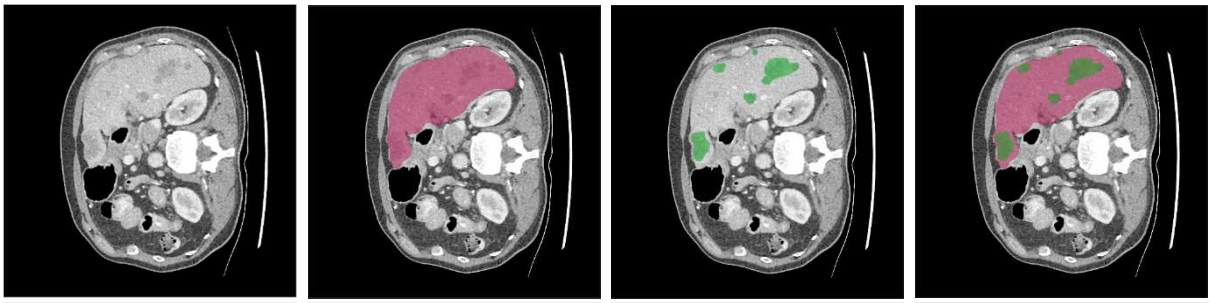
USER GUIDE

Manual liver and tumor segmentation

INFORMATION

In the first part, you will be asked to perform liver and tumor segmentation manually using adobe photoshop program. There are 15 slices of CT scan images from different patients that needs to be segmented which will take approximately 10-15 minutes.

INSTRUCTION



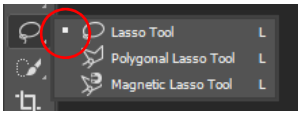
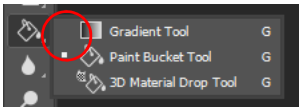
Step 1
Press "F2" to preprocessing the image

Step 2
Press "F3" to start segmenting the liver area (red color)

Step 3
Press "F4" to start segmenting the tumor area (green color)

Step 4
Press "F5" to preview the liver and tumor area and save the image

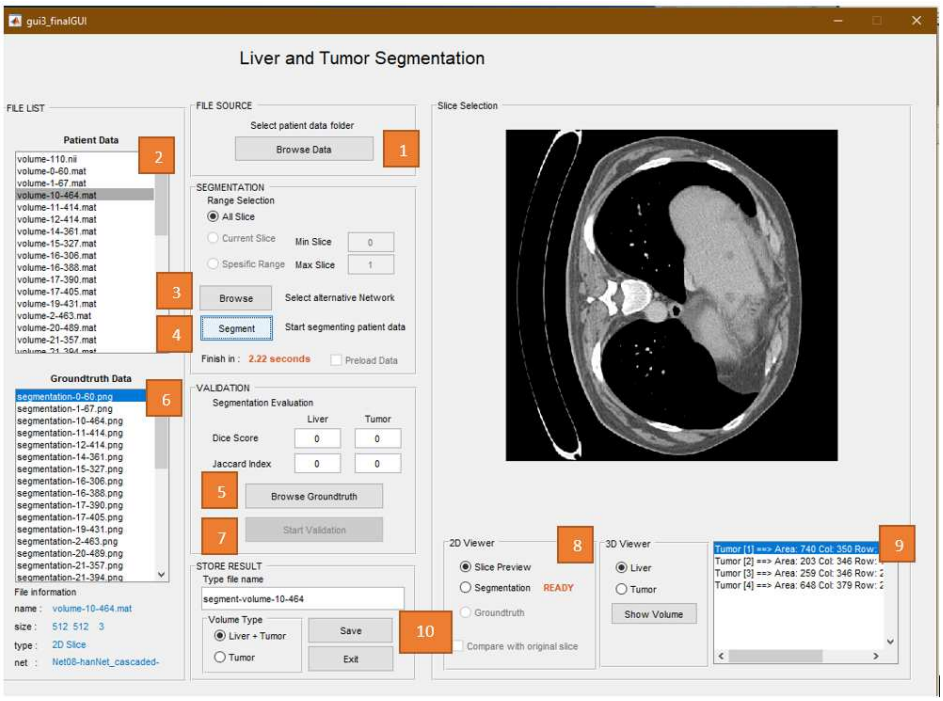
IMPORTANT SHORTCUT

Tools	Description	Shortcut
	Start area selection by clicking left mouse and drawing area in the image **Note on selecting multiple region:** - Press 'Shift' first then start drawing again - or, select region and fill the color then select a new region and fill the color again	L
	Fill the selection area with foreground color, red for liver and green for tumor	G
F2	Action Script: Preparing image before starting the segmentation process	F2
F3	Action Script: Start liver segmentation	F3
F4	Action Script: Start tumor segmentation	F4
Shift+	Action Script: Reset liver and tumor area	Shift + F5
F5	Action Script: Preview liver and tumor region and save the image	F5

Appendix 3. User Guide Automatic Segmentation

USER GUIDE

Automatic liver and tumor segmentation



INFORMATION

In the second part, you will be asked to segment liver and tumor automatically using application that we have developed. Image data is the same with the data from the first part. Additionally, you will be also asked to inspect the segmentation result carefully in order to complete the questionnaire which will be given in the third part of this experiment.

INSTRUCTION

- Select patient data by double clicking patient data in file list *(box 2)*
- Press "Segment" button *(box 4)* to start segmentation process
- Result can be checked by pressing "Segmentation" button in 2D viewer panel *(box 8)*
- To store result, in store panel, type the "file name" and press "Save" button *(box 10)*

ADDITIONAL INFORMATION

File Source Panel

[1] : Browse folder that contains patient data, it supports NIFTI file (*.nii) or Matlab file (*.mat)
[2] : Show the list of patient data

Segmentation Panel

[3] : Select alternative network that will be used for segmentation process
[4] : Start segmentation process of liver and tumor

Validation Panel

[5] : Browse folder that contain ground truth of liver and tumor
[6] : Show the list of ground truth data
[7] : Start validation process by calculating the Dice score and Jaccard index

Preview Panel

[8] : Show image in the frame, it supports slice preview, Segmentation and Ground truth
[9] : Show tumor list that detected in the image

Store Panel

[10] : Save the segmentation result where 2D slice as PNG data and 3D slice as NII and STL data

Appendix 4. Questionnaire Form

Evaluation form to measure the robustness and simplicity of the automatic liver and tumor segmentation application.

Page 1 of 2

Participant Number *

Your answer

Please tick the appropriate box to indicate your degree of satisfaction of automatic liver and tumor segmentation application *

	Excellent	Good	Satisfactory	Poor	Very poor
Segmentation result of liver area	○	○	○	○	○
Segmentation result of tumor area	○	○	○	○	○
Detection of small tumor (diameter < 10mm)	○	○	○	○	○
Detection of large tumor (diameter > 10mm)	○	○	○	○	○
Simplicity	○	○	○	○	○
Processing Time	○	○	○	○	○
Total Impression	○	○	○	○	○

Comment/Improvement for liver and tumor segmentation application

Your answer

Which kind of information do you require from liver and tumor segmentation application (You may choose more than one): *

- ☐ Liver Volume
- ☐ Tumor Volume
- ☐ Location of tumor in CT Slice
- ☐ Longest diameter of each tumor
- ☐ Time Processing
- ☐ Other:

Page 2 of 2

Please tick the approriate box to indicate your degree of application interface *

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Information for every component in the application is sufficient	○	○	○	○	○
Performing liver and tumor segmentation is very simple	○	○	○	○	○
Instruction to operate the application is clear	○	○	○	○	○
Application is useful for helping clinician work	○	○	○	○	○
Application can be easily implemented in the medical work	○	○	○	○	○

Remarks

Your answer

SUBMIT