



Sudan University of Science and Technology
College of Graduate Studies
Faculty of Computer Science and Information Technology,

A New Model to Enhance the Automatic Liver Segmentation based on U-net Architecture

نموذج جديد لتحسين التقسيم التلقائي للكبد بناءً على بنية U-net

BY:

Fatima Abdalbagi Mohammed

A dissertation submitted in Partial filament of the requirements for the degree of
Doctor of Philosophy

In

(Computer Science)

Supervisor: Prof. Serestina Viriri

March 2022

ABSTRACT

In computer vision, image segmentation is defined as the process of partitioning an image into several regions with homogeneous features. The region of our interest here in this thesis is the liver.

The main goal of the liver segmentation process is to divide the pixels of the medical image depending on specific criteria into two groups: pixels that belong to the object of interest (liver) and the rest of pixels that don't belong to the liver. It is an essential task in oncological therapy monitoring and radio-therapeutic treatment where tumor information is vital for correct dosimetry calculations.

Usually, the liver segmentation has been done manually by trained clinicians but it is time-consuming and requires much effort also different from one clinician to another because of the observer variability; as a result of that, an automatic liver segmentation system would thus be a great boon for performing these tasks. Because of the complexity of liver shapes and variable liver sizes among patients, the segmentation of the liver from medical images is very difficult and also due to low contrast between the liver and surrounding organs like the stomach, pancreas, kidney, and muscles.

Before the deep learning revolution, traditional handcrafted features were used for liver segmentation but with deep learning, the features are obtained automatically. There are numerous semi-automatic and fully automatic methodologies that have been proposed to improve liver segmentation some of them use deep learning techniques for segmentation and others use a classical based method for segmentation but still, there are no none of them achieve a hundred percent of accuracy.

In this thesis, we use the deep learning technique in particular U-net architecture to enhance the Automatic Liver Segmentation process. MICCA and 3D-IRCAD datasets are used to training and testing the model. The proposed Unet model, it was able to achieve the Dice similarity coefficient for MICCA dataset is equal to 0.97% and for a 3D-IRCAD dataset is equal to 0.96%.

مستخلص البحث

في مجال رؤية الكمبيوتر computer vision ، يتم تعريف تجزئة الصورة على أنها عملية لتقسيم الصورة في عدد من المناطق ذات الميزات المتجانسة. منطقة اهتمامنا هنا في هذه الأطروحة هي الكبد.

الهدف الرئيسي لعملية تجزئة الكبد هو تقسيم وحدات البكسل للصورة الطبية اعتماداً على معايير محددة إلى مجموعتين: وحدات البكسل التي تنتمي إلى موضوع الاهتمام (الكبد) وبقية وحدات البكسل التي لا تنتمي إلى الكبد. إنها مهام أساسية في مراقبة علاج الأورام والعلاج الإشعاعي حيث تكون معلومات الورم حيوية لحسابات قياس الجرعات الصحيحة.

عادة ، يتم إجراء تجزئة الكبد يدوياً بواسطة أطباء مدرّبين ، ولكنها تستغرق وقتاً طويلاً وتتطلب الكثير من الجهد ، كما أنها تختلف من طبيب إلى آخر بسبب تباين المراقب ؛ ونتيجة لذلك ، فإن نظام تجزئة الكبد التلقائي سيكون بمثابة نعمة كبيرة لأداء هذه المهام. بسبب تعقيد أشكال الكبد وتغير أحجام الكبد بين المرضى ، فإن تجزئة الكبد من الصور الطبية أمر صعب للغاية وأيضاً بسبب قلة التباين بين الكبد والأعضاء المحيطة مثل المعدة والبنكرياس والكلية والعضلات

قبل ثورة التعلم العميق ، تم استخدام الميزات التقليدية المصنوعة يدوياً لتجزئة الكبد ولكن مع التعلم العميق ، يتم الحصول على الميزات تلقائياً. هناك العديد من المنهجيات شبه الأوتوماتيكية والتلقائية بالكامل التي تم اقتراحها لتحسين تجزئة الكبد ، بعضها يستخدم تقنيات التعلم العميق للتجزئة والبعض الآخر يستخدم أسلوباً تقليدياً للتجزئة ولكن لا يزال لا يوجد أي منها يحقق مائة بالمائة من صحة.

في هذه الأطروحة ، نستخدم تقنية التعلم العميق، بنية U-net على وجه الخصوص لتحسين عملية الانقسام التلقائي للكبد. نستخدم مجموعات البيانات MICCA و 3D-IRCAD لتدريب النموذج واختباره. معامل تشابه النرد (Dice) لنموذج U-net المقترح كان قادراً على تحقيق معامل تشابه للنرد 0.97% عند استخدام مجموعة بيانات MICCA وتحصل على 0.96% عند استخدام مجموعة بيانات 3D-IRCAD.

TABLE OF CONTENTS

ABSTRACT.....	I
مستخلص البحث	II
TABLE OF CONTENTS.....	III
LIST OF TABLES	VI
LIST OF FIGURES	VII
LIST OF PUBLICATION	VIII
CHAPTER 1.....	1
INTRODUCTION	1
1.1 BACKGROUND	1
1.2 PROBLEM STATEMENT	4
1.3 RESEARCH OBJECTIVES.....	5
1.4 MOTIVATION	5
1.5 CONTRIBUTIONS OF THE RESEARCH	6
1.6 RESEARCH ORGANIZATION.....	7
CHAPTER 2.....	8
DEEP LEARNING	8
2.1 INTRODUCTION	8
2.2 MACHINE LEARNING	9

2.3	DEEP LEARNING	9
2.3.1	CONVOLUTIONAL NEURAL NETWORK.....	12
2.3.1.1	CONVNET ARCHITECTURE.....	12
2.4	EVALUATION METRIC.....	17
CHAPTER 3.....		19
LITERATURE REVIEW		19
3.1	INTRODUCTION	19
3.2	CLICAL BACKGROUND ON LIVER	20
3.3	CT IMAGE.....	21
3.4	LIVER SEGMENTATION METHOD	23
3.5	SEMIAUTOMATIC AND FULLY AUTOMATIC APPROACHES....	25
3.5.1	CLASSICAL METHOD FOR LIVER SEGMENTATION	25
3.5.2	DEEP LEARNING TECHNIQUES FOR LIVER SEGMENTATION	31
CHAPTER 4.....		38
PROPOSED MODEL FOR LIVER SEGMENTATION		38
4.1	INTRODUCTION	38
4.4	U-NET	39
4.5	METHODOLOGY	43
4.5.1	PRE-PROCESSING	45
4.5.2	THE ARCHITECTURE OF MODEL.....	46

4.5.3	POST-PROCESSING.....	48
	CHAPTER 5.....	51
	RESULTS AND DISCUSSION.....	51
5.1	INTRODUCTION.....	51
5.2	EXPERIMENTAL SETUP.....	51
5.3	PROPOSED UNET MODEL.....	51
5.3.1	TRAINING AND VALIDATION.....	51
5.3.2	TESTING PROPOSED UNET MODEL.....	53
	CHAPTER 6.....	60
	CONCLUSION AND FUTURE WORK.....	60
6.1	INTRODUCTION.....	60
6.2	SUMMARY.....	60
6.3	CONCLUSION.....	62
6.4	FUTURE WORK.....	62
	REFERENCES.....	63
	APPENDIX A - RESEARCH CODES.....	67

LIST OF TABLES

Table 3. 1: Classical Based methods for liver Segmentation	29
Table 3. 2: Deep learning methods for liver Segmentation	36
Table 5.1: Results of proposed Unet Model for MICCAI dataset.....	53
Table 5. 2: Results of proposed Unet model for 3D-IRCAD dataset.....	55
Table 5. 3: Comparison between proposed Unet model and some of related works.....	57

LIST OF FIGURES

Figure2. 1: Relation between the AI, ML and DL.....	8
Figure2. 2: Representations Learned By Deep-Learning	10
Figure2. 3: How Deep Learning Work	11
Figure2. 4: ConvNet Architecture	13
Figure2. 5: Convnet Operations.....	13
Figure2. 6: 2D convolution process on input volume (7x7x3).....	14
Figure2. 7: the Zero-padding process	15
Figure2. 8: The Pooling layer	16
Figure3. 1: Human Liver Anatomy.....	20
Figure3. 2: Liver in abdominal CT scan	22
Figure3. 3: Liver Segmentation Methods Categorizing.....	24
Figure 4. 1: example of obtained segmentation result by (Ronneberger et al., 2015).....	40
Figure 4. 2: U-net Architecture (Ronneberger et al., 2015).....	41
Figure 4. 3: Detailed architecture (Lamba, 2019).....	42
Figure 4. 4: The flowchart of our proposed Unet model.	44
Figure 4. 5: Overview of the applied CT Windowing	45
Figure 4. 6: Architecture of proposed Unet Model.....	47
Figure 4. 7: Show result of the thresholding.....	48
Figure 4. 8: Show the result of the LCCS process.....	49
Figure 4. 9: Show the result of the Holes Filling process.....	50

LIST OF PUBLICATION

- [1] F. A. Mohammed and S. Viriri, “Liver segmentation: A survey of the state-of-The-Art,” in *Proceedings of: 2017 Sudan Conference on Computer Science and Information Technology, SCCSIT 2017*, 2018, vol. 2017–Novem.
- [2] F. Abdalbagi, S. Viriri, and M. T. Mohammed, “Bata-Unet: Deep Learning Model for Liver Segmentation,” *Signal Image Process. An Int. J.*, vol. 11, no. 5, pp. 75–87, 2020.
- [3] F. Abdalbagi, S. Viriri, and M. T. Mohammed, “Batch Normalized Convolution Neural Network For Liver Segmentation” *Signal Image Process. An Int. J.*, vol. 11, no. 5, pp. 75–87, 2020.

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

In the computer vision field, images are considered as one of the most important of moderate carrying information. Extracted information from these images can be used for many tasks for example identification of particular features, recognition of objects, detection of cancerous cells, finding injurious tissues from body scans, or segmentation. Image segmentation is a method used to understand an image and extract information or objects (Agrawal, 2014) and also could be defined as the process of a partition of an image in several regions with homogeneous features. These regions might be a set of border pixels grouped into such structures or can be defined as groups of pixels having both a border and a particular shape. These regions are named Regions of Interest (ROIs) (Pratt, 2000). Every pixel in an image is allocated to one of these regions. A good segmentation is typically one in which the pixels in the same regions have similar greyscale of multivariate values and form a connected region, and neighboring pixels which are in different regions have dissimilar values.(Glasbey, C.A. and Horgan, 1995)

Medical images are considered as a set of techniques, processes of creating visual representations (images) inside a body for clinical analysis and medical intervention. Medical imaging is today an invaluable tool for diagnosis and treatment planning. Imaging modalities such as X-ray, Ultrasound (US), and Single-photon emission computed tomography (SPECT), Positron emission tomography (PET), computed tomography (CT), and magnetic resonance imaging (MRI). Almost all specialties of a modern hospital handled some kind of image on your clinical routine. The increasing introduction of digital imaging modalities allows efficient storage thereof and what is more important, the possibility of image processing and analysis to obtain quantitative data from them (P, 2015).

Modern surgeries rely on Computer-Aided Diagnosis (CAD) systems to assist doctors in the diagnosis of medical images, surgical planning, and simulation. CAD has become a part of the routine clinical work in medical imaging and diagnostic radiology therefore it has become one of the major research subjects (Fujita *et al.*, 2008). Because any successful treatment depends on preoperative such as planning to understand the complex internal structure of the liver and precisely localize the liver surface and its segments, the tumors, the topography of blood vessels to do that an automatic Liver segmentation is required which aim to divide the pixels (voxel) of the image depending on certain criteria into two groups: voxels that belong to the object of interest (liver) and the rest of voxels that don't belong to the liver. Liver pathologies such as cirrhosis, liver cancer, and fulminant hepatic failure can be diagnosed by using Medical imaging which is a non-invasive technique.

The diagnosticians prefer the CT images because of their higher signal to noise ratio, better spatial resolution and they provide more accurate anatomical information about the visualized structures (Campadelli, Casiraghi and Esposito, 2009). Computed Tomography (CT) was developed by British engineers named Sir Godfrey Hounsfield and Dr. Alan Cormack and has since had a profound effect on medicine. CT uses sophisticated X-ray technology to aid in the detection of a variety of diseases and conditions and is fast, painless, non-invasive, and accurate (Lawrence M. Davis, 2019) . CT scanning employs numerous X-ray beams and a set of electronic X-ray detectors which rotate around the patient, measuring the amount of radiation being absorbed throughout the body. A large volume of data is processed and creates 2D cross-sectional images of the body referred to as slices which are reassembled by computer software to produce a detailed multidimensional view of the body's anatomy.

Accurate, efficient, and automatic methods for liver segmentation are demanded because traditionally the radiologists and physicians have to manually delineate the liver region slice by slice, which is tedious and time-consuming due to a large amount of data (Nakayama *et al.*, 2006). There are several clinical applications for automatic liver segmentation such as measuring the graft volume before living donor liver transplantations(Hermoye *et al.*, 2005) , locating vessels and tumors, it also useful to monitor patients with liver metastases, where the disease is related to an enlargement of the liver, and it plays an important role in surgery planning where it is the most time-

consuming step (Heimann, Wolf and Meinzer, 2006). Information about precise size for liver, localize the tumors and accurate liver surface segmentation required in liver local treatment (Massoptier and Casciaro, 2008)

In the last few decades, a large variety of semiautomatic and fully automatic approaches have been proposed to improve the liver segmentation procedure, such as region growing, clustering, deformable models or level sets, statistical shape models (SSMs), probabilistic atlases, graph cuts and recently, deep convolution neural networks (Nakayama *et al.*, 2006) (Litjens *et al.*, 2017). Because of the complexity of liver shapes and variable liver sizes among patients, the segmentation of the liver from medical images is very difficult and also due to low contrast between the liver and the surrounding organs like stomach, pancreas, kidney ,and muscles (Zhang *et al.*, 2010). Moreover, challenge is the presence large tumors and other liver pathologies because the livers with pathologies are different from healthy ones and that may result either under-segmentation or over-segmentation.

Recently, deep learning has achieved state-of-the-art performance in medical image analysis (Lakhani *et al.*, 2018). Deep learning techniques have been applied to medical image analysis to let computers learn the features that optimally represent the problem data at hand (Litjens *et al.*, 2017). The main applications that deep learning techniques are used for are classification, detection ,and segmentation (Mazurowski and Buda, 2018). Within classification, deep learning techniques have been used for image classification where an image is taken as input and a single output is produced stating which class the image belongs to. Another application is object or lesion classification. The use of deep learning for object, region, and landmark localization in CAD, entails finding a region, object, or lesion of interest, which forms a key part of a diagnosis. Segmentation of an organ or its substructures allows for quantitative analysis which may lead to empirical medical findings to the organ of interest. An example of this is the segmentation of lesions which play an important role in the diagnosis and prognosis of diseases and abnormalities. Other applications are image registration, content-based image retrieval, image generation, and enhancement as well as combining image data with reports.

The organ of interest in this research work is the liver. The focus is on segmenting the liver from abdominal CT scans. Segmentation of the liver from these CT scans plays an important role in the study of liver functions and can assist in the diagnosis of liver diseases. Accurately detecting and delineating the liver separates the liver from its surrounding organs and isolates it for intricate analysis (Gotra *et al.*, 2017). Segmenting the liver may reduce the computation time required in the analysis as the liver only occupies a portion of the abdominal CT scan. Accurate liver segmentation ensures that the whole liver is analyzed and the surrounding organs and tissues are eliminated from the region of interest. Inspired by the use of deep learning and its success in organ segmentation, two automatic segmentation models using convolutional neural networks for liver segmentation are proposed in this research work. Wikipedia defines the convolutional neural network as: "a convolutional neural network (CNN or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery". They have applications in image and video recognition, recommender systems, image classification, medical image analysis, natural language processing, and financial time series.

1.2 PROBLEM STATEMENT

Liver segmentation is an essential task in oncological therapy monitoring and radio therapeutic treatment where tumor information is vital for correct dosimetry calculations (Wong *et al.*, 2008). Typically, this has been manually done by trained clinicians but it is time-consuming and requires much effort, and also it subjective because of observer variability; as a result of that, an automatic liver segmentation system would thus be a great boon for performing these tasks. The existence of tumors in the liver poses further difficulty. The tumor should be segmented as part of the liver and show affected parts and healthy parts of the liver. Each method has challenges; either depends on training examples which could be computationally expensive or depending on the experience level of the users (Peng, Zhang and Yang, 2010) which could be observer variability. Until now there is still a gap in the process of liver segmentation because none of the current research work achieves a hundred percent of accuracy. Bridging this gap is an open research problem.

1.3 RESEARCH OBJECTIVES

The main objective of this work is to propose a model for liver segmentation using deep learning techniques in particular U-net architecture to enhance the accuracy of Automatic Liver Segmentation process. Deep learning techniques have gained significant attention for solving computer vision tasks such as object recognition, classification, and segmentation (Long, Shelhamer and Darrell, 2015) The proposed method is considered as a binary labelling problem which assigning the labels to each voxel in the original image into either belonging to the liver or not, and it will be carried out by using a convolutional neural network; resulting in a probability map which it identifying the initial liver surface. Therefore, the particular objectives pursued in this work include the following:

1. To conduct a thorough survey of the state-of-the-art of liver segmentation.
2. To build a model for automatic liver segmentation using deep learning.
3. To compare the results of the model with other similar model.

1.4 MOTIVATION

Despite the vast amount of research available on the techniques explored to achieve accurate liver segmentation, it seems challenging to develop methods that will achieve higher accuracy rates. Many successful methods have been proposed however; they have their advantages as well as disadvantages. Different types of techniques exist such as Classical-based methods and learning-based methods. Classical-based methods which include region growing, graph-cut and level set methods, they have a drawback in that they need user interaction, more refinement, and may be sensitive to initial contours (Hu et al., 2016a). It stem from the fact that they are mainly based on gray-level information.

learning-based methods which include CNNs. Examining the learning-based methods, either a series of 2D slices of the CT scan or a while 3D CT scan volume is fed as input into the neural network models in current methods. Generally, they are lacking in the size of the dataset and some techniques fail to segment special cases due to the complex liver shape, so there are needs for more accurate model to enhance the accuracy of liver segmentation process.

1.5 CONTRIBUTIONS OF THE RESEARCH

The main contributions of this dissertation are the implementation of two models for liver segmentation with two different datasets. These contributions are described as follows:

- We introduce an enhance Unet model for automatic segmentation of the liver using CT scans on MICCAI and 3Dircadb datasets. This was achieved by using network 19 convolution layers, 18 BatchNormalization layers, and 4 max-pooling layers, 4 concatenate layers, 4 dropout layers, and 4 Transposed convolution layers (Deconvolution layer) as well as a softmax layer.
- We show a quantitative assessment of the segmentation results based on metrics: Dice Similarity Coefficient (**Dice**), Volume Overlap Error (**VOE**), Relative Volume Difference (**RVD**), Average Symmetric Surface Distance (**ASD**), And Maximum Surface Distance (**MSD**)
- This thesis provides a comparative analysis between the proposed Unet-model with some other work results.

1.6 RESEARCH ORGANIZATION

The organization of the following chapters in this research is as follows:

Chapter 2: Deep learning: presents definitions of the deep learning and convolutional neural networks (convnet).

Chapter 3: Literature Review: This chapter investigates background clinical background on the liver and related work to the current methods of liver segmentation.

Chapter 4: Liver Segmentation Model: presents the idea and implementation of the proposed model and the details of it.

Chapter 5: Results And Discussion: presents and discuss the research results of the model and compare it with some other related works.

Chapter 6: Conclusion and Future Work

CHAPTER 2

DEEP LEARNING

2.1 INTRODUCTION

Deep learning is a specific subfield of machine learning. In this chapter, we define deep learning and how it works and defines what is the Convolutional neural networks (ConvNets). Deep Learning is a new area of Machine Learning research, as seen in Figure 2.1 that explains the relationship between artificial intelligence, machine learning, and deep learning (Chollet, 2011).

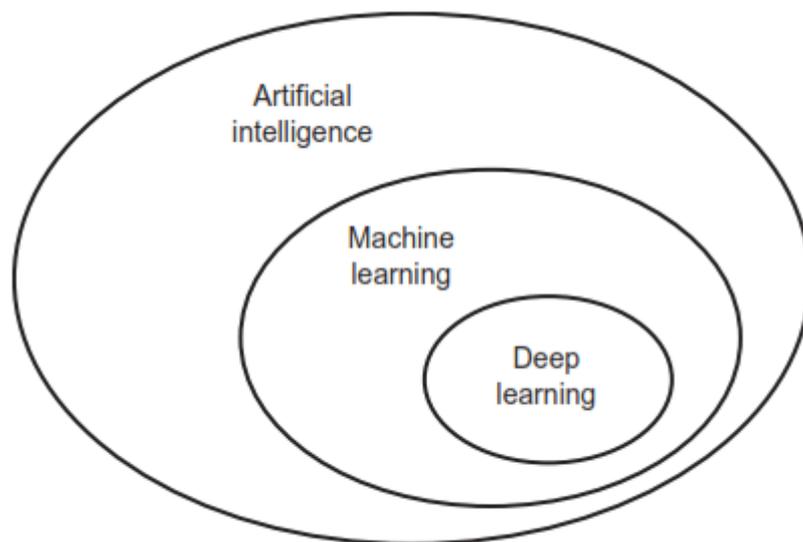


Figure2. 1: Relation between the AI, ML and DL

2.2 MACHINE LEARNING

Machine learning is aim to discover rules to execute a data-processing task, it transforms its input data into meaningful outputs, a process that is “learned” from publicity to known samples of inputs and outputs. Consequently, the central problem in machine learning and deep learning is to meaningfully transform data: to learn useful representations of the input data at hand, these representations get us closer to the predicted output. All Machine-learning models are focused on data transformation which is finding appropriate representation for input data.

2.3 DEEP LEARNING

Deep learning is a new subfield of machine learning that admits to learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations. There is no longer a need for the term “just like our minds” which hypothetical links between deep learning and biology. Deep learning is a mathematical framework for learning representations from data (Chollet, 2011).

Liu, Tianyi also define Deep learning as it refers to a subfield of machine learning that is based on learning levels of representations, corresponding to a hierarchy of features, factors, or concepts, where higher-level concepts are defined from lower-level ones, and the same lower-lever concepts can help to define many higher-lever concepts. Deep learning is learning multiple levels of representation and abstraction, which helps to understand the data such as images, audio, and text. The concept of Deep Learning comes from the study of Artificial Neural networks; Multilayer Perceptron which contains more hidden layers is a Deep Learning structure (Liu et al., 2015)

Deep learning is a type of machine learning in which a model learns to perform classification tasks directly from images, text, or sound, it is usually implemented using neural network architecture, and the term “deep” refers to the number of layers in the network. Traditional neural networks contain only 2 or 3 layers, while deep networks can have hundreds (Mathworks, 2017).

The representations learned by a deep-learning algorithm look like in Figure 2.2, which explains how the layers represent the image. The network transforms each image into representations that are increasingly different from the original image and increasingly informative about the final result. So, we can think of a deep network as a multistage information distillation operation, where information goes through successive filters and comes out increasingly purified. So deep learning is a multistage way to learn data representations.

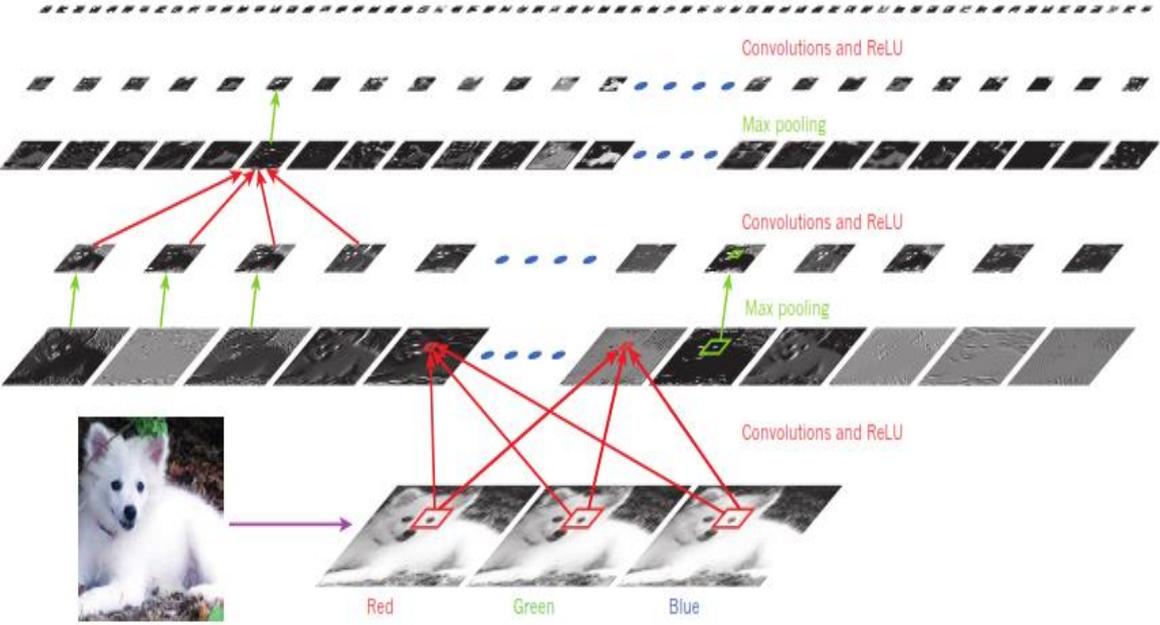


Figure2. 2: Representations Learned By Deep-Learning

To explain how deep learning works, we can visualize it as seen in Figure 2.3, which is composed of layers that are chained together which maps the input data to predictions. Then, the loss function compares these predictions to the targets, producing a loss value: a measure of how well the network’s predictions match what was expected. The optimizer uses this loss value to update the network’s weights; which is called the Backpropagation algorithm: the central algorithm in deep learning.

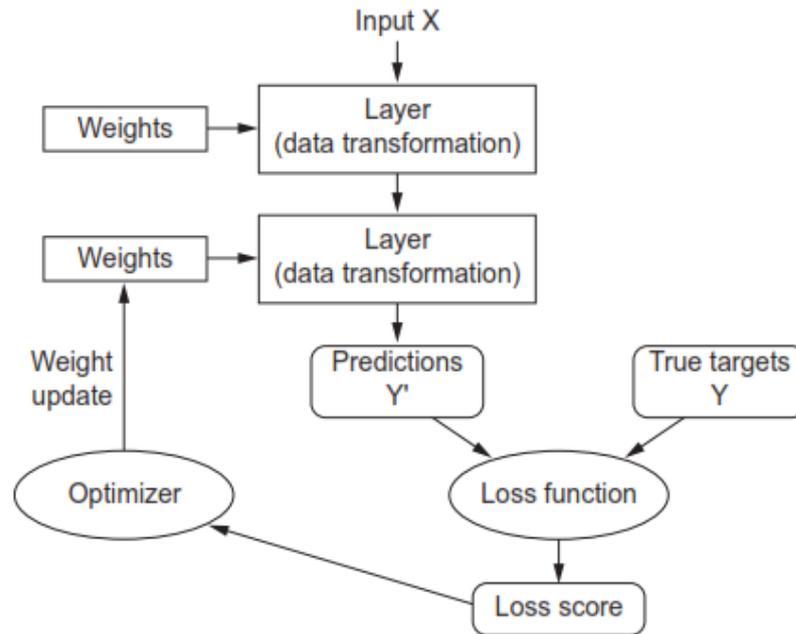


Figure2. 3: How Deep Learning Work

Deep learning has several properties that justify its status as an AI revolution. These important properties can be:

1. **Simplicity:** Deep learning removes the need for feature engineering, replacing complex, brittle, engineering-heavy pipelines with simple, end-to-end trainable models that are typically built using only five or six different operations
2. **Scalability:** Deep learning is highly amenable to parallelization on GPUs or TPUs. In addition, deep-learning models trained by iterating over small batches of data, allowing them to be trained on datasets of arbitrary size.
3. **Versatility and reusability:** deep-learning models can train on additional data without restarting from scratch, making them viable for continuous online learning. Furthermore, trained deep-learning models are re-purposable and thus reusable: for instance, it is possible to take a deep-learning model trained for image classification and drop it into a video processing pipeline. This allows us to reinvest previous work into increasingly

2.3.1 CONVOLUTIONAL NEURAL NETWORK

Convolutional networks are also known as convolutional neural networks (convnet or CNNs) are a specialized kind of neural network for processing data that has a known, grid-like topology. Examples include image data, which can be thought of as a 2D grid of pixels, and time-series data, which can be thought of as a 1D grid taking samples at regular time intervals. The name "convolutional neural network" indicates that the network employs a mathematical operation called "convolution" which is a specialized kind of linear operation. So we can define the Convolutional networks as are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers. Convolutional networks have been tremendously successful in practical applications and have shown excellent performance in many computer vision, machine learning, and pattern recognition problems (Goodfellow Ian and Courville Aaron, 2019). (Mathworks, 2017), they define the convolutional neural network (ConvNet) as one of the most popular algorithms for deep learning with images and video. Like other neural networks, it's composed of an input layer, an output layer, and many hidden layers in between.

2.3.1.1 CONVNET ARCHITECTURE

ConvNets are designed to process data that come in the form of multiple arrays, for example, a colour image composed of three 2D arrays containing pixel intensities in the three colour channels. Many data modalities are in the form of multiple arrays: 1D for signals and sequences, including language; 2D for images or audio spectrograms; and 3D for video or volumetric images. There are four key ideas behind ConvNets that take advantage of the properties of natural signals: local connections, shared weights, pooling, and the use of many layers. (Lecun, Bengio and Hinton, 2015)

A Convolutional Neural Network (ConvNet) architecture shown in Figure 2.4, it typically has convolutional layers combined with pooling (or sub-sampling) layers and then followed by fully connected layers as in a standard. is designed to better utilize such spatial and configuration information by taking 2D or 3D images as input (Hung-II, 2017).

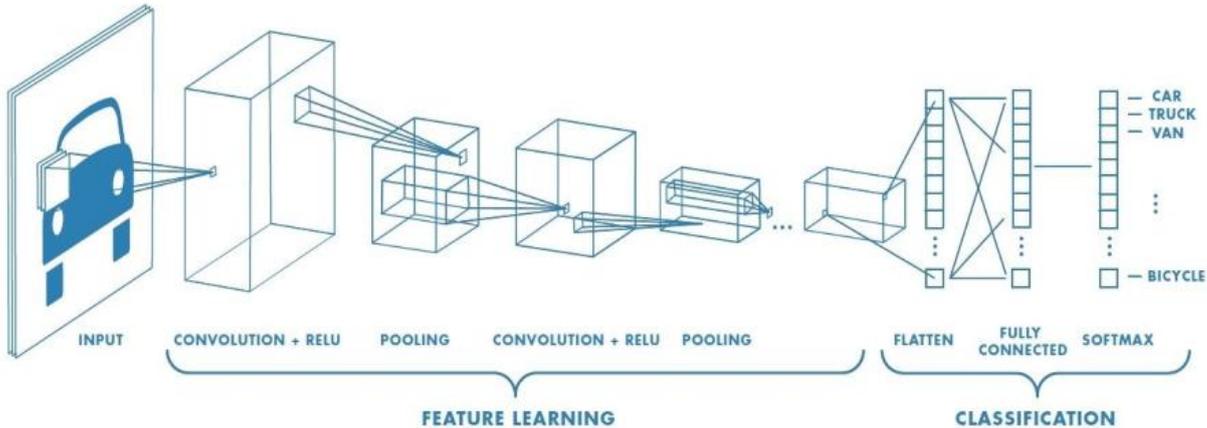


Figure2. 4: ConvNet Architecture

Convnet has two main operations: convolution operation and sampling operation as shown in Figure 2.5. The Convolution operation: use a trainable filter F_x , deconvolution of the input image to output with the feature image of each layer, namely *Feature Map*, then it adds a bias b_x , we can get convolution layer C_x . A sampling operation: n pixels of each neighborhood through pooling steps, become a pixel, and then by scalar weighting W_{x+1} weighted, add bias b_{x+1} , and then by an activation function, produce a narrow n times feature map S_{x+1} (Liu *et al.*, 2015).

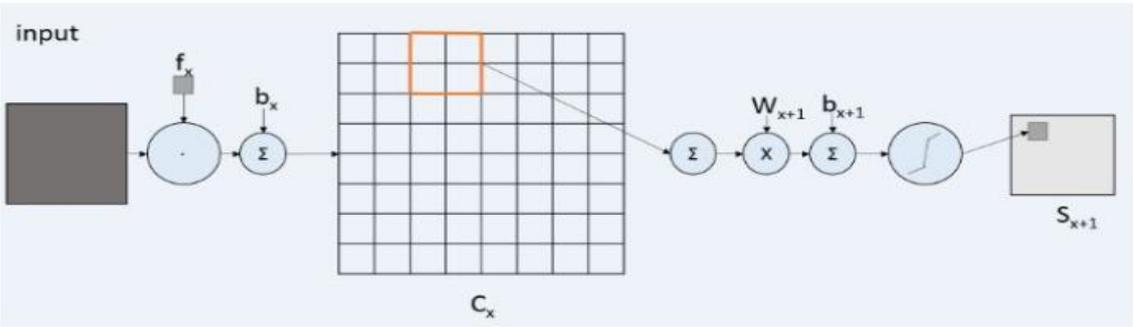


Figure2. 5: Convnet Operations

(Mathworks, 2017), they divide the layers of ConvNet into two types: Feature Detection Layers and Classification Layers. The Feature Detection Layers are performing one of three types of operations on the data: convolution, pooling, or rectified linear unit (ReLU). These three operations are repeated over tens or hundreds of layers, with each layer learning to detect different features.

1- The convolution layer computes the output of neurons that are connected to local regions or receptive fields in the input, each computing a dot product between their weights and a small receptive field to which they are connected to in the input volume (Albawi, Mohammed and Al-Zawi, 2018). Each computation leads to extraction of a feature map from the input image, as shown in Figure2. 6.

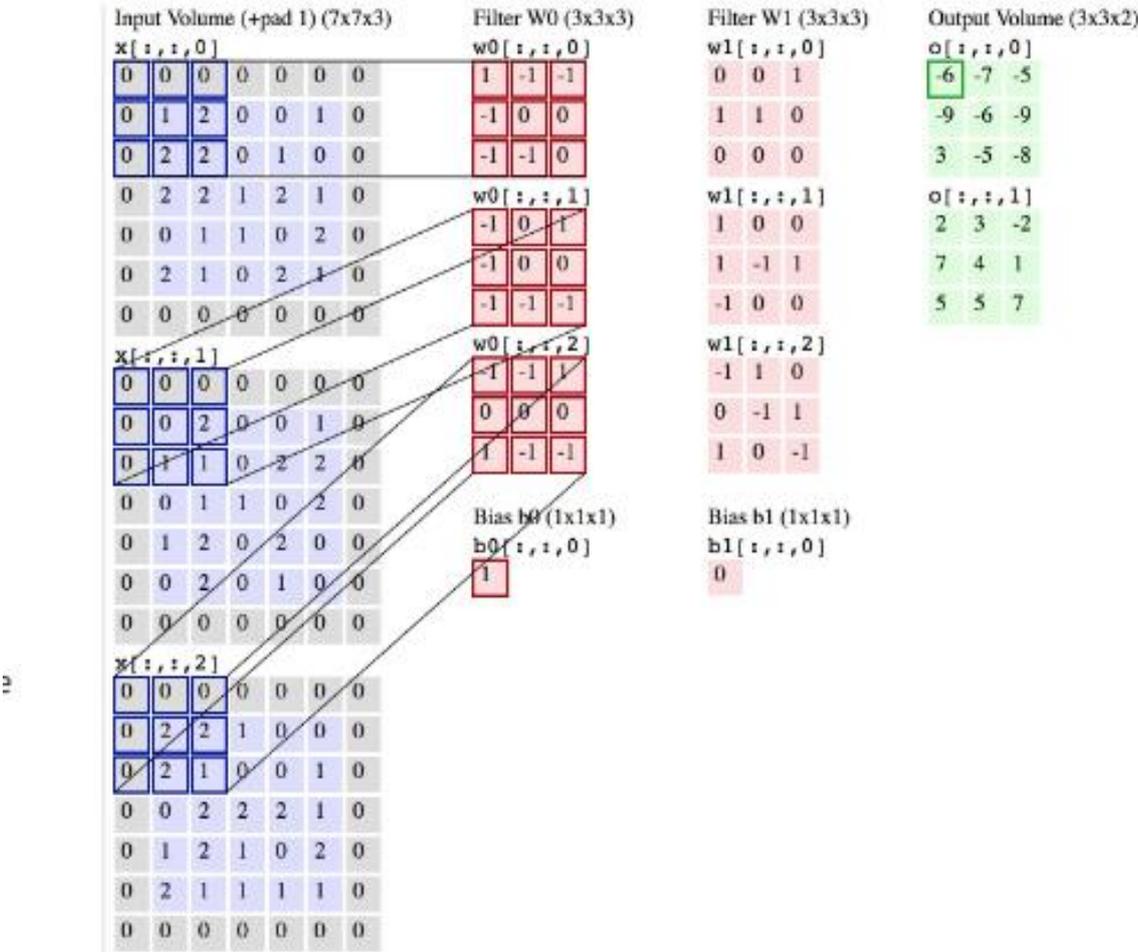


Figure2. 6: 2D convolution process on input volume (7x7x3)

In other words, visualise you have an image represented as a 5x5 matrix of values, and you take a 3x3 matrix and slide that 3x3 window or kernel around the image. At each position of that matrix, you multiply the values of your 3x3 window by the values in the image that are currently being covered by the window. As a result, we'll get a single number that represents all the values in that window of the images. You use this layer to filtering: as the window moves over the image, you check for patterns in that section of the image. This works because of filters, which are multiplied by the values outputted by the convolution.

One of the drawbacks of the convolution step is the loss of information that might exist on the border of the image. Because they are only captured when the filter slides, they never have the chance to be seen. A very simple, yet efficient method to resolve the issue is to use zero-padding. The other benefit of zero padding is to manage the output size. For example, in Figure2. 7, with $N=7$ and $F=3$ and stride 1, the output will be 5×5 which shrinks from a 7×7 input. (Albawi, Mohammed and Al-Zawi, 2018).

0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

Figure2. 7: the Zero-padding process

2- The Pooling layer simplifies the output by performing nonlinear downsampling to reducing the number of parameters that the network needs to learn about. The objective of subsampling is to get an input representation by reducing its dimensions, which helps in reducing overfitting. One of the techniques of subsampling is max pooling. With this technique, we select the highest pixel value from a region depending on its size. In other words, max-pooling takes the largest value from the window of the image currently covered by the kernel. For example, you can have a max-pooling layer of size 2 x 2 that will select the maximum pixel intensity valuze from the 2 x 2 region as shown Figure 2.8. The only difference between the pooling layer and the convolution layer is the function that is applied to the kernel and the image window isn't linear.

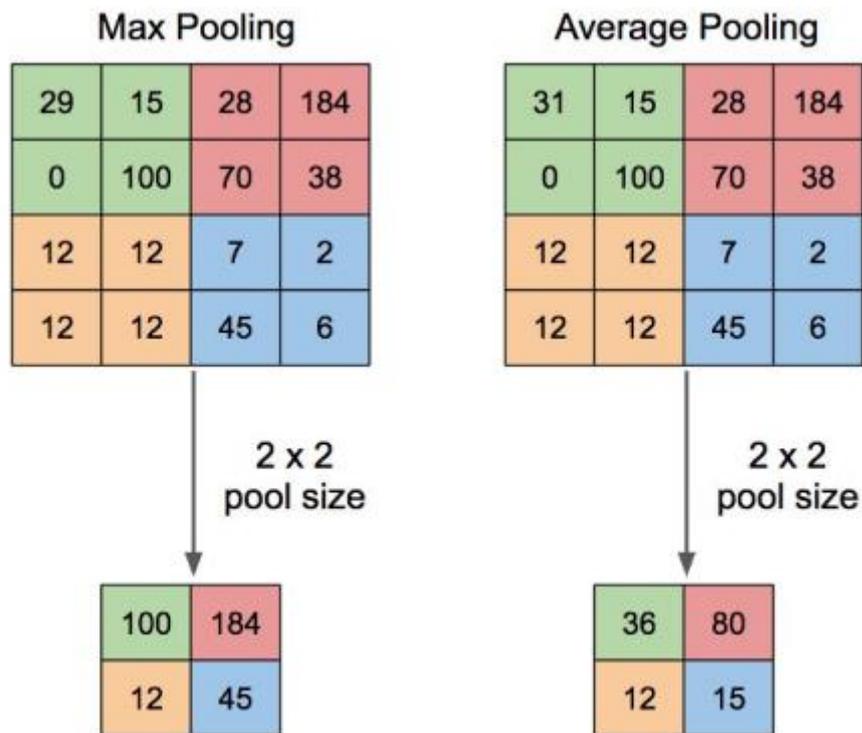


Figure2. 8: The Pooling layer

3- Rectified linear unit (ReLU) allows for faster and more effective training by mapping negative values to zero and maintaining positive values.

After feature detection, the architecture of a CNN shifts to classification. The next-to-last layer is a fully connected layer (FC) that outputs a vector of K dimensions where K is the number of classes that the network will be able to predict. This vector contains the probabilities for each class of any image being classified. The final layer of the CNN architecture uses a softmax function to provide the classification output.

The objective of the fully connected layer is to flatten the high-level features that are learned by convolutional layers and combining all the features. It passes the flattened output to the output layer where you use a softmax classifier or a sigmoid to predict the input class label.

2.4 EVALUATION METRIC

The two models are evaluated during training and after training. During the training, training and validation errors (loss) are used to evaluate the network. After training the model, the networks are used to segment the liver. Thereafter, the resulting segmented images are evaluated using some popular and well-approved evaluation metrics in the field of medical image processing including the Dice Similarity Coefficient (**Dice**), Volume Overlap Error (**VOE**), Relative Volume Difference (**RVD**), Average Symmetric Surface Distance (**ASD**) And Maximum Surface Distance (**MSD**) (Seo *et al.*, 2020).

Dice Similarity Coefficient (Dice): is a statistic that measures the similarity and overlap between two samples A and B. where A is stand for the segmentation results achieved. And B stands for the ground truth segmentation. The performance index ranges from zero to one with an index zero signifying no overlap between A and B while index one signifies a perfect overlap between them. Equation (2.1) defines Dice.

$$\text{Dice} = 2 \frac{B \cap A}{|B| + |A|} \quad (2.1)$$

Volume Overlap Error (VOE): VOE is a type of error measurement usually expressed in percentage. VOE is calculated as the ratio of the intersecting area between the segmentation volume and the ground truth volume as shown in Equation (2.2). This calculates the percentage of regions where the two volumes do not overlap with each other. A perfect segmentation produces an overlap error percentage of 0.

$$VOE = 100 \left(1 - \frac{A \cap B}{A \cup B} \right) \quad (2.2)$$

Relative Volume Difference (RVD): the relative volume difference is the percentage of the difference between the segmentation volume and the ground truth volume. The value is in the range of [-1, +1]. The perfect segmentation produces a 0 percent. The Negative values suggest that the segmentation result is smaller than the ground truth which denotes under-segmentation whereas positive values suggest that the segmentation result is larger than the ground truth denoting over-segmentation. it is calculated by using Equation (2.3).

$$RVD = 100 \left(\frac{|A| - |B|}{|B|} \right) \quad (2.3)$$

Average Symmetric Surface Distance (ASD): The average symmetric surface distance is calculated in millimeters using Equation (2.4). For a perfect segmentation, this distance is 0. Let us consider $S(A)$ as a set of surface voxels of A and $S(B)$ as a set of surface voxels of B. The shortest distance of a voxel in $S(B)$ to $S(A)$ is defined as $d(S; S(A))$ and a voxel in $S(A)$ to $S(B)$ is defined as $d(S; S(B))$.

$$ASD = \frac{1}{|S(A)| + |S(B)|} \sum_{S_A \in S(A)} d(S_A, S(B)) + \sum_{S_B \in S(B)} d(S_B, S(A)) \quad (2.4)$$

Maximum Surface Distance (MSD): MSD is the surface-based evaluation method and it's calculated between the segmentation volume and the ground truth volume in millimeters as in Equation (2.5). Unlike average in ASD, in MSD the maximum distance is taken. Let us consume that the A is the segmented volume, B is the ground truth volume, $S(A)$ is the set of surface voxels of A, $S(B)$ is the set of surface voxels of B. The distance of a voxel in $S(B)$ to $S(A)$ is defined as $d(SB; S(A))$ and the distance of a voxel in $S(A)$ to $S(B)$ is defined as $d(SA; S(B))$. For a perfect segmentation, this distance is 0.

$$MSD = \max_{S_A \in S(A)} (S_A, S(B)), \max_{S_B \in S(B)} d(S_B, S(A)) \quad (2.5)$$

CHAPTER 3

LITERATURE REVIEW

3.1 INTRODUCTION

This research focuses on the localization and segmentation of the liver in abdominal CT scans using deep learning through the implementation of two convolutional neural networks. Various methods have been proposed to achieve liver segmentation however; many of the existing techniques have some kind of limitation (Lu et al., 2016). There are interactive methods, semi-automatic methods as well as automatic methods. Interactive methods are dependent on user input to perform liver segmentation whereas automatic methods can perform liver segmentation independently of user input.

Liver segmentation methods can be categorized as anti-learning-based methods, classical learning-based methods, and deep learning methods. In the most recent work done in liver segmentation, the use of deep learning is very popular for feature extraction (Litjens et al., 2017). To gain a better understanding of how to choose an efficient technique for liver segmentation, it is important to have some knowledge about the liver anatomy and its appearance in CT scans. It is also important to conduct a study of related literature. In this chapter, the anatomy of the liver and the abdominal CT scan are briefly introduced, related literature regarding liver segmentation, and the various methods of liver segmentation were discussed.

3.2 CLICAL BACKGROUND ON LIVER

The liver is one of the largest and vital organs in the human body. It is shaped like a cone and is located in the upper right-hand portion of the abdominal cavity, below the diaphragm, and above the stomach, right kidney, and intestines. More than 500 bodily functions have been associated with the liver and some of them include regulation of most chemical levels in the blood, excretion of bile and filtering blood that comes from the digestive tract(*Johns Hopkins Medicine, 2019*).

The liver consists of four distinct lobes; the left, right, caudate, and quadrate lobes. The left and right lobes are the largest lobes and the right lobe is about five to six times larger than the tapered left lobe. The small caudate lobe extends from the posterior side of the right lobe and wraps around the inferior vena cava while the small quadrate lobe is inferior to the caudate lobe and extends from the posterior side of the right lobe and wraps around the gallbladder (Barclay, 2019). The liver anatomy is illustrated in Figure 3.1.

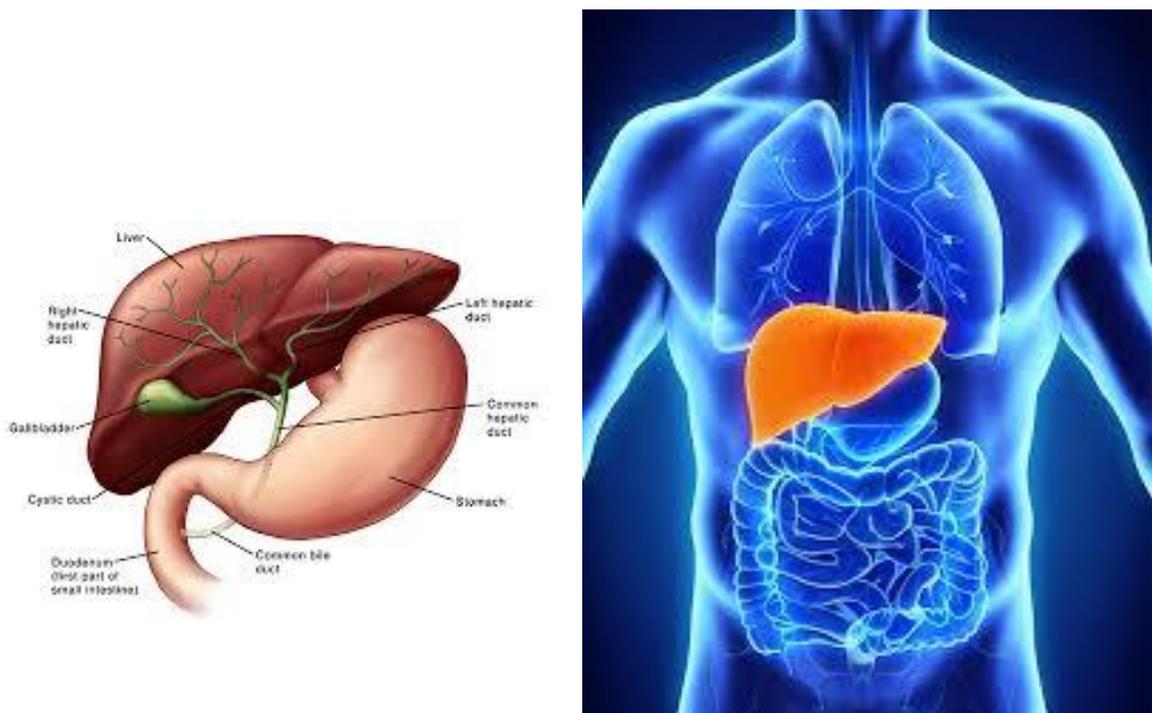


Figure3. 1: Human Liver Anatomy

The liver is a high variability organ, these variability even increases with many types of pathologies. Because of its specific location, the liver is responsible for the highest number of functions, and a very important one is to keep the body pure from toxins and harmful substances (Dixit and Pruthi, 2014). The tumor is an abnormal growth or mass found in the liver. There are several liver pathologies and most of them are related to liver tumors. Some examples of liver pathologies are liver cancer such as Hepatocellular Carcinoma (HCC), Cirrhosis, metastases, fatty liver, and fulminant hepatic failure.

When the liver contains a tumor it will be unable to function properly; these tumors can be secondary or primary. The primary tumor is that originates in the liver may be benign or malignant and the secondary tumor is that has spread to the liver from its source of origin in another part of the body. More than 40000 persons per year in the world suffer from the liver tumor (Deore, 2014). The liver is a challenging organ for surgery and requires preoperative planning before any liver operation because of: highly shape variations of the liver, inhomogeneous appearances, liver pathologies, liver tumors especially that reside near the boundary, liver subregions, fuzzy boundaries, and low contrast between liver and surrounding tissues.

3.3 CT IMAGE

3D imaging techniques have made a generous contribution in medicine in areas such as visualization, analysis, and diagnostics. Computed Tomography (CT) scans are commonly used for scanning large areas of the body. In this dissertation, the types of CT used are abdominal CT scans which are used to view the anatomical structures within the abdominal cavity. A beam of X-rays is aimed and rotated around the body to generate individual slices called tomographic images. These tomographic images contain more detailed information than conventional X-rays because they gather detail from multiple angles (Abdominal and Pelvic CT, 2018).

During the scan, different body parts absorb the X-rays in varying degrees and it is this crucial difference in absorption that allows the body parts to be distinguished from one another on an X-ray film or CT electronic image. A series of slices are produced and these can be viewed in two ways, either as individual slices or they can be stacked to produce an image volume. The abdominal CT scan can be viewed from the axial view, sagittal view or coronal view. The slices in this study are from the axial view. The different CT views are showed in Figure 3.2.

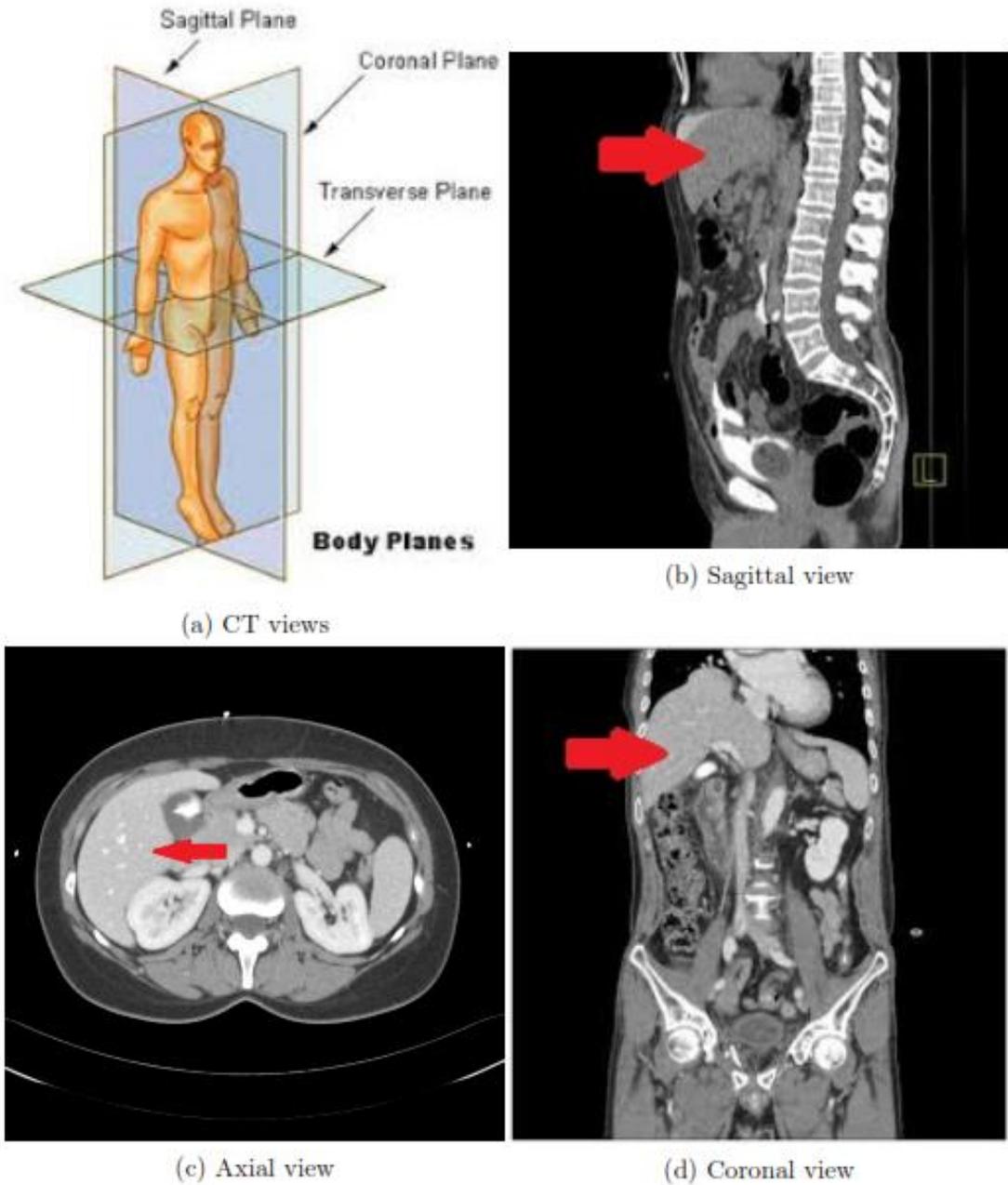


Figure3. 2: Liver in abdominal CT scan

3.4 LIVER SEGMENTATION METHOD

Liver segmentation is aim to divide the pixels of the image depending on certain criteria into two groups: pixels that belong to the object of interest (liver) and the rest of pixels that don't belong to the liver. Because of the complexity of liver shapes and variable liver sizes among patients, the segmentation of the liver from medical images is very difficult and also due to low contrast between the liver and surrounding organs like the stomach, pancreas, kidney, and muscles (Hermoye et al., 2005). Moreover, the challenge is the presence of tumors and other liver pathologies because the livers with pathologies are different from healthy ones, and that result either under segmentation or over-segmentation.

In the last few decades, a large variety of semiautomatic and fully automatic approaches have been proposed to improve the liver segmentation procedure, such as region growing, clustering, deformable models or level sets, statistical shape models (SSMs), probabilistic atlases, graph cuts and recently, deep convolution neural networks (Heimann, Wolf, and Meinzer, 2006)(Massoptier and Casciaro, 2008a). There are many efforts to survey the methods for liver segmentation and each one of them divides and categories the methods based on the different points of view, such as in (Litjens et al., 2017); they divide the automatic liver segmentation methods according to the image feature it works on, into three main classes including gray level based method, structure-based method and texture-based method as shown in Figure 3.3.

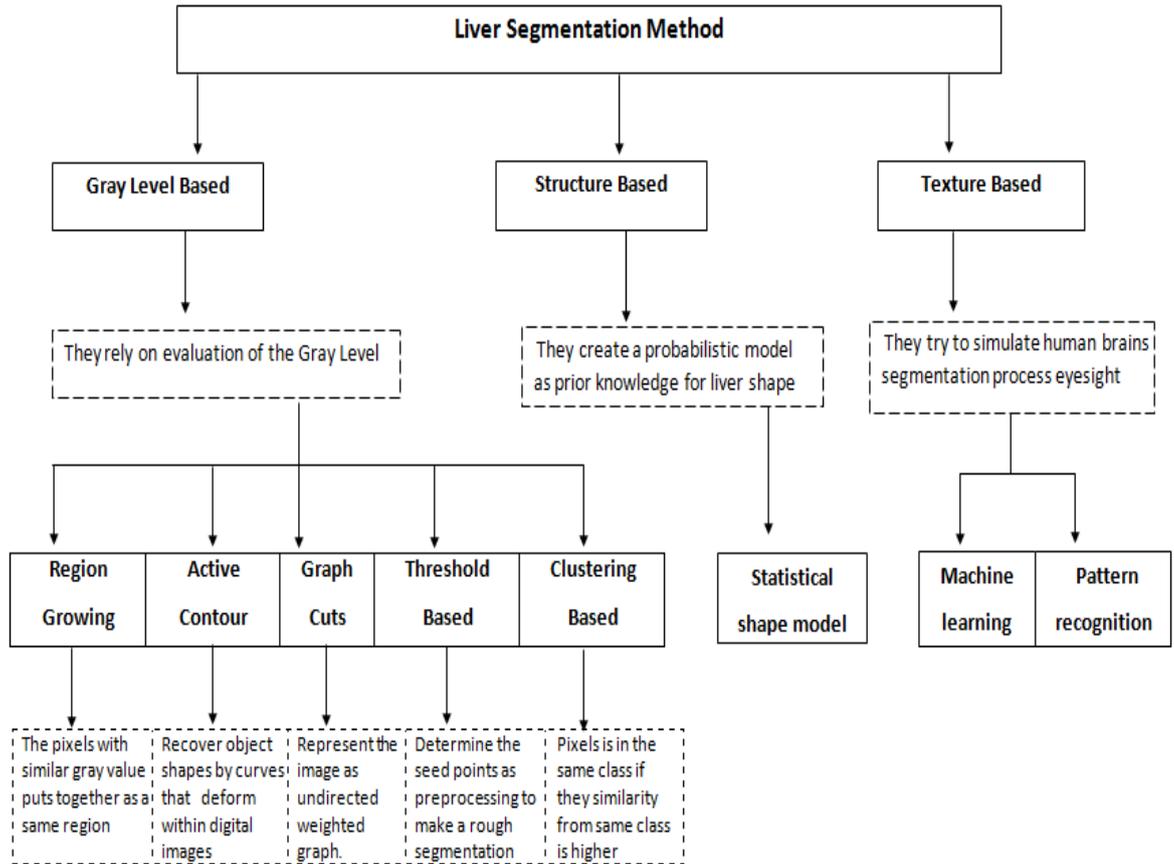


Figure3. 3: Liver Segmentation Methods Categorizing

The various algorithms used for liver segmentation can be categorized depending on the degree of automation into three groups: automatic methods, semiautomatic methods, and interactive methods (Massoptier and Casciaro, 2007a). Also the algorithms can be categorized into two groups: propagation approaches based on 2D slices and direct 3D. Generally, 3D segmentation-based methods can be classified into two classes: image-based and prior model-based (Hu et al., 2016a).

Some examples for image-based methods are region growing, thresholding, level-set-based methods, graph-cut-based methods, and others (Wu et al., 2016c). The graph cuts based methods are numerous treated problems in computer vision, such as regularization and denoising (Boykov, Veksler and Zabih, 1998) (Boykov, Veksler and Zabih, 2001), segmentation (Boykov and Kolmogorov, 2003)(Boykov and Jolly, 20010)(Rother, Kolmogorov and Blake, 2004) (Shi and Malik, 2000).

Recently, deep learning techniques are also used to solve the segmentation problems (Long, Shelhamer and Darrell, 2015)(Ronneberger, Fischer and Brox, 2015)(Çiçek et al., 2016)(Milletari et al., 2017).

3.5 SEMIAUTOMATIC AND FULLY AUTOMATIC APPROACHES

Many semiautomatic and fully automatic approaches have been proposed to improve the liver segmentation procedure some of them use a Classical Based method for segmentation and other one-use deep learning techniques for segmentation which are summarized in Table 3.1 and Table 3.2, respectively.

3.5.1 CLASSICAL METHOD FOR LIVER SEGMENTATION

A novel method was proposed in (Wu et al., 2016a) for automatic segmentation of the liver using super voxel-based graph cuts. They automatically extracted the Liver Volume of Interest (VOI) and the foreground/background seed points for graph cuts. Firstly, they were determining the region of the abdomen by using the Maximum Intensity Projection (MIP) and thresholding methods. And extract the specific liver VOI from the region of the abdomen according to prior knowledge about locating of liver organ and by using a histogram-based adaptive thresholding method and morphological operations. They generated the super voxels of the liver VOI by using the Simple Linear Iterative Clustering (SLIC) method. Secondly, foreground/background seeds for graph cuts were selected on the largest liver slice, and the graph cuts algorithm was applied to the VOI super voxels.

A single-block linear detection algorithm (SBLDA) for automatic liver segmentation from abdominal CT images was proposed in (Huang et al., 2016) they successfully reveal satisfactory segmentation results in abdominal CT images with low contrast, also it decreases the computational time because it does not require iteration and initialization. Mainly their proposed method consists of three major parts: image pre-processing, liver edge extraction with SBLDA, and image post-processing.

They compared their proposed algorithm with Shi's method (MLR-SSC); where their method is not affected by initial shapes and saves considerable runtime, on the contrary, Shi's method is affected by initial shapes which result in leading to significant segmentation error.

The authors (Peng et al., 2015) proposed a novel 3D liver segmentation method based on multi-region appearance and graph cuts approach to reducing user interaction and improving the accuracy and efficiency. The liver could contain tumor or metastasis which results in the liver with multiple sub-regions, for such a case they introduced a novel multi-region-appearance model and appearance selection scheme to segment the target multi-region object. The graph cuts approach was used to optimize the proposed energy function. They had compared their work with other graph cuts-based methods, state-of-the-art semiautomatic and interactive methods, and prior model-based methods. They found that the proposed model needs only initial seeds in the liver when compared with the other graph cuts methods and it required low interactive compared with the other semiautomatic and interactive methods and also the proposed model can be applied to livers with any shape because it was not restricted by specific training data when compared with prior model-based methods.

Liver tumors detection and segmentation method were developed in (Huang et al., 2013), using the fast learning algorithm Extreme Learning Machine (ELM). A two-class ELM classifier was used to detect and segment the suspicious region of the tumor which firstly was learned for voxel classification using the tumor/non-tumor samples selected by the user, followed with erosion and dilation operation to morphological smoothing. They also proposed one-class ELM for tumor detection where the user only needs to select healthy liver samples; and compared them with two-class ELM. A semi-automatic approach is used to extract the boundary of a tumor by randomly selecting samples within a limited region of interest for classifier training where each voxel is associated with set features such as entropy, Law's features, and sum-and-difference histograms. Their proposed kernel-based ELM achieves encouraging results when compared with traditional ELM, and is also faster than SVM. And because of the available more tumor information in two-class ELM; it had shown a relatively higher accuracy when compared with one-class ELM, but not in the case of an unknown tumor.

A Fully automatic CT liver segmentation using a novel statistical shape model approach was presented by (Erdt et al., 2010) that combines learned local shape constraints with observed shape deviation during adaptation.

(Massoptier and Casciaro, 2008a), Presents a new hybrid fully automatic method for fast segmentation of the liver and the lesions inside the liver by using a statistical model-based approach and active contour technique. There is no required interaction with the user. The role of the statistical model approach was to distinguish the liver tissue from other surrounding abdominal organs. On the other hand, the active contour technique is used in order to obtain more natural and smoother liver surface segmentation. They compared the length of processing time of their work and the accuracy with a manual contour-drawing approach made by an expert radiologist who delineated each liver slice using a public-domain image processing program (ITK-SNAP). The assessment expressed a good result but the algorithm faced some limitations in special pathological and anatomical situations, such case is when the heart and the liver were in close contact, so difficult to distinguish the limit between these two organs, as a result of that part of the heart included into the liver segmentation.

A semi-automatic method to identify tumors from 3D CT scans based on 2D region growing with knowledge-based constraints was proposed by (Wong et al., 2008). Firstly, they decompose the 3D scan image into component slices. Then, they apply 2D region growing with knowledge-based constraints on each slice. Finally, they load up the individual segmented lesions together to generate a 3D volume. The region-growing approach is used to calculate the seed point and feature vectors and also to label the voxels. The Knowledge-based constraints were used to ensure that the segmented region size and shape are within acceptable parameters. The method required minimal user involvement in order to define an approximate region of interest around the lesion in each slice image which improved the performance of region growth, as well as reduced computational requirements.

A supervised learning algorithm for liver segmentation in CT image is presented in (Seghers et al., 2007), which is the minimal cost path segmentation method; and also it is similar to Active Shape Model segmentation methods in the sense that it models the object as a set of landmarks (vertices) and connections between neighboring landmarks

(edges), but it differs in applying multiple local shape models instead of using a global shape model. For the local shape and gray-level appearance, statistical models were built based on a set of training images and corresponding surface meshes. The proposed work demonstrates the potential of segmenting the liver in contrast-enhanced CT images when validated in the first implementation and it participated in the Grand Challenge on 3D segmentation (MICCAI 2007).

Lipková et al. propose an unsupervised method for liver lesions segmentation using a phase separation approach (Lipková et al., 2017). They were depending on assumption that the liver is a combination of two parts: healthy liver and lesions, represented by dissimilar image intensities polluted by noise. To remove this noise and also to separate the liver part into two distinct phases with well-defined interfaces they used the Cahn-Hilliard separation (CHS) method. The 3Dircadb and LITS dataset was used to test the proposed method.

A semiautomatic method based on improved fuzzy C-means (FCM) and graph cuts was proposed by Wu et al for liver tumor segmentation in CT images (Wu et al., 2017). Their algorithm consists of two major steps: first, the tumor volume of interest (VOI) is extracted using a confidence-connected region growing algorithm to reduce computational cost. Then, initial foreground/background regions were labeled automatically, and a kernelized FCM with spatial information was incorporated in graph cuts segmentation to increase segmentation accuracy. They evaluate their work on the public clinical dataset (3Dircadb) and achieved an accurate result for 3D liver tumor segmentation with a reduction of processing time.

An application of minimal surfaces and Markov random fields to the segmentation of liver tumors were presented in (Stawiaski et al., 2017) (Stawiaski et al., 2017). They were using a region graph instead of a pixel graph in applying these models which leads to an interactive method used to segment tumors in 3D CT. their strategy depended on the manual definition of a sub-volume containing one or more tumors that need to be segmented based on simple observation which is: the liver presents two types of tissues: tumoral and healthy tissues to classification the liver pixels they were model the liver pixels as a Markov Random Field and the classification is performed through the maximum a posteriori estimation. But this classification step is supervised by a user in defining the markers that specify both tumoral and healthy tissues. These

markers are used to locate the tumors and to estimate the grey levels characteristics of these structures. However, the extracted liver boundaries are needed also in the liver classification. This is done by computing a minimal surface based on user defined markers. So finally the user has to specify this information: normal liver tissues, tumoral tissues, and external tissues surrounding the liver if necessary. Have compared their work with the radiologist's segmentation on a set of 5 CT images presenting 10 tumors and found that the mean surface distance between their segmentations and the radiologists is approximately 1.5 mm. approximately 71 % of their segmentations is in perfect match with the radiologist's segmentation. The total mean score obtained on the training data set was equal to 88. And the computation time needed for the tumor segmentation is approximately equal to five, up to eight minutes.

Table 3. 1: Classical Based methods for liver Segmentation

Authors	Year	Used Method	Datasets	Result
W. Wu et al.	2016	Supervoxel Based Graph Cuts	Sliver07	VOE = 7.87% RVD = 1.31% ASD = 1.286 mm RMSD = 2.498 mm MaxD = 23.563 mm
L. Huang et al.	2016	Single-Block Linear Detection Algorithm (SBLDA)	3D-IRCAD	sensitivity = 96.59% accuracy = 98.65% specificity = 99.03%
Peng J et al.	2015	A Novel Region-Appearance And Graph Cuts.	MICCAI + MICCAI 2007 + local hospitals	VOE = 4.58%±0.51% RVD = 1.08%±0.80% ASD = 0.68±0.14 mm RMSD = 1.45±0.36 mm MSD = 16.89±3.69 mm overall score = 83.4±3.1
W. Huang et al.	2013	fast learning algorithm Extreme Learning Machine (ELM),	From Different Hospitals	mean VO = 67.15% mean VD = 14.16% mean ASD = 2.27mm mean RMSD = 2.47mm mean MSD = 8.46mm
Erdt et al.	2010	learned local shape constraints with observed shape deviation	3D-IRCAD	average mean surface distance = 1.3--1.85 mm processing time = 45 s

Authors	Year	Used Method	Datasets	Result
Massoptier et al.	2008	Statistical Model-Based Approach And Active Contour Technique	From Different Hospitals	VO = 94.2% Sensitivity= 82.6% specificity =87.5%, accuracy of liver surface segmentation = 3.7 mm
D. Wong et al.	2008	2D region growing with knowledge-based constraints	From Different Hospitals	Ave Overlap Error =39.40 % AVD= 24.20% ASD=2.20mm RMSSD =3.02mm MSD=12.69mm Total Score =64
Seghers et al	2007	Supervised Learning Algorithm	Sliver07	The method has potential to result validated segmentation
Lipková et al	2017	Cahn-Hilliard separation (CHS)	3Dircadb and LITS	Dice = 0:61 Sensitivity =0:64 Specificity =0:99 Precision =0:65 Detection =0:73
Wu et al	2017	fuzzy C-means (FCM) and graph cuts	3Dircadb	VOE =29.04% RVD = 2.20% ASD =0.72 mm RMSD =1.10 mm MaxD = 4.25 mm DICE = 0.83
Stawiaski et al	2017	graph-cuts and a watershed	Sliver07	VOE =29.49 % RVD = 23.87 % ASD = 1.50 mm RMSD =2.07 mm MaxD =8.29 mm

3.5.2 DEEP LEARNING TECHNIQUES FOR LIVER

SEGMENTATION

A fully automatic framework was proposed for liver segmentation based on a 3D convolutional neural network (CNN) and globally optimized surface evolution (Hu et al., 2016a). Firstly, the deep 3D CNN gives the initial liver surface after it was trained to learn a subject-specific probability map of the liver. Then, refining the initial liver segmentation by using the prior information about novel energy function; finally, propagated the initial liver surface to the optimal position by minimizing the energy function using a global optimization-based approach.

Lu et al. develop a fully automatic liver segmentation framework without any user interaction (Lu et al., 2016). This combines a deep learning algorithm and graph cut approach; mainly their framework consists of two main steps: (1) liver detection and segmentation using convolutional neural networks(CNN) model which is used to learn the liver likelihood map to automatically identify the liver surface. (2) Refinement the initial liver segmentation from the first step by incorporating the learned liver probability map into a graph cut model. To evaluate their framework they use 40 contrast-enhanced CT volumes from two public databases MICCAISliver07 and 3Dircadb. They found that their framework can increase the efficiency of the physician.

Christ et al. present a combined automatic segmentation of the liver and its lesions in CT and MRI abdomen images using two cascaded fully convolutional neural networks (CFCNs) one for the segmentation of the liver and the other for its lesions (Christ et al., 2017a). They use an abdominal CT dataset comprising 100 hepatic tumor volumes for training the CFCN models. In the first step of their work, they pre-processed the CT/MRI image with HU-windowing or N4 bias correction. Then in the second step, segmenting the liver from abdomen CT/MRI scans using first pertained FCN after that they use the output of the last step to be input for a second FCN which segments lesions from the given segmented liver ROI. And finally, the last step was the post-processing using 3D Conditional Random Field (3D CRF). Their results show that the CFCN achieves Dice scores over 94% for the liver with computation times below 100s per

volume. And in contrast to prior work, their method could be generalized to segment multiple organs in medical data using multiple cascaded FCNs.

Vivanti et al presented a new automatic algorithm for liver tumor segmentation in follow-up CT studies (Vivanti et al., 2015). This method combines a follow-up-based detection with CNN-based segmentation where the inputs are a baseline CT scan and a delineation of the tumors in it and a follow-up scan and the outputs are delineated tumors in the follow-up scan. The presented method consists of registration, deep learning, and segmentation. The registration stage begins with automatically computing a liver mask using a liver segmentation method that relies on Bayesian classification, adaptive morphological operations, and active contours. This segmentation is performed for baseline and follow-up scans. Next, an ROI is defined that contains the follow-up tumor with high probability. The baseline delineation is transferred to the follow-up scan and the follow-up tumor ROI is doubled in each direction to account for possible tumor growth. A CNN is used to classify each voxel as being healthy liver or tumor. The classification is based on voxel intensities in an axis-aligned square centered at the voxel. The segmentation of the ROI is done by classifying all its voxels in four steps. First, the trained CNN is run in feed-forward to classify each patch. Then, the non-liver voxels are classified as healthy tissue using the liver mask of the follow-up scan. The experiments were carried out by two metrics: VOE and ASD. The results obtained were 16.75% for VOE and 2.05mm for ASD. The overall success rate for this method was 90.47%.

A novel 3D deeply supervised network (DSN) for automatic liver segmentation was presented by Dou et al (Dou et al., 2016). This method employs a fully convolutional architecture to produce a high-quality score map which is processed further by the employment of a conditional random field (CRF) to obtain a refined segmentation. A per-pixel-wise binary classification error minimization problem concerning the ground-truth mask is formulated for the learning of the 3D network. Additional supervision is injected into some hidden layers to counteract the effects of gradient vanishing which makes the loss back-propagation ineffective in the first few layers. Some lower-level and middle-level feature volumes are up-scaled using additional de-convolutional layers and then the softmax layer is employed to obtain dense predictions for calculating classification errors. Gradients are derived from these branch predictions

and the last output layer alleviates the effects of gradient vanishing. Although the 3D DSN generates high-quality probability maps, a graphical model is employed to re-new segmentation results due to imprecise segmentation in ambiguous regions. A fully connected CRF model on the transverse plane is exploited which solves an energy function to re-new the segmentation. The MICCAI-SLiver07 dataset was used. This method was trained on twenty CT scans and tested on ten CT scans. The learning process of the proposed 3D DSN showed that it converges much faster and achieves lower training/validation errors than that of a pure 3D CNN. The segmentation results achieved were a VOE of 5.42% and an average symmetric surface distance of 0.79mm. It can be concluded that the presented framework is effective and efficient and achieves competitive segmentation results to state-of-the-art approaches.

Ben-Cohen et al explored the use of a fully convolutional network (FCN) for liver segmentation and liver metastases detection in CT examinations (Avi Ben-Cohen, Idit Diamant, Eyal Klang, Michal Amitai², 2016). They proposed a network architecture that has 16 layers where all fully connected layers are converted to convolutions. The initial network used is an FCN-8s DAG network which learned to combine coarse, high information with fine, low layer information. The addition of a lower level linking layer was also explored creating an FCN-4s DAG network. Input images and their corresponding segmentation maps are used for training with stochastic gradient descent and GPU acceleration. Two networks are trained here, one for liver segmentation and the other for lesion detection. The softmax log-loss function was calculated pixel-wise with different weights for each pixel class. Two framework variations were used for evaluation as well as the initial framework. The segmentation performance was evaluated using the Dice index, sensitivity, and positive predicate values (PPV). The FCN-8s with 3 slices produced a Dice of 0.89, the FCN-8s produced a Dice of 0.88, and the FCN- 36 4s with 3 slices produced a Dice of 0.87. The FCN-8s with 3 slices produced the best results with an average sensitivity of 0.86 and an average PPV of 0.95. The lesion detection performance was evaluated using two metrics: true positive rate (TPR) and false positive per case (FPC). The dataset used is the MICCAI-SLiver07 challenge dataset. Here, the FCN-4s with 3 slices performed the best producing a TPR of 0.88 and an FPC of 0.74. The method presented here is promising for both lesion detection and liver segmentation. However, it should be noted that no significant pre-

processing or post-processing was implemented which could have improved the results achieved by the presented method.

This study (Chung et al., 2020) introduced a CNN for liver segmentation on abdominal computed tomography (CT) images with focusing on the performance of generalization. They show high generalization performance and accuracy. They proposed an auto-context neural network; it achieved an effective estimation to obtain the shape prior. They use a self-supervised contour scheme to extend their network. They achieved a better accuracy when compared to the state-of-the-art networks by reducing 10.31% of the Hausdorff distance.

In this paper (Ibtehaz and Rahman, 2020), the authors propose some modifications to U-net model architecture by developing a novel model called MultiResUNet which is an enhanced version of U-net. The hypothesis was that there was a contradiction between the features passed from the encoder network and the features propagating through the decoder network and to harmonize these contradictions they proposed additional processing called Respaths, also to increase the ability of multi-resolution analysis they proposed MultiRes blocks which were inspirations from Inception blocks. They test their model using five variety medical image datasets of different modalities. The five datasets are Murphy Lab, ISBI-2012, ISIC-2018, CVC-ClinicDB, and BraTS17. They had obtained a relative improvement in performance of 10.15%, 5.07%, 2.63%, 1.41%, and 0.62% respectively. They also experimented with a 3D version of MultiResUNet, and it outperforms the standard 3D U-net as well.

A literature review of medical image segmentation based on U-net was presented by (Du et al., 2020). They were focused on the successful segmentation experience of U-net in six medical imaging systems including computed tomography (CT), magnetic resonance imaging (MRI), ultrasound, X-ray, optical coherence tomography (OCT), and positron emission computed tomography (PET). There are many kinds of lesion regions extracted by this application. Also in this study, they have introduced the method of combining the original U-net architecture with deep learning and a method for improving the U-net network. They came out that the six imaging systems mentioned in their article, are not perfect in some imaging systems. Consequently, it needs to be improved in future studies for application in various imaging systems.

A new, more powerful deeply-supervised encoder-decoder segmentation architecture based on nested and dense skip connections named U-net++ was proposed by (Zhou et al., 2018). U-net++ is different from U-net in that the feature maps of the encoder are directly received in the decoder in U-net but the U-net++ model makes them undergo a nested convolution block. They evaluated their proposed architecture in comparison with U-net and wide U-net architectures across four medical imaging datasets including lung nodule segmentation, colon polyp segmentation, cell nuclei segmentation, and liver segmentation. They came out that U-net++ with deep supervision achieved an average IoU gain of 3.9 for U-net and 3.4 points for wide U-net. According to the results of their experiments, the proposed architecture in this study was effective, yielding significant performance gain over U-net and wide U-net. U-net++ was used in comparison in this study (Shrivastava et al., 2020) where it was compared with two other deep learning models: CE-Net and MultiResUNet for the segmentation of Corpus Callosum in the Brain MRI images.

Two deep encoder-decoder convolutional neural networks (EDCNN) were constructed and trained to cascade segments of both the liver and lesions in CT images in this paper (Budak et al., 2020). The first EDCNN was responsible for segmenting the liver image which was been input for the training of a second EDCNN; where it then responsible for segments the tumor regions within the liver ROI regions as predicted by the first EDCNN. To evaluate their performance the proposed EDCNN networks produced an average DICE score of 95.22% for the test set of CT images using a public dataset (3DIRCADb) and it was compared with some of the existing methods. The experimental results demonstrated that the proposed EDCNN achieved improved performance in segmentation accuracy over some existing methods.

Modified U-net (mU-net) network architecture was proposed by (Seo et al., 2020), they were dependent on processing an object-dependent upsampling and restructures the residual path and the skip connection by combining features in the residual path into features in the skip connection. The mUNet is differ from the U-net architecture it handles edge information and morphologic information of the objects more effectively than the U-net. Furthermore, to extract high-level global features of small object inputs the proposed architecture has additional convolution layers in the skip connection. The proposed modified U-Net (mU-net) was evaluated using two public datasets which are

Liver tumor segmentation (LiTS) challenge 2017 and 3D Image Reconstruction for Comparison of Algorithm Database (3Dircadb). They came out of this result for liver segmentation respectively: DSC of 98.51 %, DSCs was 96.01 % for the liver.

Table 3. 2: Deep learning methods for liver Segmentation

Authors	Method	Detailed Architecture	Datasets	Result
Hu P et al. 2016	3D Convolutional Neural Network (CNN) and Globally Optimized Surface Evolution.	-----	Sliver07 and local hospitals	VOE = 5.35±1.23% RVD = - 0.17 ±1.34% ASD = 0.84±0.25 mm RMSD = 1.78 ±0.56 mm MSD = 19.58± 3.07 mm overall score = ±80.34.5
Lu et al., 2016	Deep learning algorithm and graph cut	11 conv layer 2 pool layer	MICCAI and 3Dircadb	For MICCAI-Sliver07 : mean VOE =5.9% mean RVD =2.7% mean ASD =0.91% mean RMSD = 1.88 mm mean MSD= 18.94 mm For 3Dircadb mean VOE =9.36% mean RVD =0.97% mean ASD =1.89% mean RMSD = 4.15 mm mean MSD= 33.14 mm
Christ et al., 2017	two cascaded fully convolutional neural networks	-----	3Dircadb	Dice=93.1 VOE=12.8 RVD=-3.3 ASD=2.3 MSD=46.7
Vivanti et al,2015	convolutional neural networks	-----	From different patient	VOE =16.75% ASD=2.05mm overall success rate for this method was 90.47%
Dou et al, 2016	A novel 3D deeply supervised network (DSN)	-----	MICCAI	VOE =5.42% ASD=0.79mm
Ben-Cohen et al 2016	fully convolutional network (FCN)	8 conv layer 5 pool layer 2 Upsample layer	MICCAI	Dice of 0.89 average sensitivity=0.86 average PPV =0.95

Authors	Used Method	Detailed Architecture	Datasets	Result
Chung et al. 2020	auto-context neural network;	-----	From different datasets	DSC =0.96 Precision= 0.95 Sensitivity =0.97 HD [mm] =14.96 ASSD [mm]=0.82
Ibtehaz and Rahman 2020	modifications to U-Net model (MultiResUNet)	-----	From different datasets	Dice =0.0802 Accuracy = 0.7971 Sensitivity=0.9813 Specificity= 0.7955
Zhou et al., 2018	deeply-supervised encoder-decoder	-----	From different datasets	IoU=82.90
Budak et al., 2020	Two deep encoder-decoder convolutional neural networks (EDCNN)	10 conv layer 5 pool layer 5 Up-pool Softmax layer	3Dircadb	VOE =9.05 % RVD=7.03 % ASSD=1.43 mm MSD=19.37 mm DICE=95.22%
Seo et al., 2020	Modified U-Net (mU-Net)	17 conv layer 4 pool layer 3 Upsample 4 deconv layer	LiTS and 3Dircadb	(LiTS) DSC =98.51 % (3Dircadb) DSCs =96.01 %

CHAPTER 4

PROPOSED MODEL FOR LIVER SEGMENTATION

4.1 INTRODUCTION

In this chapter, we will explore the model for liver segmentation through a convolutional neural network based on the U-net model. The U-net structures will be described and applied to real CT-Scan data from MICCAI and 3Dircad datasets and check how it performs.

4.2 MICCAI DATASET

The datasets used in this research are the Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2007 grand challenge dataset and 3D Image Reconstruction for Comparison of Algorithm Database (3D-IRCAD). All CT images in MICCAI enhanced with a different operator and scanned in the focal venous stage on an assortment of scanners (Styner et al., 2008). As it is CT, all datasets acquired in the crossover direction. The pixel dispersing varied somewhere in the range of 0.55 and 0.80 mm and inter-slice distance varied from one to three mm. The majority of the images pathological and include tumors, metastasis, and sores of different sizes. The dataset used for preparing is the MICCAI 2007 grand challenge preparing set (MICCAI-Training) which comprises 20 volume images with corresponding ground truth segmentations. The dataset used for testing is the MICCAI 2007 grand challenge testing data (MICCAI-Testing) which comprises 10 volume images.

4.3 3D-IRCAD DATABASE

The 3D Image Reconstruction for Comparison of Algorithm Database (3D-IRCAD) database is composed of the CT scan of 20 patients (10 men and 10 women) with hepatic tumors in 75% of cases. The images are provided by the authors in DICOM and VTK format in 512x512 pixels. It also contains handmade true segmentation for liver, bones, tumors, and others by medical specialists for all images of the 20 patients. The CT image in 3D-IRCAD is 512x512 but may take a very long time to complete training such images on the personal computer, so the images were scaled down to 128x128 pixels.

Each patient data is distributed across two folders. Inside each of these two folders, there are 20 folders numbered from 1 to 20 representing each one of the patient's numbers. The first folder is the PATIENT_DICOM folder containing the real CT scan image and another folder is the MASKS_DICOM folder which contains the corresponding segmentation mask files. For each patient data in the MASKS_FOLDER, there is one folder per segmented biological structure (i.e.: liver, bones, tumors, etc.). Thus our ROI is only liver we only kept the liver file in the MASKS_DICOM folder.

4.4 U-NET

U-net is an artificial neural network kind of Convolutional Neural Networks (ConvNets) approach that is able to produce visual information. When visualize U-net architecture it appears similar to the letter U so it takes its name from here. U-nets were first proposed by Olaf Ronneberger, Phillip Fischer, and Thomas Brox in 2015 for Bio Medical Image Segmentation for three different segmentation tasks, the first task is the segmentation of neuronal structures in electron microscopic recordings. Second task was on a cell segmentation task in light microscope images (Ronneberger, Fischer and Brox, 2015). An example of their obtained segmentation result is displayed in Figure 4.1.

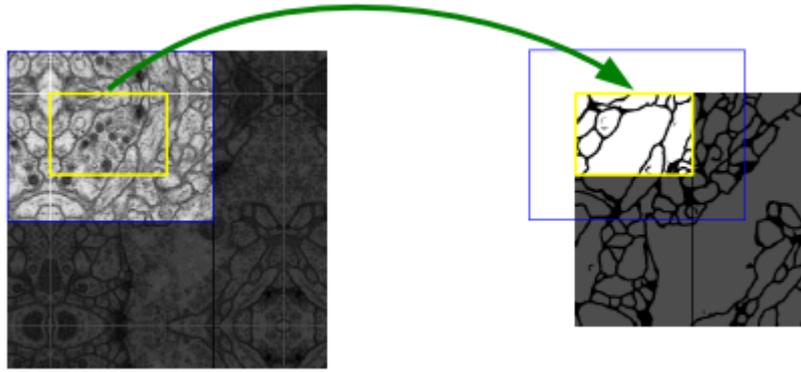


Figure 4. 1: example of obtained segmentation result by (Ronneberger et al., 2015).

U-net is an image segmentation technique developed originally for image segmentation tasks and it was adoption as the primary tool for segmentation tasks in medical imaging because it is success of evident in its widespread use in nearly all major image modalities, from CT scans and MRI to X-rays and microscopy. The great idea about U-net is that it is able to receive an image as input and produce another image as output with size equal to the size of input , which is pretty useful for generating segmentation images. Furthermore, while U-net is largely a segmentation tool, there have been instances of the use of U-net in other applications (Siddique et al., 2021).

U-net architecture shown in Figure 4.2, it contains two paths: contraction path (also called as the encoder) and expanding path (also called as the decoder). The encoder part is used to capture the context in the image using convolutions layer. The decoder part is used to enable precise localization using transposed convolutions (Ronneberger, Fischer and Brox, 2015). (Lamba, 2019) show a detailed explanation of the original architecture which shown in Figure 4.3.

The main idea behind the U-net is that during the training phase the first half which is the contracting path is responsible for producing the relevant information by minimizing a cost function related to the operation desired and at the second half which is the expanding path the network it would be able to construct the output image.

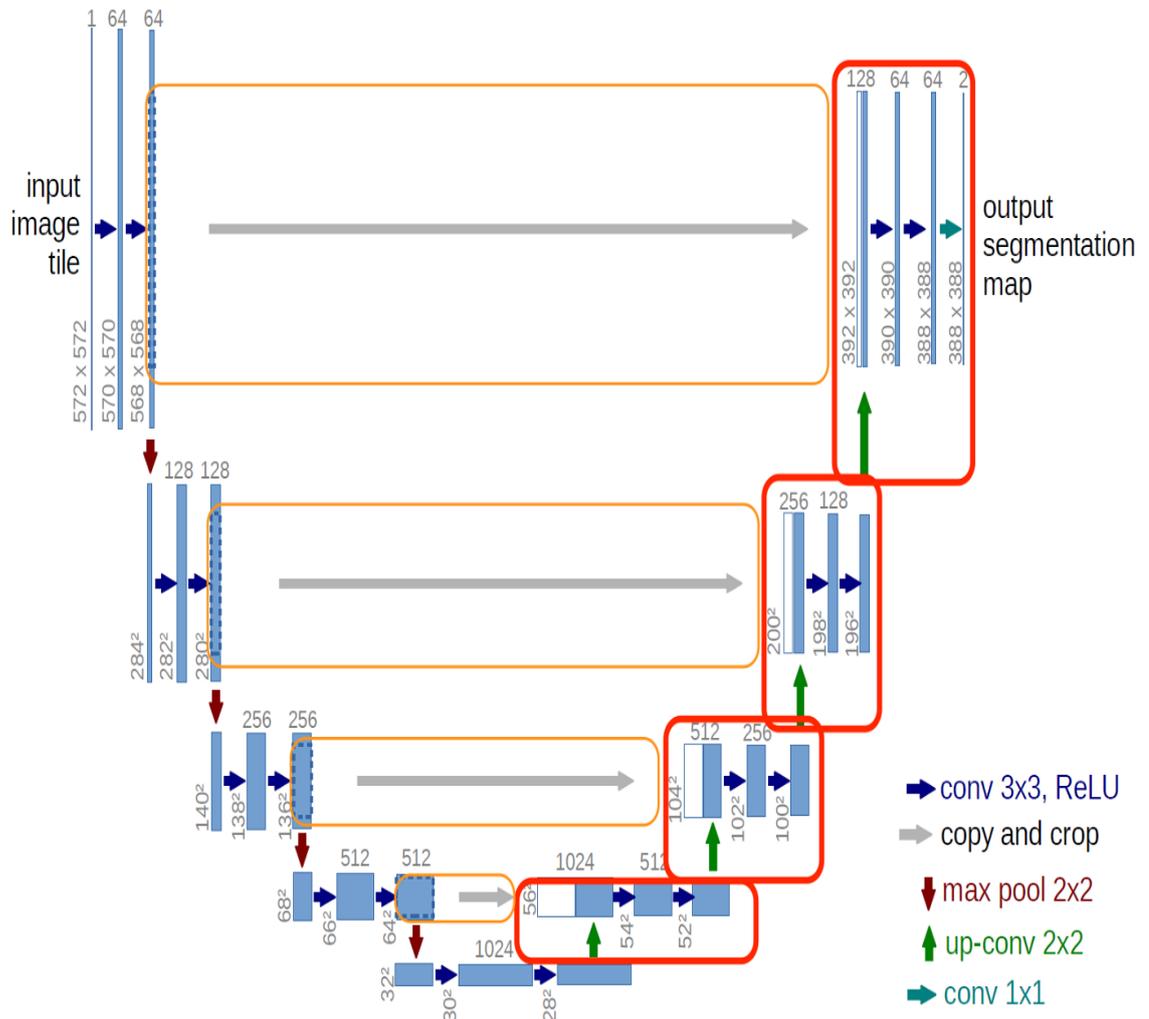


Figure 4. 2: U-net Architecture (Ronneberger et al., 2015).

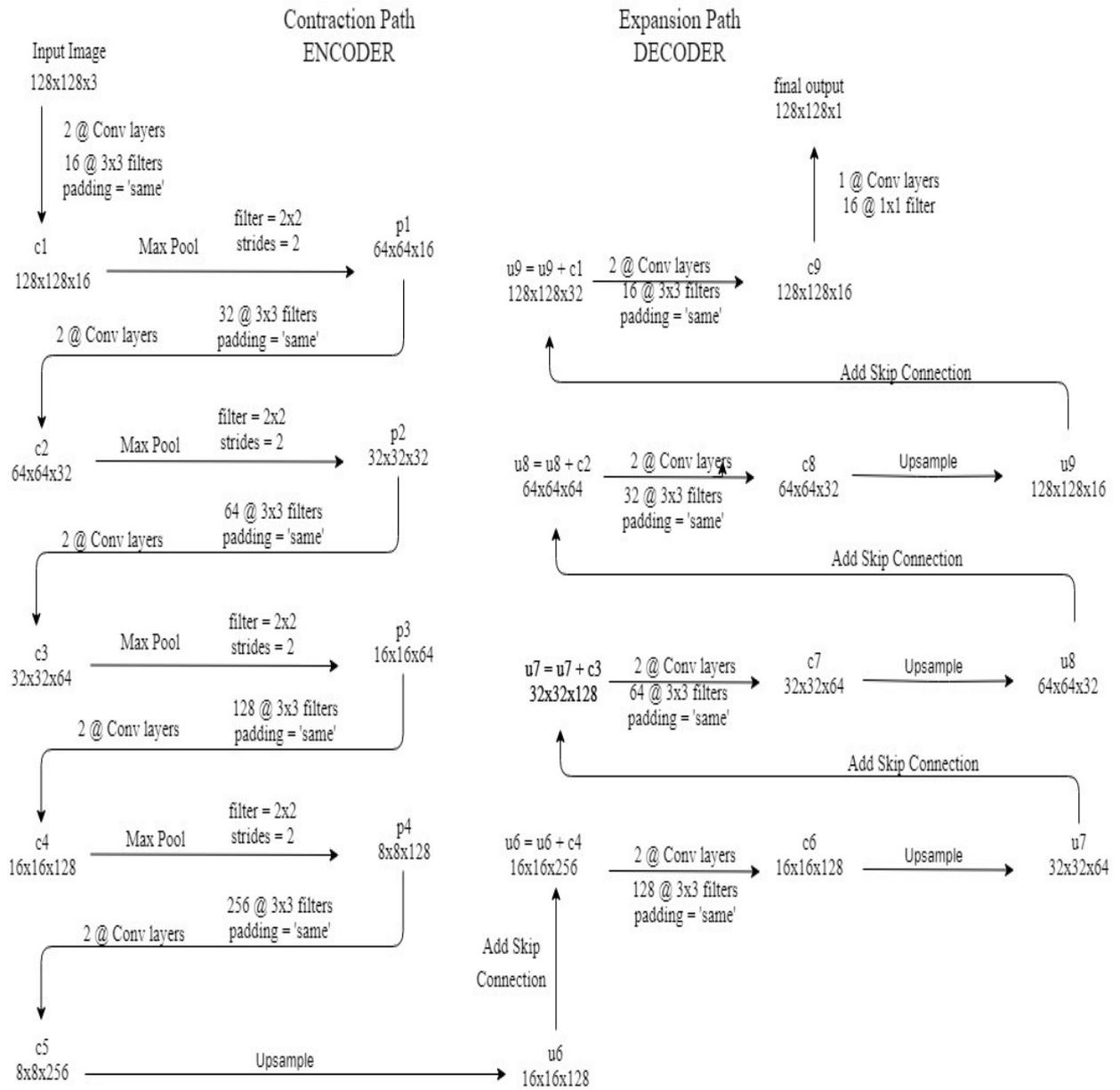


Figure 4. 3: Detailed architecture (Lamba, 2019)

4.5 METHODOLOGY

The Methodology of proposed Unet model is shown in the flowchart in Figure 4.4. It consists of four steps: First is pre-processing the CT images. The second step is training the proposed Unet model for liver segmentation. Then, testing the trained network produces a probability map as a subject-specific prior, which assigns each pixel the likelihood of being the liver for the target image. The last step is the post-processing step to maximize the resulting efficiency.

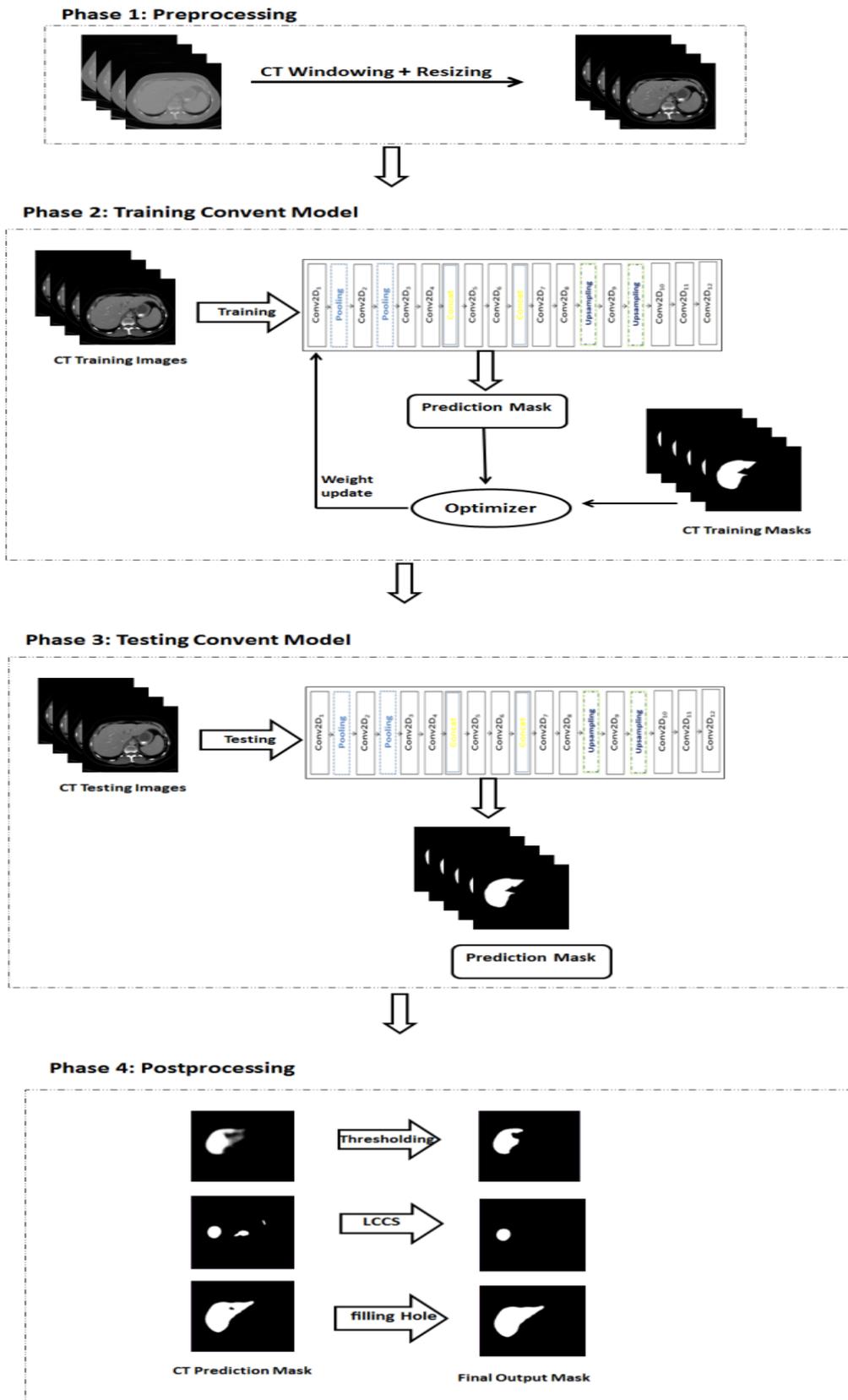


Figure 4. 4: The flowchart of our proposed Unet model.

4.5.1 PRE-PROCESSING

CT image is firstly resized to 128x128 before fitting to the network because all CT images in MICCAI and 3D IRCAD datasets are 512x512, but training such images on the personal computer may take a very long time to complete. Also, any empty images were discarded from both the original image and its mask because they would not assist in the training process.

Although convnet requires significantly less image pre-processing than traditional methods, this is still an essential task that can improve training results. CT Windowing is an important procedure in a CT scan, also known as grey-level mapping, or contrast enhancement is the process in which the CT image Grayscale component of an image is manipulated via the CT numbers, Pixel intensity measured in Hounsfield Units (HU). The value of the "Liver Window" given by Sahi et al is [-62, 238] HU (Sahi et al., 2014) and P. F. Christ et al. was truncated the image intensity of all CT scans to the range [-100, 400] HU (Christ et al., 2017b) .As such, our CT scans range was set to the range [-150, 230] HU as shown in Figure 4.5.

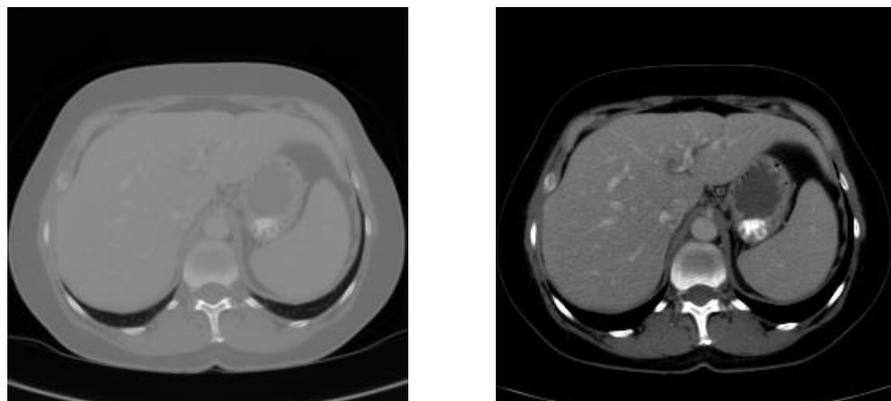


Figure 4. 5: Overview of the applied CT Windowing

4.5.2 THE ARCHITECTURE OF MODEL

In 2016, Christ et al worked for automatic liver segmentation on CT scan images using U-net and get 93.1% for Dice (Christ et al., 2016). We propose some changes to the original architecture to enhance the Dice that got it by the original U-net, where we add a batch normalization layer after each convolutions layer. The Batch Normalization layer used to normalize the values going into each activation function by applying a transformation that maintains the mean output close to zero and the output standard deviation close to one; which enhance the overall segmentation result. Our proposed model architecture consists of four parts: The contracting path (convolutions), the Middle layer, the expanding path (De-convolutions), and the output layer.

The contracting path follows the typical architecture of the U-net network with adding a dropout layer after 2 convolutions and one max-pooling layer. It consists of 3×3 convolutions repeated two times, each one followed by a batch normalization layer and a rectified linear unit (ReLU) activation, which it followed by a 2×2 max-pooling layer and dropout layer. The purpose of the dropout layer is to avoid overfitting problems. This block of two convolutions, one max-pooling layer, and the dropout layer repeated four times. After the contracting path, there is a middle layer consisting of convolutions layers.

The expansive path consists of a transpose convolutional layer known as De-convolution layers and concatenation with the corresponding cropped feature map from the contracting path, followed by 3×3 convolution repeated two times. This block of De-convolution layers, concatenation, and two 3×3 convolutions layers repeated four times. At the final layer, 1×1 convolution used to map each component feature vector to the desired number of classes. In total the proposed Unet model consists of 19 convolution layers, 18 BatchNormalization layers, and 4 max-pooling layers, 4 concatenate layers, 4 dropout layers, and 4 Transposed convolution layers (Deconvolution layer) as well as a softmax layer.

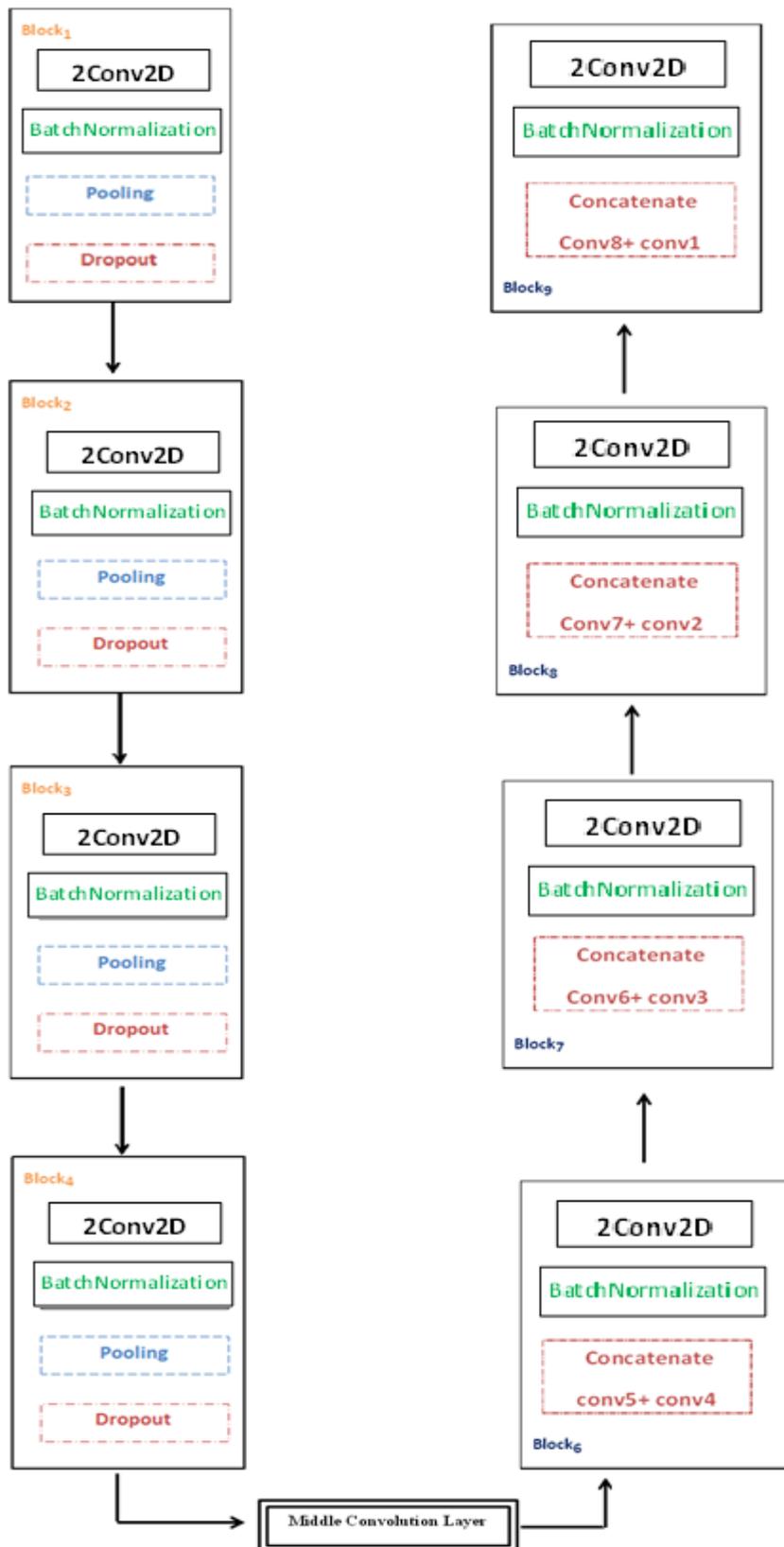


Figure 4. 6: Architecture of proposed Unet Model

4.5.3 POST-PROCESSING

Post-processing is used to refine the segmentation results produced by the previous segmentation step. To enhance the overall segmentation results we did a post-processing step, which remove the small area in the final segmentation mask by following two steps. First, we calculate the largest connected area in the final segmentation mask to set it as the threshold volume by setting the background pixel intensity to zero, and each area that is not zero within the mask labeled. Then, remove all the areas that are less than the threshold.

A. Thresholding

The Binary Thresholding is used to eliminate the pixels with a probability below 50 percent in the probability map, which in our case is a pixel value 127 which is 50% of the maximum pixel value of 255 to do that we use a *cv2.threshold()* function in *OpenCV* library. The result of the thresholding post-processing can be seen in Figure 4.7 where (a) and (b) are slices before and after thresholding, respectively.

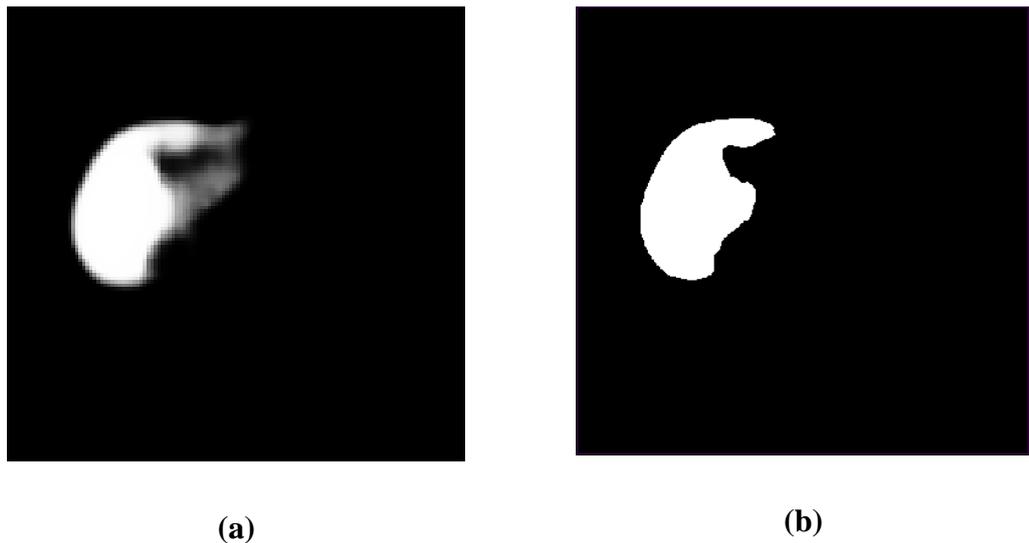


Figure 4. 7: Show result of the thresholding.

B. Largest Connected Component Selection (LCCS)

The Largest connected component selection (LCCS) technique has been used as a primary segmentation or as post-processing. In our case, we used it as post-processing to refine the final result. We calculate the Largest connected component in the resulting segmentation volume by setting the background pixel intensity to zero and each component that is not zero within the volume is labeled. Then using python Image Processing Toolbox to calculate the volume of each component and the component with the largest volume is selected.

This volume is then set as the threshold volume and all components with a volume less than this threshold volume are removed using morphological operators, this is done using the `remove_small_objects()` function in the image processing Python package `scikit-image`. The result of the LCCS can be seen in Figure 4.8 where (a) and (b) are slices before and after the LCCS process, respectively.

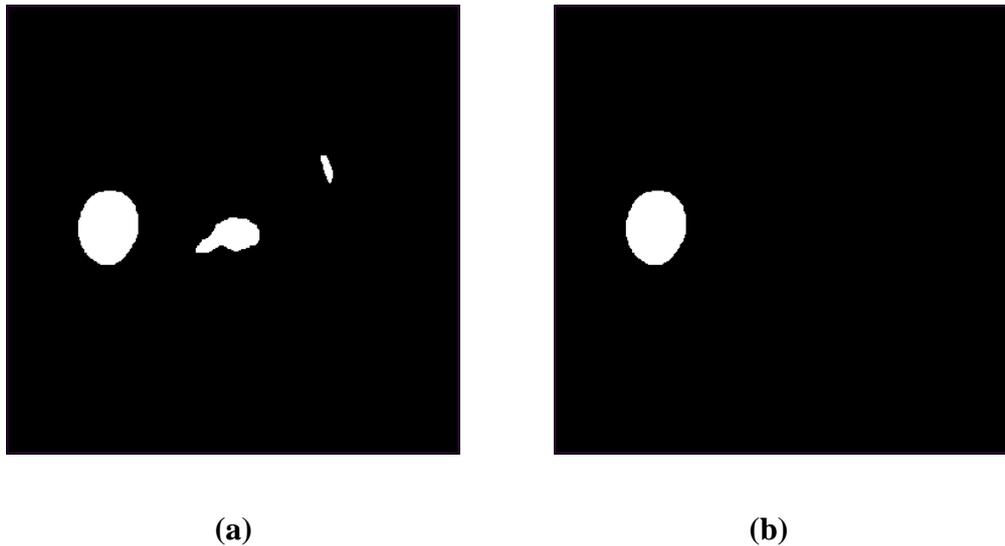


Figure 4. 8: Show the result of the LCCS process.

C. Filling The Hole

After the largest connected component is detected and the smaller components are removed, the resulting volume is processed using Morphological closing to remove the hole from it by using the `remove_small_holes()` function in `scikit-image`. The result of the holes filling process can be seen in Figure 4.9 where (a) and (b) are slices before and after the process, respectively.

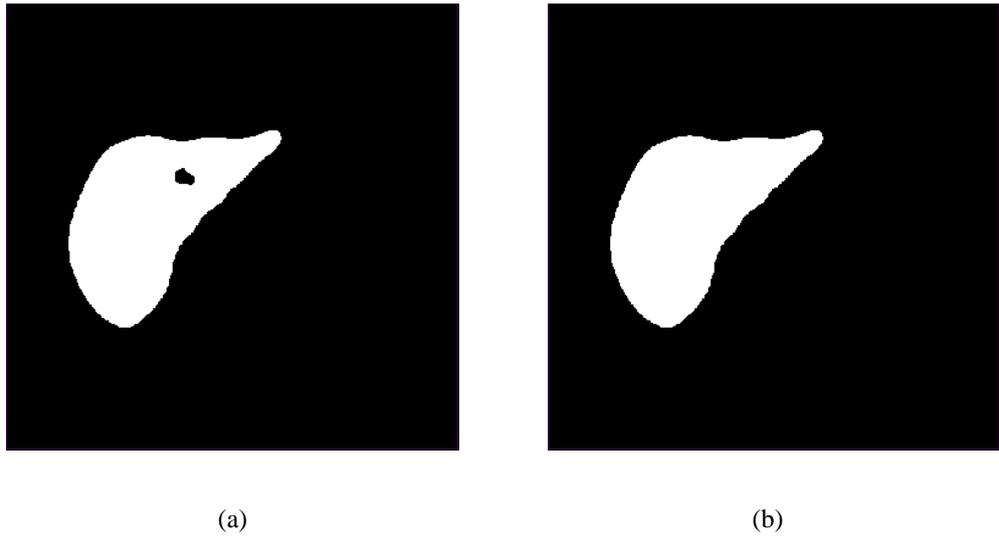


Figure 4. 9: Show the result of the Holes Filling process.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 INTRODUCTION

The summary of the proposed Unet Model is well described in this chapter; the results and experimental setup are also described.

5.2 EXPERIMENTAL SETUP

The Liver Segmentation model were implemented and run on a laptop with Intel Core(TM) i7-6500U processor and CPU @ 2.59 GHz with 8.00 GB RAM and 64-bit operating system. The implementation is done using Python Programming Language and OpenCV library for programming functions with Keras and TensorFlow open-source neural-network library for the implementation of deep learning techniques.

5.3 PROPOSED UNET MODEL

The proposed Unet Model architecture is based on the U-net architecture described in section 4.2 with some differences which are that each convolutions layer in our proposed network is followed by Batch Normalization layer and it consists of 19 convolution layers, 18 BatchNormalization layers, 4 max-pooling layers, 4 concatenate layers, 4 dropout layers and 4 Transposed convolution layers (Deconvolution layer) as well as a softmax layer.

5.3.1 TRAINING AND VALIDATION

The MICCAI and 3DIRCAD datasets used in the experimental of the proposed Unet Model. The experiments were configured with processor Intel Core(TM) i7-6500U, 8.00

GB memory. For training the network, we employed an Adam optimizer with a learning rate of $1e-3$. For the MICCAI dataset, the network trained for 25 epochs with a batch size of 41 as shown in the accuracy and loss curves in Figure.5.1, and for the 3DIRCAD dataset, the network trained for 250 epochs with a batch size of 41 as shown in the accuracy and loss curves in Figure 5.2.

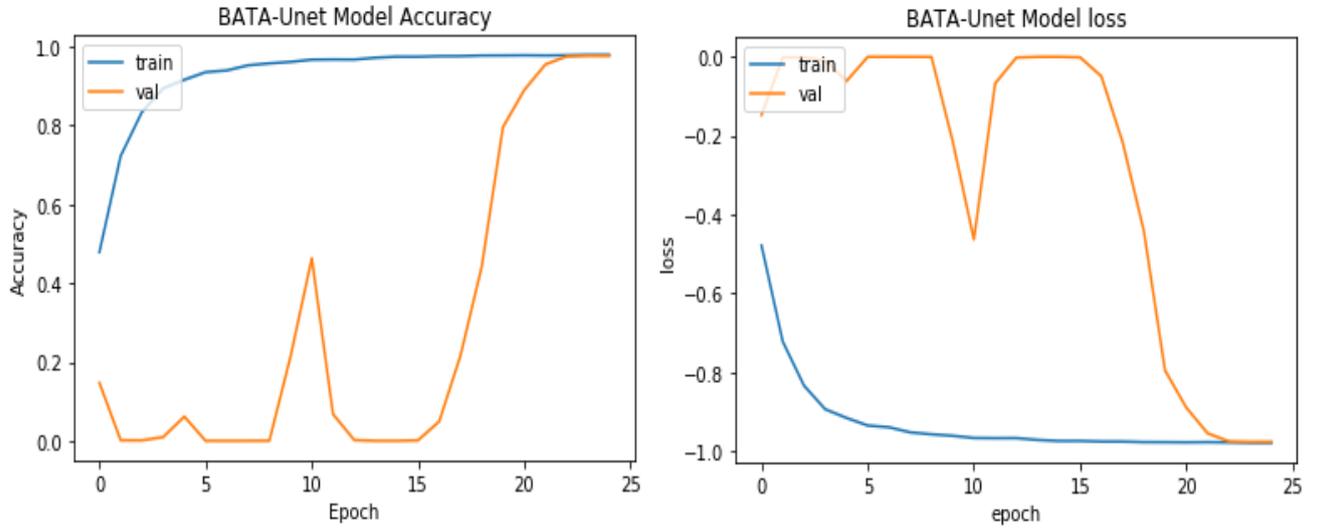


Figure 5. 1: Training, validation loss and accuracy curves for MICCAI dataset

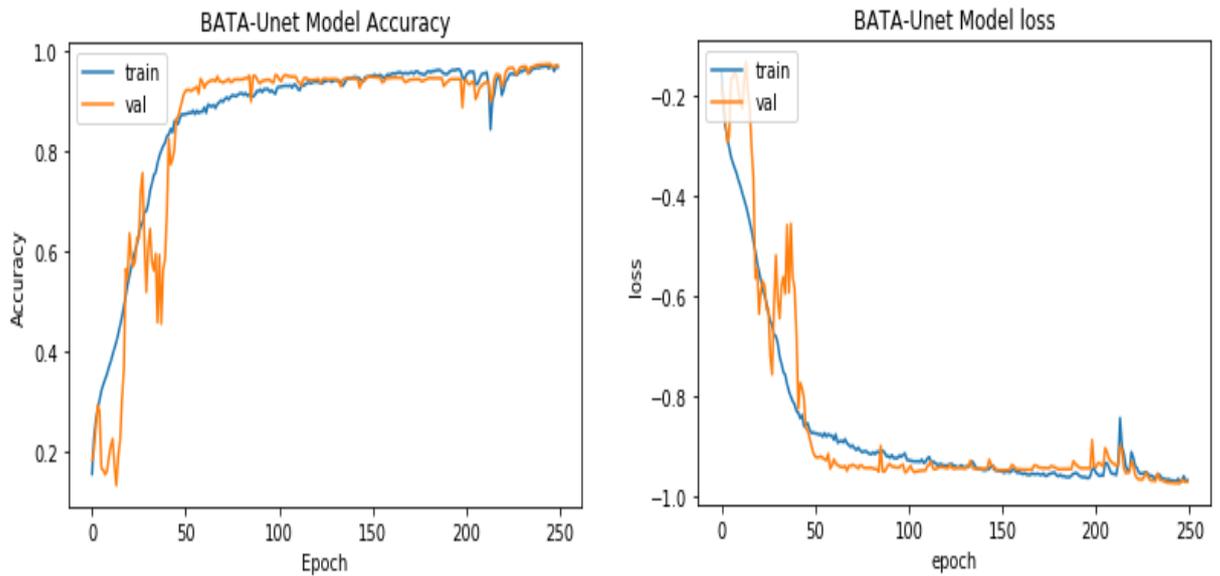


Figure 5. 2: Training ,validation loss and accuracy curves for 3DIRCAD dataset

5.3.2 TESTING PROPOSED UNET MODEL

The Dice Similarity Coefficient (Dice), Volume Overlap Error (VOE), Relative Volume Difference (RVD), Average Symmetric Surface Distance (ASD), And Maximum Surface Distance (MSD) utilized to evaluate the proposed Unet model. After training the model, results of some samples from MICCAI and 3D-IRCAD datasets shown in Table 5.1 and Table 5.2, respectively. The average results using MICCAI are 97.88%, 4.50%, 0.04%, 0.05mm, and 0.08mm for Dice, VOE, RVD, ASD, and MSD, respectively. In addition, the average results using 3D-IRCAD are 96.71%, 11.50%, 0.08%, 0.14mm, and 0.16mm for Dice, VOE, RVD, ASD, and MSD, respectively.

Table 5.1: Results of proposed Unet Model for MICCAI dataset.

#image	Dice	VOE	RVD	ASD	MSD
0	0.976	8.197	0.089	0.082	0.082
1	0.981	3.789	0.028	0.033	0.033
2	0.986	2.475	0.010	0.017	0.017
3	0.982	3.762	0.032	0.035	0.035
4	0.987	1.913	0.019	0.019	0.019
5	0.983	2.691	0.018	0.023	0.023
6	0.979	3.805	0.031	0.034	0.034
7	1.040	2.933	0.030	0.029	0.029
8	0.985	4.607	0.048	0.046	0.046
9	0.993	1.802	0.017	0.017	0.017
10	0.987	2.335	0.005	0.014	0.014
11	0.978	6.650	0.071	0.066	0.066

12	0.981	3.614	-0.001	0.018	0.019
13	0.986	3.160	0.026	0.028	0.028
14	0.983	3.031	-0.005	0.013	0.022
15	0.986	3.704	0.038	0.037	0.037
16	0.992	1.478	0.007	0.011	0.011
17	0.989	2.096	0.017	0.019	0.019
18	0.288	85.000	-0.231	0.965	2.001
19	0.981	3.438	0.023	0.028	0.028
20	0.992	2.070	0.013	0.017	0.017
21	0.990	2.126	0.008	0.015	0.015
22	0.988	2.867	0.025	0.027	0.027
23	1.076	4.134	0.043	0.042	0.042
24	0.980	4.097	0.006	0.024	0.024
25	0.977	5.285	0.052	0.060	0.060
26	0.990	2.698	0.028	0.027	0.027
27	0.976	5.128	0.029	0.040	0.124
28	0.992	2.130	0.018	0.020	0.020
29	0.993	1.924	0.014	0.017	0.017
30	0.988	2.415	0.009	0.017	0.017
Average Result for MICCAI	97.88%	4.50%	0.04mm	0.05mm	0.08mm

Table 5. 2: Results of proposed Unet model for 3D-IRCAD dataset.

#image	Dice	VOE	RVD	ASD	MSD
0	0.802	34.783	-0.100	0.167	0.250
1	0.953	9.589	-0.069	0.015	0.083
2	0.923	14.420	-0.103	0.025	0.230
3	0.976	8.071	0.065	0.091	0.091
4	0.986	5.121	0.046	0.050	0.050
5	0.975	7.103	0.036	0.057	0.057
6	1.033	25.358	0.340	0.370	0.370
7	0.982	7.858	0.068	0.078	0.078
8	0.980	8.511	0.079	0.082	0.082
9	0.978	4.374	-0.007	0.019	0.027
10	0.964	29.870	0.426	0.509	0.509
11	0.987	5.735	0.051	0.060	0.060
12	0.980	3.923	-0.016	0.013	0.028
13	0.967	8.979	0.032	0.064	0.064
14	0.970	9.019	0.078	0.098	0.098
15	0.951	16.026	0.191	0.175	0.175

16	0.980	6.898	0.040	0.061	0.061
17	0.956	8.434	0.038	0.062	0.062
18	0.978	4.294	-0.008	0.018	0.028
19	0.982	6.776	0.055	0.067	0.067
20	0.969	5.990	0.019	0.040	0.040
21	0.983	5.986	0.059	0.152	0.152
22	0.973	7.456	0.067	0.079	0.079
23	0.855	54.286	1.000	0.608	0.608
24	0.970	5.938	-0.038	0.012	0.105
25	0.853	26.316	0.129	0.229	0.229
26	0.971	8.964	0.062	0.080	0.080
27	0.985	4.912	0.041	0.047	0.047
28	0.967	9.593	0.066	0.084	0.084
29	0.964	10.285	0.078	0.163	0.163
30	0.976	7.706	0.031	0.057	0.057
Average Result for 3D-IRCAD	96.71%	11.50%	0.08mm	0.14mm	0.16mm

The results are compared with some of the related works including Cascaded U-net (Christ et al., 2017a), EDCNN (Budak et al., 2020), H-DenseUNet (Li et al., 2018), and mU-net (Seo et al., 2020) as shown in Table 5.3. It was obvious from Table 5.3 that our model is outperformed the other state-of-the-art model for Dice evaluation metrics which equals 97.88% when tested using the MICCAI database and it equals 96.71% when tested using the 3D-IRCAD database. Also for VOE evaluation metrics, it is equal to 4.50% using MICCAI and 11.50% using 3DIRCAD.

For RVD evaluation metric, which it is equal to 0.04% using MICCAI database and it equals to 0.08% using 3DIRCAD database. Also our proposed model is outperformed the other state-of-the-art model for ASD evaluation metric which equals 0.05mm when tested using the MICCAI database and it equals 0.14mm when tested using the 3D-IRCAD database and 0.08mm for MSD using MICCAI and 0.16mm using 3DIRCAD.

Table 5. 3: Comparison between proposed Unet model and some of related works

Related Work	Dice	VOE	RVD	ASD	MSD
Cascaded U-net (Christ <i>et al.</i> , 2017)	93.1	12.8	-3.3	2.3	46.7
Cascaded U-net + 3D CRF (Christ <i>et al.</i> , 2017)	94.3	10.7	-1.4	1.5	24.0
EDCNN (Budak <i>et al.</i> , 2020)	95.22	9.05	7.03	1.43	19.37
H-DenseUNet (Li <i>et al.</i> , 2018)	94.7	10.02	-0.01	4.06	9.63
mU-net (Seo <i>et al.</i> , 2020)	96.01	9.73	0.38	3.11	9.20
Result for MICCAI	97.88%	4.50%	0.04%	0.05mm	0.08mm
Result for 3D-IRCAD	96.71%	11.50%	0.08%	0.14mm	0.16mm

Figure.5.3 and Figure.5.4 are showing some of the Segmentation results for the proposed Unet model using MICCAI and 3DIRCAD respectively. The left column shows original image slices, the middle column shows the ground truth mask and the right column shows the segmentation map obtained using the proposed Unet model.

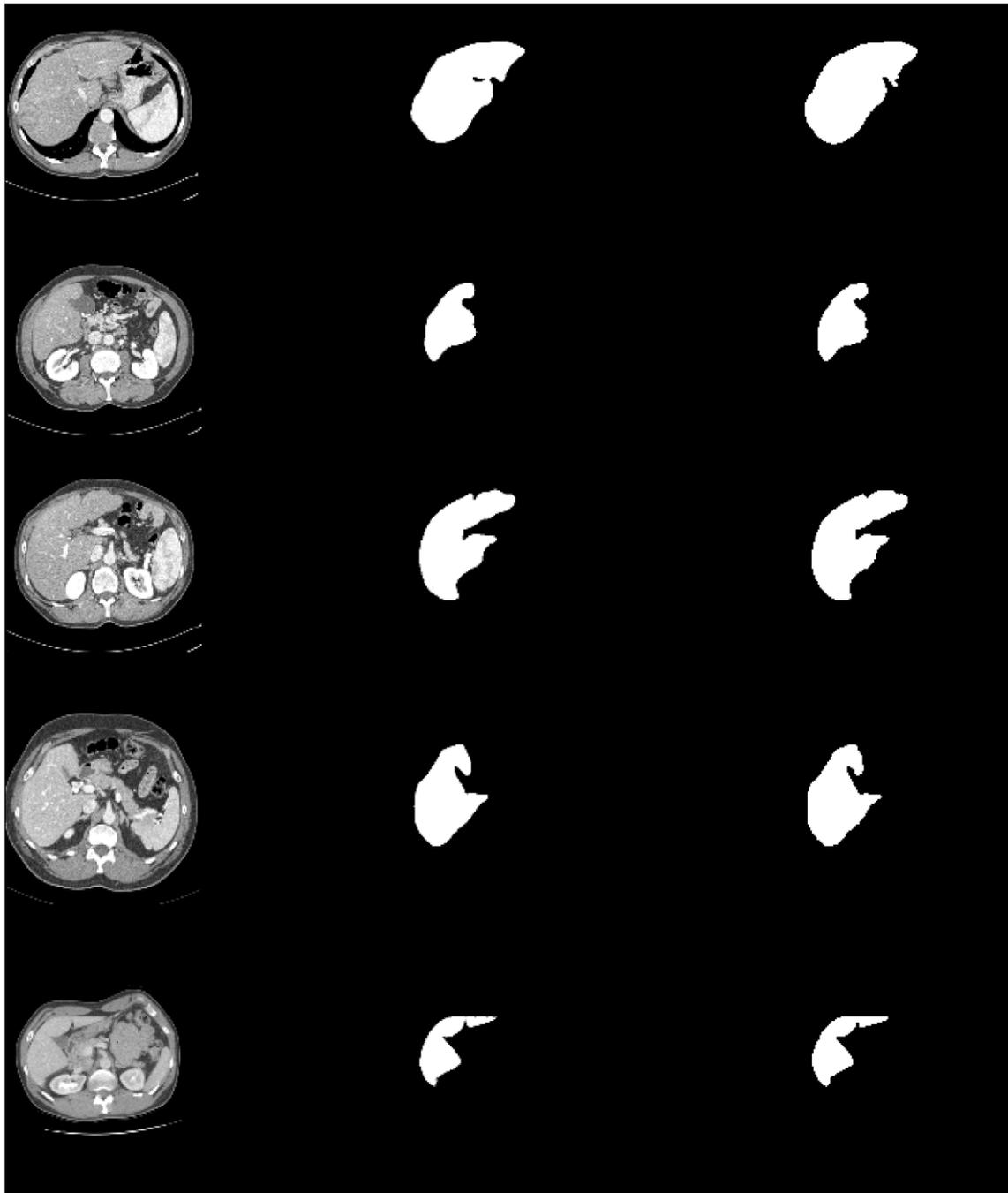


Figure 5. 3: Segmentation results of the proposed Unet model for MICCAI dataset.

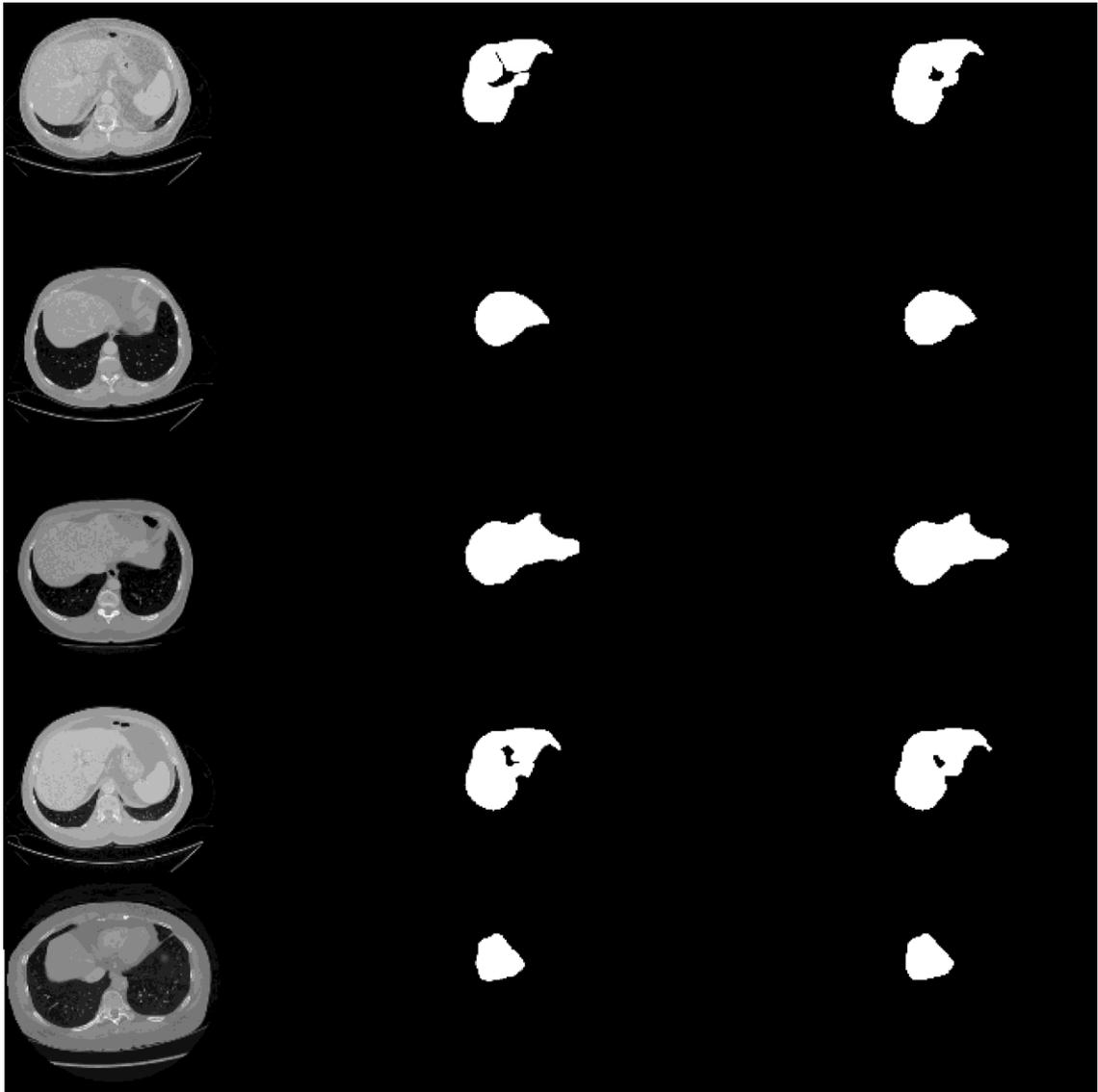


Figure 5. 4: Segmentation results of the proposed Unet model for 3DIRCAD dataset.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 INTRODUCTION

In this chapter, we conclude the thesis. In Section 6.2, we describe the background of this research work, the objective, the contribution, and list the results obtained by this thesis. Section 6.3 discusses limitations and ideas about improvement and in Section 6.4 discusses the future work.

6.2 SUMMARY

The Computer-aided diagnosis (CAD) systems in the medical field are the most advanced expertise and intelligence systems in the interface of medicine and computer science. one of the CAD systems processes is the Segmentation process is considered the most important task in computer vision one of the most recognized research areas in medical image analysis. When having a good segmentation approach may lead to a more objective diagnostic performance. If segmentation is insufficient, then a CAD system may misdiagnose. Many types of methods and algorithms developed to handle segmentation in medical imaging (Yanase and Triantaphyllou, 2019).

Liver segmentation from abdominal CT scan images is the main goal of this thesis, which is a key task for many clinical applications. The segmentation of the liver is a very challenging task due to the complexity of the liver surface, variation in liver and tumor size throughout the CT image slices, ambiguity in boundaries of like intensity tissues and nearby organs. Automatic methods for Liver segmentation is reduce time, effort and provide quality assistance to experts.

In this research work, a U-net model for liver segmentation proposed for enhances the accuracy of the segmentation process. The model is implemented using Python Programming Language and OpenCV library for programming functions with Keras and TensorFlow Open-Source Neural-Network library.

The Methodology of the model is consists of four steps: First is pre-processing the CT images. The second step is training the models for liver segmentation. Then, testing the trained networks produces probability maps as subject-specific prior, which assigns each pixel the likelihood of being the liver for the target image. The last step is the post-processing step to maximize the results efficiency of each model.

Based on the U-net architecture described in section 4.4 we built our proposed model, it consists of 19 convolution layers, 18 BatchNormalization layers, and 4 max-pooling layers, 4 concatenate layers, 4 dropout layers, and 4 Transposed convolution layers (Deconvolution layer). All convolutions layer in our proposed network followed by Batch Normalization layer, which is responsible for normalizing the input values going into each layer by applying the transformation.

MICCAI and 3D-IRCAD datasets used in the experimentation and evaluation of the model. The average results obtained by the U-net model using MICCAI are 97.88%, 4.50%, 0.04%, 0.05mm and 0.08mm. In addition, results using 3DIRCAD are 96.71%, 11.50%, 0.08%, 0.14mm and 0.16mm. For this evaluation metric respectively: DICE, VOE, RVD, ASD, and MSD.

In comparison, we compared it with some of the related works including Cascaded U-net (Christ et al., 2017a), EDCNN (Budak et al., 2020), H-DenseUNet (Li et al., 2018), and mU-net (Seo et al., 2020). It was obvious from Table 5.3 that our model is outperformed the other state-of-the-art model for Dice evaluation metrics which equals 97.88% when tested using the MICCAI database and it equals 96.71% when tested using the 3D-IRCAD database. Also for VOE evaluation metrics, it is equal to 4.50% using MICCAI and 11.50% using 3DIRCAD.

6.3 CONCLUSION

After trying many experiments for the architecture of our network, we come out with proposing to employment of deep learning especially the Unet network, which has been the most promising method for the accurate segmentation of a liver.

We exploit the benefit of the Unet architecture and propose a new method for liver segmentation by applying the concept of the Batch Normalization process. According to Ioffe and Szegedy (Ioffe and Szegedy, 2015), the Batch Normalization is increase the constancy of the network, which normalizes the inputs to a layer in the network by applying a transformation that subtracting the batch mean and dividing by the batch standard deviation. It has accelerated the training process of the network and has improved the performance of the model. We proposed a Unet model based on the U-net architecture by adding the Batch Normalization layer after each convolutions layer.

We compare it with other state-of-the-art models and it has outperformed them. We concluded that the two presented models take a significant step in providing discernment in the field of medical image processing for automatic liver segmentation.

6.4 FUTURE WORK

In the future, we recommend increasing the number of medical images in the dataset used for the training of the convolutional network models. In addition, using some other features such as 3D Convolutional Neural Networks known as Volumetric CNNs in trained and tested the model. In addition, we recommend the use of the same model to detect both liver and the tumors that reside at the liver boundary.

REFERENCES

- Agrawal, S. (2014) ‘Survey on Image Segmentation Techniques and Color Models’, 5(3), pp. 3025–3030.
- Albawi, S., Mohammed, T. A. and Al-Zawi, S. (2018) ‘Understanding of a convolutional neural network’, *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, 2018–Janua(August), pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.
- Barclay, T. (2019) *Liver – Anatomy and Function of the Human Liver*.
- Budak, Ü. *et al.* (2020) ‘Cascaded deep convolutional encoder-decoder neural networks for efficient liver tumor segmentation’, *Medical Hypotheses*, 134(September 2019). doi: 10.1016/j.mehy.2019.109431.
- Campadelli, P., Casiraghi, E. and Esposito, A. (2009) ‘Liver segmentation from computed tomography scans: A survey and a new algorithm’, *Artificial Intelligence in Medicine*, 45(2–3), pp. 185–196. doi: 10.1016/j.artmed.2008.07.020.
- Christ, P. F. *et al.* (2016) ‘Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9901 LNCS, pp. 415–423. doi: 10.1007/978-3-319-46723-8_48.
- Christ, P. F. *et al.* (2017) ‘Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks’, pp. 1–20. Available at: <http://arxiv.org/abs/1702.05970>.
- Fujita, H. *et al.* (2008) ‘Computer-aided diagnosis: The emerging of three CAD systems induced by Japanese health care needs’, *Computer Methods and Programs in Biomedicine*, 92(3), pp. 238–248. doi: 10.1016/j.cmpb.2008.04.003.

Glasbey, C.A. and Horgan, G. W. (1995) 'Chapter 4', *Image Analysis for the Biological Sciences*, pp. 1–31.

Gotra, A. *et al.* (2017) 'Liver segmentation: indications, techniques and future directions', *Insights into Imaging*, 8(4), pp. 377–392. doi: 10.1007/s13244-017-0558-1.

Heimann, T., Wolf, I. and Meinzer, H.-P. (2006) 'Active shape models for a fully automated 3D segmentation of the liver--an evaluation on clinical data.', *Proc. Medical Image Computing and Computer-Assisted Intervention*, 9(Pt 2), pp. 41–8. doi: 10.1007/11866763_6.

Hermoye, L. *et al.* (2005) 'Liver Segmentation in Living Liver Transplant Donors: Comparison of Semiautomatic and Manual Methods', *Radiology*, 234(1), pp. 171–178. doi: 10.1148/radiol.2341031801.

Hung-II, S. (2017) 'Chapter 1 - An Introduction to Neural Networks and Deep Learning', *Deep Learning for Medical Image Analysis*, pp. 3–24.

Johns Hopkins Medicine (2019).

Lakhani, P. *et al.* (2018) 'Hello World Deep Learning in Medical Imaging'. *Journal of Digital Imaging*.

Lawrence M. Davis, M. (2019) *CT Scan (CAT Scan) Procedure Side Effects, Purpose, CT vs.*

Lecun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', *Nature*, 521(7553), pp. 436–444. doi: 10.1038/nature14539.

Li, X. *et al.* (2018) 'H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes', *IEEE Transactions on Medical Imaging*, 37(12), pp. 2663–2674. doi: 10.1109/TMI.2018.2845918.

Litjens, G. *et al.* (2017) 'A Survey on Deep Learning in Medical Image Analysis', (February). doi: 10.1016/j.media.2017.07.005.

- Liu, T. *et al.* (2015) ‘Implementation of Training Convolutional Neural Networks’. Available at: <http://arxiv.org/abs/1506.01195>.
- Long, J., Shelhamer, E. and Darrell, T. (2015) ‘Fully convolutional networks for semantic segmentation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07–12–June, pp. 3431–3440. doi: 10.1109/CVPR.2015.7298965.
- Lu, F. *et al.* (2016) ‘Automatic 3D liver location and segmentation via convolutional neural networks and graph cut’. doi: 10.1007/s11548-016-1467-3.
- Massoptier, L. and Casciaro, S. (2008) ‘A new fully automatic and robust algorithm for fast segmentation of liver tissue and tumors from CT scans’, *European Radiology*, 18(8), pp. 1658–1665. doi: 10.1007/s00330-008-0924-y.
- Mathworks (2017) ‘Introducing Deep Learning with MATLAB’, *Introducing Deep Learning with MATLAB*, p. 15. doi: 10.1109/ISBI.2018.8363547.
- Mazurowski, M. and Buda, M. (2018) ‘Deep learning in radiology : an overview of the concepts and a survey of the state of the art’, (March). doi: 10.1002/jmri.26534.
- Nakayama, Y. *et al.* (2006) ‘Automated hepatic volumetry for living related liver transplantation at multisection CT.’, *Radiology*, 240(3), pp. 743–748. doi: 10.1148/radiol.2403050850.
- P, J. A. (2015) ‘Computer Vision and Medical Image Processing ’, pp. 152–159.
- Pratt, W. K. (2000) ‘Image Segmentation’, *Computer Vision*, pp. 579–622. doi: 10.1002/9780470097434.ch17.
- Seo, H. *et al.* (2020) ‘Modified U-Net (mU-Net) with Incorporation of Object-Dependent High Level Features for Improved Liver and Liver-Tumor Segmentation in CT Images’, *IEEE Transactions on Medical Imaging*. IEEE, 39(5), pp. 1316–1325. doi: 10.1109/TMI.2019.2948320.

Ronneberger, O., Fischer, P. and Brox, T. (2015) ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’, *Miccai*, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.p

Zhang, X. Z. X. *et al.* (2010) ‘Automatic liver segmentation using a statistical shape model with optimal surface detection’, *Biomedical Engineering, IEEE Transactions on*, 57(10), pp. 2622–2626. doi: 10.1109/TBME.2010.2056369.

APPENDIX A - RESEARCH CODES

1- Architecture of the Proposed U-Net Model

```
1 from keras.layers import *
2 #BATA-UNet
3 def Batc_unetmodel(optimizer, loss_metric, metrics, lr=1e-3):
4     inputs = Input((128, 128, 1))
5     conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(inputs)
6     conv1 = BatchNormalization()(conv1)
7     conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv1)
8     conv1 = BatchNormalization()(conv1)
9     pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
10    drop1 = Dropout(0.5)(pool1)
11
12    conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(drop1)
13    conv2 = BatchNormalization()(conv2)
14    conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv2)
15    conv2 = BatchNormalization()(conv2)
16    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
17    drop2 = Dropout(0.5)(pool2)
18
19    conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(drop2)
20    conv3 = BatchNormalization()(conv3)
21    conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv3)
22    conv3 = BatchNormalization()(conv3)
23    pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)
24    drop3 = Dropout(0.3)(pool3)
25
26    conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(drop3)
27    conv4 = BatchNormalization()(conv4)
28    conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv4)
29    conv4 = BatchNormalization()(conv4)
30    pool4 = MaxPooling2D(pool_size=(2, 2))(conv4)
31    drop4 = Dropout(0.3)(pool4)
32
33
34    conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(drop4)
35    conv5 = BatchNormalization()(conv5)
36    conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(conv5)
37    conv5 = BatchNormalization()(conv5)
38
39    up6 = concatenate([Conv2DTranspose(256, (2, 2), strides=(2, 2), padding='same')
40    conv6 = Conv2D(256, (3, 3), activation='relu', padding='same')(up6)
41    conv6 = BatchNormalization()(conv6)
42    conv6 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv6)
43    conv6 = BatchNormalization()(conv6)
44
45    up7 = concatenate([Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same')
46    conv7 = Conv2D(128, (3, 3), activation='relu', padding='same')(up7)
47    conv7 = BatchNormalization()(conv7)
48    conv7 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv7)
49    conv7 = BatchNormalization()(conv7)
50
51    up8 = concatenate([Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same')
52    conv8 = Conv2D(64, (3, 3), activation='relu', padding='same')(up8)
53    conv8 = BatchNormalization()(conv8)
54    conv8 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv8)
55    conv8 = BatchNormalization()(conv8)
56
57    up9 = concatenate([Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same')
58    conv9 = Conv2D(32, (3, 3), activation='relu', padding='same')(up9)
59    conv9 = BatchNormalization()(conv9)
60    conv9 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv9)
61    conv9 = BatchNormalization()(conv9)
62
63    conv10 = Conv2D(1, (1, 1), activation='sigmoid')(conv9)
64
65    model = Model(inputs=[inputs], outputs=[conv10])
66
67    model.compile(optimizer=optimizer(lr=lr), loss=loss_metric, metrics=metrics)
68
69    return model
```

2- Training the Proposed U-Net model

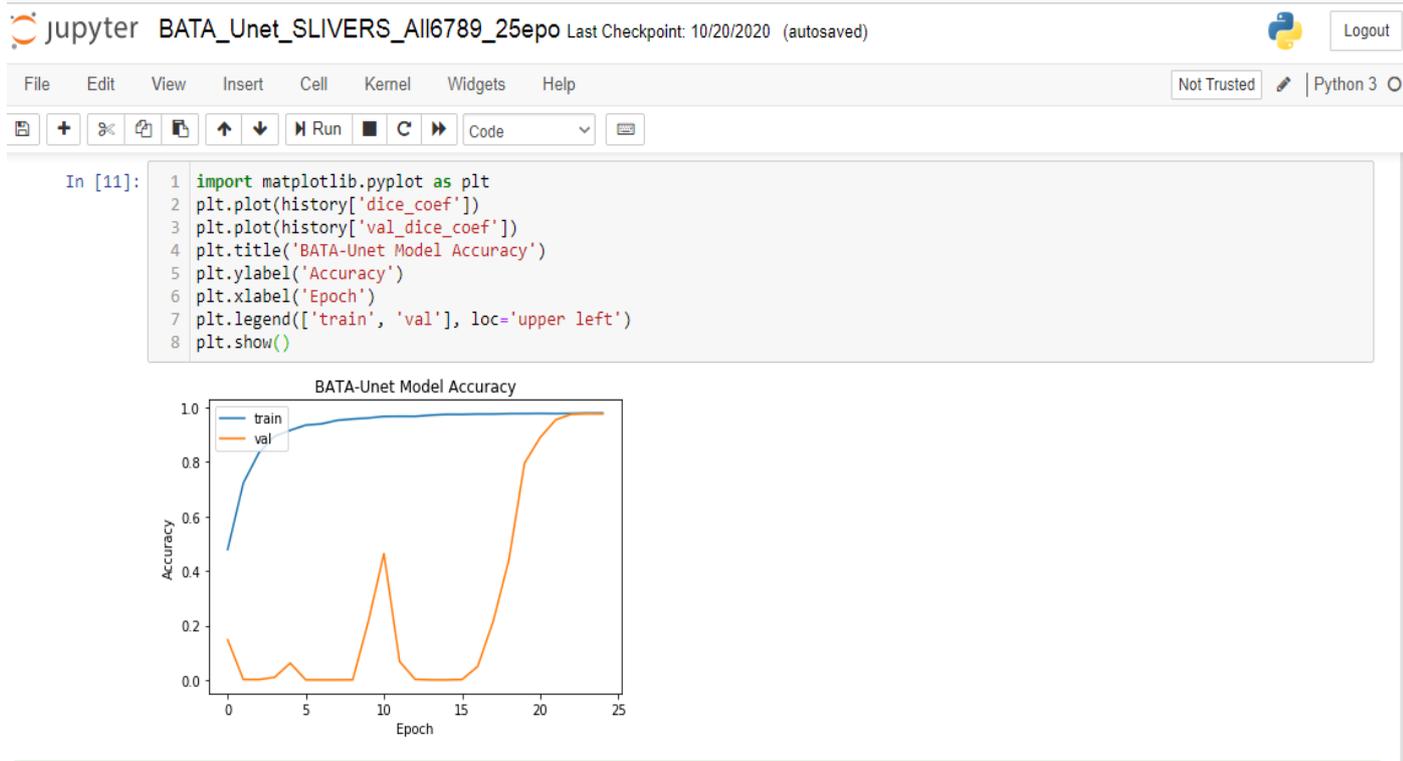
```
jupyter BATA_Unet_SLIVERS_All6789_25epo Last Checkpoint: 10/20/2020 (unsaved changes) Python 3 O
```

```
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 O
```

```
In [10]: 1 hist=BATA_Unets.fit(train_x, train_y, batch_size=41,
2         epochs=25, verbose=1, shuffle=True, validation_split=0.2,
3         callbacks=[model_checkpoint, model_reset])
```

```
Train on 1320 samples, validate on 330 samples
Epoch 1/25
1320/1320 [=====] - 2232s 2s/step - loss: -0.4786 - dice_coef: 0.4786 - val_loss: -0.1470 - val_dice_c
oef: 0.1470
Epoch 2/25
1320/1320 [=====] - 2372s 2s/step - loss: -0.7220 - dice_coef: 0.7220 - val_loss: -0.0015 - val_dice_c
oef: 0.0015
Epoch 3/25
1320/1320 [=====] - 1834s 1s/step - loss: -0.8334 - dice_coef: 0.8334 - val_loss: -0.0013 - val_dice_c
oef: 0.0013
Epoch 4/25
1320/1320 [=====] - 1211s 917ms/step - loss: -0.8933 - dice_coef: 0.8933 - val_loss: -0.0097 - val_dic
e_coef: 0.0097
Epoch 5/25
1320/1320 [=====] - 1781s 1s/step - loss: -0.9153 - dice_coef: 0.9153 - val_loss: -0.0616 - val_dice_c
oef: 0.0616
Epoch 6/25
1320/1320 [=====] - 2451s 2s/step - loss: -0.9346 - dice_coef: 0.9346 - val_loss: -2.0188e-04 - val_di
ce_coef: 2.0188e-04
Epoch 7/25
1320/1320 [=====] - 2369s 2s/step - loss: -0.9390 - dice_coef: 0.9390 - val_loss: -3.9290e-05 - val_di
ce_coef: 3.9290e-05
Epoch 8/25
1320/1320 [=====] - 2342s 2s/step - loss: -0.9519 - dice_coef: 0.9519 - val_loss: -1.9504e-04 - val_di
ce_coef: 1.9504e-04
Epoch 9/25
1320/1320 [=====] - 2359s 2s/step - loss: -0.9568 - dice_coef: 0.9568 - val_loss: -3.9096e-04 - val_di
ce_coef: 3.9096e-04
Epoch 10/25
1320/1320 [=====] - 2355s 2s/step - loss: -0.9606 - dice_coef: 0.9606 - val_loss: -0.2140 - val_dice_c
oef: 0.2140
Epoch 11/25
1320/1320 [=====] - 2341s 2s/step - loss: -0.9662 - dice_coef: 0.9662 - val_loss: -0.4631 - val_dice_c
oef: 0.4631
Epoch 12/25
1320/1320 [=====] - 2350s 2s/step - loss: -0.9666 - dice_coef: 0.9666 - val_loss: -0.0675 - val_dice_c
oef: 0.0675
Epoch 13/25
1320/1320 [=====] - 2352s 2s/step - loss: -0.9664 - dice_coef: 0.9664 - val_loss: -0.0020 - val_dice_c
oef: 0.0020
Epoch 14/25
1320/1320 [=====] - 2351s 2s/step - loss: -0.9711 - dice_coef: 0.9711 - val_loss: -1.5641e-04 - val_di
ce_coef: 1.5641e-04
Epoch 15/25
1320/1320 [=====] - 2336s 2s/step - loss: -0.9737 - dice_coef: 0.9737 - val_loss: -9.3822e-05 - val_di
ce_coef: 9.3822e-05
Epoch 16/25
1320/1320 [=====] - 2337s 2s/step - loss: -0.9736 - dice_coef: 0.9736 - val_loss: -0.0014 - val_dice_c
oef: 0.0014
Epoch 17/25
1320/1320 [=====] - 2344s 2s/step - loss: -0.9751 - dice_coef: 0.9751 - val_loss: -0.0493 - val_dice_c
oef: 0.0493
Epoch 18/25
1320/1320 [=====] - 2337s 2s/step - loss: -0.9752 - dice_coef: 0.9752 - val_loss: -0.2160 - val_dice_c
```

3- Plotting the accuracy curve



4- Plotting the loss curve



5- Results of proposed Unet Model for MICCAI dataset

jupyter BATA_Unet_SLIVERS_All6789_25epo Last Checkpoint: 10/20/2020 (unsaved changes)



File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3

+ ↶ ↷ ↵ ↴ ↵ ↴ ▶ Run ■ ↺ ▶ Code ⌵ ⌨

```
In [32]: 1 import pandas as pd
2 from IPython.display import display
3 data_df = pd.DataFrame(columns=['Dice','VOE', 'RVD', 'ASD', 'MSD'])
4 cou=0
5 Total_Voe = 0.
6 for i in range(len(imgs_mask_test)):
7     if numpy.count_nonzero(imgs_mask_test[i])!=0:
8         r = dice_coef_2(test_y[i],imgs_mask_test[i])
9         a = vc.overlapError(test_y[i],imgs_mask_test[i])
10        b = vc.ravd(test_y[i],imgs_mask_test[i])
11        c = vc.asd(test_y[i],imgs_mask_test[i])
12        d = maxasd(test_y[i],imgs_mask_test[i])
13        Total_Voe += c
14        data_df.loc[cou] = [format(r, '.3f'),format(a, '.3f'),format(b, '.3f'), format(c, '.3f'),format(d, '.3f')]
15        cou += 1
16
17 avg_avd = Total_Voe/float(len(imgs_mask_test))
18 print(avg_avd)
19 print (len(data_df))
20 pd.options.display.max_columns = None
21 pd.options.display.max_rows=None
22 data_df
```

Out[32]:

	Dice	VOE	RVD	ASD	MSD
0	0.976	8.197	0.089	0.082	0.082
1	0.981	3.789	0.028	0.033	0.033
2	0.986	2.475	0.010	0.017	0.017
3	0.982	3.762	0.032	0.035	0.035
4	0.987	1.913	0.019	0.019	0.019
5	0.983	2.691	0.018	0.023	0.023
6	0.979	3.805	0.031	0.034	0.034
7	1.040	2.933	0.030	0.029	0.029
8	0.985	4.607	0.048	0.046	0.046
9	0.993	1.802	0.017	0.017	0.017
10	0.987	2.335	0.005	0.014	0.014
11	0.978	6.650	0.071	0.066	0.066
12	0.981	3.614	-0.001	0.018	0.019
13	0.986	3.160	0.026	0.028	0.028
14	0.983	3.031	-0.005	0.013	0.022
15	0.986	3.704	0.038	0.037	0.037
16	0.992	1.478	0.007	0.011	0.011
17	0.989	2.096	0.017	0.019	0.019
18	0.288	85.000	-0.231	0.965	2.001
19	0.981	3.438	0.023	0.028	0.028
20	0.992	2.070	0.013	0.017	0.017

6- Segmentation results of the proposed Unet model for MICCAI dataset

Jupyter BATA_Unet_SLIVERS_All6789_25epo Last Checkpoint: 10/20/2020 (autosaved) Python 3

```

File Edit View Insert Cell Kernel Widgets Help
Not Trusted Python 3
+ %< > Run C Code
In [25]: 1 # Displaying some results
2 lines = 14
3 fig, axarr = plt.subplots(lines, 4, figsize=(60,lines*10), sharex=True, sharey=False)
4 s=10
5 output_width, output_height = (128, 128)
6 for i in range(0,lines):
7     axarr[i,0].imshow(test_x[i+s].reshape(output_width, output_height), cmap='gray')
8     axarr[i,1].imshow(test_y[i+s].reshape(output_width, output_height), cmap='gray')
9     axarr[i,2].imshow(imgs_mask_test[i+s].reshape(output_width, output_height), cmap='gray')
10
11     post=getLargestComponent(imgs_mask_test[i+s])
12
13     axarr[i,3].imshow(post.reshape(output_width, output_height), cmap='gray')
14
15
16     axarr[i,2].set_title('Dice Coef = ' + str(dices[i+s]),fontsize=45,ha='center')
17
18     for x in range(3):
19         axarr[i,x].axis('off')
20
21 #fig.suptitle('Dice average = ' + str(avg_dice) ,fontsize=60)
22 fig.tight_layout()
23 fig.subplots_adjust(top=0.975)
24 plt.show()
    
```

Dice Coef = 0.9868900341660739

Dice Coef = 0.9792256310563168

Dice Coef = 0.9808532591787437

Activate Windows
Go to PC settings to activate Windows.

7- Segmentation results of the proposed Unet model for 3D-IRCAD dataset

Jupyter BATA_Unet_SLIVERS_All6789_25epo Last Checkpoint: 10/20/2020 (autosaved) Python 3

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3

Run Code

```
In [25]: 1 # Displaying some results
2 lines = 14
3 fig, axarr = plt.subplots(lines, 4, figsize=(60,lines*10), sharex=True, sharey=False)
4 s=10
5 output_width, output_height = (128, 128)
6 for i in range(0,lines):
7     axarr[i,0].imshow(test_x[i+s].reshape(output_width, output_height), cmap='gray')
8     axarr[i,1].imshow(test_y[i+s].reshape(output_width, output_height), cmap='gray')
9     axarr[i,2].imshow(imgs_mask_test[i+s].reshape(output_width, output_height), cmap='gray')
10
11     post=getLargestComponent(imgs_mask_test[i+s])
12
13     axarr[i,3].imshow(post.reshape(output_width, output_height), cmap='gray')
14
15
16     axarr[i,2].set_title('Dice Coef = ' + str(dices[i+s]),fontsize=45,ha='center')
17
18     for x in range(3):
19         axarr[i,x].axis('off')
20
21 #fig.suptitle('Dice average = ' + str(avg_dice) ,fontsize=60)
22 fig.tight_layout()
23 fig.subplots_adjust(top=0.975)
24 plt.show()
```

