UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY
HONORS PROGRAM

PHẠM MINH KHÔI - NGUYỄN HỒ THĂNG LONG

SEMI-SUPERVISED ORGAN SEGMENTATION IN 3D VOLUME WITH MASK-PROPAGATION REFINEMENT

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

HO CHI MINH CITY, 2022

UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY
HONORS PROGRAM

PHẠM MINH KHÔI - NGUYỄN HỒ THĂNG LONG

SEMI-SUPERVISED ORGAN SEGMENTATION IN 3D VOLUME WITH MASK-PROPAGATION REFINEMENT

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

THESIS ADVISORS:
ASSOC. PROF. TRẦN MINH TRIẾT

ACADEMIC YEAR 2018-2022

COMMENT OF THESIS'S ADVISOR

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Tp HCM, ngày .... tháng .... năm 2022

Giáo viên hướng dẫn

COMMENTS OF THESIS'S REVIEWER

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Khóa luận đáp ứng yêu cầu của Khóa luận cử nhân CNTT.

Tp HCM, ngày .... tháng .... năm 2022

Giáo viên phản biện

ACKNOWLEDGEMENT

First and foremost, we would like to express our respect to our esteemed supervisor Associate Professor Tran Minh Triet, who has always been there for us during our work on this thesis. Thanks to his helpful advice, useful suggestion, continuous encouragement and inspirational motivation, that we are able to have enough strength to complete everything in this work, as well as acquire ourselves an immeasurable level of knowledge. We really do appreciate that.

Secondly, we want to offer our gratitude to the Faculty of Information Technology at the University of Science, VNU-HCM for their dedicated lecturers. Most of the core knowledge in this report originates from all the mandatory courses that are well-delivered by them.

Thirdly, we are beholden to all our friends and colleagues at SELAB, University of Science who are always willing to help us relentlessly. In fact, it would be unlikely for us to develop the content of this thesis without all the constructive discussion and interesting suggestions from them.

Last but not least, with our deepest gratefulness, we are wholeheartedly thankful for our family members, for their immeasurable unconditional love and also infinite support, both mental and physical. They have always been our source of motivation to survive our long-lasting four-year university period and especially on the completion of this thesis.

THESIS PROPOSAL

Thesis title: Semi-Supervised Organ Segmentation for 3D CT Volumes with Mask-Propagation Refinement
Advisor: Assoc. Prof. Trần Minh Triết
Duration: December, 2021 to July, 2022
Student: Phạm Minh Khôi (18120043) - Nguyễn Hồ Thăng Long (18120134)
Content: In organs volume segmentation problem, scanned images from medical equipment are the main source of information that qualitatively support doctors to identify the condition of the patients in order to give better treatment decisions. Nevertheless, this source of data is usually scarce in the number of annotations. Our proposed approach come up with a method of propagating masks between image slices with the least number of annotations while maintaining its effectiveness. Another contribution of ours is the introduction of a labeling tool with the user's ability to iteratively interact (e.g., scribble or click) to refine the results until satisfaction. Our aim is that doctors will be facilitated with valuable analysis software that provides reasonable suggestions for their patient treatment.

Methods:

Our proposed method consists of building a full 2D segmentation pipeline for CT volume medical images, from the rational preprocessing method to the novel two-staged segmentation method and finalize with the user-friendly application. We list our targets as bullet points below:

- In the processing stage, we hope to exploit the expert knowledge of the domain in doing this. With proper data-specific pre-preprocessing method, the medical data should be easier for neural networks to comprehend while providing details that are explainable.
- In the reference model, we aim to employ semi-supervised learning and active learning techniques to utilize unlabeled data based on uncertainty estimation to generate pseudo-labeled data and selectively choose valuable data samples for training.
  As regards to the propagation module, we aim to adapt the Mask Propagation algorithm to further refine the masks from the reference module while inheriting some of its beneficial properties.
- Lastly, we concentrate on deploying a minimal software with user-friendly GUI to apply the proposed segmentation model into practice where doctors and nurses or anyone with medical expertise but lack of programming skills, can easily label patient data by hand.

Expected Results:

- An efficient method for organ segmentation for 3D CT volumes.
- An annotation tool that can be used easily by people without technical knowledge.
- Achieving comparable top rank at FLARE 2022, a challenge of the conference MICCAI 2022

Research timeline:

- **January-February 2022**: Conduct research for recent mask propagation methods and the previously related works. Simultaneously, intensively investigate the basics of medical subjects and the prevalent applications of AI in this field.
- **March 2022**: Implement a code template for training and evaluation, try and adapt baseline code from MiVoS, STCN to toy datasets. Search for suitable opening medical challenges from MICCAI 2022's workshops, results in joining FLARE22.
- **April 2022**: Analyze FLARE22 datasets, experiment splitting, preprocessing, post-processing for the specific 3D volume data, with reference from top solutions of previous challenge. Visualize every step to avoid fatal bugs and errors before fully training.
- **May - June 2022**: Full training and evaluating models while constantly suggesting and implementing improvements for the performance models regarding both accuracy and efficiency.
- **July 2022**: Final submission to FLARE 2022. Develop an application integrated with aforementioned features and AI algorithm. Possibly conduct a number of surveys about the software usability.

	May 9th, 2022
	Students
Advisor	
resources/signatures/tmtriet.png	Phạm Minh Khôi
Assoc. Prof. Trần Minh Triết	
	Nguyễn Hồ Thăng Long

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

There is no doubt that artificial intelligence is one of the most talked-about topics of recent interest. And within the scope of AI, a more popular area is its application to healthcare – or more specifically, to extracting information from medical images. This is an urgent problem, as **volumetric CT images** can contain a great deal of valuable information about patients' health. But accurately extracting that information requires sophisticated image processing techniques – which is where AI comes in.

In this thesis, we focus on the problem of **volumetric CT images segmentation**: specifically, **organ segmentation**. We investigate two different approaches to this problem: **semi-supervised learning** and **active learning**. Especially, our method is an improvement of the traditional 2D approach, which is a concept with two components including **references** and **propagation**: references would be supporting the expert and give preliminary results, and propagation will be from the results referenced and predicted across the slices that have partial similarity in terms of slices. The reason behind that is the model could keep the resolution without any complex preprocessing methods; The proposed method has a high adaptive quality, which is proven by the reusability and **unlimited classes prediction ability** from our implementation application.

To use the full potential of unlabelled data, we approach it in two ways as follows. In the references process, we use semi-supervised by designing training models and tactics based on **Cross Pseudo Supervision** strategy, which helps capture information at a general level. In the propagation process, we use the same type of method to force the model to learn visual similarity and domain-specific features, this positive result can be applied in many types of parts or unseen contexts.

In medical images, slices have different importance and influence, for example in CT volume images, slices that are centered and contain more classes will carry more information. For that reason, we propose a training strategy that prioritizes using **active learning** to generate **pseudo labels** with high confidence, which is used to suggest potentially usable unlabelled samples to contribute to training a better model.

Our results show that both methods are effective at accurately identifying organs in CT scans. The result is boosted **from 0.5 to 0.78** and comparable with state-of-the-art methods using 3D approaches. Our method using semi-supervised using active learning outperforms traditional 2D method by a significant margin. Furthermore, we also implement an application to demonstrate the usability and practicality of the thesis.

CHAPTER 1
INTRODUCTION

3D CT volume organ segmentation is a critical task for many medical applications such as image-guided surgery, radiation therapy planning, and 3D printing. The purpose of this chapter is to provide an overview of the 3D CT volume organ segmentation problem and variants scenarios. We first discuss the problem statement and challenges involved in 3D CT volume organ segmentation. Next, we describe some key applications that can benefit from accurate 3D organ segments especially in interactive scenario. After that, we demonstrate our proposed approach overview for efficiency and accurately extracting organs from 3D CT volumes. Finally the thesis structure that we present will be overviewed.

1.1 Overview

Subclinical examination undoubtedly plays an important role in all medical treatment processes. By accurately quantifying human parts, doctors can identify illnesses and abnormalities with much less effort. With the help of deep learning algorithms, human organs can be identified automatically with effectiveness and efficiency; thus enabling doctors for faster diagnoses. But for deep learning agents to achieve high performance, it often comes with a vast amount of high-quality labeled data for the training stage. However, obtaining a sufficient amount of medical data is quite expensive and time-consuming, not to mention the need for medical labels to be evaluated by experts to ensure accuracy for usability. Because of the lack of useful data and scarce medical experts, it makes the problem becomes more challenging to tackle for today's machines. In light of that, new ideas have been devised, including an interactive framework

that enables doctors, or even non-expert ones to have the ability to manually annotate patient data effortlessly. Recently, there have been several promising advances in this area, especially semi-supervised learning algorithms that could potentially be used for CT volume organs segmentation.

1.1.1 CT volume organ segmentation

The goal of computer vision, as far-fetched as it sounds, is to replicate the human understanding level. However with the exponential growth of deep learning nowadays, human ambition is redirecting to a more achievable and narrower target instead: to imitate the expert knowledge in only a certain domain or field.

In order to create a computer vision system that can "understand" an image, we must first figure out what it means for a machine to be able to "perceive" and "analyze" objects in an image. Essentially, we need to develop a way for the computer to identify and understand the various elements that make up an image, as well as the relationships between them. This is no easy task, given the ambiguity and difficulty of interpretation presented in most images. However, by breaking down this problem into smaller pieces and developing appropriate algorithms, we can eventually create a system that is capable of understanding images on its own in a well-defined context.

In the main problems of the computer vision field, including classification, detection, and segmentation, the highest level of information extraction level is the segmentation mission. Since the segmentation task requires providing the results in pixel-level precision, meaning that it both solves classification and object localization in the most detailed way, especially it implies separating overlapping objects to avoid ambiguous cases in detection. In the field of healthcare, solving the task of object segmentation on patients' 3D volumes can supply medical

staffs with lots of useful insights.

In practice, volume object segmentation is a critical task for many applications in medical image analysis. The goal of it is to create another segmentation volume mask that highlights particular areas in the volume. Meaningful insights can be deduced from this mask to help diagnose medical conditions or to improve our understanding. There are many different approaches to volume object segmentation, but all share a common goal: accurately identifying and separating individual objects within a given 3D image volume, as in Figure **??**.

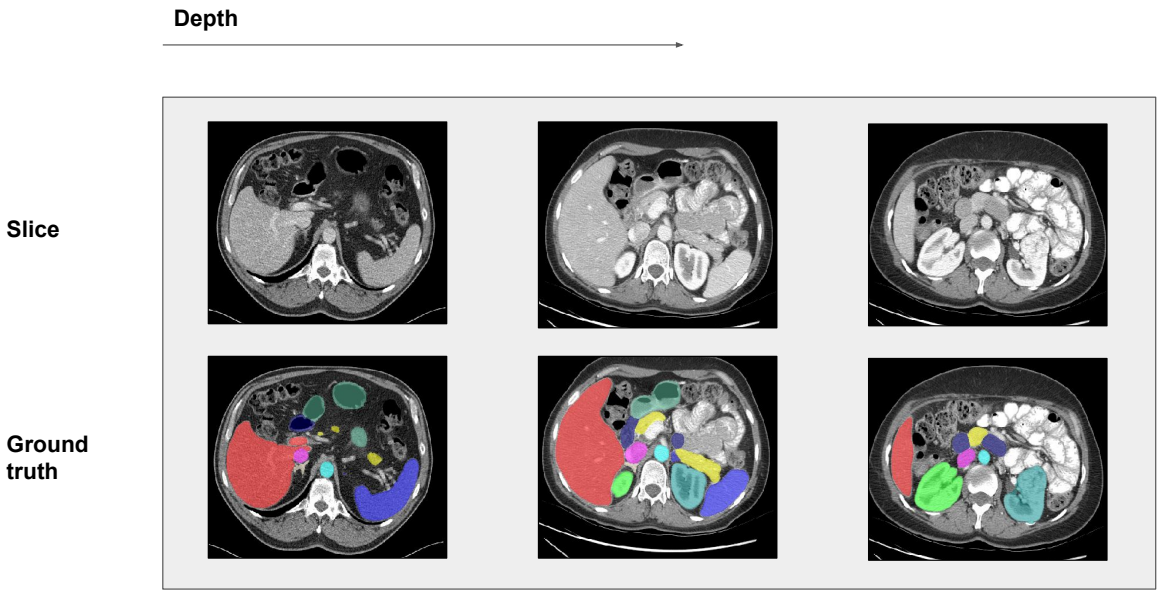

Figure 1.1: Slice-level annotation of volume organs segmentation

In volume object segmentation, there are two main approaches including:

- **Approach by capturing the whole 3D volume:** The state-of-the-art approaches for 3D medical image segmentation are to use a model that has the exact volume and shape of the medical image (figure **??**). This allows for more 3-dimensional relationships instead of just 2 dimensions,

which leads to smoother and more accurate segmentation masks. The model convergence is also faster than techniques that rely on 2d images, and the prediction speed is also satisfactory. However, this approach does have some disadvantages. For example, it highly consumes internal memory resources or needs a complex pre-processing strategy for different configurations like spacing and normalizing. These are sensitive to out-of-distribution data and can lead to terrible predictions.

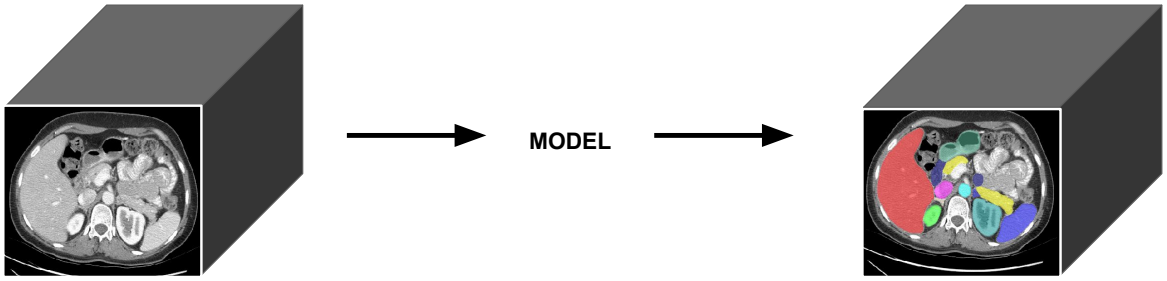

Figure 1.2: The normal pipeline for CT volume object segmentation

- **Approach by looping slice by slice through the volume (figure ??):** for this approach, the analyzing progress is as same as for video sequence data. Since the computational cost is light enough, the image input to the models can keep the original resolution instead of having to be reduced as 3D models usually handle. Furthermore, by looping and keeping temporary memory, the model focuses on just a part of the volume, which makes it avoid redundancy information. The fatal disadvantage of the two-

dimensional approaches to medical images is that they are not possible to see the entire object at once like in 3D space. This causes defects in distantly dissected parts, making it difficult for the model to retrieve information from slices that are far apart. Because of that, the development of 2D models in research for medical images is lacking and not many people try to improve the results. However, we think that this approach has a lot of potentials and can be developed into common behaviors based on doctors' habits or knowledge. We aim to make the model transferable between different domains so that it can be used more effectively. But with our targets, the challenges we facing is also daunting. Since we are treating the 3d volume as a video, and because it often requires attentive handling of object interaction, occlusion, distortion, motion blur, scaling, and tracking through the frame, we also have to handle challenges like being out of view, small objects, suddenly appearing, objects disappearing, and reappearing. The model needs to gain time-spatial understanding to prove more beneficial in many ways.

1.1.2 Interactive CT Volume Organ Segmentation

The process of volume object segmentation is an important tool for analyzing medical images. By labeling and refining the object boundaries interactively, the doctors can get a better understanding of the image data. However, this process can be time-consuming and tedious. A semi-automated process could help to speed things up without sacrificing accuracy. The algorithm would be based on the user's observations to interpolate the remaining frames. This would allow us to quickly obtain a detailed analysis of medical images while still maintaining precision.

As mentioned above, preparing medical ground truth for the supervised algorithm is resource-intensive. That leads to the limitation of algorithms developed

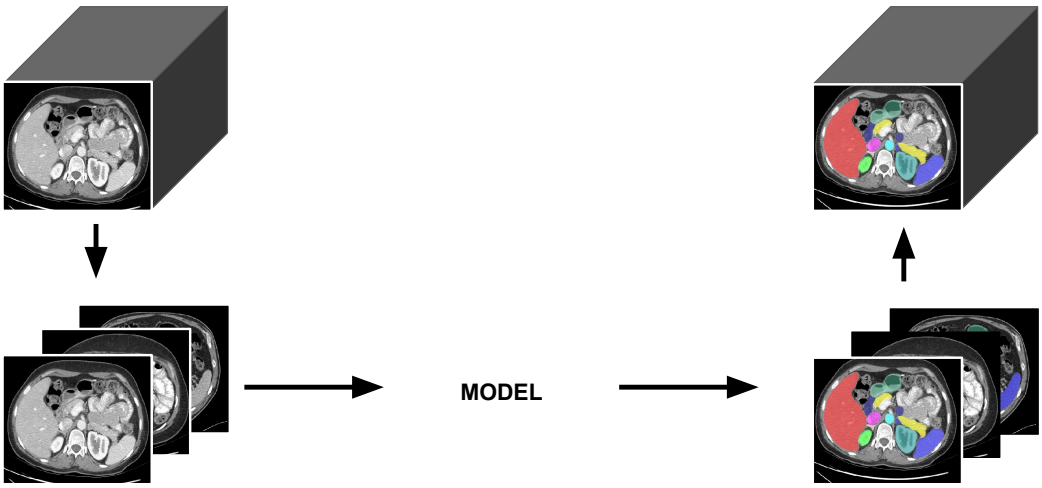

Figure 1.3: 2D pipeline approach for CT volume object segmentation

for medical diagnosis and treatment. Considering the semi-automated context, when the algorithm is still incomplete, semi-automatic labeling is necessary when expert manpower is scarce and takes years of training. Therefore, 3D dataset creation using interactive scenario appears to tackle this problem and speed up the labeling process as human in a loop. Unlabelled data takes a lot of benefits from this appearance.

The process of interactive volume object segmentation is as follows: the doctor, also known as an annotator, selects the most valuable frame and proceeds to label each object in this frame. Based on this data, the model task is to predict the segmentation mask of those objects in the remaining slices of volume by propagating the annotation throughout it. In the next interaction, the annotator selects a slice with the highest uncertainty and provides suggestions as to the corresponding scribbles in this slice. These scribbles indicate positive pixels and negative pixels by marking false positives and false negatives. This interaction between doctors and models allows for more accurate predictions and helps

ensure that results are as accurate as possible. This process is repeated until the annotator is satisfied with the resulting segmentation of the entire volume.

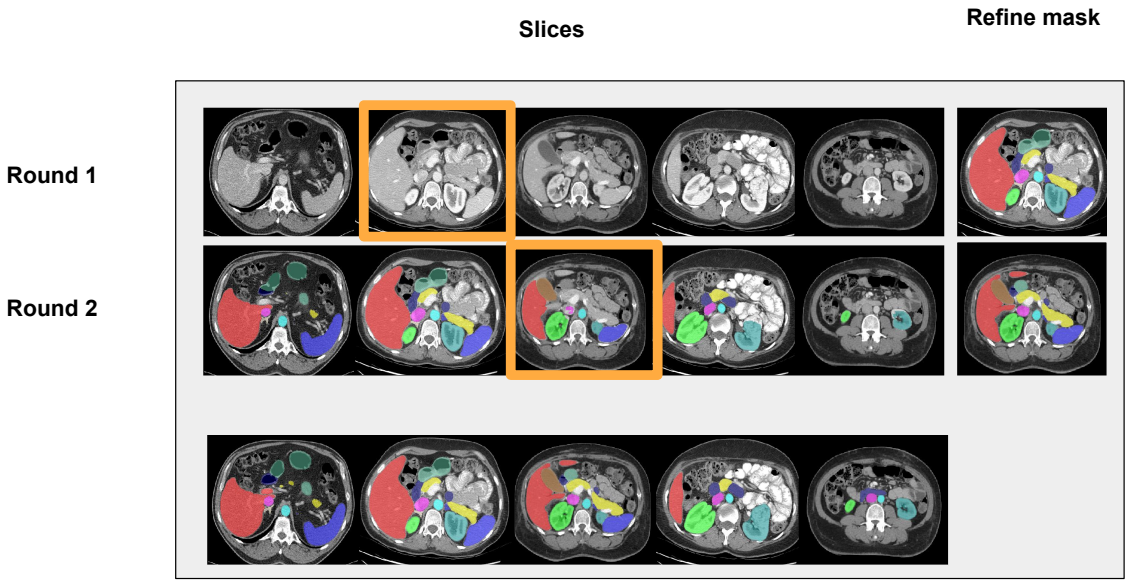

Figure 1.4: Round-based interactive volume organs segmentation.

There are many different ways to approach the interactive volume object segmentation task. However, we believe the most appropriate method must be to appreciate the significance of multiple criteria when making the decision. Speed is important, as we want to produce initial results that satisfy the user. Reasonable quantitative evaluation is also crucial, as it allows us to improve accuracy after annotator interactions.

1.2 Motivations

Segmentation is one of the problems in which the labels for models are scarce because of their high complexity and complication. If we cannot accurately label our data sets, then our models are hardly to be as accurate as we desire. This is a real issue in the medical community, where human resources are often restricted.

It is difficult to generate data when we have to prioritize human resources over machine labor.

The supervised scenario is preferable to the unsupervised scenario in medical data because it is more accurate. The user has to manually annotate the pixel locations of interest with whole volume data. However, the problem occurs that for contiguous frames the difference is not much but the annotator has to time-consuming to perform the annotate action of similar positions on slices with not too different visual features. The unsupervised lack in the low-quality segmentation mask should be limited in terms of results and difficult to put into real life, especially for a field that requires absolute accuracy like medical.

The semi-supervised approach is a great way to take advantage of the benefits of both labeled and unlabeled data. By using this method, we can get more precise quantitative results based on the labeled data. This is important because it allows the model to exploit information from other raw data without any human guide. Additionally, the semi-supervised approach is able to do this while still maintaining accuracy in its results.

Nevertheless, in any scenario above, the process of manually labeling data is slow and inefficient. It can be difficult to reuse previous models, and the adaptability is low. This means that when labeling a new dataset, the process does not take advantage of the power of previous models. As a result, this slows down research and development in the medical field.

Video object segmentation is a pixel-level tracking object through sequence images problem. There are some similarities to when the expert user conducts labeling, although volume data is a 3-dimensional block, when performing labeling, the annotator has to go through each slice and make labeling, but cannot compare. Works conveniently with volume blocks. So now the volume becomes a sequence image and has many similarities with video object segmentation.

Interactively volume object segmentation is a problem that has been proposed recently and has some promising directions for development, but it can be leveraged in the use of techniques that can be applied to problems encountered by medical data.

The points made above motivate us to investigate the potential of interactive segmentation with active learning. Such tools would work with people to find valuable positions for refinement and satisfaction of the result. With a certain number of slices, the tool could achieve results that are as good or even better than those produced by annotators.

1.3 Objectives

The main objective of this thesis is to pioneer the application of the recent state-of-the-art method of semi-supervised video object segmentation into the problem of CT volume segmentation, with the hope of improving effectiveness, efficiency and pragmatism. On top of that, we are ambitious to build an interactive annotation tool integrated with the algorithm that can assist even novice users in precisely marking regions of interest on a 3D CT volume. Overall, the main contribution of this thesis includes:

- Propose an automatic pipeline to segment human organ parts from CT volumes while utilizing an extensive amount of unlabeled data
- Propose a method that combines pseudo-labeling with active learning to plausibly generate usable data for retraining and improving the networks
- Develop an annotation tool with an interactive and user-friendly GUI that adopts our adjusted propagation method, which can reduce human effort and expert-level knowledge requirement
- Propose a way to fuse the 3-dimensional information into 2D architectures,

by using a positional encoder

The detailed work that we have done in this thesis includes:

- Conduct various research about topics of Deep learning in the medical area, especially 3D image segmentation and also works related to Semi-supervised learning and Active learning.
- Participate in FLARE22 Challenge of MICCAI and thoroughly study the given datasets and follow all of the challenge's information such as the requirements, evaluation metrics, and previous top methods.
- Conduct research on building an interactive tool, experiment repeatedly to investigate the tool's usability and user experience.
- Implement and inherit code bases to develop our own solution for the task, and considerably benchmark them to prove their effectiveness.

1.4 Thesis content

This thesis is structured into 7 chapters:

Chapter 1 In chapter 1, we present about the 3D CT volume organ segmentation task and its applications for many medical fields such as image-guided surgery, radiation therapy planning, and 3D printing. The purpose of this chapter is to provide an overview of the 3D CT volume organ segmentation problem and variants scenarios. We first discuss the problem statement and challenges involved in 3D CT volume organ segmentation. Next, we describe some key applications that can benefit from accurate 3D organ segments especially in interactive scenario. After that, we demonstrate our proposed approach overview for efficiency and accurately extracting organs from 3D CT volumes. Finally the thesis structure that we present will be overviewed.

Chapter 2 In chapter 2, we discuss all necessary background knowledge related to our work. First of all, we start with the very beginning concept in machine learning and then introduce some fundamental models used in processing complex data such as time-series or digital images. Then we introduce a list of Computer Vision problems which play an important role in our proposed approach.

Chapter 3 In chapter 3, we introduce the volume organs segmentation problem, inspire from video object segmentation problem and is most related to our work. We also discuss the semi-supervised approaches which are widely used to solve medical image segmentation task and details about the state-of-the-art model that we utilized in our system. Next, we introduce related works which have a same scenario and motivation to solve the interactive segmentation concept. We provide detailed discussions about the Interactive methods and Positional encoding technique which applied directly to our propose architecture.

Chapter 4 In this chapter, we present our solution for tackle the interactive volume object segmentation. Our proposed approach include reference and propagation module. At first, the pipeline of the method is shown. Then, each module is introduced and the detailed implementation is presented. We apply the distillation technique not only in architecture design but also in the training strategy stage. Finally we dive deep in the potential of loss function especially for medical.

Chapter 5 In this chapter, we describe the experiment dataset, evaluation metrics, and challenge platform. Besides, we provide the details configurations and implementation details of our proposed method. Based on that, we also analyze the result with on our observations.

Chapter 6 In this chapter, we present the applications of our proposed method for interactive video volume segmentation, including a annotation tool.

Chapter 7 In this chapter, we report the results and discuss about the future

works for improving our proposed method and its applications. In our thesis, we proposed a novel pipeline to tackle the problem of CT volume organ segmentation along with an interactive tool that can help labeling process become less complicated. Overall, through ablation study, our method shows improvements over the original baseline, yet still have weaknesses. If these shortcomings can be resolved, we believe it can bring many breakthrough for both the deep leaning and medical fields. Having said that, it is left for the future works.

CHAPTER 2
BACKGROUND

In this chapter, we discuss all necessary background knowledge related to our work. First of all, we start with the very beginning concept in machine learning and then introduce some fundamental models used in processing complex data such as time-series or digital images. Then we introduce a list of Computer Vision problems which play an important role in our proposed approach.

2.1 CT Volume Image

CT images are two-dimensional pictures that represent three-dimensional physical objects. The images are made by converting electrical energy (moving electrons) into X-ray photons, passing the photons through an object, and then converting the measured photons back into electrons. The number of X-rays that pass through the object is inversely proportional to the density of the object. Objects imaged by CT consist of parts that vary in density.

Image slices can either be displayed individually or stacked together by the computer to generate a 3D image of the patient that shows the skeleton, organs, and tissues as well as any abnormalities the physician is trying to identify. This method has many advantages make doctors easier to find the exact place where a problem may be located or identification of basic structures as well as possible tumors or abnormalities.

2.1.1 Digital image representation

In machine computing, image is a well-known definition. The image is construct by multiple pixels, each of them contains multiple values representing its visual information. The value range is often between from 0 to 255, for example it

can handle the image brightness or the affect of a specific color (in a colored image). Every image is define by structure information, such as image shape like channel, width, height. If the number of channel is 1, the image should be grayscale. And in color image, the number of channel would be 3 (corresponding with red, green and blue). And there are multiple variants of image type, like medical image shape would be constructed by width, height and depth. The depth channel could be thousand and pixel value range could be a negative number.

2.1.2 Hounsfield Units

The CT detectors measure the degree that the scanned tissues physical density, and the image processor storing the data as pixels calculated by to convert byte data into a range of 5000 values. The scale's range of values is named for Hounsfield; each value on the scale is termed a Hounsfield unit (HU). Densities of various substances have been assigned relative values. The density of the substances in the patient (both natural tissues and any medical implants) and around the patient are calculated based on a linear transformation of the measured **X-ray attenuation coefficients**. This transformation is based on the standard density measurement of two substances, distilled water (set as 0 HU) and air (set as -1000 HU). HU for various scanned tissues are computed from the following equation:

$$HU = 1000 \times (tissue\mu - water\mu)/water\mu \tag{2.1}$$

Where μ is the linear attenuation coefficient. CT scanners used in medical practice can present HU within a range of –1024 HU to +3071 HU. Different publications define different ranges for certain tissues and substances.

2.2 Machine Learning

For the past decades, Machine Learning (ML) has become one of the most powerful tools, allowing individuals to perform complex analyses for better insights. It has been bringing significant benefits in various vital fields, such as business, healthcare, agriculture or traffic management, etc.

The term "Machine Learning" was popularized in 1959 by Arthur Samuel [**?**]. He defined ML as "the field of study that gives computers the ability to learn without being explicitly programmed". In 1997, Mitchell [**?**] provided the definition “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E”. In other words, ML can be explained as automating and improving the learning process of computers based on their experiences without being actually programmed i.e. without any human assistance.

In the area of big data, ML derives insightful information from large volumes of data by leveraging algorithms to identify patterns and learn in an iterative process. The system applies some transformations to the input data to extract frequent or special patterns, which are usually called features, then uses them to predict the output. The performance is then evaluated by appropriate metrics for the given task. Based on these scores, the system can choose the best transformation for each problem among a set of predefined functions (also known as hypothesis space).

In general, machine learning algorithms are divided into three common types: supervised learning, unsupervised learning, and reinforcement learning.

- **Supervised learning** [**?**]: Supervised learning is when the model getting trained on a labeled dataset, which allows the model to learn and grow

more accurate over time. A labeled dataset is one that has both input and output parameters. Supervised machine learning is the mostly-used type today.

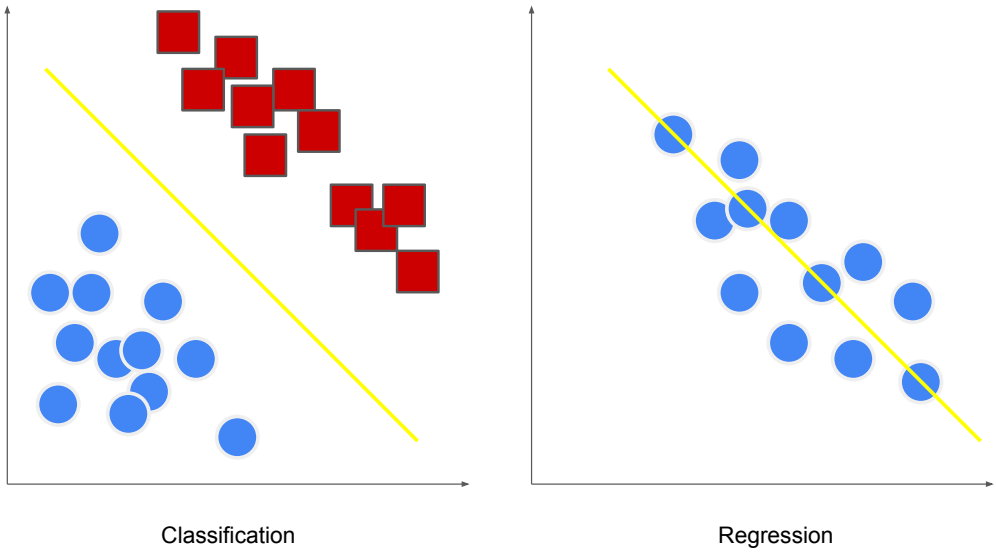

Figure 2.1: Classification task and regression task

Two common types of supervised learning algorithms are classification and regression. Classification is the task of assigning the data into multiple categorical classes. For example, an algorithm would be trained with pictures of dogs and cats, all labeled by humans, and the machine would learn ways to identify pictures of dogs or cats on its own. Meanwhile, regression is the task of distinguishing the data into continuous real values. Examples of this type of task include predicting the house prices from their properties or predicting the age of people based on their face image. Figure **??** demonstrates the difference between the two tasks.

- **Unsupervised learning** [**?**]: The technique analyzes and clusters unlabeled datasets using machine learning algorithms. These algorithms find

hidden patterns and data without any human intervention, i.e., we don't give output to our model. The training model has only input parameter values and it can group unsorted information according to similarities, patterns and differences without any prior training of data.

Two common unsupervised learning algorithms are clustering and association. Clustering techniques are applied to group data based on different patterns that our machine model finds, such as similarities or differences, whereas association rule learning is a rule-based ML technique that finds out some very useful relations between parameters of the large portions of data.

Figure **??** depicts two use cases of unsupervised learning. Clustering can be used to segment customers based on their distinct attributes and association can help suggesting what customers should buy based on what others have bought.

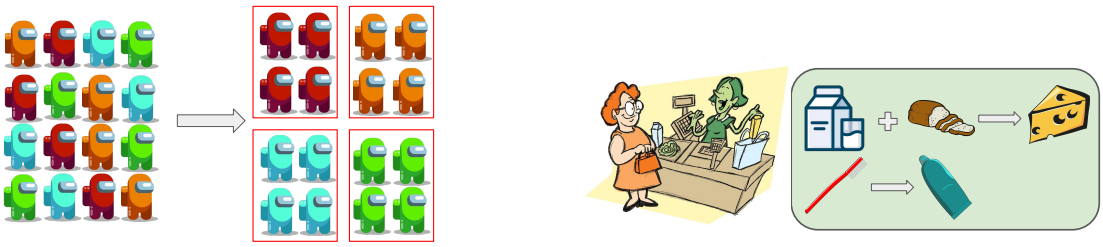

Figure 2.2: Clustering task and Association task

- **Reinforcement learning** [**?**]: In this technique, the model (or the agent) learns the behavior or pattern by interacting with the environment and assessing the reward to increase its performance. The main goal of the agent is to take action in order to maximize the notion of cumulative reward. Reinforcement learning differs from supervised learning in not needing la-

beled input/output pairs to be presented, and in not needing sub-optimal actions to be explicitly corrected. These algorithms are specific to a particular problem e.g. Google Self Driving car [**?**], AlphaGo [**?**] where a bot competes with humans and even itself to get better and better performers in Go Game. Each time we feed in data, they learn and add the data to their knowledge. So, the more it learns the better it gets trained and hence experienced.

Looking at the Figure **??**, the agent is assigned a mission to obtain the reward while minimizing the path cost. Each time it reaches the reward, it learns a new behaviour and optimize its own path for the following iteration.

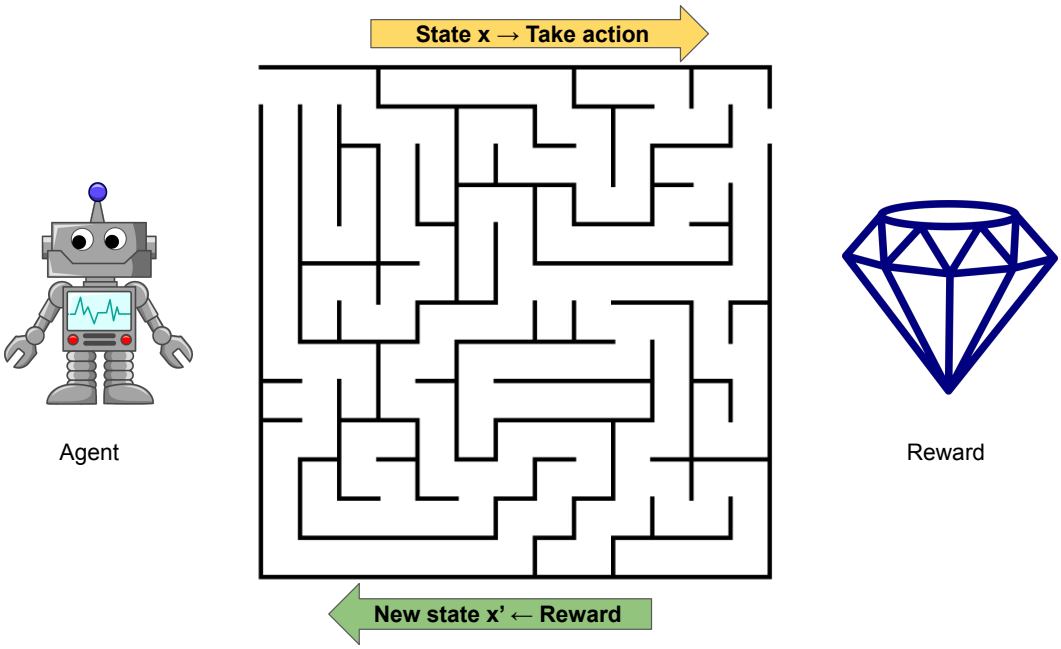

Figure 2.3: An example for Reinforcement learning. The agent is tasked to acquire the reward while searching for the most optimized track.

2.3 Deep Learning and Neural Network

2.3.1 Overview

As a subfield of machine learning, deep learning is formed as complex neural network architectures, containing more processing layers to learn hidden representations of the input data. This improvement is constructive in perceptual problems, where hand-crafted filters are merely impossible to achieve good results. In the following sections, we discuss from the original neural network to the further advanced ones.

2.3.2 Perceptron

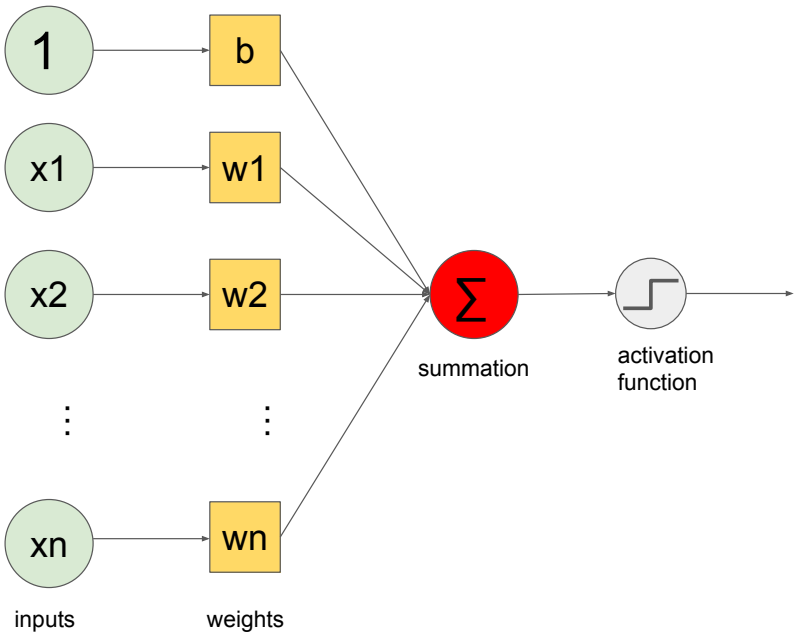

Figure 2.4: A single perceptron

We first recap the early neural network, perceptron, which was introduced by Frank Rosenblatt in the 1950s [**?**] and 1960s [**?**]. The perceptron is an algorithm or a function for learning a binary classifier. The function takes a d-dimensional

vector $\mathbf{x}$ as input along with a weight vector $\mathbf{w}$ and calculates a single output value.

$$a = f(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b) \tag{2.2}$$

where $\mathbf{w}^T\mathbf{x} = \sum_{i=0}^{d} \omega_i x_i$ denotes the inner product between the input and weight vector and b is bias. The perceptron behaves as a nonlinear function σ of a linear combination of the input where each element x_i is multiplied with it's corresponding coefficient ω_i. Figure **??** illustrates the process of a single perceptron process.

At first, the perceptron function is designed to solve the *binary classification* problem in which each datapoint $\mathbf{x}$ belongs to a single class (denoted as **1** or **0**) and the weight vector $\mathbf{w}$ represents a boundary hyperplane that splits the space into two classes seperately. Therefore, points that lie at the same side of the plane are grouped to the same class. And the *sign* function is used as activation function σ in **??** to return 1 if the combination is positive and 0 otherwise.

$$a = f(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad \mathbf{w}^T\mathbf{x} + b \geq 0 \\ 0 & \text{if} \quad \mathbf{w}^T\mathbf{x} + b < 0 \end{cases} \tag{2.3}$$

2.3.3 Activation functions

Different from perceptron, activation function here is continuous, which makes the network weights differentiable with respect to output signals.

Here we list some of those activation functions commonly used in neural network modeling:

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \tag{2.4}$$

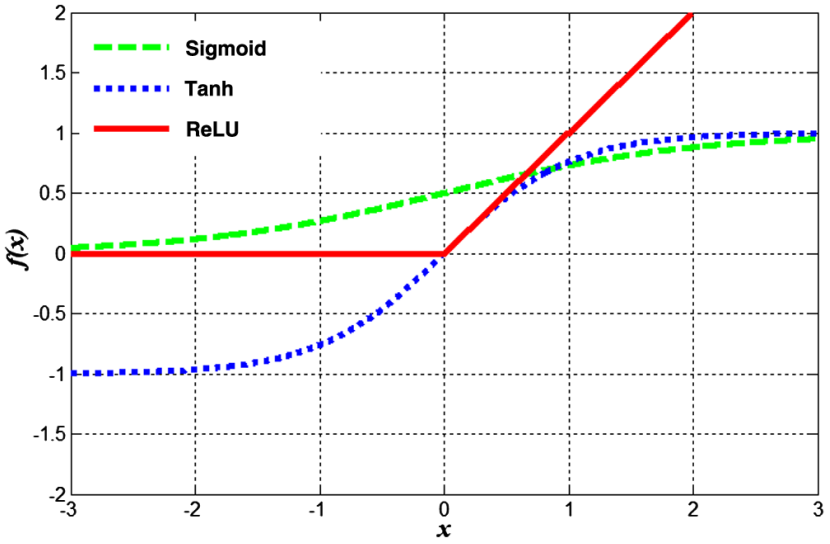

Figure 2.5: Plot of Sigmoid, Tanh and ReLU activation functions.

$$\tanh(x) = \frac{\exp(2x-1)}{\exp(2x+1)} = 2 \times \text{sigmoid}(2x) - 1 \tag{2.5}$$

$$\text{relu}(x) = \max(0, x) \tag{2.6}$$

The *sigmoid* function (**??**) scales real numbers to the range $[0, 1]$, where the large positive numbers change to 1 and the large negative ones become 0. Similar to *sigmoid* but aims to produce zero-centered signals, *tanh* function (**??**) is a scaled version of *sigmoid* with values range of $[-1, 1]$.

The drawback of both *sigmoid* and *tanh* is that when the numbers are large, the squashed values are saturated, hence their gradients is extremely small, which leads to the vanish problem when training deep neural networks. In such cases, Rectified Linear Unit (*ReLU*, **??**) is preferably used, which converts negative values to zero while keep the positive ones. Figure **??** illustrates the three functions in the same input range.

2.3.4 Artificial Neural Networks

The study of artificial neural networks (ANNs) has been inspired by the observation that biological learning systems are built of very complex webs of interconnected neurons in brains. The human brain contains a densely interconnected network of approximately trillion of neurons. ANN systems are motivated to capture this kind of highly parallel computation based on distributed representations.

Generally, ANNs is a combination of a large number of interconnected processing neuron organized in a multi-layer architecture, where signal from a neuron can be passed to another from nearby layers (figure **??**). Each neuron is also formulated the same form as **??**. In general, neural network has an input layer, an output layer and a predefined number of hidden layers. Each hidden layer stands for a single transformation step in the process, which aims to extract hidden patterns of the input stored in it's neurons. In recent years, state-of-the-art neural networks have multiple hidden layers. The number of layers determines the depth of the network, in contrast to the width of the network, which is determined by the maximum number of neurons between the hidden layers. In many cases, increasing depth increases the complexity of the model, creates more space to store information and therefore increase its learning capability. These information is continually passed to following layers to extract more essential features. Due to the fact that sequence of linear transformations always equals to a single linear one, neural network uses nonlinear activation function σ to element-wise apply to each processed neuron.

Researchers have been actively conducting research on designing the architecture for ANNs to tackle machine learning and deep learning problem. Each new network design establishes on top of the older ones and each aims to solve a specific task independently. In order for the networks to deal with different tasks,

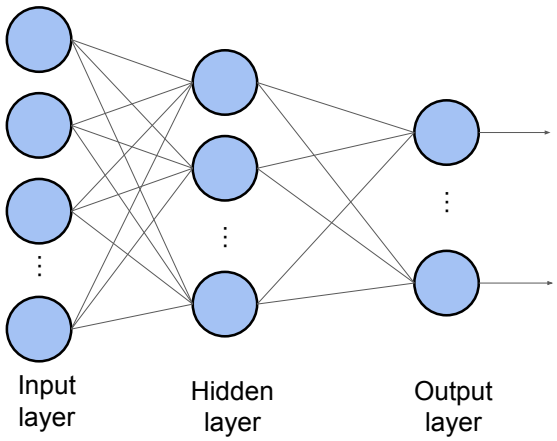

Figure 2.6: A simple three-layer neural network

such as classification, object detection, human action recognition, it cannot be without the usage of loss functions.

2.3.5 Loss functions

As stated in **??**, supervised-learning neural networks must be trained with plenty of inputs-outputs data pairs. With every input, the network is expected to give a predictive outcome. While training, the network continuously adjusts its weights to minimize the discrepancy between its predicted outcome and the real outcome. The value of the discrepancy is often measured by using loss functions.

Loss functions are very diverse, each aims to solve different things and also is used for different tasks. For instance, in the regression task, the most used loss function is the least mean squared error. Suppose we have a model with parameters θ, given a training sample (x, y) where x is the input and y is the ground-truth, the loss for this single sample is calculated as follows

$$\mathcal{L}(\theta, x, y) = \frac{1}{2}[y - h_\theta(x)]^2 \tag{2.7}$$

Similarly with classification task, the most frequently used one is cross-entropy

loss. With the same notation as the example above, the loss function is as follows

$$\mathcal{L}(\theta, x, y) = -(1 - y_i)\log(1 - h_\theta(x_i)) - y_i \log h_\theta(x_i) \tag{2.8}$$

There are various factors involved in choosing a loss function for specific problem such as type of machine learning algorithm chosen, ease of calculating the derivatives and to some degree the percentage of outliers in the data set.

So far, loss functions are used to determine the error between the output of our algorithms and the given target value. For the networks to perform better, it is essential to minimize the loss value. For only one training sample to be minimized, it is undoubtedly easy. However, in practice, the loss value should be estimated for every possible training samples; this is nearly impossible. Therefore, the process of training can be formulated as an optimization problem:

$$\mathcal{J}(\theta) = E_{x,y}[L(\theta, x, y)] \approx \frac{1}{n} \sum_{i=1}^{n} L(\theta, x_i, y_i) \tag{2.9}$$

where (x_i, y_i) is the i^{th} oair and n is the number of pairs in the dataset. The target is to search for a set of parameters θ which minimize the expected loss function, or more realistically, calculate its approximation. An effective algorithm for this task is gradient descent, which is a fundamental tool of modern machine learning problems.

2.3.6 Gradient descent

Gradient descent (GD) is an iterative optimization algorithm used to find the local minimum of an objective function $J(\theta)$ by updating the parameters θ in the opposite direction of the gradient of that objective function. A gradient

simply measures the change in all weights with regard to the change in error. In mathematical terms, a gradient is a partial derivative with respect to its inputs.

How big the steps the GD takes into the direction of the local minimum are determined by the learning rate α, which figures out how fast or slow we will move towards the optimal weights (as illustrated in Figure **??**). Small value of α might lead to consistency but slow progress while larger one can result in faster progress but risk divergence. Thus, it must be carefully chosen.

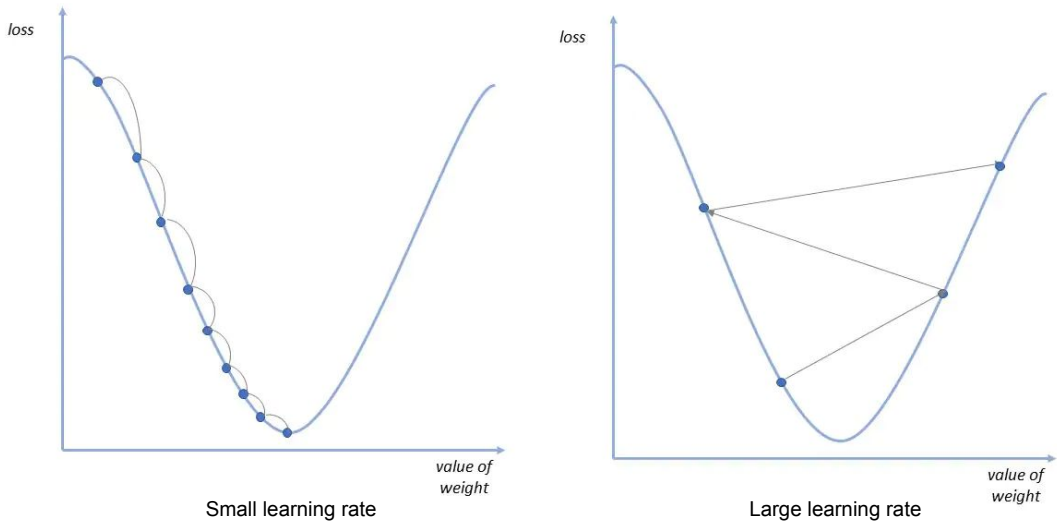

Figure 2.7: Gradient descent

Mathematically, the parameter θ_i in the set of parameters θ is updated as follows:

$$\theta_i := \theta_i - \alpha \frac{\partial J(\theta)}{\partial \theta_i} \tag{2.10}$$

While the original GD, also called vanilla GD, performs weights update after one pass through the whole dataset, which can be costly in practical scenario, there has been various variations that can alleviate this problem. Stochastic gradient

descent (SGD) [**?**] updates the parameters for each calculated training example one by one, which can help faster convergence than vanilla GD depending on the problem. Additionally, the frequency of those updates can result in noisy gradients, which may cause the error rate to fluctuate instead of slowly decreasing. For the most strategic method, it must be the Mini-batch GD variant. It simply splits the training dataset into small batches and performs an update for each of those batches. From this variant, others with more parameters and mechanisms beside learning rate are introduced to improve the consistency and convergence speed of the algorithm, such as Adam [**?**] or RMSProp.

2.3.7 Feed-forward and Back-propagation

Feed-forward describes the process of information flowing forward through the entire neural network. Given input data x, it is propagated through the intermediate layers to finally produce $\hat{y}$ at the output layer. Feed-forward is used both in the training and the testing phase to make predictions for any given input. In the training stage, the error is estimated between the network's output and ground truth; that essentially provides gradient information to update the network parameters.

Back-propagation With the loss computed, back-propagation computes the gradient with respect to the weights of the network for a single input-output pair, and then flow the gradient information backward through the network . At each of the layers, new gradients are computed based on the following layers by the chain rule, then it is used to update the parameters in these layers using gradient descent. The process happens again until the gradients at the input layer are calculated and updated.

The process of feed-forward and back-propagation keep on repeating, and the weights keep updating continuously until the network becomes better fitted to

the data that was fed into it, with reasonable evaluation result.

2.4 Convolution Neural Network

Perceptron algorithm has been proving their potential power by achieving many initial success. Most notable is the convolutional neural nets (convnets or CNNs), beginning with the model for handwritten digit recognition problems introduced by Yann LeCun at ATT Bell Labs [**?**]. Nowadays, CNNs have become the essential technique in any module or state-of-the-art model of computer vision. In fact, the convolution operation tackles the exponential parameters growth by the weight sharing kernel.

The first CNN architecture was introduced by Yan Le Cunn et al. [**?**] to process the visual data, which is inspired from the way human vison system works. One of the most common usage of the CNN is as a feature extraction module. It takes the images $H \times W \times C$ (or $H \times W \times D$) and extracts the information through multi-level downsampling layers, which is know as feature maps. Each smaller feature map holding the comprehensive information and relationship of nearby pixels. Feature map can also transform to another form, which is called feature vector.

2.4.1 Convolution in signal-processing

Convolution technique first appeared in the signal processing field. Given two signals f and g as two vectors with corresponding length are m and n, f is the input signal and g is the kernel. The convolution definite output signal will be $h = f * g$. (equation **??**)

$$f * g(i) = \sum_{n=0}^{N-1} g(n) f(i-n) \tag{2.11}$$

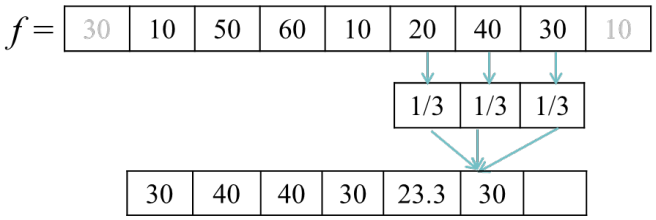

Figure 2.8: Example of convolution 1D

- Stride is the number of element that the kernel skip as it slides on the vector. For example, when the stride is 2, the kernel moves 2 cell when it slides on the vector.

- Padding is the added cell to the original vector before applying the convolution operation. Without padding, the output of 1×7 with kernel 1×3 will be 1×5. The common padding technique using in signal processing is replicate the first and last value, which means to solve it with out shape changing. For example we will add the padding on the raw signal f by assign $f(-1) = f(N-1) = 30, f(N) = f(0) = 10$ (figure **??**).

2.4.2 Convolution in multi-dimension

In image processing, convolution is an operator that combines the pixel information from neighbor pixels based on the kernel weight. The formal form of convolution operator in 2D is

$$f * g(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) g(x - m, y - n) \tag{2.12}$$

Where f is the image size $M \times N$ and g is the kernel. The intuitive illustration is show in fig **??**. Visually, a convolution is imagined as sliding the filter across the image, and in each step it produce a value by taking the linear combination of nearby pixel. As same as the convolution in one dimensional, convolution in im-

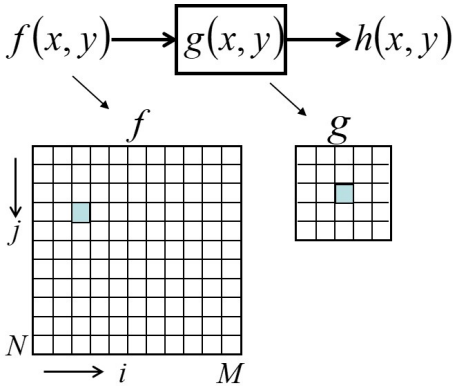

Figure 2.9: Illustration of convolution 2D

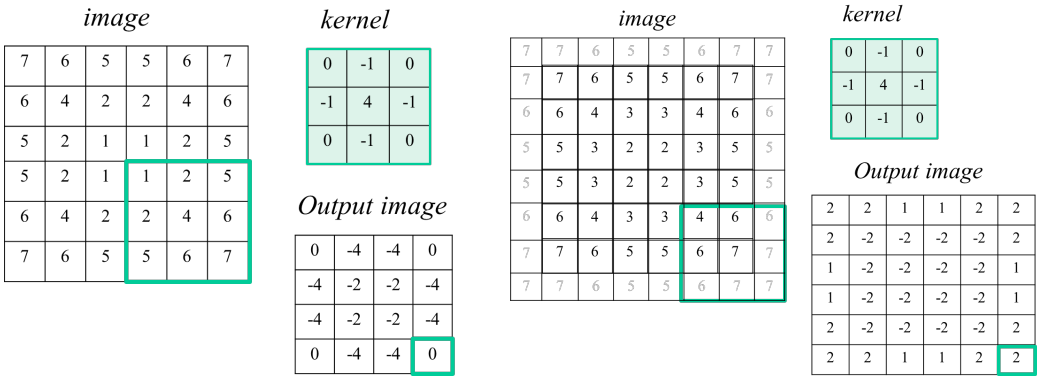

Figure 2.10: Comparation of convolution with padding (right) and without padding (left)

age still faces some limitations. To deal with it, **stride** and **padding** have been used. The difference with one dimensional case is the kernel moves a number of stride value both vertical and horizontal when it slides; the common padding technique types are zero-padding and pixel-replicate-padding. However by extracting the information by convolution, the information loss occurs.

The convolution operators appears long time ago, but inherits from the neural network strength, CNN is self generate the kernel parameters automatically. In other words, in the stage of training the network, the values of the kernel are "learned" from the given data by back propagation process. In addition to exploiting the spatial relationships, the kernel still holding less number of pa-

rameters than traditional feature extraction way.

2.4.3 Pooling layer

Pooling layer is use to reduce to image size. This lead to the efficiency in the reduction of the number of model parameters. The popular pooling types are max-pooling and average-pooling.

2.5 Transformer

"Attention is All you Need" (Vaswani, et al., 2017) [**?**], is impactful work in both natural language processing or computer vision field. It presented a lot of improvements and the proposed "transformer" model is entirely built on the self-attention mechanisms without using sequence-aligned recurrent architecture (which will describe in section **??**). The Transformer model has an encoder-decoder architecture, as commonly used in many natural machine translation models. Later decoder-only Transformer was shown to achieve great performance in language modeling tasks, like in GPT and BERT. Nowadays, transformer also use in computer vision because of the context-understanding ability of it and become one of the state-of-the-art.

2.5.1 Sequence to sequence (Seq2Seq)

The seq2seq model was introduced by Sutskever [**?**], et al in the language modeling field. It tackle the transformation of an source sequence to a target one, firsly apply in neural machine translation (figure **??**). The seq2seq model normally has an encoder-decoder architecture, composed of:

- **Encoder**: to compress the information of input sequence and represent it as an context vector. This representation is expected to be summery the whole source sequence.

Figure 2.11: The sequence to sequence overview

- **Decoder**: is initialized with the context vector to emit the transformed output. Seq2seq only used the last embedding state of the encoder network as the decoder initial input.

This architecture is the foundation of many sequential models nowadays, but still exist few limitations such as use a fixed-length context vector. And one of the the most improvements handle this problem is attention mechanism.

2.5.2 Attention and Self-Attention mechanism

The attention mechanism uses to memorize long source sentences in neural machine translation. Define the attention mechanism in a scientific way. Say, we have source sequence x have length n and target sequence y have length m

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]; \mathbf{y} = [y_1, y_2, \ldots, y_m] \tag{2.13}$$

The key idea is it create weighted shortcuts between the context vector and the entire source input. The alignment between the source and target is learned by a context vector. The context vector is a sum of hidden states $\boldsymbol{h}_i$ of the encoder sequence and weighted by alignment scores $\alpha_{t,i}$, in there α_{t_i} is calculate from the hidden states $\boldsymbol{s}_i$ from the decoder.

$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \boldsymbol{h}_i; \text{ Where } c_t \text{ is the context vector for output } y_t \tag{2.14}$$

The alignment score $\alpha_{t,i}$ is assign by a model to the pair of input at position and output at position based on how well they match. The set of pairs are weights how much of each source hidden state should be considered for each output. The

score function is therefore in the following form below at timestamp t:

$$e_{ij} = a(s_i, h_j), \qquad \alpha_{i,j} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})} \tag{2.15}$$

Self-attention is an attention mechanism variant, in that it relating different positions of a single sequence in order to compute a representation of the same sequence. The self-attention mechanism use to learn the correlation between the current token and the previous part of the sequence.

2.5.3 Multi-head Self-Attention

In the paper attention is all you need, transformer component have re-define the attention by using retrieval concept. The encoded representation of the input as a set of key-value pairs $\mathbf{K}, \mathbf{V}$. In the decoder, the previous output is compressed into a query $\mathbf{Q}$ and the next output is produced by mapping this query and the set of keys and values.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{n}})\mathbf{V} \tag{2.16}$$

Where n is the dimension of the source hidden state. And we have a scalar score a_{ij} follow

$$a_{ij} = \text{softmax}(\frac{\mathbf{q}_i {\mathbf{k}_j}^\top}{\sqrt{n}}) = \frac{\exp(\mathbf{q}_i {\mathbf{k}_j}^\top)}{\sqrt{n}\sum_{r \in S_i} \exp(\mathbf{q}_i {\mathbf{k}_r}^\top)} \tag{2.17}$$

The reason behind this because the attention operation can be thought of as a retrieval process as well. As mention in eq **??**, if we remove the constraint that the weight α_t is a one-hot vector, the operation could be thought as a retrieval process according to the α_t as a probability vector. This way efficiency computes the vector α_t as a matrix multiply to measure the similarity by project s and h to a common space instead of we have to go through the network $n \times m$ times

to acquire all the attention scores a_{ij}.

$$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \frac{\boldsymbol{s}_t^\top \boldsymbol{h}_i}{\sqrt{n}} \tag{2.18}$$

The multi-head self-attention module is a key component in transformer. In summary, the idea of multi-head attention is to split the feature dimension of the input into many parts and attention over these sub-dimensions. Then computes the scaled dot-product attention (eq **??**) over each subspace in parallel. Each independent attention outputs are simply concatenated and linearly transformed into expected dimensions.

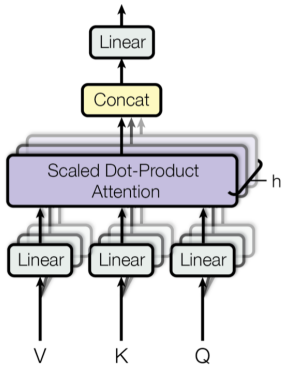

Figure 2.12: Multi-head scaled dot-product attention mechanism [**?**]

2.5.4 Transformer architecture

Transformer model has an encoder-decoder architecture. The encoder generates a representation from a large context. It constructed by two main submodules, a *multi-head self-attention* layer and a *projection network*.

In the transformer decoder is retrieve information from the encoded representation. The architecture is quite similar to the encoder. Figure **??** shown the whole propose architecture

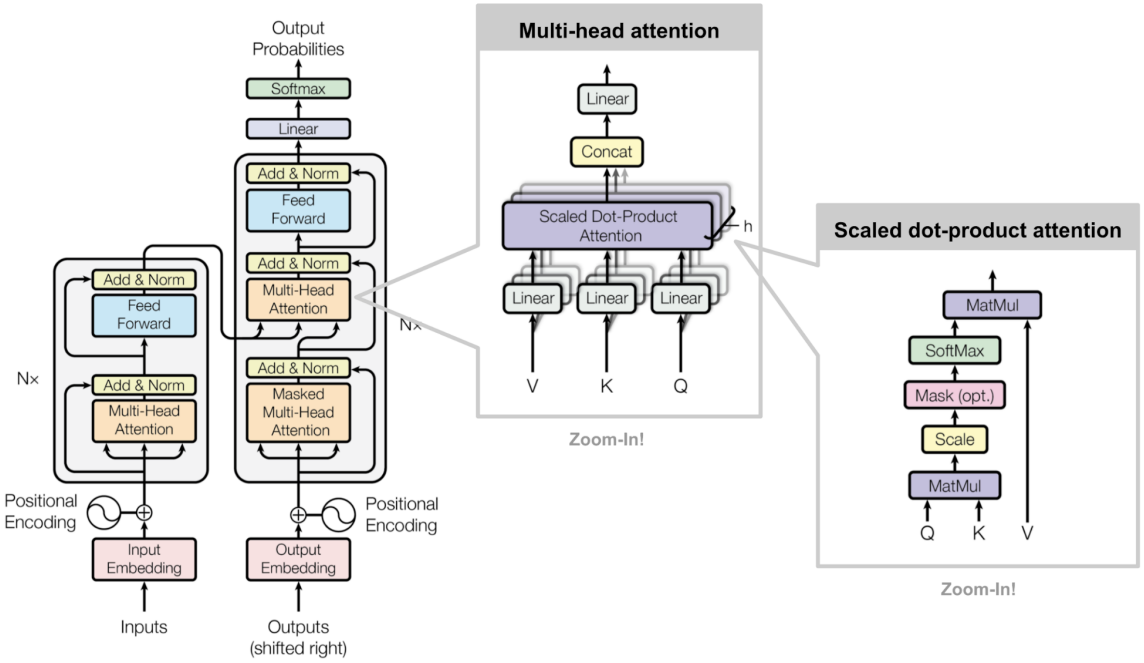

Figure 2.13: The full transformer architecture [**?**]

CHAPTER 3

RELATED WORKS

In this chapter, introduce the volume organs segmentation problem, inspire from video object segmentation problem and is most related to our work. We also discuss the semi-supervised approaches which are widely used to solve medical image segmentation task and details about the state-of-the-art model that we utilized in our system in section **??** *and* **??** *Next, we introduce related works which have a same scenario and motivation to solve the interactive segmentation concept. We provide detailed discussions about the Interactive methods and Positional encoding technique which applied directly to our propose architecture in section* **??** *and section* **??***).*

3.1 Segmentation

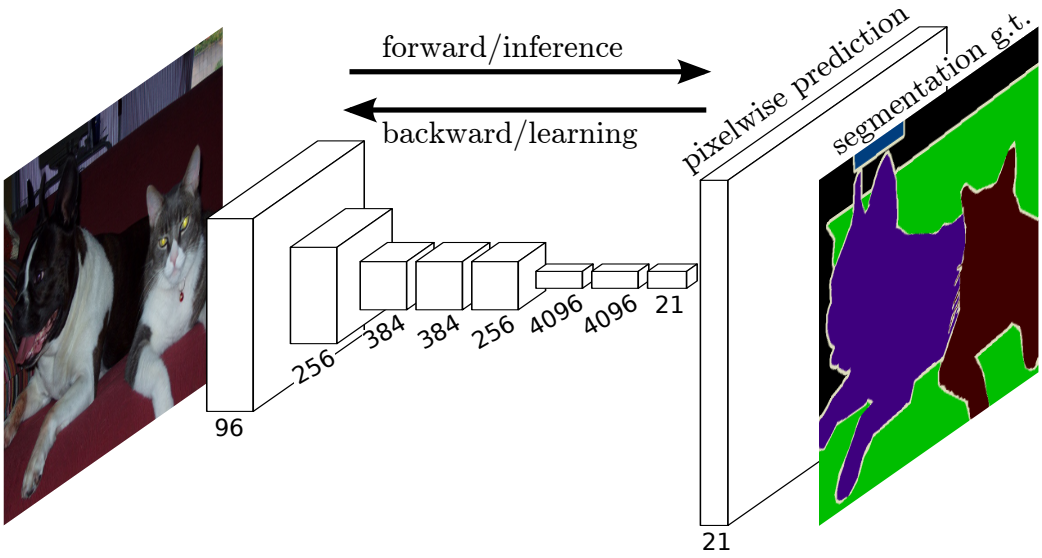

Figure 3.1: Fully Convolutional Networks Architecture

Segmentation is one of the most important tasks in image processing and has

a long history of development. It has grown strongly since neural networks and computing resources starting to explode. One of the basic ideas behind segmentation is that you can use convolution layers as a feature extraction mechanism. By downsampling the input image after passing it through a backbone CNN model containing multiple pooling layers, you can generate a prediction at a smaller scale that needs to be upsampled to the original image size. This approach, which uses fully convolutional layers to perform classification for each pixel in the feature map, is known as Fully Convolutional Networks (FCNs)[**?**].

The reason behind this architecture that combine deep semantic information in deeper layers and spatial information in shallow layers together. However, there are still many limitations, such as losing spatial information in the downsampling process; its inability to leverage global context information; and the lack of a mechanism for variant-scales.

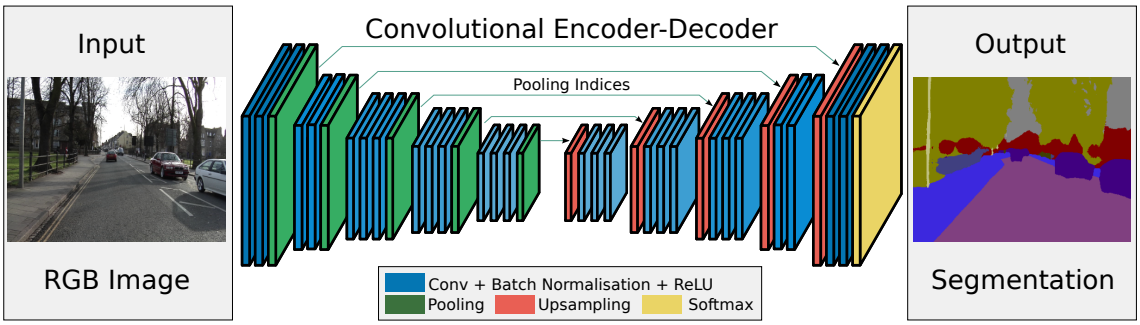

Figure 3.2: SegNet Architecture

The next popular method is using encoder and decoder based architecture. Segnet[**?**] is the pioneer when do a symmetric architecture between encoder and decoder. In details, decoder mirror copy of the decoder and augment with information from encoder. There are multi descendants inspire from this and popular nowadays like Unet[**?**]

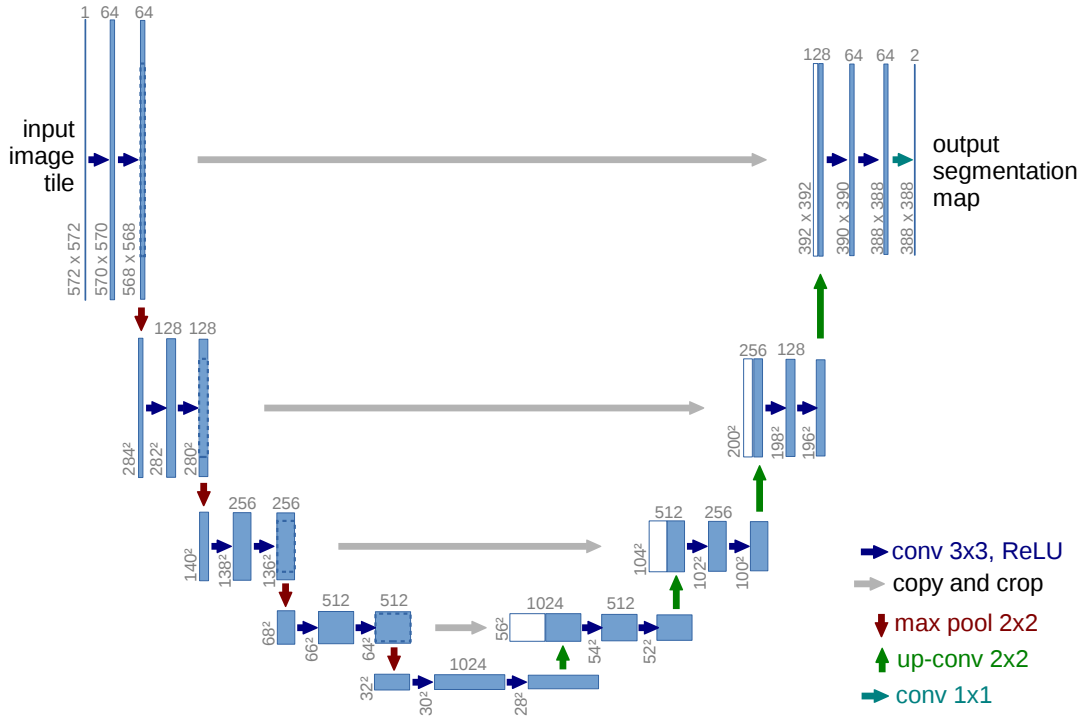

Figure 3.3: Unet Architecture

Unet[**?**] is one of the most popular architectures in computer vision, and it has been shown to be very effective in medical applications. The encoder provides spatial information that helps to decode the features more accurately. There are variants of Unet that have been developed specifically for better performance, such as Unet++[**?**] and Attention U-Net[**?**]. These variants continue to show promise for semantic segmentation tasks, with improved accuracy over other methods.

Another model to approach the fixed kernel size problem is of the family of segmentation models called DeepLab[**?**]. DeepLab model family is a promising solution to the fixed kernel size problem. They use dilated convolution (also known as atrous convolution) to learn features from a larger region without increasing computational expenses. This allows for deeper neural networks, which can better capture complex patterns in data. Atrous Convolutions with differ-

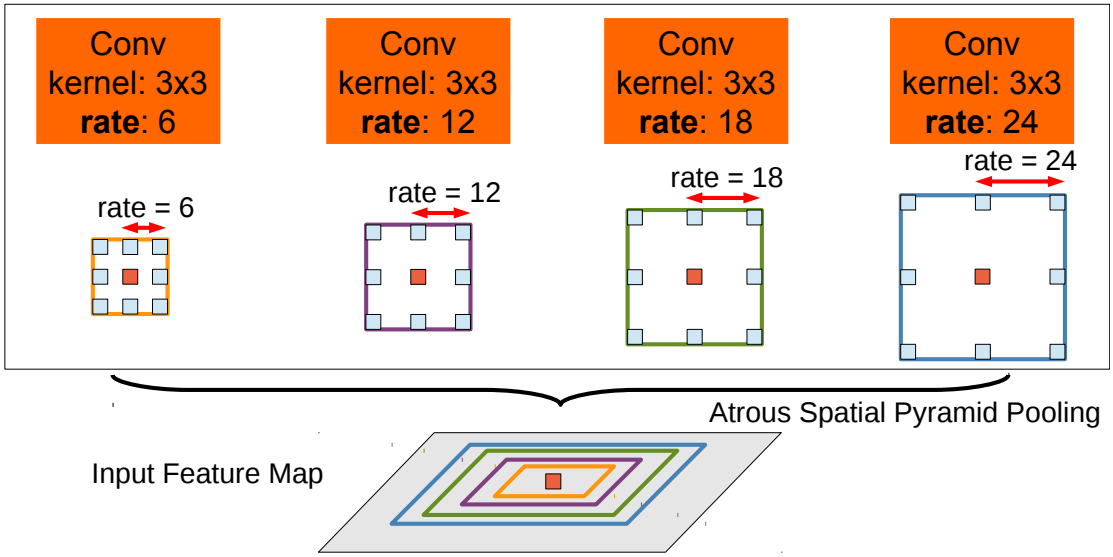

Figure 3.4: Atrous Spatial Pyramid Pooling

ent rates are stacked in the encoder, called the Atrous Spatial Pyramid Pooling module. This module offers the ability to learn multi-scale features in the encoding phase, which is important for accurately capturing information about objects and scenes.

Computer vision has come a long way in the past few years. Researchers have found that transformer power is one of the potential ways to improve computer vision. Transformers are famous for their ability in downstream task in natural language processing. However, the naive usage that replaces transformer vision as a feature extraction is not exploiting almost its capability. Due to that TransUnet[**?**] use the strong point of them ideas, which is transformer leverage both detailed high-resolution spatial information from CNN features and the global context encoded by Transformers, also inspired by the u-shaped architectural design

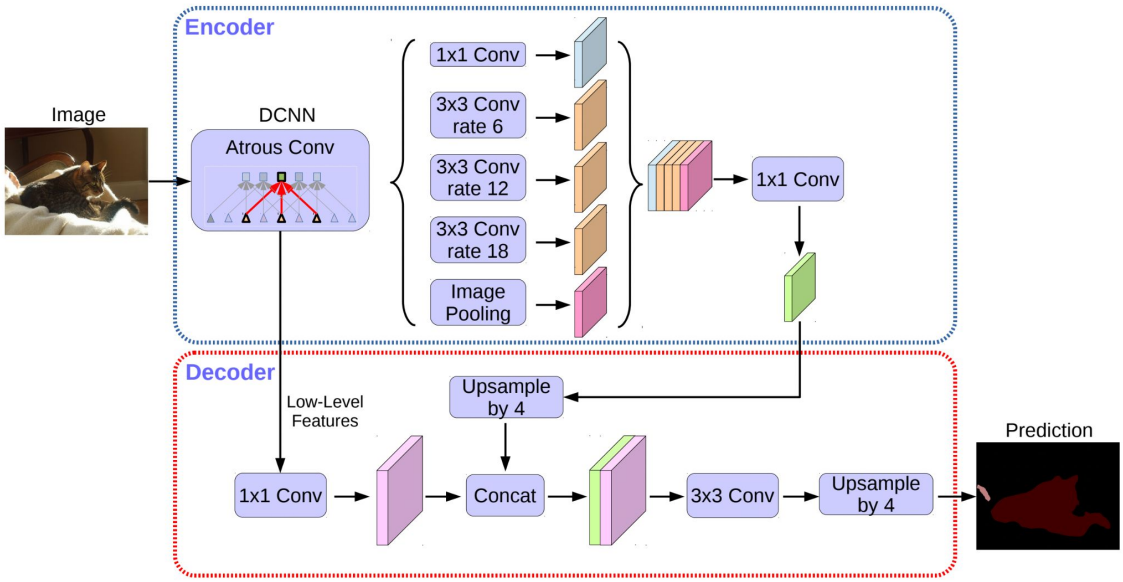

Figure 3.5: DeepLabv3 Architecture

3.2 Uncertainty Estimation

Data is growing bigger everyday, but that is still not enough for deep neural networks. The empirical report of [**?**], [**?**] suggests that the performance of recent deep networks is not yet saturated with respect to the size of training data, since data are mostly unlabeled. For this reason, learning methods from semi-supervised learning to unsupervised learning are attracting attention of many researchers. However, given a fixed amount of data, performance of semi-supervised or unsupervised still cannot match with that of fully-supervised learning. Thus, the data annotation has a vital part in uplifting the performance of neural networks Having said that, what then is the suitable approach while the budget for annotation is limited? [**?**] first proposed active learning where a model actively selects data points that the model is uncertain of. The core idea of active learning is that the most informative data point would be more beneficial to model improvement than a randomly chosen data point.

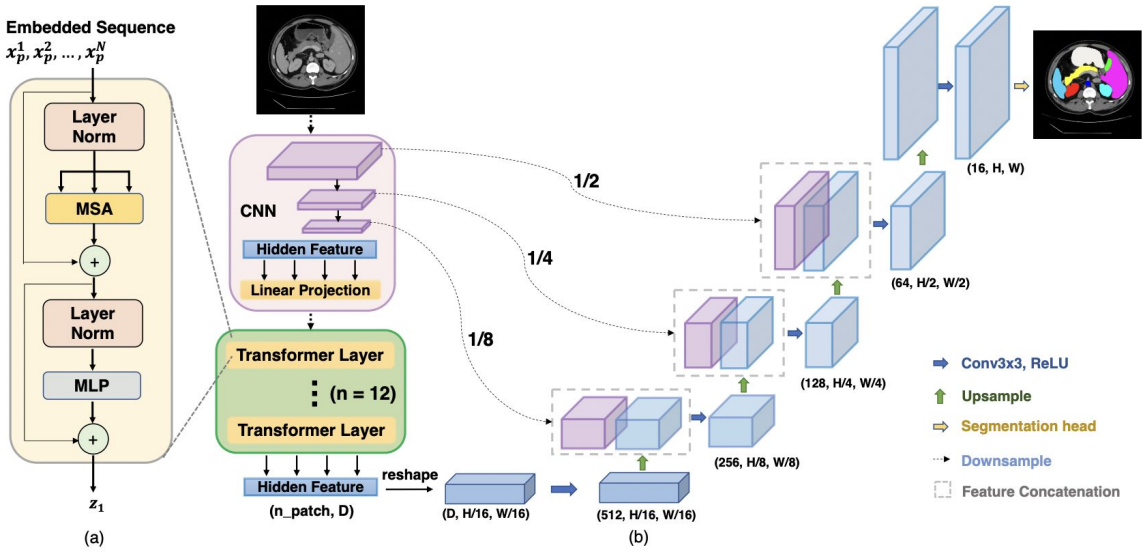

Figure 3.6: (a) schematic of the Transformer layer; (b) architecture of the TransUNet

Active learning has been advancing in the recent decades. Given a scenario where a labeled dataset L and an unlabeled dataset U is available, active learning aims to select a fixed number of subset of samples from U to be labeled such that they can lead to improvement in model performance. To identify the most valuable examples to be labeled, many sampling strategies have been proposed in previous research, and it is mostly the main focus of those. Data points chosen by these strategies are expected to be able assist model to become more generalized and comprehensive after training.

Given a pool of unlabeled data, there have been two major approaches according to the selection criteria: uncertainty-based, diversity-based and expected model change

Uncertainty-based strategy [[**?**], [**?**], [**?**], [**?**], [**?**]] selects samples for which the model produces most uncertain predictions while the diversity approach [[**?**

], [?], [?]] selects diverse data points that can represent the entire distribution of the unlabeled pool. Expected model change [[?], [?], [?]] selects data points that brings great impact to the training model parameters or its outputs.

The most straightforward method of the uncertainty approach is to utilize class posterior probabilities to define uncertainty. Despite its simplicity, this approach has performed remarkably well in various tasks, such as object detection [?], semantic segmentation [?] and human pose estimation [?].

Recently, Gal et al. [?] obtains uncertainty estimation from deep networks through multiple forward passes by Monte Carlo Dropout [?]. It was shown to be effective for classification with small datasets, but according to [32], it does not scale to larger datasets.

Interestingly, [?] train an additional regression module, which uses the training loss as optimization target, to predict a score for each unlabeled sample to evaluate its worthiness for labeling.

The majority of empirical results from previous researches suggest that active learning is actually reducing the annotation cost. The problem is that most of methods require task-specific design or are not efficient in the recent deep networks.

As a task-agnostic uncertainty approach, [?], [?] train multiple models to construct a committee, and measure the consensus between the multiple predictions from the committee. [?] goes with the same strategy where the authors compute the entropy between numerous trained agents, as depicted in Figure ??.

We follow [?] to use deep ensemble to generate prediction for each unlabeled sample, then based on consensus entropy to select good samples. While previous works aim at finding the most uncertain samples to be manually labeled by experts, but in our context, manual annotation is forbidden, therefore we try to

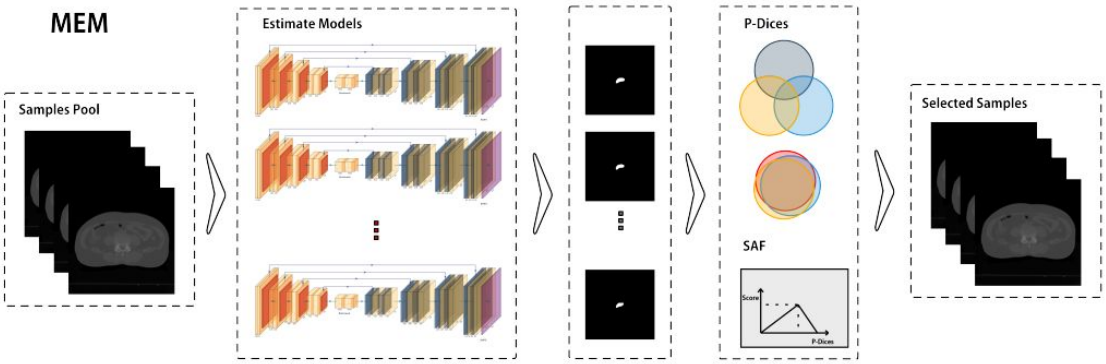

Figure 3.7: A pipeline for active learning proposed in [**?**], which calculate unlabeled sample score through the two-step query strategy: P-Dices and SAF

find most certain samples that can be used for retraining.

3.3 Semi-supervised learning

With the constantly increasing volume of non-annotated raw data in practice, semi-supervised learning approaches have been gaining popularity in the area of deep learning recently to effectively utilize this source of data. Recent works focus on two of the most common semi-supervised methods, which are consistency regularization and pseudo-labeling.

Consistency regularization aims at enforcing the consistency between the predictions (or intermediate features) of two different views of an unlabeled input sample. Some prior works [**?**], [**?**] add perturbation to the input sample, then forward both the original and perturbed one through the same network to generate two augmented outputs. These two outputs are forced to be similar by the regularization function. This approach can be seen as attaching an unsupervised

branch to the supervised learning paradigm.

On the other hand, pseudo-labeling, a.k.a., self-learning, self-labeling, or decision-directed learning, is initially developed for using unlabeled data in classification. Recently, it is applied for semi-supervised segmentation ([**?**], [**?**], [**?**], [**?**]). Frequently, it involves more than one model to produce pseudo-labels by converting model predictions on unlabeled samples into soft or hard labels as optimization targets for retraining. The process can be iterated several times. Various schemes are introduced on how to decide the pseudo segmentation maps. For example, the GAN-based methods ([**?**], [**?**], [**?**]), use the discriminator learned for distinguishing the predictions and the ground-truth segmentation to select high-confident segmentation predictions on unlabeled images as pseudo segmentation.

One common type of approach is teacher-student settings. The teacher, which usually is the larger network, is fully-supervisedly trained on the labeled data while generating pseudo-labels on unlabeled data to guide the student model to learn more stably ([**?**], [**?**], [**?**], [**?**], [**?**]). Often, the teacher's parameters are updated using an exponential moving average from the student's parameters [**?**].

Apart from the teacher-student paradigm, many proposals are the variance of it. As discussed in [**?**], because of exponential moving average, teacher network in mean-teacher tends to be very close to the student when the training process converges.

Dual-student [[**?**], [**?**], [**?**]], for instance, is proposed to use two independently initialized student network without teacher and has been achieving great performance as well. One examplar is [**?**], which has been adopted in our work, trains 2 segmentation models simultaneously on both labeled and unlabeled data. Basically, with two source images as input, they generate a cutmix version

by combining them as one image and mixing the two pseudo segmentation maps obtained from the models. Afterward, the mixed pair of image and mask is used as supervision of the two segmentation networks. Developed from that, [**?**] incorporates the idea that those two models should have different architectures to boost the performance by far, one of them should be conventional CNNs while the other has a transformers-based structure.

Recently, SSL has been widely used for medical image computing to reduce the annotation efforts [[**?**], [**?**], [**?**], [**?**]]. Bai et al [**?**] developed an iterative framework where in each iteration, pseudo labels for unannotated images are predicted by the network and refined by a Conditional Random Field (CRF) then the new pseudo labels are used to update the network.

We desire to build the same framework as [**?**] in which multiple models, in which choosen architectures are inspired from [**?**], are used to refine the pseudo labels.

However, one inherent weakness of the pseudo label learning is that the pseudo label usually contains noisy predictions. Despite the fact that most pseudo labels are correct, wrong labels also exist, which could compromise the subsequent training. If the model is fine-tuned on the noisy label, the error would also be transferred to the adapted model. Therefore, we also integrate active learning to decide which pseudo labels are suitable for retraining the model.

3.4 Mask Propagation

Video object segmentation is a task that requires specific objects to be segmented in a given video. In terms of Semi-supervised settings, a first-frame annotation is provided by the user, and the method segments objects in all remaining frames automatically. This technique, which uses a mask as prior knowledge to predict other masks, is called Mask Propagation.

To propagate these sparse labels through the entire video sequence, traditional

methods often solve an optimization problem with an energy defined over a graph structure [**?**], [**?**], [**?**].

Early deep neural network methods rely on fine-tuning the networks at test time to make segmentation networks focus on target objects and then inference on all other frames, which are extremely slow. Among them, OSVOS [**?**] and MoNet [**?**] finetune pre-trained networks on the first-frame ground-truth at test time. OnAVOS [**?**] extends the first-frame fine-tuning by introducing an online adaptation mechanism. Following these approaches, MaskTrack [**?**] and PReM [**?**] utilize optical flow to help propagate the segmentation mask from one frame to the next. Despite achieving promising results, the test-time fine-tuning restricts networks' efficiency. Faster approaches have been proposed such as temporal CNN, capsule routing [**?**], tracking, space-time matching, and memory-based methods:

Temporal CNN. Recurrent methods propagate information often from the most recent frames, either via a mask [**?**] or via a hidden representation [**?**], [**?**]. These methods are prone to drifting and struggle with occlusions.

Tracking In contrast, tracking-based approaches [**?**], [**?**], [**?**] perform frame-to-frame propagation and are thus efficient at test-time. They however lack long-term context and often lose track after object occlusions.

Space-time matching. While some methods [**?**], [**?**] [**?**] also include the first reference frame for global matching, the context is still limited and it becomes harder to match as the video progresses

Memory-based. To address the context limitation, recent state-of-the-art methods use more past frames as feature memory [**?**], [**?**], [**?**] ,[**?**], [**?**], [**?**], [**?**]. These approaches leverage a memory network to embed past-frame predictions into memory and apply non-local attention mechanisms to propagate object information from the memory to the current frame. Our work strongly adopts the

memory-based style.

Oh et. al [**?**] introduces a framework which extract embeddings from multiple intermediate frames and store inside the memory. Then in forwarding stage, these memories are combined with the query embeddings to segment the object

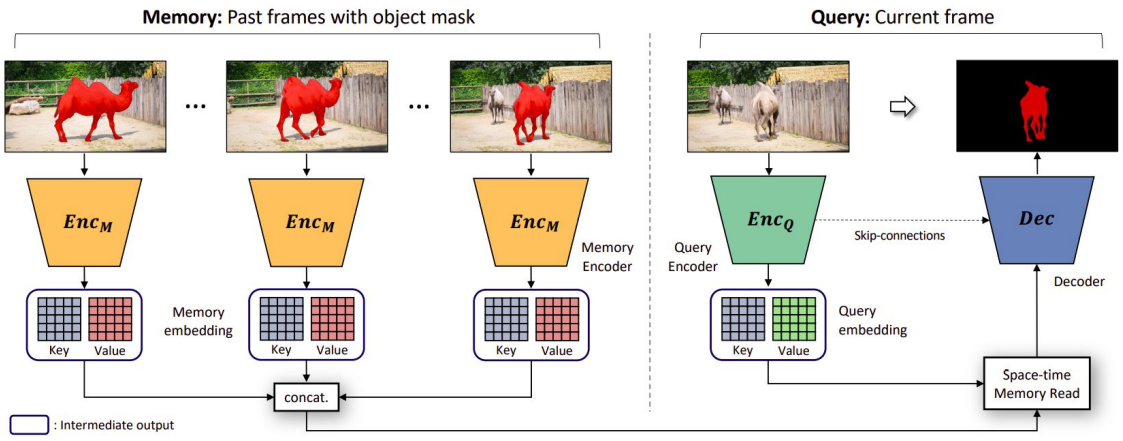

Figure 3.8: Overview architecture of STM [**?**].

Cheng et. al [**?**] proposes a framework in which the user initially scribbles a mask of an object, the scribble is transformed into the real binary mask (S2M), then that mask is propagated through every video frame based on STM [**?**]. Then the user can choose some incorrect frames and scribble to guide the fusion with the propagated frame to output more accurate masks (Different-aware fusion).

Among these extensions, we adapt STCN [**?**] as one of our main modules for refinement as it is simple and effective. but with minor modification. However, most variants cannot handle long videos due to the ever-expanding feature memory bank of STM.

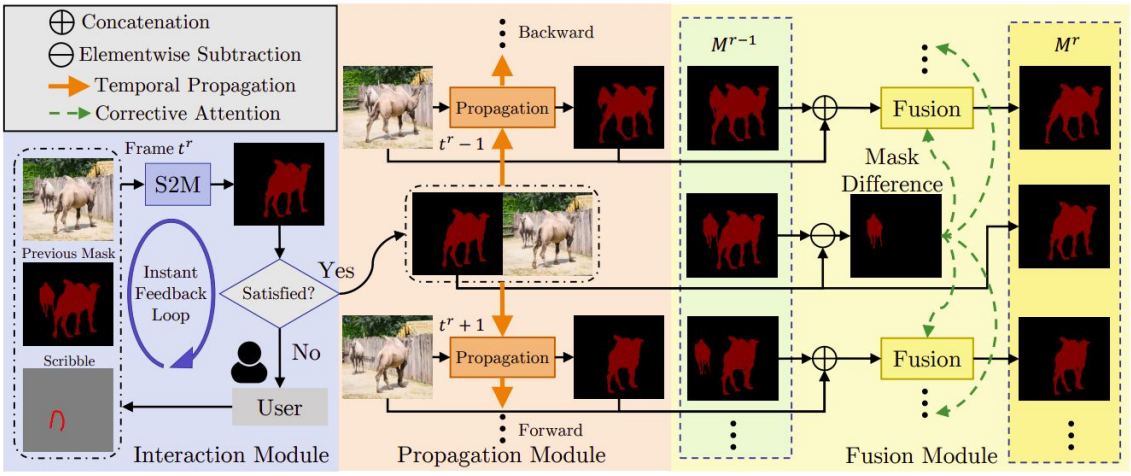

Figure 3.9: Overview architecture of MiVOS [**?**]

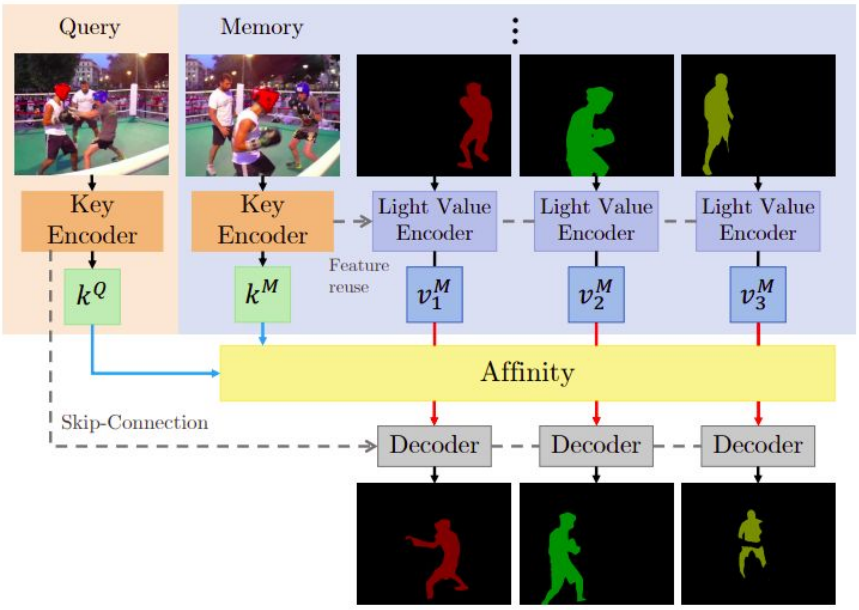

Figure 3.10: Overview architecture of STCN [**?**]

Nonetheless, the mask propagation and segmentation are still computed individually for different objects. The problem restricts the application and devel-

opment of the VOS with multiple targets. Hence, [**?**] proposes AOT to associate and decode multiple targets simultaneously, as efficiently as processing a single object. Moreover, it is the first method to utilize Transformer modules inside mask-propagation

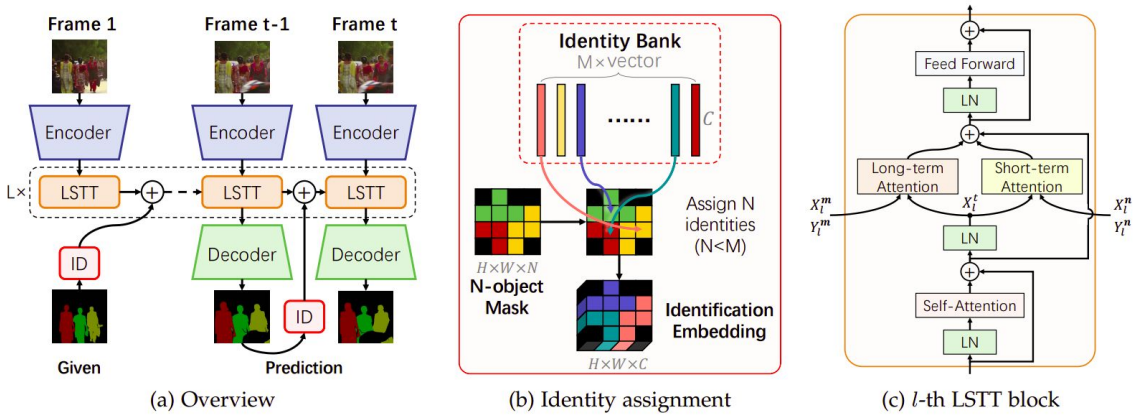

Figure 3.11: Overview architecture of AOT [**?**]

Yet, it still does not solve the GPU memory explosion problem. Recently, authors of STCN [**?**], develop XMem [**?**] which uses multiple memory stores to capture different temporal contexts while keeping the GPU memory usage strictly bounded due to the long-term memory and consolidation.

Although video object segmentation has yielded promising results in a number of domains, these approaches have not been exploited for medical dataset creation. The use of medical images for training and testing machine learning algorithms is critical to the development of accurate models that can be used to diagnose and treat diseases. In order to improve the accuracy and effectiveness of machine learning algorithms for medical image analysis, it is important to develop novel

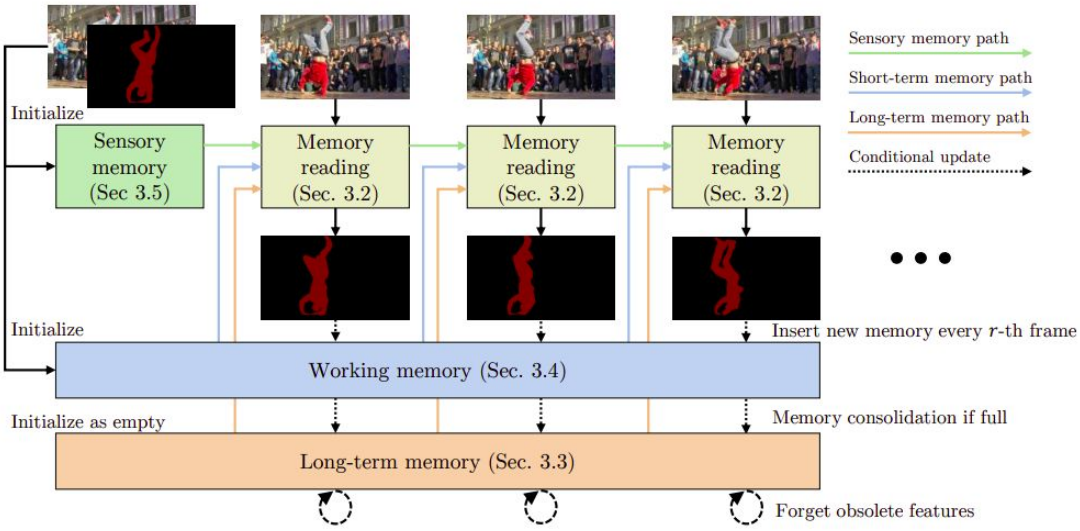

Figure 3.12: Overview architecture of XMem [**?**]

methods specifically tailored for this domain. Mask propagation is a popular technique in interactive scenarios of video object segmentation.

The user will refine the inputs to the algorithm in order to segment the target objects more accurately. Our proposed concept is heavily inspired by the DAVIS Challenge on Video Object Segmentation. After the first raw prediction of the associate model, the user will interact with some of the valuable slices. After that, they will submit their predicted masks to a server for review. In each of these subsequent interactions, from a list of slices specified by themselves, they choose one frame with an uncertain prediction and provide it to a server who then provides them with a new set of scribbles in this frame which points out regions that are false positives or negatives

3.5 Temporal Position Encoding

After the breakthrough of deep learning in still-image recognition originated with the introduction of the AlexNet model [**?**], there has been active research

devoted to the design of deep networks for a sequence of images.

Many attempts in the past leverage CNNs trained on images to extract features from the individual frames and then perform temporal integration of such features into a fixed-size descriptor using pooling, high-dimensional feature encoding, or recurrent neural networks. Karpathy et al. [**?**] presented a thorough study on how to fuse temporal information in CNNs and proposed a "slow fusion" model that extends the connectivity of all convolutional layers in time and computes activations through temporal convolutions in addition to spatial convolutions. Time-series models have also been widely studied in this task to exploit the capabilities of long-term memory features, like LSTM [**?**] or GRU [**?**] or most recently, Transformer [**?**].

However, different from RNN and LSTM, the self-attention inside Transformer cannot capture position information by design. This is critical, considering that the model is otherwise entirely invariant to the sequence order, which is harmful to video encoding.

One common approach is to use position encodings which are combined with input elements to expose position information to the model. These position encodings can be a deterministic function of position ([**?**]; [**?**]) or learned representations. Convolutional neural networks inherently capture relative positions within the kernel size of each convolution. They have been shown to still benefit from position encodings ([**?**]), however. For the Transformer, which employs neither convolution nor recurrence, incorporating explicit representations of position information is an especially important consideration since the model is otherwise entirely invariant to sequence order.

To overcome this issue, [**?**] adopts relative position embedding [**?**], which ensures translation-equivariance property and allows the model to generalize unseen sequence length during training. Empirically, [**?**] confirms that relative position

indeed helps to capture the sequential properties of video content, improving the video encoder' performance further.

In our work, we implement a simple way to inject the position information into the feature vector for the model to learn, in the hope that the target model will be capable of understanding relatively the location of specific objects within a long sequence, in our case a CT volume.

In fact, looking at the problem of abdomen organ segmentation in CT volume, this position embedding can be seen as a model constraint since any organ of a human only appear in certain areas. Therefore, with the introduction of this information, the model can comprehend the knowledge of organ relative position to make a better prediction.

Many previous works manipulate model outputs by incorporating conditions to the model inputs or intermediate features. [**?**] first introduce encoding label information into both discriminator and generator to control the generation of the image. [**?**] also has its own way to concatenate encoded bounding-boxes values to the input vector for the model to learn. All these mentioned research uses only simple embedding layers to embed the scalar values, which encourages us to do the same.

CHAPTER 4
PROPOSED METHOD

In this chapter, we present our solution for tackle the interactive volume object segmentation. Our proposed approach include reference and propagation module. At first, the pipeline of the method is shown. Then, each module is introduced and the detailed implementation is presented. We apply the distillation technique not only in architecture design but also in the training strategy stage. Finally we dive deep in the potential of loss function especially for medical.

4.1 Method overview

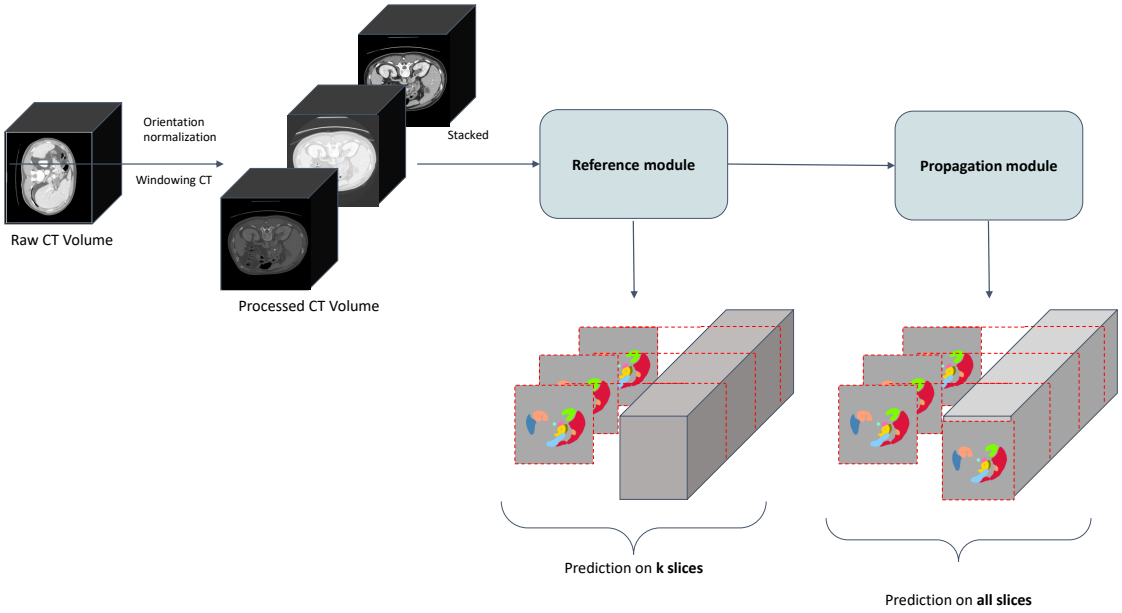

Figure 4.1: The overview description of our proposed pipeline. After the preprocessing stage, a small number of masks are generated across the CT volume by the Reference module. Then these masks are inherited by the Propagation module for the prediction of the remaining slices.

Our proposed method contains two main modules, a reference module and a

propagation module. Firstly, the entire CT volume is processed using windowing CT to get a stack of three-channel slices [**?**].

Next, a small yet sufficient amount of potential slices are sampled from the stack using a reference strategy to become candidates. These candidate slices progress through the Reference module to obtain the minimal number of preliminary masks (as described in **??**). Afterward, the Propagation module (as described in Section **??**) disseminates the information deduced from these initial masks to the remaining slices. Ultimately, the result is obtained after the final post-processing stage, which includes re-orientation and converting into appropriate format for later usage.

Furthermore, to effectively create pseudo-labels that are consistent for the unlabeled data, we follow the uncertainty estimation technique based on consensus voting scheme. These pseudo-labels that have high certainty level are used for retraining in the next cycle (as described in **??**).

The overview of our method is explained in Figure **??**

4.2 Preprocessing

For preprocessing, we apply Windowing technique [**?**] with different levels and widths to target specific parts of human organs. Windowing, also known as grey-level mapping, contrast stretching, histogram modification or contrast enhancement is the process in which the CT image grayscale component of an image is manipulated via the CT numbers; doing this will change the appearance of the picture to highlight particular structures. The brightness of the image is adjusted via the window level, window level determines the central or midpoint grey value for the range of HU displayed in the image. Ideal imaging of different tissues depends on the window level. The contrast is adjusted via the window width. A wide window (400-2000 HU) is suitable for examining structures of vastly dif-

Table 4.1: Window parameters

Group	Window level	Window width
Spine-bone	900	1400
Tissues	200	350
Chest	787	2137

ferent attenuation values. For instance, chest structures are best viewed using a wide window.

In or experiments, we apply 3 wide windows corresponding with 3 different versions of a single slice by highlight the abdomen, chest, and spine groups and stack it to one as a three-channel image (Fig. **??**). The window parameters in our experiments are shown in table **??**

Where the group of the soft tissue masses can be determined by merging of the abdomen soft tissues and liver. The chest group, which consists of the lungs and mediastinum. By re-calculating the window information from the min value and max value of selected range, these window parameters can be accurately determined.

In addition, we choose the axial plane to cut the slices from the CT volumes since this plane has various dimension sizes. Due to some relatively small organs, it might be better to keep the original size of the slices without any cropping, resampling, or resizing methods. The image is rotated to a predefined angle, then divided by 255 for normalization before going through the next step.

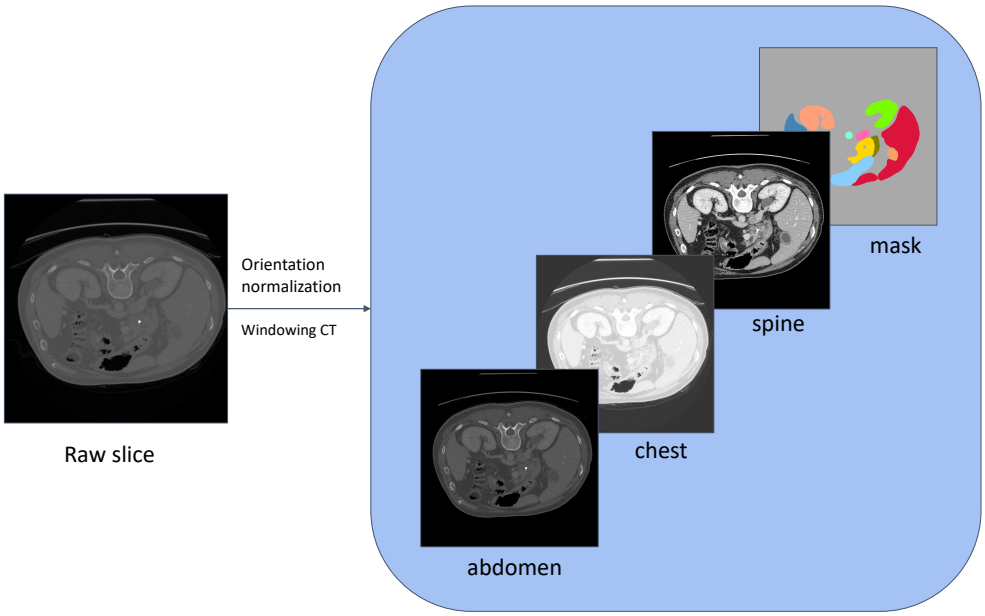

Figure 4.2: Windowing CT. 3 different versions are generated from an original slice.

4.3 Reference module

This module is expected to provide a suggestion of a minimal amount of slices and predicted masks that might contain the most information describing the entire CT Volume. Fig **??** describes the details of this module.

To utilize the enormous number of unlabeled data, we apply the recent semi-supervised method that performs effectively on several other datasets, which is called Cross Pseudo Supervision (CPS) [**?**] (yellow cube in Fig. **??**). CPS enables the usage of unlabeled data by following the dual students technique, where two models are trained simultaneously on labeled data while generating pseudo data for their "peer" to learn. In the testing phase, two models predict the same image, and the result is aggregated by summing up.

We adopt two prominent state-of-the-arts 2D segmentation models with highly different learning paradigms for this CPS framework, which is TransUNet [**?**]

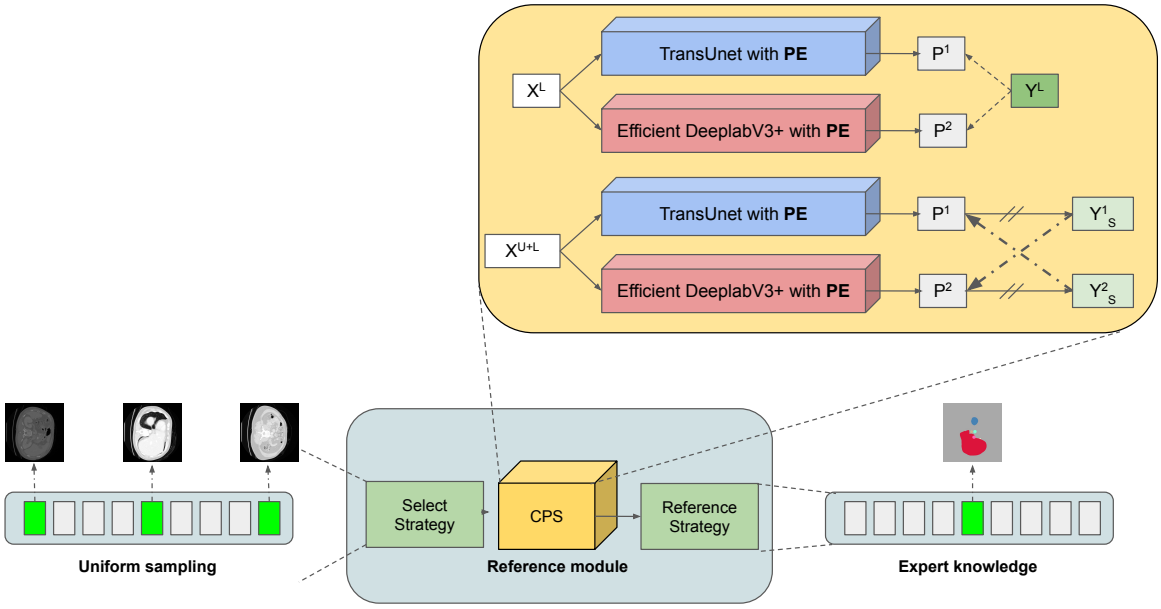

Figure 4.3: The reference module. The semi-supervised technique CPS is applied in both training and inference stage to enhance the precision of model prediction. Strategies are used to smartly choose slices that are informative for the next stage.

and DeeplabV3+ [**?**]. While DeeplabV3+ traditionally focuses more on the local information, transformers model the long-range relation, so the cross training can help to learn a unified segmenter with these two properties at the same time. In short, we choose TransUNet and DeeplabV3+ due to their ability to compensate each other for better performance [**?**].

Positional Encoding. In order for the model to make use of the order of the sequence, we need to inject somne information about the position of the frames. For simplicity, we add an additional embedding layer to embed the relative position of the frame. Specifically, the embedded position index is concatenated with the hidden features before the final segmentation head. The relative position of k^{th} frame of CT volume i with length T_i is calculated as:

$$PE(k) = \frac{k}{T_i} \tag{4.1}$$

We attach this layer to both DeeplabV3+ and TransUnet. Since they follows conventional structure of segmentation models, which comprise of encoder and decoder phases, we manage to attach the layer in a similar way for both of them, as can be generally seen in Figure **??**

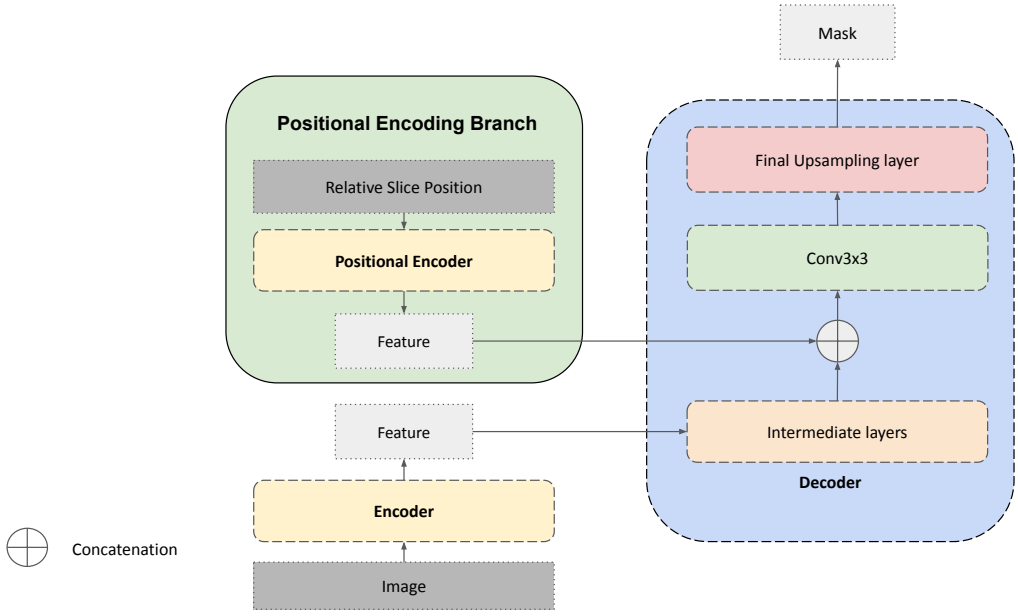

Figure 4.4: A general and simple way to attach a Positional Encoding branch into segmentation models .

In addition, we also propose a both logical and specialist-based strategy to choose which slices can be further used to boost the performance of the Propagation module. The goal of this action is to preserve only some of the most useful information for the refinement stage.

To elaborate on these strategies, prior to being put into the CPS module for

prediction, a small number of slices are uniformly sampled from the processed CT volume. After CPS produces segmentation masks for these slices, another selection step is performed to pick only some of the masks that contain the organs having the largest areas.

4.4 Propagation module

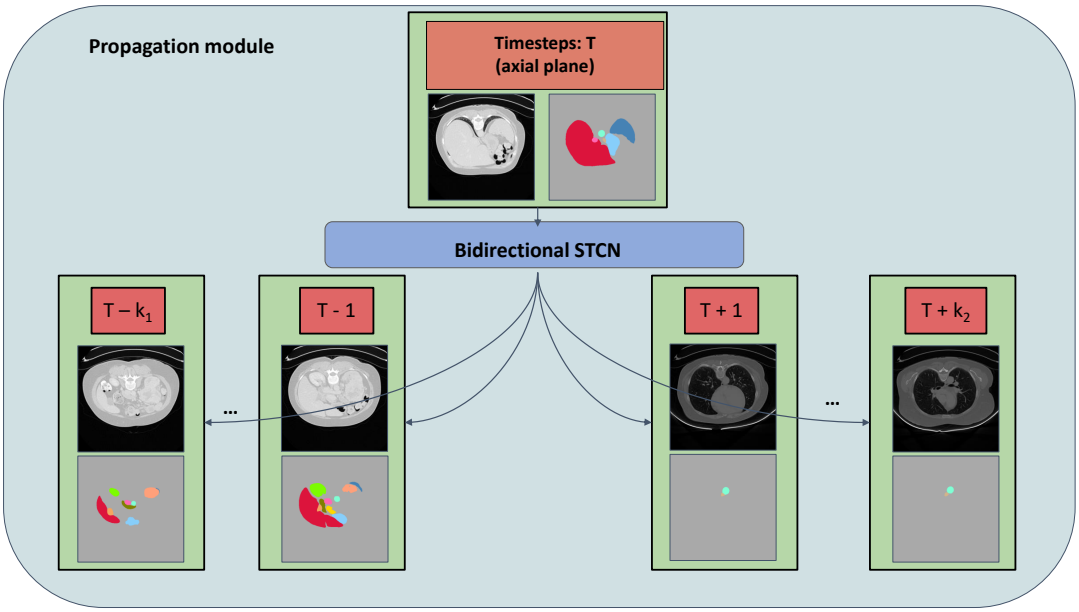

Figure 4.5: The propagation module. From an annotated slice of CT, at timestep T, STCN can make use of that to spread the information through the entire defined range $[T - k_1, T + k_2]$.

This module aims to utilize prior knowledge of given annotated slices from the Reference module to make prediction on the remaining slices, this mechanism can be referred as mask (or label) propagation.

Intuitively, the conventional 2D CNNs cannot comprehend the third dimension information within a CT volume. Thus, in hope of the ability to capture the "temporal" information along the axial plane, we adapt the Space-Time Cor-

respondence Networks (STCN) [**?**], which is a semi-supervised segmentation algorithm that has achieved promising results on Video object segmentation problem, to this 3D manner.

Basically, STCN proposes the use of a memory bank that stores information about previous frames and their corresponding masks and uses them later as prior knowledge. To generate the mask for the current frame, a pairwise affinity matrix is calculated between the query frame and memory frames based on negative squared Euclidean distance, then it is used for supporting the current mask generation [**?**].

Different from the original STCN, we slightly modify it to match the current problem. In the original work, they use only a single dense mask to propagate through the entire video, therefore for the model to perfectly work, that selected mask must contain information about all available classes. For our case to achieve that, we enable the usage of multiple masks for propagation, so that all of these masks should contain enough information about every organs. We also allow the STCN to work in a bidirectional way to enhance the refinement. Fig **??** illustrates this process.

Specially, STCN can be simply trained in the binary manner, meaning that each of the abdominal organs can be learned separately. Therefore, the knowledge can be transferred well between different organ classes.

4.5 Pseudo Labeling

Given a vast amount of unlabeled CT volumes, we apply a uncertainty estimation technique to effectively maximize the utilization of the data.

Firstly, several CPS models are trained on the provided labeled data. Then, we use these trained CPS models to obtain pseudo masks on the unlabeled set. Inspired from [**?**], we calculate the dice scores between these pseudo masks

and the aggregated one. The mean of these dice scores will be compared with a threshold to determine whether the aggregated pseudo masks are qualified. Simply speaking, consensus-based assessment is used to evaluate the quality of pseudo labels.

We determine a single score for the i^{th} volume in the unlabeled set as the formulation below:

$$score_i = \frac{1}{K \times M} \sum_{k=1}^{K^i} \sum_{m=1}^{M} \mathrm{DSC}(\mathcal{Y}_m^{k,i}, \mathcal{Y}_{AVG}^{k,i}) \tag{4.2}$$

$$\mathrm{dsc} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{4.3}$$

DSC represents the Dice Score evaluation metric calculating the overlapping area of prediction X and ground truth Y. Here $\mathcal{Y}_m^{k,i}$ indicates the m^{th} model's output of the k^{th} slice of volume i while $\mathcal{Y}_{AVG}^{k,i}$ is the mask averaged from all M models for the same slice. The easier the sample is, the more inclined the segmentation are to get a similar output if the sample is easier. In contrast, hard samples are more likely to be segmented differently by different models. Hence, we use the proposed score to measure the certainty between models' predictions. Higher score gives more credibility to the prediction, as it is more consistent.

All aggregated samples that have high certainty are then reused for the next supervised training cycle. And after the training finishes, the same labeling process is repeated until all aforementioned models achieve satisfied performance or every unlabeled data has been used.

4.6 Loss functions

For the Reference module, we use the prevalent combination of dice loss and cross entropy loss with smoothing value to alleviate the imbalanced number of the small organs, which occurs due to our splitting into slices process. The same settings are used for CPS in its supervised branch whereas only the dice loss is setup for the unsupervised branch.

For the Propagation module, we implement the online hard example cross entropy (OhemCE or Bootstrapping CE) [**?**] and also calculate the Lovasz loss [**?**] at the same time. OhemCE can help reduce the contribution of background label to the final loss. And since STCN is trained on binary task, OhemCE can direct the model to focus on visible difficult objects. Meanwhile, Lovasz loss is commonly used in the past.

Given y the ground truth mask, and the predicted mask of the model, we reformulate all the losses that we have used in training our networks as below:

Dice loss

The Dice coefficient is widely used metric in computer vision community to calculate the similarity between two matrices. In recent years, it has also been adapted as loss function known as Dice Loss [**?**]

$$DL(y, \hat{y}) = 1 - \frac{2\hat{y}y + \epsilon}{\hat{y} + y + \epsilon} \tag{4.4}$$

Here, ϵ is added in both numerator and denominator to ensure that the function is not undefined in edge case scenarios such as when $\hat{y} = y = 0$.

Cross-entropy loss

Cross-entropy is a traditional loss function that is widely used for classification objective, and as segmentation is pixel level classification it works well. It is defined in [**?**] as a measure of the difference between two probability distributions for a given random variable or set of events. Binary Cross-Entropy is defined as:

$$L_{BCE}(y, \hat{y}) = -(ylog(\hat{y}) + (1-y)log(1-\hat{y})) \tag{4.5}$$

Online hard example cross-entropy loss

We et. al [**?**] propose an online bootstrapping method, which forces networks to focus on hard (and so more valuable) pixels during training.

Let there be K different categories c_j in a label space. For simplicity, suppose that there is only one image crop per mini-batch, and let there be N pixels a_i to predict in this crop. Let y_i denotes the ground truth label of pixel a_i , and p_{ij} denotes the predicted probability of pixel a_i belonging to category c_j . Then, the loss function can be defined as

$$L_{OhemCE} = -\frac{1}{\sum_i^N \sum_j^K 1\{y_i = j \text{ and } p_{ij} < t\}} (\sum_i^N \sum_j^K 1\{y_i = j \text{ and } p_{ij} < t\} log p_{ij}) \tag{4.6}$$

where t (0, 1] is a threshold. Here 1· equals one when the condition inside holds, and otherwise equals zero. In practice, there should be a reasonable number of pixels kept per mini-batch. Hence, the threshold t will be increased accordingly if the current model performs pretty well on a specific mini-batch.

Lovasz-Softmax loss

Berman et. al [**?**] incorporates the softmax operation in the Lovasz extension. Having the similar target as the Dice Loss, The Lovasz extension is a means by which we can achieve direct optimization of the mean intersection-over-union loss in neural networks. In this respect, the Lovasz-Softmax loss can be formulated as follows:

$$L_{lovasz} = \frac{1}{|C|} \sum_{c \in C} \overline{\Delta_{J_c}}(m(c)) \tag{4.7}$$

$$m_i(c) = \begin{cases} 1 - x_i(c) & \text{if } c = y_i(c) \\ x_i(c) & \text{otherwise} \end{cases} \tag{4.8}$$

where $|C|$ represents the class number, Δ_{J_c} defines the Lovasz extension of the Jaccard index, $x_i(c) \in [0, 1]$ and $y_i(c) \in \{-1, 1\}$ hold the predicted probability and ground truth label of pixel i for class c, respectively.

CHAPTER 5
EXPERIMENT

In this chapter, we describe the experiment dataset, evaluation metrics, and challenge platform. Besides, we provide the details configurations and implementation details of our proposed method. Based on that, we also analyze the result with on our observations.

5.1 Dataset and evaluation measures

5.1.1 FLARE22 dataset

The FLARE2022 dataset is curated from more than 20 medical groups under the license permission, including MSD [**?**], KiTS [**? ?**], AbdomenCT-1K [**?**], and TCIA [**?**]. It is extended from the FLARE 2021 Challenge from fully supervised settings to a semi-supervised setting that focuses on how to use unlabeled data. Specifically, they provide a small number of labeled cases (50) and a large number of unlabeled cases (2000) in the training set, 50 visible cases for validation, and 200 hidden cases for testing. The segmentation targets include 13 organs: liver, spleen, pancreas, right kidney, left kidney, stomach, gallbladder, esophagus, aorta, inferior vena cava, right adrenal gland, left adrenal gland, and duodenum. Compare to the FLARE 2021 challenge, the dataset is 4x larger and the segmentations targets are increased to 13 organs.

5.1.2 Evaluation metrics

The evaluation measures consist of two accuracy measures: the region-based Dice Similarity Coefficient (DSC) and the boundary-based Normalized Surface Dice (NSD) [**?**], and three running efficiency measures: running time, area under GPU memory-time curve, and area under CPU utilization-time curve. All measures will be used to compute the ranking. Moreover, the GPU memory consumption

has a 2 GB tolerance.

Let G, S denote the ground truth and the segmentation result, respectively. $|\partial G|$ and $|\partial S|$ are the number of voxels of the ground truth and the segmentation results, respectively. Both metrics take the scores in $[0, 1]$ and higher scores indicate better segmentation performance. We formulate the definitions of the two measures in the two following sections.

5.1.2.1 Dice Similarity Coefficient

The Dice similarity coefficient, also known as the Sørensen–Dice index or simply Dice coefficient, is a statistical function which measures the similarity between two sets of data. The DSC has become arguably the most widely used tool in several segmentation benchmarks [**?**], [**?**]. In term of segmentation task, it is used to evaluate the region overlap between the predicted and the target mask. The formulation is as follows:

$$\text{DSC(G,S)} = \frac{2|G \cap S|}{|G| + |S|} \tag{5.1}$$

5.1.2.2 Normalized surface Dice

Despite its popularity, DSC does not take boundary precision into account, while that is a crucial aspect in clinical medical tasks. In contrast, NSD is sensitive to this boundary error and is used to evaluate how close the segmentation and ground truth surfaces are to each other at a specified tolerance τ. NSD can be formualzed as below:

$$\text{NSD(G,S)} = \frac{|\partial G \cap \mathcal{B}^{\tau}_{\partial S}| + |\partial S \cap \mathcal{B}^{\tau}_{\partial G}|}{|\partial G| + |\partial S|} \tag{5.2}$$

where $\mathcal{B}^{\tau}_{\partial G} = \{x \in \mathcal{R}^3 | \exists \tilde{x} \in \partial G, ||x - \tilde{x}|| \leq \tau$, $\mathcal{B}^{\tau}_{\partial S} = \{x \in \mathcal{R}^3 | \exists \tilde{x} \in \partial S, ||x - \tilde{x}|| \leq \tau$ denote the border region of the ground truth and the segmentation surface at tolerance τ, respectively. According to [**?**], FLARE22 Challenge uses $\tau = 1$mm.

5.1.3 Implementation details

5.1.3.1 Environment settings

The development environments and requirements are presented in Table **??**.

Table 5.1: Development environments and requirements.

Windows/Ubuntu version	Ubuntu 18.04.5 LTS
CPU	Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz
RAM	1×32GB;
GPU (number and type)	One Quadro RTX 5000 16G
CUDA version	11.6
Programming language	Python 3.10
Deep learning framework	Pytorch (Torch 1.11.0, torchvision 0.12.0)

5.1.3.2 Preprocessing protocols

In the beginning of both training and inference phase, the CT volumes are splitted into slices of 2D images before applying Windowing CT. As mentioned before, we keep the original size of the volumes. Then,for each of these slices, Windowing CT is applied with three different settings to generate three version of each. The three settings aim to highlight three particular parts of the human organs, which are: the chest, the abdomen, and the spine. Detailed configuration is shown in Table **??**

5.1.3.3 Training protocols

Currently, we find that using only simple 2D transform functions such as horizontal/vertical flipping or rotating might be enough for both modules to generalize. In the training stage, the Reference module follow traditional training process, in which two models are concurrently trained. For the Propagation module, we inherit the same process as in [**?**] which samples 3 neighboring slices at a time.

Table **??** and Table **??** mention the training protocols for Reference module and Propagation module, respectively. In both settings, we use the original-sized images, which is $[512, 512]$ for the training and inference phases.

Table 5.2: Training protocols for Reference module: CPS of TransUnet and Efficientnet DeeplabV3+

Network initialization	Random initialization
Batch size	2 (labeled) + 2 (unlabeled)
Patch size	None
Total iterations	50000
Optimizer	AdamW
Initial learning rate (lr)	0.0001
Lr decay schedule	multiplied by 0.5 every iteration at $[40000, 45000]$
Training time	48 hours
Number of model parameters	105M (TransUnet Resnet50) + 11M (Efficientnet DeeplabV3+) [1]
Number of flops	108G (TransUnet Resnet50) + 1,3G (Efficientnet DeeplabV3+) [2]

5.1.3.4 Testing protocols

In the inference phase, after applying windowing CT technique for pre-processing, we then resize all the slices to 512×512 prior to predicting. When the model generate a mask prediction, which apparently the same size as 512, we interpolate

Table 5.3: Training protocols for Propagation module: STCN with Resnet backbone

Network initialization	Random initialization
Batch size	8
Patch size	None
Total iterations	50000
Optimizer	AdamW
Initial learning rate (lr)	0.0001
Lr decay schedule	multiplied by 0.5 every iteration at $[40000, 45000]$
Training time	48 hours
Number of model parameters	54,416,065 [3]

that to the original size.

Initially, in the Reference stage, we do not forward the entire volume since it would increase the GPU resource tremendously. Here, we follow a simple strategy which is uniform sampling. For each CT volume, we sample a slice every 5 steps, then group them as a batch and forward through the Reference module. As mentioned earlier, the Reference module consists of two 2D segmentation models: TransUnet with Resnet50 as backbone and DeeplabV3+ with EfficientNetB3 as backbone, they are used simultaneously to generate the prediction. The masks outputed from this module are filtered once more to select only potentially informative slices and its annotation. Another sampling strategy is used here, to pick the possible slices, we iterate through them and calculate the areas of the visible organs. Then for each organ, we choose the slice that have the largest area of it, which leaves us with only 13 candidate slices maximum.

Then comes the Propagation module, with these only 13 slices, this module has to propagate all the information across the remaining slices. This also includes a propagation strategy as well. For each of the annotated slices as the pivot points, the module perform bidirectional propagation strategy, then sum them

up before finally deciding which classes they are. In more detail, the STCN architecture inside the Propagation module has a memory bank that can store up past information and use them to predict the current input. For every k_1 frame, the bank encode the frame and save the information once, then find top k_2 encoded frames that is most similar with the current one to generate the mask. In addition, the memory bank gets flushed every k_3 frames stored. In all our experiment, we find that $k_1 = 5$, $k_2 = 20$ and $k_3 = 200$ gives best overall results while keeping the GPU memory not overloaded.

Regarding the pseudo-labeling process, we use multiple trained CPS to create a pool of pseudo-labeled data and measure the uncertainty between them. Following the Equation **??**, we filter out samples with scores higher than 0.9 and use them for the next training cycle. The cycle loops until all unlabeled samples become high quality.

5.2 Quantitative results

Here we present both quantitative and qualitative results of our proposed method. We also include the ablation study (Table **??**) to further analyze the effectiveness of each of our modules.

Some interesting insights can be spotted in Table **??**. Overall, we can see that using the pseudo-labeled data for training, helps boost the performance of the model by a great amount. Unfortunately, we have yet to fully explore every unlabeled sample (only 700 samples were used for training in our submission), but intuitively, the number of used unlabeled samples is likely to be directly proportional to the evaluation result. Another notable observation is that the DSC for some small human organs (gallbladder and adrenal glands) can hardly be improved because of the class imbalance problem.

Table **??** shows that each module contributes to the final score of our submission.

Table 5.4: Comparison between using and not using the pseudo labels as supervised training data. The model that is used for the report is TransUnet [**?**]. The highlighted figures emphasize the highest values in each row. The Dice Scores are reported on the public test set, and are given from the public leaderboard.

Number of pseudo-labeled samples	0	200	700
Liver	0.9141	0.9555	**0.9604**
RK	0.645	0.7944	**0.8014**
Spleen	0.8071	0.9144	**0.9255**
Pancreas	0.6152	0.7309	**0.7567**
Aorta	0.8992	0.9272	**0.9335**
IVC	0.7398	0.7967	**0.8207**
RAG	0.5786	0.6507	**0.6545**
LAG	0.4376	**0.6179**	0.6138
Gallbladder	0.474	**0.5889**	0.5885
Esophagus	0.74	**0.7784**	0.7783
Stomach	0.7678	0.8403	**0.8424**
Duodenum	0.4425	0.5617	**0.5679**
LK	0.7015	**0.8112**	0.8026
Mean DSC	0.674	0.7668	**0.7728**

Table 5.5: Ablation experiment on each proposed modules and techniques. Scores are reported on the public test set, which in available on the public leaderboard.

No.	**CPS**	**Windowing CT**	**UE**	**MP**	**Mean DSC**
1					0.547
2	✓				0.762
3	✓	✓	✓		0.770
4	✓	✓	✓	✓	0.784

The third and fourth rows where both Cross Pseudo Supervision (CPS) and Uncertainty Estimation (UE) is used mean that pseudo-labels that are qualified by UE are used as supervised inputs in CPS workflow whereas the remaining unlabeled data are used as unsupervised inputs. Noticeably, in the fourth row, with the Mask propagation (MP) applied, DSC score is enhanced. It is surprising that MP only looks upon the minority of the slices to fully propagate through the whole volume. We recommend looking at the qualitative result in Section **??** to fully understand what each component contributes to the total score. The detailed evaluation for our final submission on the public test set is shown in

Table **??**.

Table 5.6: The final evaluation score for our final submission on the public test set

Mean DSC	**Liver**	**RK**	**Spleen**	**Pancreas**	**Aorta**	**IVC**
0.7841	0.9591	0.8149	0.9244	0.7499	0.9383	0.8262
RAG	**LAG**	**Gallbladder**	**Esophagus**	**Stomach**	**Duodenum**	**LK**
0.6456	0.601	0.6877	0.7986	0.8446	0.5808	0.8219

5.3 Qualitative results

5.3.1 Overall

Looking at examples that are well-predicted by our approach in Fig **??** (1b, 2b, 3b), it demonstrates good segmentation masks with clear and smooth mask boundaries. Some small organs can also be seen segmented successfully and precisely meaning that both proposed modules can work effectively with organs having various sizes. For reference, we use [**?**] for all the CT volume visualization in our project.

On the other hand, our models suffer from various difficult cases where organs are missing. Generally, there are two cases that negatively affect our approach:

1. Relatively small organs (adrenal glands (Fig **??** (1e)), gallbladder (Fig **??** (1e)), and esophagus (Fig **??** (3e))) account for the lowest DSC since they usually are failed to be identified by the Reference module.

2. Other organs (pancreas (Fig **??** (1e)) and duodenum (Fig **??** (2e))) despite having a larger size, yet their lengths on the axial plane are short and sometimes occluded by many surrounding organs, which can affect how the information propagating through the slices, causing class confusion in the result.

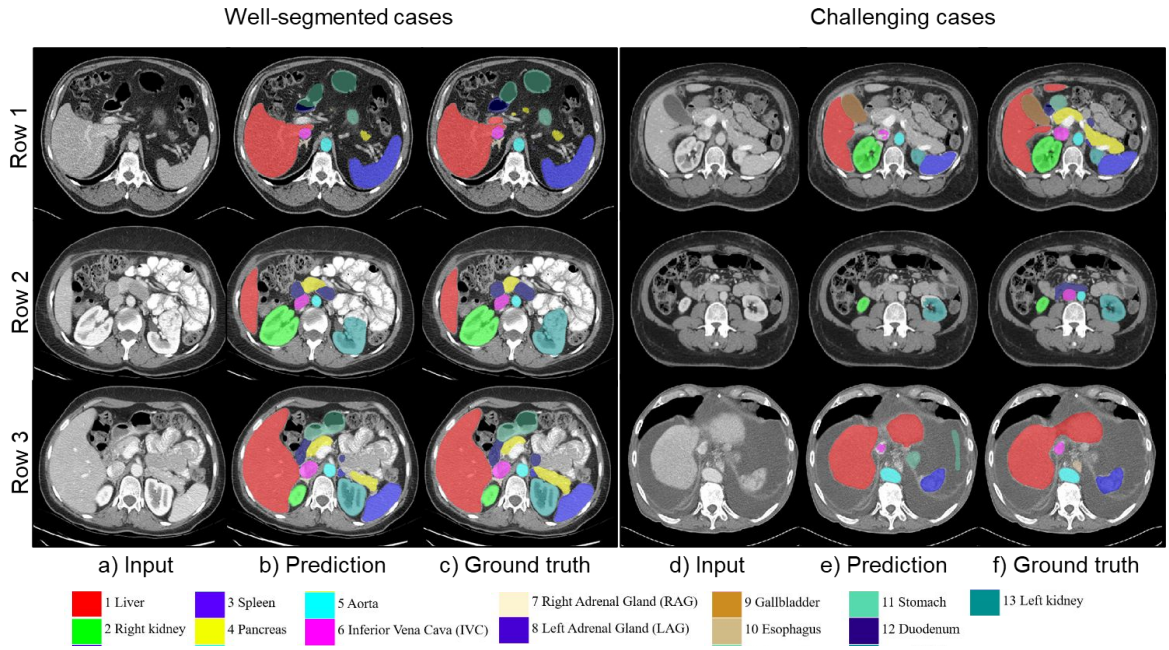

Figure 5.1: Qualitative results from the validation set. We illustrate both well-segmented and challenging examples for our proposed segmentation pipeline

Furthermore, due to our two-staged pipeline, for the results of the second stage to be good really relies on the first stage's performance. If the reference stage miss-segments any organ, that one will be missed during the entire propagation process. Having said that, this issue mostly just occurs in organs that have a short-size length on the axial plane.

5.3.2 Improvement of Mask Propagation

Figure ?? and Figure ?? showcases the refinement results of our proposed Propagation module. It can be recognized that the "Without propagation" columns represent the output masks of the Reference stage only. We can see that with the second stage applied, some organ masks become smoother and more precise along the coronal and sagittal axis (the left kidney in Figure ?? and both the gallbladder, pancreas in Figure ??).

While the masks from the first stage are usually inconsistent along the tem-

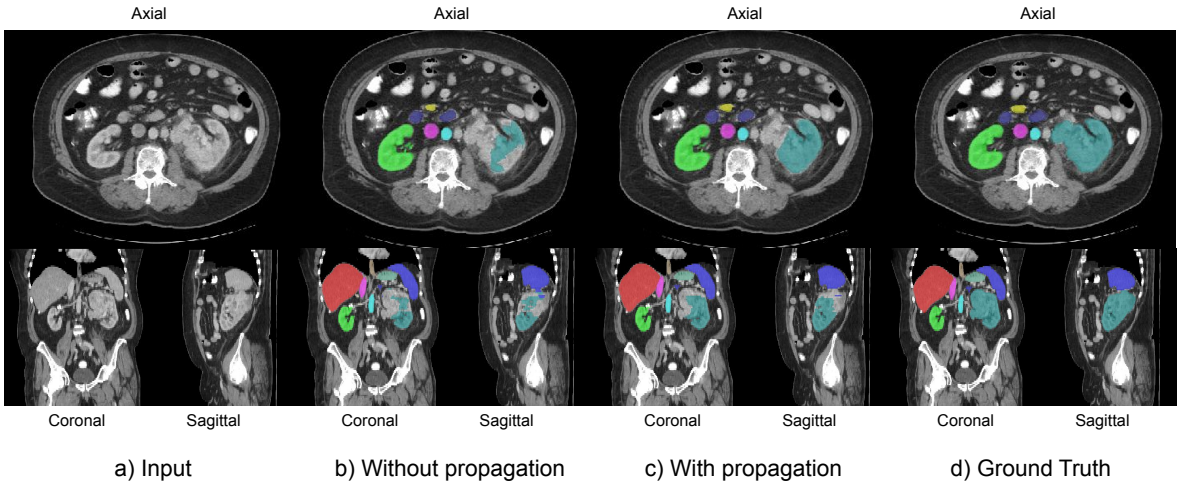

Figure 5.2: Qualitative comparison between before and after mask-propagated refinement.

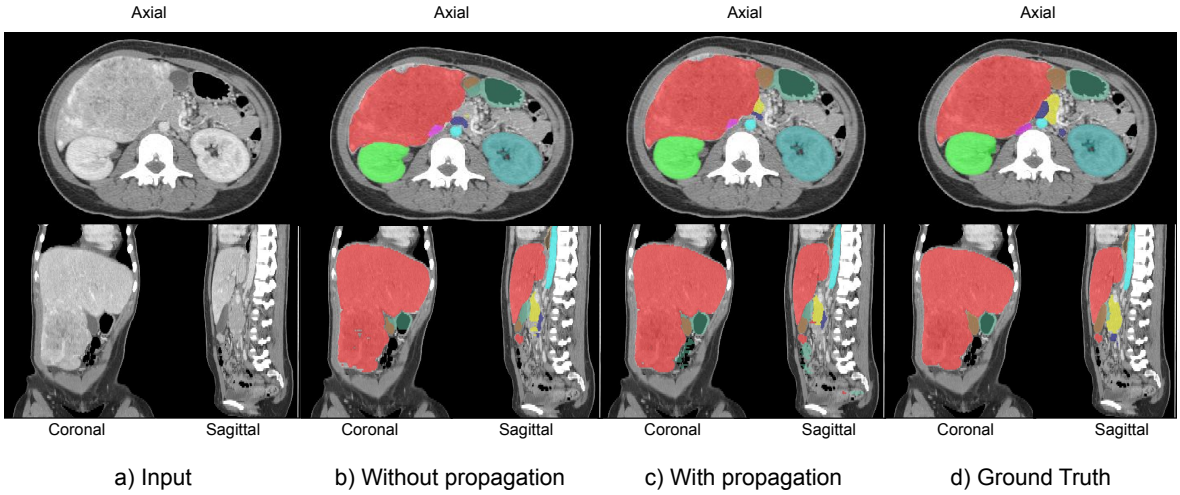

Figure 5.3: Another qualitative comparison between before and after mask-propagated refinement

poral dimension, even between two consecutive frames, the Mask Propagation algorithm helps to stabilize the differences between them.

5.3.3 Improvement of Cross Pseudo Supervision

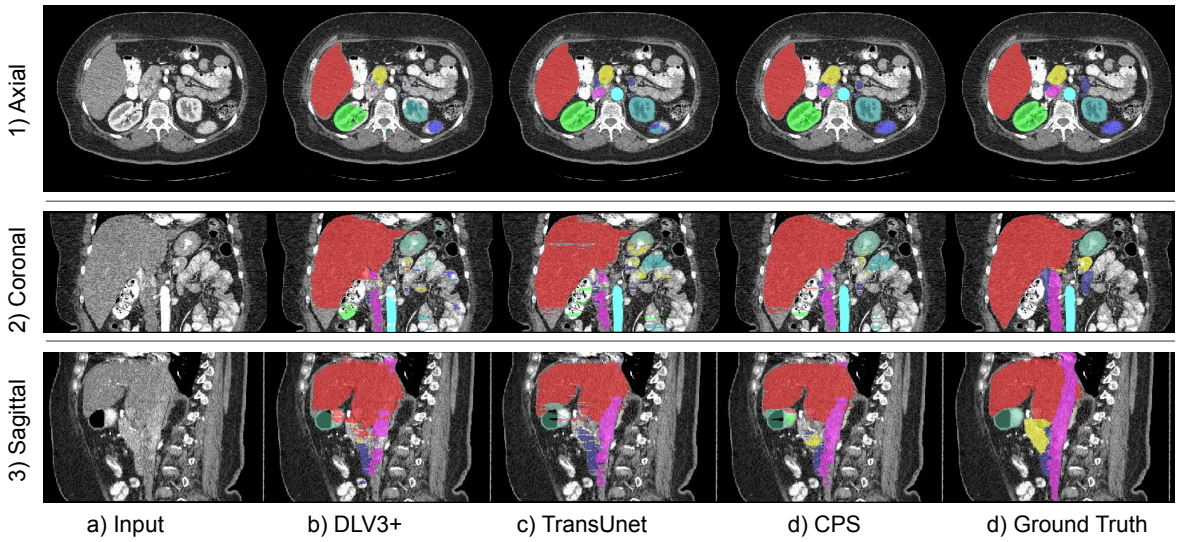

Figure 5.4: Comparison between before and after using Cross Pseudo Supervision in the Reference stage

The effectiveness of using CPS can be seen in Figure **??**. It seems that the DeeplabV3+ and the TransUnet suffer from segmenting some different specific organs with small areas. CPS, on the other hand, gives more stable results (the small part of the spleen in the bottom right in the axial view, or the reduction in inconsistency along the sagittal plane, both in Figure **??**).

5.3.4 Improvement of Positional Encoding

The positional encoding includes the temporal information in the input of the Reference module. Therefore, it is expected to reduce the discordance between

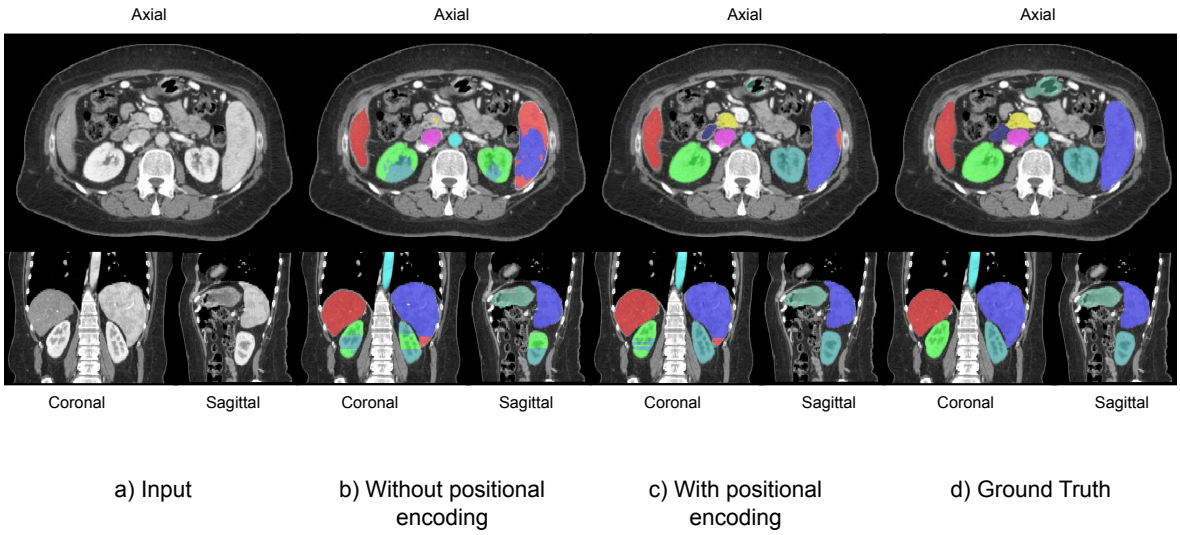

Figure 5.5: Comparison between before and after integrating positional encoding into the TransUnet in the Reference stage

the sequence of slices. For example, the right kidney in Figure **??** b), viewed in coronal and sagittal, is constantly mistaken for the right kidney. With the introduction of positional information, it is segmented accordingly. Moreover, it also accurately identifies the location of the pancreas and the left adrenal gland.

5.3.5 Improvement of Pseudo Labeling with Uncertainty Estimation

It is clear that the contribution of pseudo-labeling is significant to our final results. In Figure **??**, most of the organ masks are notably enhanced after re-training with pseudo labels. However, we still have yet to resolve the imbalanced problem, which leads to the difficulty in identifying small organs like the gall-bladder in the Figure.

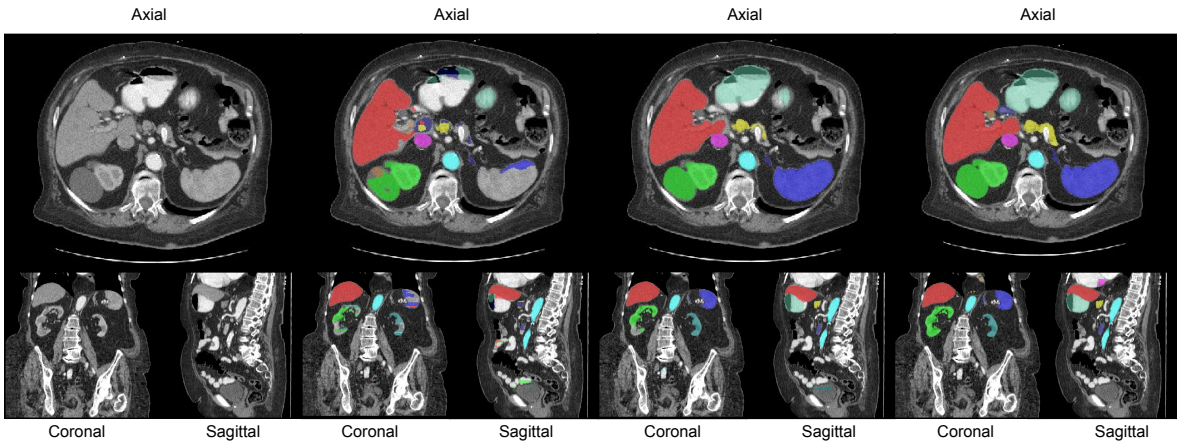

Figure 5.6: Comparison between before using pseudo labeling and after retraining the model with pseudo labeling

CHAPTER 6

APPLICATION

In this chapter, we present the implemented application of our proposed method for interactive video volume segmentation, including a annotation tool.

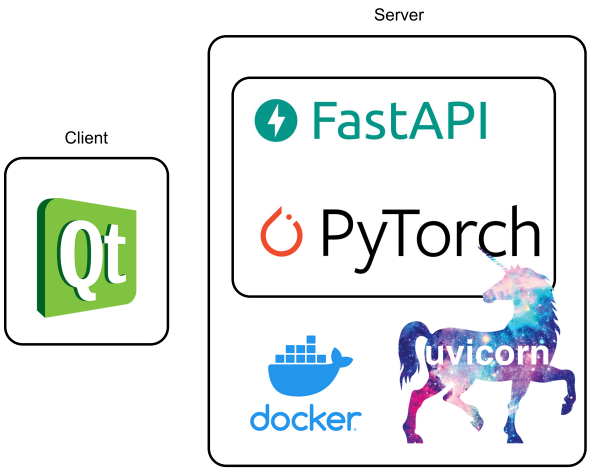

Figure 6.1: Overall architecture of application

To prove our claim before, that our concept can easily adapt with out-of-domain data, we introduce an annotation tool. As usual, it comprises of two separated sides, the client and the server. The back-end server is hosted based on requirements of the provided service and communicate with the Graphic User Interface (GUI), which runs on the client machine, through REST API via HTTPs protocol. This setup potentially helps reduce the dependency between the client and server, each side can be developed and deployed independently. We use the QT framework to build a simple GUI for our application while the server is wrapped up by Uvicorn, which is an ASGI web server implementation in Python. Docker is also used to pack the whole server into a bundle for maintaining version compatibility (figure **??**). The application is used to demonstrate our proposed

method for interactive volume organs segmentation.

6.0.1 Main features

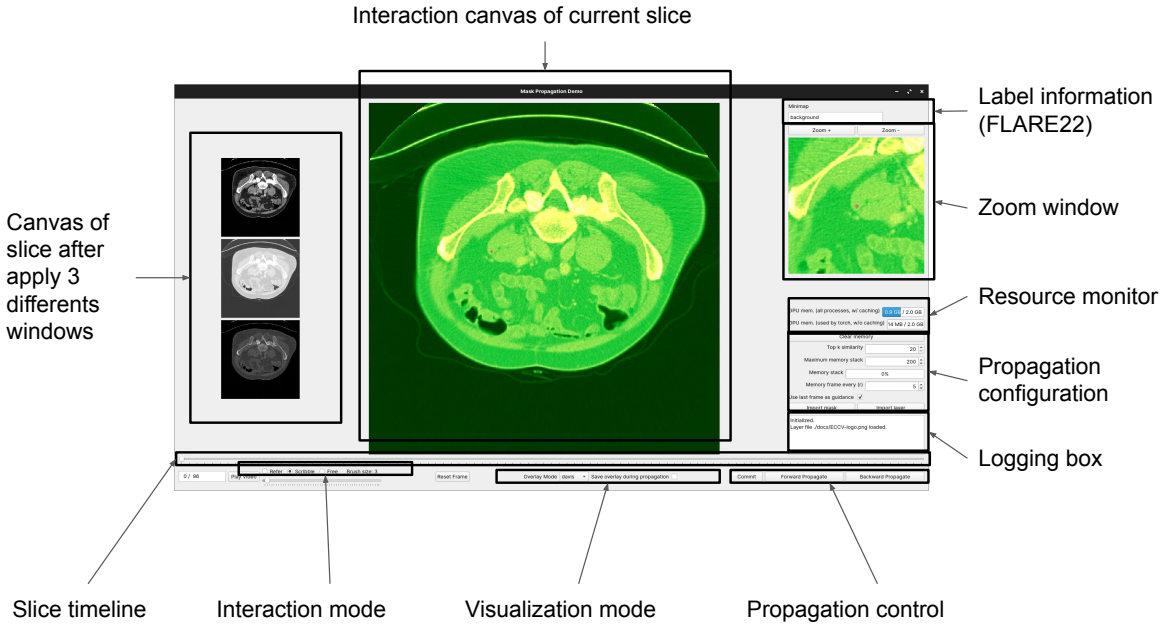

Figure 6.2: The layout of our annotation tool

Selecting and previewing the annotated slices: By using the **slice timeline**, the annotator can choose any arbitrary slice just by dragging the slider. In each timestamp, two main canvas are shown: on the left side is the visualization of applied image on different windows due to the aforementioned pre-processing stage (as in **??**); The larger canvas in the middle displays the stacked version of the 3 images as a RGB image. We also provide the user with an additional canvas on the left side, which illustrates the zooming view of the main canvas for small details interaction. The zoom-in ratio could be modified by available controller buttons, or by scrolling the mouse wheel.Users can also navigate through the timeline by pressing the left arrow key and the right arrow key.

Performing slice annotation: We provide 3 type of annotation interactions

including: scribbles to mask (s2m), free scribbles and automatic reference. In the free scribble mode, the user can freely draw on the canvas to create object masks. Meanwhile, The s2m and reference options offer the ability to use deep learning models to automatically generate the masks. The list of available objects can be shown by pressing "L". Then to choose a specific object that the user desire to label, corresponding keypad should be pressed. For example, to annotate an object that belongs to class "1", keypad "1" should be pressed. In both the scribbles modes, the user can use left and right mouse buttons to draw new masks or to erase existing masks respectively. After finishing, the user can press space to obtain the mask. With the initial mask formed, the user can perform forward/backward propagation to disseminate the annotated mask to other slices across the timeline. The **propagation control** provides options to accomplish that.

The monitors: Stats are printed in the **logging box**. The **resource monitor** also displays the current usage of GPU memory. Label information is also shown on the top right, indicating the information of the object that is being hovered on.

6.0.2 Annotation flow

The annotation flow in Figure **??** shown in the diagram illustrates how users, clients, and servers interact with the other. After initialization, the annotator will select valuable slices for the refinement stage. The client will then wrap these slices and send them to the server. The request and response process will be communicated through a proxy. After the server has done its prediction on the whole volume image based on annotations from the refinements made by the annotator, this process will repeat until an acceptable level of results is achieved.

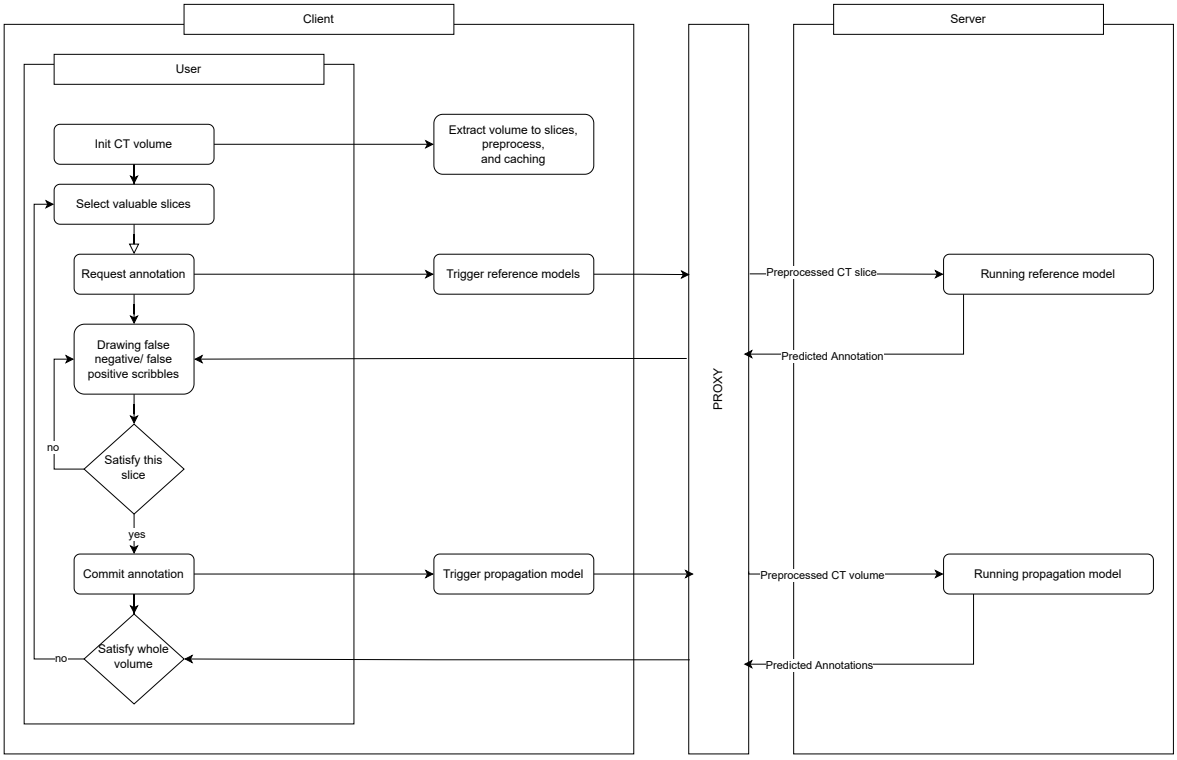

Figure 6.3: The annotation flow for a CT volume

6.0.2.1 Request parsing

The serialization and deserialization processes is very important when it comes to exchanging data between the client and server. By using request parsing, we are able to use JSON for both the client and server messages. This makes the process quick and easy, while also ensuring that all of the data is properly transferred. There are some basic types that can be easily jsonified, such as numbers and strings. However, images and masks need to be encoded using the latin-1 encoder in order to ensure successful transmission.

6.0.3 Future works

In the thesis range, the application tool is not strictly required, but the system design must be flexible and scalable. Our implementation satisfies preliminary targets, but there are many points that can still be improved. For example,

data management, cloud storing or transferring process optimization can all be enhanced through caching or similar techniques.

CHAPTER 7
CONCLUSION

In this chapter, we report the results and discuss about the future works for improving our proposed method and its applications. In our thesis, we proposed a novel pipeline to tackle the problem of CT volume organ segmentation along with an interactive tool that can help labeling process become less complicated. Overall, through ablation study, our method shows improvements over the original baseline, yet still have weaknesses. If these shortcomings can be resolved, we believe it can bring many breakthrough for both the deep leaning and medical fields. Having said that, it is left for the future works.

7.1 Results

With new advancements in technology, there are endless possibilities for what can be achieved. In the medical field, one of the most common problems that doctors face is accurately segmenting 3D objects from 2D images or volume sequences. We propose a novel two-stage pipeline that can leverage the strength of many state-of-the-art 2D deep learning algorithms and techniques in videos and images, into the task of 3D object segmentation. This proposal aims to introduce a novel and inspirational approach to solving one of the most common problems in the medical field. In addition, to break the barrier of differences in medical pipeline processes, our solution is able to transfer and exploit the power of multiple domain data to create more accurate results.

In summary, we enhance the initial results by presenting the application of new techniques and modules:

- The semi-supervised Cross Pseudo Supervision module helps to exploit the enormous amount of unlabeled data successfully. According to our abla-

tion study, with the usage of only a quarter of the unlabeled source, the validation results were boosted significantly. Additionally, this module also contributes to the pseudo-labeling stage to generate more data for other module to learn.

- The Propagation module which inherit the power of STCN model with some modification for adaptation. By using this module, the initial results were refined to obtain better masks. This module can also be attached to the annotation tool with ease and efficiency. However, its performance is strictly dependent on the precision of the previous stage, which is the Reference module.

- The rational uncertainty estimation approach for the pseudo-labeling process, which helps the process becomes more explainable and productive. Despite that, its simplicity allows it to be further analyzed and upgraded in the future.

On top of that, we also come up with a user-friendly annotation tool that provides support and guidance for doctor to quickly generate usable data for the medical field.

Although our research shows improvements over traditional 2D methods, it still cannot be comparable to other 3D methods, as can be seen on the leaderboard of FLARE22 challenge, since those methods can comprehend the dimensional information of a full CT volume. Officially, our method achieves the final result of 0.78 DSC. Still, our method does introduce a novel approach yet promising one and can be further developed in the future.

7.2 Future works

It is undouted that there is a lot of space for our method to improve. Although it is more resource-efficient than the original method, it requires huge memory bank to store information during the testing phase. There have been many recent research with solution to alleviate this problem by using multiple memory storage mechanism [**?**], or an identity assignment bank to associate multiple objects at once [**?**]. Unfortunately, at the time they are introduced, we have nearly come to the end of our thesis. It would be highly recommended for readers to look into these research.

Some drawback of the model comes from the unique property of the dataset, however we lack of expert knowledge to carefully look at those error case by case. It is encouraged in the future for researchers to perform in-depth investigation to obtain more valuable insights.

APPENDICES

LIST OF PUBLICATIONS

This section contains the paper of ours for the proposed method. It is currently under review at MICCAI 2022, for the FLARE22 Challenge. As regards to the official score on the private testset, it will be announced at MICCAI 2022.

Semi-supervised organ segmentation with Mask Propagation Refinement and Uncertainty Estimation for Data Generation

Minh-Khoi Pham[1,2][0000-0003-3211-9076], Thang-Long Nguyen-Ho[1,2][0000-0003-1953-7679] *, and Minh-Triet Tran[1,2,3][0000-0003-3046-3041]

[1] University of Science, Ho Chi Minh City, Vietnam
[2] Viet Nam National University, Ho Chi Minh City, Vietnam
[3] John von Neumann Institute, Ho Chi Minh City, Vietnam

Abstract. We present a novel two-staged method that employs various 2D-based techniques to deal with the 3D segmentation task. In most of the previous challenges, it is unlikely for 2D CNNs to be comparable with other 3D CNNs since 2D models can hardly capture the temporal information. In light of that, we propose using the recent state-of-the-art technique in video object segmentation, combining with other semi-supervised training techniques to leverage the extensive unlabeled data. Moreover, we also generate pseudo-labeled data that is both plausible and consistent for further retraining by using uncertainty estimation. Overall, our method achieves ... on the validation set of FLARE22.

Keywords: 2D semi-supervised segmentation · Mask Propagation · Uncertainty Estimation

1 Introduction

Organs segmentation hold an important step in clinical stage because it affects factors such as abnormal organ detection, disease diagnosis, etc. However, quantifying agencies accurately is quite expensive and time consuming, while medical labels need to be evaluated and labeled by experts to ensure accuracy before releasing a valuable data set to the community. Because of the lack of data or human resources with high expertise in labeling, or the trade off between spending too much time on labeling, the work of medical staff must always be prioritized, so it is very difficult to generate data when human resources are scarce.

In the supervised scenario, that is time consuming to manually annotate the pixel locations of interest. As for the unsupervised lack in the low-quality segmentation mask, it is difficult to apply in real scenario. With the semisupervise approach, we can take advantage of the labeled data combined with the unlabelled data, the approach still provide accurate evaluation based on the labeled data.

* The first two authors share the equal contribution.

However, in any scenario, data labeling must be done manually and sequentially to ensure absolute accuracy in medical data. But now, state of the art method are gradually proposed, but there is not aim to the reusability or adaptability, so when labeling a completely new data, the work does not take advantage of the knowledge from previous models. Our propose is not just about how we utilize unlabelled data, but also the potential of how we can reuse it to help with efficient data generation.

The objective of our approach is to propose a novel method for interactive scenario, which is a semi-automated process include interact and refine, this labeling process speeds up the annotate process. By recommending propagation module using binary manner learning strategy, we do not need to care about the number of classes when doing the inference stage. The model can understand the context or class we want to reference, and then infer the entire volume. Proposed architecture design referencer that takes advantage of unlabelled data and suggests the propagation model valuable slices. The propagation result is satisfying all designed goals, including quickly, propagated less often to satisfy results, the results are always exactly comparable to the reference level.

2 Method

2.1 Preprocessing

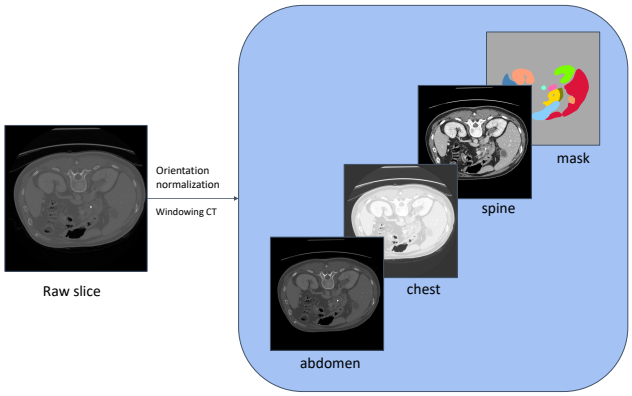

Fig. 1. Windowing CT

With some organs that are relatively small, it might be better to keep the original size of the slices without any cropping, resampling or resizing methods.

Because the Hounsfield Unit scale has a very wide range, and might not perform well with the approach on conventional visual models, we apply a window

based on prior knowledge to display map to 256 range. The motivation is with disease tissue it is only visible when applying windows, if we use raw data, the diseased tissue is mixed with normal tissue, it is almost impossible to extract information from the visual image.

Windowing [2], also known as grey-level mapping, contrast stretching, histogram modification or contrast enhancement is the process in which the CT image grayscale component of an image is manipulated via the CT numbers; doing this will change the appearance of the picture to highlight particular structures. The brightness of the image is adjusted via the window level. The contrast is adjusted via the window width. In or experiments, we create 3 different versions of a single slice by highlight the abdomen, chest, and spine groups and stack it to one as a three-channel image (Fig. 1).

In addition, we choose the axial plane to cut the slices from the CT volumes since this plane has various dimension sizes. The image is rotated to a predefined angle, then is divided by 255 for normalization before going through the next step.

2.2 Proposed Method

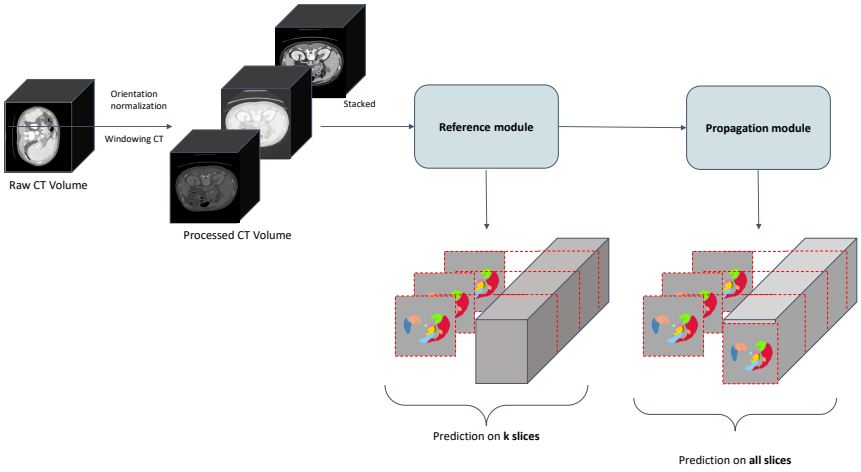

Fig. 2. Our overall proposed pipeline. Firstly, the entire CT Volume is processed using windowing CT to get a stack of three-channel slices. Then the slices progress through the reference modules to obtain minimal number of preliminary masks. Lastly, the Propagation module refine these initial masks to finalize the result.

Our method composes of two main modules: Reference module and Propagation module, as can be seen in Fig 2.

In the beginning, we heuristically select only a small number of slices from the CT Volume to be our initial candidates. Next, these slices are processed by using Windowing technique 2. Afterwards, these slices are put through the Reference module, in which performs the standard multiclass segmentation, then the preliminary k masks can be obtained. Based on these pairs of potential slices and masks as prior knowledge, the Propagation module can utilize them to propagate the objects transformation information to the remaining slices across the CT volume length. The final output of this module is a 3D dense mask prediction, with each voxel indicating a class.

Reference module This module is expected to provide a suggestion of a minimal amount of slices and predicted masks that might contains the most information describing the entire CT Volume. Fig 3 describes the details of this module.

To utilize the enormous number of unlabeled data, we apply the recent semi-supervised method that performs effectively on several other datasets, which is called Cross Pseudo Supervision (CPS) [7] (yellow cube in Fig. 3. CPS enables the usage of unlabeled data by following the dual students technique, where two models are trained simultaneously on labeled data while generating pseudo data for their "peer" to learn. In the testing phase, two models predict on the same image and the result is aggregated by summing up.

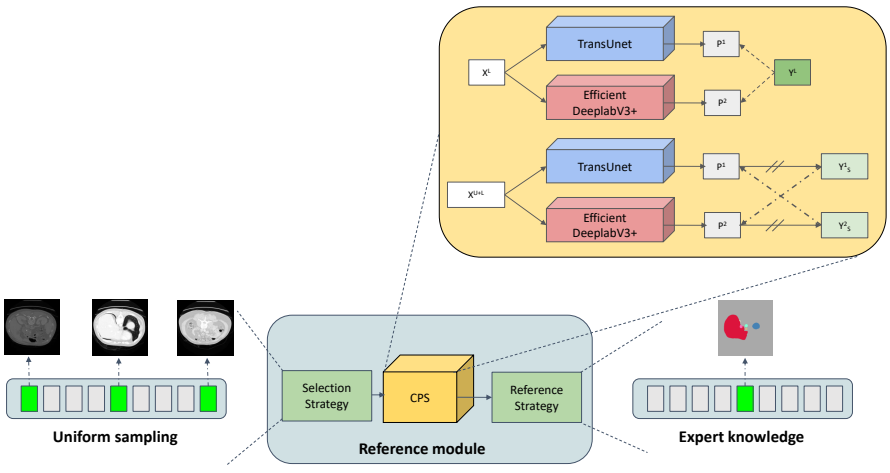

Fig. 3. The reference module. The semi-supervised technique CPS is applied in both training and inference stage to enhance the precision of model prediction. Strategies are used to smartly choose slices that are informative for the next stage.

We adopt two prominent state-of-the-arts 2D segmentation models with highly different learning paradigms for this CPS framework, which is TransUNet [5] and DeeplabV3+ [6].

While DeeplabV3+ traditionally focus more on the local information, transformers model the long-range relation, so the cross training can help to learn a unified segmenter with these two properties at the same time. In short, we choose TransUNet and DeeplabV3+ due to the their ability to compensate each other for better performance. [13]

In addition, we also propose a both logical and specialist-based strategy to choose which slices that can be further used to boost the performance of the Propagation module. The goal of this action is to preserve only some of the most useful information for the refinement stage.

To elaborate on these strategies, prior to being put into the CPS module for prediction, a small number of slices are uniformly sampled from the processed CT volume. After CPS produces segmentation masks for these slices, another selection step is performed to picking only some of the masks that contains the organs having the largest areas.

Propagation module This module aims to utilize prior knowledge of given annotated slices from the Reference module to make prediction on the remaining slices, this mechanism can be referred as mask (or label) propagation.

Intuitively, the conventional 2D CNNs cannot comprehend the third dimension information within a CT volume. Thus, in hope of the ability to capture the "temporal" information along the axial plane, we adapt the Space-Time Correspondence Networks (STCN) [8], which is a semi-supervised segmentation algorithm that has achieved promising results on Video object segmentation problem, to this 3D manner.

Basically, STCN proposes the use of memory bank that stores information of previous frames and their corresponding masks, and uses them later as prior knowledge. To generate the mask for the current frame, a pairwise affinity matrix is calculated between the query frame and memory frames based on negative squared Euclidean distance, then it is used for supporting the current mask generation. [8]

Different from the original STCN, we slightly modify it to match the current problem. As the original work, they use only a single dense mask to propagate through the entire video, therefore for the model to perfectly work, that selected mask must contains information of all available classes. For our case to achieve that, we enable the usage of multiple masks for propagation, so that all of these mask should contain enough information of every organs. We also allow the STCN to work in a bidirectional way to enhance the refinement. Fig 4 illustrates this process.

Specially, STCN can be simply trained in the binary manner, meaning that each of the abdominal organs can be learned separately. Therefore, the knowledge can be transferred well between different organ classes.

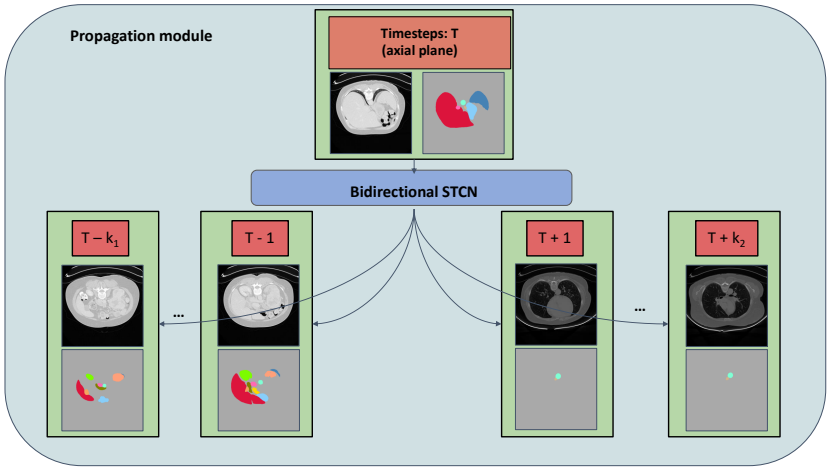

Fig. 4. The propagation module. From an annotated slice of CT, at timestep T, STCN can make use of that to spread the information through the entire defined range $[T - k_1, T + k_2]$.

Pseudo labeling with Uncertainty Estimation Given a vast amount of unlabeled CT volumes, we apply a uncertainty estimation technique to effectively maximize the utilization of the data.

Firstly, several CPS models are trained on the provided labeled data. Then, we use these trained CPS models to obtain pseudo masks on the unlabeled set. Inspired from [20], we calculate the dice scores between these pseudo masks and the aggregated one. The mean of these dice scores will be compared with a threshold to determine whether the pseudo masks are qualified. Simply speaking, consensus-based assessment is used to evaluate the quality of pseudo labels.

All samples that have high certainty are then reused for the next supervised training cycle. And after the training finishes, the same labeling process is repeated until all aforementioned models achieve satisfied performance or every unlabeled data has been used.

Loss function For the Reference module, we use the prevalent combination of dice loss and cross entropy loss with smoothing value to alleviate the imbalanced number of the small organs, which occurs due to our splitting into slices process. The same settings are used for CPS in its supervised branch whereas only the dice loss is setup for the unsupervised branch.

For the Propagation module, we implement the online hard example cross entropy (OhemCE or Bootstrapping CE) [21] and also calculate the Lovasz loss [3] at the same time. OhemCE can help reduce the contribution of background label to the final loss, Since STCN training on binary task, OhemCE can di-

rect the model to focus on visible difficult objects. Meanwhile, Lovasz loss is commonly used in the past.

2.3 Post-processing

We do not use any post-processing techniques because no complex pre-processing ones are used, and we conduct all our experiment on the nearly-original image volumes apart from the orientation settings. Thus, before submitting to the evaluation system, the mask must be transformed back to the original orientation.

3 Experiments

3.1 Dataset and evaluation measures

The FLARE2022 dataset is curated from more than 20 medical groups under the license permission, including MSD [19], KiTS [10,11], AbdomenCT-1K [15], and TCIA [9]. The training set includes 50 labelled CT scans with pancreas disease and 2000 unlabelled CT scans with liver, kidney, spleen, or pancreas diseases. The validation set includes 50 CT scans with liver, kidney, spleen, or pancreas diseases. The testing set includes 200 CT scans where 100 cases has liver, kidney, spleen, or pancreas diseases and the other 100 cases has uterine corpus endometrial, urothelial bladder, stomach, sarcomas, or ovarian diseases. All the CT scans only have image information and the center information is not available.

The evaluation measures consist of two accuracy measures: Dice Similarity Coefficient (DSC) and Normalized Surface Dice (NSD), and three running efficiency measures: running time, area under GPU memory-time curve, and area under CPU utilization-time curve. All measures will be used to compute the ranking. Moreover, the GPU memory consumption has a 2 GB tolerance.

3.2 Implementation details

Environment settings The development environments and requirements are presented in Table 1.

Training protocols Currently, we find that using only simple 2D transform functions such as horizontal/vertical flipping or rotating might be enough for both modules to generalize.

4 Results & Discussion

Validation results

Qualitative results later

Table 1. Development environments and requirements.

Windows/Ubuntu version	Ubuntu 18.04.5 LTS
CPU	Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz
RAM	1×32GB;
GPU (number and type)	One Quadro RTX 5000 16G
CUDA version	11.6
Programming language	Python 3.10
Deep learning framework	Pytorch (Torch 1.11.0, torchvision 0.12.0)
Specific dependencies	
(Optional) Link to code	

Table 2. Training protocols for Reference module: CPS of TransUnet and Efficientnet DeeplabV3+

Network initialization	Random initialization
Batch size	2 (labeled) + 2 (unlabeled)
Patch size	None
Total iterations	50000
Optimizer	AdamW
Initial learning rate (lr)	0.0001
Lr decay schedule	multiplied by 0.5 for every iteration at $[40000, 45000]$
Training time	48 hours
Number of model parameters	117,597,362 [4]
Number of flops	[5]
CO_2eq	$_Kg$[6]

Table 3. Training protocols for Propagation module: STCN with Resnet backbone

Network initialization	Random initialization
Batch size	8
Patch size	None
Total iterations	50000
Optimizer	AdamW
Initial learning rate (lr)	0.0001
Lr decay schedule	multiplied by 0.5 for every iteration at $[40000, 45000]$
Training time	48 hours
Number of model parameters	54,416,065 [7]
Number of flops	[8]
CO_2eq	$_Kg$[9]

Table 4. Ablation experiment on proposed modules and techniques

No.	CPS	Windowing CT	UE	MP	Mean DSC
1					0.547
2	✓				0.762
3	✓	✓	✓		0.770
4	✓	✓	✓	✓	0.784

Limitation and future work Apparently, although our proposed method has yet to achieve the high result, we believe it can be further improved if these limitation that we identify here are solved. First of all, the problem of imbalanced dataset has arisen because we perceive this as a 2D problem. Due to the slices splitting process, small organs (such as pancreas, gallbladder or adrenal glands) only appear in a small amount of slices, while larger objects have wider range of appearance. Therefore, it leads to the problem of imbalanced dataset. We tried some ways to tackle the problem, for instance: smart sampling, or imbalanced loss, however only slightly improvement was seen. Secondly, the proposed approach is a two-stage method, the second stage is undoubtedly dependent of the first one. If there are any organs that are missed by the reference module, it definitely cannot be recovered in the propagation phase. In the future, it is encouraged to focus on boosting the performance of the reference module by fully exploiting the temporal information.

5 Conclusion

In summary, we present a two-stage pipeline, which can leverage the strength of many state-of-the-art 2D deep learning algorithms and techniques in videos and images, into the task of 3D object segmentation. Our proposal aims to introduce a novel and inspirational approach in solving one of the most common problem in the medical field.

6 Acknowledgements

The authors of this paper declare that the segmentation method they implemented for participation in the FLARE 2022 challenge has not used any pretrained models nor additional datasets other than those provided by the organizers.

Furthermore, no manual intervention has been made in the contribution to the results of the proposed method

References

1. NIH Pancreas. https://wiki.cancerimagingarchive.net/display/Public/Pancreas-CT (2020), [Online; Accessed: Aug. 2020]

2. Baba, Y., Murphy, A.: Windowing (ct) (Mar 2017). https://doi.org/10.53347/rid-52108, http://dx.doi.org/10.53347/rID-52108 3
3. Berman, M., Triki, A.R., Blaschko, M.B.: The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 4413–4421. Computer Vision Foundation / IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00464, http://openaccess.thecvf.com/content_cvpr_2018/html/Berman_The_LovaSz-Softmax_Loss_CVPR_2018_paper.html 6
4. Bilic, P., Christ, P.F., Vorontsov, E., Chlebus, G., Chen, H., Dou, Q., Fu, C.W., Han, X., Heng, P.A., Hesser, J., et al.: The liver tumor segmentation benchmark (lits). arXiv preprint arXiv:1901.04056 (2019)
5. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: Transunet: Transformers make strong encoders for medical image segmentation. CoRR **abs/2102.04306** (2021), https://arxiv.org/abs/2102.04306 5
6. Chen, L., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII. Lecture Notes in Computer Science, vol. 11211, pp. 833–851. Springer (2018). https://doi.org/10.1007/978-3-030-01234-2_49, https://doi.org/10.1007/978-3-030-01234-2_49 5
7. Chen, X., Yuan, Y., Zeng, G., Wang, J.: Semi-supervised semantic segmentation with cross pseudo supervision. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 2613–2622. Computer Vision Foundation / IEEE (2021), https://openaccess.thecvf.com/content/CVPR2021/html/Chen_Semi-Supervised_Semantic_Segmentation_With_Cross_Pseudo_Supervision_CVPR_2021_paper.html 4
8. Cheng, H.K., Tai, Y., Tang, C.: Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual. pp. 11781–11794 (2021), https://proceedings.neurips.cc/paper/2021/hash/61b4a64be663682e8cb037d9719ad8cd-Abstract.html 5
9. Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., et al.: The cancer imaging archive (tcia): maintaining and operating a public information repository. Journal of Digital Imaging **26**(6), 1045–1057 (2013) 7
10. Heller, N., Isensee, F., Maier-Hein, K.H., Hou, X., Xie, C., Li, F., Nan, Y., Mu, G., Lin, Z., Han, M., et al.: The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: Results of the kits19 challenge. Medical Image Analysis **67**, 101821 (2021) 7
11. Heller, N., McSweeney, S., Peterson, M.T., Peterson, S., Rickman, J., Stai, B., Tejpaul, R., Oestreich, M., Blake, P., Rosenberg, J., et al.: An international challenge to use artificial intelligence to define the state-of-the-art in kidney and kidney tumor segmentation in ct imaging. American Society of Clinical Oncology **38**(6), 626–626 (2020) 7

12. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature Methods **18**(2), 203–211 (2021)
13. Luo, X., Hu, M., Song, T., Wang, G., Zhang, S.: Semi-supervised medical image segmentation via cross teaching between CNN and transformer. CoRR **abs/2112.04894** (2021), https://arxiv.org/abs/2112.04894 5
14. Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., Martel, A.L.: Loss odyssey in medical image segmentation. Medical Image Analysis **71**, 102035 (2021)
15. Ma, J., Zhang, Y., Gu, S., Zhu, C., Ge, C., Zhang, Y., An, X., Wang, C., Wang, Q., Liu, X., Cao, S., Zhang, Q., Liu, S., Wang, Y., Li, Y., He, J., Yang, X.: Abdomenct-1k: Is abdominal organ segmentation a solved problem? IEEE Transactions on Pattern Analysis and Machine Intelligence (2021). https://doi.org/10.1109/TPAMI.2021.3100536 7
16. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241 (2015)
17. Roth, H., Farag, A., Turkbey, E., Lu, L., Liu, J., Summers, R.: Data from pancreas-ct. the cancer imaging archive (2016)
18. Roth, H.R., Lu, L., Farag, A., Shin, H.C., Liu, J., Turkbey, E.B., Summers, R.M.: Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation. In: International conference on medical image computing and computer-assisted intervention. pp. 556–564. Springer (2015)
19. Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., Van Ginneken, B., Kopp-Schneider, A., Landman, B.A., Litjens, G., Menze, B., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063 (2019) 7
20. Wang, J., Chen, Z., Wang, L., Zhou, Q.: An active learning with two-step query for medical image segmentation. In: 2019 International Conference on Medical Imaging Physics and Engineering (ICMIPE). pp. 1–5. IEEE (2019) 6
21. Wu, Z., Shen, C., van den Hengel, A.: High-performance semantic segmentation using very deep fully convolutional networks. CoRR **abs/1604.04339** (2016), http://arxiv.org/abs/1604.04339 6