Deep Learning Methods to Identify Human Cranium, Brain Ventricles, and Intracranial Hemorrhage Using Tissue Pulsatility Ultrasound Imaging

Nhut Minh Phan

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Computer Science & Software Engineering

University of Washington

2022

Reading Committee:

Erika Parsons, Chair

Pierre Mourad

Michael Stiber

Program Authorized to Offer Degree:

Computer Science & Software Engineering

University of Washington

Abstract

Deep Learning Methods to Identify Human Cranium, Brain Ventricles, and Intracranial Hemorrhage Using Tissue Pulsatility Ultrasound Imaging

Nhut Minh Phan

Chair of the Supervisory Committee:
Dr. Erika Parsons
Computer Science & Software Engineering

Traumatic Brain Injury (TBI) is a serious medical condition when a person experiences trauma in the head, resulting in intracranial hemorrhage (bleeding) and potential deformation of head-enclosed anatomical structures. Detecting these abnormalities early is the key to saving lives and improving survival outcomes. Standard methods of detecting intracranial hemorrhage are Computed Tomography (CT) and Magnetic Resonant Imaging (MRI). However, they are not readily available on the battlefield and in low-income settings. A team of researchers from the University of Washington developed a novel ultrasound signal processing technique called Tissue Pulsatility Imaging (TPI) that operates on raw ultrasound data collected using a hand-held tablet-like ultrasound device. This research work aims to build segmentation deep-learning models that take the input TPI data and detect the skull, ventricles, and intracranial hemorrhage in a patient's head. We employed the U-Net architecture and four of its variants for this purpose. Results show that the proposed methods can segment the brain-enclosing skull and is relatively successful in ventricle detection, while more work is needed to produce a model that can reliably segment intracranial hemorrhage.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

CNN: Convolutional Neural Network

CT: Computer Tomography.

CTBI: closed Traumatic Brain Injury.

GPU: Graphical Processing Unit

IED: Improvised Explosive Device.

IRB: Institutional Review Board.

LSTM: Long Short-Term Memory.

PTBI: penetrating Traumatic Brain Injury.

RELU: Rectified Linear Unit

RNN: Recurrent Neural Network.

ROI: Region of Interest

TBI: Traumatic Brain Injury.

TPI: Tissue Pulsatility Imaging.

WHO: World Health Organization.

ACKNOWLEDGMENTS

I want to express sincere appreciation to Dr. Pierre Mourad and Dr. Michael Stiber for accepting my request to be on the committee for this thesis. I am also grateful for the support received from Dr. John C. Kucewicz and Nina LaPiana. They contributed significantly to data collection, registration, and signal processing to enable this study. Additionally, I want to extend my gratitude to the IT staff at the University of Washington, Bothell. They simplified the remote development experience tremendously. Finally, I offer my special thanks to Dr. Erika Parsons for being the Chair of my committee, and without her support and guidance, this thesis would not have been possible.

DEDICATION

I dedicate this thesis to my family, who have meant and continue to mean so much to me. There is a special feeling of gratitude to my parents, Cac Phan and Nga Tran. Their support has allowed me to reach my academic destination. My sister Anh Phan has never left my side during the challenges of graduate school and life. My wife, Yen Nguyen, whose words of encouragement keep me straight on the path to my goal. I also dedicate this work to my friends, Kevin Mun and Josiah Lim. They cheered me on at the low points and helped prepare me for the Master's defense.

Chapter 1

INTRODUCTION

1.1 Background

Traumatic Brain Injury (TBI) is a serious medical condition that occurs when a sudden physical force causes damage to the brain. It may happen in one of two ways: close brain injury (cTBI) and penetrating brain injury (pTBI) [3]. Closed brain injuries occur when the injury does not penetrate through the skull and brain. These injuries are caused by rapid movements and shaking of head. Penetrating brain injuries happen when a foreign object penetrates the skull and travel through the brain tissue. An example of pTBI is when a bullet goes through the head.

A large percentage of deployed U.S. soldiers (40% to 60% of surviving soldiers) suffer from closed-head injuries caused by the blast effect of an improvised explosive device (IED) explosion [20]. These injuries could result in intracranial hemorrhage, causing long-term neurological damages if left untreated. For severe TBI cases, the patients must be evacuated to the nearest combat hospital with equipment to support neurosurgery, airway protection, mechanical ventilation, and other means for critical care. However, severe cTBI patients often do not survive more than one year after injury [20]. Thus, early diagnosis is essential to improve the clinical outcome and provide medical personnel with information to make decisions when resources are scarce.

Besides the relevance on the battlefield, TBI is pressing public health and medical problems worldwide. According to the World Health Organization (WHO), TBI affects an estimated 10 million people annually [29]. Low and middle-income countries face higher risk factors for causes of TBI due to inadequate health care systems.

Early diagnosis and immediate medical care are critical in improving the clinical outcomes

for TBI patients [4]. Currently, Computer Tomography (CT) and Magnetic Resonance Imaging (MRI) are the standard methods for identifying intracranial hemorrhage [25]. The main disadvantage of these imaging modalities is the complexity, size, and cost of the required equipment, making them inaccessible in combat settings and low-income countries. In contrast, ultrasound imaging could be performed with relatively affordable equipment, which could be as small as a standard tablet. An example of such a system is a tablet-like device from Terason (the company website is at https://www.terason.com). Ultrasound imaging has a major drawback: ultrasound waves do not penetrate bones very well, making ultrasound imaging more suitable for young infants. At this age, the skull is not yet fused completely, allowing the ultrasound waves to penetrate more effectively than the adult's skull does[1].

A team of researchers from the University of Washington developed a novel ultrasound technique called tissue pulsatility imaging (TPI) that captures the pulsation of the brain tissue as blood infuses the brain during a cardiac cycle [36]. The resulting data is named displacement data after the tissue displacement caused by blood flow. The team collected data from civilian patients who suffer moderate to severe cTBI. The working hypothesis is that the difference in the movements of brain tissue versus TBI lesions allows one to detect intracranial hemorrhage through computer-assisted means. This research work aims to employ the power of deep learning to automatically identify intracranial hemorrhage and other essential brain structures from TPI data.

1.2 Problem Statement and Scope

This research focuses on the core algorithm that detects the regions of intracranial hemorrhage within a patient's brain tissue and other abnormalities associated with TBI. Since TBI could also cause damage to the skull and a network of cerebrospinal-fluid-filled cavities called ventricles, the research also aims to segment the skull and ventricles. Therefore, this research's scope is limited to building deep learning models that can detect the skull, the ventricles, and intracranial hemorrhage from displacement data. The proposed models are based on the U-Net [51], an image segmentation model that has been widely used in medical

applications. Our goal is to build more effective deep learning models for the mentioned tasks and evaluate them objectively. We used two different loss functions suited for image segmentation: dice loss and a hybrid loss function. Furthermore, we evaluated the models using the dice coefficient score, Intersection over Union (IoU) score, precision, and recall.

1.3 Report Structure

The rest of this thesis is organized into the following four chapters. In chapter 2, Related works, we discuss the existing supervised learning algorithms that have been successfully implemented to solve medical segmentation problems and related works in detecting intracranial hemorrhage. Chapter 3 provides the details of data collection, data processing, and the implementations of the deep learning models and overall software system. The performance evaluation of the deep learning models and their results are discussed in chapter 4, Experiment and result. Finally, in chapter 5, Conclusion and future work, we summarize the achievements and shortcomings of the research and lay out possible improvements.

Chapter 2
RELATED WORKS

Medical image segmentation aims to identify structures such as organs or lesions in an image. It plays a crucial role in computer aid diagnosis. The two main categories of image segmentation are semantic segmentation and instant segmentation [38]. Semantic segmentation involves assigning a corresponding class for each pixel in an image. In contrast, instant segmentation is more complex, distinguishing instances based on specific categories and outputting the basic semantic segmentation's result. This research focuses on semantic segmentation. For the rest of this thesis, we refer to semantic segmentation when image segmentation is mentioned. Deep learning has been used widely for medical image segmentation with huge success. This chapter discusses the recent progress of deep learning in the field of medical image segmentation. Specifically, we focus on deep learning applications, supervised deep learning techniques, and data augmentation. In addition, we will discuss related works in intracranial hemorrhage detection.

2.1 Applications

Popular medical image segmentation tasks includes, but not limited to, segmentation of the brain regions and brain lesions [42, 13], segmentation of the liver and liver tumors [39, 60], segmentation of the lungs and lung nodules [61, 50], segmentation of the optic nerve head [12, 19], and segmentation of different cell types [51, 55]. The data used for medical diagnosis are commonly 2D gray scale or three-channel RBG images and 3D volumetric data. Deep learning practitioners get their data from the four popular imaging modalities, X-ray, ultrasound, Computed Tomography (CT), and Magnetic Resonance Imaging (MRI) [38], in addition to typical three-channel RBG images. When the structures of interest is at cellular

scale, the input images also come from Electron Microscopy (EM) [51].

2.2 Supervised Learning

Supervised learning is the most popular method for medical image segmentation tasks. This section reviews the techniques that deep learning researchers have attempted to solve medical image segmentation problems. Specifically, we will discuss the two important components of a deep learning model, the backbone network and the network block.

2.2.1 Backbone network

The backbone network is the high-level architecture of a deep learning model. It draws a big picture of how different pixel-level computational operations are chained together to process the input and output of a segmentation map. Researchers have proposed various backbone architectures for deep learning in the last decade, including 2D and 3D U-Net-based models, Recurrent Neural Network (RNN), the skip connection, and cascade models.

U-Net One of the most popular deep learning architectures used for image segmentation is the encoder-decoder structure. U-Net [51], fully convolution network (FCN) [40], and Deeplab [11] are examples of the encoder-decoder structure. Encoder-decoder architectures have two main components: the encoder and the decoder. The encoder reduces the dimensions of the inputs and extracts image features, while the decoder uses these features to determine the locations of the objects of interest and outputs the segmented images. U-Net has been used extensively for various image segmentation tasks and is considered the benchmark to compare the performance of newly developed models [38]. Many modern architectures were inspired by U-Net [62, 65, 33, 5, 28, 10, 9]. Figure 2.1 shows the architecture of the U-Net. The left side of the U-Net works as an encoder. It consists of convolution filters that extract features and max-pooling layers that shrink the inputs. The right side is the decoder that recovers the original image dimensions by four up-convolution operations. The U-Net also has skip connections that concatenate the input features of the encoder to

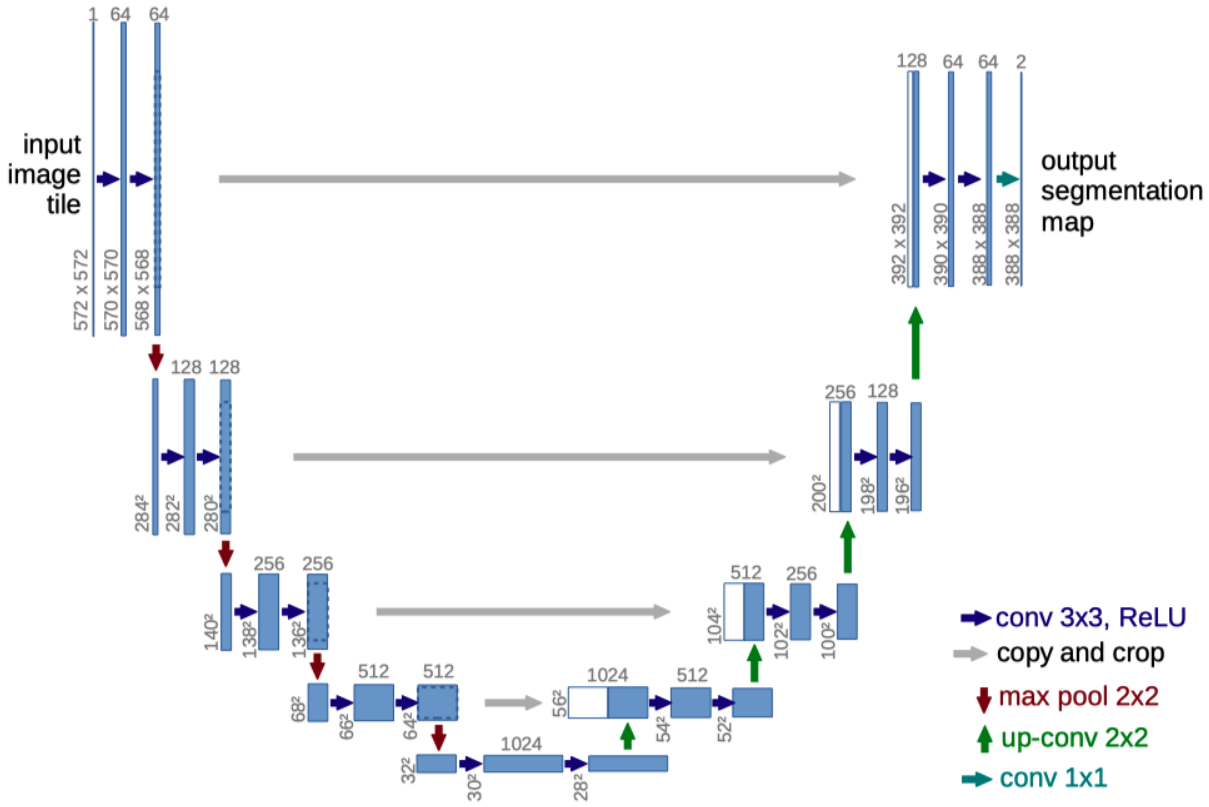

Figure 2.1: The U-Net architecture. The left and right side are the encoder and decoder, respectively[51].

the output of the decoder. This structure effectively fuses low-level (high resolution) with high-level (low resolution) image features through skip connections. These connections allow U-Net to segment medical images, which often contain noise and show blurred boundaries.

3D Network Motivated by the success of U-Net, the authors of the U-Net paper extended the architecture to work with 3D medical data [66]. Such data are common in the form of volumetric CT and MRI data. This architecture has more parameters and requires much higher computing resources than the U-Net. Due to insufficient computing capability, the authors limited the 3D U-Net to only three down-sampling layers, reducing the model's

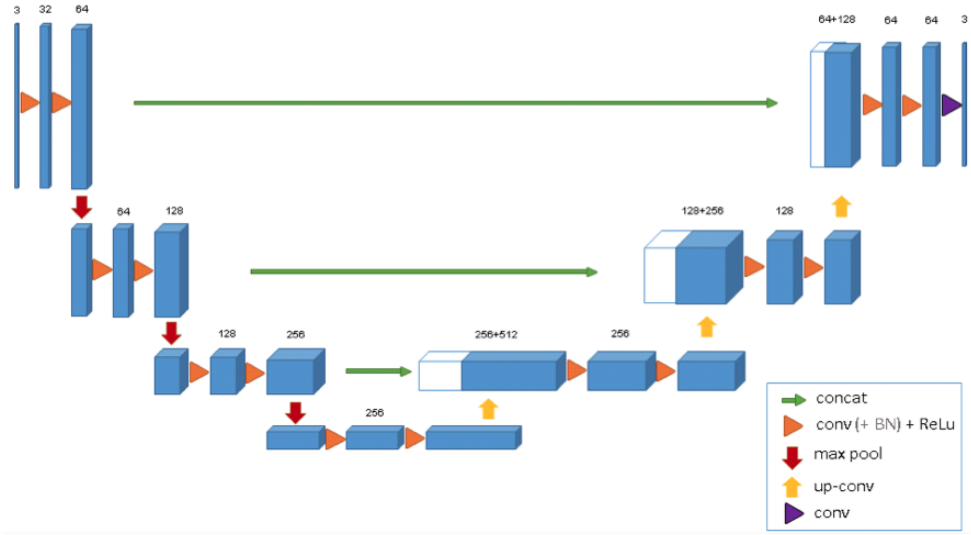

Figure 2.2: The 3D U-Net architecture[66].

segmentation accuracy when tested. Figure 2.2 depicts this architecture. To improve the 3D U-Net, a similar structure called V-Net was developed [45] with the architecture shown in Figure 2.3. V-Net was developed to allow stacking more network layers, providing better feature representation.

One problem that deep neural networks face is vanishing gradient [24]. A neural network's gradients (derivative) are found using backpropagation during the training process. By the chain rule, the derivatives of the hidden layers are multiplied from the output layer to the input layer to compute the derivative of the initial layer. If the derivatives are small, multiplication causes the gradient to decrease quickly, causing a vanishing gradient. Residual connection is one of the solutions for this problem [24]. The residual connection, as shown in Figure 2.4, is essentially a skip connection that brings weights from an earlier stage to a later stage. These weights do not go through the activation function, so they do not suffer from the vanishing gradient problem. The V-Net employs residual connection in the design, allowing it to have four down-sampling layers.

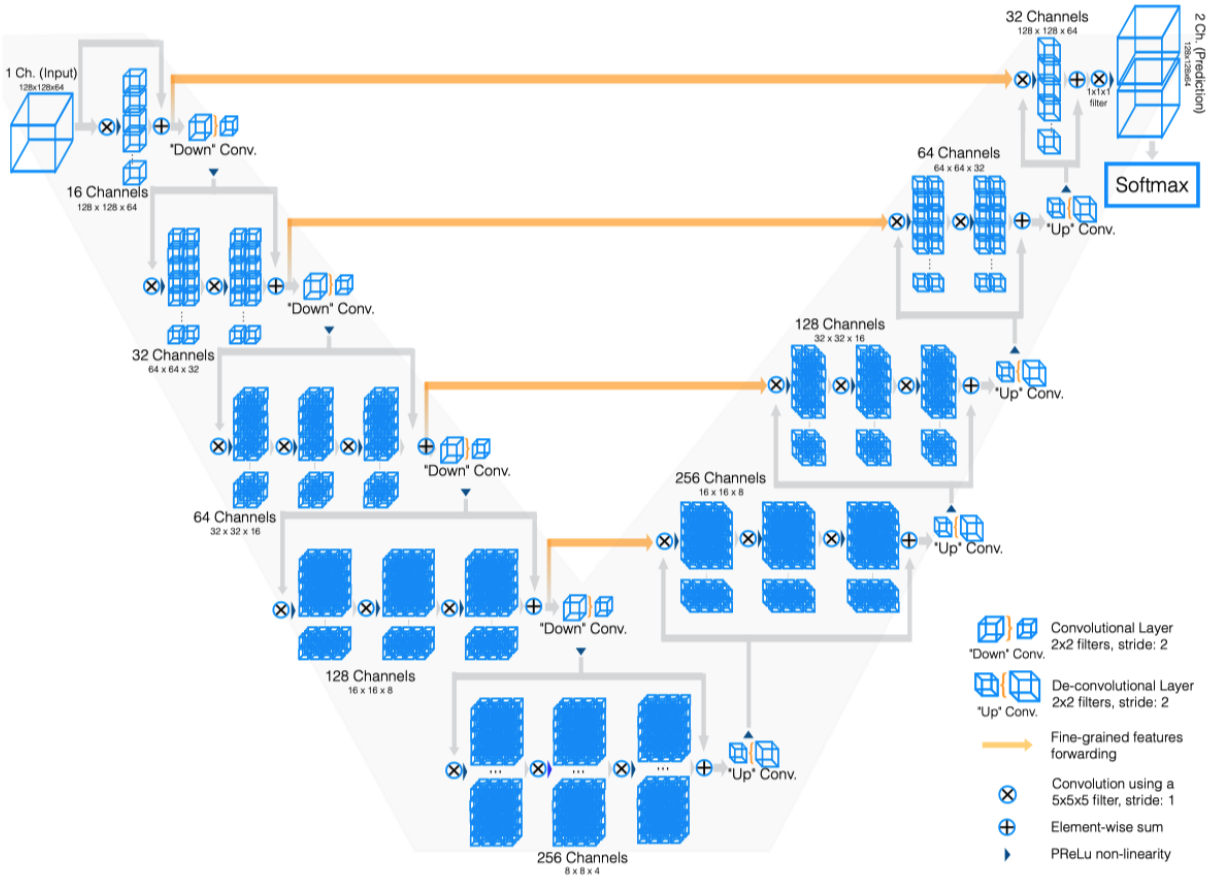

Figure 2.3: The V-Net architecture[45].

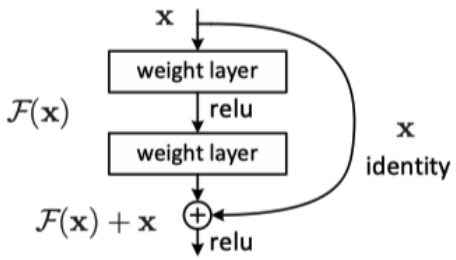

Figure 2.4: Residual connection (identity) [24] adds the input to the output of subsequent layer to prevent vanishing gradient.

Recurrent Neural Network A recurrent neural network (RNN) is normally used to solve sequence problems such as voice recognition, language translation, and time-series sequences. Hochreiter et al. developed a popular RNN model called Long Short-Term Memory (LSTM) network in 1997 [26]. It can mitigate the vanishing gradient problem by introducing a self-loop. Since RNNs can model the relationship of time-series sequences, LSTM and other RNNs have influenced the development of deep learning networks that extract the temporal relationship of images. Alom et al. introduced a recurrent convolutional neural network based on U-Net (RU-Net), as shown in Figure 2.5. This model improves feature representation for medical image segmentation tasks by recursive residual convolution layers [5]. Gao et al. proposed the fully convolution structured LSTM (FCSLSTM), a combination of LSTM and CNN, to study the progression of disease from brain MRI data [21]. This model relies on CNN and LSTM to extract spatial and temporal information, respectively. Bai et al. applied a similar network to track disease development in human aorta [6]. Their model consists of a series of U-Net and LSTM connected in a sequential manner.

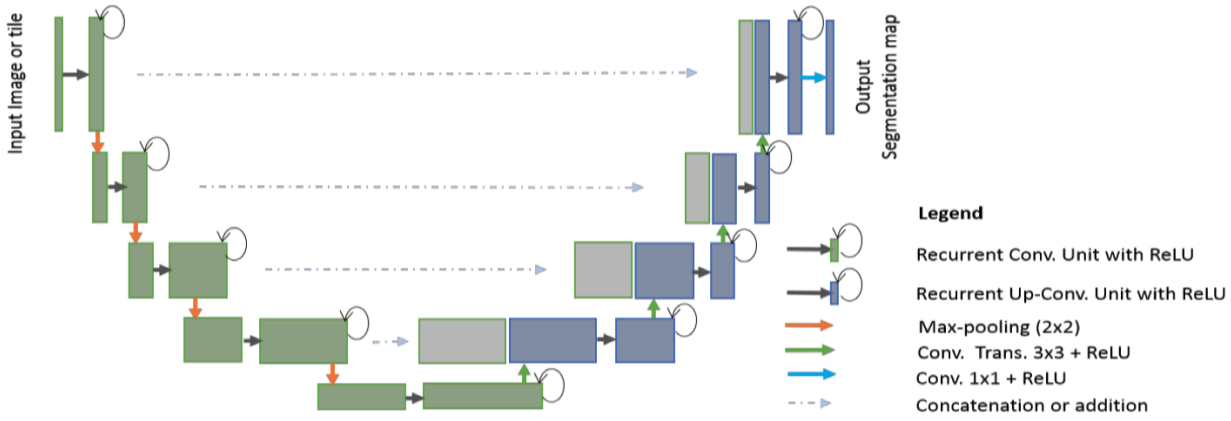

Figure 2.5: RU-Net architecture with convolution encoding and decoding units using recurrent convolution layers (RCL) based on U-Net architecture.[5].

Skip connection The U-Net's skip connections aim to improve feature representation by fusing the outputs of the encoder (low resolution) to the corresponding decoder layer (high resolution). A disadvantage of the skip connection is that the outputs of the encoder and decoder differ significantly in terms of resolution, causing the resultant feature maps to be blurry [38]. Ibtehaz et al. improved the original U-Net in multiple ways, including replacing the ordinary skip connection with convolution operations on the encoder's features before fusing with the decoder's features [30]. They tested the model on five different datasets, including EM images for cell segmentation and endoscopy images for colon cancer detection, and reported better performance than U-Net. Seo et al. [54] improved results for liver and liver-tumor segmentation by adding a residual path and activation operations to the skip connection.

Cascade Models The accuracy of image segmentation tasks could be improved by training multiple models and chaining their input and output together. This type of architecture has three main categories, coarse-fine segmentation, detection segmentation, and mixed segmentation [38]. We are interested mostly in the coarse-fine segmentation type. In coarse-fine segmentation networks, two different convolutional networks are linked together into one model. The first network provides a coarse segmentation of the region of interest (ROI), and the second network refines the coarse segmentation with details of the objects to be detected. Researchers have developed this type of network for liver and liver tumor segmentation using 3D CT images. Christ et al. proposed the Cascaded Fully Convolution Neural Networks (CFCNs) to segment tumors from the liver using CT images [14]. They employed an FCN to segment the liver, then fed the result to a second FCN for liver tumor segmentation. Other coarse-fine segmentation networks for liver and liver tumor segmentation are described in [58, 35, 17].

2.2.2 Network Block

The backbone networks discussed previously contain blocks that perform different computations on the input to generate the final output. Researchers have devised a variety of network blocks to effectively extract features maps within the larger network from medical images. This section will discuss two network blocks: dense connection and the attention mechanism.

Dense Connection In a dense connection network, the input of the previous layers is fed into subsequent layers. Guan et al. replaced the last block of every layer of the U-Net model with a dense block [22]. Figure 2.6 shows the diagram of this dense block. They named this architecture Fully Dense UNet (FD-UNet). The architecture outperformed U-Net for removing artifacts from reconstructed 2D photoacoustic tomography images. Zhou et al. [65] extended this idea to the larger model as shown in Figure 2.7 and named the new architecture UNet++. This network contains intermediate layers and dense skip connections to allow the feature maps to emerge gradually, bridging the semantic gap between the encoder and decoder. UNet++ outperforms U-Net in cell segmentation, brain tumor segmentation, liver segmentation, and lung nodule segmentation.

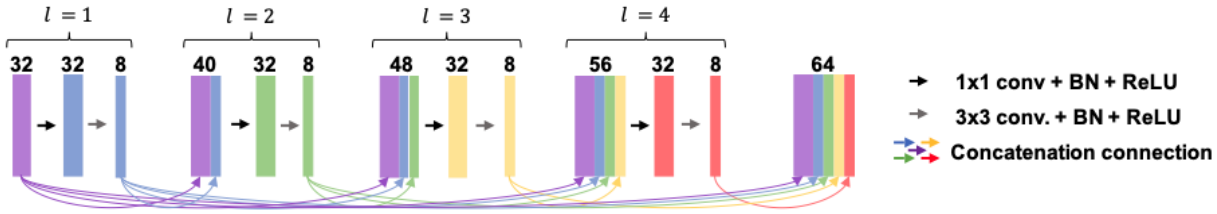

Figure 2.6: The dense block with four layers. The feature maps from preceding layers are concatenated as input to the following layers. BN stands for batch normalization [22].

Attention Mechanism Another network block that has been shown to improve a network's performance for medical image segmentation is the attention block. The basic of

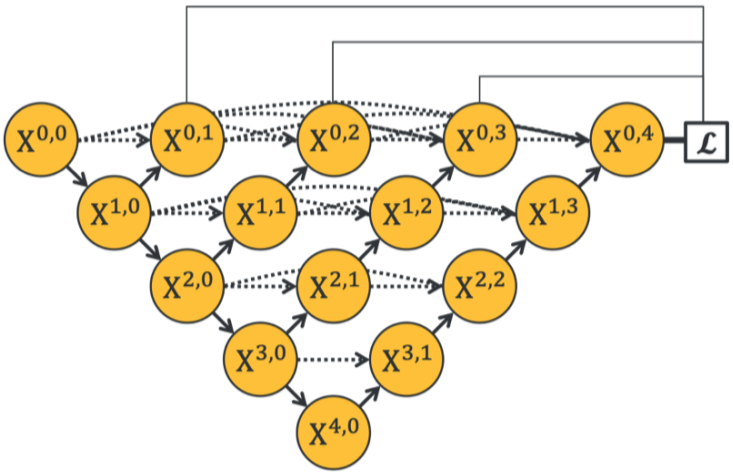

Figure 2.7: The U-Net++ architecture. Solid up and down arrows represent the encoding and decoding path, respectively, while dotted arrows represents skip connections. The yellow circles are convolution blocks [22].

the attention block is that it can assign different weights to the input features according to their importance. Oktay et al. incorporated the attention mechanism into the U-Net and proposed attention U-Net [49]. Figure 2.8 shows the overall architecture of attention U-Net. It is very similar to the original U-Net besides the addition of the attention gate, as shown in Figure 2.9. This network block modifies the output of the encoder before it is concatenated to the decoder features. The attention gate effectively controls the feature importance of input pixels.

Another type of attention mechanism is self-attention. The U-Net and many of its variations have two main limitations. The first one is low effectiveness: the encoder and decoder depend on local operators; the second is low efficiency: the models generate a large number of feature maps by increasing the output channels by a factor of two at each step [62]. The Non-local U-Net was proposed to solve these problems [62]. Non-local U-Net contains global aggregation blocks based on self-attention operators to gather global information without relying on a deep encoder with many parameters.

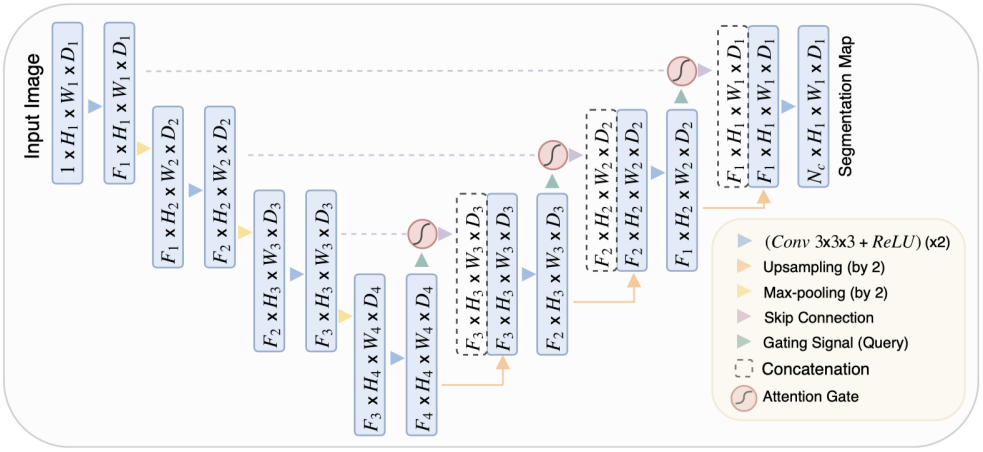

Figure 2.8: The architecture of the attention U-Net [49].

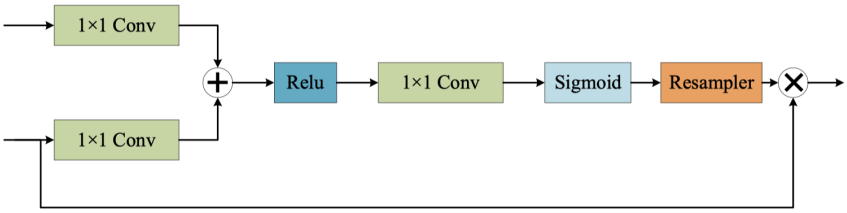

Figure 2.9: Attention gate from Attention U-Net[49], redrawn by Lei et al. [38].

2.3 Data Augmentation

The quality of the data dictates the segmentation results of deep learning models. However, in the medical field, it is challenging to build high-quality datasets since the cost of data acquisition and labeling is very high [38]. Researchers have to rely on data augmentation when large labeled datasets are not available at their disposal. Conditional generative adversarial nets (cGAN) are frequently used to generate additional data for various medical segmentation tasks. Guibas et al. proposed an architecture that contains a GAN and a cGAN to create synthetic images of retinal fundi images [23]. Another work by Mahapatra et al. overcame the challenge of limited chest X-ray images by using cGAN [41]. They employed a

cGAN and conditioned it on actual samples to synthesize realistic chest X-ray images.

2.4 Methods for Diagnosis of Intracranial Hemorrhage

Intracranial hemorrhage is commonly diagnosed using CT by a radiologist's trained eyes. In recent years, researchers have applied various deep learning methods to automatically detect intracranial hemorrhage from CT scans without the assistance of an expert. Hssayeni et al. collected CT scans from TBI patients and segmented the intracranial hemorrhage using a U-Net model [27]. They reported a dice coefficient score of 0.31. For a different dataset, Nemcek et al. proposed a weakly supervised deep learning algorithm that achieved a dice coefficient score of 0.58 [47].

Intracranial hemorrhage has five distinct subtypes, which are characterized by the anatomic location of the bleeding: intraparenchymal, subarachnoid, subdural, epidural, and intraventricular hemorrhage [18]. In addition to semantic segmentation, predicting a specific type of intracranial hemorrhage is another prevalent task. Nguyen et al. proposed a method that combined a CNN with an LSTM mechanism and made use of the whole 3D CT space [48] to identify the specific type of intracranial hemorrhage. The CNN acts as a feature extractor to process individual CT slices in a CT volume, while the LSTM accumulates and links the features across the different slices. Another work by Yassine et al. applied an ensemble of CNN and vision transformer to classify the type of intracranial hemorrhage in a CT scan [7].

Ultrasound is not a reliable tool to detect intracranial hemorrhage in an adult's head. It is more commonly used for very young infants. A trained expert can examine B-mode ultrasound images to diagnose neonates for intracranial hemorrhage. Johnson et al. [34] and Sauerbrei et al. [53] showed that B-mode ultrasound could be used to detect intracranial hemorrhage in neonates with high sensitivity and specificity. While in adults, ultrasound application is limited due to the poor insonation conditions [43]. B-mode ultrasound is efficient for adults when the scan is taken through the brain tissue with parts of the skull removed [59]. At the time of this writing, there is yet any existing work related to the automatic detection of intracranial hemorrhage in adults from ultrasound imaging.

2.5 Existing Work Related to Tissue Pulsatility Imaging Data

A previous student approached the problem of detecting intracranial hemorrhage using a different representation of tissue displacement data, whose description is not in the scope of this writing. The student attempted several different deep learning models, including MobileNetV2 [52], pix2pix [31], and ResNeSt [63]. The architectures of the mentioned models were used unmodified. The loss function used for model optimization was categorical loss entropy. The metrics used for evaluation are precision and recall. The best model was reported to be ResNeSt, with a reported precision score of about 0.07 and recall of about 0.2 after 70 training epochs.

The evaluation results showed that the previous deep learning methods and data preprocessing might not be suitable for detecting intracranial hemorrhage from displacement data. In addition, the choice of the loss function and evaluation metrics is poor. Categorical loss entropy considers the predicted results of all pixel an image equally. The result is that the loss value could be very low, but the predictive power for a class of interest is very low. The precision and recall scores evaluate the quantity of true positive, true negative, false positive, and false negative in relation to each other, but they do not indicate how well a model learns the correct label of a pixel or how well the pixels of the ground truth overlaps with the pixels in the detected areas.

Chapter 3

METHOD

This chapter discusses the detailed implementation of the proposed system. In deep learning, the quality of input data to a model dictates output quality. Therefore, it is essential to understand the data before building any model. This chapter goes into the details of the procedures that generate the patient data that were used for model development. Data generation in this context involves the collection, signal processing, and mask (label) generation. In addition, the chapter also describes the detailed implementation of the deep learning models employed to detect the skull, ventricles, and intracranial hemorrhage in a patient's head.

3.1 Data Collection

The data used in this research were collected from actual TBI patients. Therefore, the data collection is controlled by the policies of the University of Washington's Human Subjects Division. The research team filed a Zipline application to the Human Subjects Divisions' e-IRB system of the University of Washington. The application was approved by the Institutional Review Board (IRB) before any patient data were gathered.

The data were collected from patients admitted to Harborview Medical Center (HMC) in Seattle, the only Level 1 trauma center in the area capable of providing total care for every aspect of traumatic injuries to the brain [2]. The criteria for selecting patients are:

- Moderate to severe cTBI at HMC's Neuro ICU, with or without poly-trauma
- Patients 18 years or older
- Not prisoners

- Not from Native American or non-U.S. indigenous populations.

Obtaining data from prisoners, Native Americans, or indigenous population requires additional authorization from the government and tribal chiefs. Therefore, the last two criteria simplify the application for data collection. When TBI patients arrived at HMC, the patients received a life-saving treatment if necessary, including diagnosis of TBI via CT imaging. A research coordinator then screened them for the inclusion criteria. If they had an injury the team was interested in, the coordinator asked them or their family for consent. Ultrasound raw data were collected using a Terason (Burlington, MA) u3200t, a tablet-based, general-purpose scanner with a 4V-2 phased array transducer (64 elements, 2.5 MHz RF sampling frequency, 128 scanlines per frame). The scan rate was configured to 30 frames per second. A certified medical sonographer used the Terason device in the hospital room to collect ultrasound data. Data could come from patients hours after an injury or days after an injury. Data collection happened after the patients were diagnosed using CT or MRI and did not interfere with or interrupt routine hospital clinical care.

The sonographer collected data through the temporal window of the head (See Figure 3.1) without shaving the head, spending approximately five minutes aiming the ultrasound toward the known intracranial hemorrhage with each of the following orientations: coronal, axial, and oblique. The axial is a horizontal slice of the brain at zero degree; the coronal is a 90-degree slice of the brain; the oblique is a slice at any angle from one degree to 179 degrees, excluding the coronal. Data were collected from the left and right sides of the head. The sonographer also used a pulse oximeter to capture phases of the cardiac cycle. Any patient identifying information, including name, age, and gender, was removed from the data. Three-digit numbers distinguish one patient from another.

Before restrictions were imposed due to COVID-19 pandemic, the sonographer collected 15 scans from each side of the head equally divided between the three scan planes. The amount of data collected following COVID-19 restriction was limited to one axial, one coronal, and four oblique scans from each side of the head.

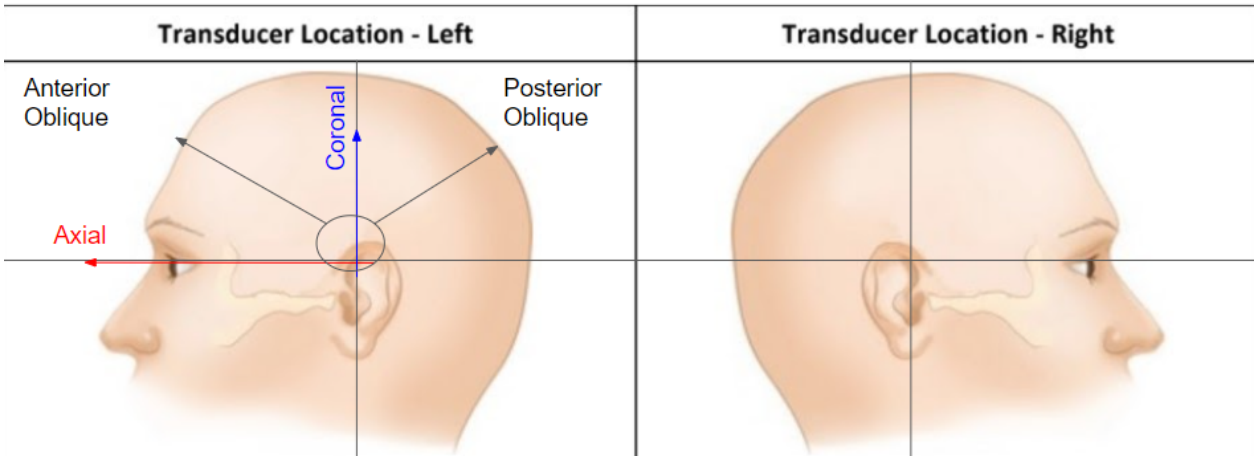

Figure 3.1: The planes at which ultrasound data are collected: left or right side of the head and at axial, coronal, or oblique.

3.2 Signal Processing

This section describes the signal processing algorithm, which was developed by an ultrasound signal processing expert Dr. John Kucewicz, employed to generate the displacement data. Medical diagnostic ultrasound transmits and receives short bursts of high-frequency sound, typically between 1 and 10 MHz. As sound propagates through the body, some fraction of the sound is scattered back towards the ultrasound transducer due to differences in local acoustic impedance. Because the speed of sound in tissue is relatively constant, two-dimensional images of tissue structure can be created based on the amplitude of the received ultrasound and the time between a sound burst's transmission and its reception. Images of blood or tissue velocity can be created by transmitting two or more ultrasound bursts and measuring the spatially localized relative temporal shifts in the received ultrasound from burst to burst. In ultrasound parlance, images of structure are referred to as B-Mode, i.e., brightness mode. Images of velocity are referred to as Doppler ultrasound or tissue Doppler ultrasound if tissue velocity is being measured.

Tissue Pulsatility Imaging (TPI) is a variation of tissue Doppler designed to measure

the naturally occurring pulsatile motion of tissue due to blood flow [37, 36]. During systole, blood accumulates in the arterial vascular, causing the tissue to expand by a fraction of a percent. Later in the cardiac cycle, accumulated blood flows through the capillary bed into the venous vasculature and back towards the heart, allowing the tissue to relax to its pre-systolic blood volume. TPI was inspired by plethysmography, an older technology used to measure gross changes in tissue blood volume in, for example, an entire limb. TPI extends this idea to measure local changes in tissue blood volume within the body.

For the larger study led by Dr. Pierre Mourad, it is hypothesized that TPI can be used to detect variations in tissue pulsatility localized to areas of the brain experiencing hemorrhage. The measurement of the TPI signal is based on standard ultrasound signal processing methods [16]. The ultrasound data was filtered into eight frequency sub-bands, quadrature demodulated, lag-1 autocorrelation was used to compute the velocity for each sub-band, and the velocities were averaged weighted by the power of the sub-band yielding an eight-second velocity waveform for every pixel in the image sequence.

Following the velocity calculation, an unpublished vector Doppler method was used to suppress common-mode motion introduced by the gross motion of the ultrasound transducer relative to the patient. Tissue displacement was calculated from the time integral of velocity combined with a band-pass filter to emphasize cardiac pulsatility. Lastly, individual cardiac cycles within the 8 seconds were detected and resampled such that all cycles were of uniform duration, i.e., 30 samples per cardiac cycle. Figure 3.2 summarizes the signal processing steps.

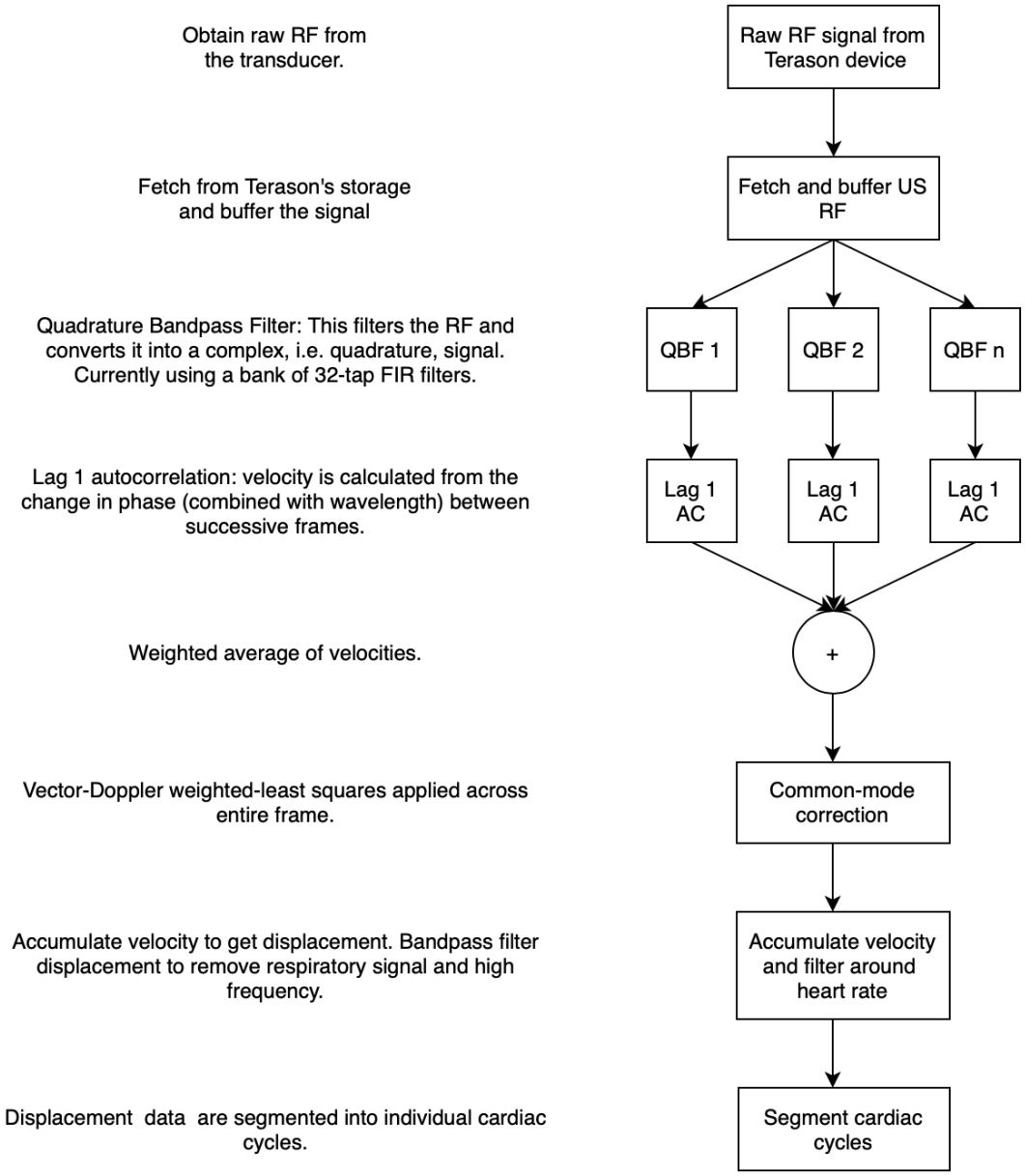

Figure 3.2: The raw ultrasound data were processed using various signal processing techniques to produce displacement data. The data were also synchronized with the heart's cardiac cycle.

3.3 CT Registration to Create Masks

Patients recruited for this research work were diagnosed with TBI using CT images of their heads. Our research coordinator, Nina LaPiana, obtained the CT images and aligned them with the ultrasound planes to create the brain masks, skull masks, ventricle masks, and blood (intracranial hemorrhage) masks for each ultrasound plane. This process is called registration. The procedure relies on an interactive MATLAB user interface. Figure 3.3 demonstrates an example of this procedure, and Figure 3.4 depicts the output brain and skull mask.

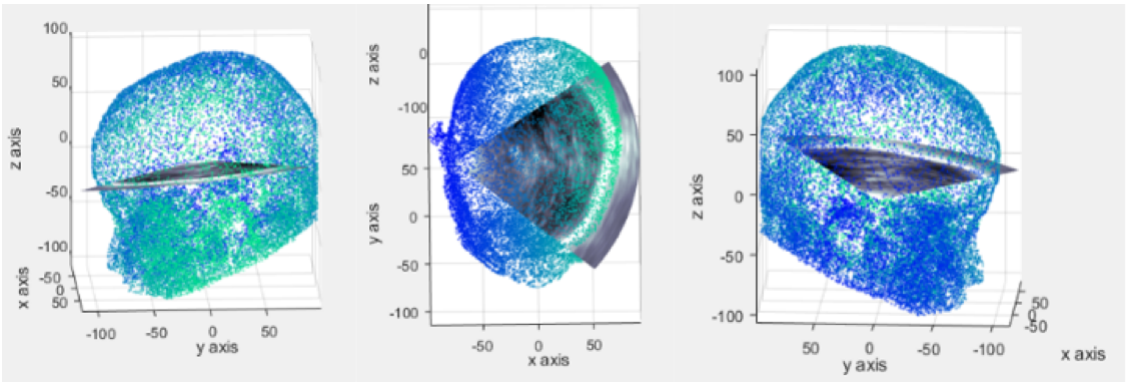

Figure 3.3: Registering a 2D ultrasound scan plane to the 3D CT data. The 2D B-mode scan is shown relative to a point cloud representing the skin surface of the head in the 3D CT data.

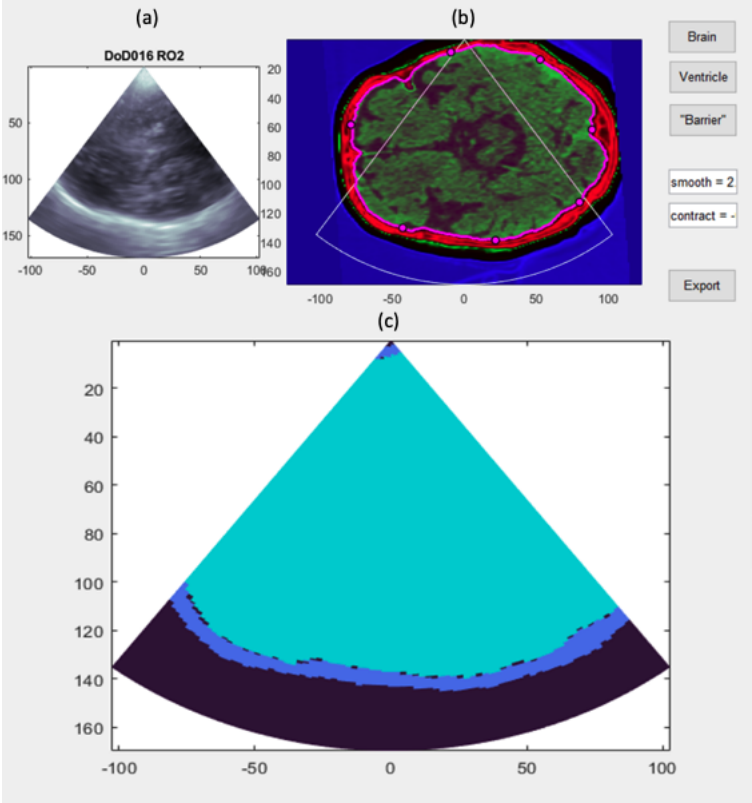

Figure 3.4: CT brain mask constructed by registration with B-Mode ultrasound for patient # 16 at the right oblique plane. (a) B-mode ultrasound data. (b) An overlay of the ultrasound plane on the CT image. (c) Resulting brain mask (cyan) and skull mask (blue).

3.4 System Design

This research is a part of a more extensive study that aims to deploy a portable ultrasound system to the battlefield to support field medics/doctors diagnosing TBI patients. Figure 3.5 depicts the use case of the Terason ultrasound system, which contains the deep learning models and the actors (the patient and the operator) who interact with it. The deep learning models are hidden from this view since the operator and patient do not use the algorithm directly. Instead, they benefit from the output of the algorithm displayed on the screen of the Terason system. The operator could be a nurse, a field medic, a doctor, or anyone who

can operate the Terason to collect sonograms. This person uses the included transducer to collect data from the patient and observes the output from the Terason system's screen.

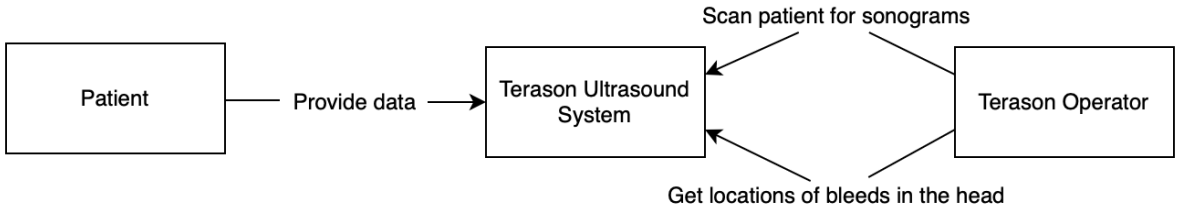

Figure 3.5: Context diagram show the use case of the system.

Figure 3.6 shows the general flow of data from the patient to the display of the Terason system. An operator uses a Terason compatible transducer to collect sonograms from the patient. This data is fed into the signal processing algorithm developed by Dr. John Kucewicz. This algorithm is written in MATLAB and interfaces with the Terason system's low-level processes via the SDK provided by the manufacturer. The resulting data is then prepared and fed into trained deep-learning models for the detection of structures of interest.

This research focuses primarily on the feasibility study of using deep learning models to provide the diagnosis. The deployment of the models onto the Terason system is out of scope. The development process follows the standard machine learning workflow(see Figure 3.7, where model development consists of data pre-processing, model selection and tuning, model training, and evaluation. The work has two phases: model development and inference. The majority of the research concentrates on model development, in which the data pre-processing steps for displacement data and masks are conceived, and three different models are developed to detect the skull, ventricle, and intracranial hemorrhage (blood). The inference phase represents the deployment of the models but without the Terason system. The user can select one of the three outputs (skull, ventricle, or blood) to display.

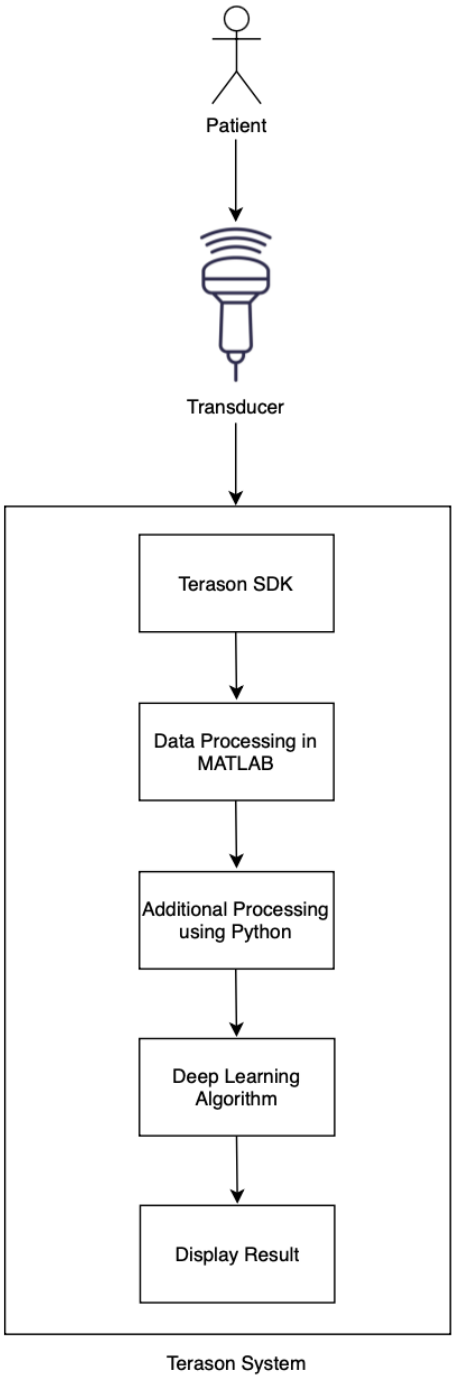

Figure 3.6: General data flow diagram of the system.

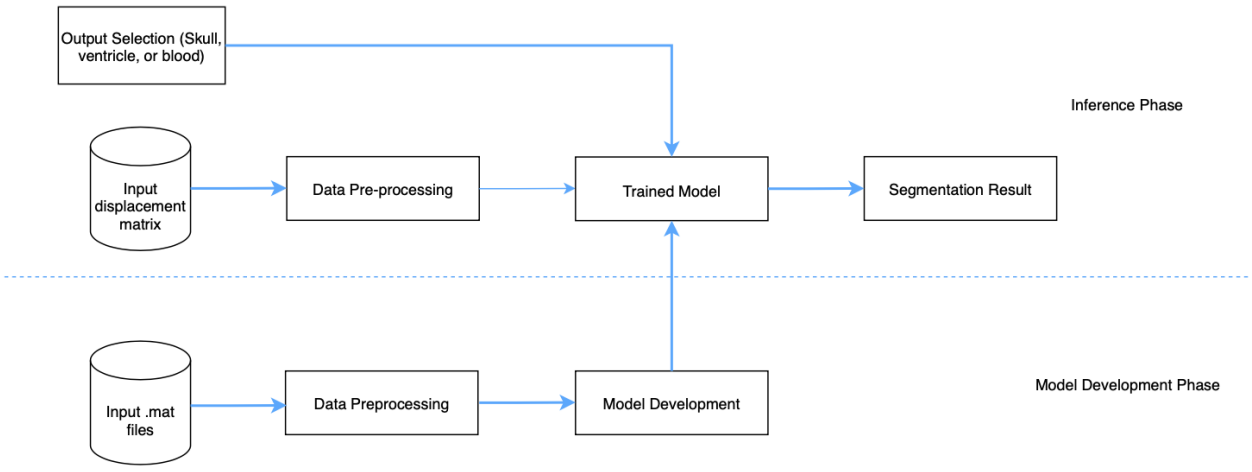

Figure 3.7: An overview of the development process.

3.5 Data Description

The number of total participating TBI patients is 104 as of October 2021. Data from each patient are ranked one to five according to the quality of the scan, with one being the lowest quality and five being the highest quality. A set of patient data with a quality rating of three is deemed suitable for use since they contain fewer artifacts from movement during data collection. The resulting number of patients whose data are included in this work is 96.

Every patient enrolled in the study received a subject ID (starting with 'DoD001'). Personal information, such as name, date of birth, gender, etc., is omitted. Prior to COVID-19 pandemic, each subject was scanned 30 times, where a scan is eight seconds of ultrasound data. The data collection protocol called for 15 scans from the left side of the head and 15 scans from the right side with five axial scans, five coronal scans, and five oblique scans from each side. After COVID-19 pandemic took place, participating patients received fewer scans due to pandemic restrictions. Thus, a patient could have a maximum of 30 data files, which correspond to 30 scans.

The data are available as .mat files and stored in a shared Google Drive, from which team

members can freely download. Files are named based on the subject ID, the overall scan number, the side (left or right), and the orientation of the transducer. For instance, a file prepended with 'DoD005_Ter007_RO3' refers to the third, right-side oblique scan (seventh scan overall) for the fifth subject enrolled in the study. Each file contains the following variables.

- xAxis: 2D matrix with the lateral coordinate for each pixel in the unit of millimeter. The sizes of this matrix can vary from patient to patient and scan to scan.

- zAxis: 2D matrix with the depth coordinate for each pixel in the unit of millimeter. he sizes of this matrix can vary from patient to patient and scan to scan.

- timeAxis: Vector with timestamp in seconds for each ultrasound frames.

- bMode: 3D matrix of "Brightness" of the received ultrasound data used for creating images of tissue structure. The sizes of the first two dimensions match zAxis and zAxis. The third dimension corresponds to the frame number. Its size is one less than the length of the timeAxis vector. For visualization, this should be compressed using something like log-compression to visualize both bone and soft-tissue.

- displacement: 3D matrix of displacement in the unit of millimeter. The sizes of the first two dimensions match xAxis and zAxis. The third dimension corresponds to the frame number. Its size is equal to the length of the timeAxis vector. This data is the common-mode motion corrected and high-pass filtered.

- bloodMask: Mask of intracranial hemorrhage in the brain based on CT data registered to the ultrasound imaging plan. A value of 1 indicates blood in a pixel.

- bloodMaskThick: Probability mask of blood in the brain based on CT registered to the ultrasound imaging plan along with neighboring planes. Values range from 0.0 to 1.0 with higher numbers indicating greater likelihood of blood being presented in a pixel.

- skullMask: Mask of the skull based on CT data registered to the ultrasound imaging plane. A value of 1 indicates skull in a pixel.

- skullMaskThick: Probability mask of skull based on CT registered to the ultrasound imaging plane along with neighboring planes. Values range from 0.0 to 1.0 with higher numbers indicating greater likelihood of skull being presented in a pixel.

- ventMask: Mask of the ventricles based on CT data registered to the ultrasound imaging plane. A value of 1 indicates ventricle in a pixel.

- ventMaskThick: Probability mask of ventricles based on CT registered to the ultrasound imaging plane along with neighboring planes. Values range from 0.0 to 1.0 with higher numbers indicating greater likelihood of skull being presented in a pixel.

- brainMask: Mask of the brain based on CT data registered to the ultrasound imaging plane. A value of 1 indicates brain in a pixel. This does not differentiate between pixels with and without blood.

- transPos: Position (x, y) and orientation (roll, pitch, yaw) of the transducer in CT space.

- dataCT: CT slice estimated to correspond to the ultrasound data. This is in Hounsfield unites with an offset of 1024.

- hr: Estimated heart rate during the scan in beats per second.

- hrTimes: Estimated time of the beginning/ending time for each cardiac cycle during the scan. These time were estimated from interpolated data and so are likely to occur between frame timestamps stored in the timeAxis vector.

- displacementNorm: Displacement data broken up into individual cardiac cycle and stretched or shrunk to be an equivalent number of the samples. The first two dimensions match xAxis and zAxis. The third dimension is time within a cardiac cycle. The fourth dimension is the cardiac cycle.

- bModeNorm: Similar to displacementNorm with one bMode frame for each cardiac cycle.

From each data file, the relevant variables are xAxis, zAxis, bModeNorm, displacementNorm, bloodMaskThick, skullMaskThick, ventMaskThick, and brainMask. The matrices xAxis and yAxis are used for visualization purposes. They allow an ultrasound image to be displayed within a cone-shaped area. bModeNorm matrix is also not an input to the development of deep learning models; it is solely used to show the corresponding B-Mode image of a displacement frame. The matrices bloodMaskThick, skullMaskThick, ventMaskThick, and brainMask serve as the ground truths for detecting intracranial hemorrhage, skull, ventricle, and brain tissue, respectively. The matrix displacementNorm contains the input displacement data (images) that are used to train and evaluate deep learning models. It is the output of the signal processing steps described in Section 3.2.

3.6 Data preparation and Processing

Preprocessing was carried out in a frame-wise fashion. The displacement data is a time series of the degree of movement of brain tissue over eight seconds. Even though there are changes from one frame of displacement to another, these changes are on a sub-millimeter scale. Extracting multiple displacement frames from a single ultrasound plane would result in similar images with the same label (mask). The redundancy would result in over-fitted models. Therefore, only a single frame is selected from the entire displacement time series. The eighth frame of the first cardiac cycle is extracted to maximize the contrast between normal pulsatility due to tissue perfusion and static hemorrhage or other brain structures (skull and ventricles). This frame correlates to the point where the blood volume (pressure)

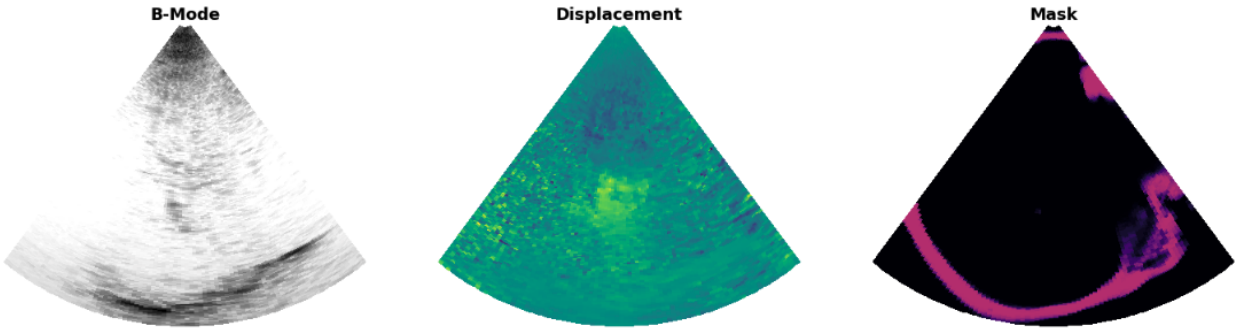

Figure 3.8: Sample data for detecting the skull. B-mode image is only for visualization of the normal ultrasound image. Displacement is a the processed displacement frame. Mask is the resized skull mask.

is the greatest in the brain. Each displacement frame is then resized to 256x80 pixels. Since the range of values in a displacement frame is large, the displacement values in a frame are standardized by shifting the mean to zero and scaling the standard deviation to one. This scaling would allow a model to generalize more effectively. The skull, brain, ventricle, and blood masks are also resized to 256x80 pixels. Figure 3.8, 3.9, 3.10, 3.11 show examples of the input data post-processing. The cone shape is for visualization only; inputs are rectangular images of size 256x80 pixels.

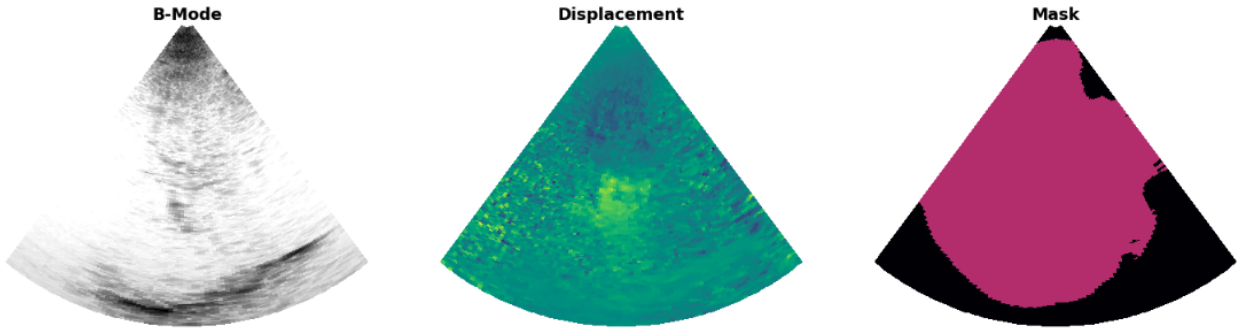

Figure 3.9: Sample data for detecting the brain. Brain detection is not a main task but is an intermediate step for detecting the ventricles and blood). B-mode image is only for visualization of the normal ultrasound image. Displacement is a the processed displacement frame. Mask is the resized brain mask.

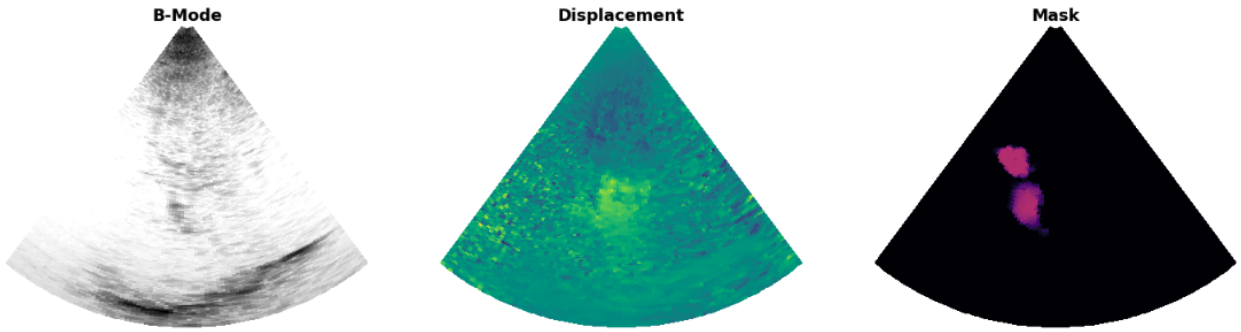

Figure 3.10: Sample data for detecting the ventricles. B-mode image is only for visualization of the normal ultrasound image. Displacement is a the processed displacement frame. Mask is the resized ventricle mask.

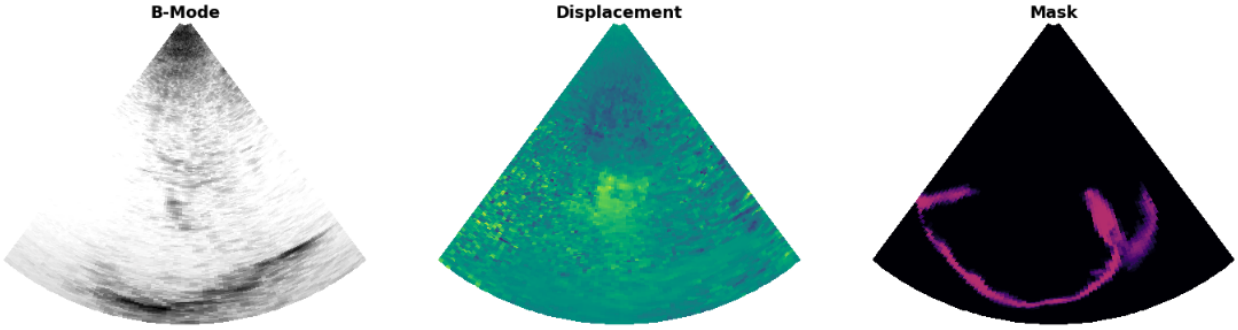

Figure 3.11: Sample data for detecting intracranial hemorrhage (blood). B-mode image is only for visualization of the normal ultrasound image. Displacement is a the processed displacement frame. Mask is the resized blood mask.

3.7 Loss Function Selection

The loss function of a deep learning model is the objective that the training process attempts to minimize. A loss function compares the model's output to the ground truth directly or indirectly, depending on its empirical formula. The loss functions used in this work are dice loss, balanced cross entropy, and a hybrid loss function consisting of dice loss and balanced binary cross entropy.

The skull, ventricles, and blood occupy a small percentage of the total pixels in a displacement frame, as shown in Figure 3.8, 3.10, and 3.11. This severe class imbalance is a challenge for a model to learn intricate features of the skull, ventricles, and blood. Dice loss function was selected since it was shown to perform well in binary-class segmentation tasks that have the class-imbalance problem [57]. The dice loss function was derived from dice coefficient equation, which is a widely used metric in the computer vision community to measure the similarity between two images [32]. The expression for dice loss is as follows:

$$DL(y, \hat{y}) = 1 - \frac{2\sum_{i=1}^{N} y_i \hat{y}_i + \epsilon}{N(\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i + \epsilon)}$$

Here, y is the ground truth; $\hat{y}$ is the predicted value by the prediction model; N is the

number of training examples; ϵ is a constant added to to nominator and denominator to avoid the edge case when $(y = \hat{y} = 0)$.

The standard dice loss shown above is modified with squared terms in the denominator. It was shown that the modified version allows the training process to reach the global minimum more effectively [44]. The modified dice loss is called soft dice loss, whose equation is as follow:

$$DL(y, \hat{y}) = 1 - \frac{2\sum_{i=1}^{N} y_i \hat{y}_i + \epsilon}{N(\sum_{i=1}^{N} y_i^2 + \sum_{i=1}^{N} \hat{y}_i^2 + \epsilon)}$$

Class imbalance is the issue during the training process. In contrast, output imbalance happens at inference. Output imbalance refers to the difference in the number of false positives and false negatives in the output. False negatives refer to the pixels that are erroneously labeled as background; false positives refer to the pixels that are erroneously labeled as the target object. For certain applications, it is preferable to eliminate more false positives than false negatives, or vice versa [57]. Balanced cross entropy [32] is, therefore, another loss function used in this work. It is a variant of the famous cross entropy function. Balanced cross entropy has a scaling factor to adjust the amount of false positives/negatives. The expression is as follows:

$$BCE(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} \beta(y_i ln \hat{y}_i) + (1 - \beta)[(1 - y_i) ln(1 - \hat{y}_i)]$$

Here, $\beta \in [0, 1]$ is the scaling coefficient for the false negatives and false positives. When β is set to a value less than 0.5, false positives are penalized more than false negatives. Balanced cross entropy was not used alone but in a combo function as discussed next.

To leverage the power of both dice loss and weighted binary cross entropy, we also used a combo loss function [57]. The equation of this loss function is as follows:

$$CL(y, \hat{y}) = \alpha \left(-\frac{1}{N}\sum_{i=1}^{N} \beta(y_i ln \hat{y}_i) + (1 - \beta)[(1 - y_i) ln(1 - \hat{y}_i)] \right)$$
$$+(1 - \alpha)\frac{1}{N}\sum_{i=1}^{N} \left(\frac{2y_i \hat{y}_i + \epsilon}{y_i^2 + \hat{y}_i^2 + \epsilon} \right)$$

Here, $\alpha \in [0, 1]$ is a weighting factor that determines the contribution of dice loss in the equation.

3.8 Skull Detection

Since the literature reported promising results for U-Net [51] in various types of medical segmentation tasks, the model of choice for this work is the U-Net. All of the models developed in this research have the U-Net as the backbone network. This section describes the model architectures that were used to detect the skull. Figure 3.12 illustrates the U-Net backbone that is the basis of all models we used. The results of this model were used as the benchmarks for other models. It has a contracting path (down-sampling) on the left and an expansive path (up-sampling) on the right. The contracting path primarily consists of two 3x3 zero-padded convolutions that are used four times. Each convolution has a rectified linear unit (ReLU) as the activation function, followed by a 2x2 max pooling operation with a zero stride for down-sampling. At a down-sampling step, the number of feature channels is doubled (see the number of filters of 2D convolution blocks in Figure 3.12).

Each step in the up-sampling path consists of a 3x3 zero-padded transposed convolution layer with a 2x2 stride followed by a ReLU activation function. The output of the transposed convolution layer is concatenated with the feature map from the contracting path, then goes through a dropout layer for regularization. Two 3x3 convolutions, each followed by a ReLU, operate on the resultant feature map. The final layer is a 1x1 zero-padded convolution with one output filter and a sigmoid activation. The number of output filters (layers of the output feature map n at the first convolution layer) dictates the following convolution layers' output filters. It is a hyperparameter that can be adjusted. The chapter 4 will discuss the selection of this number. Another difference from the original U-net is that this implementation has a dropout regularization layer after every max-pooling layer in the contracting path and after each up-convolution step in the expansive path. In this way, the dropout regularization is applied to each element within the resultant feature map of the previous layer. dropout regularization probabilistically removes or "drops out" input units with a frequency of a dropout rate at each step during training time. dropout regularization helps prevent overfitting. The dropout rate is another hyperparameter that could be adjusted

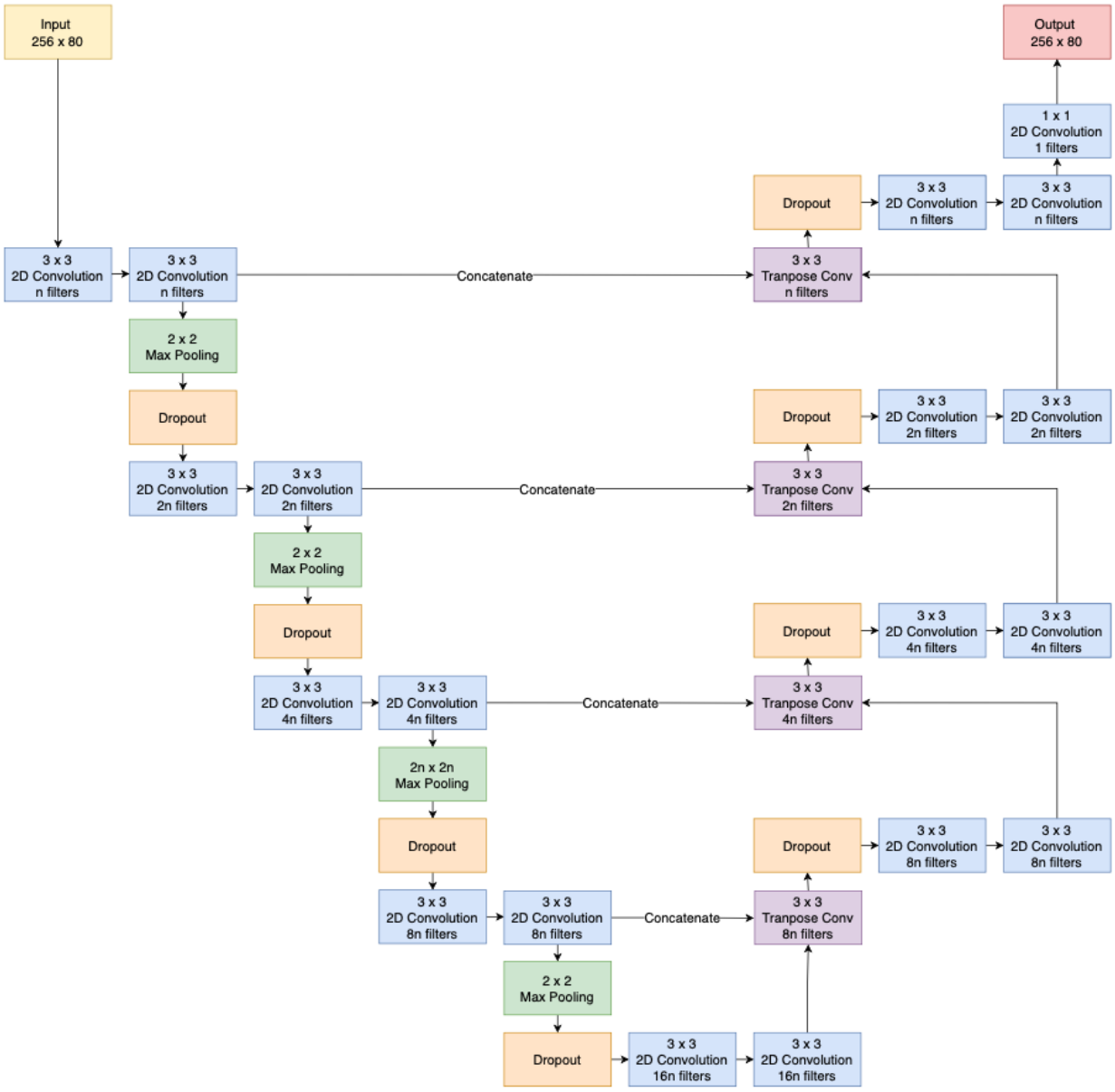

Figure 3.12: The U-Net implementation used in this research. 3 x 3 is the kernel size of the convolution layers. n is the number of output filters of the first convolution layer. n is doubled at each subsequent level.

to achieve the highest performance.

The second model used for this task is the U-Net++ [65]. U-Net++ was chosen for its ability to improve the gradient flow. Figure 3.13 illustrate the implementation of U-Net++ used in this research. The overall structure of the model is adapted from the original U-Net++ paper. Similar to U-Net, this model consists of a main down-sampling and up-sampling path, with the addition of redesigned skip pathways and dense skip connections. The redesigned skip pathways (shown in green) bridge the semantic gap between the contracting and expansive paths. In U-Net, skip connections directly connect the feature maps from the contracting path to the expansive path, potentially fussing semantically dissimilar feature maps. The convolution layers within the redesigned skip pathways allow feature maps to evolve gradually, reducing the semantic gap between the contracting path and the expansive path. As a result, the optimizer used during training can optimize for the loss function better.

The implementation of the down-sampling block (D-block) and up-sampling (U-block) within the architecture of U-Net++ is shown in Figure 3.14. The down-sampling block contains two 3x3 zero-padded convolution layers, each followed by a ReLU activation function, a 2x2 max-pooling layer with a zero stride, and a dropout layer, whose dropout rate was tuned. The up-sampling block contains a 3x3 zero-padded transpose convolution layer with a 2x2 stride followed by a ReLU, a dropout layer, and two convolution layers similar to those of the corresponding down-sampling block. The output of the convolution layers in the down-sampling block is concatenated with the output of the corresponding transpose convolution to create the re-designed skip pathways and dense skip connections. Like U-Net, the number of output filters n in each convolution layer was adjusted to maximize the performance while reducing the computational cost.

The authors of U-Net++ also introduced deep supervision so that the complexity of the model can be adjusted, trading inference speed for performance. This idea is especially attractive considering the ideal deployment scenario is on the non-GPU Terason system. However within the scope of this work, deep supervision was not explored.

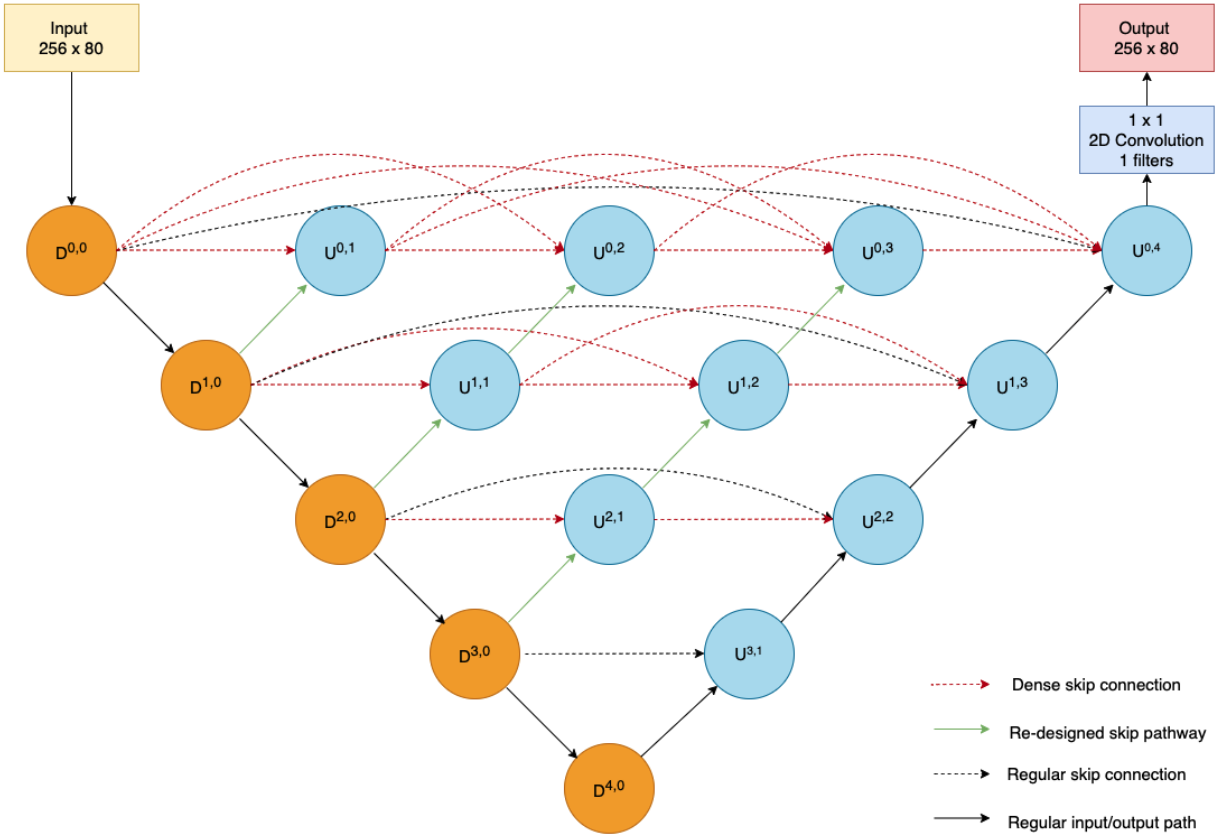

Figure 3.13: The implementation of U-Net++ used in this research. The orange D-blocks are similar operations on the contracting (down-sampling) path. The blue U-blocks are similar operations on the expansive (up-sampling) path. The solid arrow indicates that the output of a block is the input of another block. The dotted arrow indicates the intermediate pre-maxpooling output of a block is concatenate to the another block. The superscript of a block indicates the depth level. Refer to Figure 3.14 for the details of D-blocks and U-blocks.

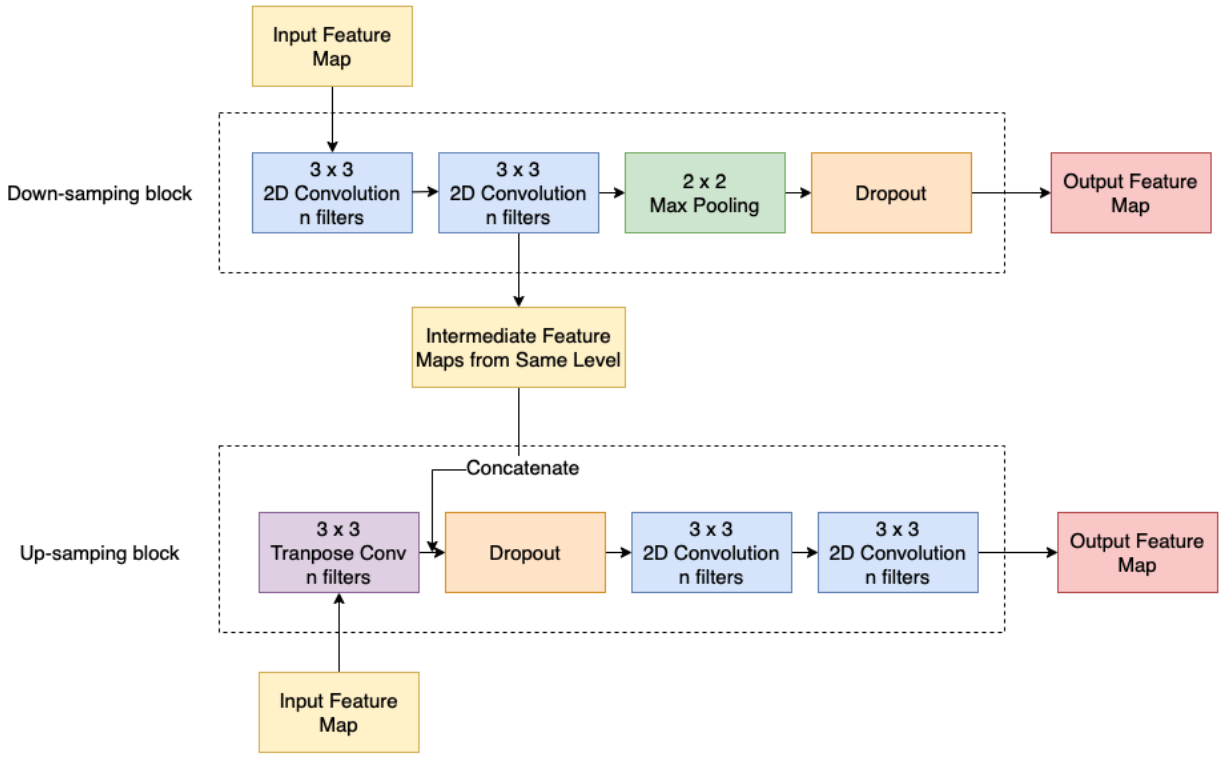

Figure 3.14: The up-sampling and down-sampling block of the U-Net++ implementation. The number of filters n in the convolution layer and transpose convolution layer corresponds to the deep of the level that it belongs to.

3.9 Ventricles and Intracranial hemorrhage detection

Since the mask, or label, for each segmentation task is independent of each other, the model architectures described in the previous section for skull detection could be re-purposed to detect the ventricles and intracranial hemorrhage (blood). The U-Net was again used as the benchmark for the detection of ventricles and intracranial hemorrhage. Both ventricles and blood occupy a small percentage of the total pixels in a displacement frame. Therefore, the ordinary U-Net and U-Net++ might not work well for ventricles and blood. We need a mechanism that allows a model to "zoom" into a region of interest to find the ventricles and blood. The region of interest (ROI) here is the brain tissue area, which contains the pixels corresponding to ventricles and blood. From the literature review, we identified two types of models that would allow the zooming operation: attention mechanism and cascade network. We used independent models that have similar architectures for the tasks of detecting the ventricles and detecting blood.

Attention mechanism, in the context of image segmentation, is a way to highlight only the relevant activations during training. This technique allows a network to focus on a certain part of an image. The attention mechanism was added to the U-Net to improve the skip connection. Our attention U-Net's implementation is shown in Figure 3.15. This implementation is largely similar with that described in Section 3.8, with the addition of the attention block, which is adapted from [49]. The attention block takes two input vectors, the output of the previous up-sampling step (or gating signal) and the output of the corresponding convolution in the down-sampling path (or input features). The input features go through a zero-padded 3x3 stride convolution so that their dimensions are the same as the gating signal after it undergoes a zero-padded 1x1 convolution. The two vectors are summed element-wise, resulting in aligned weights becoming larger while unaligned weights becoming relatively smaller. The resultant vector passes through a ReLU activation layer and a zero-padded 1x1 convolution layer that collapses the dimensions further. It then goes through a sigmoid layer to scale its values between the range 0 and 1, where 1 indicates more

relevant features; this is the attention coefficient. It is then upsampled back to the original input features' dimensions. The attention coefficients are multiplied element-wise with the input features to scale them according to relevance. The attention output is then passed on and concatenated as a skip connection in the U-Net.

An alternative to the attention mechanism is a cascade of networks. For ventricles and blood detection, such a cascade network consists of two separate models, a coarse model that identifies the brain tissue and a fine model that detects the ventricles or blood. The fine model uses the output of the coarse model as the region of interest (ROI) and then only looks for relevant features within the ROI. We will discuss two different implementations of cascade networks.

Zhang et al. [64] inspired the first cascade network, called Cascade A. Figure 3.17 illustrates the high-level view of this model. Net A and Net B are U-Nets with similar architecture as described previously. They are trained separately and are only fused at inference. Net A was trained using the input displacement data and the CT brain mask to detect the brain tissue area. Its job is to produce the brain mask that is used by Net B as the ROI. The output of Net A (dimension 256 x 8 x 1) is then concatenated with the one-channel displacement input frame. Net B processes the resultant two-channel combination to segment the ventricle or blood.

Eppel et al. [15] inspired the second cascade network, named Cascade B. Figure 3.18 illustrates the high-level view of this model. The implementations of Net A and Net B in this model are similar to those in Figure 3.17. The difference is in the way Net B processes the inputs. The output of Net A goes through an independent 2D convolution layer to down-sample it to the same dimensions as the input displacement frame after the first down-sampling block of Net B. The resultant vectors are multiplied element-wise and are passed through Net B.

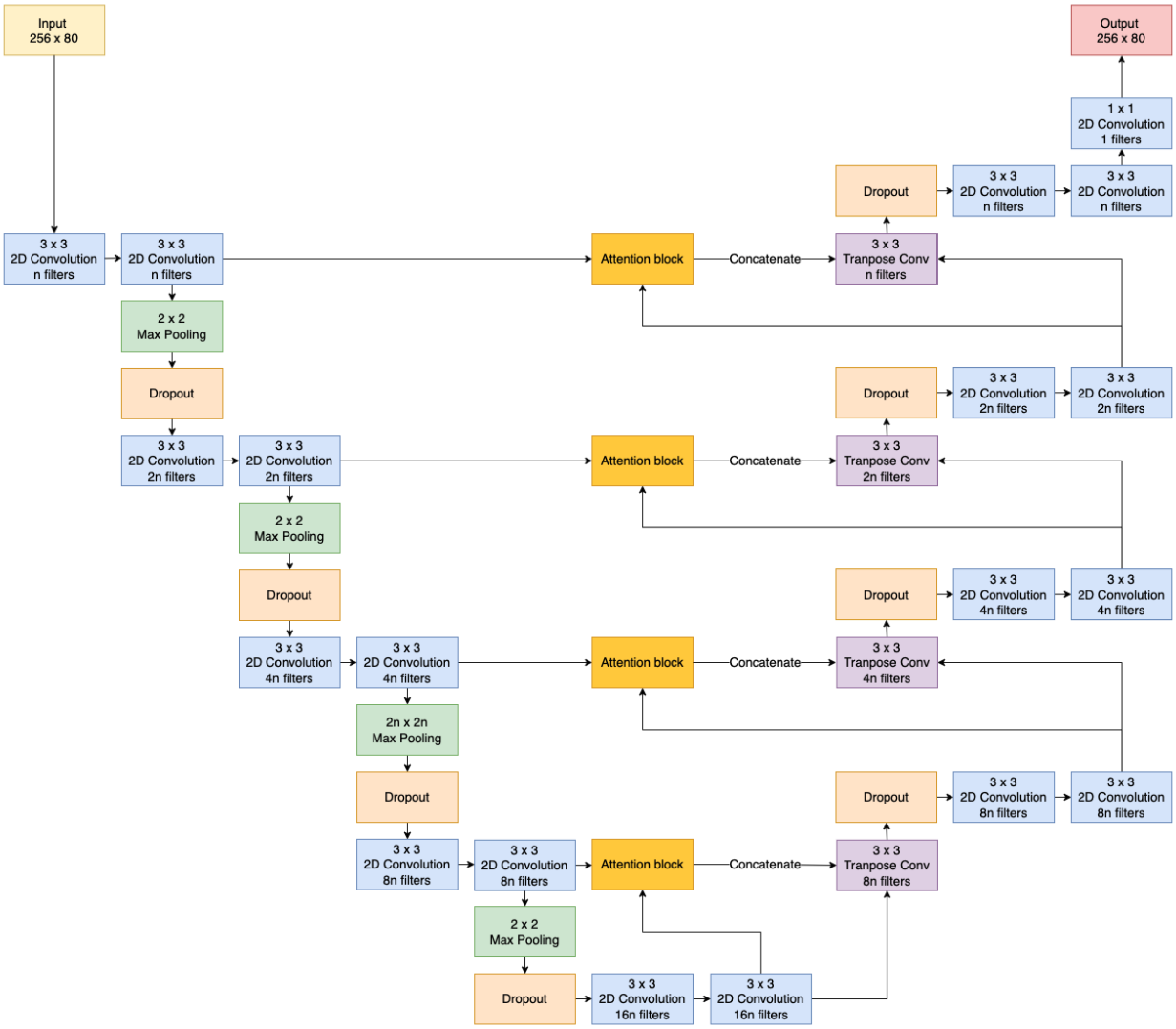

Figure 3.15: The implementation of Attention U-Net. The implementation of the U-Net is similar to that seen in Figure 3.12, with the addition of the attention block.

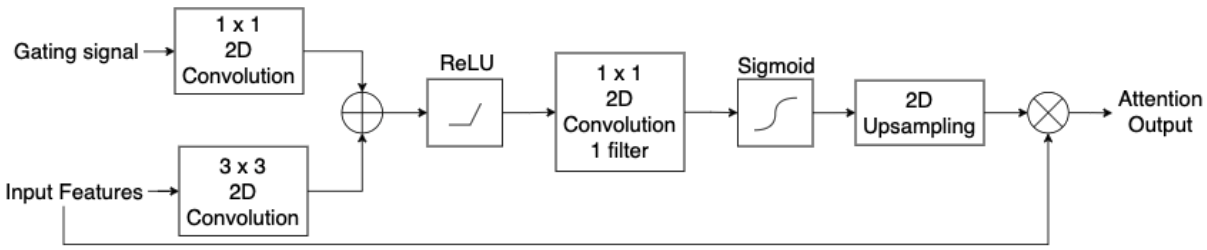

Figure 3.16: The implementation of the attention block is adapted from [49].

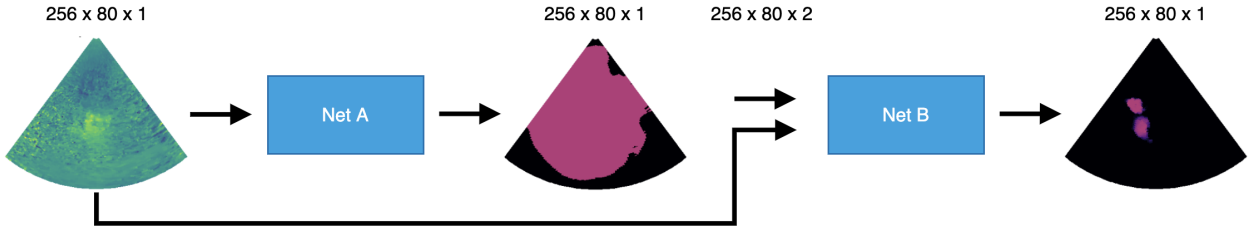

Figure 3.17: The overview of the implement of Cascade A model. Net A is trained to identify the brain tissue while Net B concatenate the input and Net A's resultant brain mask to detect ventricles or blood. The numbers above the diagram indicate the dimensions of the matrices.

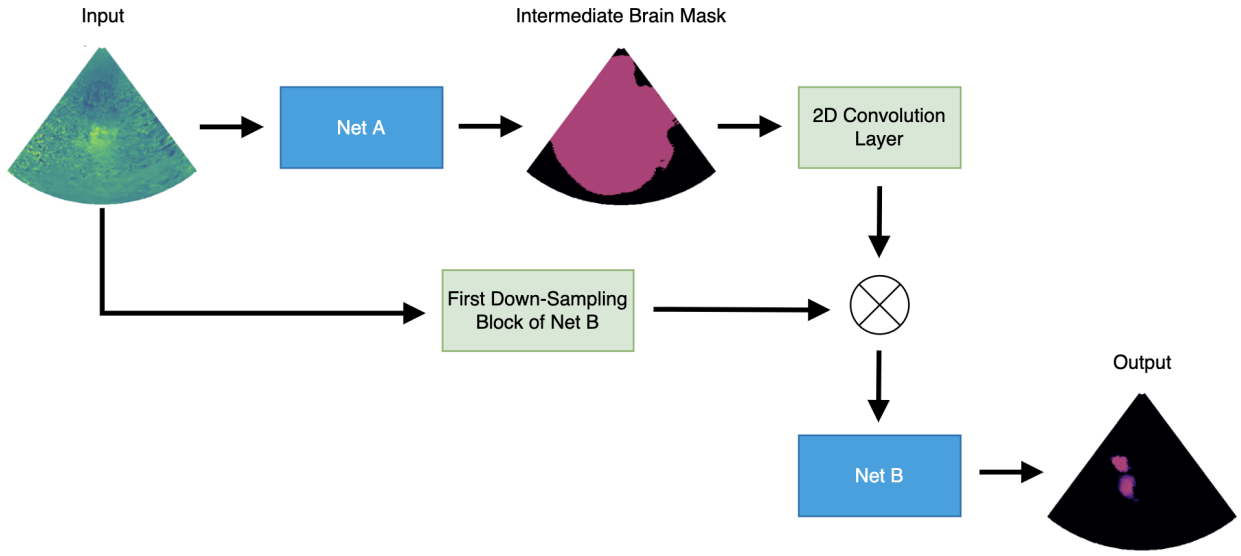

Figure 3.18: The overview of the implement of Cascade B model. The brain mask goes through a 2D convolution layer whose output is multiplied element-wise with the output of first down sampling block of Net B.

Chapter 4

EXPERIMENT AND RESULT

This chapter aims to evaluate the proposed methods for their efficacy in detecting the features in the head of TBI patients using TPI displacement data. The goal is to identify a model that provides the highest result, in terms of the evaluation metrics, for each detection task (skull, ventricles, and intracranial hemorrhage). This chapter describes the experiment setup, equipment, and results of the proposed models for detecting skull, ventricles, and intracranial hemorrhage (blood).

4.1 Experiment Set Up

The patients were shuffled and randomized into a two dataset:

- A training set used for training, hyperparameter tuning, and cross-validation. The training set contains 86 patients (1661 examples).

- A testing set used for gauging the models' performance on unseen data. The testing set includes ten patients (131 examples)

This data allocation method ensures complete independence between the training and testing sets, preventing data from the same patient from being in both. Within the training set, the data were again shuffled, regardless of which patient they belonged to. 20% of the data is further reserved as a hold-out validation set to gauge the performance of the models during their development.

The main computing resource is a high-performance computer system called Otachi provided by the University of Washington, Bothell. The following software and hardware were used to perform experiments:

- Programming language: Python 3.9.2
- Deep learning framework: Tensorflow 2.6.0
- CUDA version: 11.2
- Operating system: Linux Debian 4.19.181-1 (2021-03-19) x86
- CPU: Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz
- GPU: NVIDIA Tesla V100 Tensor Core

We employed the Adam optimizer with a learning rate of 0.001 to train the models. Adam is an optimization algorithm that is an alternative to the classical stochastic gradient descent to update a model's weights iterative using training data. The stochastic gradient descent is an optimization algorithm that estimates the error, between the ground truth (label or mask) and the output, for the current state of a model using examples from the training dataset. The resultant error is then used to update the model's weights using backpropagation. The learning rate provides a starting point for the learning process to converge at a minimum of the objective (loss) function. The value of 0.001 is selected because early experiments show that a larger learning rate arrives at poor results. In addition, the learning rate is scheduled to be reduced by a factor of ten when a plateau in model performance is detected. The intent is to slow down the training process and fine-tune the model weights when the optimization algorithm approaches a minimum of the objective function. The scheduling is implemented using the ReduceLROnPlateau callback of Tensorflow.

We evaluated the models using popular image segmentation metrics: dice coefficient score, intersection over union (IoU) score, precision, and recall. Precision and recall scores are calculated using Tensorflow 2.6.0's built-in precision and recall, while dice coefficient and IoU scores are implemented from scratch in Python. The values of these metrics range from 0 to 1, with 1 being the perfect score, meaning the model can segment the feature of interest with 100% accuracy.

4.2 Loss Function Selection

Since skull detection is the most achievable task out of three (skull, ventricles, blood) due to the highest number of relevant pixels and a greater contrast in input displacement images (solid bone is less likely to be displaced), the U-Net built for skull detection was used to evaluate the two loss functions mentioned in Chapter 3, soft dice loss and combo loss. A U-Net model was built and trained for 50 epochs with the following hyperparameters: the number of start filters n of 16, batch size of 32, and a dropout rate of 0.5. At the end of the 50 epochs, the training curves plateau, allowing the validation metrics shown in Table 4.1. We see that even though the value of the loss function is lower with the combo loss function, the resultant dice, IoU, precision, and recall scores are not necessarily better. The benefit of using the combo loss for the segmentation tasks in this research is not significant. Therefore, the dice loss function is the only loss function used to train deep learning models in the remaining sections of this chapter.

Table 4.1: Validation results of U-Net for skull detection using different loss functions after training for 50 epoch with a learning rate of 0.001. The validation was performed on the hold-out validation data

Loss Function	**Loss**	**Dice**	**IoU**	**Precision**	**Recall**
Soft Dice	0.2678	0.5021	0.3345	0.8526	0.4075
Combo with $\alpha = 0.3, \beta = 0.5$	0.2275	0.4907	0.3252	0.8456	0.4034
Combo with $\alpha = 0.5, \beta = 0.5$	0.1993	0.4848	0.3200	0.8531	0.3962
Combo with $\alpha = 0.5, \beta = 0.7$	0.1967	0.4913	0.3257	0.8231	0.4580
Combo with $\alpha = 0.5, \beta = 0.3$	0.1954	0.4832	0.3186	0.8679	0.3717
Combo with $\alpha = 0.7, \beta = 0.5$	0.1690	0.4871	0.3220	0.8584	0.3817

4.3 Skull Detection

The proposed models used for skull segmentation are the U-Net and U-Net++. They were trained and tested on the training data and hold-out set for comparison. The following hyperparameters were used: number of filters in the first convolution layer n of 16, dropout rate of 0.5, batch size of 32, and 100 training epochs. Figure 4.1 illustrates the value of the loss function (soft dice loss) over 100 epochs for U-Net and U-Net++. The lower the loss value, the better the model performs. Figure 4.2 shows the performance of these models in terms of dice coefficient score at the same epochs. The higher this value, the better the model performs. In the beginning, the loss value is high, and the dice score is low. As the training proceeds, the loss decreases while the dice score increases, indicating that the models gradually learn the feature representation from the examples in the training set. The curves eventually plateau at around 60 epochs. Using the above set of hyperparameters, U-Net and U-Net++ have similar performance, even though U-Net++ is a more complex model of the two.

Furthermore, for both models, the training curve shows better results than the validation curve, indicating a degree of overfitting occurs but not by a large margin. Overfitting means that a model learns feature representation well on what it has seen in training but fails to generalize on unseen examples. Because of overfitting and that both models perform similarly, the input displacement data might not contain enough information for the models to learn from.

We also performed a grid search for U-Net and U-Net++ to explore the results from using different combinations of hyperparameters (filters in the first convolution layer, dropout rate, and batch size). Grid search is a hyperparameter tuning technique that aims to select a set of hyperparameters that provide the highest performance. Since grid search is resource-intensive, and early experiments showed that U-Net and U-Net++'s learning curves plateau at around 60 epochs, grid search was performed for 60 epochs. Upon the completion of the grid search, we found the hyperparameter combinations shown in Table 4.2. The details of

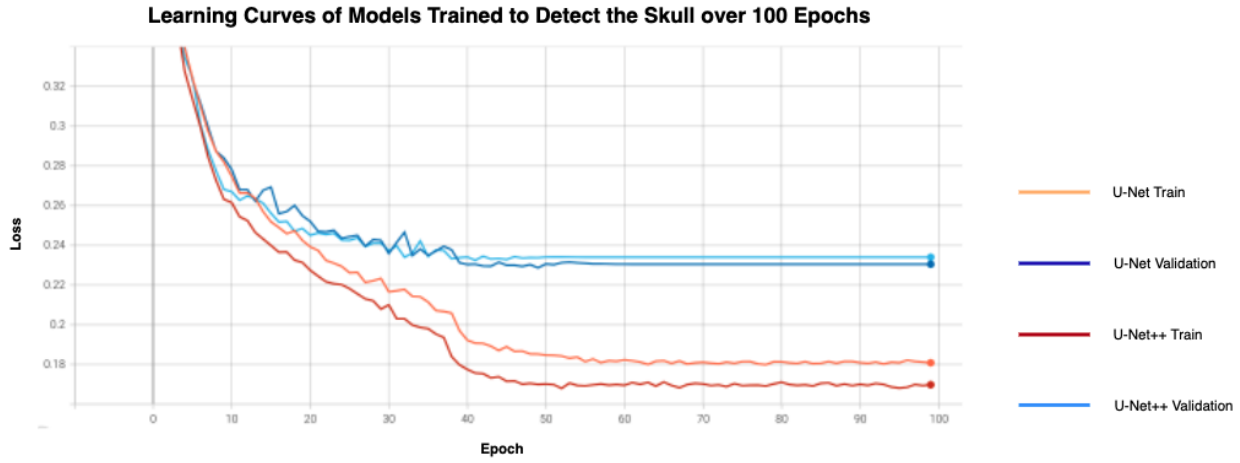

Figure 4.1: Loss curves of U-Net and U-Net++ trained for skull detection over 100 epochs.

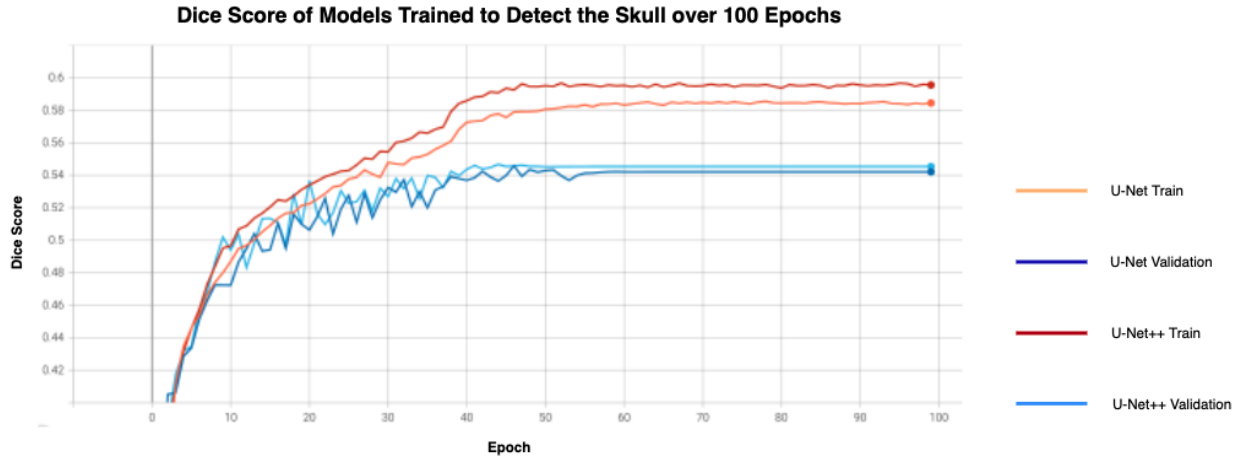

Figure 4.2: Dice score of U-Net and U-Net++ trained for skull detection over 100 epochs.

the grid search experiment could be found in Appendix A.

Since the dataset is relatively small, we also performed 10-fold cross validation to evaluate the models built with the parameters in Table 4.2. 10-fold cross validation was done by randomly splitting the training set into ten equal partitions, training on nine parts, and validating on one different part. The process was repeated ten times (ten folds). The results

Table 4.2: Results of grid search for skull detection.

Model	**n Start Filters**	**dropout Rate**	**Batch size**
U-Net	32	0.3	32
U-Net++	32	0.4	20

(mean and standard deviation) are summarized in Table 4.3. The standard deviations across all metrics are minimal, indicating that the results of different folds are very similar. In addition, we see that U-Net and U-Net++ arrive at very similar results. Therefore, the final model selection depends on which model can generalize better when tested on the unseen data.

For the last round of evaluation, we retrained the models on the entire training set and evaluated on the independent testing set. Table 4.4 shows the results when the trained models were tested on unseen data. We see that the performance is slightly lower than validation performance in Table 4.3, but it is expected for unseen data. U-Net performs slightly better in terms of dice, IoU, and precision scores. Since U-Net can generalize slightly better, it was deemed the final model for skull detection. Figure 4.3 illustrates a few output examples for skull detection using the trained U-Net model. In general, the model can discern the part of the skull that does not associate with the frontal side quite well from the background. It fails where there are fine details in the bone structure. The cause of this inaccuracy is probably due to insufficient information on the frontal side of the head. The bone structure in the facial region is more complex than other parts of the skull, preventing ultrasound from picking up more information.

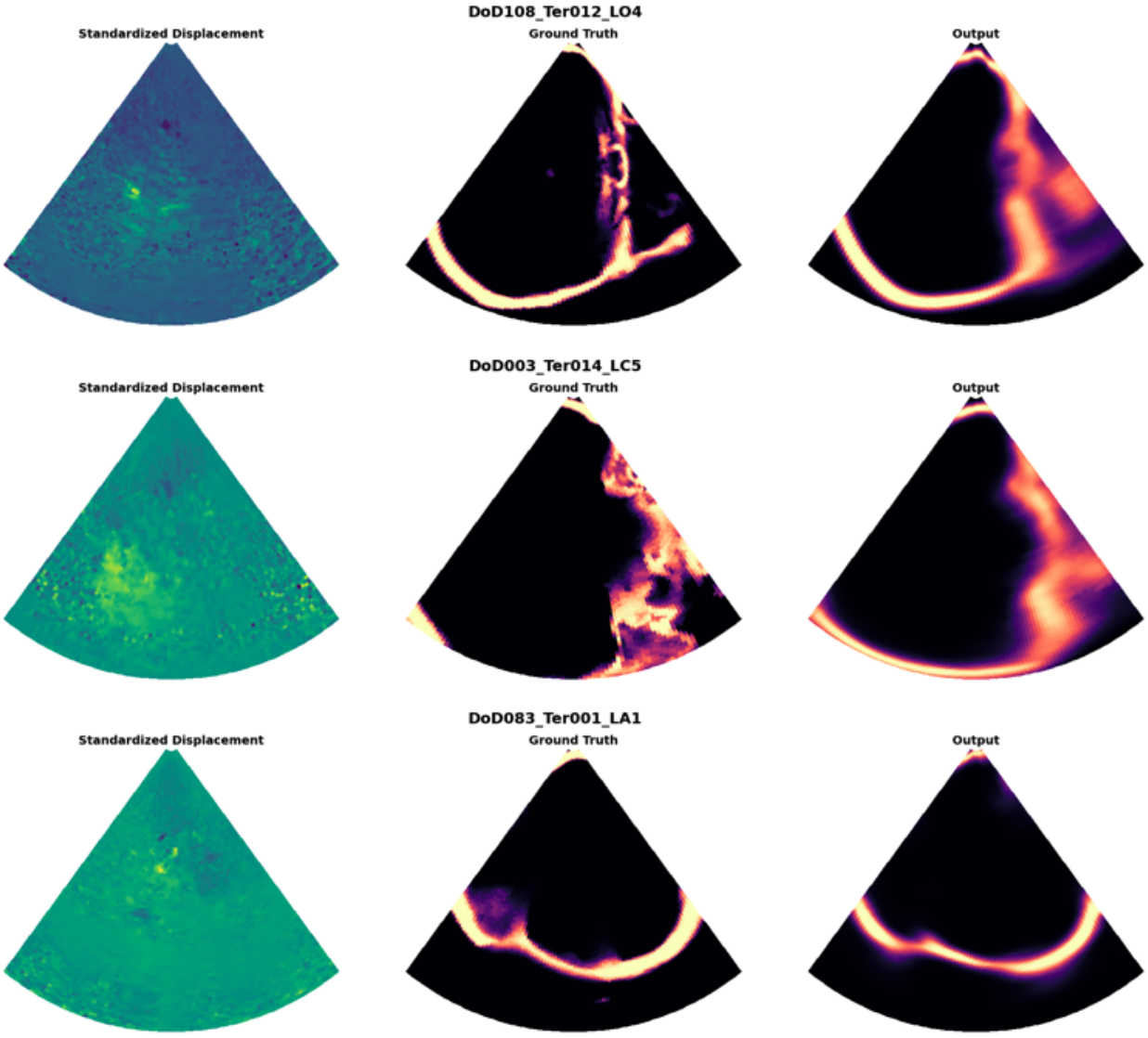

Figure 4.3: Example outputs for skull detection task using the final U-Net model. Standardized displacement is the post-processing input; ground truth is the CT skull mask; last column is the model's output.

Table 4.3: 10-Fold cross validation results of U-Net and U-Net++ for skull detection (mean and standard deviation).

Model	**Dice**	**IoU**	**Precision**	**Recall**
U-Net	0.5541 (0.0086)	0.3836 (0.0078)	0.8830 (0.0084)	0.4261 (0.0105)
U-Net++	0.5570 (0.0094)	0.3864 (0.0089)	0.8817 (0.0064)	0.4251 (0.0108)

Table 4.4: Skull detection test results from U-Net and U-Net++ on unseen data in the testing set.

Model	**Dice**	**IoU**	**Precision**	**Recall**
U-Net	0.5175	0.3491	0.8475	0.4071
U-Net++	0.5040	0.3371	0.8469	0.4106

4.4 Brain Tissue Detection

The detection of the brain tissue is not one of the main tasks in this research. However, it is required to use cascade models to detect the ventricles and intracranial hemorrhage since they are enclosed within the brain tissue. The brain mask, which represents the brain tissue, serves as the region of interest (ROI), allowing another model to focus only on the relevant area. Net A in Figure 3.17 and Figure 3.18 perform this task. Figure 4.4 and Figure 4.5 show the results of experiments for brain detection with the following hyperparameters: number of filters in the first convolution layer n of 16, a dropout rate of 0.5, a batch size of 32, and 100 training epochs. Here, U-Net++ performs better since its loss curves converge to lower values, and dice curves converge to higher values than U-Net.

We also employed grid search to find a combination of start filters n, dropout rate, and batch size that produces the highest result for U-Net and U-Net++ over 50 training epochs. Table 4.5 list the resultant hyperparameters. U-Net and U-Net++ were built with these hyperparameters and were evaluated using the 10-fold cross validation procedure. Table

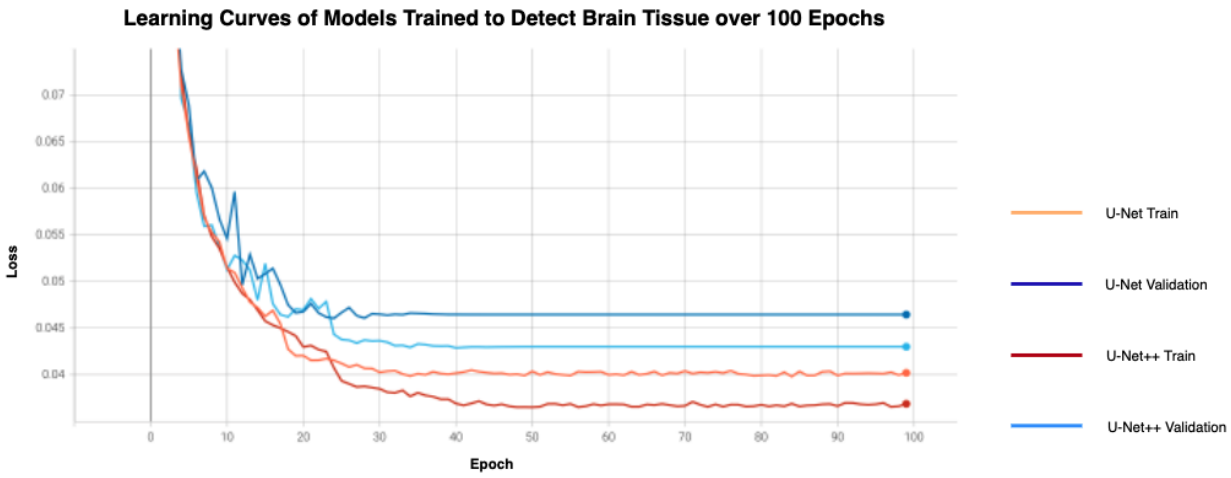

Figure 4.4: Loss curve for brain detection over 100 epochs.

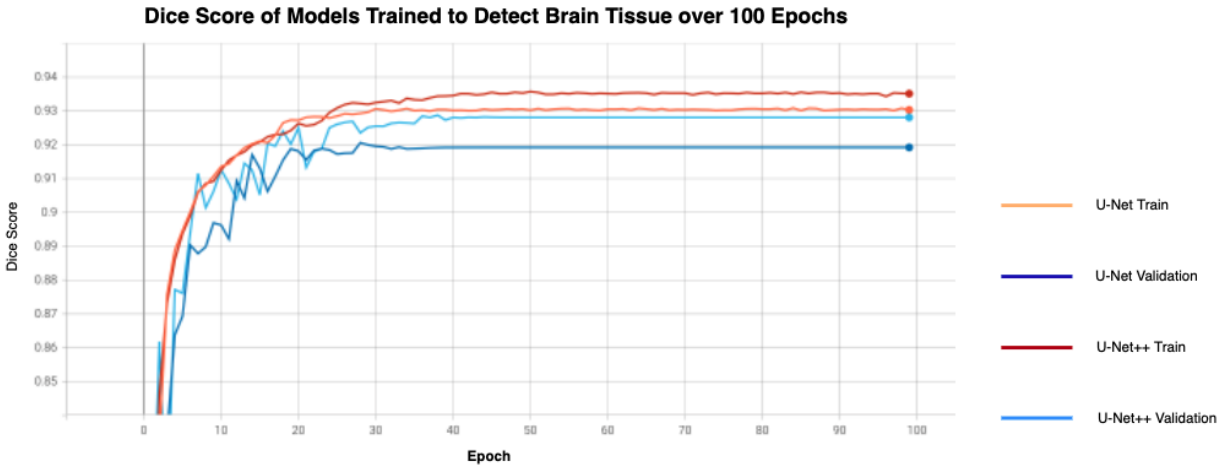

Figure 4.5: Dice score for brain detection over 100 epochs.

Table 4.5: Results of grid search for brain tissue detection.

Model	**n Start Filters**	**dropout Rate**	**Batch size**
U-Net	32	0.3	32
U-Net++	32	0.2	10

Table 4.6: 10-Fold cross validation results of U-Net and U-Net++ for brain tissue detection (mean and standard deviation).

Model	**Dice**	**IoU**	**Precision**	**Recall**
U-Net	0.9084 (0.0505)	0.8360 (0.0792)	0.8860 (0.1069)	0.9626 (0.0190)
U-Net++	0.9086 (0.0512)	0.8364 (0.0805)	0.8847 (0.1060)	0.9637 (0.0183)

Table 4.7: Brain tissue detection test results from U-Net and U-Net++ on unseen data in the testing set.

Model	**Dice**	**IoU**	**Precision**	**Recall**
U-Net	0.9210	0.8535	0.9228	0.9492
U-Net++	0.9251	0.8608	0.9288	0.9469

4.6 summarizes the 10-fold cross validation results. They were also retrained on the entire training set for 100 epochs and tested on the testing set. Table 4.7 shows the results for unseen data. U-Net++ performs slightly better in terms of dice and IoU scores in both cases. Thus, we selected U-Net++ as the final model for brain tissue detection. Using this model, we generated a few examples shown in Figure 4.6. Here, the output of U-Net++ matches the ground truth well, as evident from the dice and IoU scores.

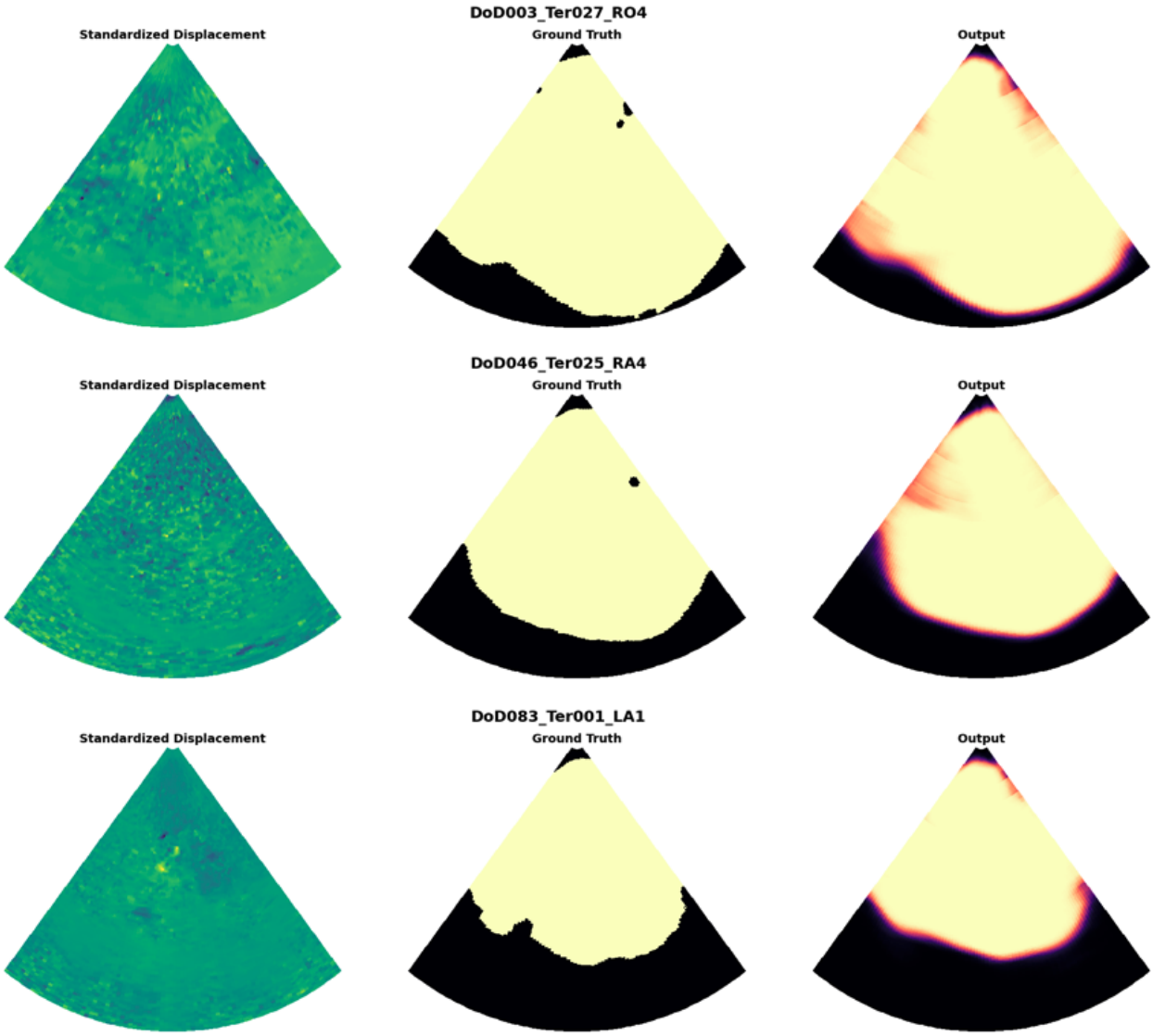

Figure 4.6: Example outputs for brain tissue detection task using the final U-Net++ model. Standardized displacement is the post-processing input; ground truth is the CT brain mask; last column is the model's output.

4.5 Ventricle Detection

This section explores the efficacy of U-Net, Attention U-Net, Cascade A, and Cascade B models in segmenting the ventricles from displacement data. It is important to note that Cascade A (Figure 3.17) and Cascade B (Figure 3.18) are trained and tested with only Net B in the models. Since the quality of ventricle detection depends on the quality of the region of interest (ROI), the ROIs used here is the CT brain mask provided originally. Cascade A and Cascade B will be re-evaluated with Net A (from brain tissue detection) and Net B in the same model in Section 4.7.

Figure 4.7 and Figure 4.8 show the results of experiments for ventricle detection with the following hyperparameters: number of filters in the first convolution layer n of 16, dropout rate of 0.5, batch size of 32, and 100 training epochs. The learning curves plateau at around 70 epochs, indicating complete training. All models exhibit a similar degree of overfitting as the gaps between their training and validation curves are similar. Attention U-Net yields the lowest results, while Cascade A and Cascade B models deliver the highest scores and almost the same plateau for loss function and dice score.

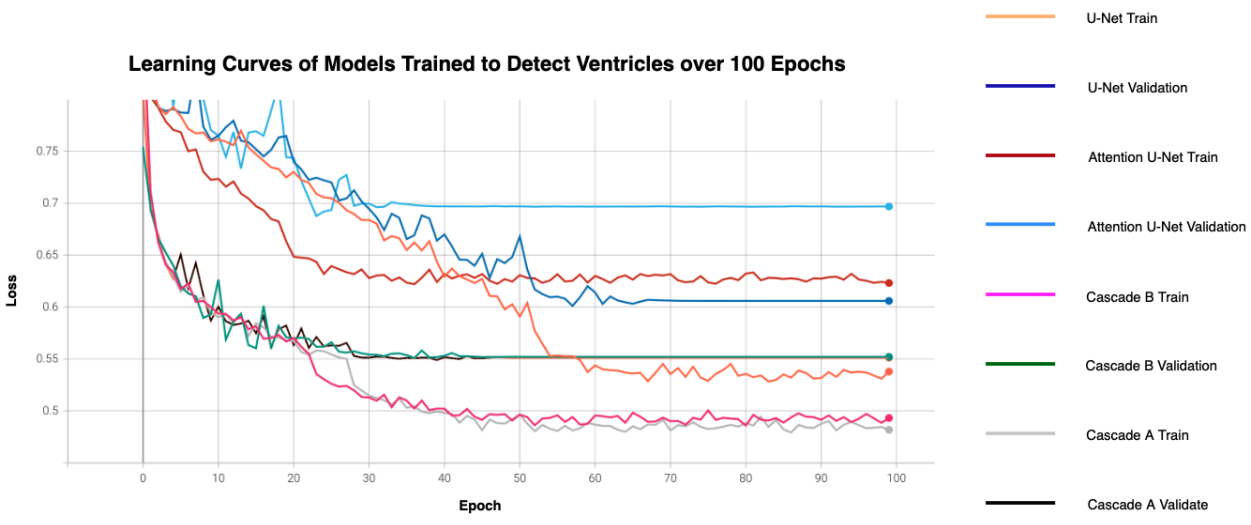

Figure 4.7: Loss curve for ventricle detection over 100 epochs.

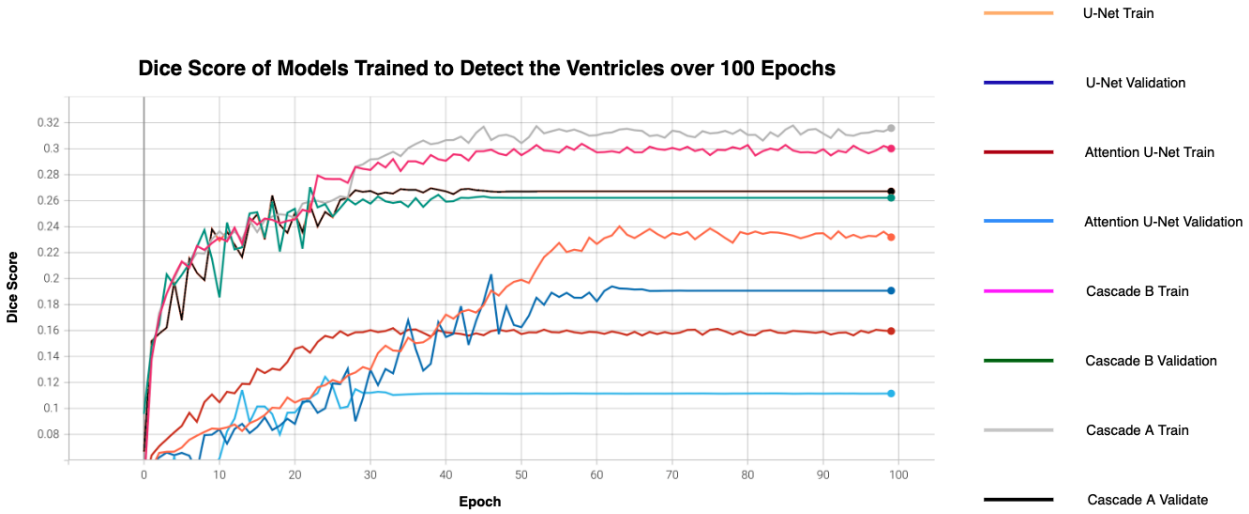

Figure 4.8: Dice score for ventricle detection over 100 epochs.

We also performed a grid search to find the most effective sets of hyperparameters for these models. The resultant hyperparameters is shown in Table 4.8 after training for 70 epochs. We re-built the models using these hyperparameters and evaluated them on the training dataset using 10-fold cross validation procedure. The results are shown in Table 4.9. Cascade A and Cascade B have higher performance in terms of the evaluation metrics than U-Net and Attention U-Net. On the final round of testing on unseen data in the testing dataset, Cascade A was selected as the final model as it yielded higher dice and IoU score. Output examples of this model on unseen data are shown in Figure 4.9. The model was able to locate the general location of the ventricles. However, it cannot match the shape of the ventricles well.

Table 4.8: Results of grid search for ventricle detection.

Model	**n Start Filters**	**dropout Rate**	**Batch size**
U-Net	16	0.5	32
Attention U-Net	32	0.3	20
Cascade A	32	0.3	30
Cascade B	8	0.2	20

Table 4.9: 10-Fold cross validation results from U-Net, Attention U-Net, and cascade models for ventricle detection (mean and standard deviation).

Model	**Dice**	**IoU**	**Precision**	**Recall**
U-Net	0.1832 (0.0261)	0.1025 (0.0165)	0.5254 (0.0814)	0.1647 (0.0235)
Attention Unet	0.1272 (0.0463)	0.0691 (0.0256)	0.3992 (0.1518)	0.1118 (0.0432)
Cascade A	0.2574 (0.0251)	0.1496 (0.0168)	0.5790 (0.0523)	0.1785 (0.0167)
Cascade B	0.2565 (0.0182)	0.1492 (0.0119)	0.6016 (0.0502)	0.1805 (0.0245)

Table 4.10: Ventricle detection test results from U-Net, Attention U-Net, and cascade models on unseen data in the testing set.

Model	**Dice**	**IoU**	**Precision**	**Recall**
U-Net	0.1065	0.0566	0.3789	0.1568
Attention U-Net	0.1353	0.0730	0.4036	0.1857
Cascade A	0.2433	0.1418	0.4905	0.2754
Cascade B	0.2425	0.1396	0.4745	0.2982

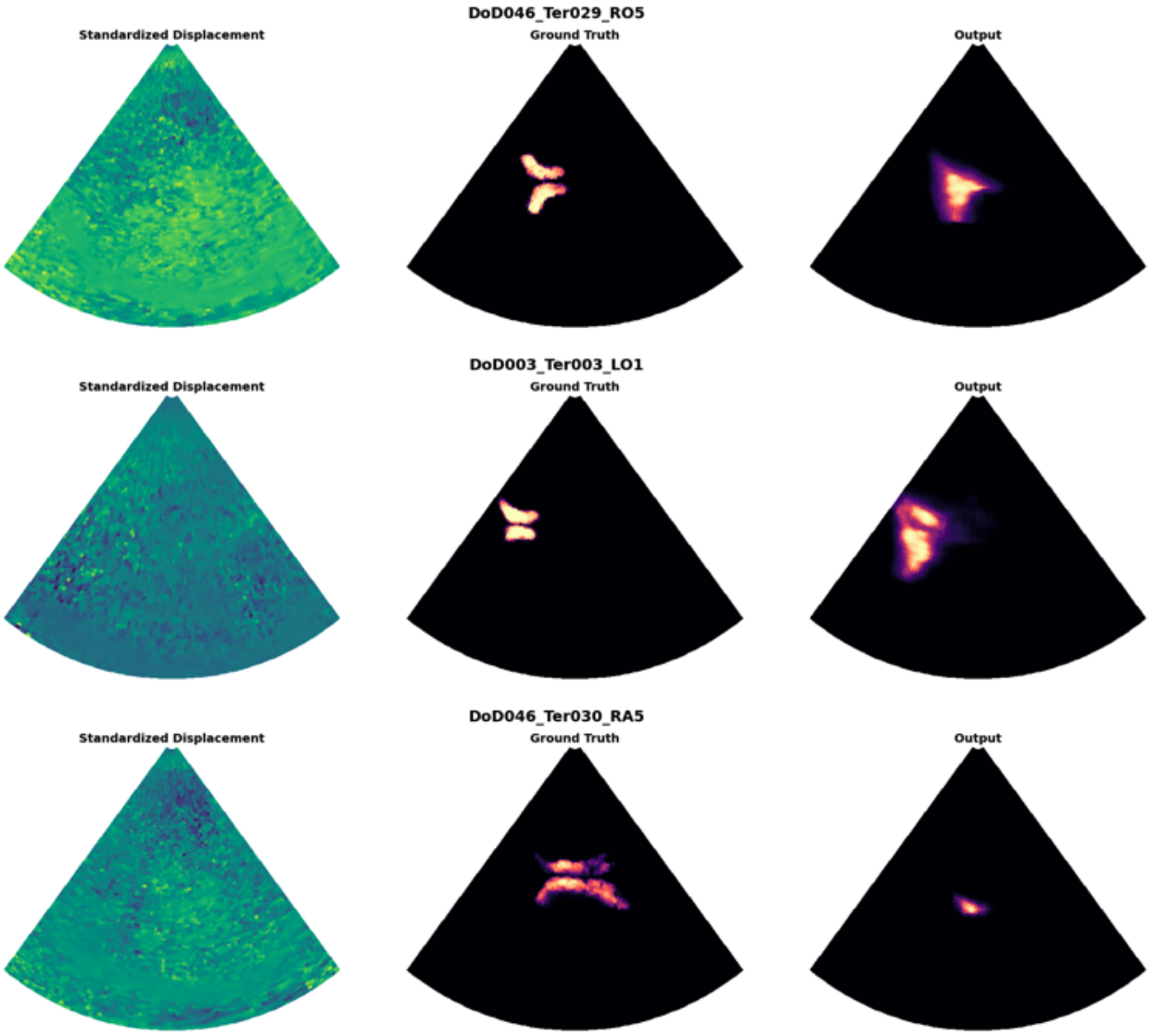

Figure 4.9: Example outputs for ventricle detection task using the model Cascade A. Standardized displacement is the post-processing input; ground truth is the CT ventricle mask; last column is the model's output.

4.6 Intracranial Hemorrhage Detection

This section explores the efficacy of U-Net, Attention U-Net, Cascade A, and Cascade B models in detecting the ventricles from displacement data. Similar to ventricle detection, Cascade A (Figure 3.17) and Cascade B (Figure 3.18) are trained and tested with only Net B in the models. Full integration of Net A and Net B will be discussed in Section 4.7. Figure 4.10 and Figure 4.11 show the results of experiments for ventricle detection with the following hyperparameters: number of filters in the first convolution layer n of 16, dropout rate of 0.5, batch size of 32, and 100 training epochs. Here, the models learning curves plateau at about 80 epochs. Cascade B model produces the highest dice score and reaches the lowest loss value, followed by Cascade A, U-Net, and Attention U-Net, in that order.

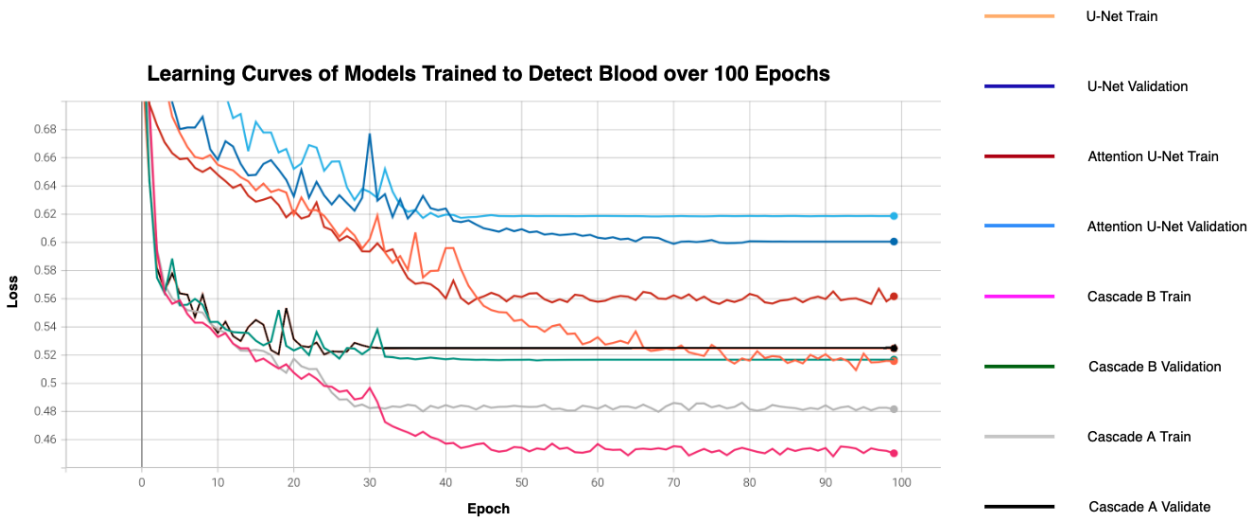

Figure 4.10: Loss curve for blood detection over 100 epochs.

Performing grid search for the four models over 80 epochs resulted in Table 4.11. The models were then built with these hyperparameters and evaluated using the 10-fold cross validation procedure. Table 4.12 demonstrates the results. Here, Cascade A came out on top in terms of dice score, IoU score, and precision, but by a small margin. When tested on the unseen data of the testing dataset, Cascade A again had the highest dice, IoU, and

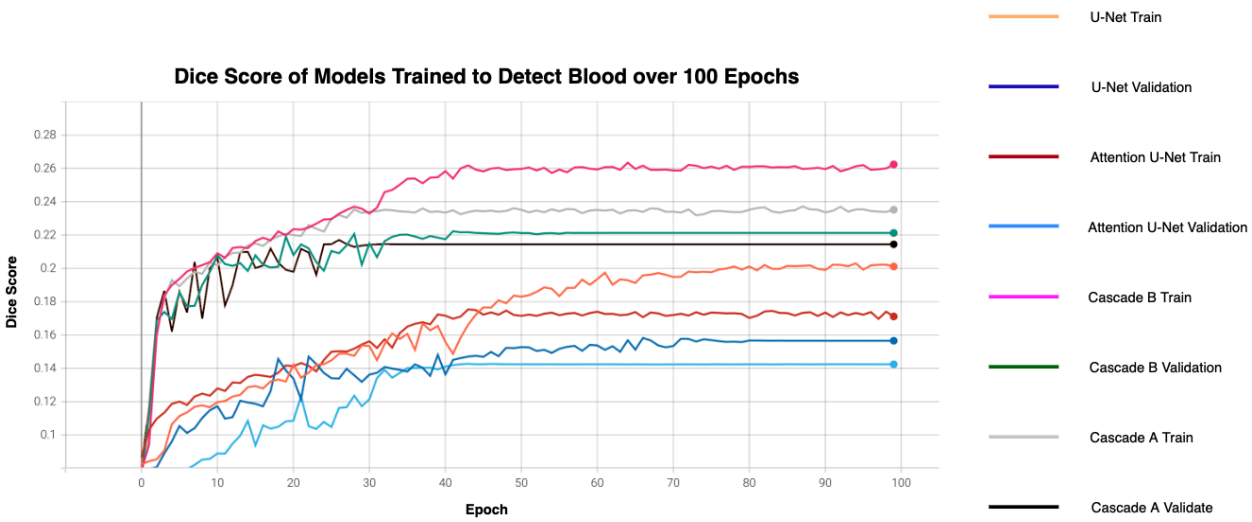

Figure 4.11: Dice score for intracranial hemorrhage detection over 100 epochs.

precision scores. Therefore, it was chosen as the final model for this task. Figure 4.12 shows what to expect when this model tries to find intracranial hemorrhage in unseen data. We see that the model performs very poorly. It cannot detect any of the blood deep in the brain tissue. Instead, it shows the outline of the brain tissue from the ROI (the ground truth from Figure 4.6).

The use of the ROI in the model is not likely the cause of this behavior since the model is also used to detect the ventricles. The outputs of the ventricle detection task do not show the outline of the brain. There are two probable explanations for the poor performance. Firstly, many training examples contain bleeding near or between the skull and the brain tissue. This issue is fatal to convolution models. Convolution-based models learn by generating filters that recognize the gradient between different objects. In other words, they learn to detect the boundaries between the objects in an image. If the boundary is ambiguous, the segmentation model will not perform well. Unfortunately, many training examples have the bleeding at the boundary of the skull and the brain, where the gradient is more profound. The gradient is due to the contrast between soft brain tissue and the solid skull. The

model was misled and learned that there was bleeding where the brain meets the skull. Secondly, the information available in a single displacement frame is not sufficient to segment intracranial hemorrhage. Since convolution-based models learn to find the contrast between two objects, the models would have difficulty segmenting the objects if the contrast is low. The intracranial hemorrhage within the brain tissue might have a faint difference from the surrounding brain tissue.

Table 4.11: Results of grid search for intracranial hemorrhage detection.

Model	**n Start Filters**	**dropout Rate**	**Batch size**
U-Net	16	0.1	20
Attention U-Net	8	0.1	32
Cascade A	8	0.4	10
Cascade B	8	0.2	20

Table 4.12: 10-Fold cross validation results from U-Net, Attention U-Net, and cascade models for intracranial hemorrhage detection (mean and standard deviation).

Model	**Dice**	**IoU**	**Precision**	**Recall**
U-Net	0.1819 (0.0124)	0.1021 (0.0080)	0.5780 (0.0292)	0.0844 (0.0102)
Attention Unet	0.1743 (0.0100)	0.0972 (0.0065)	0.5628 (0.0621)	0.1043 (0.0206)
Cascade A	0.2521 (0.0124)	0.1460 (0.0082)	0.6275 (0.0495)	0.1269 (0.0122)
Cascade B	0.2409 (0.0137)	0.1384 (0.0093)	0.5657 (0.0442)	0.1510 (0.0124)

Table 4.13: Intracranial hemorrhage detection test results from U-Net, Attention U-Net, and cascade models on unseen data in the testing set.

Model	**Dice**	**IoU**	**Precision**	**Recall**
U-Net	0.1624	0.0894	0.5237	0.0331
Attention U-Net	0.1456	0.0798	0.4877	0.0363
Cascade A	0.2299	0.1307	0.5659	0.0941
Cascade B	0.2089	0.1202	0.5426	0.1127

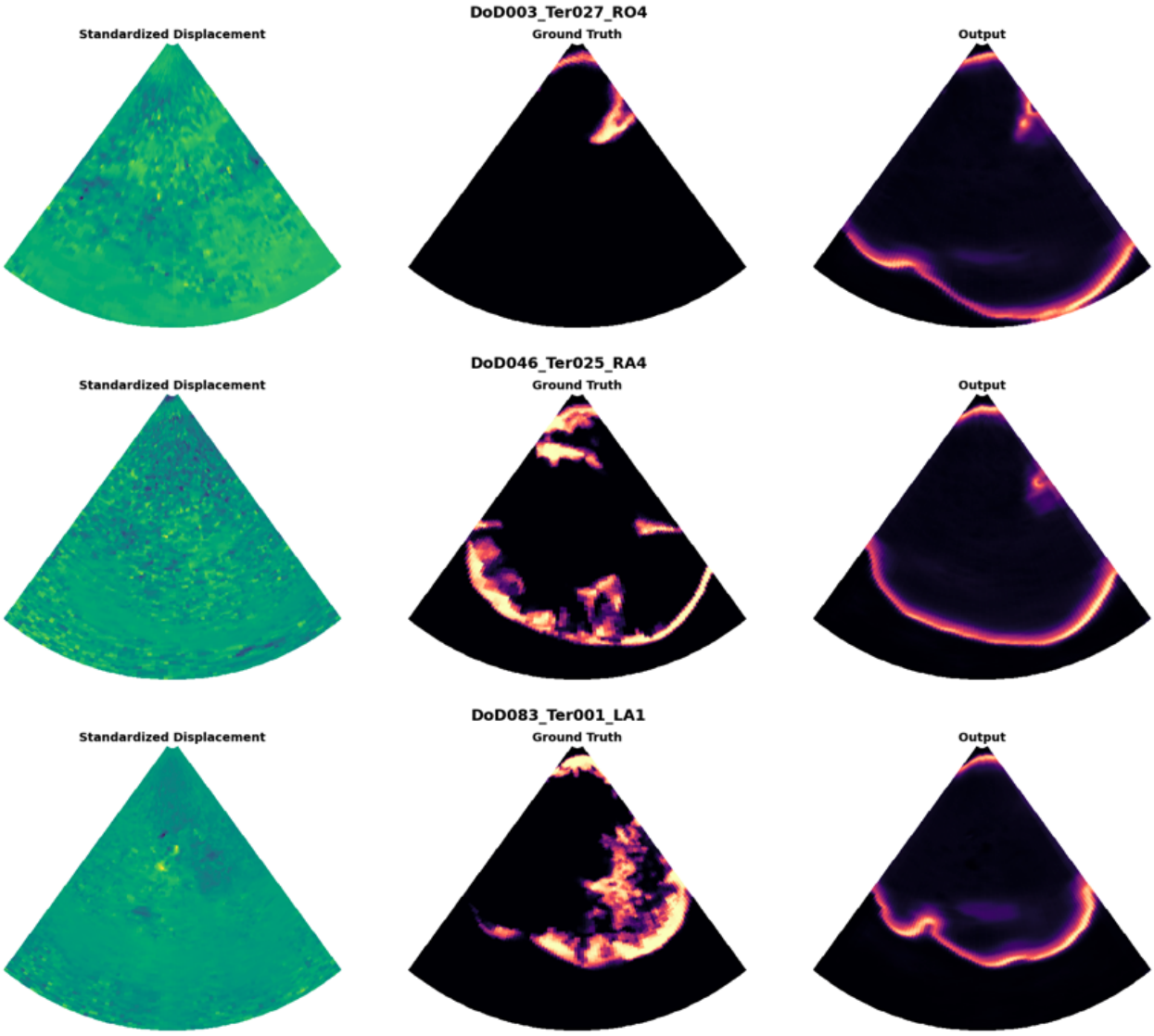

Figure 4.12: Example outputs for intracranial hemorrhage detection task using the model Cascade A. Standardized displacement is the post-processing input; ground truth is the CT blood mask; last column is the model's output.

4.7 Combine brain tissue detection with ventricle detection and blood detection

As mentioned in Section 4.5 and Section 4.6, the models Cascade A and Cascade B were built without using the ROI output of Net A. Instead, the original CT brain masks were used as the ROI. Using the CT brain mask represents the ideal case in which Net A can accurately segment the brain tissue with 100% accuracy. We call these models ideal models. However, we know from Table 4.7 that Net A (U-Net++) could only get to 0.9251 in terms of dice score. This section examines the performance of the fully cascaded model for ventricle and intracranial hemorrhage detection.

Table 4.14 and 4.15 show the evaluation metrics for the final Cascade A model described in Section 4.5 and Section 4.6 versus the complete cascade model, in which Net A (final U-Net++ for brain detection) is chained with Net B (final Cascade A model from ventricle and intracranial hemorrhage detection). For both ventricle and intracranial hemorrhage detection, the scores of the complete cascade models are lower than those of the ideal models, which is expected since Net B cannot provide a perfect segmentation of the brain. Looking at Figure 4.13, we see that the ventricle model can only locate the location of the ventricles without being able to segment the shape of the ventricles. From Figure 4.14, the intracranial hemorrhage model does not detect much bleeding and only gets the outline of the ROI.

Table 4.14: Ventricle detection test results from and the model Cascade A and the fully cascaded model on unseen data in the testing set.

Model	**Dice**	**IoU**	**Precision**	**Recall**
Ideal Cascade A	0.2433	0.1418	0.4905	0.2754
Complete Cascade A	0.1836	0.1011	0.3361	0.2759

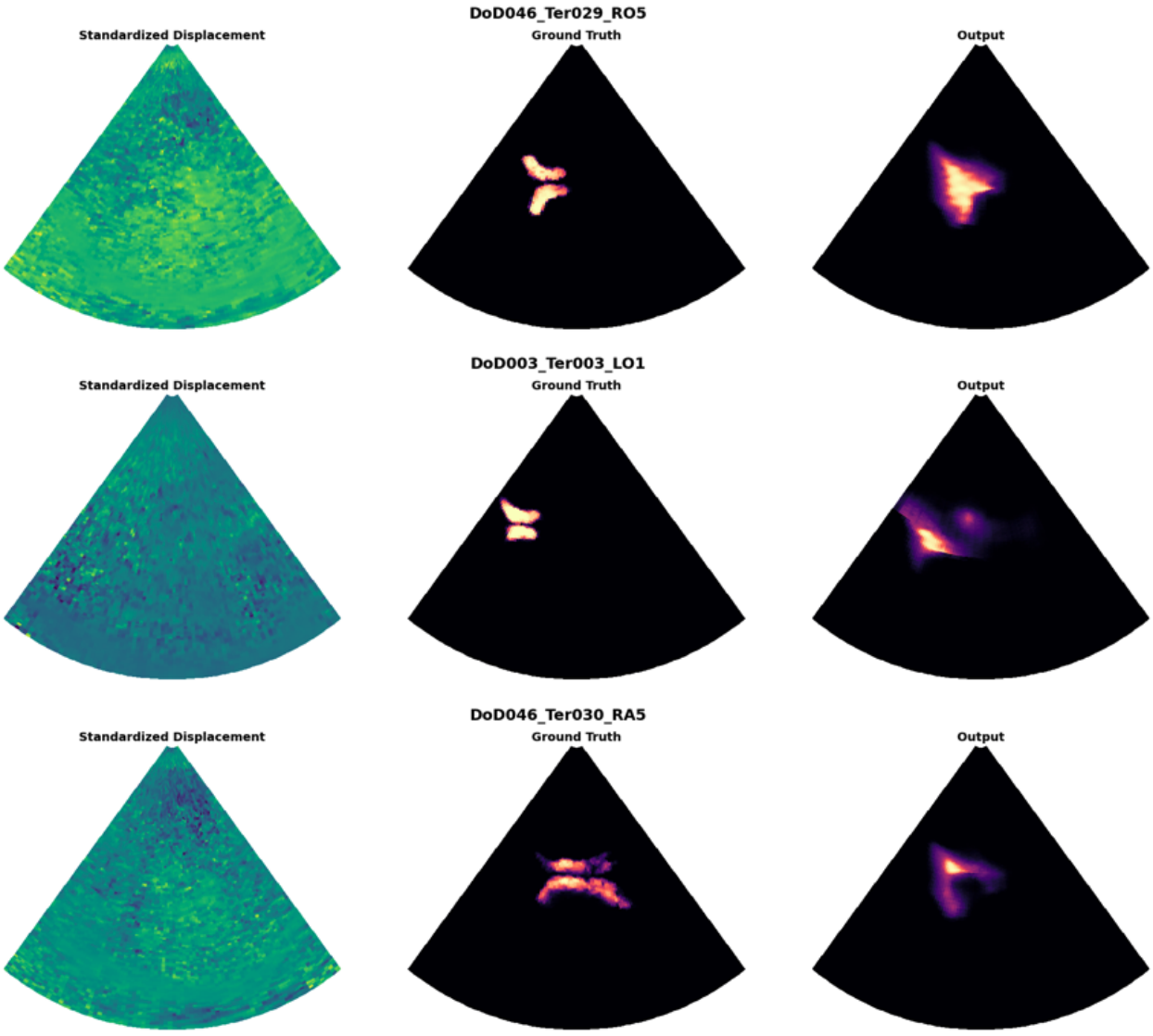

Figure 4.13: Example outputs for ventricle detection task using the fully cascaded Cascade A model. Standardized displacement is the post-processing input; ground truth is the CT blood mask; last column is the model's output.

Table 4.15: Intracranial hemorrhage detection test results from Cascade A model and fully cascaded model on unseen data in the testing set.

Model	**Dice**	**IoU**	**Precision**	**Recall**
Ideal Cascade A	0.2299	0.1307	0.5659	0.0941
Complete Cascade A	0.1573	0.0854	0.4015	0.0769

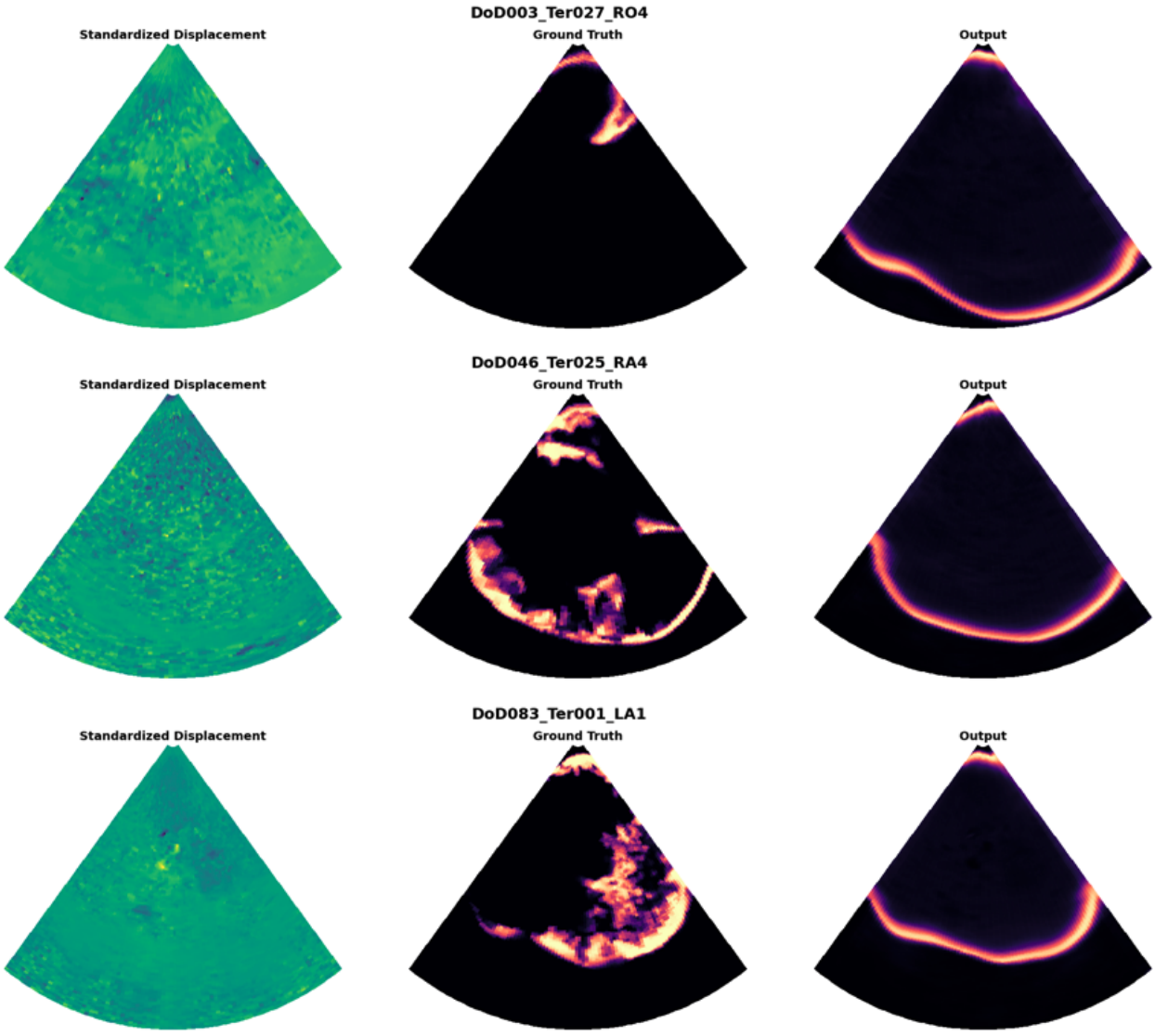

Figure 4.14: Example outputs for intracranial hemorrhage detection task using the fully cascaded Cascade A model. Standardized displacement is the post-processing input; ground truth is the CT blood mask; last column is the model's output.

Chapter 5

CONCLUSION AND FUTURE WORK

In this research, we have implemented five different U-Net-based segmentation models that take ultrasound data in the form of TPI displacement and automatically detect the skull, ventricles, and intracranial hemorrhage from TBI patients. Experiments using the models have shown that the proposed models achieve relative success in skull detection and ventricle detection while performing poorly in detecting intracranial hemorrhage. The final U-Net model successfully segment the part of the skull that encloses the brain. U-Net++ is the final model for brain segmentation, achieving a dice score of higher than 0.9 for the task. Cascade A, which concatenates the region of interest (ROI) with the input, can locate the ventricles but fails to match the ventricles' shape. Finally, we found that the proposed models cannot reliably segment the intracranial hemorrhage from displacement data. Although the result is not at a desirable level for intracranial hemorrhage, we achieved the goal of objective evaluation for convolution-based deep learning models.

Several challenges make it difficult to achieve high model performance. First, the amount of data available is relatively small. Training a deep learning model requires a large amount of data to cover as many existing cases as possible. There are 104 participating patients in this study, from whom data for only 96 patients are deemed suitable to use. Furthermore, the COVID-19 pandemic has severely impacted the data collection. Second, the blast TBI due to exposure to an explosive is the most commonly encountered cTBI on the battlefield [20]. This type of injury has different characteristics from TBI that civilians typically experience [20]. Unfortunately, our dataset consists solely of civilian TBI. This problem hinders a model's ability to generalize to TBI on the battlefield. Third, A majority of the training examples contains epidural and subdural hemorrhage, which occurs at the boundary of the skull and

brain. In the case of subdural hemorrhage, there is only a thin layer of blood between the skull and brain. This data imbalance could be the reason for the poor performance of the cascade model in detecting intracranial hemorrhage. Lastly, CT data used to create the masks (labels) and ultrasound were collected at different times, between which the patients might have received life-saving treatments. The contents of the CT scans might not align perfectly with the contents of the ultrasound scans.

In the future, five main improvements can boost the outcomes of this research. Firstly, to deploy the models to a Terason system, we need a continuous pipeline that performs signal processing on the input data, data clean-up, and inference. This pipeline consists of the Tissue Pulsatility Imaging (TPI) algorithm and the proposed work described in this thesis.

Secondly, we need to convert the code base to take advantage of multiple threads. The development and training of the deep learning models took place on computer systems that contain GPUs. However, the target Terason ultrasound system does not have any graphical computing unit that efficiently performs matrix operations required by the deep learning models. Since deep learning is resource-intensive, this limitation could lead to a long inference time if a model is deployed to the Terason system as is. A future direction is to make inference a multi-threaded process to take advantage of the multi-threading capability of Terason's onboard CPU.

Thirdly, the proposed data preprocessing method involves extracting one frame from a cardiac cycle, where the head's blood volume is the greatest. This method ignores any information pertaining to the dynamics of the pulsating brain as blood enters and leaves the capillaries. Models that utilize this information could potentially improve the performance of ventricle and intracranial-hemorrhage detection tasks.

The fourth improvement pertains to data collection. The performance of a deep-learning model depends heavily on the amount of training data. We only had access to a small pool of patients whose injuries resulted in many subdural and epidural examples. Additional data collection that diversifies the dataset by capturing more examples of other intracranial hemorrhage types will benefit the development of deep learning models significantly. Moreover,

suppose we can collect more data in the future, the CT scans and ultrasound scans should be gathered at around the same time, preferably at hospital admission, to ensure the highest data quality. Additionally, collecting medical data is expensive and relies on the patient's consent. To increase the amount of training examples, we can explore synthetic TPI data in future research.

Finally, we could employ the brain and skull detection outcomes to improve intracranial-hemorrhage detection. As aforementioned, our proposed models successfully segment the brain-enclosing skull areas and the brain tissue. On the other hand, subdural and epidural hemorrhage is extremely challenging to identify from TPI images. We hypothesized that subdural and epidural hemorrhage displace the brain tissue, creating a gap between the skull and brain. Overlapping the segmented brain mask and skull mask could reveal this gap and allow us to detect the two intracranial hemorrhage types.

BIBLIOGRAPHY

[1] Cranial ultrasound. https://www.healthlinkbc.ca/tests-treatments-medications/medical-tests/cranial-ultrasound. Accessed: 2022-02-21.

[2] Trauma care. https://www.uwmedicine.org/specialties/emergency-medicine/trauma-care. Accessed: 2022-02-22.

[3] Traumatic brain injury. https://www.hopkinsmedicine.org/health/conditions-and-diseases/traumatic-brain-injury. Accessed: 2022-02-20.

[4] Brain Tumor Foundation Inc 2000. Management and prognosis of severe traumatic brain injury.

[5] Zahangir Alom, Tarek M Taha, and Vijayan K Asari. Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation. *arXiv*, page 12, 2018.

[6] Wenjia Bai, Hideaki Suzuki, Chen Qin, Giacomo Tarroni, Ozan Oktay, Paul M. Matthews, and Daniel Rueckert. Recurrent neural networks for aortic image sequence segmentation with sparse annotations. *arXiv:1808.00273 [cs]*, August 2018. arXiv: 1808.00273.

[7] Yassine Barhoumi and Rasool Ghulam. Scopeformer: n-cnn-vit hybrid model for intracranial hemorrhage classification. 2021.

[8] J Campbell, J Clark, D White, and C Jenkins. Pulsatile echo-encephalography. *Acta Neurologica Scandinavica. Supplementum*, 45:1–57, 1970.

[9] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. 2021.

[10] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. 2021.

[11] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. 2017.

[12] Jun Cheng, Jiang Liu, Yanwu Xu, Fengshou Yin, D. W. K Wong, Ngan-Meng Tan, Dacheng Tao, Ching-Yu Cheng, Tin Aung, and Tien Yin Wong. Superpixel Classification Based Optic Disc and Optic Cup Segmentation for Glaucoma Screening. *IEEE transactions on medical imaging*, 32(6):1019–1032, 2013.

[13] Venkateswararao Cherukuri, Peter Ssenyonga, Benjamin C Warf, Abhaya V Kulkarni, Vishal Monga, and Steven J Schiff. Learning Based Segmentation of CT Brain Images: Application to Postoperative Hydrocephalic Scans. *IEEE transactions on biomedical engineering*, 65(8):1871–1884, 2018.

[14] Patrick Ferdinand Christ, Mohamed Ezzeldin A. Elshaer, Florian Ettlinger, Sunil Tatavarty, Marc Bickel, Patrick Bilic, Markus Rempfler, Marco Armbruster, Felix Hofmann, Melvin D'Anastasi, Wieland H. Sommer, Seyed-Ahmad Ahmadi, and Bjoern H. Menze. Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields. *arXiv:1610.02177 [cs]*, 9901:415–423, 2016. arXiv: 1610.02177.

[15] Sagi Eppel, Haoping Xu, and Alan Aspuru-Guzik. Computer vision for liquid samples in hospitals and medical labs using hierarchical image segmentation and relations prediction. 2021.

[16] David Evans and W. Norman McDicken. Chapter 11, Signal Processing for Colour Flow Imaging. In *Doppler Ultrasound: Physics, Instrumentation, and Signal Processing*, pages 245–287. Chichester, England: John Wiley and Sons, Ltd, 2nd edition, 2000.

[17] Xiaorui Feng, Chaoli Wang, Shuqun Cheng, and Lei Guo. Automatic Liver and Tumor Segmentation of CT Based on Cascaded U-Net. In *Proceedings of 2018 Chinese Intelligent Systems Conference*, Lecture Notes in Electrical Engineering, pages 155–164. Springer Singapore, Singapore, 2018.

[18] MD Freeman, William David and MD Aguilar, Maria I. Intracranial hemorrhage: Diagnosis and management. *Neurologic clinics*, 30(1):211–240, 2012.

[19] Huazhu Fu, Jun Cheng, Yanwu Xu, Damon Wing Kee Wong, Jiang Liu, and Xiaochun Cao. Joint Optic Disc and Cup Segmentation Based on Multi-Label Deep Network and Polar Transformation. *IEEE transactions on medical imaging*, 37(7):1597–1605, 2018.

[20] Ling G, Bandak F, Armonda R, Grant G, and Ecklund J. Explosive blast neurotrauma. *J Neurotrauma*, June 2009.

[21] Yang Gao, Jeff M. Phillips, Yan Zheng, Renqiang Min, P. Thomas Fletcher, and Guido Gerig. Fully convolutional structured LSTM networks for joint 4D medical image segmentation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1104–1108, Washington, DC, April 2018. IEEE.

[22] Steven Guan, Amir A. Khan, Siddhartha Sikdar, and Parag V. Chitnis. Fully Dense UNet for 2-D Sparse Photoacoustic Tomography Artifact Removal. *IEEE Journal of Biomedical and Health Informatics*, 24(2):568–576, February 2020.

[23] John T. Guibas, Tejpal S. Virdi, and Peter S. Li. Synthetic Medical Images from Dual Generative Adversarial Networks. *arXiv:1709.01872 [cs]*, January 2018. arXiv: 1709.01872.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv*, December 2015. arXiv: 1512.03385.

[25] Jeremy J. Heit, Michael Iv, and Max Wintermark. Imaging of intracranial hemorrhage. *Journal of Stroke*, 19(1):11–27, 2017.

[26] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997. _eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf.

[27] Murtadha D. Hssayeni, Muayad S. Croock, Aymen D. Salman, Hassan Falah Al-khafaji, Zakaria A. Yahya, and Behnaz Ghoraani. Intracranial hemorrhage segmentation using a deep convolutional model. *Data (Basel)*, 5(1):14, 2020.

[28] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation. 2020.

[29] Adnan A. Hyder, Colleen A. Wunderlich, Prasanthi Puvanachandra, G. Gururaj, and Olive C. Kobusingye. The impact of traumatic brain injuries: A global perspective. *PubMed*, 22(5):341–353, 2007.

[30] Nabil Ibtehaz and M. Sohel Rahman. MultiResUNet : Rethinking the U-Net Architecture for Multimodal Biomedical Image Segmentation. *Neural Networks*, 121:74–87, January 2020. arXiv: 1902.04049.

[31] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv:1611.07004 [cs]*, November 2018. arXiv: 1611.07004.

[32] Shruti Jadon. A survey of loss functions for semantic segmentation. 2020.

[33] Debesh Jha, Michael A. Riegler, Dag Johansen, Pål Halvorsen, and Håvard D. Johansen. DoubleU-Net: A Deep Convolutional Neural Network for Medical Image Segmentation. *arXiv:2006.04868 [cs, eess]*, June 2020. arXiv: 2006.04868.

[34] Michael L Johnson, Carol M Rumack, Eric J Mannes, and Kevin E Appareti. Detection of neonatal intracranial hemorrhage utilizing real-time and static ultrasound. *Journal of clinical ultrasound*, 9(8):427–433, 1981.

[35] Krishna Chaitanya Kaluva, Mahendra Khened, Avinash Kori, and Ganapathy Krishnamurthi. 2D-Densely Connected Convolution Neural Networks for automatic Liver and Tumor Segmentation. *arXiv:1802.02182 [cs]*, January 2018. arXiv: 1802.02182.

[36] John C. Kucewicz, Barbrina Dunmire, Nicholas D. Giardino, Daniel F. Leotta, Marla Paun, Stephen R. Dager, and Kirk W. Beach. Tissue pulsatility imaging of cerebral vasoreactivity during hyperventilation. *Ultrasound in Medicine & Biology*, 34(8):1200–1208, 2008.

[37] John C. Kucewicz, Barbrina Dunmire, Daniel F. Leotta, Heracles Panagiotides, Marla Paun, and Kirk W. Beach. Functional Tissue Pulsatility Imaging of the Brain During Visual Stimulation. *Ultrasound in Medicine & Biology*, 33(5):681–690, May 2007.

[38] Tao Lei, Risheng Wang, Yong Wan, Bingtao Zhang, Hongying Meng, and Asoke K. Nandi. Medical Image Segmentation Using Deep Learning: A Survey. *arXiv*, December 2020. arXiv: 2009.13120.

[39] Wen Li, Fucang Jia, and Qingmao Hu. Automatic Segmentation of Liver Tumor in CT Images with Deep Convolutional Neural Networks. *Journal of computer and communications*, 3(11):146–151, 2015. Publisher: Modern Science Publishers.

[40] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440. IEEE, 2015. ISSN: 1063-6919.

[41] Dwarikanath Mahapatra, Behzad Bozorgtabar, Jean-Philippe Thiran, and Mauricio Reyes. Efficient Active Learning for Image Classification and Segmentation Using a Sample Selection and Conditional Generative Adversarial Network. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Lecture Notes in Computer Science, pages 580–588, Cham, 2018. Springer International Publishing.

[42] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, Levente Lanczi, Elizabeth Gerstner, Marc-Andre Weber, Tal Arbel, Brian B Avants, Nicholas Ayache, Patricia Buendia, D. Louis Collins, Nicolas Cordier, Jason J Corso, Antonio Criminisi, Tilak Das, Herve Delingette, Cagatay Demiralp, Christopher R Durst, Michel Dojat, Senan Doyle, Joana Festa, Florence Forbes, Ezequiel Geremia, Ben Glocker, Polina Golland, Xiaotao Guo, Andac Hamamci, Khan M Iftekharuddin, Raj Jena, Nigel M John, Ender Konukoglu, Danial Lashkari, Jose Antonio Mariz, Raphael Meier, Sergio Pereira, Doina Precup, Stephen J Price, Tammy Riklin Raviv, Syed M. S Reza, Michael Ryan, Duygu Sarikaya, Lawrence Schwartz, Hoo-Chang Shin, Jamie Shotton, Carlos A Silva, Nuno Sousa, Nagesh K Subbanna, Gabor Szekely, Thomas J Taylor, Owen M Thomas, Nicholas J Tustison, Gozde Unal, Flor Vasseur, Max Wintermark, Dong Hye Ye, Liang Zhao, Binsheng Zhao, Darko Zikic, Marcel Prastawa, Mauricio Reyes, and Koen Van Leemput. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2015. Place: PISCATAWAY Publisher: IEEE.

[43] Karsten Meyer-Wiethe, Fabrizio Sallustio, and Rolf Kern. Diagnosis of intracerebral hemorrhage with transcranial ultrasound. 27(2):40–47, 2009.

[44] Fausto Milletari. *A Hough Voting Strategies for Segmentation, Detection and Tracking.* PhD dissertation, Technische Universität München, 2018.

[45] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *arXiv:1606.04797 [cs]*, June 2016. arXiv: 1606.04797.

[46] Fausto Milletary. *Hough Voting Strategies for Segmentation, Detection and Tracking.* PhD thesis, Technischen Universität München, November 2017.

[47] Jakub Nemcek, Tomas Vicar, and Roman Jakubicek. Weakly supervised deep learning-based intracranial hemorrhage localization. 2021.

[48] Nhan T Nguyen, Dat Q Tran, Nghia T Nguyen, and Ha Q Nguyen. A cnn-lstm architecture for detection of intracranial hemorrhage on ct scans. 2020.

[49] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention U-Net: Learning Where to Look for the Pancreas. *arXiv*, page 10, 2018.

[50] Yuya Onishi, Atsushi Teramoto, Masakazu Tsujimoto, Tetsuya Tsukamoto, Kuniaki Saito, Hiroshi Toyama, Kazuyoshi Imaizumi, and Hiroshi Fujita. Multiplanar analysis for pulmonary nodule classification in CT images using deep convolutional neural network and generative adversarial networks. *International journal for computer assisted radiology and surgery*, 15(1):173–178, 2019.

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241, Cham, 2015. Springer International Publishing. ISSN: 0302-9743.

[52] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv:1801.04381 [cs]*, March 2019. arXiv: 1801.04381.

[53] E E Sauerbrei, M Digney, P B Harrison, and P L Cooperberg. Ultrasonic evaluation of neonatal intracranial hemorrhage and its complications. *Radiology*, 139(3):677–685, 1981.

[54] Hyunseok Seo, Charles Huang, Maxime Bassenne, Ruoxiu Xiao, and Lei Xing. Modified U-Net (mU-Net) With Incorporation of Object-Dependent High Level Features for Improved Liver and Liver-Tumor Segmentation in CT Images. *IEEE Transactions on Medical Imaging*, 39(5):1316–1325, May 2020.

[55] Tzu-Hsi Song, Victor Sanchez, Hesham EIDaly, and Nasir M Rajpoot. Dual-Channel Active Contour Model for Megakaryocytic Cell Segmentation in Bone Marrow Trephine Histology Images. *IEEE transactions on biomedical engineering*, 64(12):2913–2923, 2017.

[56] Strandness and Summer. *Hemodynamics for Surgeons*. New York: Grune and Strattion, 1975.

[57] Saeid Asgari Taghanaki, Yefeng Zheng, S Kevin Zhou, Bogdan Georgescu, Puneet Sharma, Daguang Xu, Dorin Comaniciu, and Ghassan Hamarneh. Combo loss: Handling input and output imbalance in multi-organ segmentation. *Computerized medical imaging and graphics*, 75:24–33, 2019.

[58] Wei Tang, Dongsheng Zou, Su Yang, and Jing Shi. DSL: Automatic Liver Segmentation with Faster R-CNN and DeepLab. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, Lecture Notes in Computer Science, pages 137–147, Cham, 2018. Springer International Publishing.

[59] G Unsgaard, A Gronningsaeter, S Ommedal, and TAN Hernes. Brain operations guided by real-time two-dimensional ultrasound: New possibilities as a result of improved image quality. *Neurosurgery*, 51(2):402–411, 2002.

[60] Refael Vivanti, Ariel Ephrat, Leo Joskowicz, Naama Lev-Cohain, Onur A Karaaslan, and Jacob Sosna. Automatic Liver Tumor Segmentation in Follow-Up CT Scans: Preliminary Method and Results. In *Patch-Based Techniques in Medical Imaging*, Lecture Notes in Computer Science, pages 54–61, Cham, 2016. Springer International Publishing. ISSN: 0302-9743.

[61] Shuo Wang, Mu Zhou, Zaiyi Liu, Zhenyu Liu, Dongsheng Gu, Yali Zang, Di Dong, Olivier Gevaert, and Jie Tian. Central focused convolutional neural networks: Developing a data-driven model for lung nodule segmentation. *Medical image analysis*, 40:172–183, 2017.

[62] Zhengyang Wang, Na Zou, Dinggang Shen, and Shuiwang Ji. Non-local U-Net for Biomedical Image Segmentation. *arXiv*, February 2020. arXiv: 1812.04103.

[63] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. ResNeSt: Split-Attention Networks. *arXiv:2004.08955 [cs]*, December 2020. arXiv: 2004.08955.

[64] Yizhe Zhang, Michael T. C. Ying, Lin Yang, Anil T. Ahuja, and Danny Z. Chen. Coarse-to-fine stacked fully convolutional nets for lymph node segmentation in ultrasound images. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 443–448, 2016.

[65] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. *IEEE Transactions on Medical Imaging*, 39(6):1856–1867, June 2020.

[66] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. *arXiv:1606.06650 [cs]*, June 2016. arXiv: 1606.06650.

Appendix A

RESULTS OF GRID SEARCH EXPERIMENT FROM TENSORBOARD

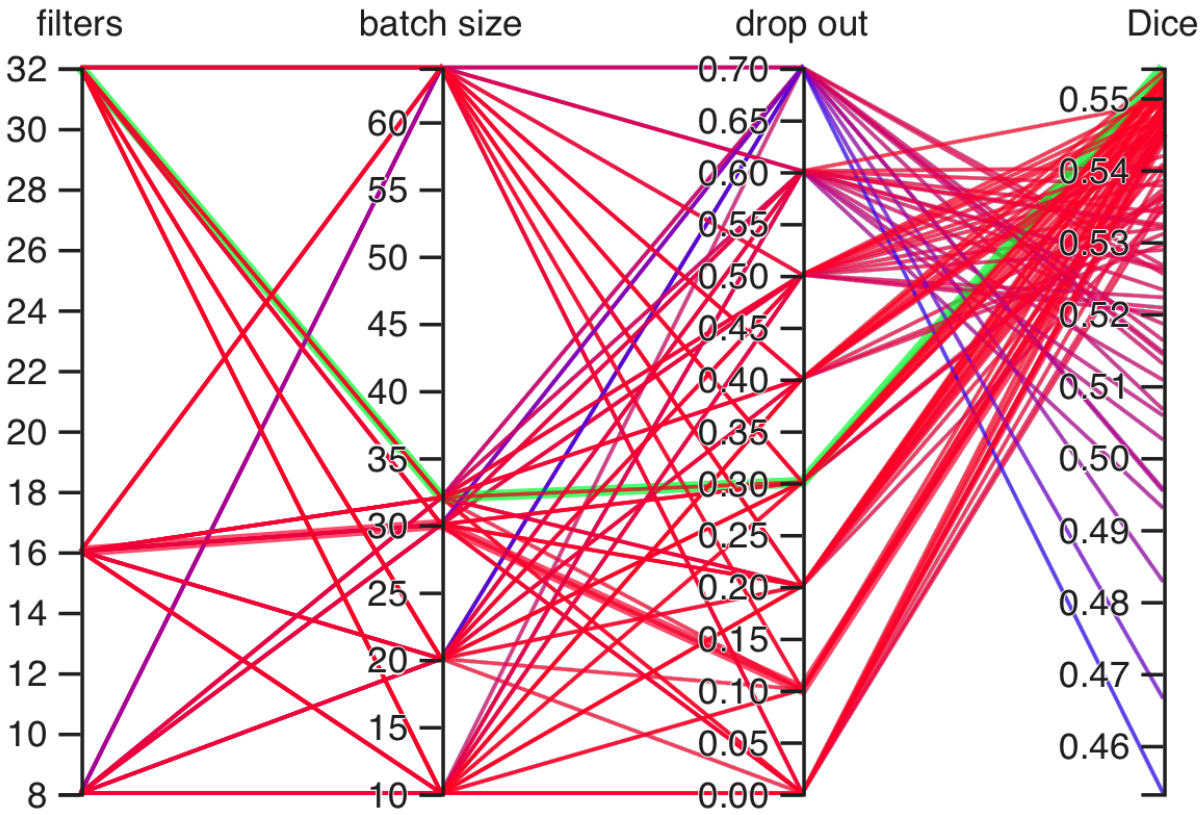

Figure A.1: Results of grid search for U-Net that performs skull detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.

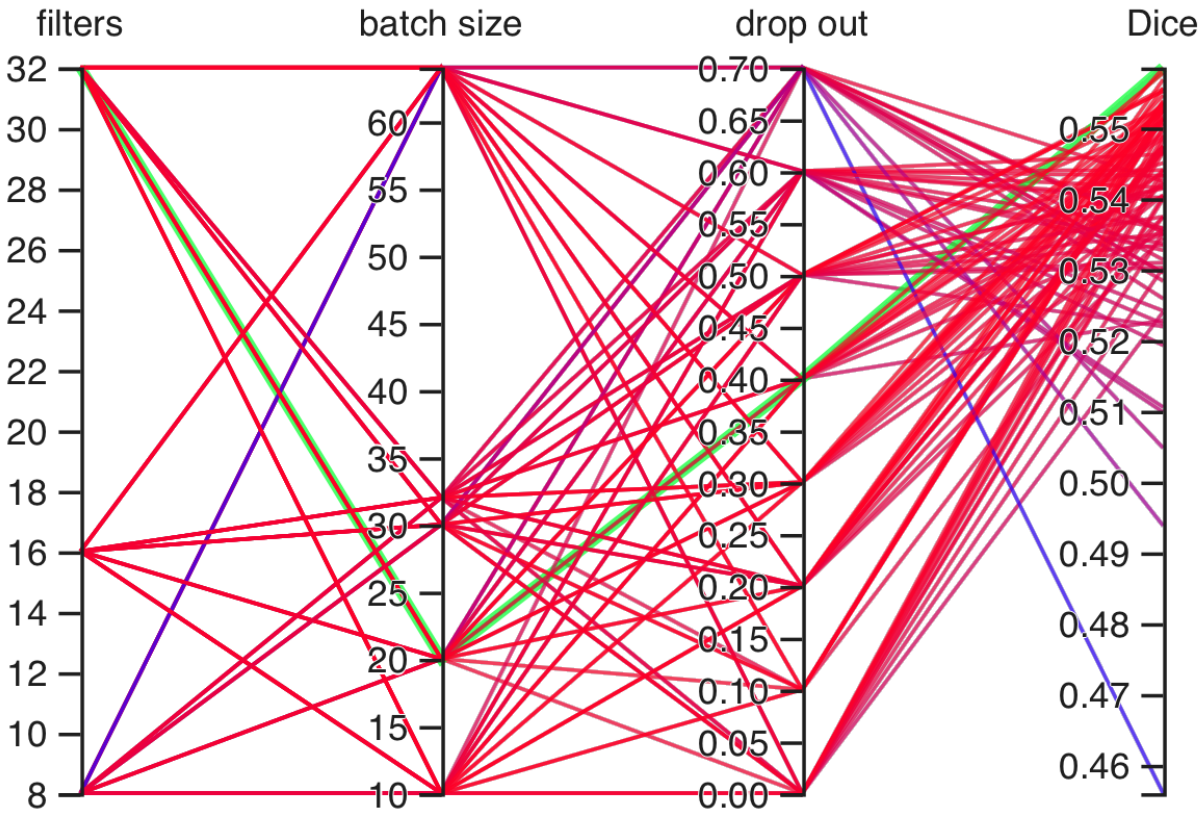

Figure A.2: Results of grid search for U-Net++ that performs skull detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score..

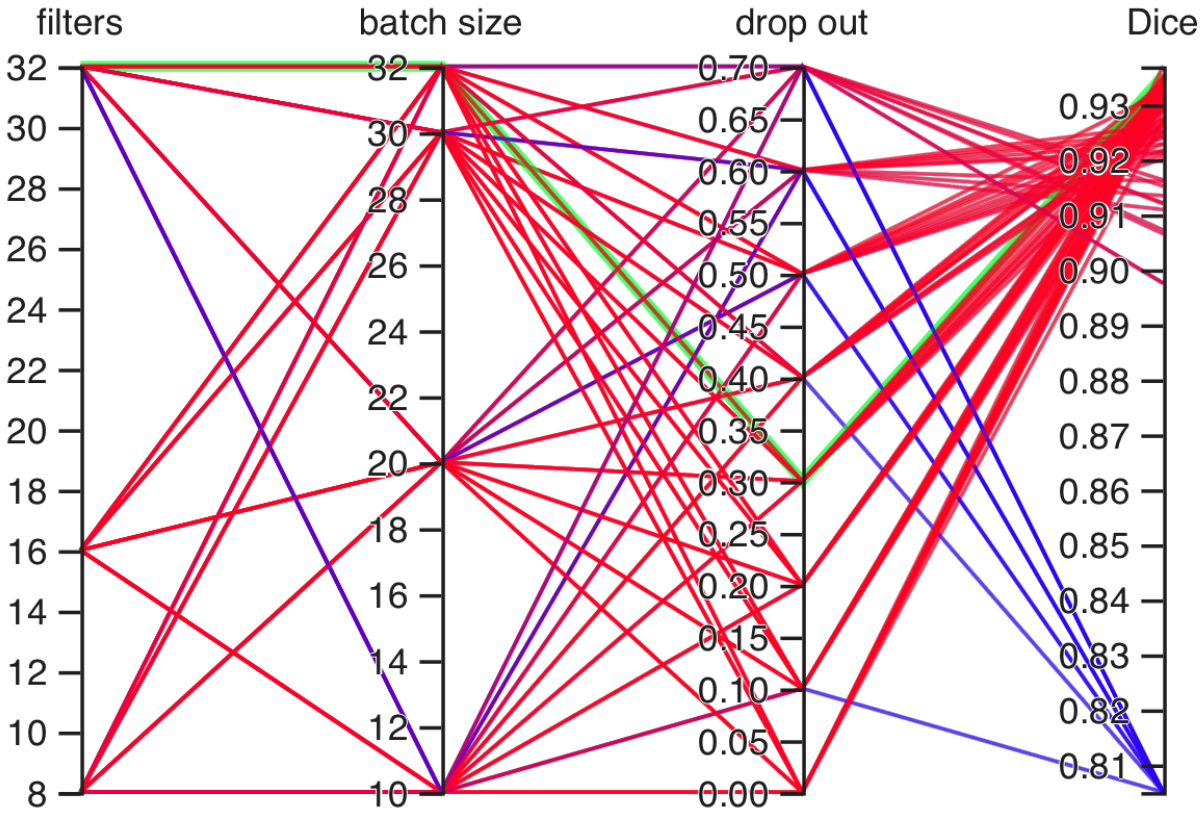

Figure A.3: Results of grid search for U-Net that performs brain tissue detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.

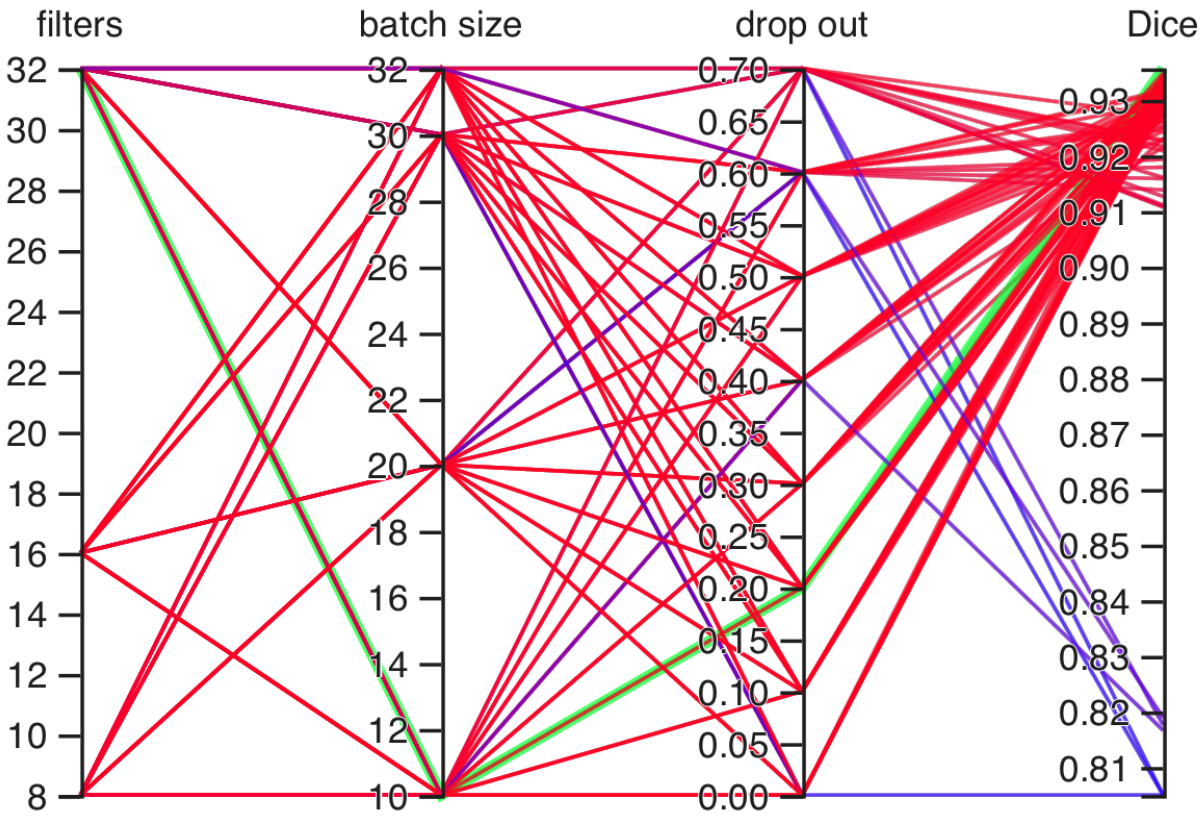

Figure A.4: Results of grid search for U-Net++ that performs brain tissue detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.

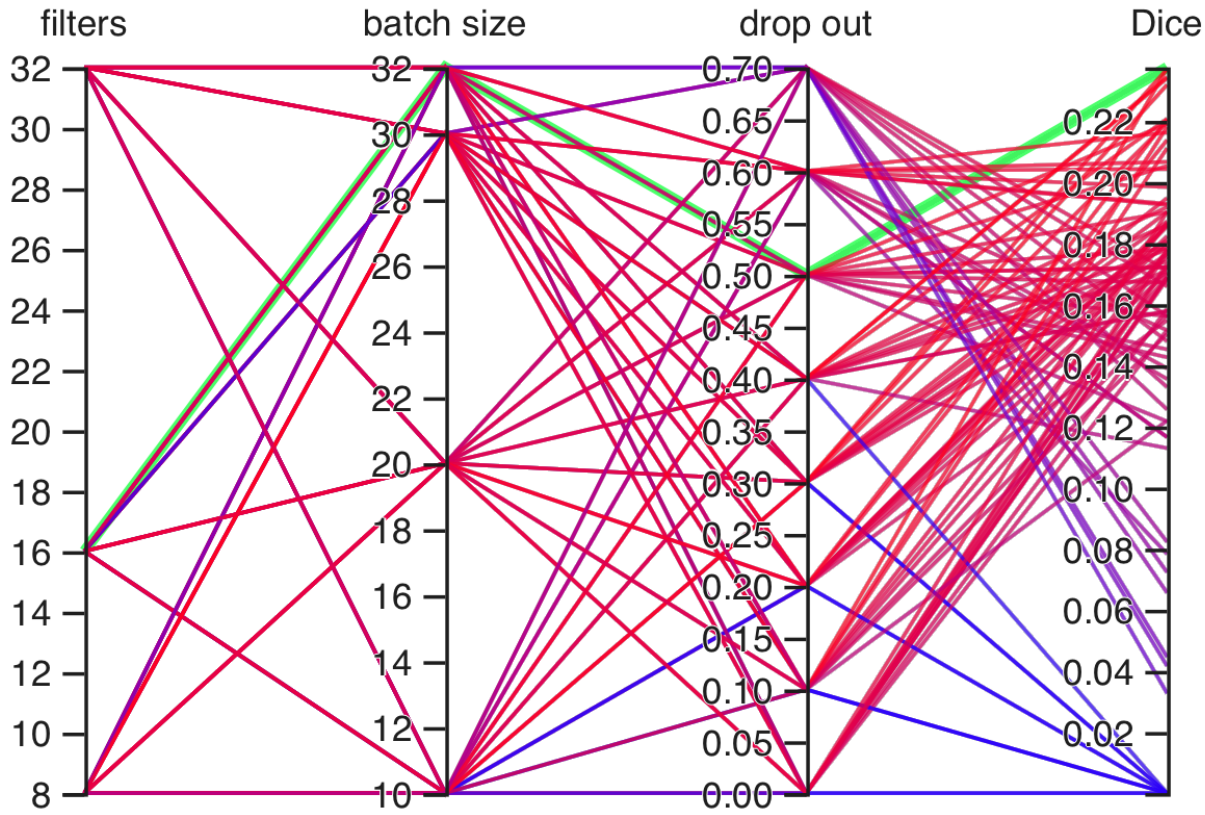

Figure A.5: Results of grid search for U-Net that performs ventricle detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.

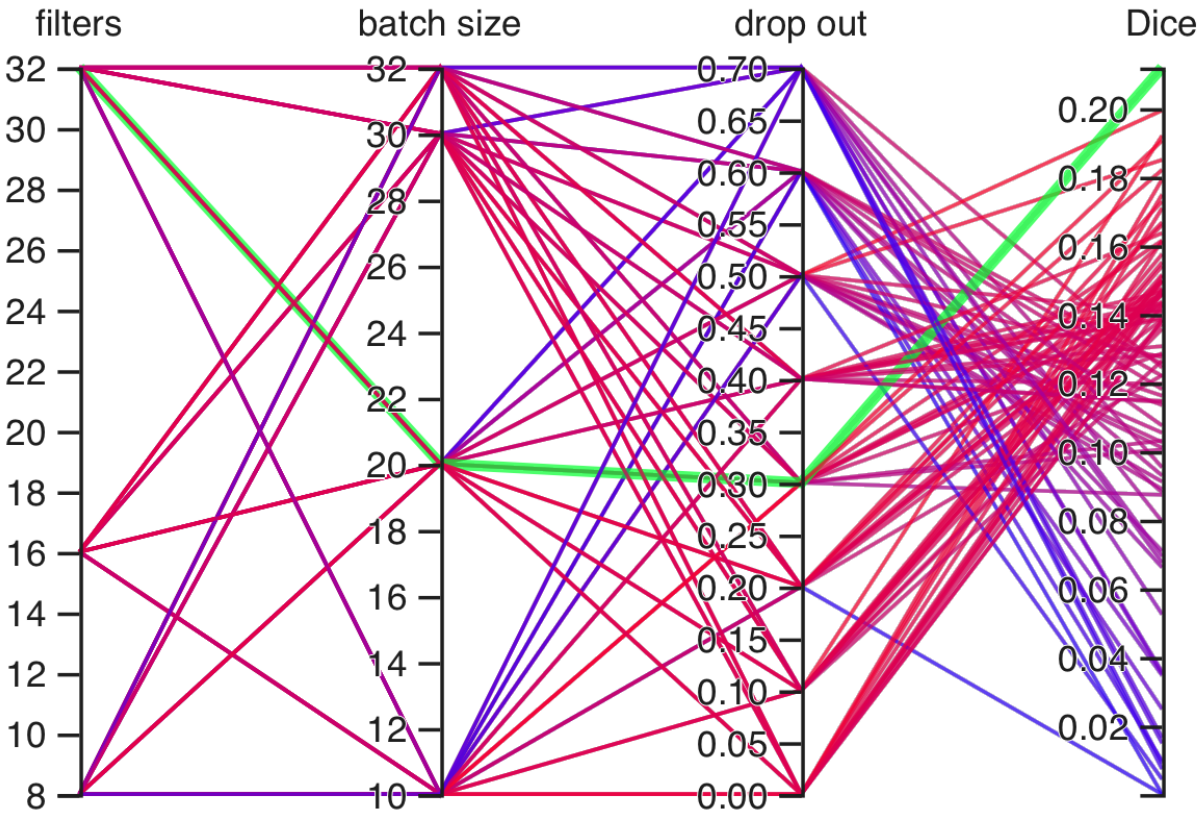

Figure A.6: Results of grid search for Attention U-Net that performs ventricle detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.

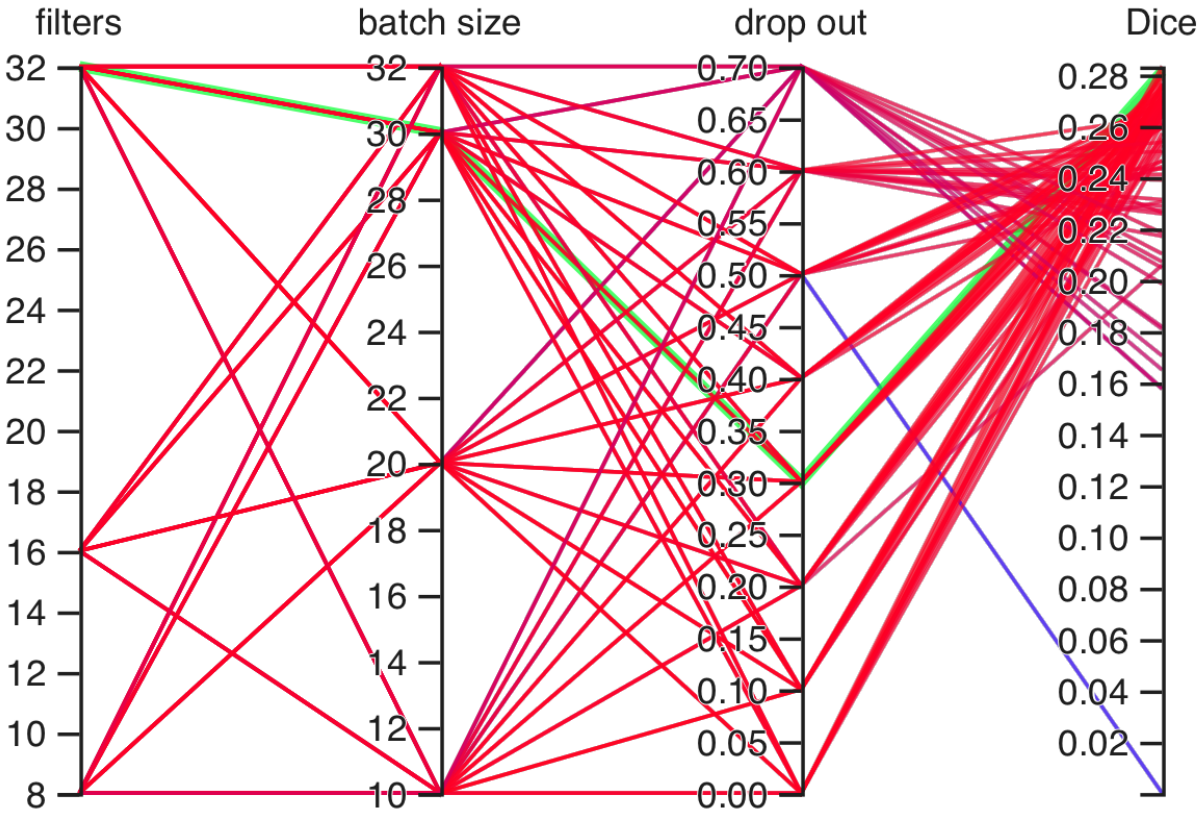

Figure A.7: Results of grid search for Cascade A model that performs ventricle detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.

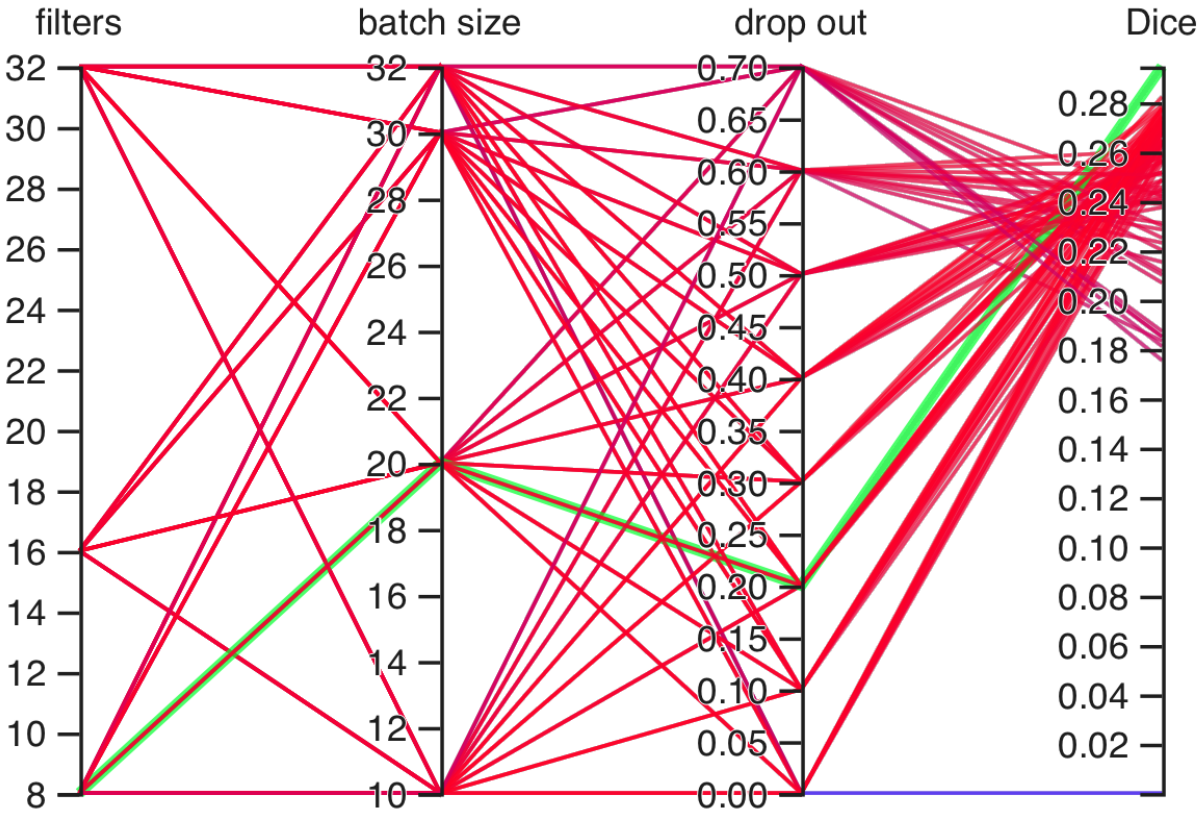

Figure A.8: Results of grid search for Cascade B model that performs ventricle detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.

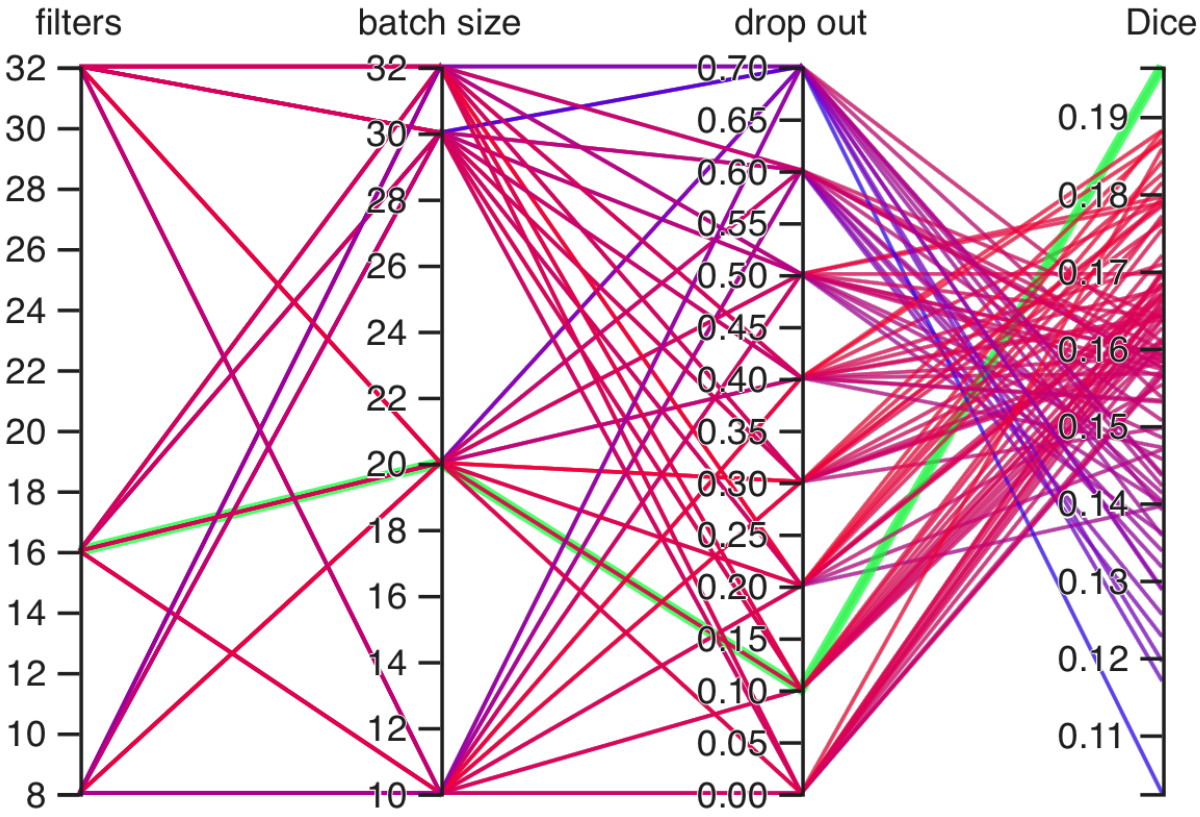

Figure A.9: Results of grid search for U-Net that performs intracranial hemorrhage detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.

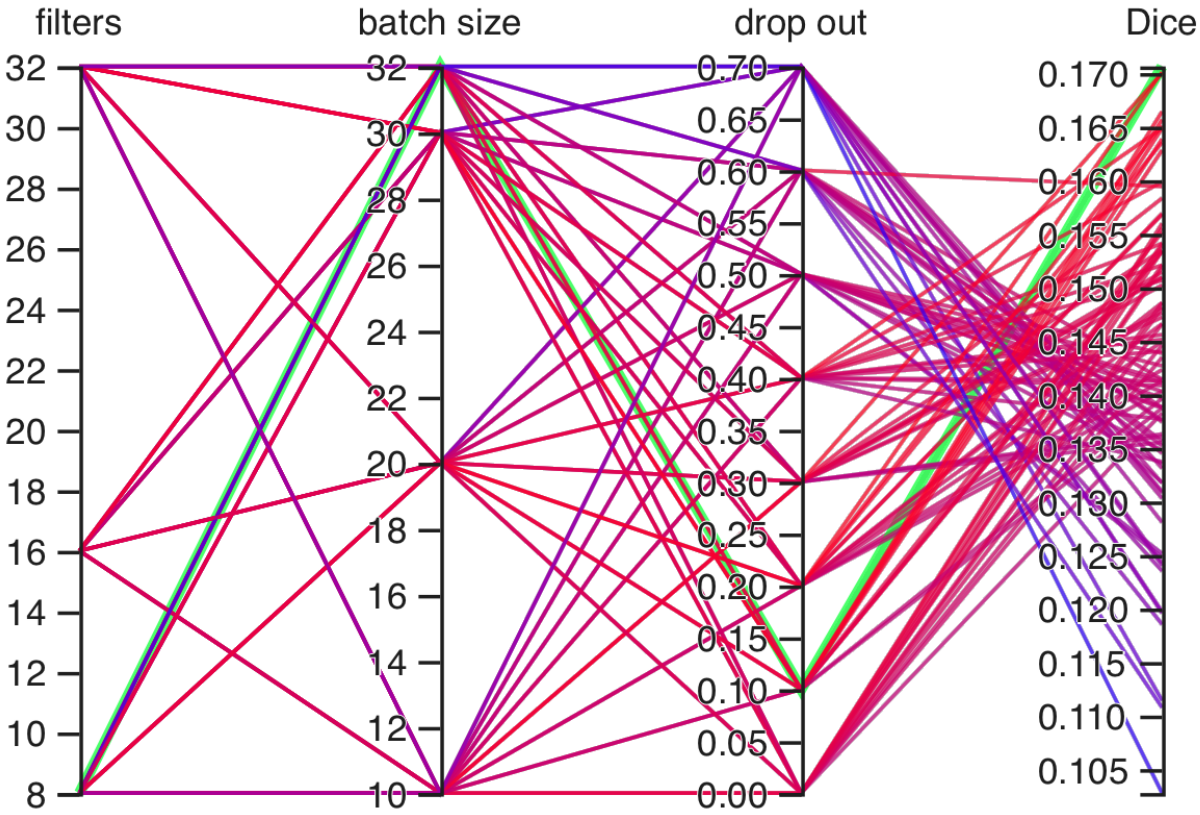

Figure A.10: Results of grid search for Attention U-Net that performs intracranial hemorrhage detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.

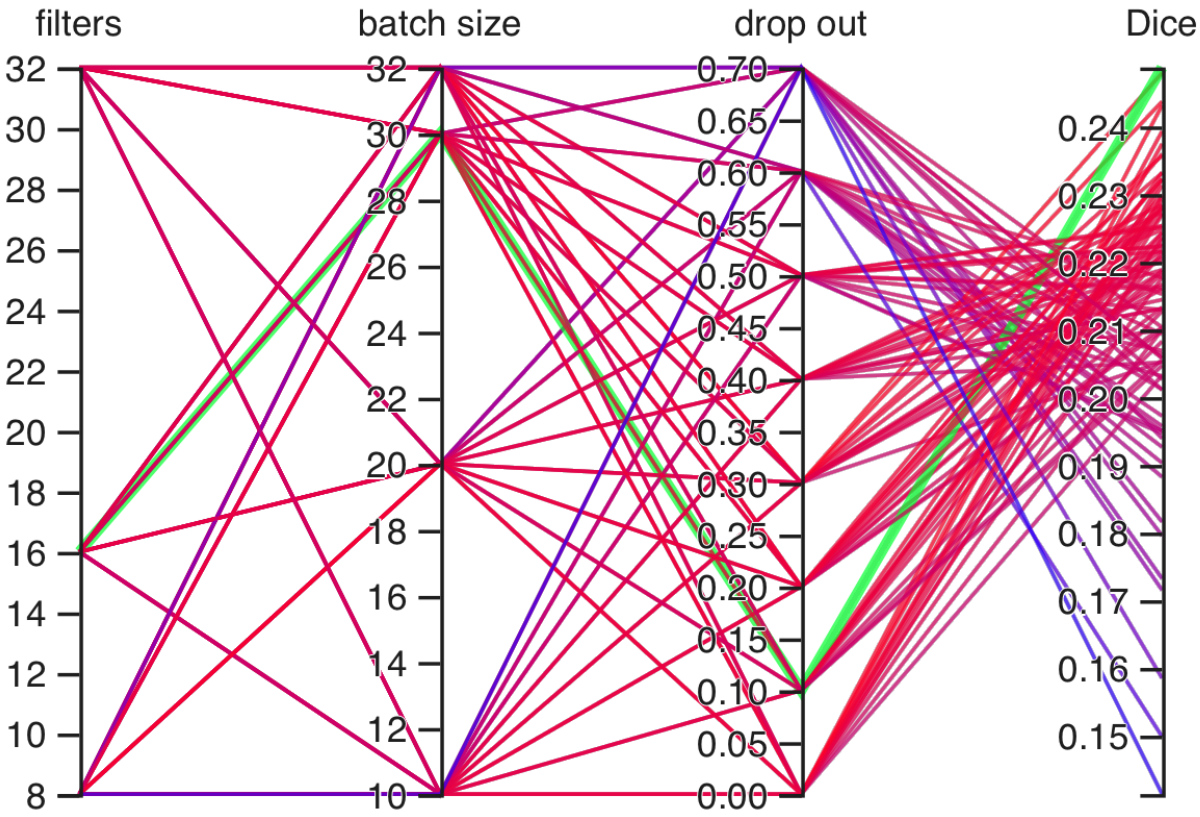

Figure A.11: Results of grid search for Cascade A model that performs intracranial hemorrhage detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.

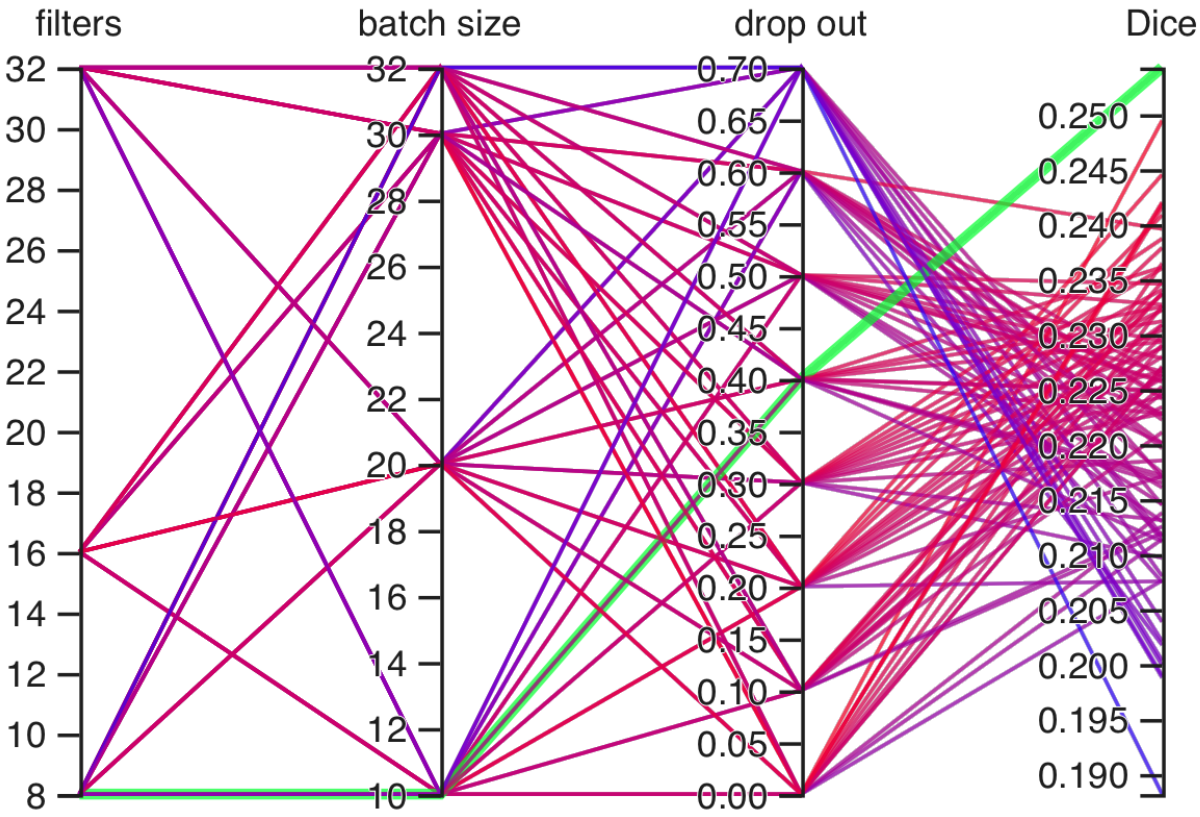

Figure A.12: Results of grid search for Cascade B model that performs intracranial hemorrhage detection. The search spaces include the number of filters in the first convolution layer, dropout rate, and training batch size. The path highlighted in green connects the parameters that produce the highest dice score.