

IMPROVING GAN LEARNING DYNAMICS FOR THYROID NODULE SEGMENTATION

by

Alisa Kunapinun

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of
Doctor of Engineering in Mechatronics

Examination Committee: Assoc. Prof. Mongkol Ekpanyapong (Chairperson)
Assoc. Prof. Dittapong Songsaeng, MD. (Co-Chair)
Prof. Matthew N. Dailey
Prof. Manukid Parnichkun
Dr. Chadaporn Keatmanee

External Examination: Assoc. Prof. Worapan Kusakunniran
Faculty of Information and Communication
Technology, Mahidol University
Nakhon Pathom, Thailand

Nationality: Thai

Previous Degree: Master of Engineering in Mechatronics
Asian Institute of Technology
Pathum Thani, Thailand

Scholarship Donor: Royal Thai Government

Asian Institute of Technology
School of Engineering and Technology
Thailand

December 2022

AUTHOR'S DECLARATION

I, Alisa Kunapinun, declare that the research work carried out for this dissertation was in accordance with the Asian Institute of Technology regulations. The work presented in it is my own and has been generated by me as the result of my original research, and if external sources were used, such sources have been cited. It is original and has not been submitted to any other institution to obtain another degree or qualification. This is a true copy of the dissertation, including final revisions.

Date: 28 September 2022

Name: Alisa Kunapinun

Signature: 

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my father, Tanut Kunapinun and my sister, Promarin Kunapinun, for providing the financial support that made which this research possible. I would like to thank my supervisor, Assoc. Prof. Mongkol Ekpanyapong, for bringing me back to school to do my Ph.D. Moreover, many thanks to Prof. Matthew N. Dailey, for his constant support and guidance throughout this project and other crazy projects I have carried on, and for giving me a chance to work as his teaching assistant and improve my teaching and technical skills. Without him, I would not have been able to complete this research. Special thanks are also due to Assoc. Prof. Dittapong Songsaeng, for his invaluable advice, continuous support, and patience during my research. He showed me the beauty of the medical imaging field. Lastly, I would like to thank the Thyroid Nodule Assessment supported the Ultrasound Images Project (NBTC) for financial support through grant # A63-1-(2)-018.

ABSTRACT

The thyroid gland, which is responsible for secretion of critical hormones that regulate the body, is susceptible to a number of pathological conditions best diagnosed using ultrasound. A common diagnosis method involves identifying then characterizing the appearance of thyroid nodules: solid or fluid-filled lumps that could be benign or malignant. Physicians could be aided in this endeavor by an automated system to precisely identify nodules in a given ultrasound image. This thesis therefore presents a novel algorithm for segmenting thyroid nodules in ultrasound images named **StableSeg GAN**. The algorithm is based on the concept of image-to-image translation, which combines traditional supervised semantic segmentation with unsupervised learning using generative adversarial networks (GANs). GANs have been found to improve semantic segmentation models' performance in specific tasks. However, GAN learning dynamics are famously unstable, oftentimes leading to mode collapse. It is well known that controlling the discriminator in a GAN to not learn too quickly often improves generator learning, making the learning smoother and avoiding mode collapse. StableSeg GANs exploit the concept of closed-loop control of the gain on the loss output of the discriminator to stabilize training. We find that gain control leads to smoother generator training and avoids the mode collapse that typically occurs when the discriminator learns too quickly relative to the generator. We also find that the combination of the supervised and unsupervised learning styles encourages both low-level accuracy and high-level consistency. As a test of the concept of controlled hybrid supervised and unsupervised semantic segmentation, StableSeg GANs use DeeplabV3+ as the generator, Resnet18 as the discriminator, and PID control to stabilize the GAN learning process. The new model is superior to the state-of-the-art DeeplabV3+ in terms of intersection over union (IoU), with a mean IoU of 81.26% over a challenging test set. The results of our thyroid nodule segmentation experiments show that StableSeg GANs have flexibility to segment nodules more accurately than supervised segmentation models or uncontrolled GANs.

Keywords: Thyroid nodules, Ultrasound, Semantic segmentation, Deep learning, CNNs, GANs, DeepLab, Automatic control

CONTENTS

	Page
ACKNOWLEDGMENTS	iii
ABSTRACT	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER 1 INTRODUCTION	1
1.1 Background of the Study	1
1.2 Statement of the Problem	3
1.3 Research Scope	4
1.4 Objectives of the Study	4
1.5 Contributions	4
1.6 Publication	5
1.7 Ethics Approval	5
1.8 Organization of the Study	5
CHAPTER 2 BACKGROUND	6
2.1 Thyroid	6
2.1.1 Ultrasound for Diagnosis	7
2.1.2 Thyroid Diseases Indicating Ultrasound Examination	7
2.1.3 Thyroid Nodule and ACR TI-RADS	7
2.2 Deep Learning in Medical Imaging	11
2.2.1 Thyroid Nodule Segmentation in Deep Learning	12
2.3 Deep Convolutional Neural Networks (CNNs) for Segmentation	13
2.4 DeepLabV3+	15
2.4.1 Atrous Convolution	15
2.4.2 DeeplabV3+ Backbone	17
2.4.3 Implementation of DeepLabV3+ for Thyroid Nodule Ultrasound images	19
2.5 Loss Function for Segmentation Networks	19
2.5.1 Binary Cross-Entropy Loss Function	20
2.5.2 Soft Dice Loss Function	21
2.5.3 Lovasz-Softmax Function	21

	Page
2.6 Measuring Segmentation Quality	22
2.7 Generative Adversarial Networks (GANs)	23
2.7.1 GANs Architecture	24
2.7.2 Dynamics of Loss Functions for GANs	25
2.7.3 Pix2Pix Model in Image Segmentation	26
2.8 GAN Stabilization	26
CHAPTER 3 METHODOLOGY	32
3.1 Dataset Preparation	32
3.2 Data Augmentation	33
3.3 Designing DeepLabV3-GANs	34
3.4 Designing StableSeg GAN	35
CHAPTER 4 EXPERIMENTAL RESULTS	39
4.1 StableSeg GAN Training Parameters	39
4.2 Generator Loss in StableSeg GANs	39
4.3 Supervised and Unsupervised Loss Function Evolution During Training	41
4.4 Ablation Study	42
4.5 Experiment with Multiple Organs	43
CHAPTER 5 CONCLUSION AND DISCUSSION	46
5.1 Discussion	46
5.2 Conclusion	46
REFERENCES	48
APPENDICES	52
APPENDIX A: JACOBIAN OF THE GRADIENT VECTOR FIELD IN GANS	53
APPENDIX B: ALL TEST RESULTS IN STABLESEG GANS	56
VITA	83

LIST OF TABLES

Tables	Page
Table 3.1 Test IoU With and Without Augmentation of Training Images	34
Table 4.1 Accuracy of DeeplabV3+GANs with 8-layer Discriminator and Different γ	40
Table 4.2 Accuracy of StableSeg GANs with 8-layer Discriminator and Different γ	41
Table 4.3 Accuracy of StableSeg GANs with ResNet18 Discriminator and Different γ	41
Table 4.4 Ablation Study: Accuracy Comparison between StableSeg GANs and Other models	43
Table 4.5 Accuracy of General StableSeg GANs with ResNet18 Discriminator	45

LIST OF FIGURES

Figures	Page
Figure 1.1 Sample Segmented Thyroid Nodules in Ultrasound Images	2
Figure 2.1 Thyroid Gland Model	6
Figure 2.2 Ultrasound Image of a Normal Thyroid Gland	8
Figure 2.3 Example Thyroid Nodules	9
Figure 2.4 ACR TI-RADS 2017	11
Figure 2.5 Current Medical Imaging and Technological Trends	12
Figure 2.6 Mask R-CNN Model	13
Figure 2.7 Conceptual Design of Autoencoder Networks	14
Figure 2.8 DeepLabV3+ Architecture	15
Figure 2.9 Atrous Convolution	16
Figure 2.10 Comparison of Atrous and Ordinary Convolution	16
Figure 2.11 DeepLabV3+ Encoder Network with Atrous Spatial Pyramid Pooling	17
Figure 2.12 EfficientNet b7 Networks	18
Figure 2.13 MBConv Block	19
Figure 2.14 Overall Process of DeepLabV3+ 1st Place Solution in MICCAI 2020 TN-SCUI Challenge. Reprinted from <a href="https://github.com/WAMAWA
MA/TNSCUI2020-Seg-Rank1st">https://github.com/WAMAWA MA/TNSCUI2020-Seg-Rank1st	20
Figure 2.15 Distribution of Medical Imaging Research Using GANs	23
Figure 2.16 Applications Using GANs in Medical Imaging	24
Figure 2.17 Different GAN Designs. Reprinted from Xin Yi (2019)	25
Figure 2.18 Characteristic Loss Function Dynamics for Stable and Unstable GAN Learning	27
Figure 3.1 Annotation Nodule Process	32
Figure 3.2 Cropping and Annotation of a Nodule Image	33
Figure 3.3 Sample Augmented Images	34
Figure 3.4 DeepLabV3+ GAN Architecture	35
Figure 3.5 StableSeg GAN Training for One Epoch	37
Figure 4.1 G/D Loss With and Without Stabilization	40
Figure 4.2 Breakdown of Generator Loss into Unsupervised Loss and Supervised Loss	42

	Page
Figure 4.3 Mask Result Comparison for a Random Sample from the Test Set	44
Figure 4.4 Generalized StableSeg GANs Network	45

CHAPTER 1

INTRODUCTION

This chapter provides an introduction to the dissertation topic, starting from background on medical imaging and a problem statement. Then the chapter explains the dissertation scope and the objectives of the study.

1.1 Background of the Study

Medical technology is important for human life. Technology improvement can help physicians diagnose and cure people. Currently, some computing technologies are being adopted in the medical field, such as machine learning, deep learning, and machine vision. Some technologies can be combined for improvement. For example, deep learning has been used to upgrade the accuracy of inference in machine vision.

Machine vision is widely used in medical imaging in many applications requiring classification, detection, or segmentation, with image types such as CT scans, MRI, X-rays, mammograms, and ultrasound. Ultrasound is a popular medical image type because it is low in cost and easy to diagnose. Ultrasound images can be used for diagnoses in many organs, such as the thyroid gland, the breast, the liver, and the uterus. Ultrasound images can show lesions, nodules, fluid, or calcification. Such abnormalities in ultrasound images can indicate specific diseases.

In the thyroid gland, many abnormalities can be detected using ultrasound images. Because the thyroid gland lies at a shallow depth under the skin, thyroid lesions are quite apparent in ultrasound images. Thus, physicians and radiologists often use ultrasound images as a first level of diagnosis in which they can classify abnormalities accurately.

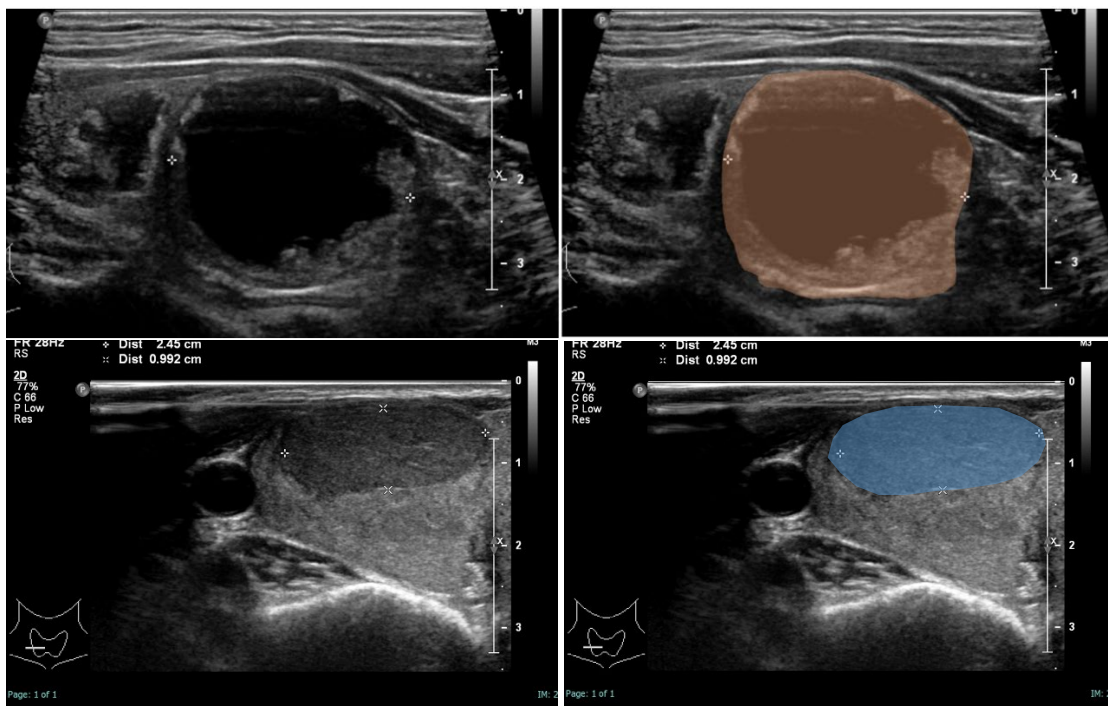
A thyroid nodule is one type of the lesion that can be observed in ultrasound images. The effect of a large nodule is to block other organs, such as the trachea and blood vessels, from working correctly. Nodules can also mutate, becoming malignant. However, once a nodule is detected, physicians can classify the nodule type as benign or malignant using the shape and characteristics of the nodule. Automatic segmentation of thyroid nodules should help physicians diagnose a nodule's type, plan a biopsy, observe growth, and track any change over time. Moreover, less experienced physicians

and trainees can use automatic segmentation tools to learn to diagnose nodules accurately. Clearly, automatic segmentation of thyroid nodules is beneficial.

A sample thyroid nodule segmentation is shown in Figure 1.1.

Figure 1.1

Sample Segmented Thyroid Nodules in Ultrasound Images.



Note. The images on the first row shows a half-cystic nodule (fluid and solid inside) in hyperechoic (the color of solid nodule is bright equal to thyroid gland.) without a clear border (ill-defined shape, difficult to detect the real shape). The image on the second row shows a solid nodule in hypoechoic (the color of solid nodule is darker than thyroid gland.) with a smooth border. The annotations have been drawn manually.

Deep learning is necessary for automatic thyroid nodule segmentation, because deep learning methods typically obtain far superior segmentation results when compared with other algorithms. Many deep learning models can be used for segmentation, such as U-Net (Ronneberger, 2015), Mask R-CNN (He, 2017), DeepLab (Chen, 2017), and YOLACT (Bolya, 2019). These models have been applied to medical images by researchers working on many organs and imaging types.

However, most existing semantic segmentation models utilize supervised learning. They cannot correctly segment structures they have not been exposed to during learning, limiting model applicability. As an alternative, some researchers have studied the applicability of unsupervised learning for the semantic segmentation problem. One such unsupervised learning model is Pix2Pix (Isola, 2017), which is a method for unsupervised learning of image transformations using generative adversarial networks (GANs). Such methods are generative and have more flexibility to assign positions in an image to symptoms.

GANs can produce good results, but it is well-known that GAN learning is unstable. The Pix2Pix model solves the unstable learning problem using supervised learning. However, the Pix2Pix approach is not the best solution when using a strong discriminator, which is ultimately necessary to obtain the best possible generator.

This thesis describes the design of a deep learning model for thyroid nodule segmentation that combines typical supervised learning loss functions for semantic segmentation with an unsupervised learning component using GANs. GANs endow the model with the ability to extract a higher-level understanding of the shapes of nodules in addition to the low-level feedback provided by the supervised loss function. To address the well-known instability of GAN learning, we stabilize the dynamics of the competition between the generator and the discriminator using the techniques of closed-loop automatic control.

1.2 Statement of the Problem

Ultrasound images can be difficult to process because of the indefinite shape, indefinite color, and the effects of brightness and other organs. This makes it difficult to develop an automatic system with high accuracy. In thyroid image processing, incorrect or missing detection of a structure should not be allowed, because it will waste radiologists' time and could even lead to thyroid biopsies in incorrect positions. In the worst case, the cost of a mistake could be a patient's life.

Since an automatic system for segmentation of thyroid nodules would be helpful for radiologists and physicians to diagnose thyroid conditions accurately, thyroid nodule detection is an active area of research in medical imaging. Most of the research focuses on detection and segmentation algorithms with supervised and unsupervised learning.

Research that has used GANs for segmentation in other organs has generally found that GANs can improve accuracy, but the work has not as of yet addressed the fundamental problem of GANs, i.e., unstable training leading to low accuracy.

1.3 Research Scope

This dissertation focuses on the design of a deep learning tool to help physicians with thyroid nodule diagnosis. The design should meet following criteria.

1. The diagnosis should be highly accurate that should be higher than 80% of IoU and accepted by physicians.
2. Inaccuracies or errors in ground truth data that annotators created incorrect, should not inordinately affect model accuracy.
3. The GANs model should be easy to trainable without mode collapse during learning.
4. Model training should exhibit balance between the generator and discriminator, both improving their loss gradually.

1.4 Objectives of the Study

The main objective of this dissertation is to improve upon the state of the art in semantic segmentation of thyroid nodules in ultrasound images. To achieve this main objective, I perform the following steps.

1. Design a GAN model for semantic segmentation of thyroid nodules in ultrasound images.
2. Analyze the issues with ordinary state-of-the-art models such as DeepLabV3+ and U-Net, and use the resulting insights to improve upon the state of the art.
3. Improve GAN learning by stabilizing the dynamics of GANs during learning.

1.5 Contributions

This dissertation presents a new algorithm for GAN learning in which the dynamics are controlled to achieve stability and avoid mode collapse. To the best of our knowledge, this is the first real application of GAN learning dynamics control. More specifically, this dissertation contributes

1. A new GAN based on DeepLabV3+.

2. A new solution to the GAN stability problem based on automatic control, dynamic stabilization, and PID control.
3. Superior results on a thyroid ultrasound dataset.

1.6 Publication

Material in this dissertation is published in *Ultrasound in Medicine and Biology (UMB)*: Kunapinun, Alisa, Dailey, Matthew N., Songsaeng, Dittapong, Parnichkun, Manukid, Keatmanee, Chadaporn, and Ekpanyapong, Mongkol (2022), Improving GAN Learning Dynamics for Thyroid Nodule Segmentation. *Ultrasound in Medicine and Biology*.

1.7 Ethics Approval

According to protocol approval from our institutional review board in ID SIRB Protocol No. 211/2564(IRB2), the informed consent is waived for the retrospective study. The patient's information and ultrasound images were utterly anonymized. Thyroid Nodule Assessment supported the Ultrasound Images Project (NBTC) with the grant number A63-1-(2)-018.

1.8 Organization of the Study

The rest of this dissertation is organized as follows. A literature review on thyroid nodules, ultrasound imaging, and deep learning models is provided in Chapter 2. The methodology, especially the design and mathematics of the model, is provided in Chapter 3. Experimental results and discussion are given in Chapter 4, and the last chapter provides a conclusion and a summary of recommendations.

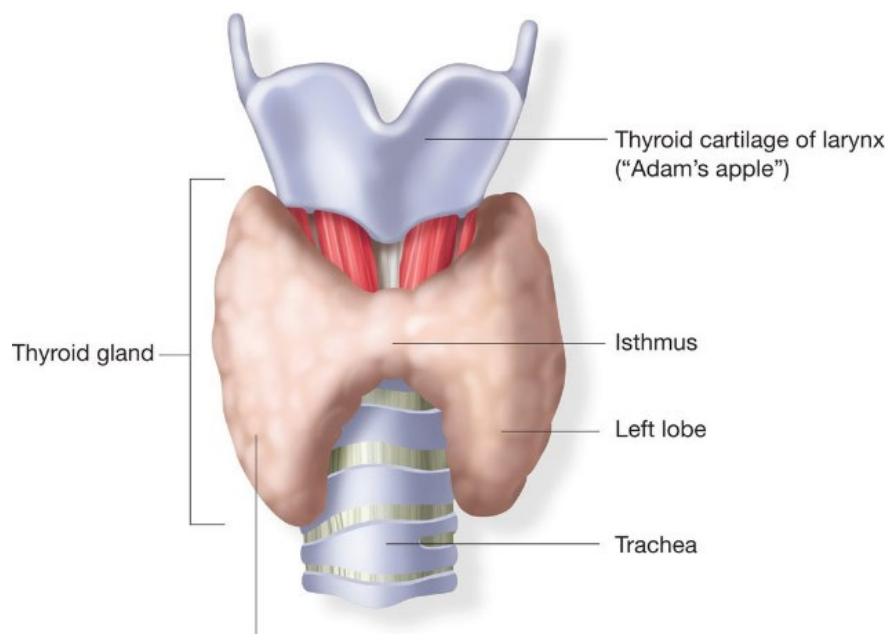
CHAPTER 2

BACKGROUND

This chapter reviews the literature related to the content of the dissertation, starting from the thyroid and thyroid lesions including nodules, the main focus of the thesis. Deep learning is then introduced, which leads to the DeepLabV3+ semantic segmentation model and GANs. Along the way, related methods are explained. These include model structure, loss functions, and model dynamics. The dynamics of ordinary GAN learning are deeply described, and the GAN instability problem is discussed in depth.

Figure 2.1

Thyroid Gland Model



Note: Image from <https://www.toppr.com/ask/en-bd/content/concept/thyroid-gland-201176/>

2.1 Thyroid

The thyroid is a butterfly-shaped gland in front of the neck, shown in Figure 2.1. The thyroid gland covers the trachea, and has two lobes, the left and the right. Some people have another small lobe called the pyramid lobe. The thyroid produces several hormones that play a role in many of the body's functions, such as metabolism, growth,

and body temperature. When disease conditions occur in the thyroid, it affects the entire body's stability.

2.1.1 Ultrasound for Diagnosis

The ultrasound machine is a tool for diagnosis that is useful for discovering and confirming medical disorders. It is used to assist medical procedures and treatments (Chaiyasut, 2017). To aid in diagnosis, ultrasound scans detect tissue and organs from the reflection, refraction, scattering, and absorption of sound waves. The result is then released as a grayscale image. Brighter intensities indicate dense tissue, such as bone and calcium, while darker intensities indicate less-dense tissues, such as blood and water (Bomeli, 2010).

Ultrasound is good for diagnosing thyroid diseases because the thyroid gland is at a shallow depth, making ultrasound images of the thyroid clear and easy to understand (Jones, 2015).

2.1.2 Thyroid Diseases Indicating Ultrasound Examination

Patients with thyroid diseases are usually diagnosed using ultrasound when a physician feels a nodule from the outside or observes the thyroid getting bigger (Radovick, 2010). Another reason for an ultrasound scan is to diagnose patients with a high risk of thyroid cancer, such as those with Hashimoto's thyroiditis or lymphoma, radiation therapy patients, or patients with a malfunction possibly caused by thyroid hormones. Moreover, an ultrasound examination is necessary as a follow up on disease progression or fine needle aspiration (FNA), a procedure for obtaining a cell sample from the target nodule for pathological examination.

2.1.3 Thyroid Nodule and ACR TI-RADS

An ultrasound image of a normal thyroid gland is shown in Figure 2.2. The image shows that the thyroid gland is just under the skin. The right and left lobes cover the trachea. The left and right carotid arteries are located behind the thyroid gland.

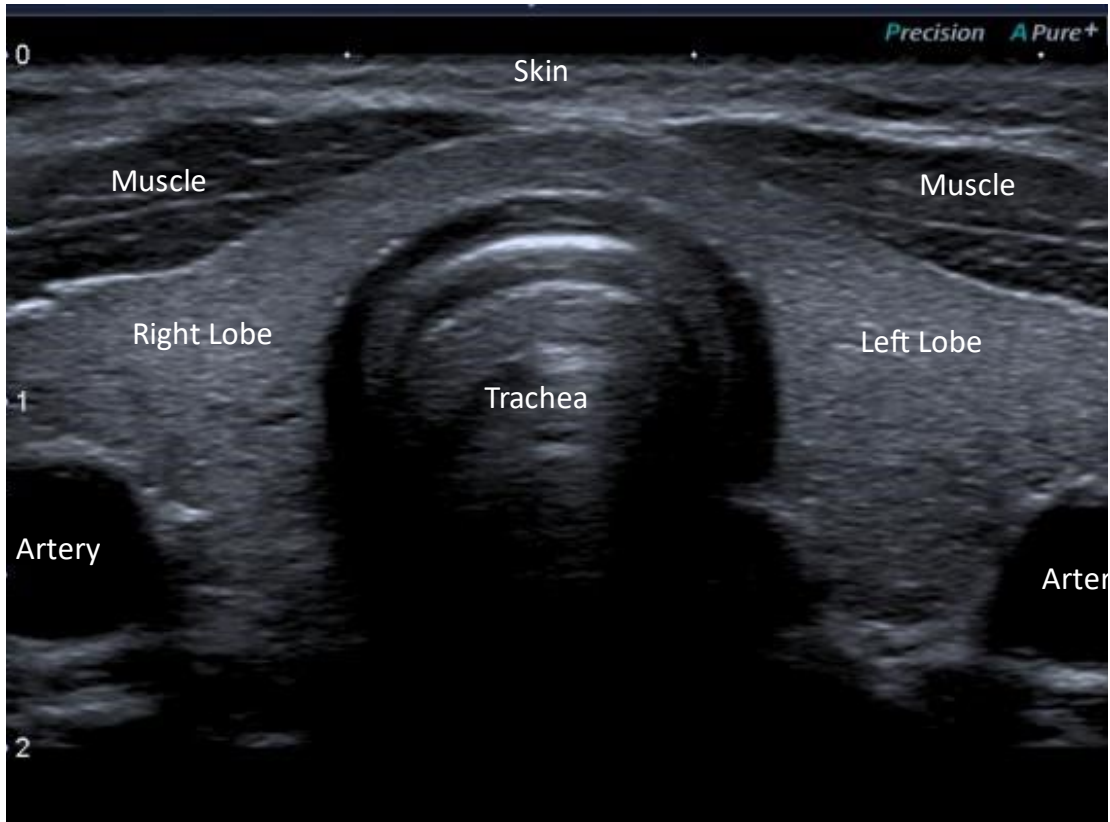
Nodules affect the thyroid gland's shape, and some nodules can impinge on and affect the function of other organs such as the trachea.

One of the most common disease symptoms visible in the thyroid gland is the presence of thyroid nodules. Figure 2.3 shows different thyroid nodules with different

malignancy risks. The malignancy risks can be classified by the nodule's shape, softness, calcium content, and fluid content. Some nodules can contain both benign and malignant material at different locations.

Figure 2.2

Ultrasound Image of a Normal Thyroid Gland

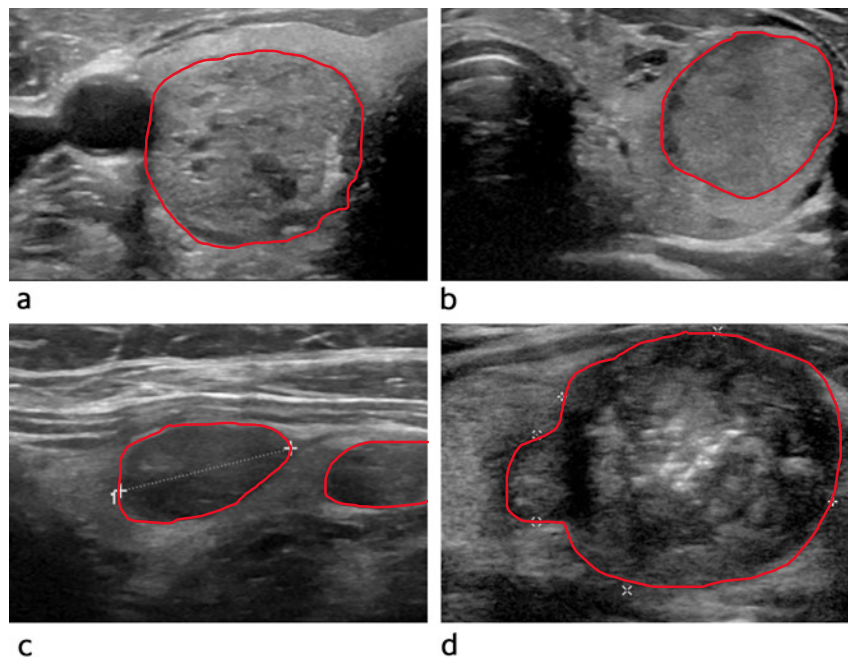


Note: Image from Chaiyasut (2017).

Medical practitioners evaluate benignness or malignancy of a nodule using the Thyroid Imaging Reporting & Data System (TI-RADS) (Tessler, 2018). There have been many revisions of the TI-RADS system for classifying nodule malignancy. The latest revision was released in 2017 with the publication of ACR TI-RADS. Most people develop thyroid nodules as they mature, but most nodules are benign. Some nodules, however, can mutate to a malignancy. When the TI-RADS score is too high, the physician will perform a FNA or thyroid biopsy. The following categories are used for scoring:

Figure 2.3

Example Thyroid Nodules.



Note: The thyroid nodules are examples of different types. a) Oval, well-defined, and spongiform nodule. b) Oval, well-defined, and isoechoic nodule. c) Oval, well-defined, and moderately hypoechoic nodule. d) Hypoechoic nodule with irregular margins and microcalcifications. Reprinted from Tangerud (2020).

1. Composition (Select one from the list):
 - a. Cystic or completely cystic: 0 points
 - b. Spongiform: 0 points
 - c. Mixed cystic and solid: 1 point
 - d. Solid or almost completely solid: 2 points
2. Echogenicity (Select one from the list):
 - a. Anechoic: 0 points
 - b. Hyperechoic or isoechoic: 1 point
 - c. Hypoechoic: 2 points
 - d. Very hypoechoic: 3 points
3. Shape (Select one from the list):
 - a. Wider than tall: 0 points
 - b. Taller than wide: 3 points

4. Margins (Select one from the list):
 - a. Smooth: 0 points
 - b. Ill-defined: 0 points
 - c. Lobulated/irregular: 2 points
 - d. Extra-thyroidal extension: 3 points
5. Echogenic foci (Select all that apply):
 - a. None: 0 points
 - b. Large comet-tail artifact: 0 points
 - c. Macrocalcifications: 1 point
 - d. Peripheral/rim calcifications: 2 points
 - e. Punctate echogenic foci: 3 points

Benign and malignant cases can be partitioned based on the TI-RADs score as follows:

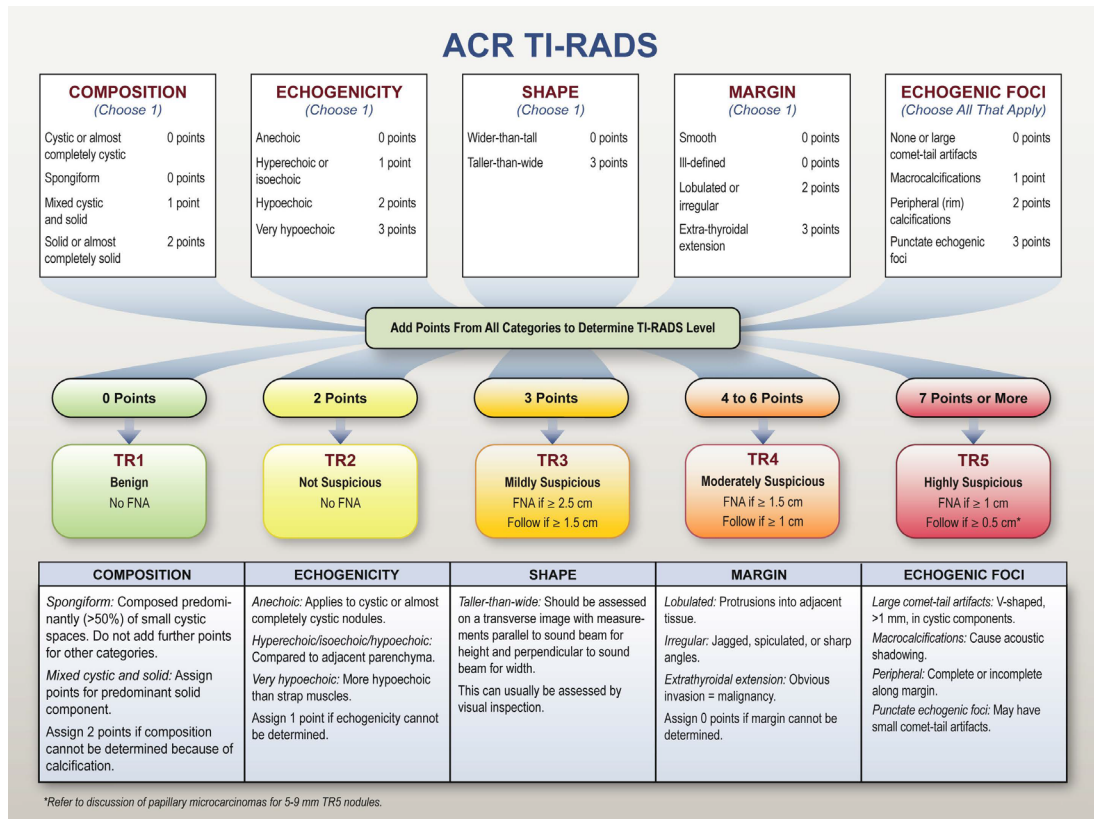
1. TR1 (0 points): entirely benign.
2. TR2 (≤ 2 points): not suspicious, most likely benign.
3. TR3 (3 points): mildly suspicious. This type should be checked with FNA when the nodule size is larger than 25 mm or followed up upon when the size is larger than 15 mm.
4. TR4 (4-6 points): moderately suspicious. This type should be checked FNA when the nodule size is larger than 15 mm or followed up upon when the size is larger than 10 mm. Likely malignancy.
5. TR5 (≥ 7 points): highly suspicious. Almost certainly malignancy. This type should be checked with FNA when the nodule size is larger than 10 mm or followed up upon when the size is larger than 5 mm. Nodules smaller than 5 mm are difficult to check with FNA. If the size is between 5-9 mm, for example a microcarcinoma, FNA is recommended.

Moreover, some defects must be classified as malignancy and checked with FNA when certain foci such as microcalcifications occur.

The ACR TI-RADS classification chart is shown in Figure 2.4.

Figure 2.4

ACR TI-RADS 2017



Note: Image from Tessler (2018).

2.2 Deep Learning in Medical Imaging

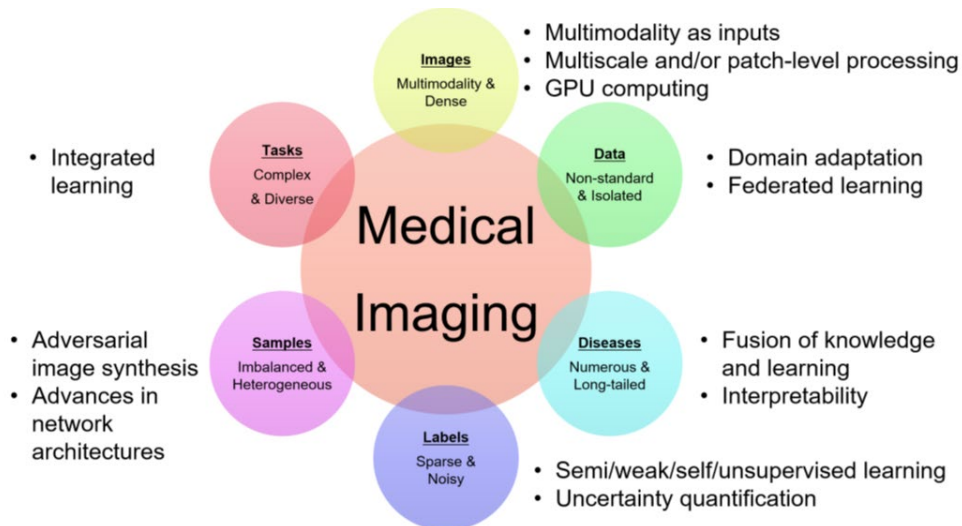
In recent years, the deep learning revolution in computer vision has led to many improvements in medical image processing. Many approaches to support physicians have been designed for different tasks. Medical imaging algorithms can solve many types of problems, such as classification, semantic segmentation, object detection, and 2D/3D synthesis (Zhou, 2021; Pluim, 2019; Ker, 2018; Sánchez, 2017; Greenspan, 2016). Figure 2.5 shows the current research problems and various types of solutions in medical imaging. The main features are the way images are processed, how data are mapped to disease classes, and how learning of multiple tasks is integrated.

Deep learning methods can be adapted to solve many medical imaging problems. For example, common deep learning algorithms such as CNNs or fully connected neural networks can be used directly for image classification. However, some research uses

image segmentation to detect symptoms and classify the disease according to the properties of the detected disease symptoms. This step can improve accuracy in some tasks.

Figure 2.5

Current Medical Imaging and Technological Trends.



Note: The main current research problems in medical imaging and the technological methods for solving them. Reprinted from Zhou (2021).

For problems similar to the example above, one of the most important medical imaging tools is semantic segmentation. Semantic segmentation can be used for analysis, diagnosis, localization, and tracking of anatomical features over time. Semantic segmentation results can be applied very efficiently so long as the pixel-level classification is sufficiently accurate.

2.2.1 Thyroid Nodule Segmentation in Deep Learning

Ultrasound sensors measure tissue density using the reflection, refraction, scattering, and absorption of sound waves. Because the thyroid gland is not very deep in the body, ultrasound is suitable for diagnosing thyroid lesions (Jones and Morgan, 2015). Thyroid nodules typically contain blood, tissue, and calcification (Bomeli, 2010), and these elements absorb varying amounts of sound waves, facilitating their detection in ultrasound images. Researchers have attempted to detect and segment thyroid nodules using a variety of machine learning methods, including deep convolutional neural

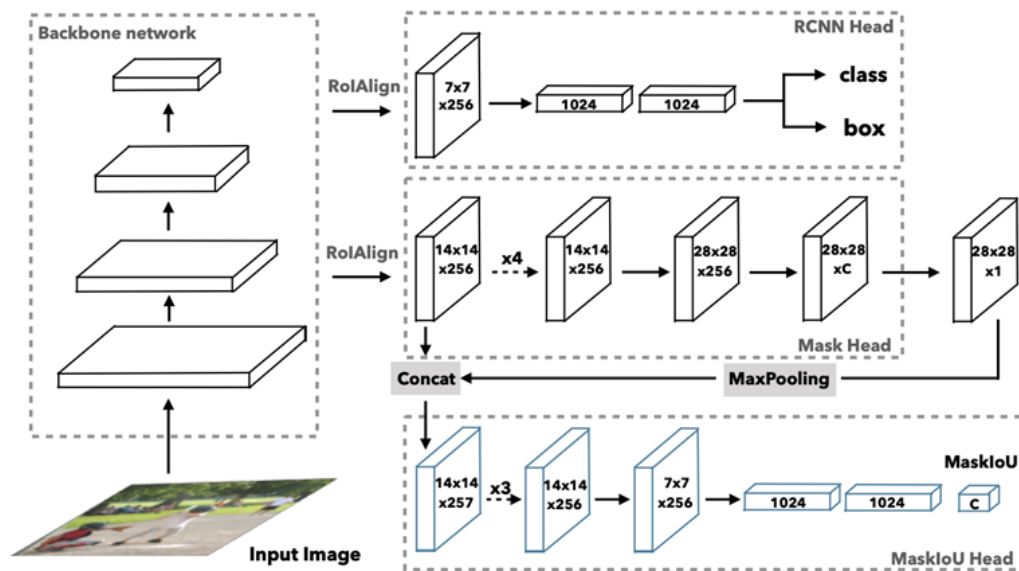
networks (Wang, 2021), Faster R-CNN for segmentation (Yu, 2020) and detection (Yao, 2020), and DeepLabV3+ (Ye, 2020). Almost all recent state of the art research finds that deep learning improves accuracy.

2.3 Deep Convolutional Neural Networks (CNNs) for Segmentation

Deep CNNs were originally developed for classification and prediction in supervised learning problems. Early CNNs such as InceptionNet (Szegedy, 2015, 2016) and ResNet (He, 2016) have been used as backbones⁸⁴ for detection and segmentation models such as Fast R-CNN (Yu, 2020), FCN (Long, 2015), and YOLO (Redmon, 2018; Bochkovskiy, 2020). Normally, the backbones for detection and segmentation are extracted from a classifier and used to transform images to feature maps that aid detection or segmentation process rather than classification.

Figure 2.6

Mask R-CNN Model.



Note: The mask R-CNN architecture (He, 2017) combines a backbone model from which feature maps are extracted to be multilayers in a pyramid shape. After feature maps are output from the backbone, the model continues with further tasks such as region proposals, classification, and segmentation.

Today's detection models are very accurate when there are many objects in an image and the precise size or boundary contour of the object is not important. However, pixel-

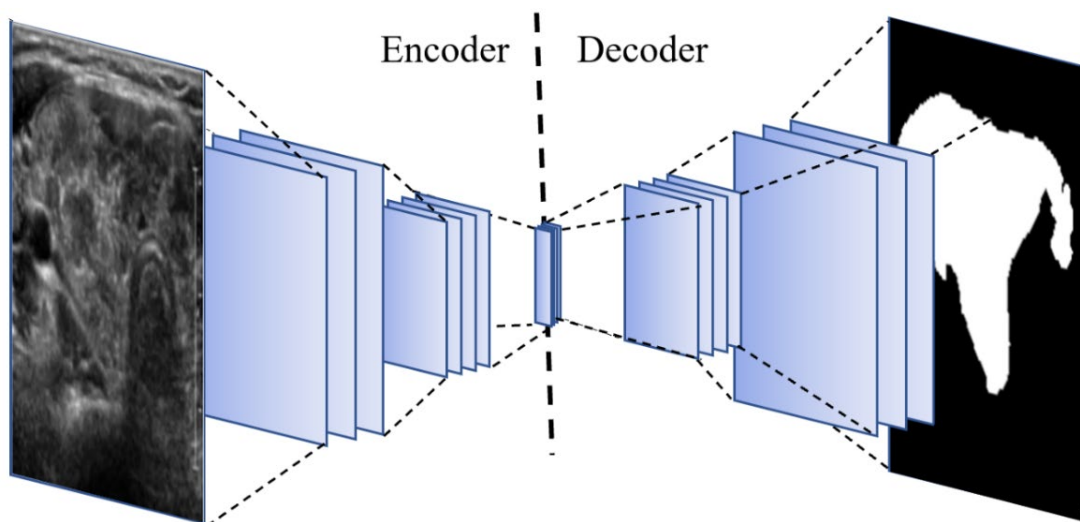
level classification with a detector is inaccurate, because background pixels within the detected boundary box are not removed. Segmentation models address this problem in a way similar to the detection solution, but instead of outputting a detected object's position and bounding box, we output the object's shape, in the form of an object class for each pixel in the region.

In 2017, Mask R-CNN (He, 2017), the first instance segmentation that was sufficiently efficient and accurate for practical use, was released. It uses the structure of Faster R-CNNs, as well as feature pyramid networks, to scan for important features and accurately segment objects. The model has been used in many applications, and medical imaging is no exception.

Building on feature extraction backbones and feature pyramids, deep autoencoder networks have also been adapted to the problem of semantic segmentation (Bank, 2020). Autoencoder-structured segmentation models combine an encoder and a decoder, as shown in Figure 2.7. Many of the modern segmentation models are designed based on autoencoder structures, including U-Net (Badrinarayanan, 2017), SegNet (Ronneberger, 2015), and DeepLab (Chen, 2017, 2014).

Figure 2.7

Conceptual Design of Autoencoder Networks.



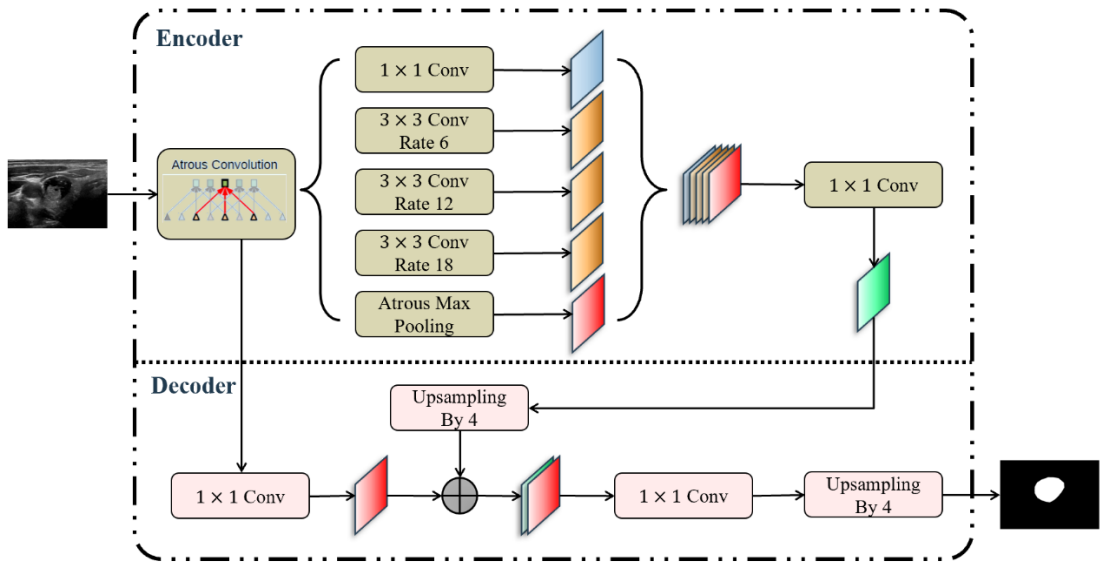
Note: Autoencoder networks compound two stages: an encoder and a decoder.

2.4 DeepLabV3+

Google’s Deeplab segmentation network was introduced in 2014 (Chen, 2014). It uses atrous convolution (Chen, 2017) in the encoder see Section 2.4.1 for a discussion of atrous convolution. Based on this and several other enhancements, the DeepLab model has been updated several times and is now in version 3+, i.e., DeepLabV3+ (Chen, 2018). This version has proven very accurate, achieving first place in the MICCAI 2020 TN-SCUI challenge (Yu, 2020). DeepLabV3+’s architecture is shown in Figure 2.8.

Figure 2.8

DeepLabV3+ Architecture.



2.4.1 Atrous Convolution

Atrous convolution, exemplified in Figure 2.9, is a general design for convolutional neural networks especially supportive of segmentation. An atrous convolution uses a wide-spread convolution kernel with “holes” rather than an increased kernel size. An atrous convolution, in comparison with an ordinary convolution, increases receptive field size without changing feature map resolution or the number of parameters required. The concept is shown schematically in Figure 2.10.

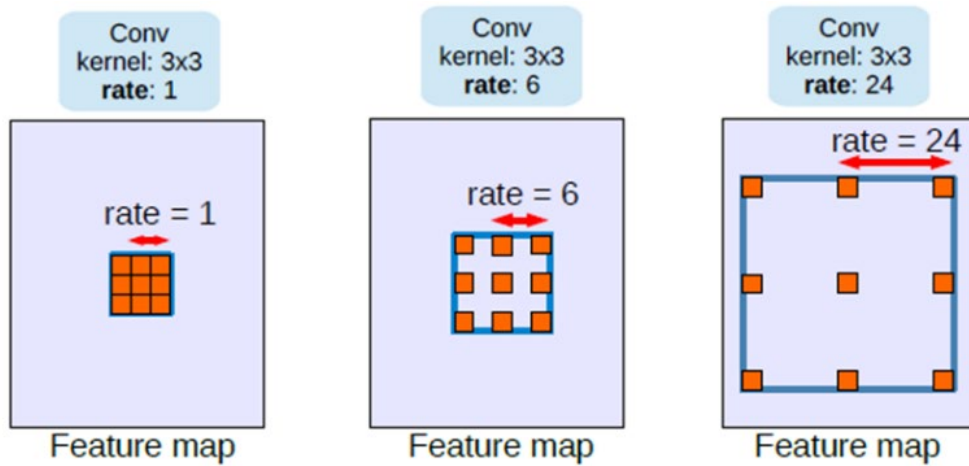
In one dimension, the atrous convolution operation can be written as

$$y[i] = \sum_k x[i + r \cdot k]w[k], \quad (1)$$

where x is the tensor input, y is the tensor output, i is the position of input and output tensor, r is the “jumping rate” in the input tensor, w is the weight tensor, and k is the position of weight tensor.

Figure 2.9

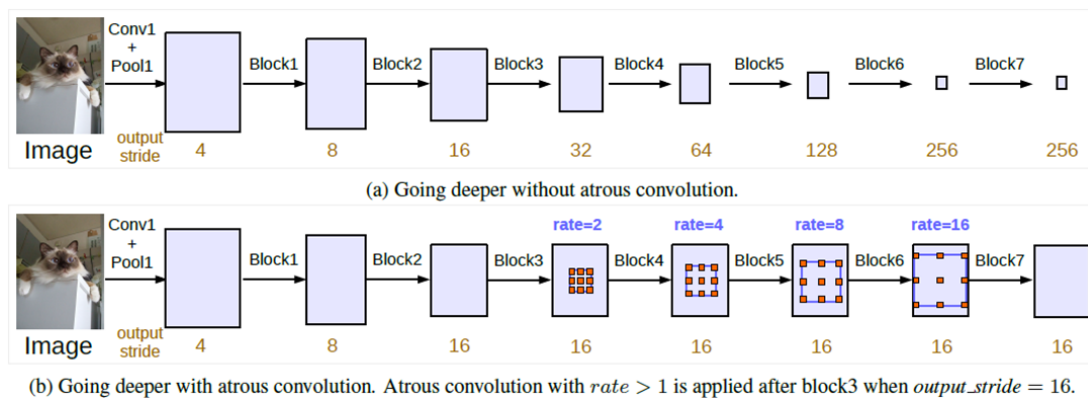
Atrous Convolution.



Note: Image from <https://towardsdatascience.com/review-deeplabv3-atrous-convolution-semantic-segmentation-6d818bfd1d74>.

Figure 2.10

Comparison of Atrous and Ordinary Convolution.

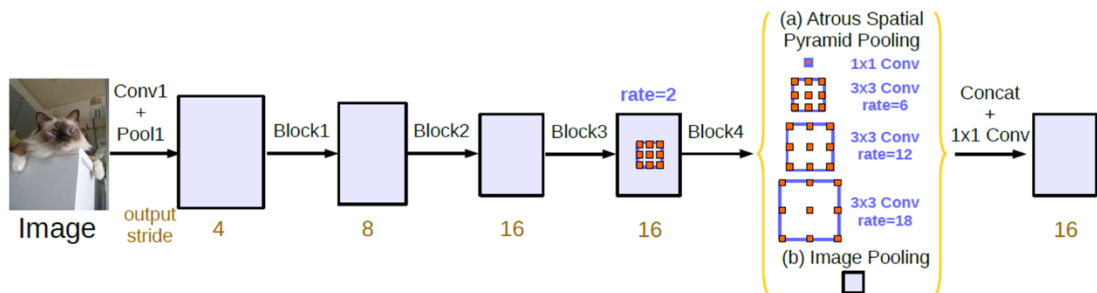


Note: a) Without atrous convolution, feature map size generally decreases when convolution stride increases. b) Atrous convolution on the other hand, typically preserves feature map size while increasing receptive field size. Image from <https://towardsdatascience.com/review-deeplabv3-atrous-convolution-semantic-segmentation-6d818bfd1d74>

The DeepLabV3+ model uses atrous convolution in its convolutional and pooling layers. The pooling layers work with many tensor shapes, from small sizes to the very large sizes in a pyramid. This structure is called “Atrous Spatial Pyramid Pooling,” as shown in Figure 2.11.

Figure 2.11

DeepLabV3+ Encoder Network with Atrous Spatial Pyramid Pooling



Note: The network contains multiple atrous convolutional layers with a jumping rate of 2. Atrous spatial pyramid pooling is applied at the last layer of the network to obtain a multiscale pyramid. Image from <https://towardsdatascience.com/review-deeplabv3-atrous-convolution-semantic-segmentation-6d818bfd1d74>.

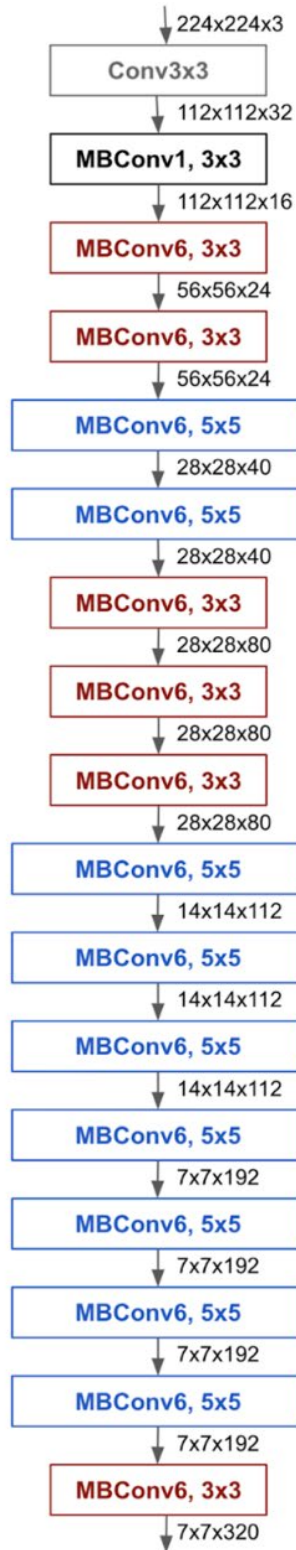
2.4.2 DeeplabV3+ Backbone

The backbone of DeepLabV3+ is EfficientNet b7, as shown in Figure 2.12. In each block of the network, a bottleneck block inspired by MobileNet called *MBConv* is used.

Figure 2.13 shows a MBconv block in detail.

Figure 2.12

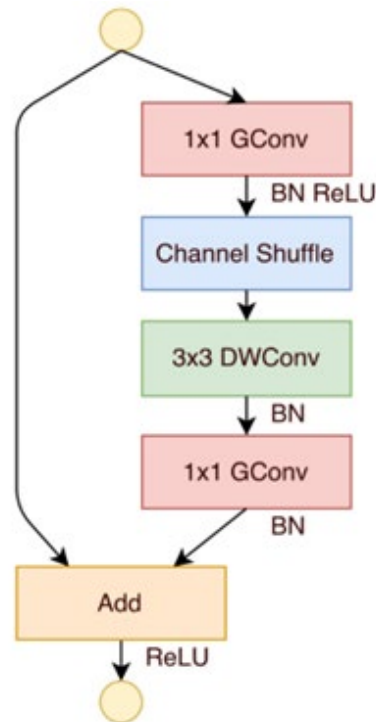
EfficientNet b7 Network.



Note: The EfficientNet b7 network composes many MBConv blocks.

Figure 2.13

MBConv Block.



Note: The MBConv block, the main block in MobileNet, is a residual bottleneck block. The module block is designed concept based on ResNet module. The first layer do 1×1 convolution and shuffle the channel tensors, and do another 3×3 depth wise convolution and 1×1 convolution, then it add the parameters with the previous tensor.

2.4.3 Implementation of DeepLabV3+ for Thyroid Nodule Ultrasound Images

DeeplabV3+ has been used for thyroid nodule detection in ultrasound images in a challenge called the MICCAI 2020 TN-SCUI challenge, and it ranked first. The system has two DeeplabV3+ stages, stage 1 and stage 2. The first stage network uses the entire image at 256×256 resolution to find the nodule roughly and then crops the image around the rough detection to find the actual nodule shape. The overall structure of the solution is shown in Figure 2.14.

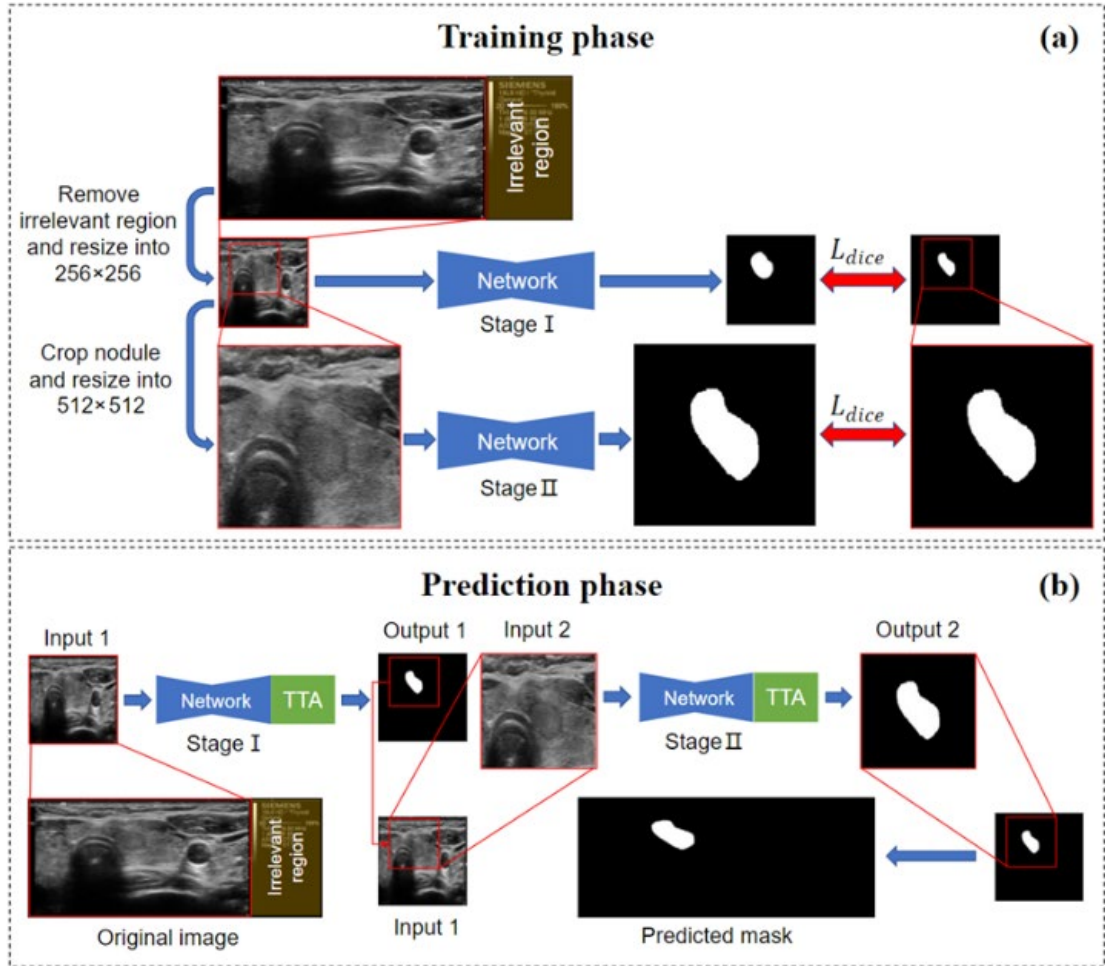
2.5 Loss Function for Segmentation Networks

All deep learning models must learn from feedback about correct and incorrect predictions. In supervised learning, we can compare prediction results with the desired

target or ground truth and fired have been made. The function that penalizes incorrect behavior and encourages correct behavior is the loss function.

Figure 2.14

Overall Process of DeepLabV3+ 1st Place Solution in MICCAI 2020 TN-SCUI Challenge. Reprinted from <https://github.com/WAMAWAMA/TNSCUI2020-SegRank1st>



2.5.1 Binary Cross-Entropy Loss Function

For semantic segmentation with one class of interest, we typically use the ordinary binary cross-entropy loss function (\mathcal{L}_B) on a per-pixel basis. This loss function is simple and it can calculate the overall of error. However, it can make the model mode collapse easily. Assume an input image \mathbf{x} of size $M \times N$ associated with a binary ground truth mask \mathbf{y} , and assume a segmentation model $g(\cdot)$ that outputs a mask prediction with the

same size as its input. The binary cross-entropy loss function for one image (Jadon, 2020) can be written as

$$\mathcal{L}_B(\mathbf{y}, g(\mathbf{x})) = -\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [\mathbf{y}_{i,j} \log g(\mathbf{x})_{i,j} + (1 - \mathbf{y}_{i,j}) \log(1 - g(\mathbf{x})_{i,j})] , \quad (2)$$

2.5.2 Soft Dice Loss Function

Another loss function often used in computer vision is the dice loss function (Jadon, 2020). It is used to measure the similarity of two images or image patches. The soft dice loss function (\mathcal{L}_S) can be written as

$$\mathcal{L}_S(\mathbf{y}, g(\mathbf{x})) = -\frac{2 \sum_i \sum_j \mathbf{y}_{i,j} g(\mathbf{x})_{i,j}}{\sum_i \sum_j \mathbf{y}_{i,j}^2 + \sum_i \sum_j g^2(\mathbf{x})_{i,j} + \epsilon} , \quad (3)$$

where ϵ is a small positive value to prevent divide by zero errors.

2.5.3 Lovasz-Softmax Function

Although soft dice loss is efficient, it is known to be difficult to optimize. The Lovasz-softmax function is designed to solve this optimization problem. The DeepLab implementation has a specific implementation of the Lovasz-softmax loss function, which we follow here. Assume p is the number of pixels in the image size $M \times N$. For each $i \in 1, \dots, p$, the binary cross-entropy loss for pixel i can be written as

$$e_i = G(\mathbf{x})_i \mathbf{y}_i + (1 - G(\mathbf{x})_i)(1 - \mathbf{y}_i) \quad (4)$$

We order the e_i in decreasing order as $\pi_1, \pi_2, \dots, \pi_p$, i.e., $\pi_1 = \operatorname{argmax}_i e_i$, and $\pi_p = \operatorname{argmin}_i e_i$. The e_i in decreasing order can help the system which pixel data is more important to consider. We can calculate the cumulative summation as

$$\Pi_i = \sum_{j=1}^i e_{\pi_j} \quad (5)$$

And the corresponding cumulative loss for the negative class can be written as

$$\Phi_i = \sum_{j=1}^i (1 - e_{\pi_j}) \quad (6)$$

Finally, the Lovasz-softmax function in DeepLab can be written as

$$\Delta_i = 1 - \frac{\Pi_p - \Pi_i}{\Pi_p - \Phi_i} \quad (7)$$

Because the loss functions mentioned above have different advantages and disadvantages in different situations, we select all three for our experiments by sum all loss output $\mathcal{L} = \mathcal{L}_B + \mathcal{L}_S + \mathcal{L}_L$.

2.6 Measuring Segmentation Quality

In this dissertation, we use IoU (intersection over union) between predicted and ground truth segmentations as the primary criterion. Besides IoU, it is useful to understand whether a model is biased toward positive or negative classification in a binary classifier or toward one or more classes in a multinomial classifier. To measure bias in a binary task, we need to compare the ground truth masks (annotated manually) and the predicted masks (annotated by the model) and partition mask pixels into four groups:

- True-positive (TP): The position is selected by the predictor, and the decision is correct.
- True-negative (TN): The position is not selected, and the decision is correct.
- False-positive (FP): The position is selected by the predictor, but the decision is incorrect.
- False-negative (FN): The position is not selected by the predictor but is incorrect.

We use five equations to measure segmentation accuracy: IoU, accuracy, sensitivity, specificity, and precision. IoU can be written as

$$IoU = \frac{TP}{TP+FP+FN} \quad . \quad (8)$$

Accuracy (Acc) can be written as

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad . \quad (9)$$

Sensitivity (Sen), also known as recall, can be written as

$$Sen = \frac{TP}{TP+FN} \quad . \quad (10)$$

Specificity (Spc) can be written as

$$Spc = \frac{TN}{FP+TN} \quad . \quad (11)$$

Precision (Prc) can be written as

$$Prc = \frac{TP}{TP+FP} \quad (12)$$

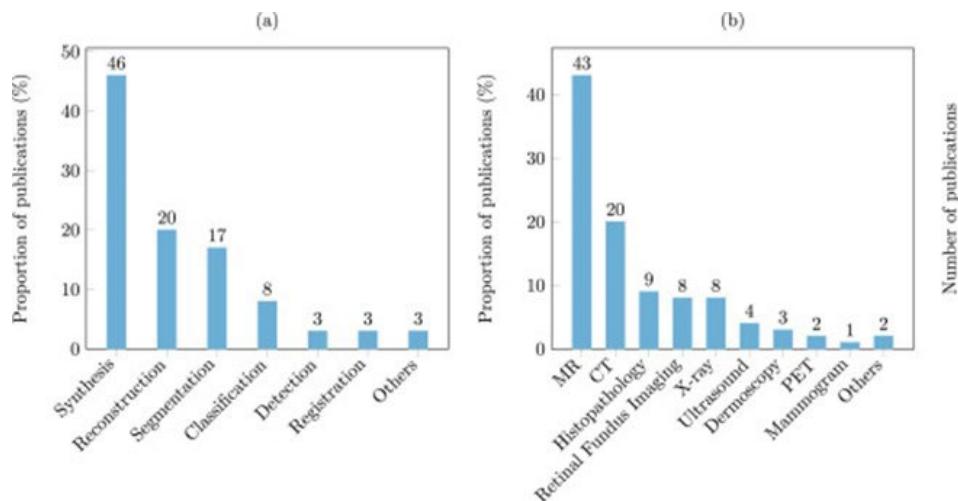
2.7 Generative Adversarial Networks (GANs)

GANs are modern deep learning architectures used for unsupervised learning. GANs were initially applied to image synthesis. Today, GANs are being used in many applications. Xin et al. (2019) review and classify GAN applications used in medicine into five categories:

- Synthesis
- Reconstruction
- Classification
- Detection
- Registration

Figure 2.15

Distribution of Medical Imaging Research Using GANs.



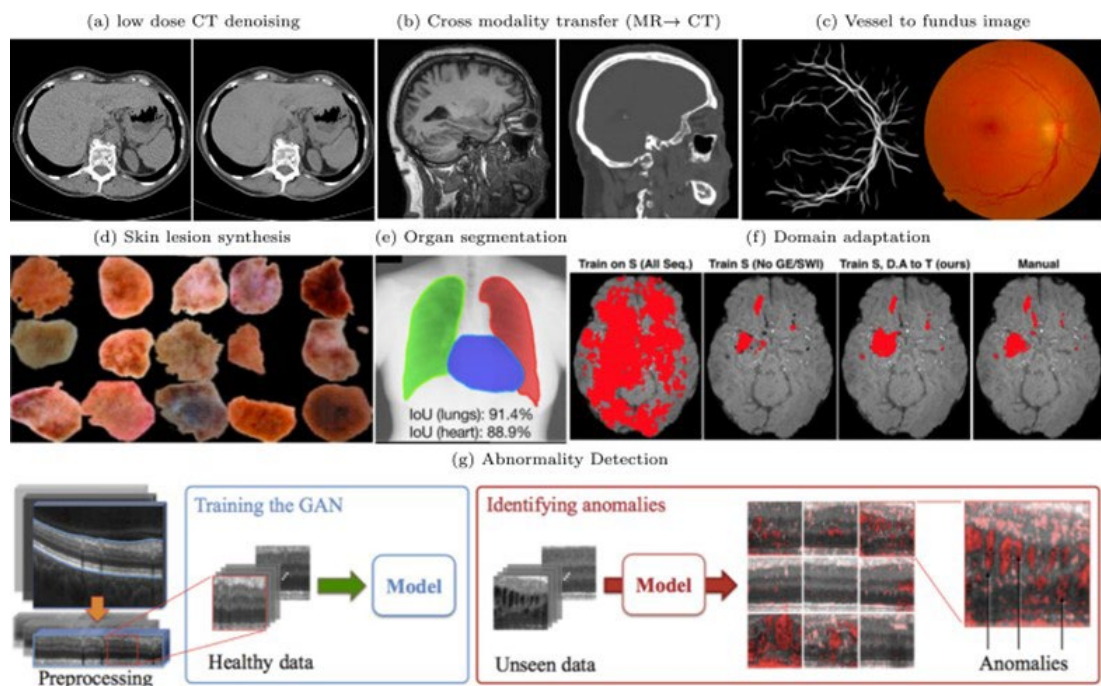
Note: a) Distribution of GAN-related publications in different applications. b) Distribution of GAN-related publications using GANs in different types of images. Reprinted from Xin et al. (2019).

In 2019, the most popular GAN application was image synthesis (Xin et al., 2019). Among these research topics, four research topics used GANs for ultrasound images.

However, until now, only one research paper reports the use of GANs with ultrasound images for segmentation applications (Dou, 2018). A summary of GAN implementations is shown in Figure 2.15, and example applications using GANs in medical imaging are shown in Figure 2.16.

Figure 2.16

Applications Using GANs in Medical Imaging.



Note: Reprinted from Xin et al. (2019)

2.7.1 GANs Architecture

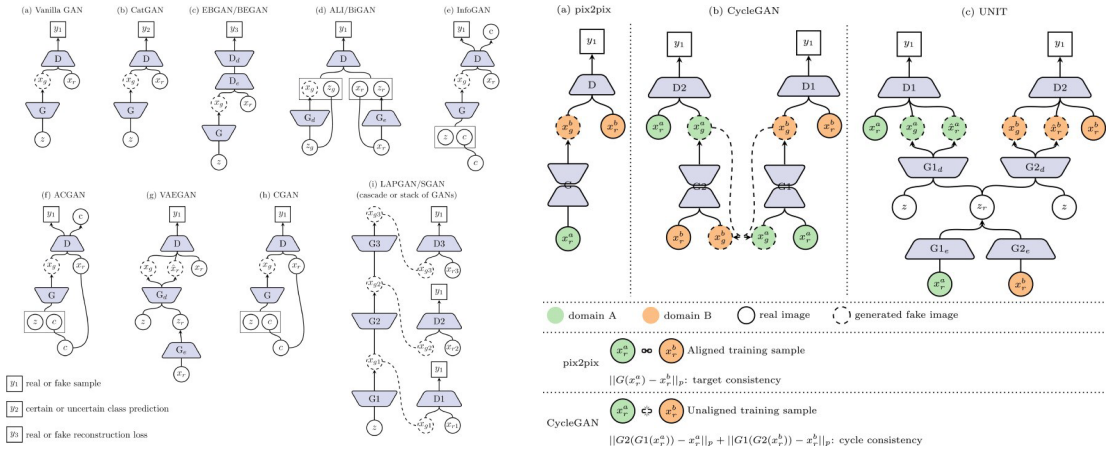
GANs have gone through many architectural iterations, including vanilla GANs, Pix2Pix, CycleGANs, and VAE GANs. However, all GAN designs must contain at least two main networks: a generator model G and a discriminator model D . The generator performs a target task such as image synthesis. In contrast, the discriminator is responsible for classifying samples as real or fake (generated).

The two networks are competitive in that the discriminator attacks the generator, forcing it to improve in order to “fool” the discriminator. Each GAN network architecture is used in different applications. Some GANs such as vanilla GAN

(Goodfellow, 2014) can learn to generate images that look real. Some GANs, such as CycleGANs (Zhu, 2017), can learn to map images between two domains, and some GANs such as Pix2Pix (Isola, 2017) can convert an image in one style into another style. Various GAN designs are shown in Figure 2.17.

Figure 2.17

Different GAN Designs. Reprinted from Xin et al. (2019).



2.7.2 Dynamics of Loss Functions for GANs

GAN model training requires evaluation of a loss function. Ordinary GANs and some specialized GAN architectures use two combinations of loss functions. The discriminator loss function encourages it to maximize classification accuracy in terms of real and fake, while the generator loss function encourages it to generate images that look real. Thus, learning involves iteration, leading to dynamics

$$\max_{\phi} \mathcal{L}_D(\phi; \theta), \quad (13)$$

$$\max_{\theta} \mathcal{L}_G(\theta; \phi), \quad (14)$$

where θ is the parameter vector for G function and ϕ is the parameter vector for D . Equations 13 and 14 can be rewritten as

$$\mathcal{L}_D(\phi; \theta) = \mathbb{E}_{\mathbf{y}}[\log(D(\mathbf{y}; \phi))] + \mathbb{E}_{G(\mathbf{x})}[\log(1 - D(G(\mathbf{x}; \theta); \phi))], \quad (15)$$

$$\mathcal{L}_G(\theta; \phi) = \mathbb{E}_{G(\mathbf{x})}[\log(D(G(\mathbf{x}; \theta); \phi))]. \quad (16)$$

where \mathbb{E}_y is the probability output of discriminator from real image, and $\mathbb{E}_{G(x)}$ is the probability output of discriminator from fake image.

2.7.3 Pix2Pix Model in Image Segmentation

According to Xin et al. (2019), the few researchers who have used GANs for semantic segmentation have primarily used Pix2Pix for the task. In the case of Pix2Pix-based segmentation models, the conceptual design of the generator network is the same as that for any other semantic segmentation network. However, instead of classifying masks as real or fake, the discriminator classifies the mask overlay output by the generator as correct or incorrect, with the original image as a second input. Thus, the discriminator sees both the original image and the generated mask, and then it can determine whether the overlay of the generated mask is correct considering the target object in the original image.

The discriminator's objective in a Pix2Pix GAN is to maximize

$$\mathcal{L}_D(\phi; \theta) = \mathbb{E}_{\mathbf{x}, \mathbf{y}}[\log(D(\mathbf{y}; \phi))] + \mathbb{E}_{\mathbf{x}, G(\mathbf{x})}[\log(1 - D(G(\mathbf{x}; \theta), \mathbf{x}; \phi))] , \quad (17)$$

while the generator's objective is to minimize the same loss ignoring terms independent of θ :

$$\mathcal{L}_G(\theta; \phi) = \mathbb{E}_{\mathbf{x}, G(\mathbf{x})}[\log(D(G(\mathbf{x}; \theta), \mathbf{x}; \phi))] + \mathcal{L}_1(\theta). \quad (18)$$

\mathcal{L}_1 represents the loss function for supervised learning, depending on the application. For semantic segmentation with Pix2Pix, $\mathcal{L}_1 = \mathcal{L}_B$, i.e., the binary cross-entropy loss function introduced earlier.

2.8 GAN Stabilization

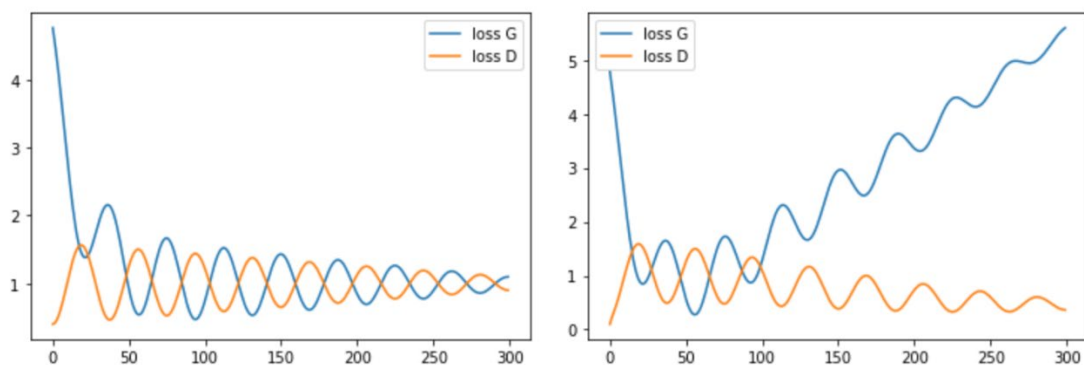
The loss \mathcal{L}_G of the generator and the loss \mathcal{L}_D of the discriminator should both change together for effective optimization. If the discriminator becomes too confident too early, the gradient signal to G will be very small. This type of vanishing gradient can cause training to be imbalanced, with the generator unable to learn. In this situation, the generator tends to output only one image. This situation is called *mode collapse*. Figure 2.18 shows stylized loss profiles of balanced training without mode collapse and imbalanced training with mode collapse.

There are three ways to balance GAN training to address mode collapse:

1. Reduce the discriminator's complexity
2. Add a supervised loss function for the generator network, turning the method into a semi-supervised method (Salimans, 2016)
3. Balance the learning dynamics

Figure 2.18

Characteristic Loss Function Dynamics for Stable and Unstable GAN Learning.



Note: Left: Stable GAN learning between the generator and the discriminator, in which the discriminator and generator compete effectively. The loss of the generator converges to a low value. Right: Unstable GAN learning between the generator and the imbalanced discriminator loss with mode collapse. Usually, the imbalance is caused by a discriminator that learns more quickly than the generator, leading to a loss in the discriminator that is very low and a loss in the generator that is very high.

The first method, reducing the discriminator's complexity, is the worst solution among the three, because weak discriminators can lead to inaccurate results. A generator will only learn well if its discriminator is robust, and conversely, a weak discriminator will provide an insufficient challenge to the generator for it to improve.

The second method, utilizing semi-supervised learning, is another way to avoid mode collapse, because supervised learning can prevent the generator from outputting results far away from the target. This method also increases generator training speed, because semi-supervised learning is faster than purely unsupervised learning.

The third approach, balancing the learning dynamics, involves finding equilibrium points and improving convergence to the equilibrium using algorithms such as Wasserstein GANs (Bottou, 2017). Methods based on control theory for stabilization have also led to promising results (Yadav, 2017).

The stochastic gradient descent method used in deep learning updates a set of weights using a loss function:

$$\theta^{(i+1)} \leftarrow \theta^{(i)} - \alpha \nabla_{\theta} \mathcal{L}(\theta^{(i)}). \quad (19)$$

From Equation 14, changing the learning rate α or scaling \mathcal{L} can change the learning speed. Thus, we can control the magnitude of the components of the loss function \mathcal{L} to have more appropriate relative contributions. In this approach, we can balance the rate of learning in G and D (Mescheder, 2018).

We assume different log sigmoid functions $h_1(\cdot)$, $h_2(\cdot)$, and $h_3(\cdot)$:

$$h_1(x) = \log(\sigma(x)) \quad (20)$$

$$h_2(x) = \log(1 - \sigma(x)) \quad (21)$$

$$h_3(x) = -\log(1 - \sigma(x)) \quad (22)$$

The derivatives and second derivatives of the log sigmoid functions are:

$$h_1(x) = \log(\sigma(x)) \quad h'_1(x) = 1 - \sigma(x) \quad h''_1(x) = -\sigma(1 - \sigma(x)) \quad (23)$$

$$h_2(x) = \log(1 - \sigma(x)) \quad h'_2(x) = -\sigma(x) \quad h''_2(x) = -\sigma(1 - \sigma(x)) \quad (24)$$

$$h_3(x) = -\log(1 - \sigma(x)) \quad h'_3(x) = \sigma(x) \quad h''_3(x) = \sigma(1 - \sigma(x)) \quad (25)$$

The discriminator's dynamics involve the iterative maximization of \mathcal{L}_D . Assuming D has a linear output (moving the sigmoid into $h_1(\cdot)$, $h_2(\cdot)$, and $h_3(\cdot)$) with p a distribution over \mathbf{y} , p_G a distribution over $G(\mathbf{x})$, and p_x a distribution over \mathbf{x} , we convert the discriminator and generator dynamics of ordinary GANs from Equations 15 and 16 as:

$$\mathcal{L}_D(\phi; \theta) = \mathbb{E}_{p(\mathbf{y})}[\mathbf{h}_1(D(\mathbf{y}; \phi))] + \mathbb{E}_{p_G(\hat{\mathbf{y}})}[\mathbf{h}_2(D(\hat{\mathbf{y}}; \phi))], \quad (26)$$

$$\mathcal{L}_G(\theta; \phi) = \mathbb{E}_{p_x(\mathbf{x})}[\mathbf{h}_3(D(G(\mathbf{x}; \theta); \phi))]. \quad (27)$$

In these equations, θ is the parameter vector for G , and ϕ is the parameter vector for D . $D(\mathbf{y}; \phi)$ is the result of applying the discriminator to a true image, and $D(G(\mathbf{x}; \theta); \phi)$ is the result of applying the discriminator to an output of the generator.

Local stability for the discriminator and generator is achieved when the gradient of both $\mathcal{L}_D(\phi; \theta)$ and $\mathcal{L}_G(\theta; \phi)$ are equal to 0. Letting $v(\phi, \theta)$ be the gradient vector field for the discriminator and generator as learning occurs over time, we write

$$v(\phi, \theta) = \begin{bmatrix} \nabla_{\phi} \mathcal{L}_D(\phi; \theta) \\ \nabla_{\theta} \mathcal{L}_G(\theta; \phi) \end{bmatrix}, \quad (28)$$

where the gradient of $\mathcal{L}_D(\phi; \theta)$ with respect to ϕ is given by

$$\nabla_{\phi} \mathcal{L}_D(\phi; \theta) = \mathbb{E}_{p(\mathbf{y})}[\mathbf{h}'_1(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi)] + \mathbb{E}_{p_G(\mathbf{y})}[\mathbf{h}'_2(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi)], \quad (29)$$

and the gradient of $\mathcal{L}_G(\theta; \phi)$ with respect to θ is

$$\nabla_{\theta} \mathcal{L}_G(\theta; \phi) = \mathbb{E}_{p_x(\mathbf{x})}[\mathbf{h}_3(D(G(\mathbf{x}; \theta); \phi)) [\nabla_{\theta} D(G(\mathbf{x}; \theta); \phi)]^T \cdot \nabla_{G(\mathbf{x})} D(G(\mathbf{x}; \theta); \phi)]. \quad (30)$$

These gradients can easily be derived using the chain rule.

At the equilibrium point $(\phi^*; \theta^*)$, at which the generator perfectly matches the data distribution p , we have, throughout the support of p , $p_G(\mathbf{y}) = p(\mathbf{y})$, $D(G(\mathbf{x}; \theta^*); \phi^*) = 0$, $\nabla_{\mathbf{y}} D(\mathbf{y}; \phi^*) = 0$, and $\nabla_{\mathbf{y}}^2 D(\mathbf{y}; \phi^*) = 0$. The first statement means that G outputs \mathbf{y} vectors with a distribution indistinguishable from the data distribution. The second statement means that the discriminator cannot determine whether its input is from G or sampled from the data distribution. The third statement means that D 's output around the support of the data distribution is smooth, and the fourth statement means that D 's output around the support of the data distribution is flat. Let the Jacobian of the gradient vector field $v(\phi, \theta)$ at (ϕ^*, θ^*) be

$$v'(\phi^*, \theta^*) = \begin{bmatrix} J_{DD} & J_{DG} \\ J_{GD} & J_{GG} \end{bmatrix}, \quad (31)$$

The terms of this Jacobian are given by

$$J_{DD} = -\frac{1}{2} \mathbb{E}_{p(\mathbf{y})} [\nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T] \Big|_{\phi=\phi^*}, \quad (32)$$

$$J_{DG} = -\frac{1}{2} \nabla_{\theta} \mathbb{E}_{p_G(\mathbf{y})} [\nabla_{\phi} D(\mathbf{y}; \phi)] \Big|_{\phi=\phi^*, \theta=\theta^*}, \quad (33)$$

$$J_{GD} = -J_{DG}^T, \quad (34)$$

$$J_{GG} = 0, \quad (35)$$

Linearizing Equation 31 around the (ϕ^*, θ^*) , we obtain

$$\begin{bmatrix} \nabla_{\phi} \mathcal{L}_D(\phi; \theta) \\ \nabla_{\theta} \mathcal{L}_G(\theta; \phi) \end{bmatrix} \approx \begin{bmatrix} J_{DD} & J_{DG} \\ -J_{DG}^T & J_{GG} \end{bmatrix} \begin{bmatrix} \phi - \phi^* \\ \theta - \theta^* \end{bmatrix}, \quad (36)$$

where ϕ^*, θ^* are constant vectors that depend on the training set input $\{\mathbf{X}, \mathbf{Y}\}$.

In order to simplify the analysis of the dynamics, consider the case of any one parameter ϕ of D and any one parameter θ of G . Then J_{DD} and J_{DG} are scalars. According to the recipe of classical control theory, we transform the dynamics of Equation 36 to the frequency domain using the Laplace transform in order to understand the system's stability around the equilibrium point.

For ϕ , we obtain

$$s\Phi(s) = J_{DD}(\Phi(s) - \Phi^*(s)) + J_{DG}(\Theta(s) - \Theta^*(s)), \quad (37)$$

$$\Phi(s) = \frac{-J_{DD}}{s-J_{DD}} \Phi^*(s) + \frac{-J_{DG}}{s-J_{DD}} (\Theta(s) - \Theta^*(s)), \quad (38)$$

The dependency of Φ on Θ can be eliminated using $\nabla_{\theta} \mathcal{L}_G$, from which we obtain

$$\Theta(s) = \frac{-J_{DG}}{s} (\Phi(s) - \Phi^*(s)), \quad (39)$$

Substituting Equation 38 into Equation 39, we obtain

$$\Phi(s) = \frac{-sJ_{DG}}{s^2-sJ_{DD}+J_{DG}^2} \Theta^*(s) + \frac{-sJ_{DD}+J_{DG}}{s^2-sJ_{DG}+J_{DG}^2} \Phi^*(s), \quad (40)$$

$$\Phi(s) = \mathcal{T}_{D_1}(s)\Theta^*(s) + \mathcal{T}_{D_2}(s)\Phi^*(s), \quad (41)$$

where \mathcal{T}_{D_1} and \mathcal{T}_{D_2} together express the system dynamics in the frequency domain. Analysis of the poles of this transfer function indicate that the dynamics may be stable or unstable depending on the specific values of θ^* and ϕ^* . Considering the many parameters of a modern deep learning model, these dynamics will almost certainly be unstable for some parameters. When the system is unstable, stabilizing the dynamics around (ϕ^*, θ^*) requires a control algorithm.

One such control algorithm is the closed loop feedback controller of Xu et al. (2019), which is shown to stabilize and improve the convergence of GANs such as the ordinary GAN and Wasserstein GAN. However, this strategy is strictly only valid around the equilibrium and does not address the stability of the trajectory the system takes from the initial state toward the equilibrium. Any such closed-loop controller also requires parameterization to avoid overshoot.

In this dissertation, we take a similar approach with an alternative to Xu et al.'s closed-loop controller, PID control of the rate of approach to the equilibrium. We apply PID control to Pix2Pix GANs as explained in Section 3.4, “Designing StableSeg GAN,” in Chapter 3.

CHAPTER 3

METHODOLOGY

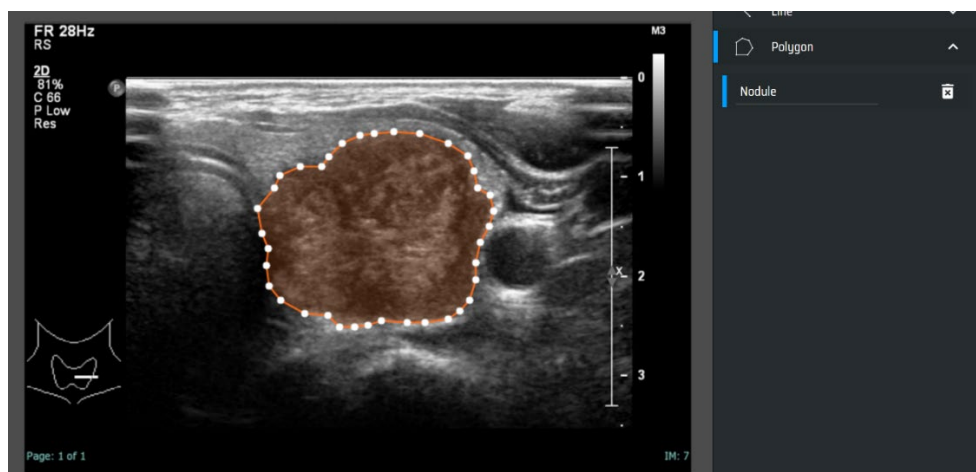
This chapter describes the work process related to the literature from the previous chapter. I explain how to prepare a thyroid nodule dataset and design the integration of DeepLabV3+GANs and the StableSeg GANs.

3.1 Dataset Preparation

We acquired 1527 thyroid nodule ultrasound images from three hospitals: Siriraj Hospital, Vajira Hospital, and Srinakharinwirot Hospital, for the training data. We randomly partitioned the data into 1328 training images and 149 test images. The training images split to 87.13% because we will add the public dataset in the next step. Then, we added 725 images from an open-access thyroid nodule image dataset (Pedraza, 2015) to the training set. We use the public dataset to the training set only because we consider the output from the hospitals. Thus, among 2252 images, 2103 are training images, and 149 are test images. All cases were reported by doctors who were at least a 3rd year resident in diagnostic radiology, and all cases were pathologically confirmed with fine needle aspirations and reported by pathologists.

Figure 3.1

Annotation Nodule Process.



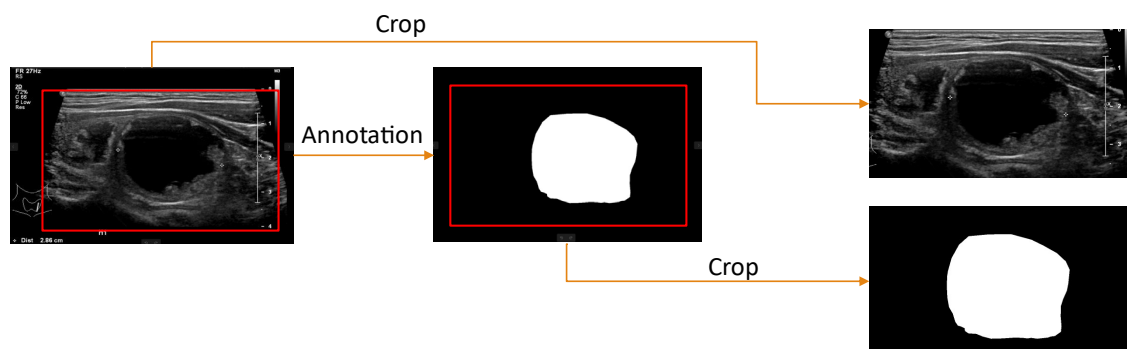
Note: Nodule annotations are specified using <https://www.makesense.ai/>. The annotator was trained to delineate nodules by a professional radiologist, and all annotations were re-confirmed by the radiologist.

Most thyroid ultrasound images are diagnosed in axial view or transverse view; very occasionally, they are diagnosed in longitudinal view. The nodules in every image were annotated by a trained annotator and re-confirmed by radiologists. A sample annotation of a nodule is shown in Figure 3.1.

To reduce noise from the input images, all images were cropped by the program automatically and re-checked manually to contain only the ultrasound data. This also ensures that the models will not “read” text data and use it to predictions that would not generalize. The automatic cropping program is calculated and crop image from the average dark rows and columns which always happen before the ultrasound data appear. The overall process is shown in Figure 3.2.

Figure 3.2

Cropping and Annotation of a Nodule Image.



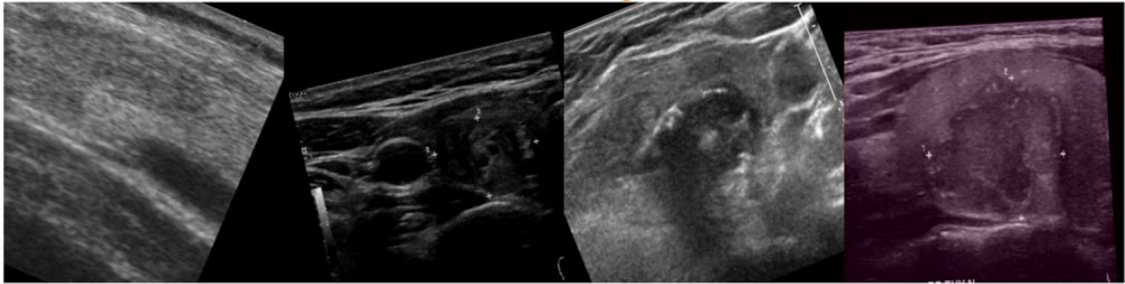
Note: Cropping prior to annotation reduces model confusion.

3.2 Data Augmentation

During training, the all training images are augmented to simulate different ultrasound machine setups. We allow up to 20% distortion in color, brightness, contrast, scale, translation, and rotation. On each epoch, a random series of transformations is applied to each image, which is then resized to 256×256 due to GPU memory constraints during training. Sample augmentations are shown in Figure 3.3.

Figure 3.3

Sample Augmented Images.



To confirm that augmentation improves accuracy, we trained a U-Net256 model two times in with un-augmented and augmented data for ten epochs and compared the result on test data. The results show that augmentation improves accuracy as measured by IoU by more than 10%, as shown in Table 3.1. The augmentation give a better result than the original data because it can create more various dataset.

Table 3.1

Test IoU With and Without Augmentation of Training Images.

Network	Without Augmentation (IoU %)	With Augmentaion (% IoU)
U-Net256	43.27%	57.68%

3.3 Designing DeepLabV3-GANs

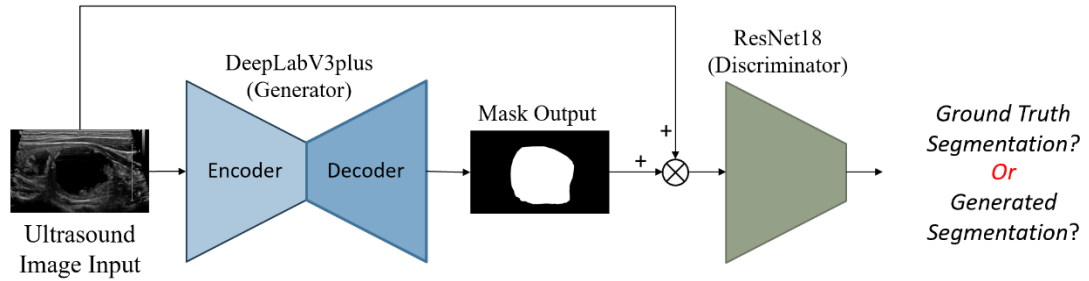
DeeplabV3+ is known to be a good segmentation model, and many researchers have observed that embedding good models into GANs can improve segmentation accuracy. Thus, it is possible that the combination of DeepLabV3+ and GANs can improve the accuracy of ordinary DeepLabV3+.

We designed the DeepLabV3+GAN used in StableSeg GANs using DeeplabV3+ as a generator. The input to the generator is a thyroid nodule ultrasound image with three channels, and the output is a mask over the nodule image with one channel. The nodule mask image and ultrasound image are concatenated and fed to the discriminator model, which classifies the pair as real or fake. This GAN structure replicates the Pix2Pix

model (Isola, 2017). The discriminator is a deep CNN that is as deep as possible without causing mode collapse. The overall architecture is shown in Figure 3.4.

Figure 3.4

DeepLabV3+ GAN Architecture.



The input to the generator is a thyroid nodule ultrasound image with three channels, and the output is a mask over the nodule image with one channel. The nodule mask image and ultrasound image are concatenated and fed to the discriminator model, which classifies the pair as real or fake. The discriminator is a deep CNN that is as deep as possible without causing mode collapse.

The supervised loss G from DeepLabV3+GANs is similar to that in an ordinary Pix2Pix GAN. However, instead of only binary cross-entropy loss, we use a combination of the three supervised learning loss functions: soft dice loss (\mathcal{L}_S), binary cross-entropy loss (\mathcal{L}_B), and Lovasz hinge loss (\mathcal{L}_L):

$$\mathcal{L}_{SG} = \mathcal{L}_S + \mathcal{L}_B + \mathcal{L}_L \quad (42)$$

Combining the supervised loss \mathcal{L}_{SG} with the unsupervised GAN loss functions, we obtain the complete semi-supervised loss for G , adding hyper-parameter γ to balance the supervised loss and the unsupervised loss:

$$\mathcal{L}_G(\theta; \phi) = \gamma \mathbb{E}_{p_G(\mathbf{x})} [h_3(D(\mathbf{x}||G(\mathbf{x})))] + (1 - \gamma) \mathcal{L}_{SG}(\theta) \quad (43)$$

3.4 Designing StableSeg GAN

In pilot studies, we found that the weakness of the DeepLabV3+GAN architecture by itself is that training tends to be unstable. The weight update contribution from the discriminator loss is usually too high or too low relative to the supervised loss, and it is

prone to over-fitting. Moreover, a too-strong discriminator or a too-weak discriminator will fail to improve the generator. Thus, we extend previous work on control of GAN learning, applying PID control to the dynamics of the approach of the discriminator model to an equilibrium point. By controlling the discriminator's descent, we balance the relative learning speed of the generator and discriminator. We first reformulate the loss functions of Equations 15 and 16 to take into account the terms for the supervised learning of the generator. The revised objectives are

$$\min_{\phi} -\mathcal{L}_D(\phi; \theta), \quad (44)$$

$$\min_{\theta} -\mathcal{L}_G(\theta; \phi), \quad (45)$$

where

$$\mathcal{L}_D(\phi; \theta) = \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [h_1(D(\mathbf{x} \parallel \mathbf{y}))] + \mathbb{E}_{p_G(\mathbf{x}, \mathbf{y})} [h_2(D(\mathbf{x} \parallel G(\mathbf{x})))], \quad (46)$$

$$\mathcal{L}_G(\theta; \phi) = \gamma \mathbb{E}_{p_G(\mathbf{x}, \mathbf{y})} [h_3(D(\mathbf{x} \parallel G(\mathbf{x})))] + (1 - \gamma) \mathcal{L}_{SG}. \quad (47)$$

To stabilize the dynamics of this GAN's learning, as explained in the GAN stabilization section of Chapter 2, we utilize the fact that at the equilibrium point $(\phi^*; \theta^*)$, we have $p_{G^*}(\mathbf{y}) = p(\mathbf{y})$, $D(\mathbf{x} \parallel G(\mathbf{x}; \theta^*); \phi^*) = 0$, $\nabla_{\mathbf{y}} D(\mathbf{x} \parallel \mathbf{y}; \phi^*) = 0$, and $\nabla_{\mathbf{y}}^2 D(\mathbf{x} \parallel \mathbf{y}; \phi^*) > 0$ throughout the support of p . Following the same steps given in Chapter 2, we obtain the typically unstable dynamics

$$\Phi(s) = \mathcal{T}_{D_1}(s)\Theta^*(s) + \mathcal{T}_{D_2}(s)\Phi^*(s), \quad (48)$$

which we stabilize using, in our case, PID closed-loop control. We call the new method *StableSeg GANs*.

An algorithm for the training process is given in Figure 3.5. The controller's transfer function in the frequency domain is

$$K_{PID}(s) = K_p + \frac{K_i}{s} + sK_d, \quad (49)$$

where K_p , K_i , and K_d are hyper-parameters for system control in the PID algorithm.

As with the closed-loop control method of Xu (2019), the dynamics can be stabilized with appropriate values of the controller transfer function parameters.

Figure 3.5

StableSeg GAN Training for One Epoch.

Algorithm 1 PID Training of StableSeg GAN (1 epoch)

Input: Training dataset DS emitting pairs (\mathbf{x}, \mathbf{y})

$\mathcal{L}_I \leftarrow 0$

$n \leftarrow 0$

while DS not empty **do**

 Sample minibatch $\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}$ from DS

$\hat{\mathbf{Y}}^{(n)} \leftarrow G(\mathbf{X}^{(n)})$

$\mathcal{L}_t \leftarrow h_1(D(\mathbf{X}^{(n)} \parallel \mathbf{Y}^{(n)}))$

$$\mathcal{L}_o \leftarrow \begin{cases} 0, & n = 0 \\ h_1(D(\mathbf{X}^{(n-1)} \parallel \mathbf{Y}^{(n-1)})) \\ \quad + h_2(D(\mathbf{X}^{(n-1)} \parallel \hat{\mathbf{Y}}^{(n-1)})), & \\ \text{otherwise} \end{cases}$$

$\mathcal{L}_P \leftarrow K_p \mathcal{L}_t$

$\mathcal{L}_I \leftarrow \mathcal{L}_I + K_i \mathcal{L}_t$

$\mathcal{L}_D \leftarrow K_d(\mathcal{L}_t - \mathcal{L}_o)$

$\mathcal{L}_c \leftarrow \mathcal{L}_P + \mathcal{L}_I + \mathcal{L}_D$

$\phi \leftarrow \phi + \alpha_\phi \nabla_\phi \mathcal{L}_c$

$\mathcal{L}_G \leftarrow \gamma h_3(D(\mathbf{X}^{(n)} \parallel \hat{\mathbf{Y}}^{(n)})) + (1 - \gamma) \mathcal{L}_{SG}$

$\theta \leftarrow \theta + \alpha_\theta \nabla_\theta \mathcal{L}_G$

$n \leftarrow n + 1$

end while

Let $\mathbf{X}^{(n)} \parallel \mathbf{Y}^{(n)}$ denote the minibatch used for the n^{th} training iteration, and let \mathcal{G} be a stochastic estimate of \mathcal{L}_D using $\mathbf{X}^{(n)} \parallel \mathbf{Y}^{(n)}$. In the time domain, the update equation for ϕ using the stochastic gradient \mathcal{G} of \mathcal{L}_D is

$$\phi^{(n+1)} = \phi^{(n)} - \alpha[\mathcal{P} + \mathcal{J} + \mathcal{D}] \quad , \quad (50)$$

where \mathcal{P} , \mathcal{J} , and \mathcal{D} are described as

$$\mathcal{P} = K_p \mathcal{G}(\phi, \mathbf{X}^{(n)} \parallel \mathbf{Y}^{(n)}; \theta) \Big|_{\phi=\phi^{(n)}}, \quad (51)$$

$$\mathcal{J} = K_i \sum_{i=1}^n \mathcal{G}(\phi, \mathbf{X}^{(n)} \parallel \mathbf{Y}^{(n)}; \theta) \Big|_{\phi=\phi^{(i)}}, \quad (52)$$

$$\mathcal{D} = K_d (\mathcal{G}(\phi, \mathbf{X}^{(n)} \parallel \mathbf{Y}^{(n)}; \theta) \Big|_{\phi=\phi^{(n)}} - \mathcal{G}(\phi, \mathbf{X}^{(n)} \parallel \mathbf{Y}^{(n)}; \theta) \Big|_{\phi=\phi^{(n-1)}}), \quad (53)$$

where $\mathcal{G}(\phi, \mathbf{X}^{(n)} \parallel \mathbf{Y}^{(n)}; \theta) \Big|_{\phi=\phi^{(n)}}$ denotes estimating the expectations in Equation 46 with $\phi^{(n)}$ and minibatch n . Figure 3.5 describes the overall process programmatically.

CHAPTER 4

EXPERIMENTAL RESULTS

This chapter gives detailed results of my experiments with DeeplabV3+GAN and StableSeg GAN.

4.1 StableSeg GAN Training Parameters

While training StableSeg GANs, we use Adam optimization with a learning rate of 0.001 with a linearly increasing learning rate from 0 to 0.001 over the first five epochs for warmup. We use PID control parameters $K_p = 0.5$, $K_i = 10$, and $K_d = 0.01$. We train until convergence. The PID parameters were hand tuned to obtain stable training curves.

In addition, we found that with $K_i = K_d = 0$ (only K_p is tuned), training loss was unstable, leading to mode collapse. However, when we changed the discriminator in StableSeg GANs to be the same as in DeepLabV3+ GAN, we found that it exhibits nearly the same behavior as the ordinary DeepLabV3+ GAN.

4.2 Generator Loss in StableSeg GANs

One of the findings of this research is that loss control for GANs reduces the problem of mode collapse, enabling more decrease in generator loss than is possible without controlled loss. This in turn allows us to use a more powerful discriminator than would otherwise be possible. We therefore replace the eight-layer CNN used for the discriminator in DeepLabV3+GANs with ResNet18 (He, 2016) because the ResNet18 is verified that it can classify in image classification task higher than 75% of ImageNet dataset.

As shown in Table 4.1, without stabilization, DeeplabV3+GANs require more weight on the supervised loss (best $\gamma = 0.5$) and fail to achieve a good generator loss, but as shown in Tables 4.2 and 4.3, StableSeg GANs with stabilization do not require such high weights on the supervised loss, γ to 0.9. However, with $\gamma = 1.0$, neither DeeplabV3+GANs nor StableSeg GANs learn well. We conclude that StableSeg GANs benefit from having more flexibility to “ignore” the supervised signal and pay more attention to fool the discriminator.

We performed a hyperparameter search experiment to find the best value of γ (the relative weight on supervised vs. unsupervised learning) and found that with stabilization, the model performs best with $\gamma = 0.9$ avoiding mode collapse, as shown in Tables 4.3.

Table 4.1

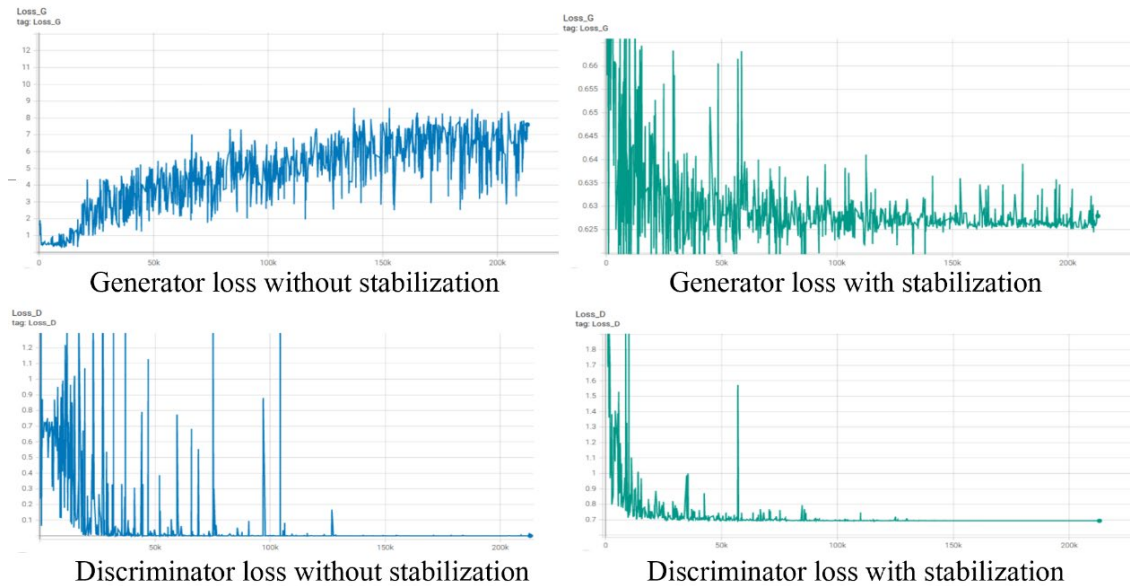
Accuracy of DeeplabV3+GANs with 8-layer Discriminator and Different γ

DeepLabV3+ GANs	IoU	Acc	Sen/Rec	Spec	Prec
$\gamma = 1.0$	16.478	34.299	76.093	25.198	17.872
$\gamma = 0.9$	78.581	94.705	88.439	95.644	85.845
$\gamma = 0.5$	79.027	96.136	88.512	97.533	88.499
$\gamma = 0.1$	78.689	94.747	89.446	95.677	86.033

Note: The results are from 149 images test set. IoU: Intersection over Union, Acc: Accuracy, Sen/Rec: Sensitivity/Recall, Spec: Specificity, Prec: Precision

Figure 4.1

StableSeg GAN G/D Loss With and Without Stabilization.



Note: With stabilization, G learns well, and D 's loss approaches the theoretical equilibrium (actual loss = 0.7, ideal loss = 0.5).

Table 4.2*Accuracy of StableSeg GANs with 8-layer Discriminator and Different γ*

StableSeg GANs	IoU	Acc	Sen/Rec	Spec	Prec
$\gamma = 1.0$	24.135	35.222	75.119	22.232	26.116
$\gamma = 0.9$	80.611	96.640	90.354	97.605	87.934
$\gamma = 0.5$	80.027	96.636	89.012	97.533	87.499
$\gamma = 0.1$	78.324	94.051	88.448	97.300	88.514

Note: The results are from 149 images test set. IoU: Intersection over Union, Acc: Accuracy, Sen/Rec: Sensitivity/Recall, Spec: Specificity, Prec: Precision

Table 4.3*Accuracy of StableSeg GANs with ResNet18 Discriminator and Different γ*

StableSeg GANs	IoU	Acc	Sen/Rec	Spec	Prec
$\gamma = 1.0$	22.232	32.811	77.638	23.143	26.786
$\gamma = 0.9$	81.973	97.142	91.079	97.862	88.211
$\gamma = 0.5$	79.992	96.444	89.265	97.196	87.800
$\gamma = 0.1$	78.252	96.107	88.673	97.243	88.320

Note: The results are from 149 images test set. IoU: Intersection over Union, Acc: Accuracy, Sen/Rec: Sensitivity/Recall, Spec: Specificity, Prec: Precision

4.3 Supervised and Unsupervised Loss Function Evolution During Training

Figure 4.2 shows a breakdown of generator loss into its supervised and unsupervised components in order to observe the effect of each component over time. Early on, supervised loss has more effect than unsupervised loss, so it dominates the early weight updates. We conjecture that during these early iterations, the model learns the rough structure of the thyroid nodule. By 1000 iterations, the effect of supervised learning is less than that of unsupervised learning. Clearly, by this time, the model is adapting mainly to improve the unsupervised loss, no doubt “ignoring” some aspects of the

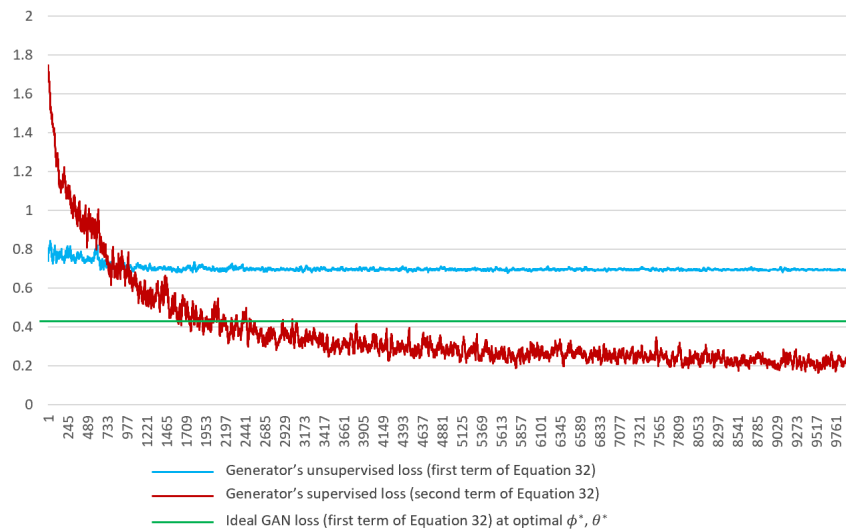
supervised targets in order to make small improvements as perceived by the discriminator.

4.4 Ablation Study

We compared StableSeg GANs with the Resnet18 discriminator against alternative models, models, including DeepLabV3+, DeepLabV3+GAN, and StableSeg GANs with the 8-layer discriminator. The results are shown in Table 4.4. The comparison of mask results in Figure 4.3 shows that StableSeg GANs admit a morphology more like the radiologists' contours. Comparing the segmentations from left to right in Figure 4.3, it is clear that the DeeplabV3+ based models segment nodule contours more effectively than the other methods. Among the DeeplabV3+ variations, DeepLabV3+ using an ordinary GAN is only a marginal improvement over ordinary DeepLabV3+. On the other hand, the StableSeg GAN with ResNet18 as a discriminator gives substantially better segmentations than the other models in the sense that the contours of the object are more similar to the radiologists' segmentations.

Figure 4.2

Breakdown of Generator Loss into Unsupervised Loss and Supervised Loss.



Note: Supervised loss ($Loss_{L1}$: red) decreases rapidly early on; later in training, unsupervised learning ($Loss_{GAN}$: blue) dominates. When using StableSeg GANs, the $Loss_{L1}$ nearly reaches zero, and $Loss_{GAN}$ nearly reaches the ideal loss for ϕ^* and θ^* .

Table 4.4*Ablation Study: Accuracy Comparison Between StableSeg GANs and Other Models.*

Network	IoU	Acc	Sen/Rec	Spec	Prec
DeepLabV3+	78.973	96.142	89.079	97.162	87.211
DeepLabV3+GANs	79.027	96.136	88.512	97.533	88.499
StableSeg GANs (8-layers discriminator)	80.611	96.640	90.354	97.605	87.934
StableSeg GANs (ResNet18 discriminator)	81.973	97.142	91.079	97.862	88.211

Note: The results are from a 149 image test set. IoU: Intersection over Union, Acc: Accuracy, Sen/Rec: Sensitivity/Recall, Spec: Specificity, Prec: Precision

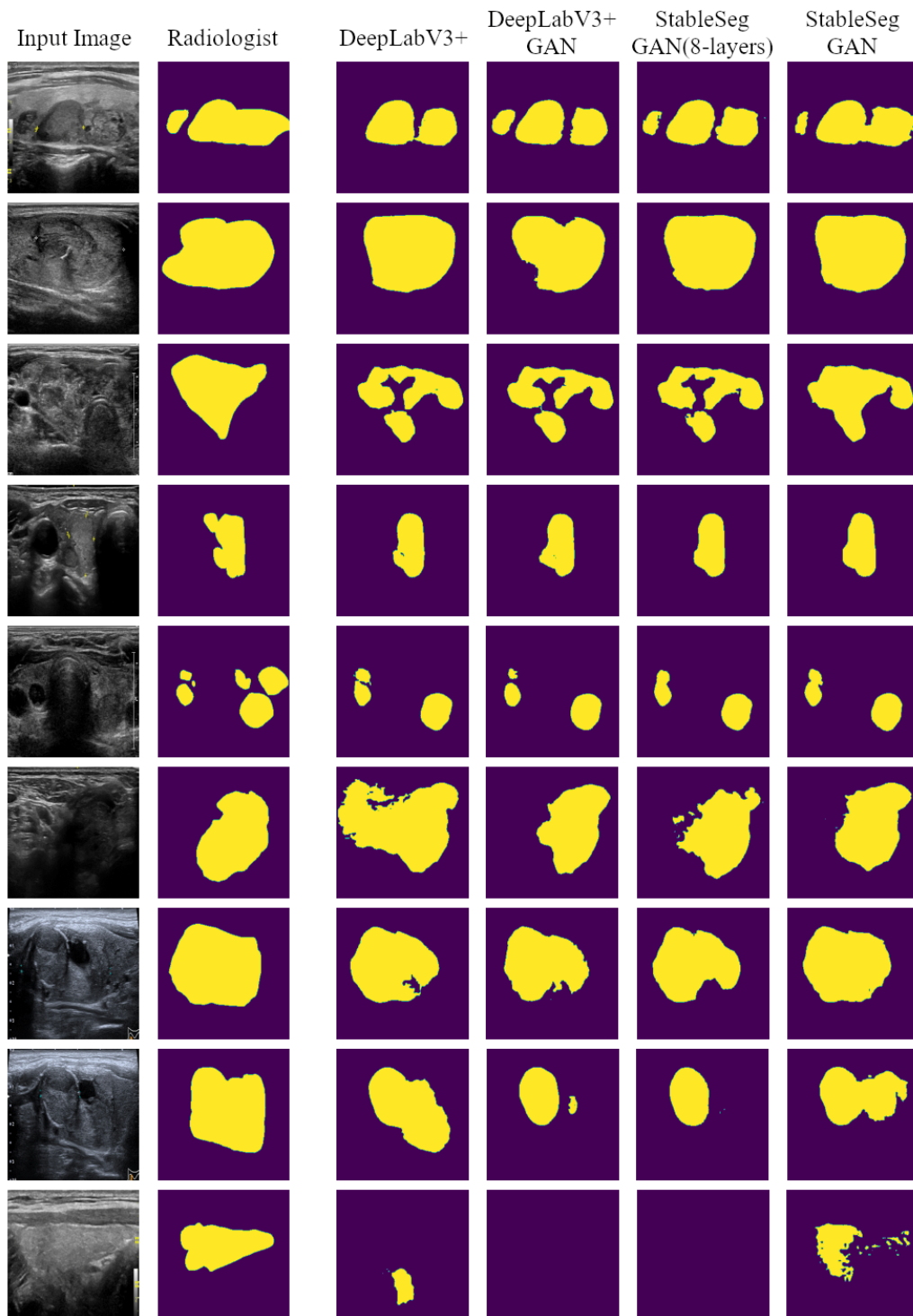
4.5 Experiment with Multiple Organs

As a test of the generality of StableSeg GANs, we performed an experiment with a new design that can segment thyroid ultrasound images as well as images of other organs such as the breast. We modified the generator input of StableSeg GANs to include not only a three-channel image but also one channel for each type of organ, enabling the same model to segment nodules in multiple organs. The discriminator input is modified to comprise the three channels of the image, one channel for the nodule mask, and n channels specifying the organ type. The structure of the new StableSeg GAN is shown in Figure 4.4.

We tested the generality of the modified model using 780 images from an open-access dataset of breast ultrasound images (Walid, 2020), and randomly separated them into 90% for the training set (702 images) and 10% for the test set (78 images). The combined results are less accurate than models with only one type of organ, as shown in Table 4.5. Thus, we conclude that StableSeg GANs can be generalized, but factors such as imbalance between breast and thyroid or ultrasound images may affect accuracy. With the small training set here, separate StableSeg GANs may be more suitable than the general one, as medical imaging requires maximum high accuracy.

Figure 4.3

Mask Result Comparison for a Random Sample from the Test Set.



Note: Left to right: Ultrasound input images, mask annotations from radiologist, mask from DeepLabV3+, mask from DeepLabV3+GAN, mask from StableSegGAN with 8 CNN layers, mask from StableSegGAN. Qualitatively, StableSegGAN masks are more similar to the radiologist's masks than are the other methods' masks.

Table 4.5

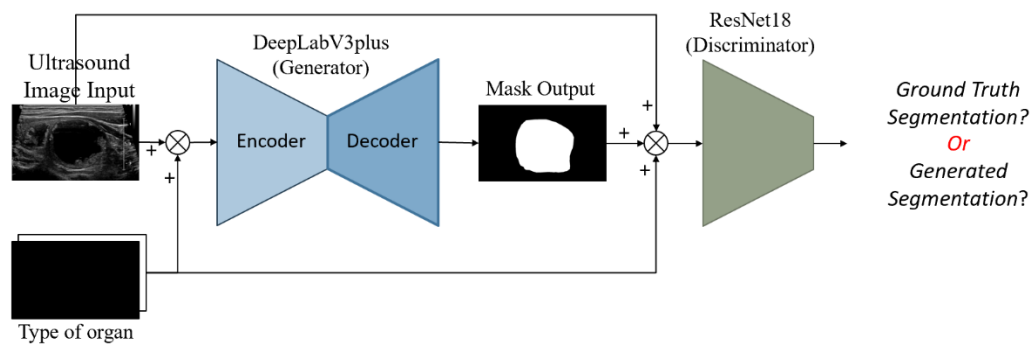
Accuracy of General StableSeg Gans with Resnet18 Discriminator

Model	Organ	IoU	Acc	Sen/Rec	Spec	Prec
Generalized StableSeg GAN	Thyroid	78.392	96.091	89.028	97.130	87.139
Generalized StableSeg GAN	Breast	77.645	95.354	88.444	97.233	87.652
StableSeg GAN	Thyroid	81.973	97.142	91.079	97.862	88.211
StableSeg GAN	Breast	80.894	96.378	90.199	98.021	89.022

Note: The results are from a 149 image thyroid ultrasound test set and a 78 image breast ultrasound test set. IoU: Intersection over Union, Acc: Accuracy, Sen/Rec: Sensitivity/Recall, Spec: Specificity, Prec: Precision

Figure 4.4

Generalized StableSeg GANs Network.



Note: The modified StableSeg GANs more general than the separate models. The model adds input channels indicating the organ to the generator and the discriminator, enabling the model to support multiple organs.

CHAPTER 5

CONCLUSION AND DISCUSSION

This chapter provides a summary of the dissertation, including the advantages and disadvantages of the approach. Moreover, it gives some possible directions for future work and improvements.

5.1 Discussion

We find that the DeeplabV3+GAN with an eight-layer discriminator can segment thyroid nodules more accurately than DeeplabV3+. However, we found that increasing the depth of the discriminator beyond eight layers led to poor segmentation. We observed that with a deeper discriminator, learning was unstable, with mode collapse. The reason for this phenomenon was that the discriminator was learning too quickly relative to the generator. Simple remedies, such as stopping discriminator learning or changing learning rates did not give satisfactory results, as there was no dynamic adaptability in the static hyperparameters.

To address this problem, we apply a control algorithm to adjust the dynamics of the GAN's loss, which is equivalent to dynamic adaptive adjustment of the learning rate. We utilize the simple PID control algorithm to guide the generator and discriminator to an equilibrium point. We find that controlling dynamics in this result was in the loss of the generator and discriminator converging to an equilibrium point even with the higher capacity ResNet18 discriminator. We call the DeeplabV3+GAN with the ResNet18 discriminator and controlled dynamics StableSeg GAN. In the new model, loss not only converges more robustly, but we also see improvement in accuracy complicated to DeeplabV3+ and DeeplabV3+GAN. On real thyroid nodule ultrasound test images, we find that the StableSeg GAN can detect some nodules that DeeplabV3+ cannot find. Overall, the extent to which controlled unsupervised learning can improve detection relative to supervised learning is striking.

5.2 Conclusion

The results of our thyroid nodule segmentation experiments show that StableSeg GANs have more flexibility to segment nodules more accurately than DeepLabV3+, which is trained using supervised learning only. We also observe that qualitatively, StableSeg

GANs more effectively capture radiologists' expertise than do DeepLabV3+GANs. They may therefore be able to predict locations of nodules that cannot be found by the supervised model or DeeplabV3+GANs in new images in the future. We conclude that generally, segmenting nodules is improved by adding well controlled unsupervised learning. To further increase the use of unsupervised learning and to further improve the GAN's performance, we introduce a relatively simple way to optimize the dynamics of the model's approach to an equilibrium using a simple yet effective closed loop control method, PID, to balance the learning between the generator and the discriminator. This has multiple benefits, allowing the use of a stronger discriminator and leading to a better generator that is stronger at its maximum performance.

The ideas behind StableSeg GANs can potentially be exploited in many applications. In medical imaging applications, we can use the same approach to optimize models for other organs or other diseases. For ultrasound images, we can predict not only locations of thyroid nodules but also other important parts of other organs such as the breast and liver.

In the future, the model can be improved to simultaneously segment and classify organ parts as diseased or not diseased. Moreover, to improve accuracy, we can increase the size of the dataset. The new data could be a mix of labeled and unlabeled data. We hypothesize that self-supervised learning using unlabelled data will improve accuracy on labelled test data. Increased data volume will allow expansion of the generator model and discriminator model. More sophisticated methods for optimization control such as sliding mode, state space, and non-linear control methods may also be beneficial in further improving StableSeg GANs.

We plan to run trials of the model for thyroid nodule cancer diagnosis with radiologists in the near future.

REFERENCES

- Anand, A. T. (2017). A Local Statistics Based Region Growing Segmentation Method for Ultrasound Medical Images. *World Academy of Science, Engineering and Technology, International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering, 1*, 564-569.
- Åse Tangerud, E. S. (2020). Thyroid nodules -- investigation and follow-up. *Tidsskriftet*. Retrieved from <https://tidsskriftet.no/en/2020/09/klinisk-oversikt/thyroid-nodules-investigation-and-follow>
- Badrinarayanan, V. a. (2017, Dec). SegNet: A Deep Convolutional {Encoder-Decoder} Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39, 2481--2495.
- Bank, D. a. (2020). Autoencoders. *arXiv*, cs.LG. doi:2003.05991
- Bochkovskiy, A. a.-Y.-Y. (2019). YOLOv4: Optimal speed and accuracy of object detection. *arXiv cs.CV*. doi:1904.02689
- Bomeli, S. R. (2010, Apr). Evaluation of a thyroid nodule. *Otolaryngol. Clin. North Am.*, 43, 229--38, vii.
- Bottou, M. A. (2017). Wasserstein Generative Adversarial Networks. *Proceedings of the 34th International Conference on Machine Learning*. 70, pp. 214--223. PMLR.
- Chaiyasut W., C. A. (2017). *Ultrasound in Clinical Practice การวินิจฉัยด้วยคลื่นเสียงความถี่สูง*. Bangkok, Thailand: Mahidol University.
- Chen, L.-C. a. (2014, Dec). Semantic image segmentation with deep convolutional nets and fully connected CRFs. *arXiv cs.CV*. doi:1412.7062
- Chen, L.-C. a. (2017, Jun). Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv cs.CV*. doi:1706.05587
- Chen, L.-C. a. (2018, Feb). Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv cs.CV*. doi:1802.02611
- Daniel Bolya, C. Z. (2019). YOLACT: Real-time Instance Segmentation. *ArXiv*. doi:arXiv:1904.02689
- Dou, Q. a.-A. (2018, Apr). Unsupervised cross-modality domain adaptation of ConvNets for biomedical image segmentations with adversarial loss. *arXiv cs.CV*. doi:1804.10916

- Goodfellow, I. J.-A.-F. (2014, Jun). Generative Adversarial Networks. *arXiv stat.ML*. doi:1406.2661
- Greenspan, H. a. (2016, May). Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Trans. Med. Imaging*, 35, 1153--1159.
- He, K. a. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE.
- He, K. a. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE.
- Hesamian, M. H. (2019, Aug). Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges. *Journal of Digital Imaging*, 32, 582--596.
- Huang, C. a. (2020). Skin lesion segmentation based on mask R-CNN. *2020 International Conference on Virtual Reality and visualization (ICVRV)*. Recife, Brazil: IEEE.
- Isola, P. a.-Y. (2017). Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE.
- Jadon, S. (2020, Jun). A survey of loss functions for semantic segmentation. *arXiv eess.IV*. doi:2006.14822
- Jones, J. a. (2015, Mar). *Assessment of thyroid lesions (ultrasound)*. Retrieved from Radiopaedia.org: Radiopaedia.org
- Ker, J. a. (2018). Deep learning applications in medical image analysis. *IEEE Access*, 6, 9375--9389.
- Long, J. a. (2015). Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE.
- Mescheder, L. a. (2018, Jan). Which Training Methods for GANs do actually Converge? *arXiv cs.LG*. doi:1801.04406
- Pedraza, L. a. (2015). An open access thyroid ultrasound image database. *Tenth International Symposium on Medical Information Processing and Analysis*. Cartagena de Indias, Colombia: SPIE.

- Pluim, V. C. (2019). Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis*, 54, 280-296.
- Radovick, F. E. (2010). Clinical Management of Thyroid Disease. *Mayo Clinic Proceeding*, 434. doi:10.4065/mcp.2010.0648
- Rajest, K. R. (2021, Apr). A Review of Medical Image Segmentation Algorithms. *EAI Endorsed Transactions on Pervasive Health and Technology*, 7. doi:10.4108/eai.12-4-2021.169184
- Redmon, J. a. (2018). YOLOv3: An Incremental Improvement. *arXiv cs.CV*. doi: 1804.02767
- Ronneberger, O. a. (2015, May). U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv cs.CV*. doi:1505.04597
- S. Kevin Zhou, H. G. (2021). A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*. doi: 10.1109/JPROC.2021.3054390
- Salimans, T. a. (2016). Improved Techniques for Training GANs. *Advances in Neural Information Processing Systems*. 29. Curran Associates, Inc.
- Sánchez, G. L. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60-88. doi:https://doi.org/10.1016/j.media.2017.07.005
- Szegedy, C. a. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE.
- Szegedy, C. a. (2016). Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE.
- Tessler, F. N. (2018). Thyroid Imaging Reporting and Data System (TI-RADS): A User's Guide,. *Radiology*, 287, 29-36. doi: 10.1148/radiol.2017171240
- Tuysuzoglu, A. a. (2018, May). Deep adversarial context-aware landmark detection for ultrasound imaging. *arXiv cs.CV*. doi:1805.10737
- Walid Al-Dhabyani, M. G. (2020). Dataset of breast ultrasound images. *Data in Brief*. doi:https://doi.org/10.1016/j.dib.2019.104863
- Wang, M. a. (2021). *Automatic segmentation and classification of thyroid nodules in ultrasound images with convolutional neural networks*. Springer International Publishing.

- Xin Yi, E. W. (2019). Generative adversarial network in medical imaging: A review. *Medical Image Analysis*, 101552. doi:10.1016/j.media.2019.101552
- Xu, K. a. (2019, Sep). Understanding and stabilizing GANs' training dynamics with control theory. *arXiv cs.LG*. doi:1909.13188
- Yadav, A. a. (2017, May). Stabilizing adversarial nets with prediction methods. *arXiv cs.LG*. doi:1705.07364
- Yang, D. a. (2017, Jul). Automatic liver segmentation using an adversarial image-to-image network. *arXiv cs.CV*, 1707.08037.
- Yao, S. a. (2020, Jun). Texture synthesis based thyroid nodule detection from medical ultrasound images: Interpreting and suppressing the adversarial effect of in-place manual annotation. *Front. Bioeng. Biotechnol.*, 8, 599.
- Ye, H. a. (2020, Aug). An intelligent platform for ultrasound diagnosis of thyroid nodules. *Sci. Rep.*, 10, 13223.
- Yu, X. a. (2020). Detection of thyroid nodules with ultrasound images based on deep learning. *Curr. Med. Imaging Rev.*, 16, 174--180.
- Zhou, S. K. (2021). A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proc. IEEE Inst. Electr. Electron. Eng.*, 109, 820--838.
- Zhu, J.-Y. a. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv cs.CV*. doi:1703.10593

APPENDICES

APPENDIX A

JACOBIAN OF THE GRADIENT VECTOR FIELD IN GANS

We declare the GANs equation in each parameter. At first, we know that the gradient vector field for the discriminator and generator for learning over time from Equation 28 can be written as

$$v(\phi, \theta) = \begin{bmatrix} \nabla_{\phi} \mathcal{L}_D(\phi; \theta) \\ \nabla_{\theta} \mathcal{L}_G(\theta; \phi) \end{bmatrix}, \quad (54)$$

where the gradient of $\mathcal{L}_D(\phi; \theta)$ and $\mathcal{L}_G(\theta; \phi)$ with respect to ϕ and θ respectively, are

$$\nabla_{\phi} \mathcal{L}_D(\phi; \theta) = \mathbb{E}_{p(\mathbf{y})} [h'_1(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi)] + \mathbb{E}_{p_G(\hat{\mathbf{y}})} [h'_2(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi)], \quad (55)$$

$$\nabla_{\theta} \mathcal{L}_G(\theta; \phi) = \mathbb{E}_{p_x(\mathbf{x})} [h_3(D(G(\mathbf{x}; \theta); \phi)) [\nabla_{\theta} D(G(\mathbf{x}; \theta); \phi)]^T \cdot \nabla_{G(\mathbf{x})} D(G(\mathbf{x}; \theta); \phi)]. \quad (56)$$

As the gradient vector field $v(\phi, \theta)$ at above, the Jacobian of the gradient vector field at the equilibrium point (ϕ^*, θ^*) can be

$$v'(\phi^*, \theta^*) = \begin{bmatrix} J_{DD} & J_{DG} \\ J_{GD} & J_{GG} \end{bmatrix}, \quad (57)$$

For J_{DD} , it can be derived from

$$\begin{aligned} J_{DD} &= \nabla_{\phi} [\nabla_{\phi} \mathcal{L}_D(\phi; \theta)] \\ &= \nabla_{\phi} [\mathbb{E}_{p(\mathbf{y})} [h'_1(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi)] + \mathbb{E}_{p_G(\hat{\mathbf{y}})} [h'_2(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi)]] \quad (58) \\ &= \nabla_{\phi} [\mathbb{E}_{p(\mathbf{y})} [h''_1(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T + h''_1(D(\mathbf{y}; \phi)) \nabla_{\phi}^2 D(\mathbf{y}; \phi)] \\ &\quad + \mathbb{E}_{p_G(\hat{\mathbf{y}})} [h''_2(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T + h'_2(D(\mathbf{y}; \phi)) \nabla_{\phi}^2 D(\mathbf{y}; \phi)]] \end{aligned}$$

At the equilibrium point (ϕ^*, θ^*) ,

at which the generator perfectly matches the data distribution p , we know that, throughout the support of p , $p_G(\mathbf{y}) = p(\mathbf{y})$, $D(G(\mathbf{x}; \theta^*); \phi^*) = 0$, $\nabla_{\mathbf{y}} D(\mathbf{y}; \phi^*) = 0$, and $\nabla_{\mathbf{y}}^2 D(\mathbf{y}; \phi^*) = 0$. Thus, we can derive Equation 58 to be

$$\begin{aligned}
J_{DD} &= \mathbb{E}_{p(\mathbf{y})} [h_1''(0) \nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T + 0] \\
&\quad + \mathbb{E}_{p_G(\mathbf{y})} [h_2''(0) \nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T + 0] \\
&= h_1''(0) \nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T + h_2''(0) \nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T \Big|_{\phi=\phi^*} \quad (59) \\
&= -\frac{1}{2} \mathbb{E}_{p(\mathbf{y})} [\nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T] \Big|_{\phi=\phi^*}
\end{aligned}$$

For J_{DG} , it can be derived as

$$\begin{aligned}
J_{DG} &= \nabla_{\theta} [\nabla_{\phi} \mathcal{L}_D(\phi; \theta)] \\
&= \nabla_{\theta} [\mathbb{E}_{p(\mathbf{y})} [h_1'(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi)] + \mathbb{E}_{p_G(\mathbf{y})} [h_2'(D(\mathbf{y})) \nabla_{\phi} D(\mathbf{y}; \phi)]] \quad (60) \\
&= \mathbb{E}_{p(\mathbf{y})} [h_1''(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T + h_1'(D(\mathbf{y}; \phi)) \nabla_{\theta} \nabla_{\phi} D(\mathbf{y}; \phi)] \\
&\quad + \mathbb{E}_{p_G(\mathbf{y})} [h_2''(D(\mathbf{y})) \nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T + \nabla_{\theta} h_2'(D(\mathbf{y})) \nabla_{\phi} D(\mathbf{y}; \phi)]
\end{aligned}$$

At equilibrium point, we can decrease the form to be

$$\begin{aligned}
J_{DG} &= h_1''(0) \mathbb{E}_{p(\mathbf{y})} [\nabla_{\phi} D(\mathbf{y}; \phi) \nabla_{\phi} D(\mathbf{y}; \phi)^T + 0] \\
&\quad + h_2''(0) \nabla_{\theta} \mathbb{E}_{p_G(\mathbf{y})} [0 + \nabla_{\phi} D(\mathbf{y}; \phi)] \Big|_{\phi=\phi^*, \theta=\theta^*} \\
&= -\frac{1}{2} \nabla_{\theta} \mathbb{E}_{p_G(\mathbf{y})} [\nabla_{\phi} D(\mathbf{y}; \phi)] \Big|_{\phi=\phi^*, \theta=\theta^*} \quad (61)
\end{aligned}$$

For J_{GD} , we know that the function $h_2'(\cdot) = h_3'(\cdot)$, and $\mathcal{L}_G(\theta; \phi)$ has the same equation of $-\mathcal{L}_D(\phi; \theta)^T$ when $\mathbb{E}_{p(\mathbf{y})} [h_1'(D(\mathbf{y}; \phi)) \nabla_{\phi} D(\mathbf{y}; \phi)] = 0$. Thus, J_{GD} can be

$$J_{GD} = \nabla_{\phi} [\nabla_{\theta} \mathcal{L}_G(\theta; \phi)] = -J_{DG}^T \quad (62)$$

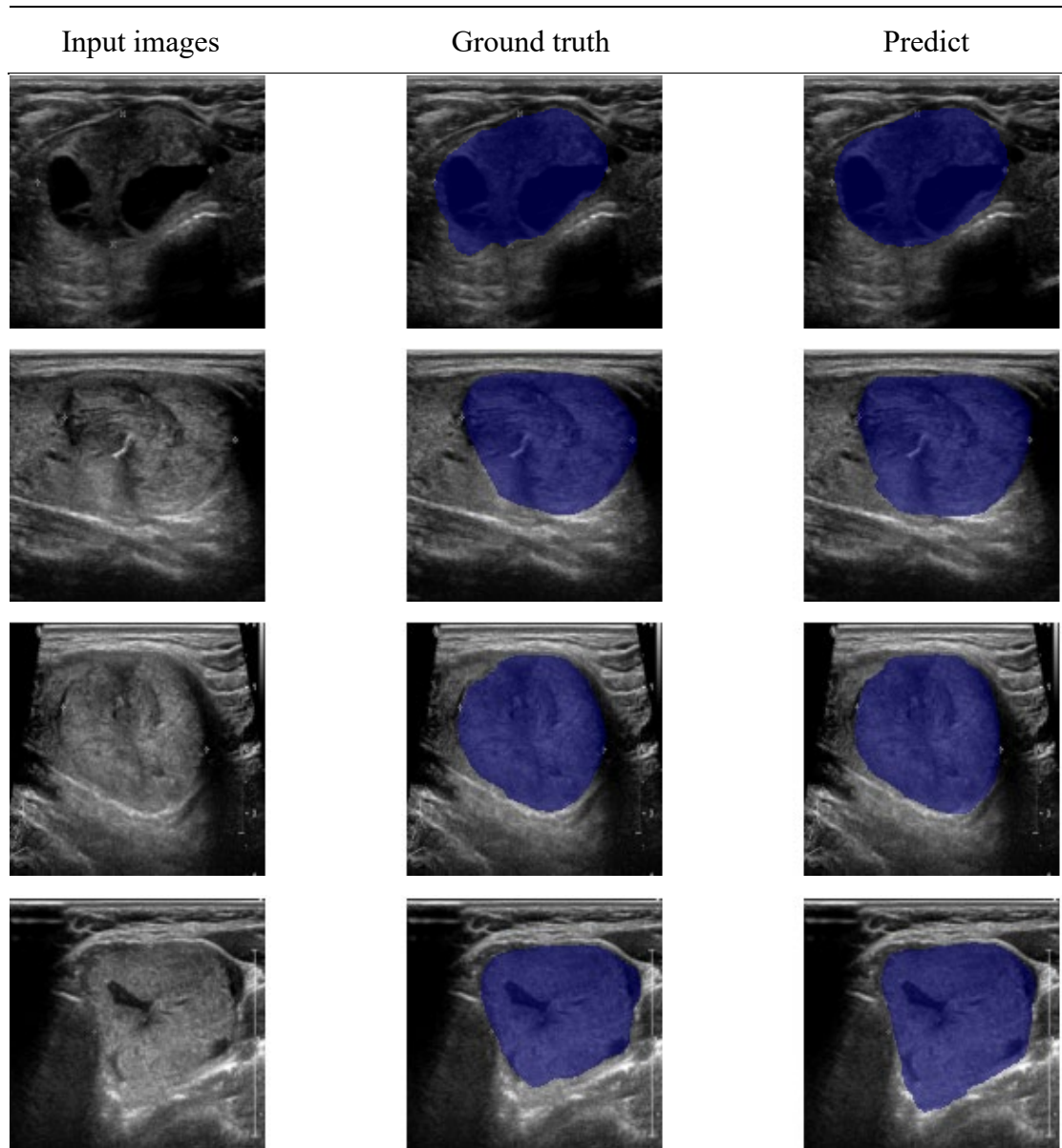
For J_{GG} , it can be derived as

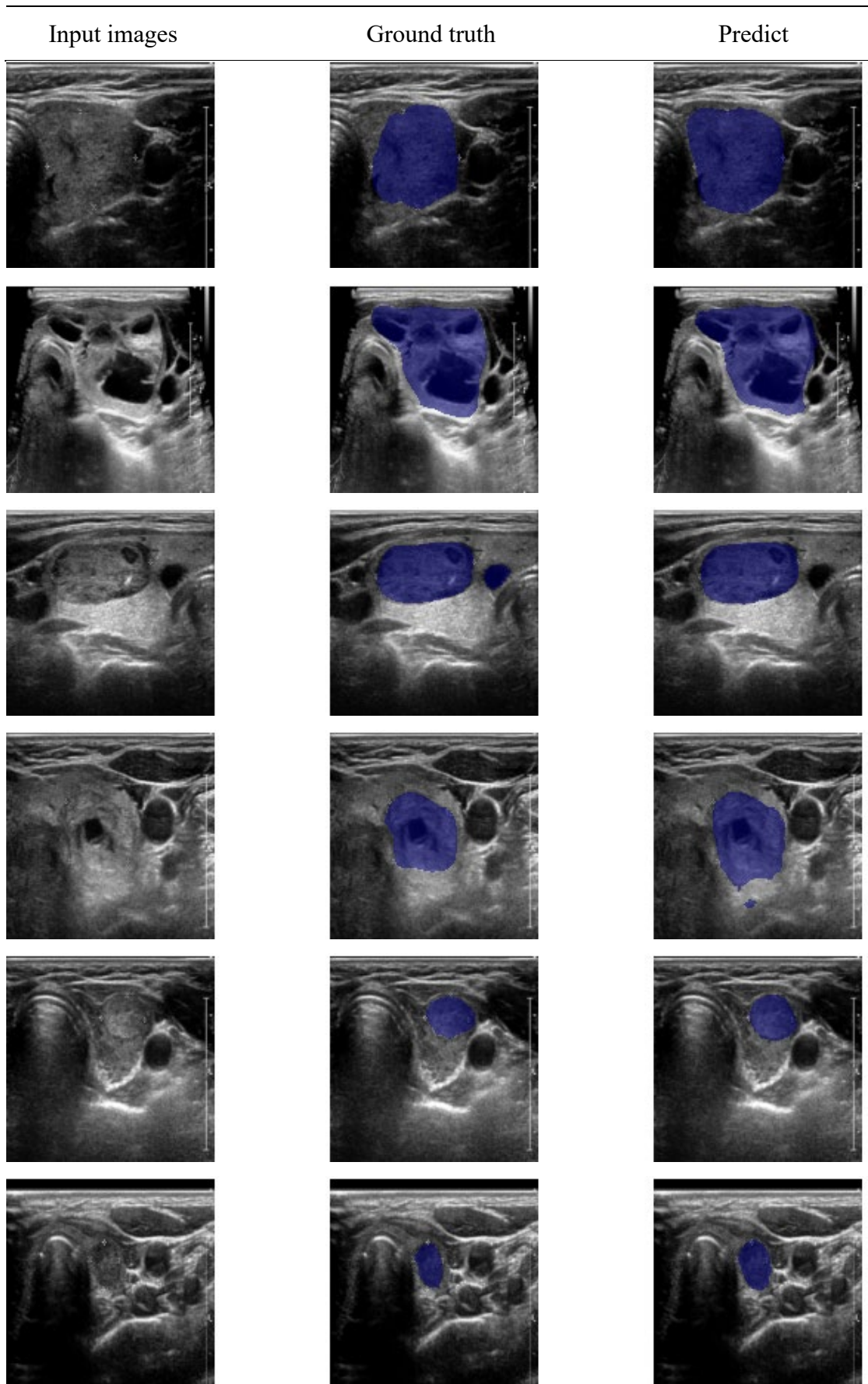
$$\begin{aligned}
J_{GG} &= \nabla_{\theta}[\nabla_{\theta}\mathcal{L}_G(\theta; \phi)] \\
&= \nabla_{\theta}[\mathbb{E}_{p(\mathbf{x})}[h'_3(D(G(\mathbf{x}; \theta); \phi))[\nabla_{\theta}D(G(\mathbf{x}; \theta); \phi)]^T \cdot \nabla_{G(\mathbf{x})}D(G(\mathbf{x}; \theta); \phi)]] \\
&= 0
\end{aligned}
\tag{63}$$

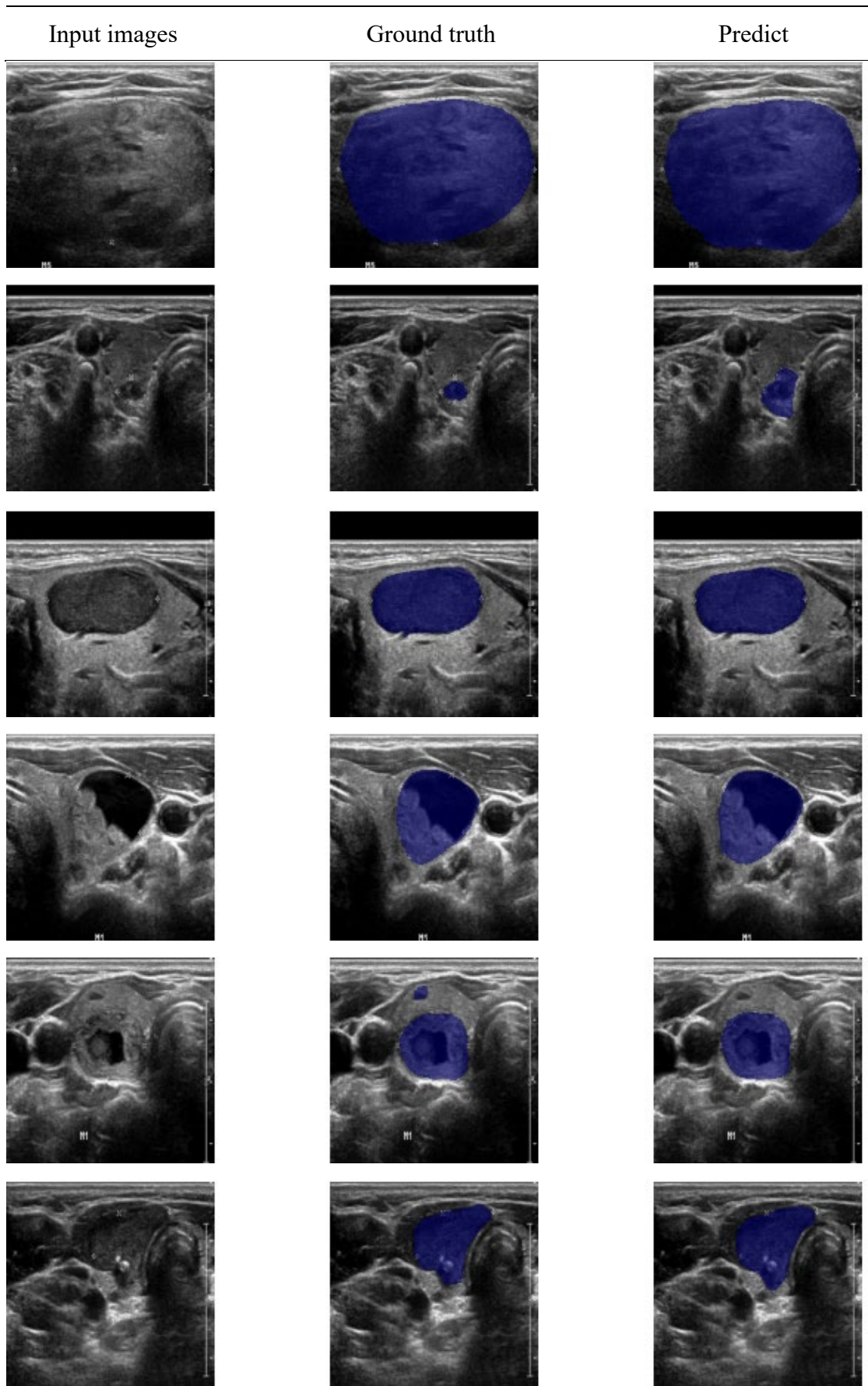
APPENDIX B

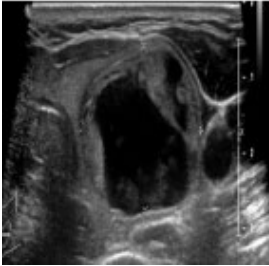
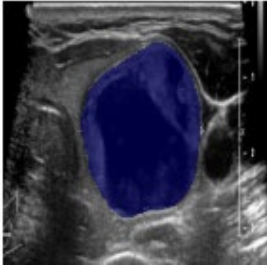
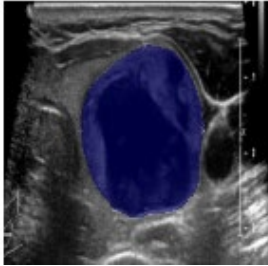

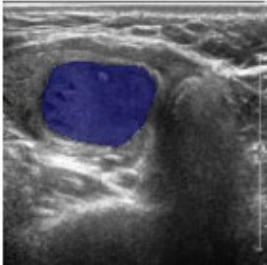
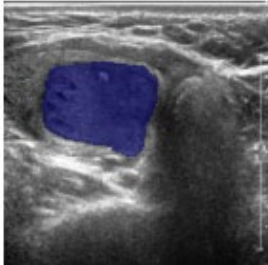

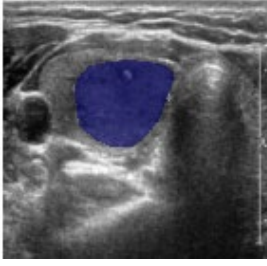
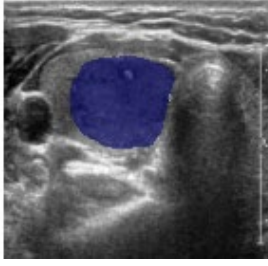

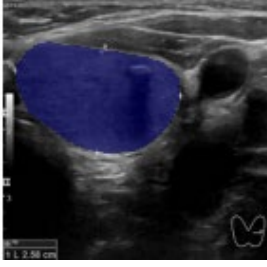
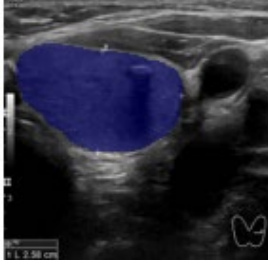

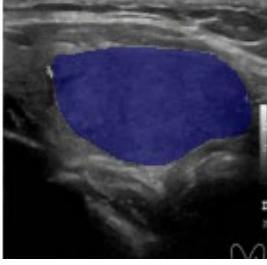
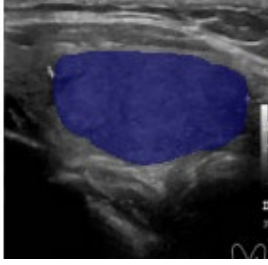

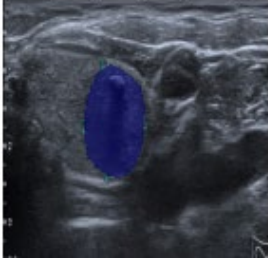

ALL TEST RESULTS IN STABLESEG GANS

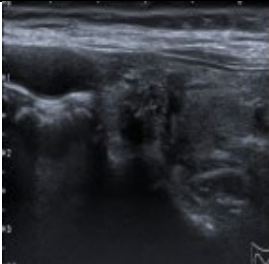
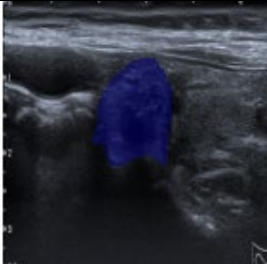
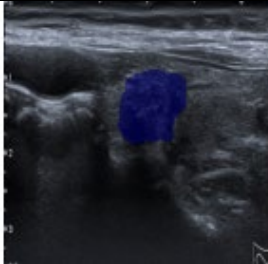
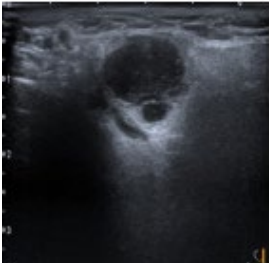

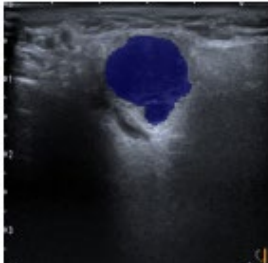
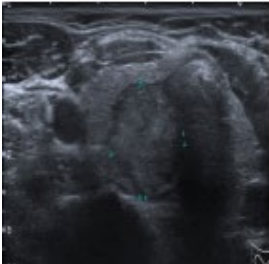
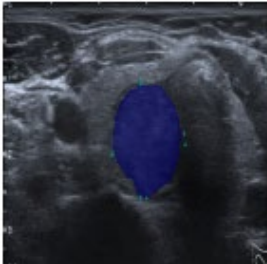
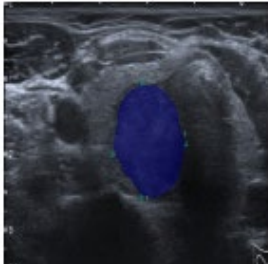
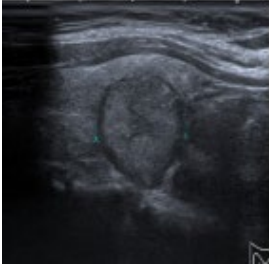
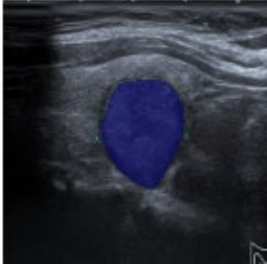
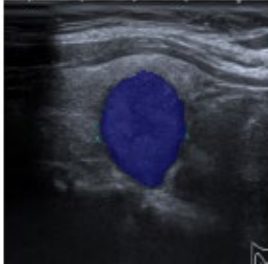
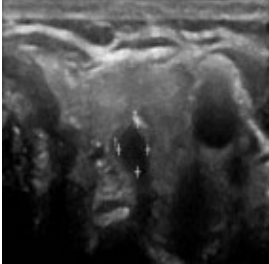
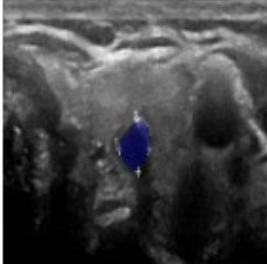
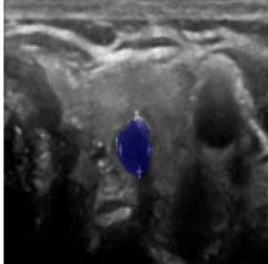
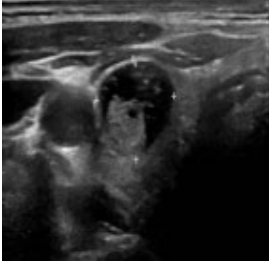
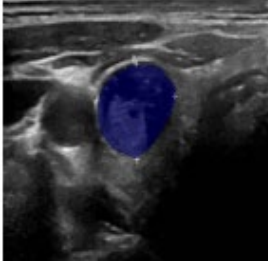
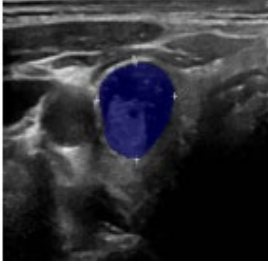
We declare all of the 149 tests set results from StableSeg into the appendix for more study. All images from left to right are the input image, the ground truth nodules mask with overlay the image, and the predicted nodules mask image.

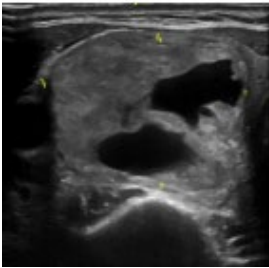
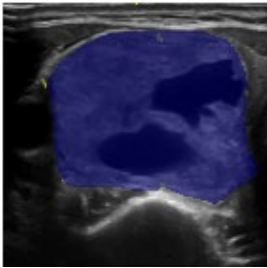
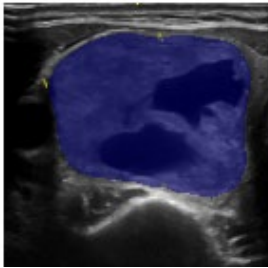
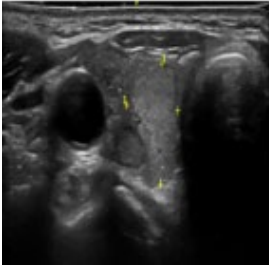
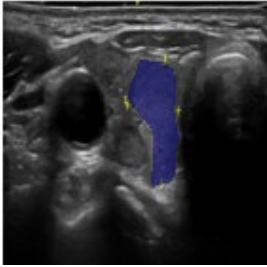
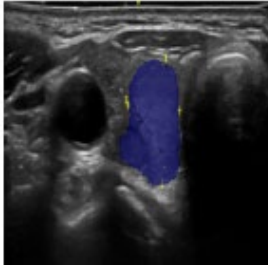
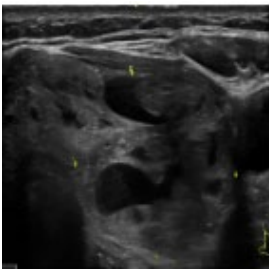
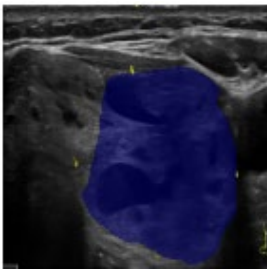
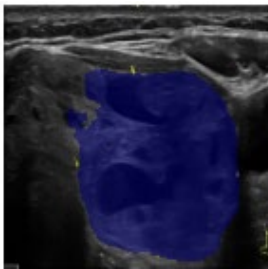
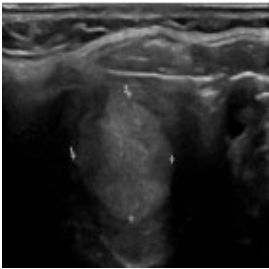
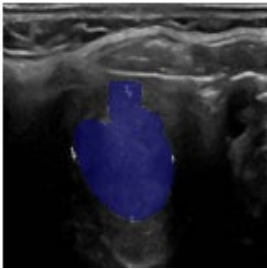
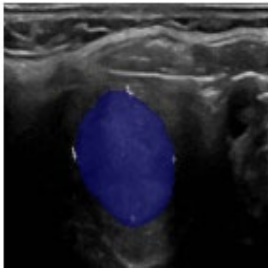
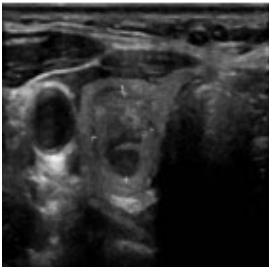
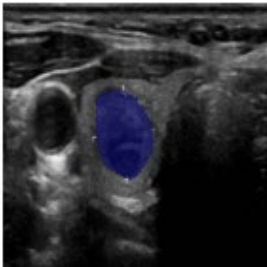
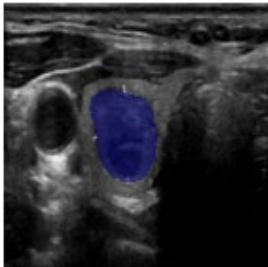
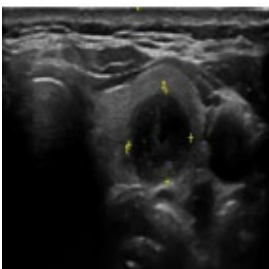
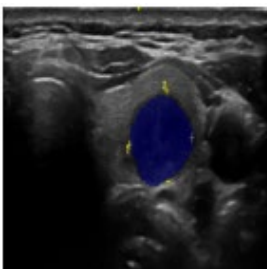
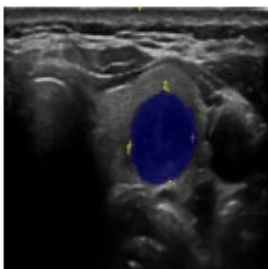


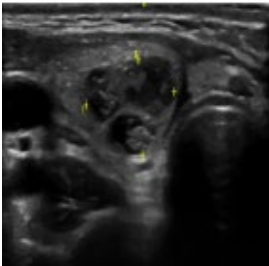
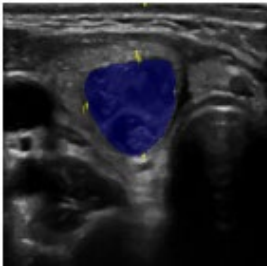
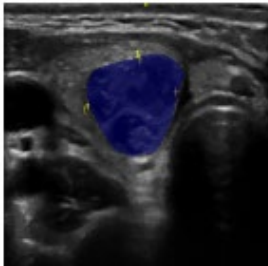
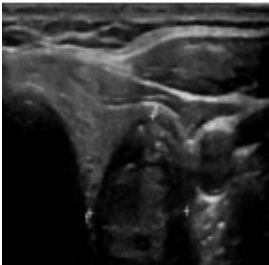
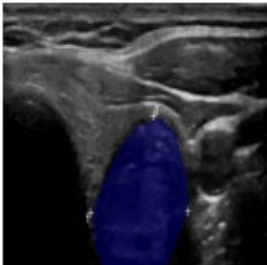
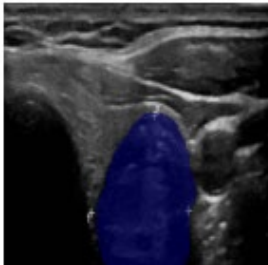
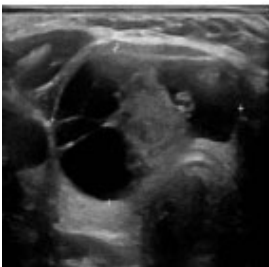
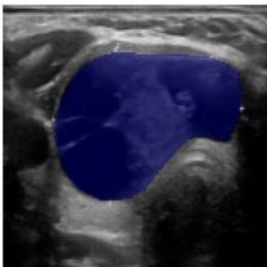
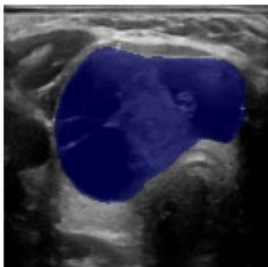
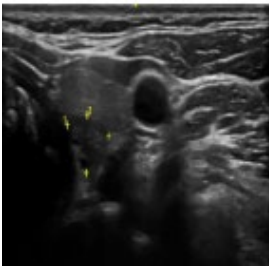
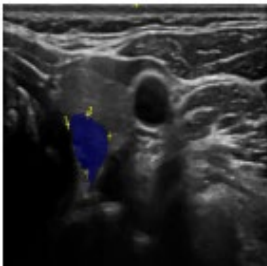
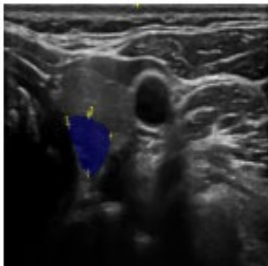
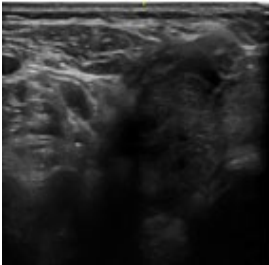
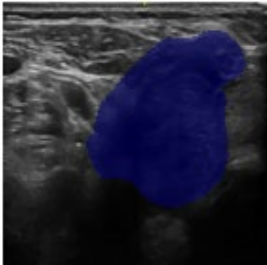
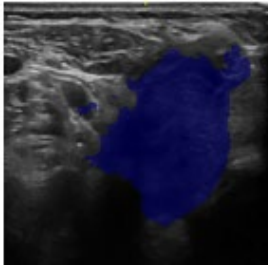


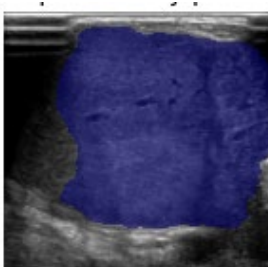


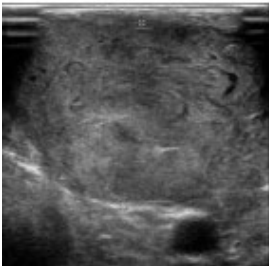
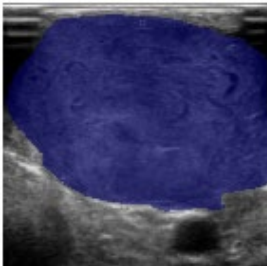
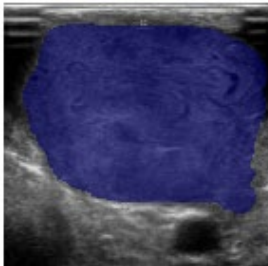










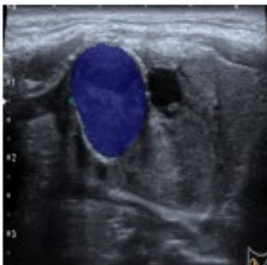
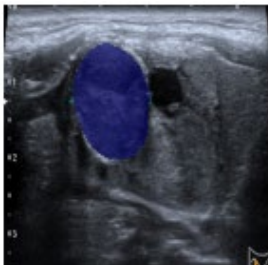
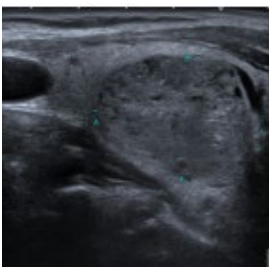
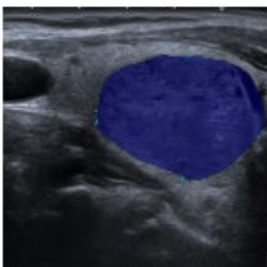
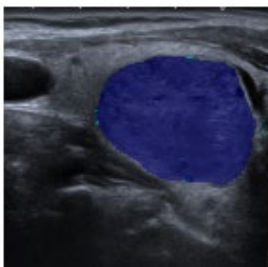


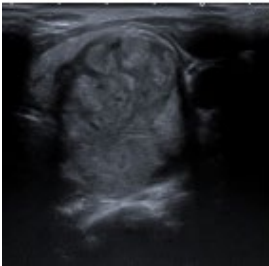
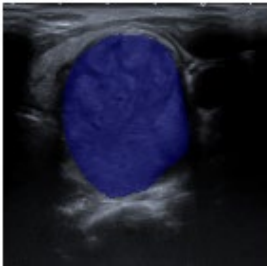
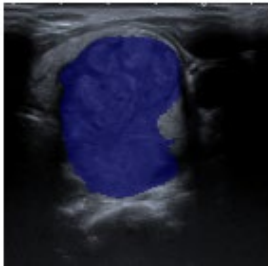
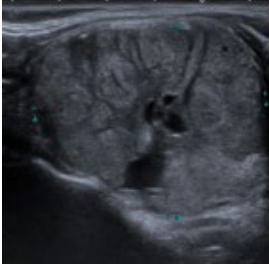
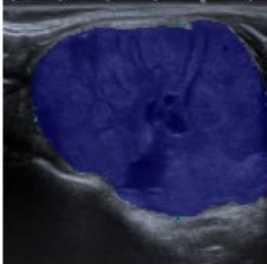
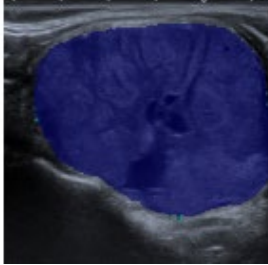
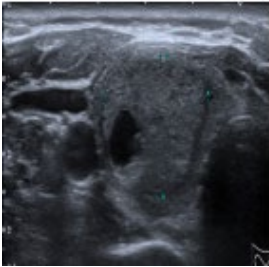
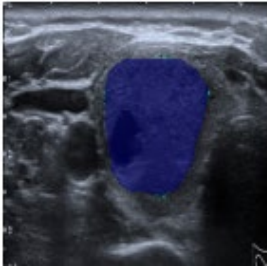
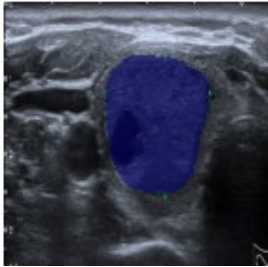
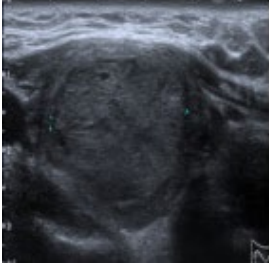
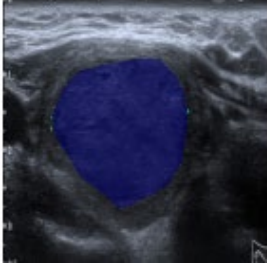
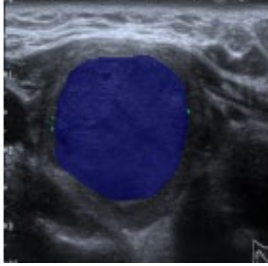


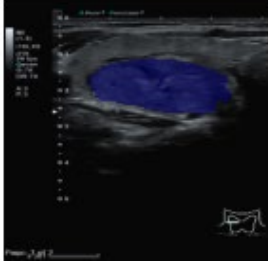

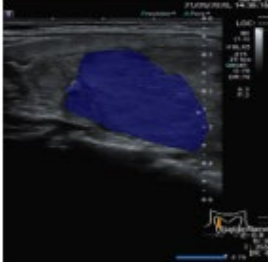
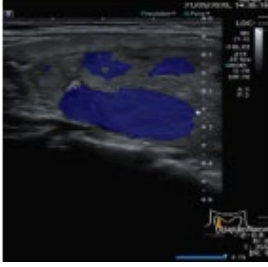
Input images	Ground truth	Predict
		
		
		
		
		
		

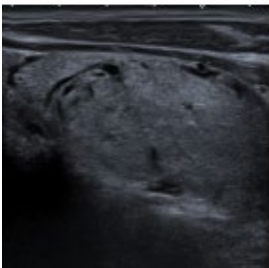
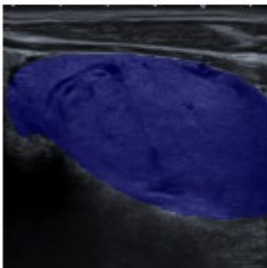
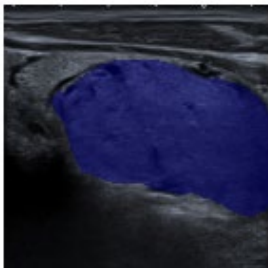
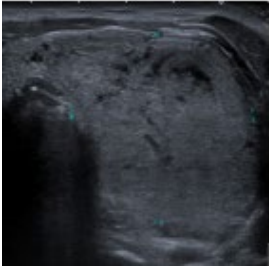
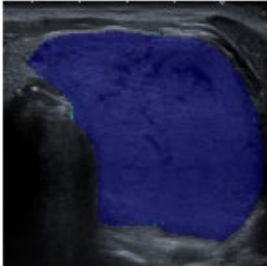
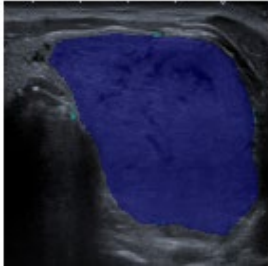

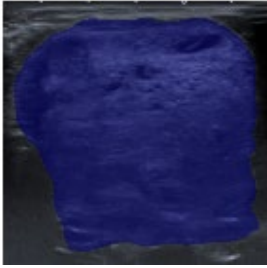
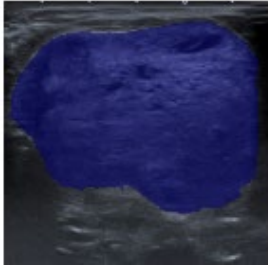
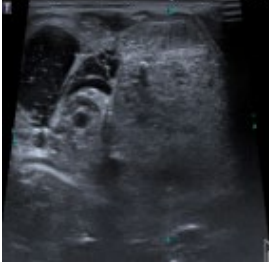

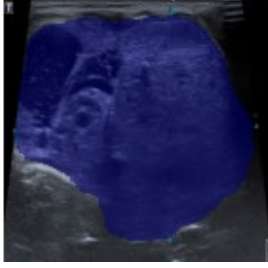
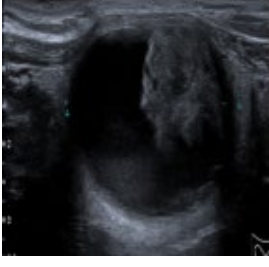
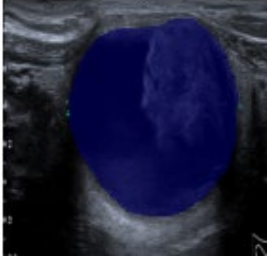
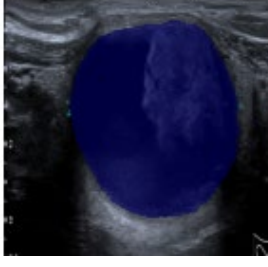
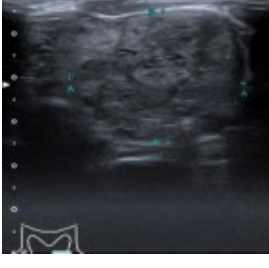
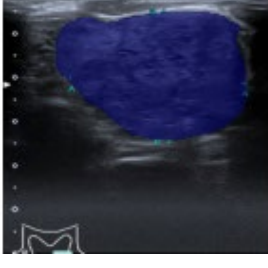

Input images	Ground truth	Predict
		
		
		
		
		
		

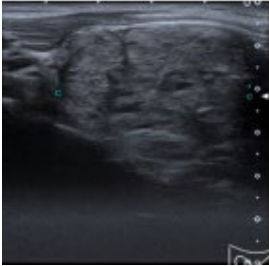
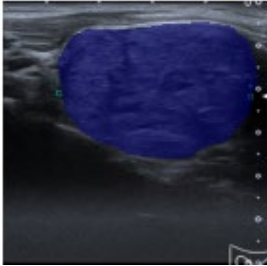
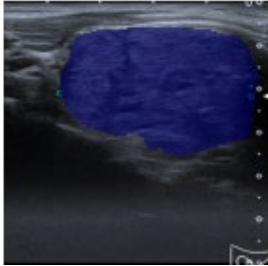
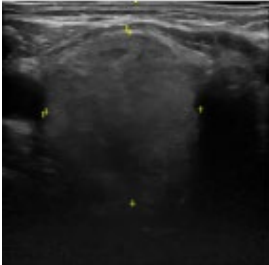
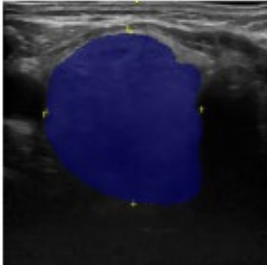
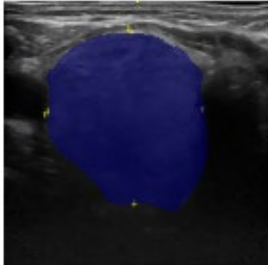
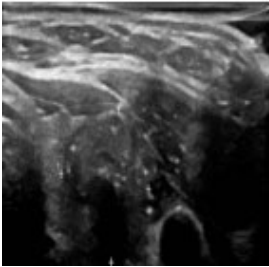
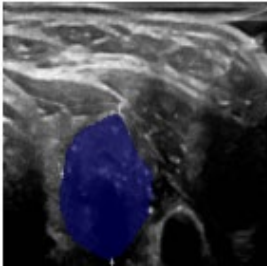
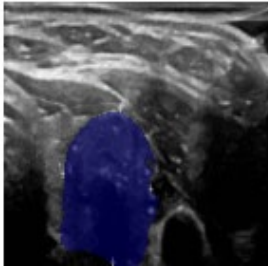
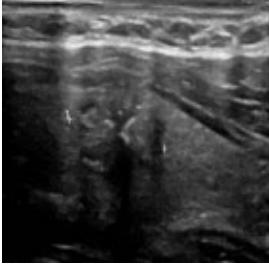
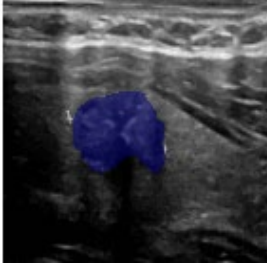
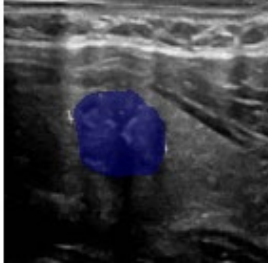
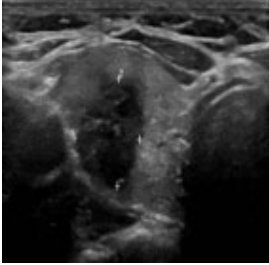
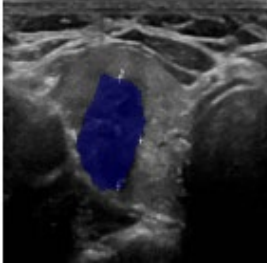
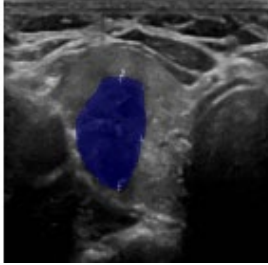
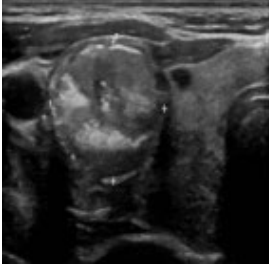
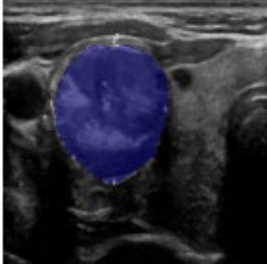
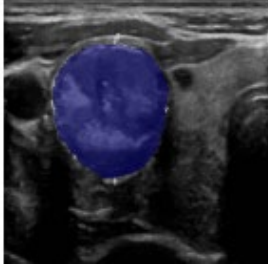
Input images	Ground truth	Predict
		
		
		
		
		
		

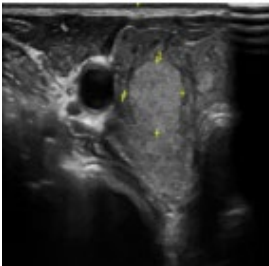
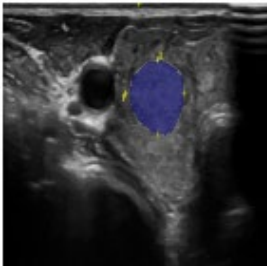
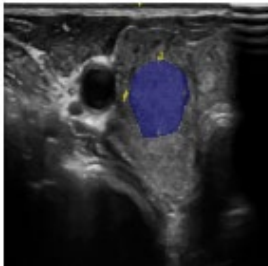
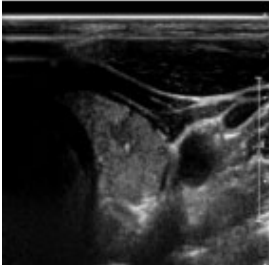
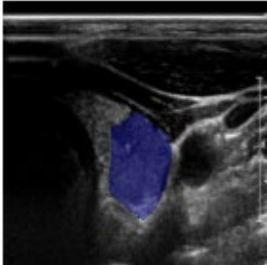
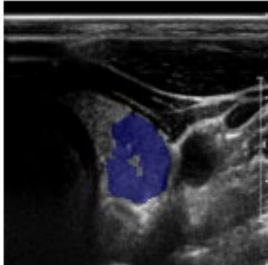
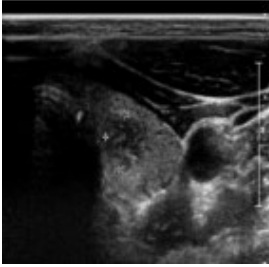
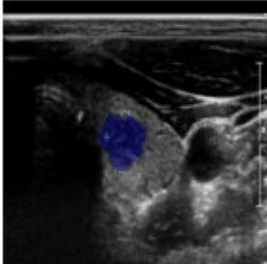
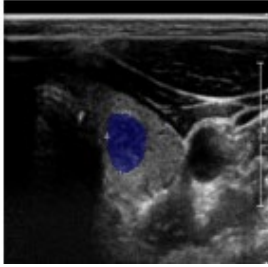

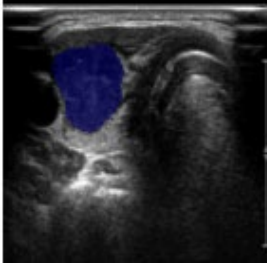
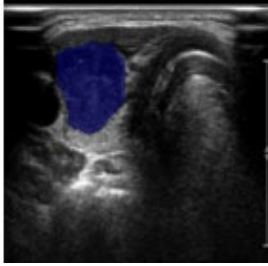
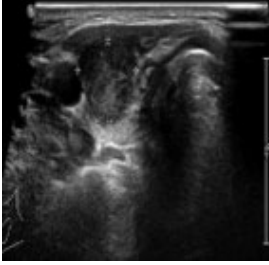
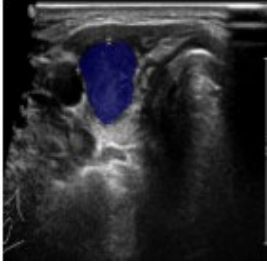
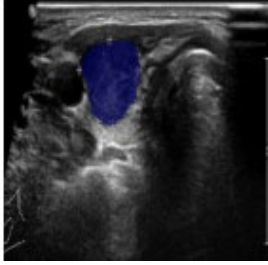
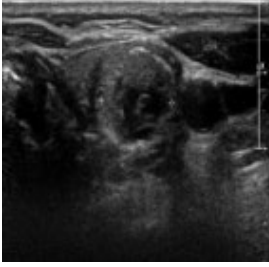
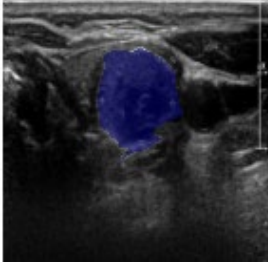
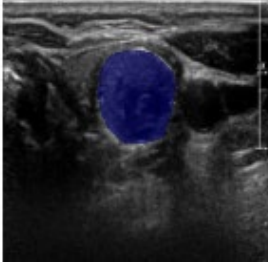
Input images	Ground truth	Predict
		
		
		
		
		
		

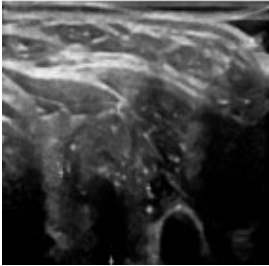
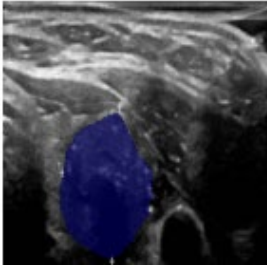
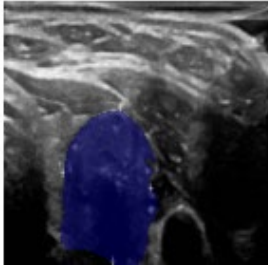
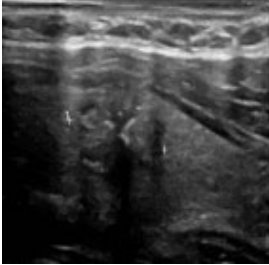
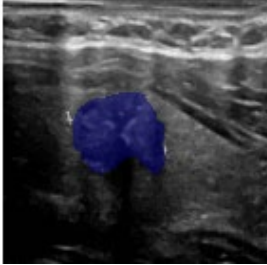
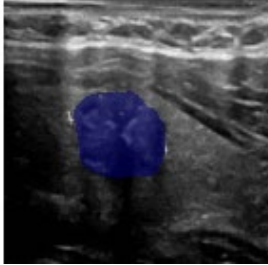
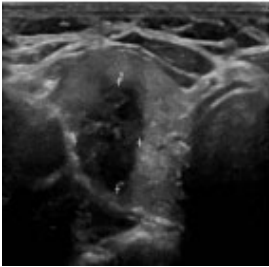
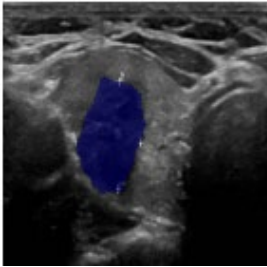
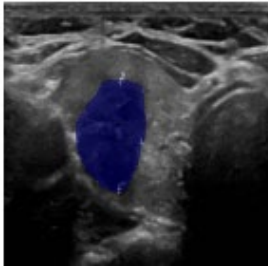
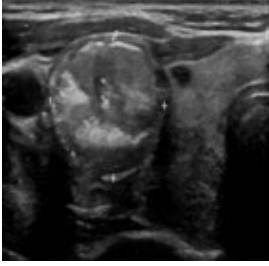
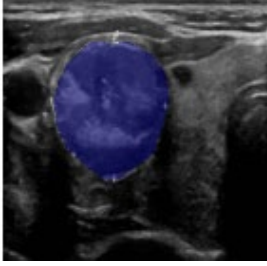
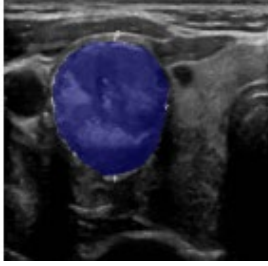
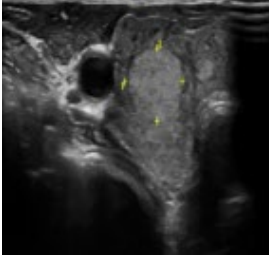
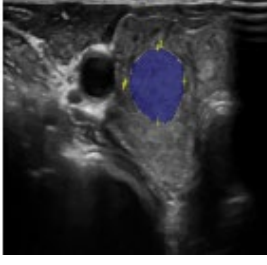
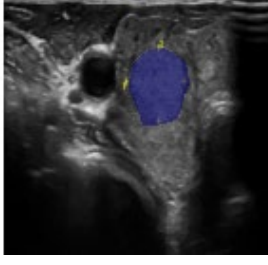
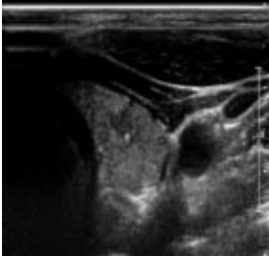
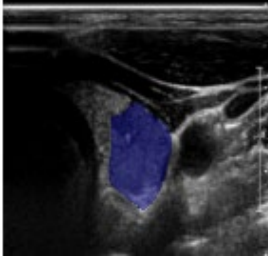
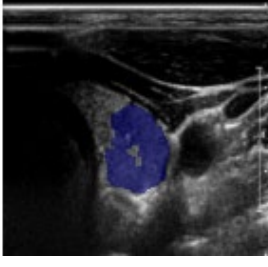
Input images	Ground truth	Predict
		
		
		
		
		
		

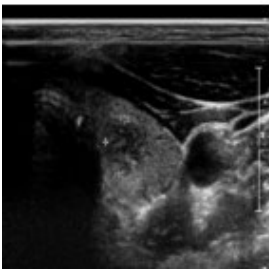
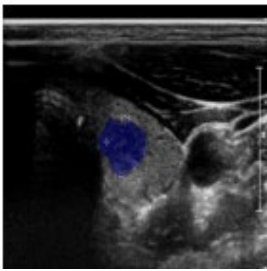
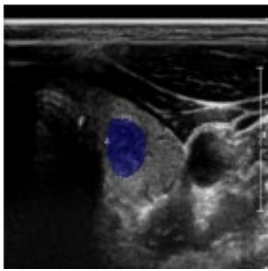

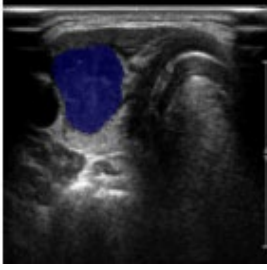
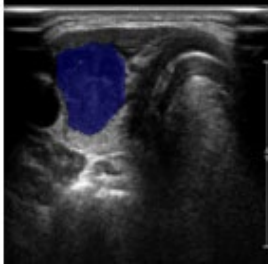
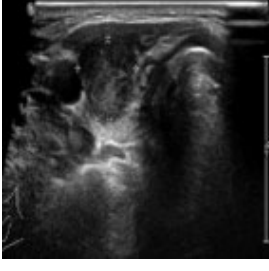
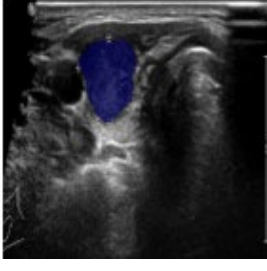
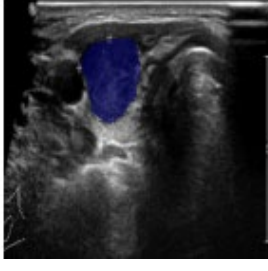

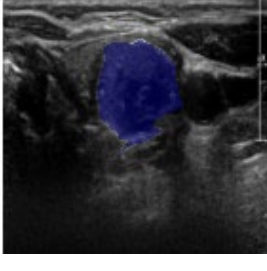
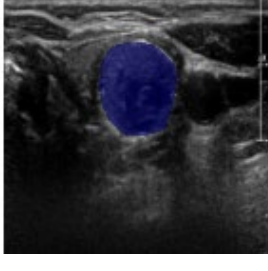


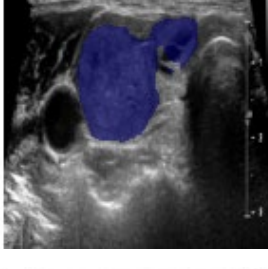
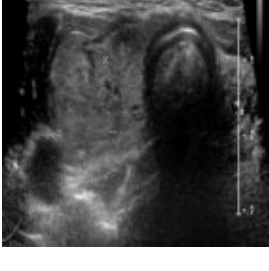
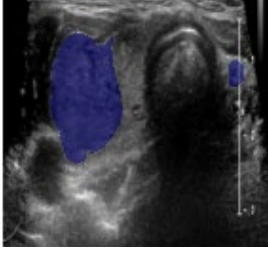
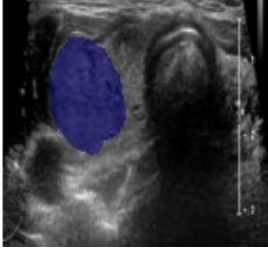
Input images	Ground truth	Predict
		
		
		
		
		
		

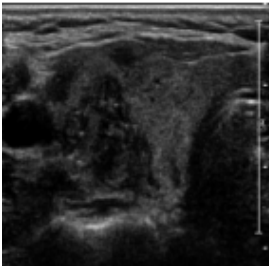
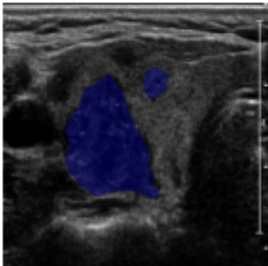
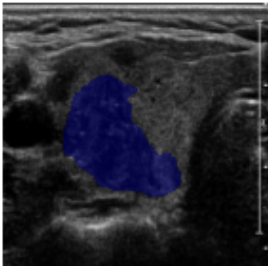
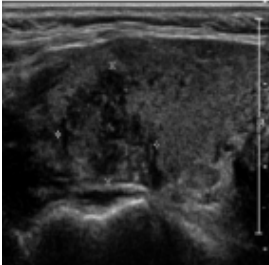
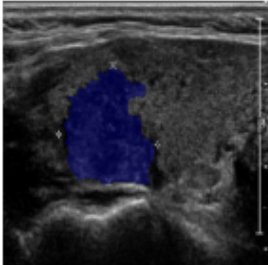
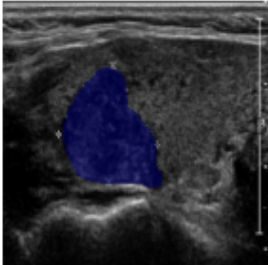
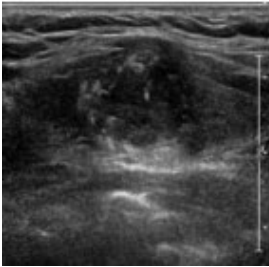
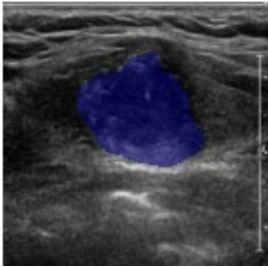
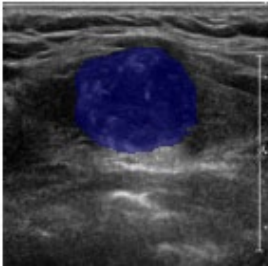
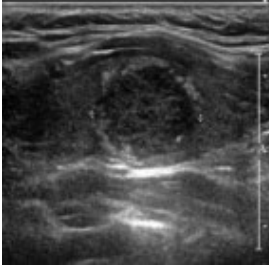
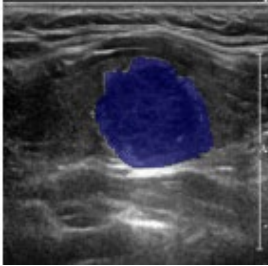
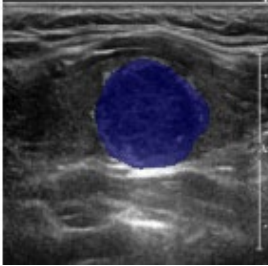
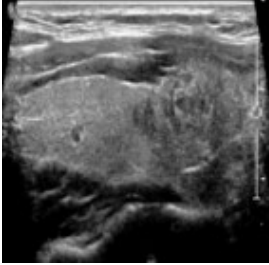
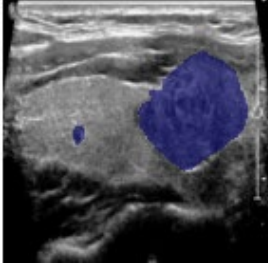
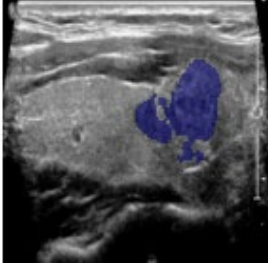
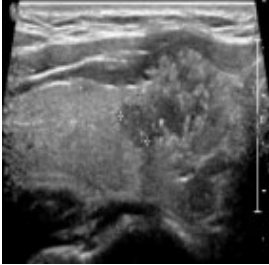
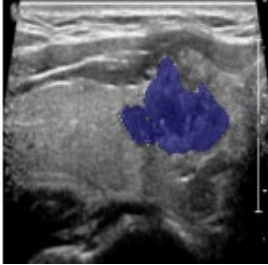
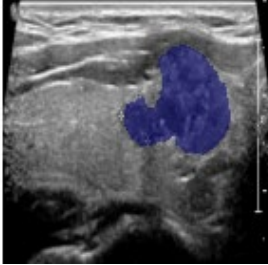
Input images	Ground truth	Predict
		
		
		
		
		
		


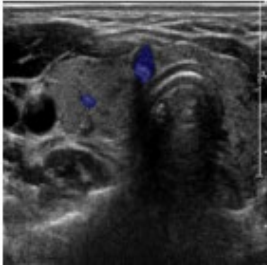
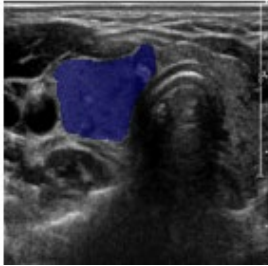
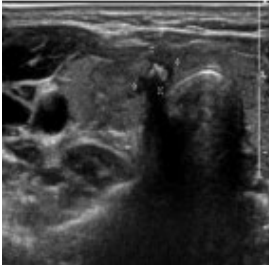
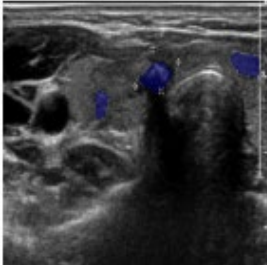
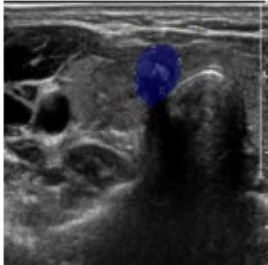

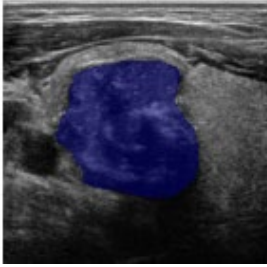
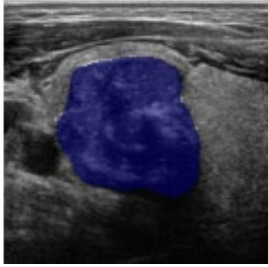
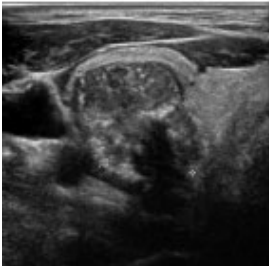
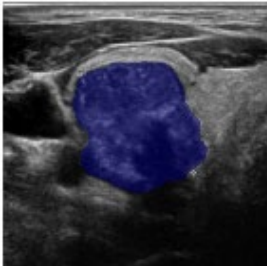
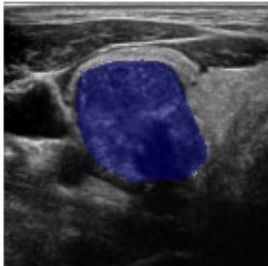
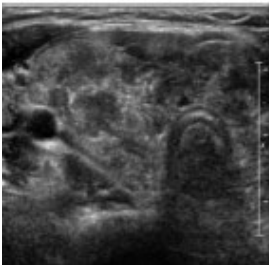
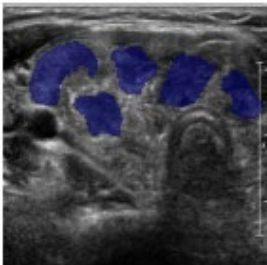
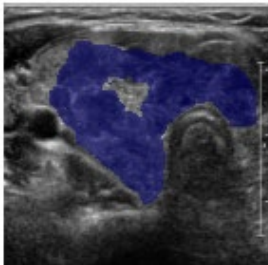
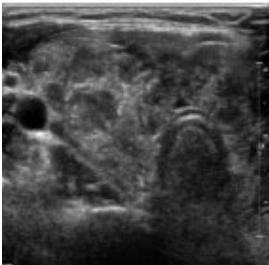
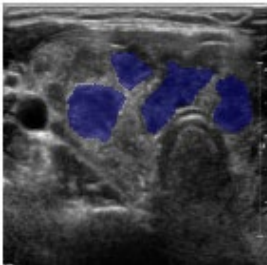
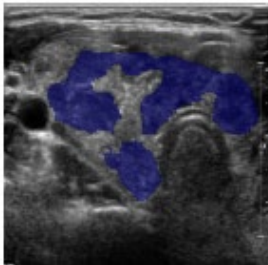
Input images	Ground truth	Predict
		
		
		
		
		
		

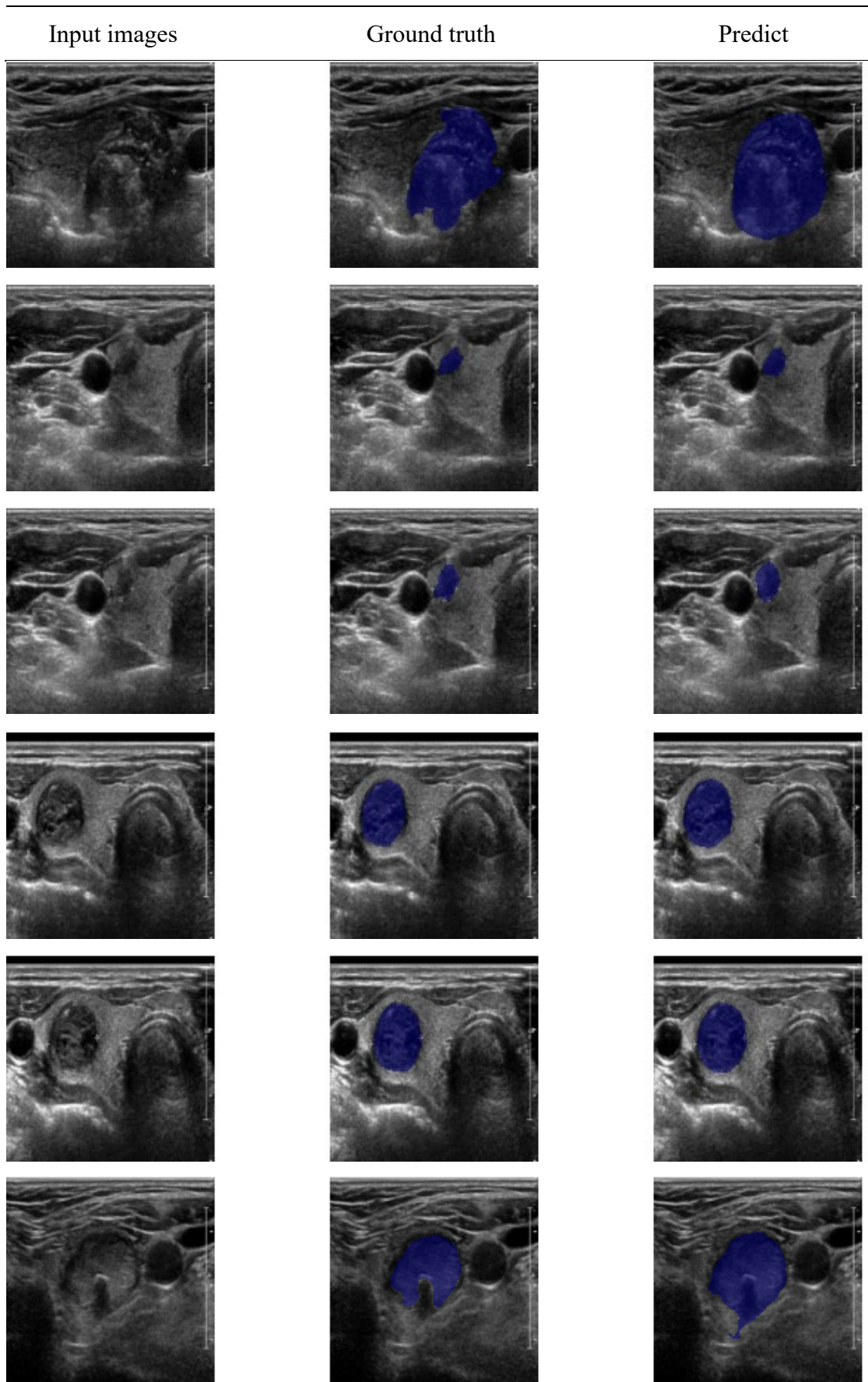
Input images	Ground truth	Predict
		
		
		
		
		
		

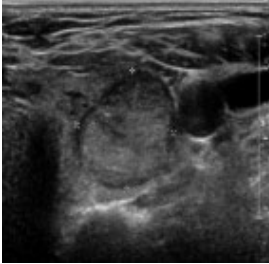
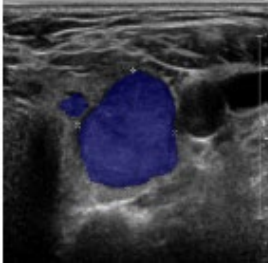
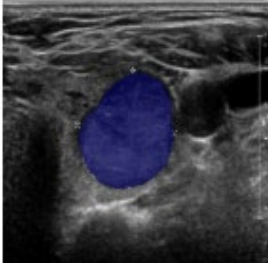
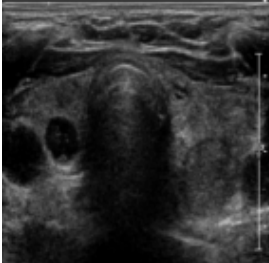
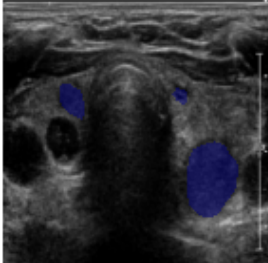
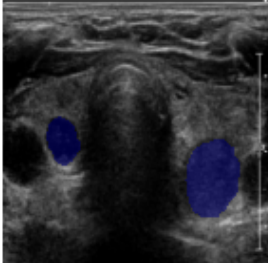
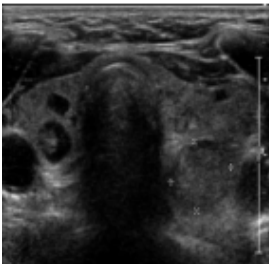
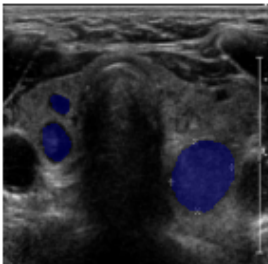
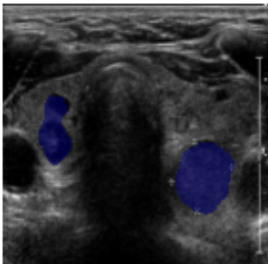
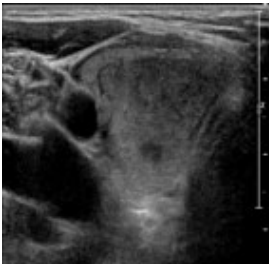
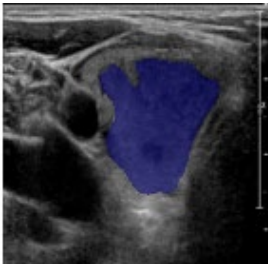
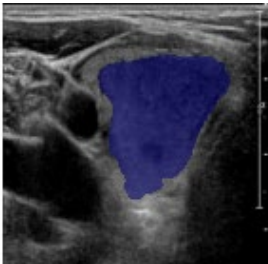
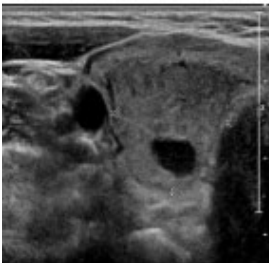
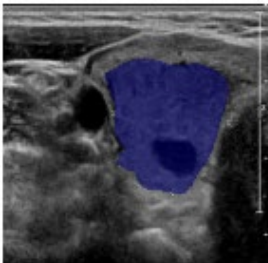
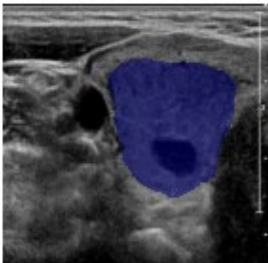
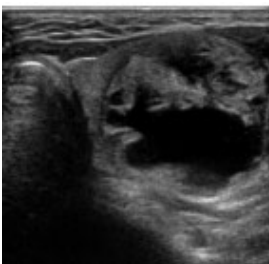
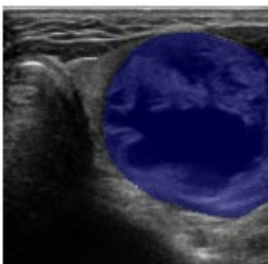
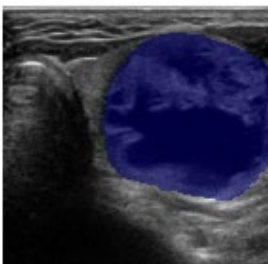
Input images	Ground truth	Predict
		
		
		
		
		
		

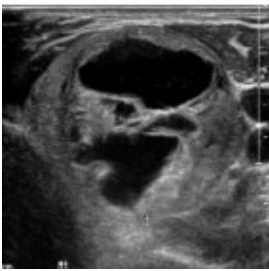
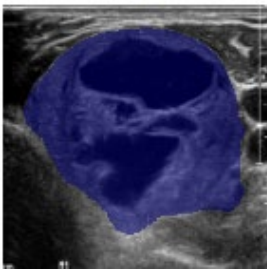
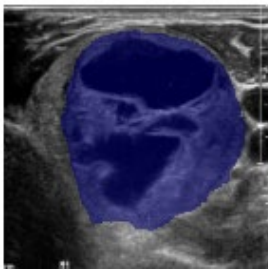
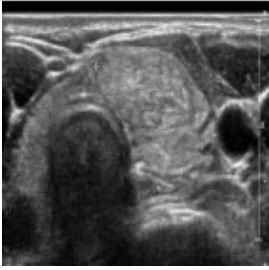
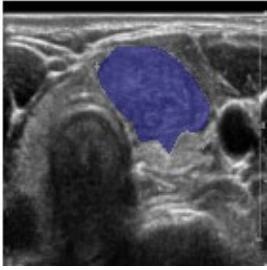
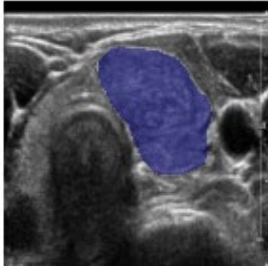
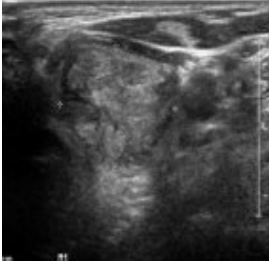
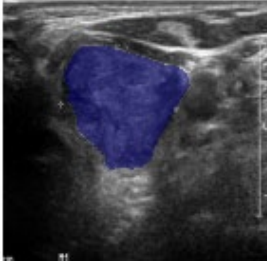
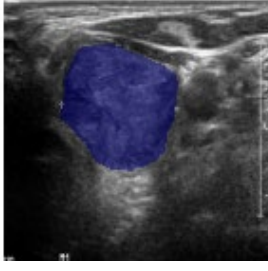
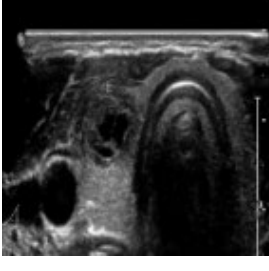
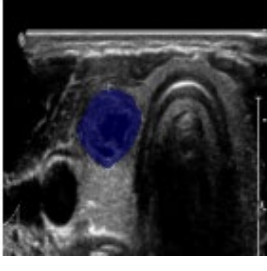
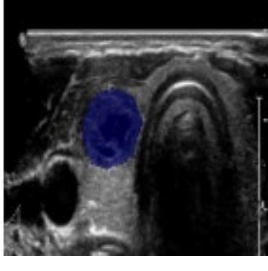

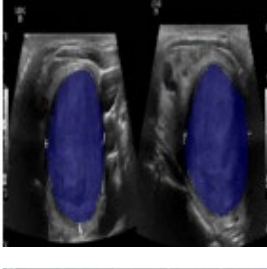
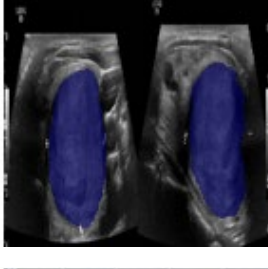

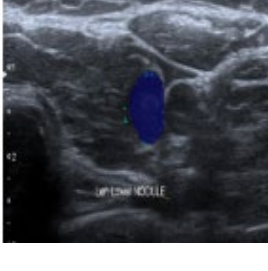
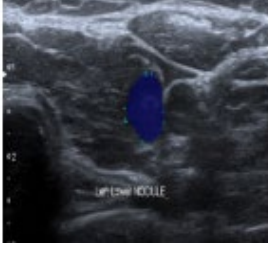
Input images	Ground truth	Predict
		
		
		
		
		
		

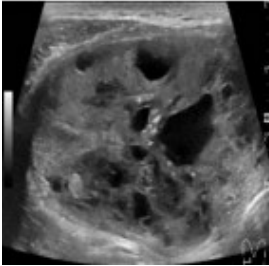
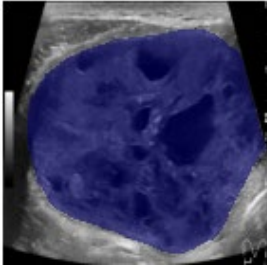
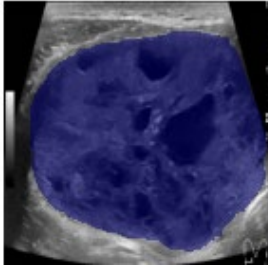
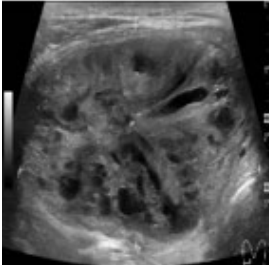
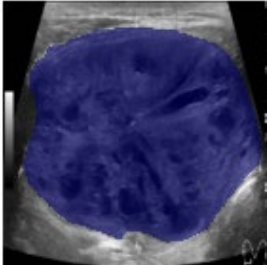
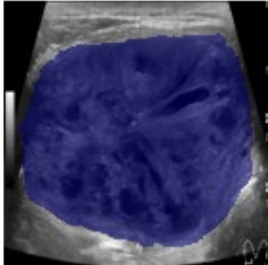
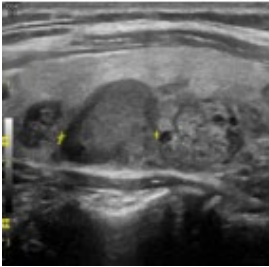
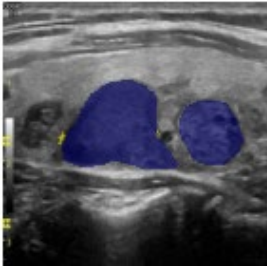
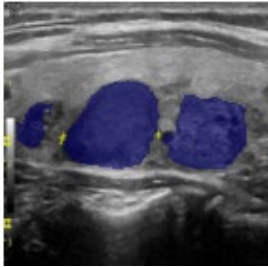
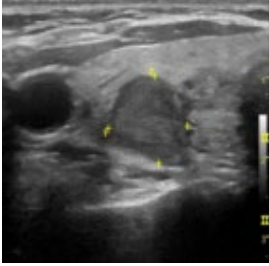
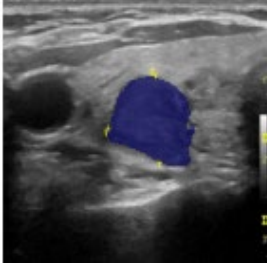
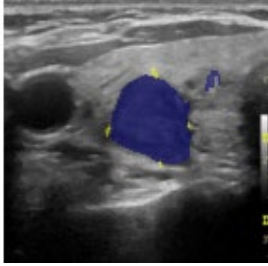

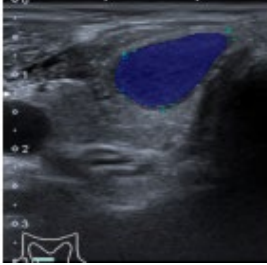

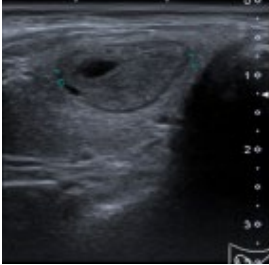
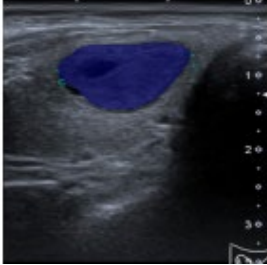
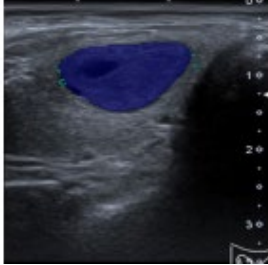
Input images	Ground truth	Predict
		
		
		
		
		
		

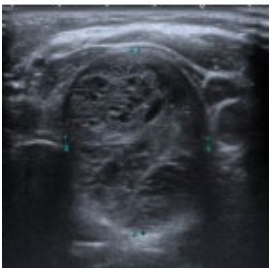
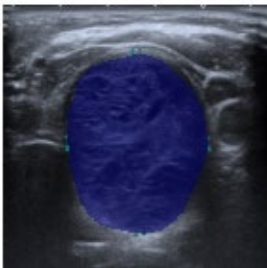
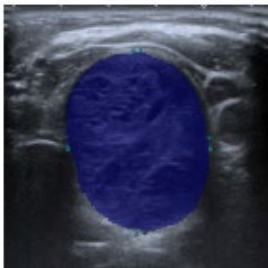
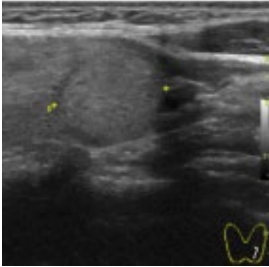
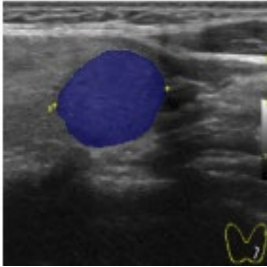
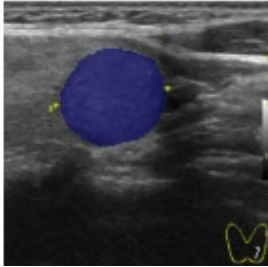
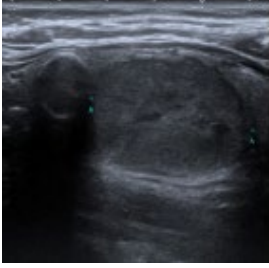
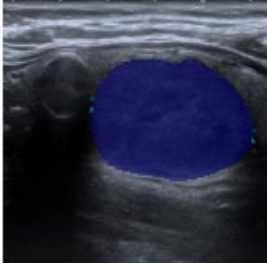
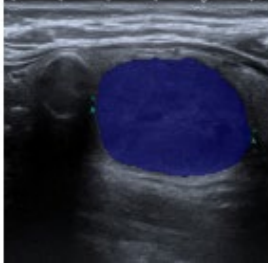



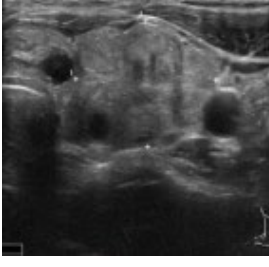
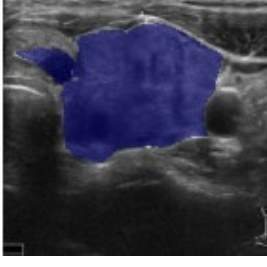
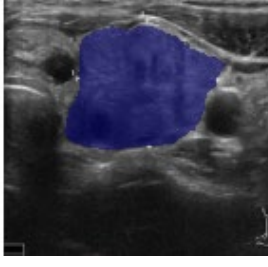
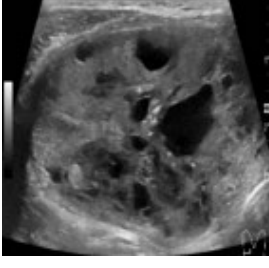
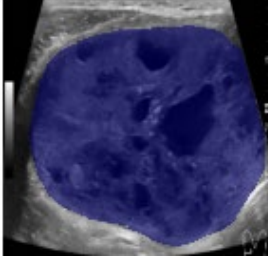
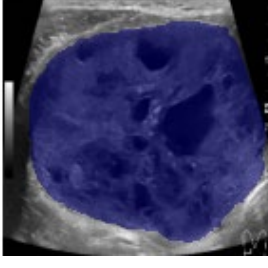
Input images	Ground truth	Predict
		
		
		
		
		
		

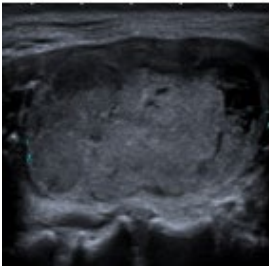
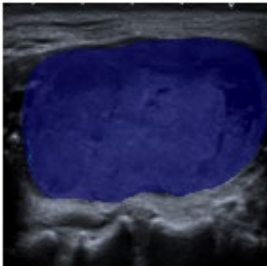
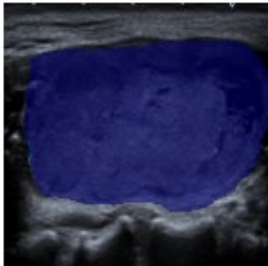
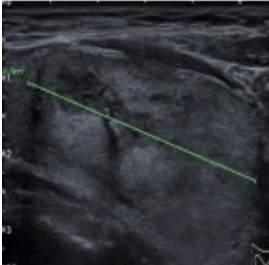
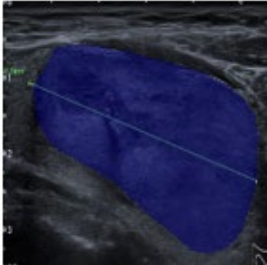
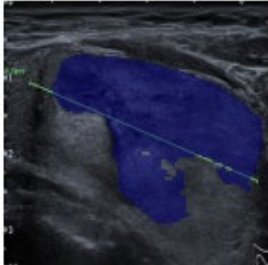
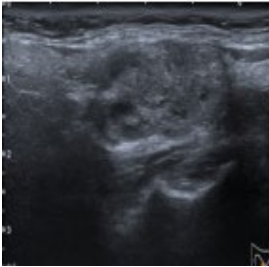

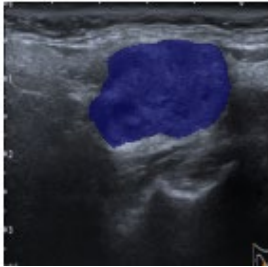
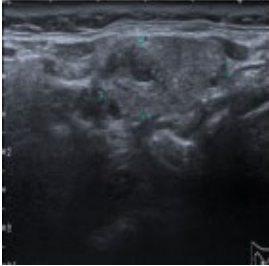
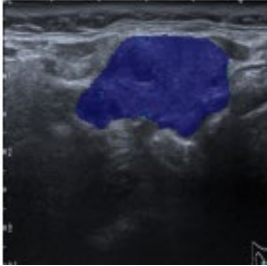
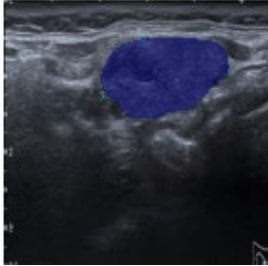
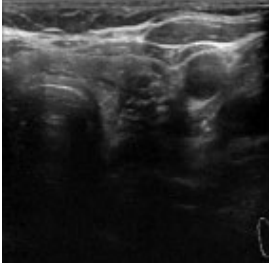
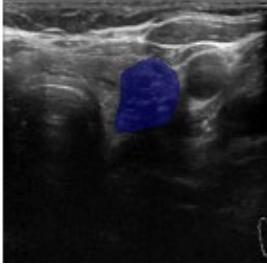
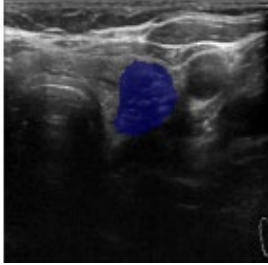
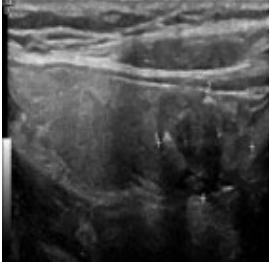
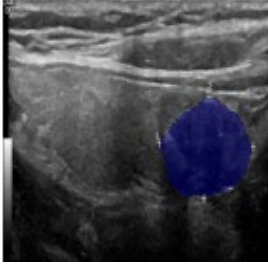
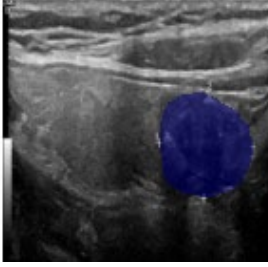


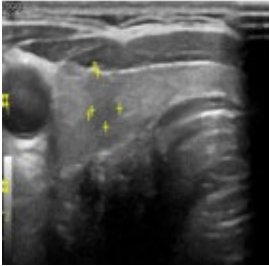
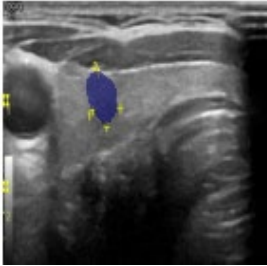
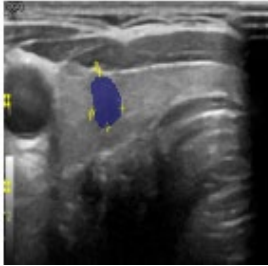
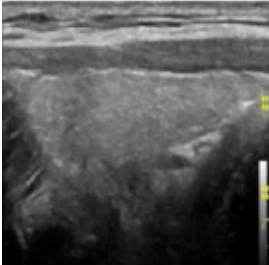
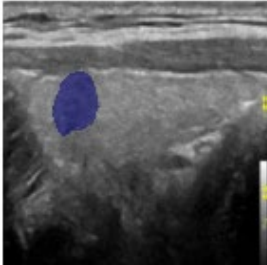
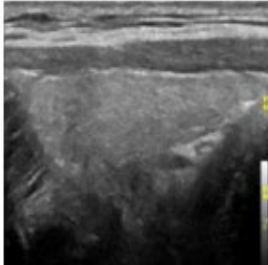
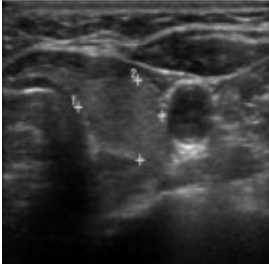
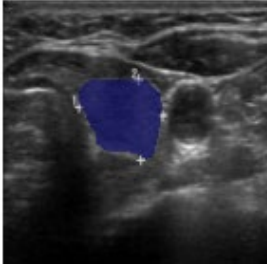
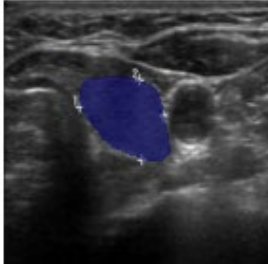
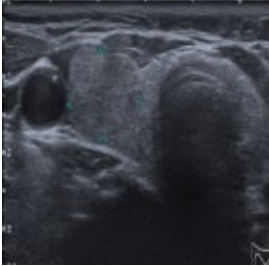
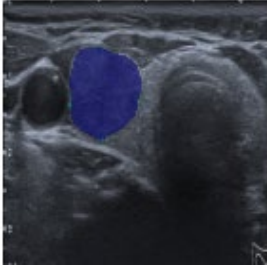
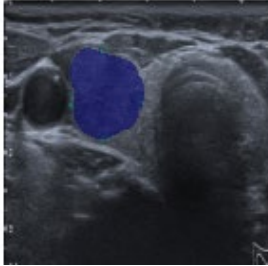
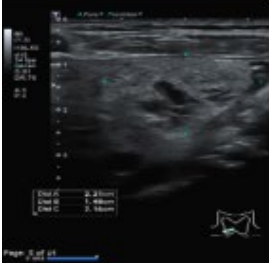
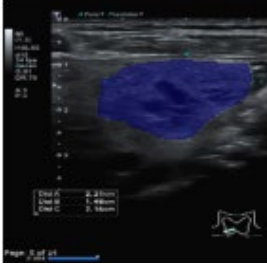



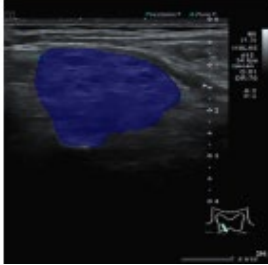
Input images	Ground truth	Predict
		
		
		
		
		
		




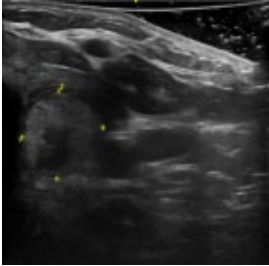
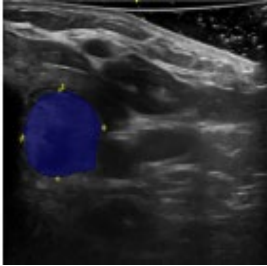
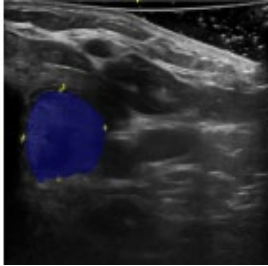
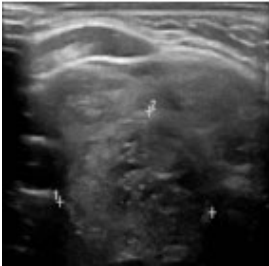
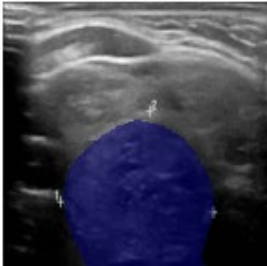
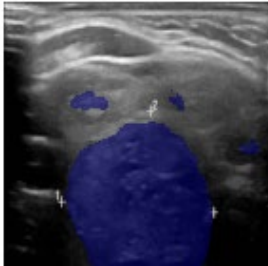
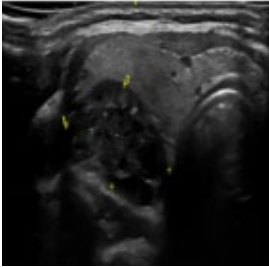
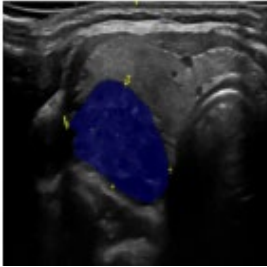
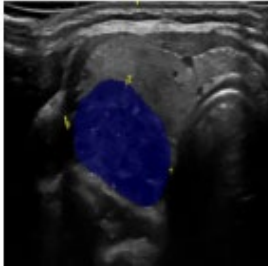
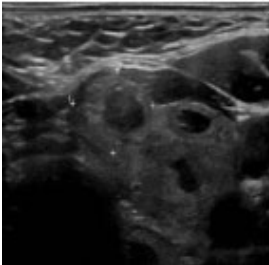
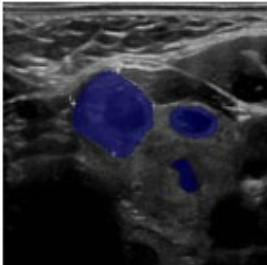
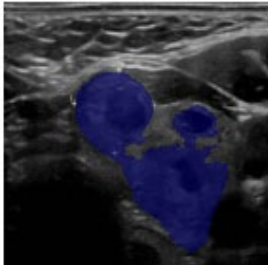
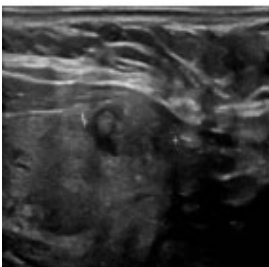
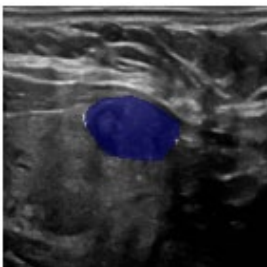
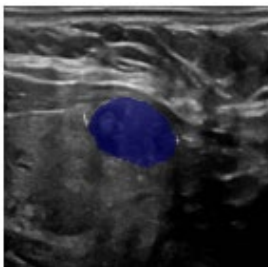
Input images	Ground truth	Predict
		
		
		
		
		
		

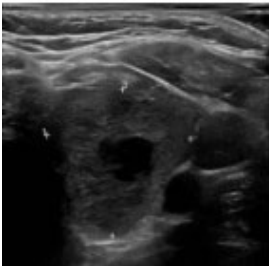
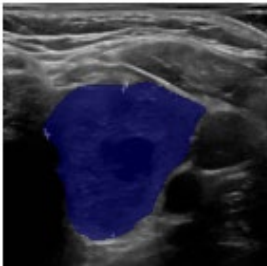
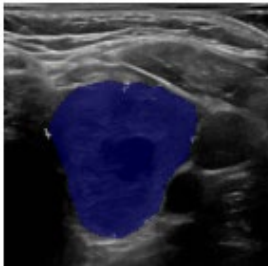
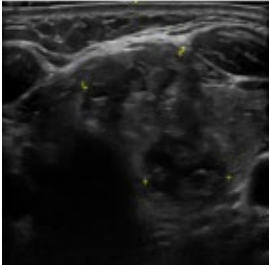
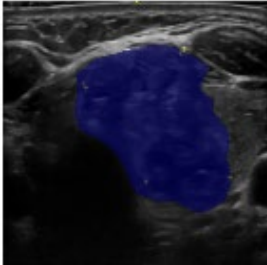
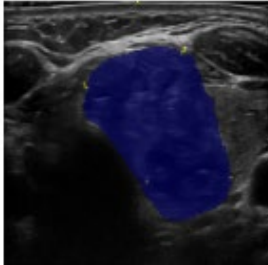
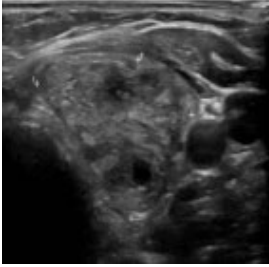
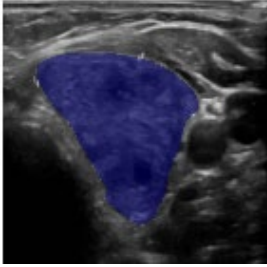
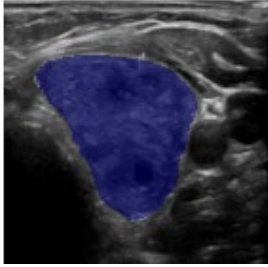
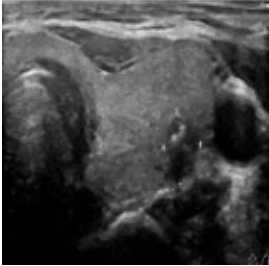
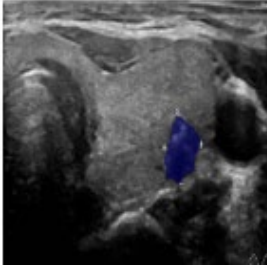
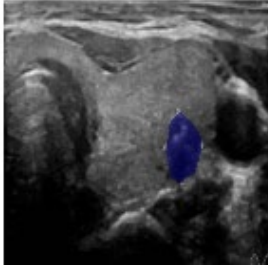
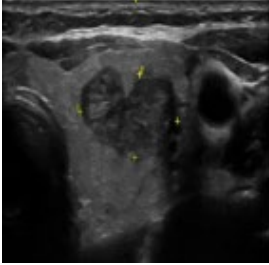
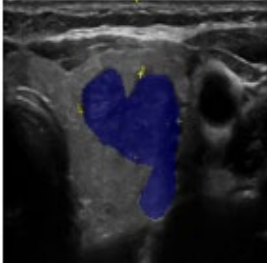
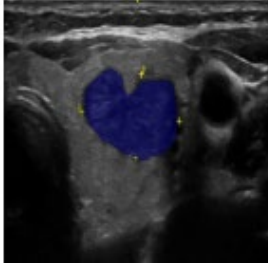
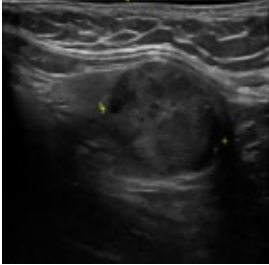
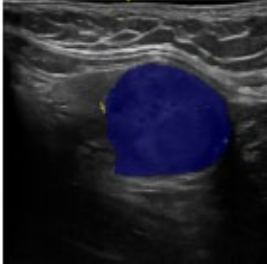
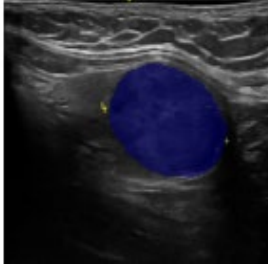
Input images	Ground truth	Predict
		
		
		
		
		
		

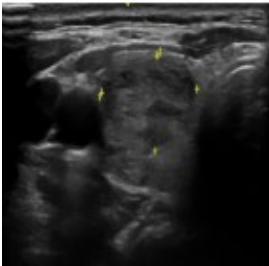
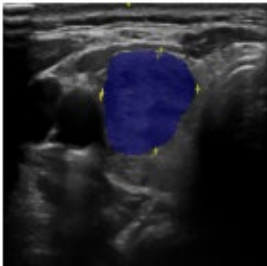
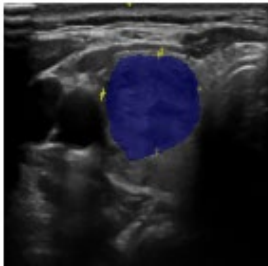
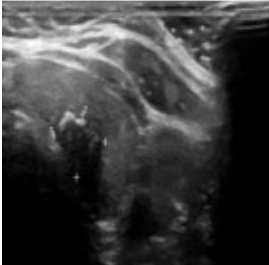
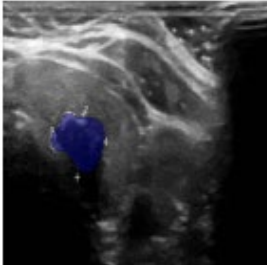
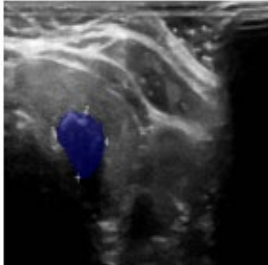
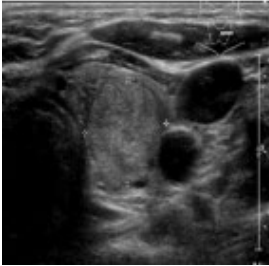
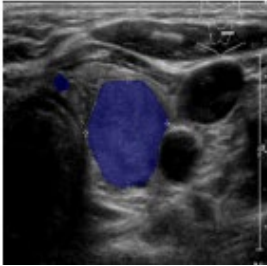
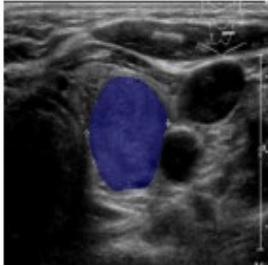

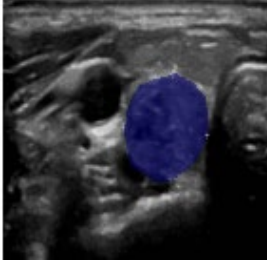
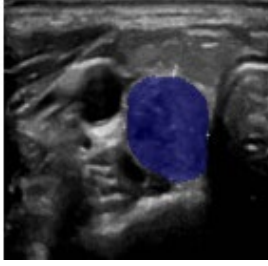

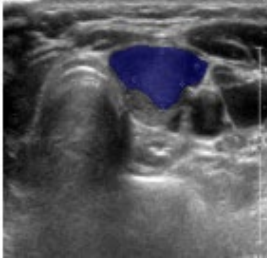
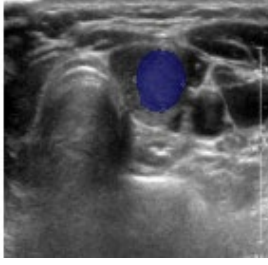
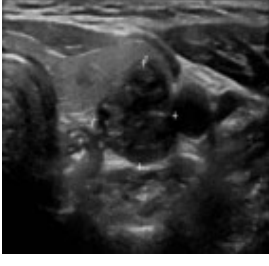
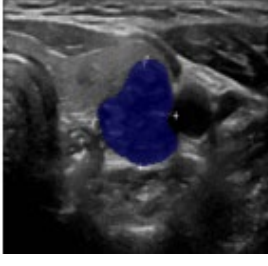
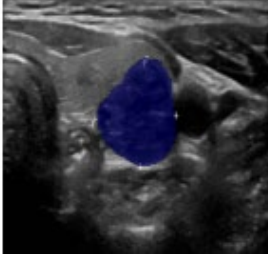
Input images	Ground truth	Predict
		
		
		
		
		
		

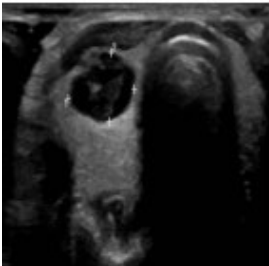
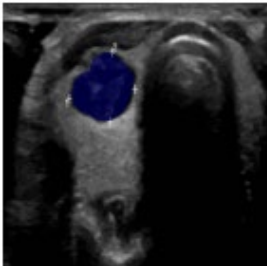
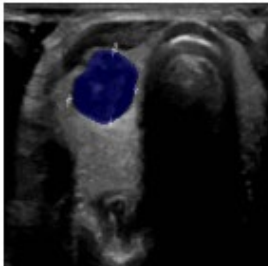
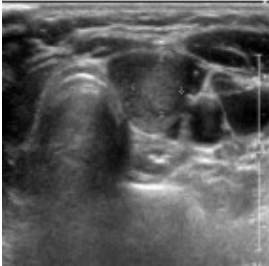
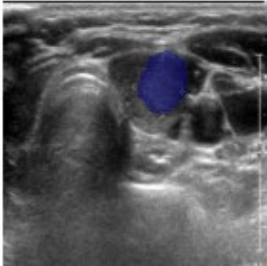
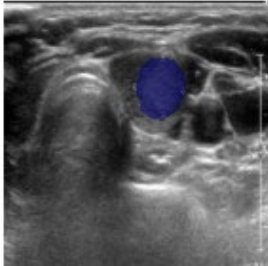
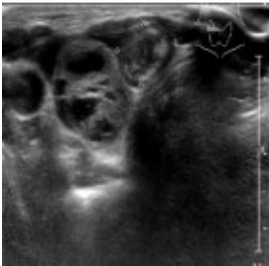
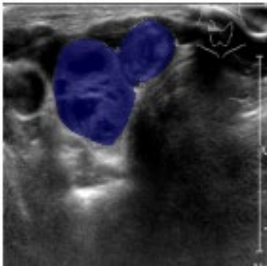
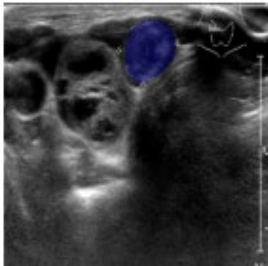

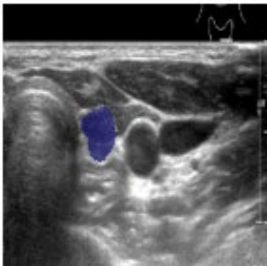
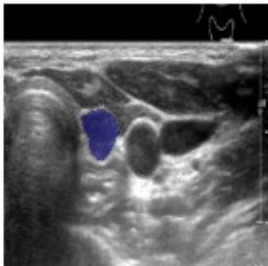
Input images	Ground truth	Predict
		
		
		
		
		
		

Input images	Ground truth	Predict
		
		
		
		
		
		

Input images	Ground truth	Predict
		
		
		
		
		
		

Input images	Ground truth	Predict
		
		
		
		
		
		

Input images	Ground truth	Predict
		
		
		
		
		
		

Input images	Ground truth	Predict
		
		
		
		

VITA

Alisa Kunapinun was born in Bangkok, Thailand. Before attending her Doctoral degree at the Asian Institute of Technology (AIT), Thailand, she worked in Cal-Comp Automation and Industrial 4.0 Co.Ltd. in the position of vice-director, and before that, she joined in Toshiba-Machine, Japan (current name is Shibaura-Machine) in the position of robot specialist. Her special skills are robotics and computer vision.

She graduated master's degree in Mechatronics from the AIT and earned a Bachelor of Engineering (Computer) from Mahidol University, Thailand.

While at AIT, Alisa worked with Prof. Matthew N. Dailey as a teaching assistant in Machine Learning, Recent Trends of Machine Learning, and Computer Vision, and she also became a teaching assistant of Assoc. Prof. Mongkol Ekpanyapong in Deep Reinforcement Learning. Moreover, she was a research assistant in the Medical Imaging Field with Ultrasound images and MRI images.