Orhan Demirci

M.Sc. Thesis

AGU 2024

TREE-NET: BOTTLENECK FEATURE SUPERVISED NETWORK FOR BIOMEDICAL IMAGE SEGMENTATION

M.Sc. THESIS
SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Orhan Demirci
July 2024

TREE-NET: BOTTLENECK FEATURE SUPERVISED NETWORK FOR BIOMEDICAL IMAGE SEGMENTATION

A THESIS
SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF
ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Orhan Demirci
July 2024

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name-Surname: Orhan Demirci
Signature :

REGULATORY COMPLIANCE

M.Sc. thesis titled "**TREE-NET: BOTTLENECK FEATURE SUPERVISED NETWORK FOR BIOMEDICAL IMAGE SEGMENTATION**" has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
Orhan Demirci	Prof. Bülent Yılmaz

Head of the Electrical and Computer Enginering Graduate Program

Assist. Prof. Samet GÜLER

ACCEPTANCE AND APPROVAL

M.Sc. thesis titled "**TREE-NET: BOTTLENECK FEATURE SUPERVISED NETWORK FOR BIOMEDICAL IMAGE SEGMENTATION**" and prepared by Orhan Demirci has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

31 /07 / 2024

JURY:

Advisor : Prof. Bülent YILMAZ

Member : Prof. Suat ÖZDEMİR

Member : Assoc. Prof. Zafer AYDIN

APPROVAL:

The acceptance of this M.Sc. has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ….. /….. / …….. and numbered ……..…… .

…….. /…….. / ……..

Graduate School Dean
Prof. Dr. İrfan Alan

ABSTRACT

TREE-NET: BOTTLENECK FEATURE SUPERVISED NETWORK FOR BIOMEDICAL IMAGE SEGMENTATION

Orhan Demirci
MSc. in Electrical and Computer Engineering
Supervisor: Prof. Bülent YILMAZ
July 2024

In this thesis, we introduce Tree-NET, a novel approach for medical image segmentation utilizing bottleneck feature supervision. This method enhances traditional segmentation algorithms by keeping supervision between bottleneck features of the network. The primary goal is to improve the model's ability to learn discriminative and robust features while simultaneously reducing computational costs. Bottleneck feature supervision involves compressing the input and label data using Autoencoders and then supervising the bottleneck features with a segmentation network named "Bridge-Net," which can be any segmentation model of choice. We applied Tree-NET to two critical medical image segmentation tasks: skin lesion segmentation and polyp segmentation. Our experiments demonstrate significant improvements in segmentation accuracy and efficiency. For instance, the U-NET backboned Tree-NET uses only 154.43 MB for executing and storing the model, which is almost 3.5 times smaller than the original U-Net while having a close number of trainable parameters. In skin lesion segmentation, Tree-NET achieved dice, Intersection-over-Union (IoU), and accuracy scores of 0.893, 0.751, and 0.977 respectively. For polyp segmentation, the scores were 0.856, 0.795, and 0.923 for dice, IoU, and accuracy respectively. Compared to traditional segmentation models, the empirical results show that Tree-NET achieves higher accuracy with reduced training time and computational cost, thus representing a significant advancement in medical image analysis by providing more reliable and efficient tools for clinical applications.

Keywords: Autoencoders, Bottleneck Feature Supervision, Medical Image Segmentation, Tree-NET, Self-tuning

ÖZET

TREE-NET: BİYOMEDİKAL GÖRÜNTÜ SEGMENTASYONU İÇİN DARBOĞAZ ÖZELLİK SÜPERVİZYONU KULLANILAN YAPAY SİNİR AĞI MODELİ

Orhan Demirci
Elektrik ve Bilgisayar Mühendisliği Yüksek Lisans Programı
Tez Yöneticisi: Prof. Dr. Bülent YILMAZ
Temmuz-2024

Bu tezde, tıbbi görüntü segmentasyonu için yeni bir yaklaşım olan Tree-NET tanıtılmaktadır. Bu yöntem, geleneksel kodlayıcı-çözücü mimarilerini, ağ içerisindeki ara özellik temsillerine veya dar boğazlara gözetim ekleyerek geliştirir. Ana hedef, modelin ayırt edici ve sağlam özellikler öğrenme yeteneğini artırırken aynı zamanda hesaplama maliyetlerini ve eğitim süresini azaltmaktır. Dar boğaz özellik gözetimi, girdileri ve etiket verilerini Otomatik Kodlayıcılar kullanarak sıkıştırır ve ardından "Bridge-Net" olarak adlandırılan bir segmentasyon ağı ile dar boğaz özelliklerini denetler, bu ağ temelde herhangi bir segmentasyon modeli olabilir. Bu yaklaşım, daha az hesaplama kaynağı tüketimi, daha yüksek doğruluk sonuçları sağlarken, neredeyse aynı toplam eğitilebilir parametre sayısını korur. Tree-NET'i iki kritik tıbbi görüntü segmentasyonu görevi olan cilt lezyon segmentasyonu ve polip segmentasyonuna uyguladık. Deneylerimiz, segmentasyon doğruluğunda ve verimliliğinde önemli iyileşmeler göstermektedir. Örneğin, U-NET tabanlı Tree-NET, modeli çalıştırmak ve depolamak için yalnızca 154.43 MB kullanır, bu da orijinal U-NET mimarisinden neredeyse 3.5 kat daha küçüktür. Cilt lezyon segmentasyonunda, Tree-NET sırasıyla 0.8938, 0.7518 ve 0.9776 dice, IoU ve doğruluk seviyelerine ulaşmıştır. Polip segmentasyonunda ise, dice, Intersection-over-Union (IoU) ve doğruluk için sırasıyla 0.8564, 0.7956 ve 0.9236 puanlarına ulaşmıştır. Geleneksel segmentasyon modelleriyle karşılaştırıldığında, ampirik sonuçlar Tree-NET'in daha yüksek doğruluk, daha düşük eğitim süresi ve hesaplama maliyeti ile daha yüksek performans gösterdiğini, bu nedenle tıbbi görüntü analizinde önemli bir ilerlemeyi temsil ettiğini ve klinik uygulamalar için daha güvenilir ve verimli araçlar sağladığını göstermektedir.

Anahtar Kelimeler: Oto-kodlayıcılar, Darboğaz Özellik Süpervizyonu, Medikal Görüntü Segmentasyonu, Tree-NET

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Bülent Yılmaz, for his support and guidance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CRC	Colorectal cancer
BS U-NET	Bottleneck Supervised U-NET
DL	Deep Learning
MSE	Mean Squared Error
CNN	Convolutional Neural Network
BCE	Binary Cross Entropy
VGG	Visual Geometry Group
IoU	Intersection-over-Union
DSC	Dice Similarity Coefficient
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
BB	Backbone
ST	Self-Tuned
ACC	Accuracy
PO	Percentage Overfitting

To my grandfather who was a great scientist

Chapter 1

Introduction

Variants of U-NET [1, 2] and fully convolutional networks (FCN) [3], built on the encoder-decoder architecture are the most widely used models used for medical image segmentation. These networks utilize skip connections that integrate deep, semantic, coarse-grained feature maps from the decoder with fine-grained, low-level feature maps from the encoder. These connections are effective in recapturing intricate details of target objects, resulting in segmentation masks that retain fine details even against challenging backgrounds. This technique is also crucial for the performance of instance-level segmentation models like Mask-RCNN [4], which allows for the segmentation of objects that are occluded.

While image segmentation has achieved commendable performance in natural images, the question remains whether these models can meet the stringent conditions of medical image segmentation. Segmenting abnormalities or lesions or in medical images demands a higher degree of precision compared to natural images. Even minor flaws in segmentation of medical images can have a major effect on clinical outcomes. For example, accurately including the subtle spiculation patterns surrounding a nodule, which can indicate malignancy, is crucial for maintaining the clinical credibility of the model [1].

Moreover, these models face several performance issues when applied to medical image segmentation. One significant challenge is the high computational complexity especially in tasks with 3-D images such as MRIs [5], leading to increased processing times and substantial demands on hardware resources, making real-time analysis difficult. Memory usage is also a concern, as high-resolution medical images and complex model architectures require significant memory, often resulting in out-of-memory errors. Training these models to achieve high accuracy is another issue, as it demands extensive

datasets and prolonged training periods, compounded by the scarcity and cost of finely annotated medical images.

Additionally, generalization remains a problem, as models trained on specific datasets may not perform well on different types of medical images due to variations in imaging techniques, patient populations, and disease presentations [6]. The interpretability of these complex models is also a critical issue, as understanding the model's decisions is essential for clinical validation and trust. Finally, even minor inaccuracies in segmentation can have significant clinical consequences. For instance, missing subtle features like spiculation patterns around nodules can impact diagnosis and treatment planning [7]. Thus, ensuring high segmentation accuracy is crucial for the clinical applicability of these models.

One notable architecture is the BS U-NET, which combines the strengths of U-NET with bottleneck feature supervision of label map to achieve superior segmentation results. It was aimed to increase the accuracy by using low level features of labels as a supportive loss value to the loss from original U-NET. Despite state of art design of BS U-NET, there are serious drawbacks about the size and accuracy of the network [8].

To address these issues, Tree-NET, a novel state-of-the-art architecture is presented for segmentation issues based on bottleneck feature supervision of both input and segmentation labels and then creating a bridge network between them to train their low-level representations. The network consists of three main components named Encoder-Net, Decoder-Net and Bridge-Net as each having separate training processes. After the trainings, the whole network is combined, and it has one final training end-to-end which is named self-tuning.

By this technique, the bridge network of choice is trained with smaller sized input and label images without losing any essential features improving the model performance, accuracy and decreasing the training size while keeping the trainable parameters almost the same number.

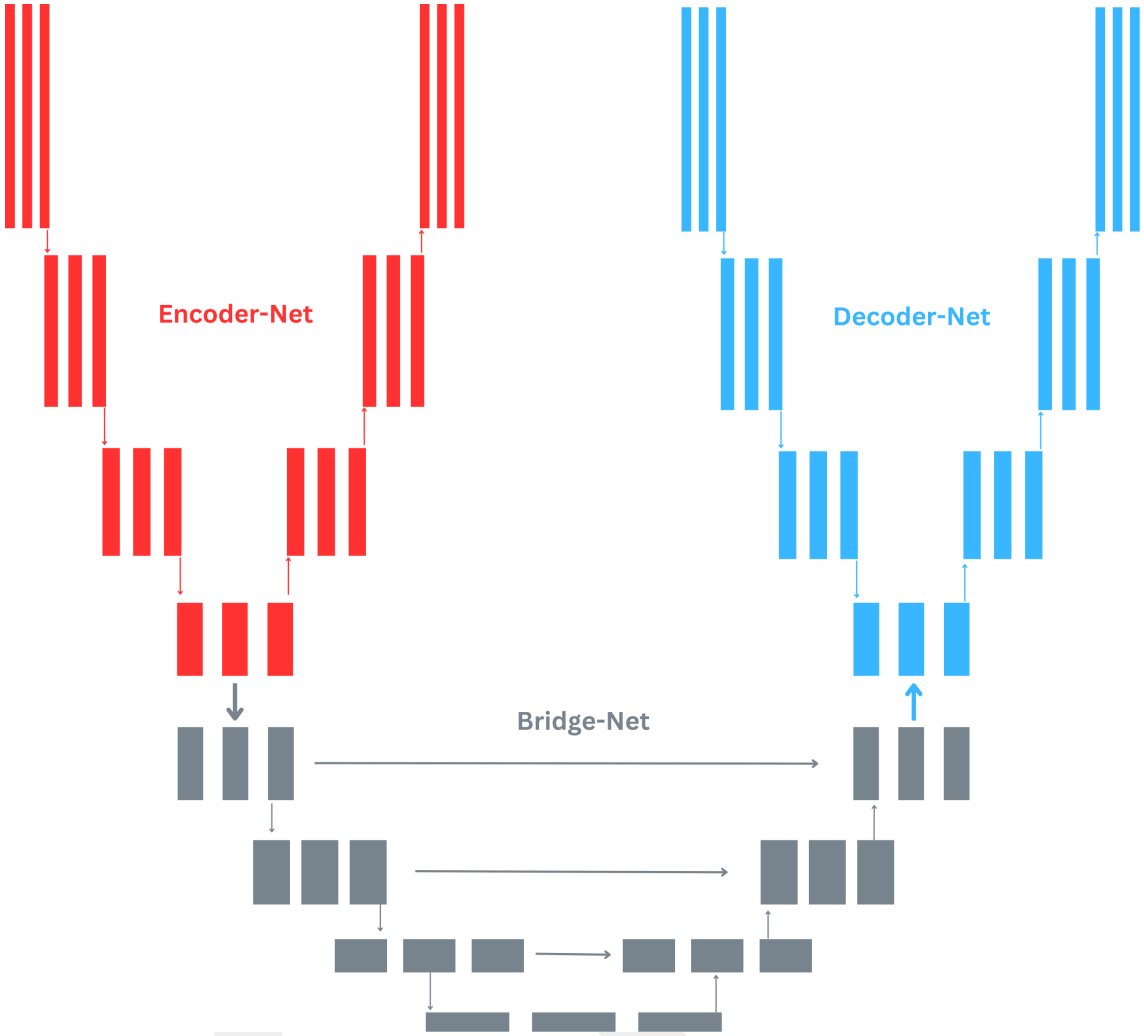


Figure 1.1 Tree-NET diagram. Red blocks denote the Encoder-Net for input encoding. Gray blocks represent the U-NET architecture as the Bridge-Net for segmentation training. Blue blocks indicate the Decoder-Net for output decoding to match the original label size.

1.1 Literature Review

U-NET shaped networks have emerged as a potent tool in the field of medical imaging analysis, highlighting their effectiveness in medical image segmentation [9-12]. Their distinctive architecture enables efficient handling of limited labeled data, employing techniques such as data augmentation. The incorporation of skip connections further enhances segmentation accuracy, making U-NET an influential framework in medical image analysis. Bottleneck Supervised U-NET (BS U-NET), a recent advancement, combines U-NET's architecture with bottleneck feature supervision, significantly enhancing segmentation performance by utilizing high-level feature representations.

BS U-NET integrates the concept of bottleneck feature supervision into the U-NET architecture. Bottleneck features, extracted from the intermediate layers of an autoencoder trained by the label data, encapsulate rich, high-level information that can improve segmentation accuracy and generalizability. BS U-NET leverages these features to address challenges such as data scarcity and variability in medical images [8].

1.2 Motivation and Contribution

Segmentation of medical images is generally more of a challenge compared with natural images segmentation. This difficulty arises from several factors: first, target objects in medical images often exhibit irregularities in size, shape, and intensity, complicating the model's ability to accurately learn patterns; second, medical image segmentation is prone to a high rate of false positives because target tissues can share similar features (intensity, size, and shape) with other tissues; third, the annotation of medical images is both costly and time-consuming, leading to a highly imbalanced training dataset with a predominance of positive or negative cases, which further exacerbates the false positive issue.

While numerous segmentation models have been proposed in the literature, few address the bottleneck feature vector, which encapsulates highly condensed information from the input images.

In this thesis, we introduce a novel network with a unique autoencoder-supported architecture that performs a significant segmentation result while lowering the computational cost. Our key contributions are as follows:

- We propose Tree-NET, a bottleneck feature supervised deep learning architecture that completes the segmentation training between the low-level representation of input and labels.
- It has an end-to-end assembled unique structure where its three main components Encoder-Net, Bridge-Net and Decoder-Net have isolated training.
- Input and Label data are being supervised by the Encoder-Net and Decoder-Net respectively before segmentation training by the Bridge-Net as it enables to learn features without overfitting.
- Bridge-Net is a conceptual segmentation model that to various architectures could be selected as the Bridge-Net.
- The Tree-NET algorithm reduces the computational cost in terms of memory usage and speed performance by shrinking the input and label size without losing information while enhancing the accuracy of the segmentation output.
- The additional step to the bottleneck feature supervision is the self-tuning approach which supervises the whole network built after the isolated supervision of bottleneck features.

Chapter 2

Background

2.1 Colorectal Cancer

Colorectal cancer (CRC) is a type of cancer that originates in the colon or rectum, both of which are parts of the digestive system. It ranks among the most prevalent cancers worldwide and is a leading cause of cancer-related mortality. CRC commonly begins as benign growths called polyps on the inner lining of the colon or rectum. These polyps can become malignant over time, developing into colorectal cancer [13].

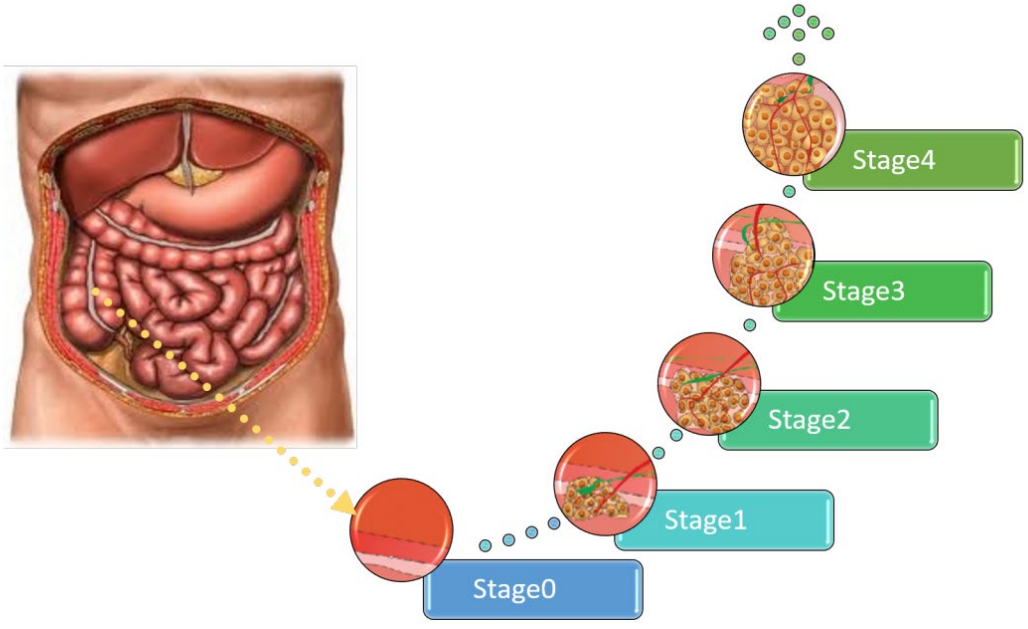


Figure 2.1 The different stages of colorectal cancer

The incidence of colorectal cancer varies across the globe, with higher rates in developed nations. This disparity is attributed to differences in diet, lifestyle, and healthcare access. Risk factors for CRC include being over the age of 50, having a family colorectal cancer or polyps' history, certain genetic disorders like familial adenomatous

polyposis and Lynch syndrome, diets high in red and processed meats, lack of physical activity, over consumption of alcohol, smoking and obesity.

In its early stages, colorectal cancer often does not cause any symptoms, highlighting the importance of regular screening for early detection. When symptoms do appear, they may include alterations in bowel habits (such as diarrhea or constipation), bleeding from the rectum or blood in the stool, persistent discomfort in the abdomen, fatigue, and unexplained weight loss. Screening options include fecal occult blood tests (FOBT), sigmoidoscopy, colonoscopy, and imaging techniques like CT colonography. Colonoscopy is considered the gold standard for diagnosing CRC as it allows direct visualization and biopsy of suspicious areas [14].

The treatment approach for colorectal cancer depends on the stage at which it is diagnosed. Early-stage CRC may be addressed with surgery alone, which involves removing the tumor and surrounding tissue. For more advanced stages, a combination of surgery, chemotherapy, and radiation therapy is often required. Targeted therapies and immunotherapies have also shown effectiveness for certain types of colorectal cancer. Prognosis varies with early detection greatly improving outcomes, resulting in a five-year survival rate of over 90% for localized cancers. However, survival rates drop significantly once the cancer has metastasized [15].

Preventing colorectal cancer involves lifestyle changes such as adhering to a nutritious diet with plenty of fruits, vegetables, and whole grains, participating in regular physical activity, refraining from smoking, and moderating alcohol consumption. Regular screening is critical, particularly for those at higher risk due to age, family history, or genetic factors. Generally, guidelines recommend that average-risk individuals begin screening between the ages of 45 and 50, with the frequency of screening depending on the method used and initial findings.

Colorectal cancer poses a major public health challenge due to its high prevalence and the severe consequences if not detected early. Enhancing public awareness, preventive strategies, and advancements in screening and treatment are essential to reduce the incidence and improve survival rates of this disease.

2.2 Skin Cancer

The integumentary system, which encompasses the skin and its appendages (hair, nails, sweat, and oil glands), is the largest organ of the human body, with an average surface area of 2.0 square meters. The skin primarily functions to protect the body from various external threats, including microbes, chemicals, and temperature extremes. It also secretes antibacterial substances, and melanin in the skin acts as a chemical protection layer against UV, which can damage skin cells [16].

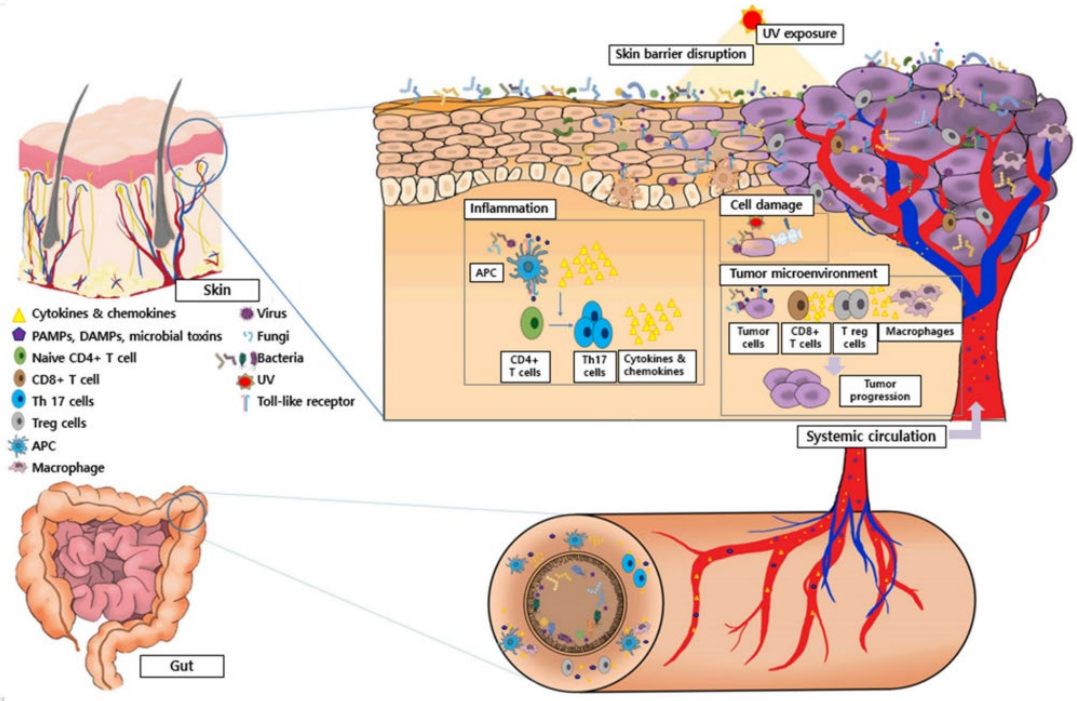


Figure 2.2 The Physiological Process of Skin Cancer Development [17]

The skin comprises multiple layers, but the two main ones are the epidermis (the outermost layer) and the dermis (the layer beneath the epidermis), as illustrated in Fig. 2.1. Skin cancer arises from uncontrolled cell growth due to DNA mutations. It ranks as the 17th most common cancer worldwide, with an estimated 1.8 million cases expected in 2020 [18]. Early detection of melanoma yields a five-year survival rate of over 99 percent, which drops to 71 percent if the cancer reaches the lymph nodes, and to 32 percent if it spreads to distant organs [19]. Most skin cancer cases are found on sun-exposed areas such as face, arms, hands, neck, chest, ears, and scalp [17].

The most common types of skin cancer include melanoma, squamous cell carcinoma, and basal cell carcinoma. Basal and squamous cell carcinomas are primarily

found in the basal and squamous layers of the skin, respectively. Both types are generally treatable but can be disfiguring and costly to manage. In contrast, melanomas, which start in melanocytes, are the most dangerous due to their tendency to metastasize to other parts of the body, particularly vital organs [17]. The primary cause of skin cancer is excessive exposure to ultraviolet (UV) rays from the sun, tanning beds, or sunlamps. UV radiation can damage skin cells, leading to sunburn and, over time, accumulate damage that results in changes in skin texture, premature aging, and occasionally skin cancer. Unlike internal cancers, skin cancers develop externally and are often visible, allowing for early detection. Early diagnosis of skin cancer provides the best opportunity for effective treatment [20].

2.3 Medical Image Segmentation

Image segmentation tasks are divided into two main categories: semantic and instance segmentation [21, 22]. Semantic segmentation involves pixel-level classification, assigning each pixel in an image to a specific category. In contrast, instance segmentation not only classifies each pixel but also differentiates distinct objects within the same category. To effectively distinguish lesion or organ pixels, segmentation methods must be designed such that it ensures to capture the essential details. Common sensors for acquiring medical imaging data include Magnetic Resonance Imaging (MRI), X-ray, Computed Tomography (CT), Positron Emission Tomography (PET and Ultrasound (US) [23].

Earlier traditional methods for medical image segmentation relied on techniques such as template matching, edge detection, graph cuts, region growing, active contour lines, and various mathematical approaches. However, in recent years, deep learning has advanced significantly across numerous fields, addressing many complex cases specific to the medical area. Convolutional neural networks (CNNs) have been particularly successful in extraction of feature representation for images, removing the necessity for manually crafted features in image segmentation, and achieving outstanding performance.

One successful model is introduced Fully Convolutional Networks (FCNs) that employ multiple convolutional blocks in the encoder path to obtain semantic

representations [3]. Similarly, in the decoding path, it uses convolutional layers jointly with up-sampling operations to achieve predictions in pixel-level. The primary motivation behind the successive up-sampling process in the decoding path is to gradually enhance the spatial dimension for detailed segmentation results.

Inspired by the structure of FCNs and encoder-decoder models, the U-NET model is developed for biomedical image segmentation. This model is specifically designed for practical application in medical image analysis and can be utilized across various imaging modalities [24-28].

2.3.1 U-NET

U-NET is a CNN based architecture with U shape structure that is used for image segmentation tasks. Fig. 1 represents the original architecture of U-NET [2]. A key feature of U-NET is its structure: the encoding path (left side) compresses the input into low level features, which is then decompressed into the output map by the decoder part (right side). As illustrated in Fig. 2.3, the feature map on the left side (in the middle) is concatenated with the feature map on the right side as followed by the arrows. The skip connections help mitigate information loss during encoding, making U-NET particularly suitable for medical images considering that some target tissues are quite small. This design allows U-NET to retain critical spatial information and ensure that fine details are preserved throughout the network. Such information might be lost in the deeper layers of the encoding path. Due to the features outlined, U-NET has been extensively applied to various medical tasks, including tumor detection, organ segmentation, and cellular structure identification [29-31].

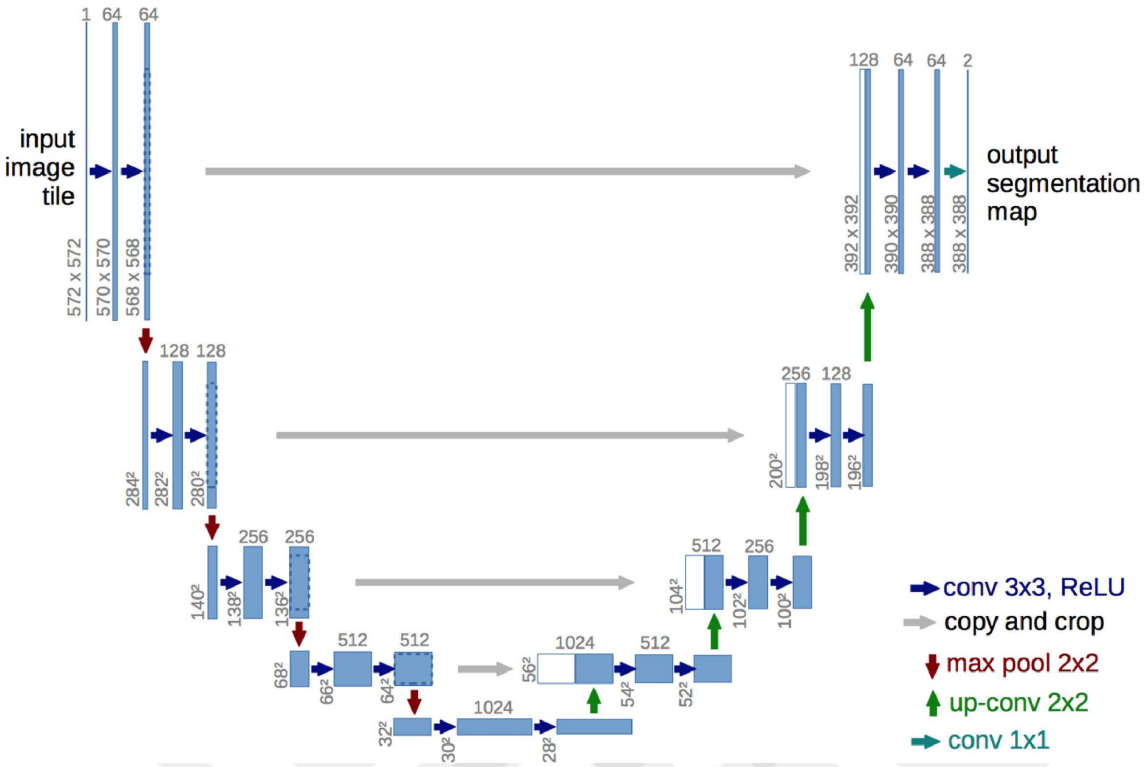


Figure 2.3 Diagram of U-NET

2.3.2 U-NET++

U-NET++ enhances the original U-NET architecture with nested and dense skip pathways. These pathways connect encoding and decoding nodes at different scales, facilitating better feature propagation and fusion. Specifically, U-NET++ introduces the following key enhancements:

Dense Skip Connections: U-NET++ employs dense skip connections that pass information through multiple layers, enhancing the network's ability to capture and integrate multi-scale features.

Nested Architectures: The nested architecture consists of multiple sub-networks that are interconnected, enabling more refined and detailed feature extraction and fusion.

Improved Feature Fusion: By leveraging dense and nested connections, U-NET++ ensures more comprehensive feature fusion, leading to improved segmentation accuracy.

Implementing U-NET++ involves constructing the nested encoder-decoder architecture with dense connections. The training process typically requires significant computational resources due to the increase in complexity of the model. Hyperparameters such as learning rate, batch size, and the choice of optimizer play crucial roles in the model's performance. [1]

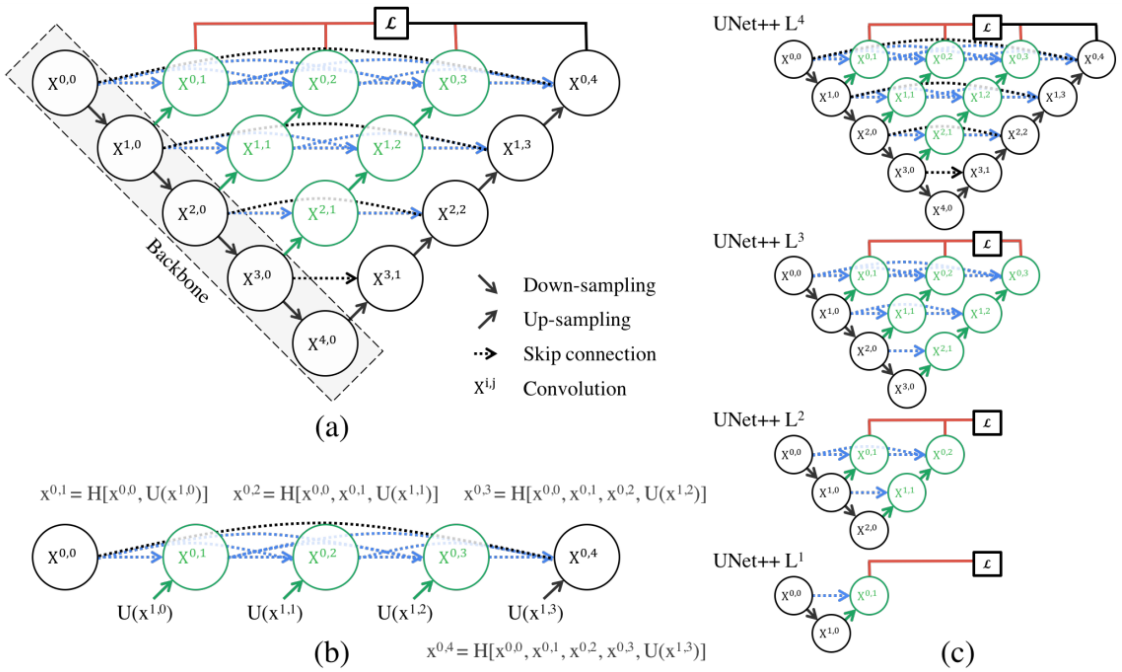


Figure 2.4 Diagram of U-NET++

2.4 Autoencoder

Autoencoders are unsupervised neural networks that specialize in compressing the given data as efficiently as possible by utilizing an hourglass shaped network. General uses of Autoencoders are dimensional reduction, noise cancellation and colorization by encoding high-dimensional data into bottleneck features. An autoencoder typically comprises two main components: an encoder and a decoder [32]. Essentially, an autoencoder can be seen as a non-recurrent feed-forward neural network having identical input and output. Notably, removing the skip connections from the U-NET would create a very similar architecture to that of an autoencoder.

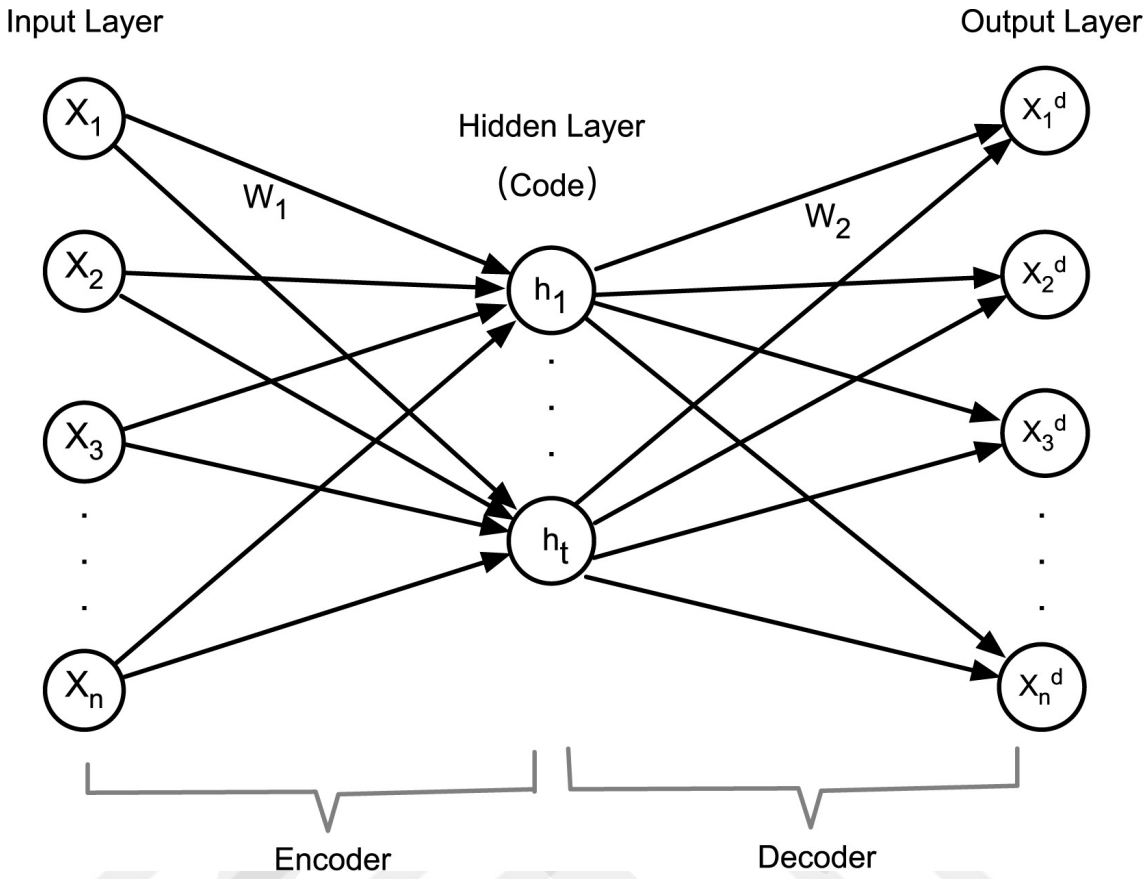


Figure 2.5 Structural diagram of an autoencoder: x represents the input data of the input layer, h denotes the hidden layer's data, and x_1 indicates the reconstructed output data in the output layer [33].

2.5 Bottleneck Feature Supervision

Bottleneck feature supervision is a method used in segmentation models like U-NET or FCN to enhance their performance in capturing detailed image information across different scales. In these models, a "bottleneck" layer represents a central point where the spatial resolution of feature maps is reduced while retaining important semantic details.

With bottleneck feature supervision, additional supervision is introduced at this bottleneck layer by incorporating extra branches or layers that predict specific semantic features. This approach helps the model learn to capture both global and local context

effectively, acts as a form of regularization to prevent overfitting, and ultimately improves segmentation accuracy by guiding the learning of more robust representations.

In BS U-NET, practically there are two main components in the algorithm such as Encoding U-Net and Segmentation U-NET as in Fig. 2.6.

Encoding U-NET: It is basically an Autoencoder that the label masks of the dataset are being fed into the network as input. The main idea here is to feed the bottleneck features of this model into the Segmentation U-NET as an additional loss. To do that, it has to be trained before the Segmentation U-NET. Dice Loss is used as the loss function for this model.

Segmentation U-NET: It is the original U-NET model with an additional bottleneck supervision supplied by the bottleneck features of Encoder-Net as an additional loss. Thus, the loss function is the weighted summation of two losses as one is the original loss and the other comes from the Encoder-Net Bottleneck features.

$$Totall\ loss = w1 * DiceLoss + w2 * EucledianLoss \quad (\mathbf{2.1})$$

where the summation of **w1** and **w2** is equal to 1.

The additional component Encoding U-NET is trained alongside the main segmentation task using intermediate feature maps to predict semantic attributes, aimed model to achieve better segmentation performance by leveraging multi-scale contextual information [8]. That means it isn't realy an assembled architecture. Encoding U-NETwill be out of the equation after it supports the Segmentation-Net with its Bottleneck Features. After the trainings of these two components, only the Segmentation U-NET will be used as the BS U-NET.

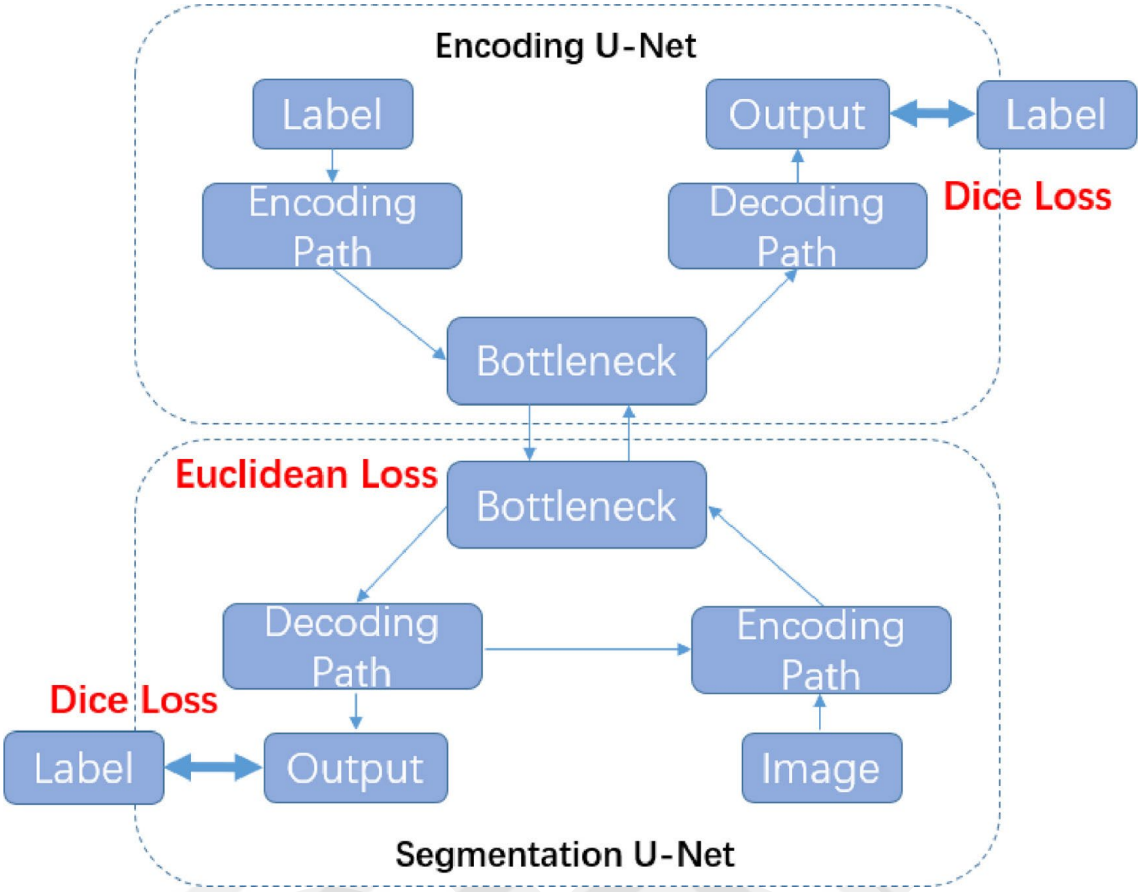


Figure 2.6 The bottleneck feature supervised (BS) U-NET

Chapter 3

Model Architecture

The Tree-NET model consists of three main components such as Encoder-Net, Bridge-Net, and Decoder-Net.

3.1 Encoder-Net

Encoder-Net is a convolutional autoencoder model consisting of encoder and decoder parts. It is designed for extracting the low-level representation (bottleneck feature) of input images. The Encoder-Net training will be separately applied on input images and bottleneck features obtained to be fed into the Bridge-Net in order to do the segmentation training.

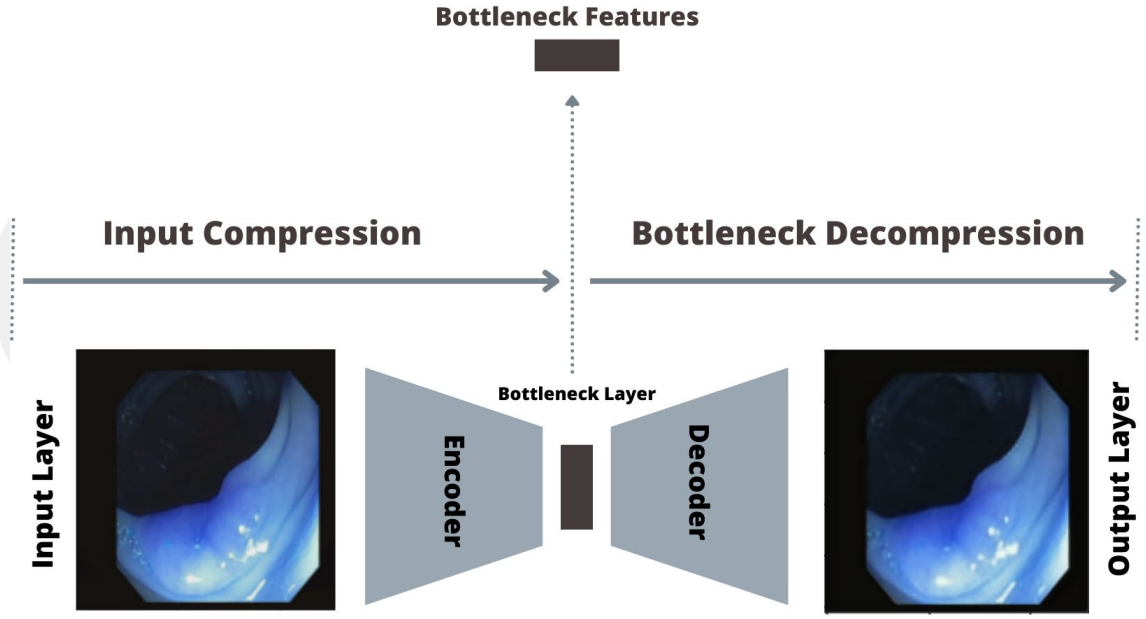


Figure 3.1 Encoder-Net

3.1.1 Network Structure

Table 3.1 Encoder-Net Structure

Layer Type	Input Channel	Output Channel	Kernel Size	Stride	Padding	Parameters
Conv2d	3	8	3x3	1	0	224
ELU						
MaxPool2d			2x2	2		
Conv2d	8	24	3x3	1	0	1,752
ELU						
Conv2d	24	32	3x3	2	0	19,232
ReLU						
Conv2d	32	18	3x3	2	0	5,202
ELU						
Conv2d	18	3	3x3	1	0	489
ReLU						
ConvTranspose 2d	3	10	3x3	1	1	280
ELU						
ConvTranspose 2d	10	24	3x3	1	1	2,184
ELU						
ConvTranspose 2d	24	32	3x3	2	1	16,192
ELU						

Up Sample			2x2	2		
ConvTranspose 2d ELU	32	24	3x3	2	0	6,936
ConvTranspose 2d ReLU	24	8	3x3	2	0	1,736
ConvTranspose 2d ReLU	8	3	3x3	2	0	219
Total						**54,446**

3.2 Bridge-Net

As the name implies, Bridge-Net is the network where the Encoder-Net and Decoder-Net are integrated. In this model, the bottleneck features from the Encoder-Net are fed into the Bridge-Net while the bottleneck features of Decoder-Net are used as the labels. Fig. 3.2 explains the process.

There is no specific design for this network so any type of segmentation model could be selected and used as the Bridge-Net. In our experiment, U-NET and U-NET++ are selected as the Bridge-Net for segmentation.

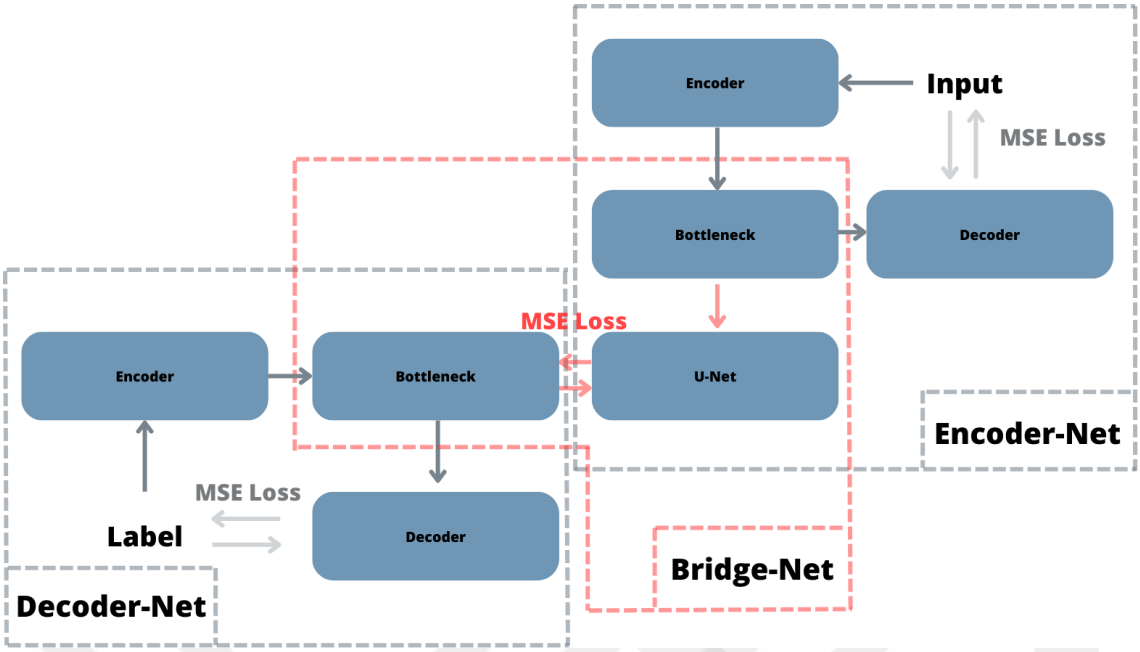


Figure 3.2 Tree-NET Components: The diagram illustrates the Encoder-Net, Bridge-Net, and Decoder-Net. Unidirectional arrows indicate the data flow through the networks, while bidirectional arrows represent the process of loss computation.

3.2.1 U-NET Structure

U-NET used in this paper consists of 16 convolution layers having a total number of 7,852,611 trainable parameters. Visual Geometry Group (VGG) Blocks are created that includes 2 convolution layers inside to decrease the design complexity.

Table 3.2 VGG Block.

Layer Type	Input Channel	Output Channel	Kernel Size	Stride	Padding	Parameters
Conv2d	n	h	3x3		1	nxhx3x3+h
BatchNorm2d						h
ReLU			3x3			
Conv2d	h	o	3x3		1	hxox3x3+o
BatchNorm2d						o

ReLU							
Total							**hx3x3x(n+o)+2h+o**

After each VGG Block, up-samplings or down-samplings are applied regarding to the layers as in Table 3.3. The hidden channel represents the hidden size of each VGG block.

Table 3.3 Structure of U-NET.

Layer Type	**Input Channel**	**Hidden Channel**	**Output Channel**	**Kernel Size**	**Stride**	**Padding**	**Parameters**
			Encoder				
VGG Block	3	32	32	3x3	1	1	10,272
MaxPool2d				2x2	2		
VGG Block	32	64	64	3x3	1	1	55,680
MaxPool2d				2x2	2		
VGG Block	64	128	128	3x3	1	1	221,952
MaxPool2d				2x2	2		
VGG Block	128	256	256	3x3	1	1	886,272
MaxPool2d				2x2	2		
VGG Block	256	512	512		1	1	3,542,016
			Decoder				

Upsample				2x2	2		
VGG Block	(256+512)	256	256	3x3	1	1	2,360,832
Upsample				2x2	2		
VGG Block	256	512	512	3x3	1	1	590,512
Upsample				2x2	2		
VGG Block	(128+64)	64	64	2x2	1	1	147,840
Upsample				2x2	2		
VGG Block	(64+32)	32	32	3x3	1	1	37,056
Conv2d	32	3	1	1x1	1	0	99
Total							**7,852,611**

3.2.2 U-NET++ Structure

The same VGG Blocks are used to build U-NET++ architecture. U-NET++ variant has more complex structure compared to the original U-NET. Thus, layer numbers corresponding to the numb are added in the table 3.4

Table 3.4 Structure of U-NET++

Layer Type	Input Channel	Hidden Channel	Output Channel	Kernel Size	Stride	Padding	Parameters
			Encoder				
			LAYER 1				
VGG Block	3	32	32	3x3	1	1	10,272

MaxPool2d				2x2	2		
VGG Block	32	64	64	3x3	1	1	55,680
			LAYER 2				
MaxPool2d				2x2	2		
VGG Block	64	128	128	3x3	1	1	221,952
			LAYER 3				
MaxPool2d				2x2	2		
VGG Block	128	256	256	3x3	1	1	886,272
			LAYER 4				
MaxPool2d				2x2	2		
VGG Block	256	512	512		1	1	3,542,016
			Decoder				
			LAYER 1				
Upsample				2x2	2		
VGG Block	(64+32)	32	32	3x3	1	1	37,056
			LAYER 2				
Upsample				2x2	2		
VGG Block	(128+64)	64	64	3x3	1	1	147,840
Upsample				2x2	2		
VGG Block	(32x2+64)	32	32	2x2	1	1	46,176

			LAYER 3				
Upsample				2x2	2		
VGG Block	(256+128)	128	128	3x3	1	1	590,592
Upsample				2x2	2		
VGG Block	(64x2+128)	128	128	3x3	1	1	442,752
Upsample				2x2	2		
VGG Block	(32x3+64)	32	32	3x3	1	1	55,488
			LAYER 4				
Upsample				2x2	2		
VGG Block	(256+512)	256	256	3x3	1	1	2,360,832
Upsample				2x2	2		
VGG Block	(128x2+256)	128	128	3x3	1	1	738,048
Upsample				2x2	2		
VGG Block	(64x3+128)	64	64	3x3	1	1	221,568
Upsample				2x2	2		
VGG Block	(32x4+64)	32	32	3x3	1	1	64,704
Conv2d	32	3	1	1x1	1	0	99
Total							**9,163,395**

3.3 Decoder-Net

Decoder-Net extracts the low-level representation (bottleneck feature) of Labels. After the model is trained using Labels as the input, Output Compression part of Decoder-Net is for obtaining the bottleneck features are used as labels for training the Bridge-Net. When the training is completed, as the name implies, Bottleneck Decompression part of

the Decoder-Net will be used to decode output into the regular size to get the desired segmentation output.

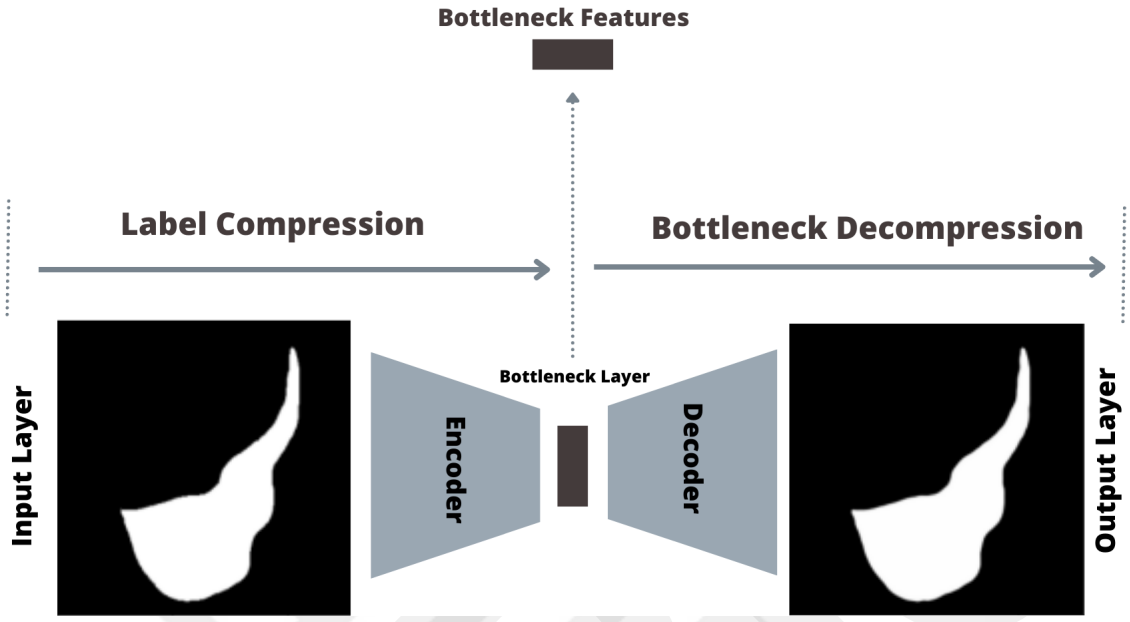


Figure 3.3 Decoder-Net

3.3.1 Network Structure

Table 3.4 Structure of Decoder-Net

Layer Type	Input Channel	Output Channel	Kernel Size	Stride	Padding	Parameters
Conv2d	1	8	3x3	1	0	80
ELU						
MaxPool2d			2x2	2		
Conv2d	8	24	3x3	1	0	1,752
ELU						
Conv2d	24	32	3x3	2	0	19,232
ELU						

Conv2d	32	5	3x3	2	0	5,202
ELU						
ConvTranspose2d	5	3	3x3	1	0	489
Sigmoid						
Conv2d	3	24	3x3	1	0	504
ELU						
Upsample			2x2	2		
ConvTranspose2d	24	32	3x3	1	0	14,464
ELU						
Upsample			2x2	2		
ConvTranspose2d	32	24	3x3	2	0	6,936
ELU						
ConvTranspose2d	24	16	3x3	2	0	1,736
ELU						
Upsample			2x2	2		
ConvTranspose2d	16	8	3x3	2	0	73
ELU						
Total						**50,468**

3.3 Overall Architecture

Figure 3.4 represents the overall architecture of the proposed model. Combining the three components such as Encoder-Net, Bridge-Net and Decoder-Net, the model could be processed through a further training process named as Self-tuning (ST). As all the three components are trained separately, this training is aimed to provide a better harmonization between them and remove unnecessary learned connections.

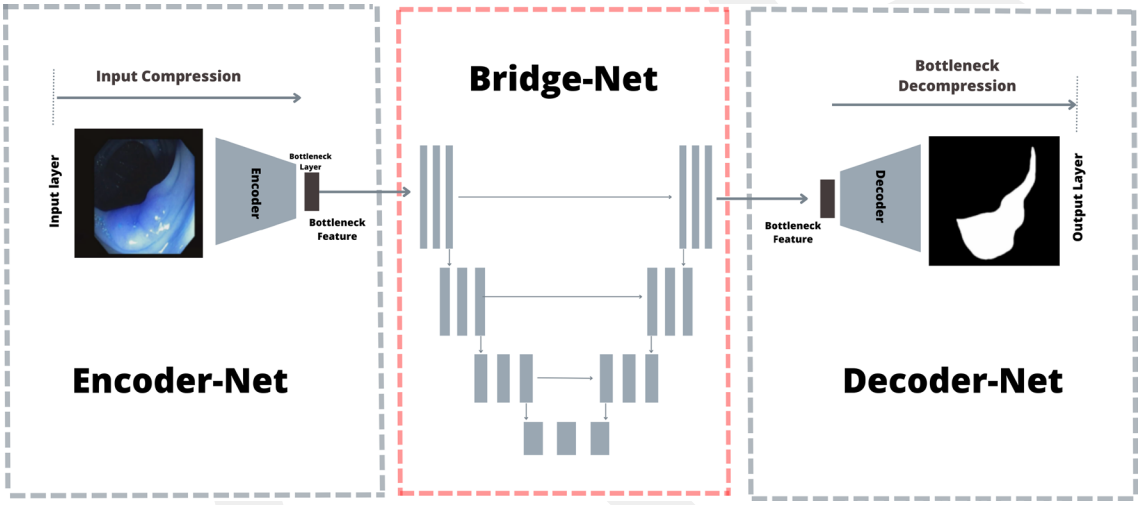


Figure 3.4 Tree-NET diagram. The Encoder part of the Encoder-Net, Bridge-Net, and Decoder part of the Decoder-Net are assembled end-to-end, forming the overall structure for self-tuning.

Chapter 4

Results

4.1 Performance Evaluation

4.1.1 Dataset Descriptions

There are two main datasets used to evaluate the proposed algorithm's performance such as colon and skin cancer segmentation.

4.1.1.1 CVC-ClinicDB Dataset

The CVC-ClinicDB dataset, developed by the Computer Vision Center (CVC), is a comprehensive resource designed to advance the field of colonoscopy image analysis through the development and benchmarking of automated diagnostic algorithms. This dataset supports several primary tasks: polyp detection, polyp segmentation, and colonoscopy image classification.

- Dataset Source: The dataset used in this study is CVC-ClinicDB, which is a collection of frames extracted from colonoscopy videos specifically curated for research in polyp detection and segmentation. CVC-ClinicDB is the official database designated for use in the training stages of the MICCAI 2015 Sub-Challenge on Automatic Polyp Detection Challenge in Colonoscopy Videos.
- Image Resolution: 384×288x3
- Dataset Size: 612
- Annotation Details: The dataset includes ground truth annotations for polyps, which consist of masks corresponding to the regions covered by polyps in the images. Each original image in the dataset is paired with a

corresponding polyp mask, where the mask indicates the precise area of the image occupied by the polyp.

For each image:

- Original Images: The original images are stored in the format original/frame_number.tiff.
- Polyp Masks: The ground truth polyp masks are provided in the format ground truth/frame_number.tiff, where frame number corresponds to the number of the frame extracted from the colonoscopy video [34].

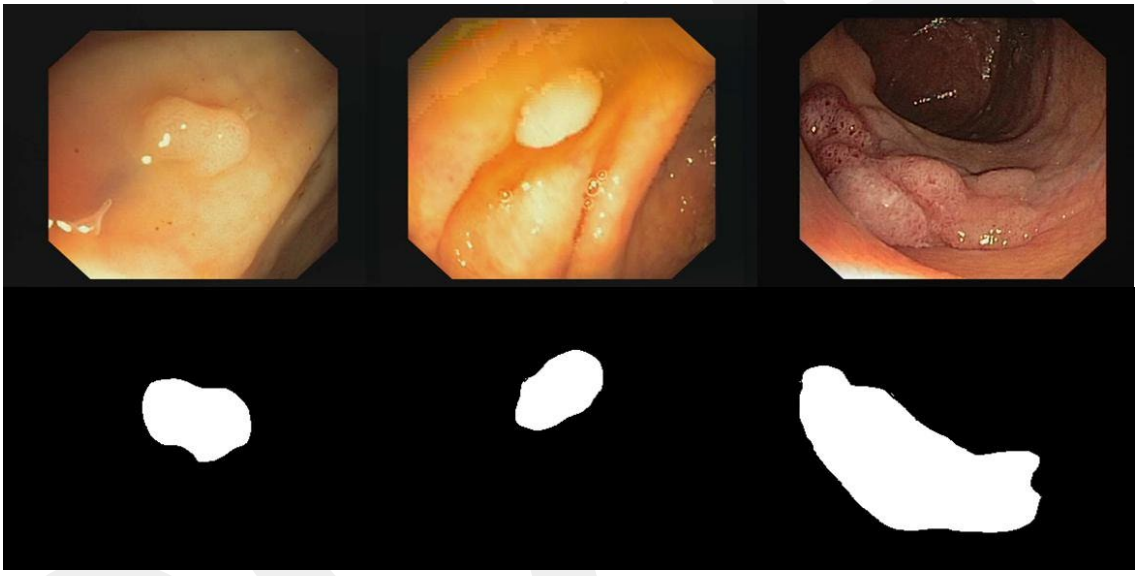


Figure 4.1 Sample Polyp Images

4.1.1.2 ISIC-2018 Dataset

The ISIC 2018 dataset, developed by the International Skin Imaging Collaboration (ISIC), is a comprehensive resource designed to advance the field of skin lesion analysis through the development and benchmarking of automated diagnostic algorithms. This dataset supports three primary tasks: lesion segmentation, lesion attribute detection, and disease classification.

- **Dataset Source**: The ISIC 2018 dataset is provided by the International Skin Imaging Collaboration (ISIC), a global partnership aimed at improving melanoma diagnosis through the development of advanced imaging and analysis techniques.

• **Dataset Size**: The dataset contains 2594 training, 100 validation and 1000 test high-resolution dermoscopic images.

• **Image Resolution**: The resolution of the images in the ISIC 2018 dataset varies, but they are typically high-resolution, allowing for detailed analysis of skin lesions. Common resolutions include images with dimensions around 1022x767 pixels, although this can vary slightly depending on the specific image.

• **Annotation Details**:

  ○ Original Images: Each image in the dataset is an original dermoscopic image of a skin lesion with .jpg format.

  ○ Polyp Masks: For each original image, there are corresponding ground truth masks that delineate the boundaries of the lesions. These masks are essential for training and evaluating segmentation algorithms [35].

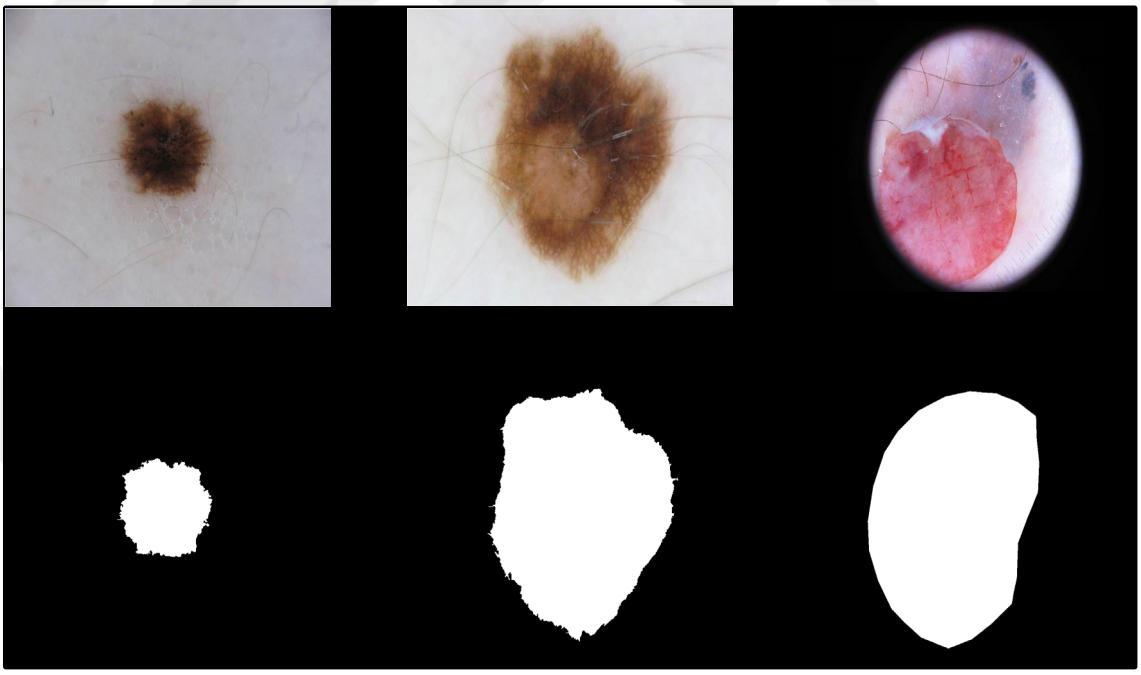


Figure 4.2 Skin Lesion Samples

4.1.2 Experimental Setup

4.1.2.1 Preprocessing

The four main parts of the preprocessing in order are as follows:

- **Data resizing:** Datasets are resized into 256x256x3.
- **Data shuffling:** For Clinic-DB, with respect to the length of dataset input 612, a randomized index is created, and index values are saved to shuffle the data for training all three components such as Encoder-Net, Bridge-Net and Decoder-Net (the identical index is applied for the other algorithms) to keep the same experiment.
- **Data division:** For Clinic-DB, the dataset is divided into three parts such as training, validation and test with the ratios of 0.8,0.1,0.1 respectively. For the ISIC-2018, the data is splitted into training, validation and test parts by default.
- **Normalization:** Inputs and Labels in dataset are normalized into 0 and 1.

4.1.2.2 Training Procedure

To complete the training of Tree-NET, Encoder-Net and Decoder-Net's is be trained initially. Then the input is fed into the Encoder-Net, and bottleneck features are collected to create the new input with size 3x64x64. The same process is also applied to label data using Decoder-Net and label with size 3x64x64 is obtained.

The selection model of Bridge-Net as U-NET and U-NET++ are fed with the created inputs and supervised by the created labels. When these three training processes are completed, encoder part of the Encoder-Net, the Bridge-Net and decoder part of the Decoder-Nets are assembled tip to tip with respectively.

After the assembly, self-tuning is applied to improve the models' performance in terms of accuracy. As all the three components are trained separately, the additional training aims to harmonize them and remove the unnecessary trained features and focus more on the ones that could improve the results.

4.1.2.3 Parameters

For the segmentation networks such as Bridge-Net, U-NET, U-NET++ and BS U-NET, most of the hyperparameters are kept the same such as the optimizer type, batch

size and the learning rate in order to keep the experimental setup the same. For the Encoder-Net and Decoder, due to the smaller number of hyperparameters and different structures, the optimum values vary from the segmentation networks. Thus, they are selected separately as shown in table 4.1.

Table 4.1 Training Parameters. Tree-NET denotes the integrated architecture comprising Encoder-Net, Decoder-Net, and Bridge-Net. "ST" refers to the process of Self Tuning.

	Encoder-Net	Bridge-Net	Decoder-Net	Tree-NET ST	U-NET	U-NET++
Input Size	3x256x256	3x64x64	1x256x256	3x256x256	3x256x256	3x256x256
Bottleneck Size	3x64x64	512x4x4	3x64x64	512x4x4	512x16x16	512x16x16
Output Size	3x256x256	3x64x64	1x256x256	1x256x256	1x256x256	1x256x256
Batch Size	8	8	8	8	8	8
Learning Rate	0.001	0.0001	0.001	0.0001	0.0001	0.0001
Loss Function	MSE	MSE	MSE	Dice + BCE	Dice + BCE	Dice + BCE
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Epoch Number	90	30	90	10	30	30

For U-NET and U-NET++ combinations of Dice and BCE are used. Dice Loss mitigates the issue of class imbalance by focusing on the overlap, while BCE ensures pixel-wise accuracy.

$$Totall\ loss = w1 * DiceLoss + w2 * BCELoss \qquad (\mathbf{4.1})$$

where the summation of $\mathbf{w1}$ and $\mathbf{w2}$ is equal to 1.

For the Bridge-Net, MSE loss is preferred compared to the Dice Loss and BCE Loss considering the features are low level representations where there are no exact 0 and 1 segmentation output features.

After the Tree-NET assembly, combinations of Dice and BCE losses are used since it becomes end-to-end network and more suitable.

4.1.2.3 Evaluation Metrics

The proposed model has been evaluated and compared with the other models in terms of accuracy, overfitting and computational usages.

For the accuracy evaluation, we utilize the Dice coefficient, Intersection Over Union (IoU), and accuracy (ACC) scores on the test dataset. These metrics provide a comprehensive assessment of the segmentation quality, capturing various aspects of the model's performance.

The Dice coefficient, also known as the Dice Similarity Coefficient (DSC), measures the overlap between the predicted segmentation and the ground truth. It is particularly useful in medical image segmentation due to its sensitivity to both false positives and false negatives. The Dice coefficient is defined as:

$$Dice\ Coefficient\ = \frac{2 \times tp}{2 \times tp + fp + fn} \qquad (\mathbf{4.2})$$

where;

True Positives (TP): The number of correctly predicted pixels as part of the target class (both the prediction and ground truth are 1).

True Negatives (TN): The number of correctly predicted pixels as part of the background class (both the prediction and ground truth are 0).

False Positives (FP): The number of incorrectly predicted pixels as part of the target class (the prediction is 1, but the ground truth is 0).

False Negatives (FN): The number of incorrectly predicted pixels as part of the background class (the prediction is 0, but the ground truth is 1).

The Intersection-Over-Union (IoU quantifies the intersection ratio to the union of the predicted and ground truth masks. It yields a clear indication of the accuracy of the

segmentation by measuring the fraction of correctly predicted pixels out of the total pixels in either the prediction or the ground truth. The IoU is calculated as:

$$IoU = \frac{tp}{tp + fp + fn} \quad (\mathbf{4.3})$$

Accuracy measures the proportion of correctly classified pixels (both true positives and true negatives) out of the total number of pixels. While accuracy is a straightforward metric, it may not be as informative in cases of class imbalance, which is common in medical image segmentation. Accuracy is given by:

$$Accuracy\ Score = \frac{tp + tn}{tp + tn + fp + fn} \quad (\mathbf{4.4})$$

To calculate the overfitting, we take the difference between the training performance and the test performance, divided by the training performance, and then express it as a percentage. By that metric, we can observe how the test performance varies from the training performance and make comments about the overfitting.

$$Percentage\ Overfitting = \frac{Training\ Score - Test\ Score}{Training\ Score} \quad (\mathbf{4.5})$$

IoU, Dice and Accuracy values will be used for training and accuracy scores in order to calculate the percentage overfitting values individually.

For the computational usage evaluation, inference speed, and memory footprint and parameter size of the model are analyzed. These analyses provide a fine insight about the model's computational performance in real world applications

For the inference speed test, we systematically measured the time taken for the model to process the test dataset. The following steps outline our methodology:

Data Preparation: The test dataset was preprocessed and loaded using a PyTorch Data Loader, ensuring consistent and efficient batching of input data.

Device Allocation: The model was deployed on both CPU or GPU devices for testing. This setup was crucial for optimizing performance measurements.

Evaluation Mode: The model was switched to evaluation mode using model.eval(). This step deactivates specific layers like dropout and batch normalization that are only relevant during the training phase, ensuring accurate performance metrics.

Warm-up Phase: An initial warm-up run was performed. This involved processing a single batch of data to stabilize performance and ensure the CPU and GPU was fully initialized. This step is particularly beneficial for obtaining consistent timing results on CPU and GPU devices.

Inference Time Measurement: The primary performance measurement was conducted by timing the model's inference process over the entire test dataset. Python's time module was utilized for precise timing. The inference loop was executed within a 'torch.no_grad()' context to prevent the computation of gradients, thereby reducing computational overhead and memory usage.

Data Analysis: The total inference time was recorded, and the average time per batch was calculated by dividing the total time by the number of batches. This metric provides a normalized measure of the model's speed, facilitating comparison across different models or configurations.

For the memory footprint and parameter size analysis of the model, the Torch summary library is utilized. Specifically, the 'torchsummary.summary()' function is employed with the input size specified, without requiring the original data. This method provides detailed information about the model's memory usage, including the input size, forward/backward pass size, parameters size, and estimated total size in megabytes.

Input Size (MB): The input size represents the memory required to store a single batch of input data during both forward and backward passes.

Forward/Backward Pass Size (MB): This metric refers to the memory consumption during the forward and backward passes of the model, including intermediate activations and gradients.

Parameter Size (MB): The parameter size represents the memory required to store the model's learnable parameters (weights and biases).

Estimated Total Size (MB): The estimated total size combines the input size, forward/backward pass size, and parameter size to provide an overall estimate of the memory usage for executing and storing the model.

By using these evaluation metrics, we can gain a comprehensive understanding of the model's performance across different aspects of segmentation accuracy and robustness. This multi-faceted evaluation helps ensure that the models are assessed thoroughly and can guide further improvements and refinements.

4.2 Comparative Analysis

4.2.1 Encoder-Net & Decoder-Net Results

Figs. 4.1 and 4.2 illustrate the output results from Encoder-Net and Decoder-Net compared with their ground truth values.

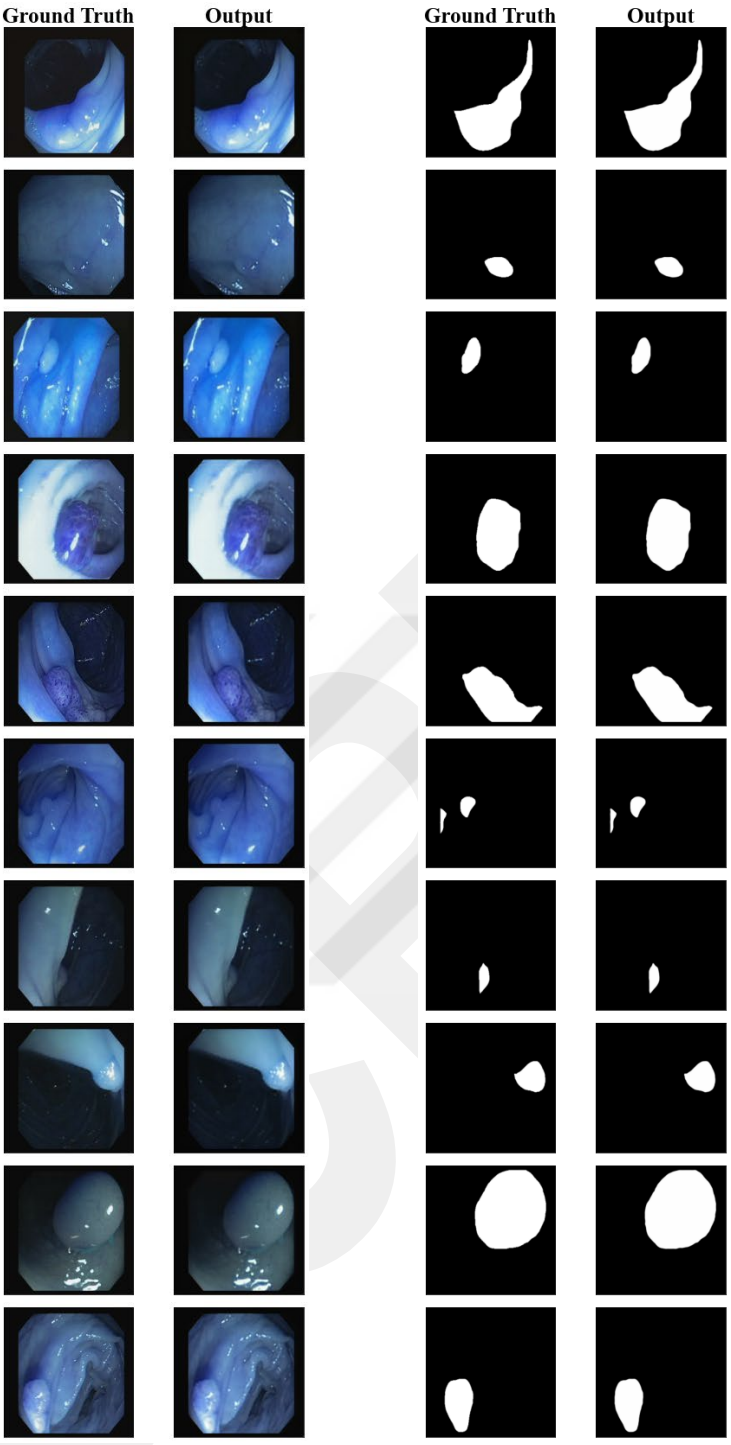


Figure 4.3 CVC-Clinic DB Sample Test Outputs versus Ground Truth of Encoder-Net and Decoder-Net from a) Output and Ground Truth of Encoder-net b) Output and Ground Truth of Decoder-Net

Figure 4.4 ISIC Sample Test Outputs versus Ground Truth of Encoder-Net and Decoder-Net from a) Output and Ground Truth of Encoder-net b) Output and Ground Truth of Decoder-Net

As it can be seen from visualized results of the Encoder-Net, the output is even clearer than the Original Input meaning that the autoencoder didn't lose much of features

from the original input and auto-handled the contrast of the images while shrinking the input size 16 times.

4.2.2 Tree-NET vs. Other Models

The algorithms performance is compared with traditional methods in terms of accuracy and computational efficiency.

4.2.2.1 Accuracy Results

The results of the proposed network are compared with other methods such as BS U-NET, U-NET, and U-NET++. The comparisons are visualized in Fig. 4.5 and Fig. 4.6. The IoU, Dice, and accuracy scores for the proposed model and the comparison models are detailed in Table 4.2. As demonstrated in the figures and table, our design surpasses the others in all metrics, showcasing superior performance across the board.

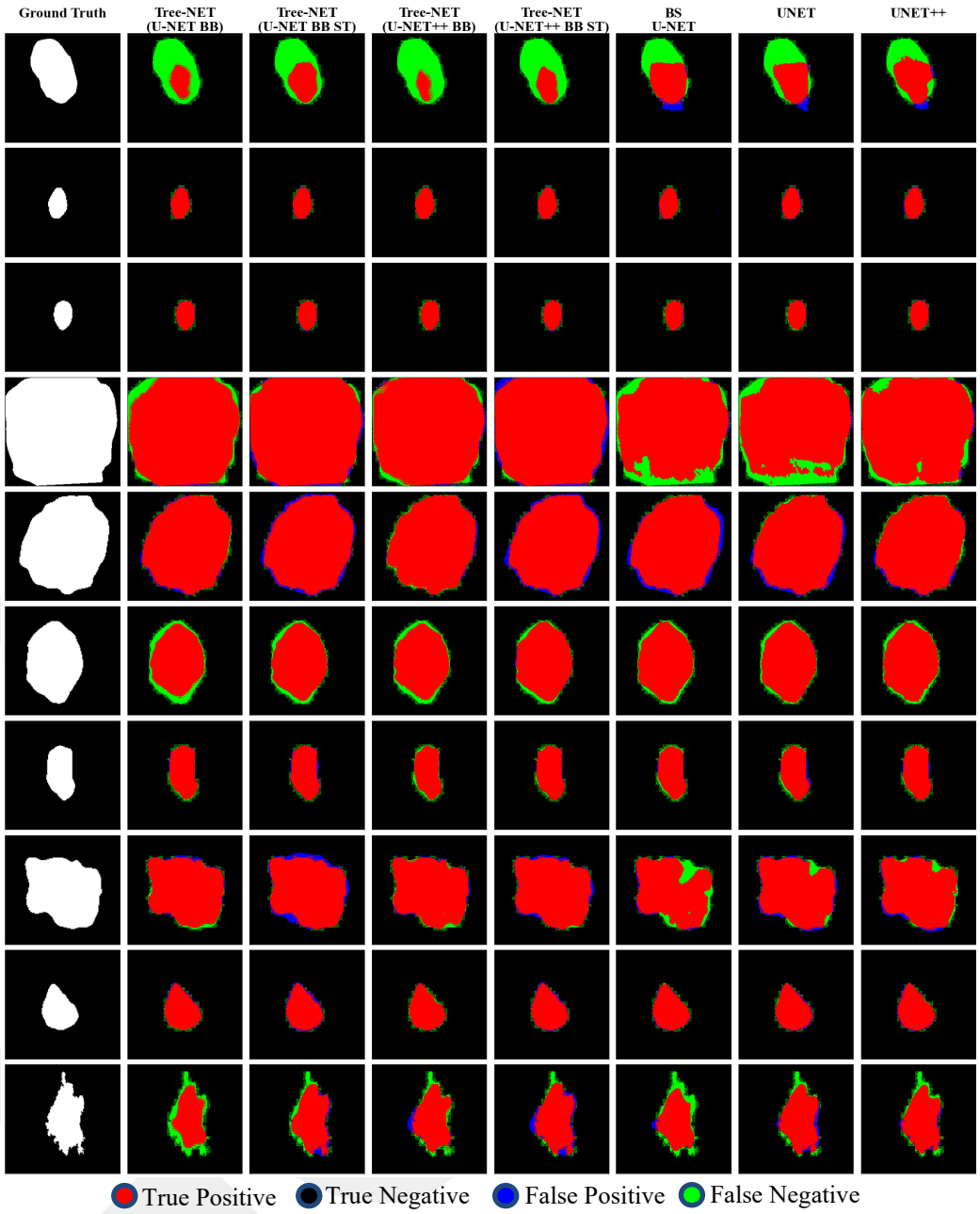


Figure 4.5 Colored visualization of segmentation results of ISIC-2018 test samples comparing the proposed network with BS U-NET, U-NET, and U-NET++

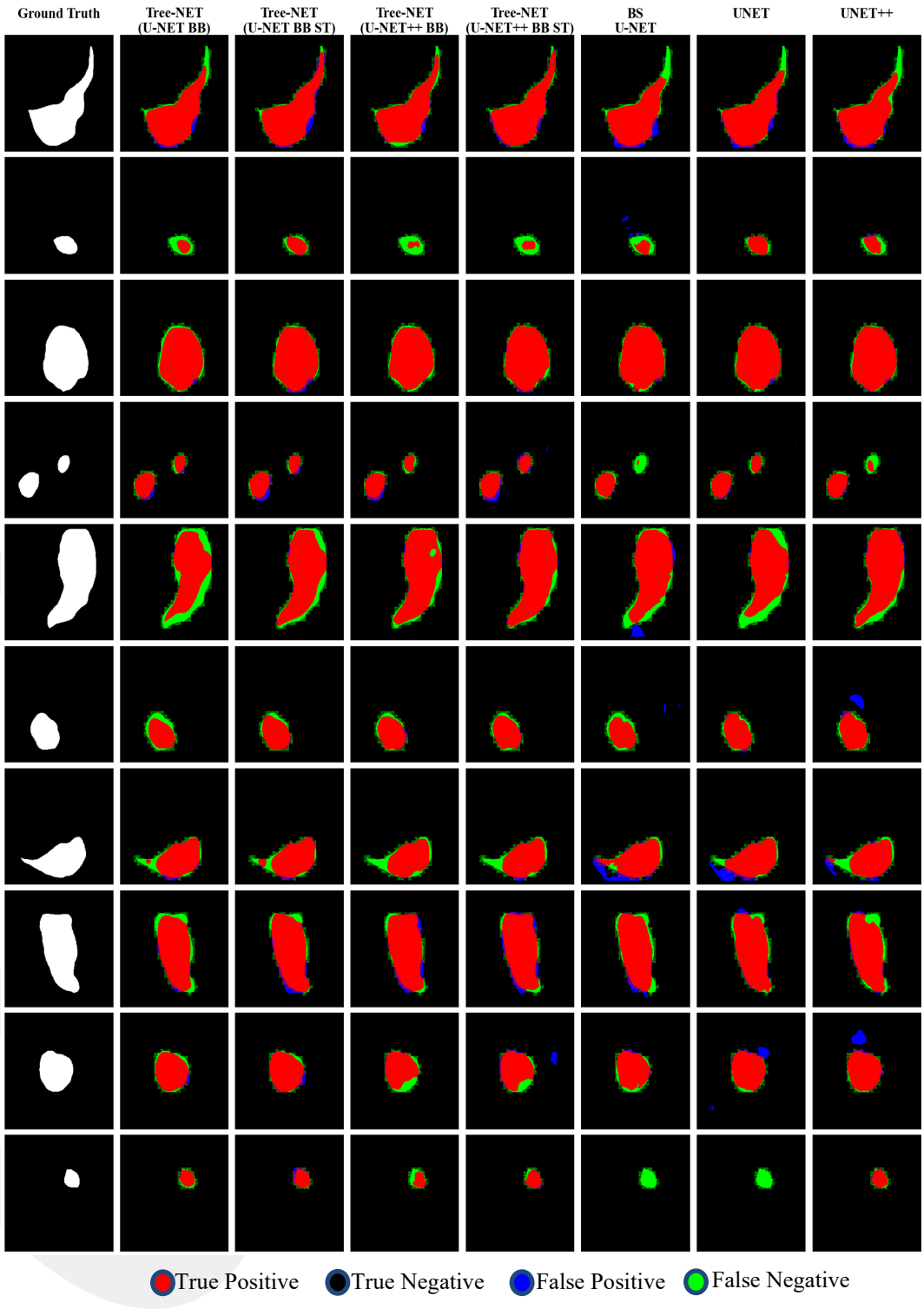


Figure 4.6 Colored visualization of segmentation results of CVC Clinic-DB test samples comparing the proposed network with BS U-NET, U-NET, and U-NET++

Table 4.2 Performance comparison of the proposed network (Tree-NET) against BS U-NET, U-NET, and U-NET++ in terms of IoU, Dice, and Accuracy scores

		Tree-NET				COMPARISON MODELS		
		U-NET BB ST	U-NET BB	U-NET++ BB ST	U-NET++ BB	BS U-NET	U-NET	U-NET++
CVC Clinic	**Dice**	0.8860	0.8894	0.8909	**0.8938**	0.8671	0.8734	0.8750
	IoU	**0.7518**	0.7288	0.7446	0.7153	0.6763	0.7103	0.7250
	ACC	0.9769	0.9754	**0.9776**	0.9775	0.9737	0.9761	0.9769
ISIC 2018	**Dice**	0.855	**0.8564**	0.8501	0.8549	0.8456	0.8392	0.8536
	IoU	0.788	**0.7956**	0.7852	0.7951	0.7792	0.7719	0.7948
	ACC	**0.9236**	0.9233	0.9218	0.9234	0.9190	0.9150	0.9228

4.2.2.2 Overfitting Check

For overfitting assessment, we calculate the percentage overfitting using Dice, IoU, and ACC score values. This evaluation is conducted on the CVC Clinic-DB dataset, which has a relatively small size and is more prone to overfitting.

Table 4.3 Overfitting check comparison of the proposed network (Tree-NET) against BS U-NET, U-NET, and U-NET++ in terms of IoU, Dice, and Accuracy scores on CVC Clinic-DB. BB stands for Backbones and PO stands for Percentage Overfitting

	Tree-NET		COMPARISON MODELS		
	U-NET BB	U-NET++ BB	BS U-NET	U-NET	U-NET++
Dice PO %	**5.751725**	5.908564	11.74165	9.734008	8.157304
IoU PO %	13.79428	16.32484	15.3801	14.52996	**11.30941**
ACC PO %	1.400634	**1.379365**	2.941047	1.432819	1.767825

The results presented in Table 4.3 indicate that the Tree-NET models exhibit lower overfitting compared to the other models overall. This demonstrates the superior generalization capability of Tree-NET.

4.2.2.3 Computational Efficiency

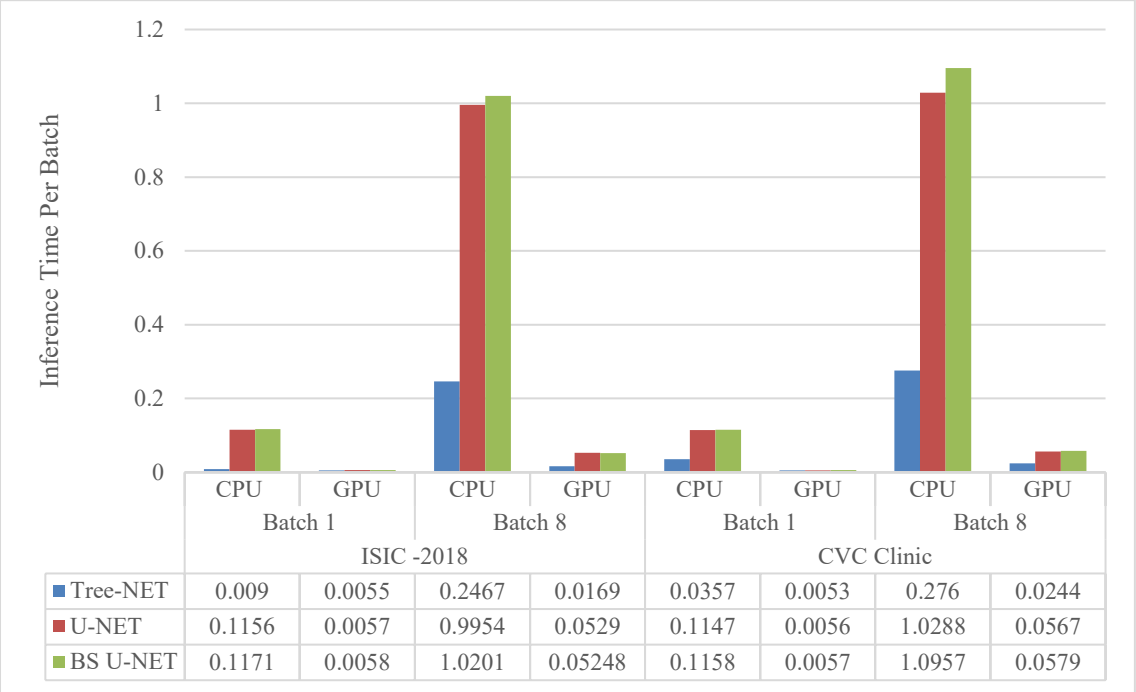
The computational efficiency of the proposed segmentation model was evaluated using Torch summary and computational speed analysis. The results from the Torch summary, which include input size, forward/backward pass size, parameters size, and estimated total size in megabytes, are summarized in Tables 4.4 and 4.5. The speed performance of different models, using the same backbones (BB), in terms of their inference times on GPU and CPU for different batch sizes is illustrated in Fig. 4.7 and 4.8.

Table 4.4 Computational Performance of Tree-NET Components

	ENCODER-NET	DECODER-NET	BRIDGE-NET (U-NET)	BRIDGE-NET (U-NET++)
Input Size (MB)	0.75	0.25	0.05	0.05
Forward/Backward Pass Size (MB)	95.94	91.84	31.00	69.25
Params Size (MB)	0.21	0.19	29.96	34.96
Estimated Total Size (MB)	96.90	92.28	61.00	104.25

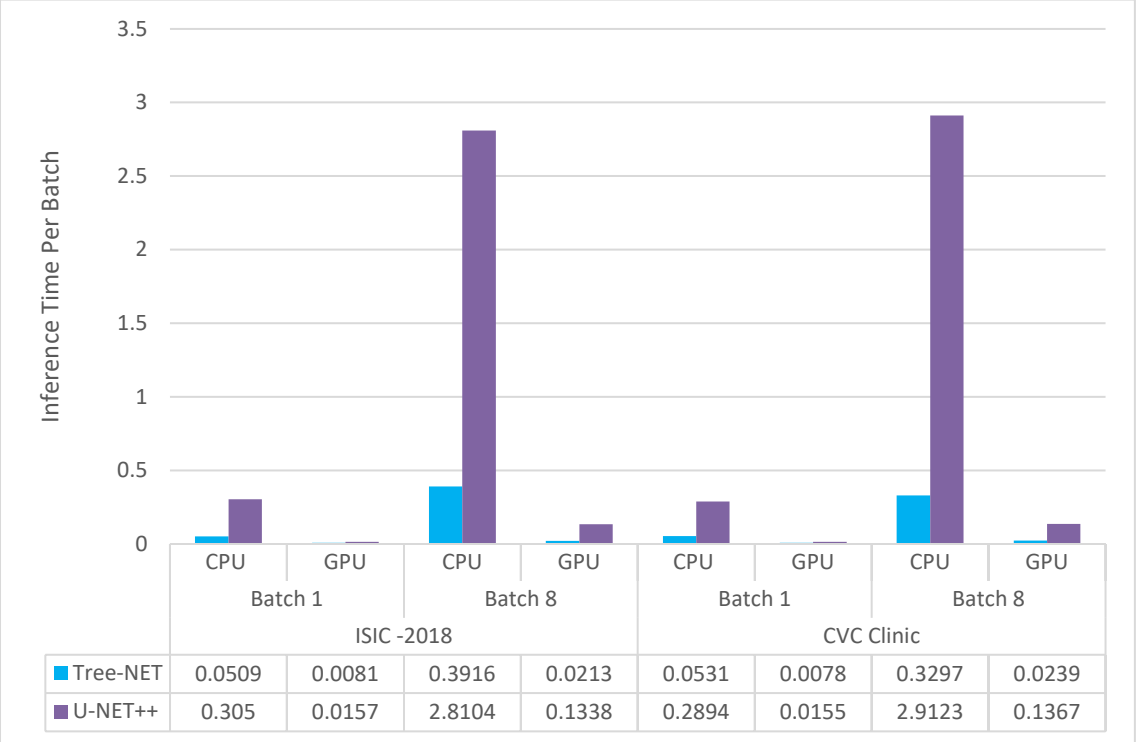
Table 4.5 Comparative Results for Computational Performance

	Tree-NET		COMPARISON MODELS		
	U-NET BB	U-NET++ BB	BS U-NET	U-NET	U-NET ++
Input Size (MB)	0.75	0.75	0.75	0.75	0.75
Forward/Backward Pass Size (MB)	**123.53**	161.78	495.02	495.00	1108.00
Parameter Size (MB)	30.15	35.15	29.98	**29.96**	34.96
Estimated Total Size (MB)	**154.43**	197.68	525.75	525.71	1143.71



	ISIC -2018				CVC Clinic			
	Batch 1		Batch 8		Batch 1		Batch 8	
	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU
Tree-NET	0.009	0.0055	0.2467	0.0169	0.0357	0.0053	0.276	0.0244
U-NET	0.1156	0.0057	0.9954	0.0529	0.1147	0.0056	1.0288	0.0567
BS U-NET	0.1171	0.0058	1.0201	0.05248	0.1158	0.0057	1.0957	0.0579

Figure 4.7 Speed Performance Comparison of U-NET BB Models on GPU and CPU for different batch sizes on both datasets



	ISIC -2018				CVC Clinic			
	Batch 1		Batch 8		Batch 1		Batch 8	
	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU
Tree-NET	0.0509	0.0081	0.3916	0.0213	0.0531	0.0078	0.3297	0.0239
U-NET++	0.305	0.0157	2.8104	0.1338	0.2894	0.0155	2.9123	0.1367

Figure 4.8 Speed Performance Comparison of U-NET++ BB Models on GPU and CPU for different batch sizes on both datasets

From these tables and figures, it is evident that the proposed model not only provides superior segmentation accuracy but also maintains high computational efficiency. It demonstrates a significantly lower estimated total size and inference time across different batches, devices, and datasets compared to several other state-of-the-art models. This efficiency is crucial for practical deployment, as it reduces computational resource demands.

4.3 Discussion

Our proposed segmentation model, Tree-NET, demonstrates superior performance and efficiency while exhibiting lower overfitting, making it highly advantageous for real-life applications. A critical factor contributing to its efficiency is the superior computational performance in terms of both model feed forward speed and memory footprint.

The memory usage difference becomes more pronounced with larger models. Specifically, the memory usage of the U-NET backboned Tree-NET is approximately 3.5 times smaller than that of the U-NET model, while the U-NET++ backboned Tree-NET is over 5.5 times smaller compared to the U-NET++ model. This increase in the memory footprint reduction ratio suggests that as the complexity of the Bridge-Net model increases, the Tree-NET model performs better in terms of memory usage. This efficiency is achieved through effective low-level representation. By processing smaller inputs with convolution operations, the model significantly reduces computational load without compromising accuracy.

To illustrate, feeding inputs of sizes (3×256×256) and (3×64×64) into the same model dramatically reduces the number of computations required, even if the number of parameters remains constant. This reduction is due to the computational complexity of convolution operations being directly related to input size. By efficiently handling smaller inputs through low-level representations, our model reduces the overall number of operations, leading to faster processing times and lower energy consumption.

Additionally, the architecture of our model, with its Encoder-Net and Decoder-Net components, is designed to be lightweight yet powerful. Each of these nets has approximately 50,000 trainable parameters, which is relatively low compared to the Bridge-Net's used and the other state-of-the-art models. Despite this compactness, the performance of our model remains unaffected, demonstrating that it can achieve high segmentation accuracy with fewer resources.

This efficiency has significant implications for real-life applications. In scenarios where computational resources are limited, such as mobile and embedded devices, our model's low computational cost and reduced memory footprint make it an ideal choice. It enables the deployment of high-performing segmentation algorithms in resource-constrained environments, expanding the applicability of advanced computer vision techniques to a broader range of devices and use cases. Moreover, the reduced computational demand translates to lower energy consumption, which is crucial for battery-powered devices and contributes to sustainability efforts by reducing the overall carbon footprint of computing operations.

The self-tuning approach performed particularly well on the CVC Clinic-DB dataset. The IoU score of the U-NET Backboned Tree-NET increased significantly from 0.7288 to 0.7518, demonstrating a promising improvement in accuracy. The key reason behind the effectiveness of this approach lies in maintaining harmonization between the separately trained components by further training them after assembly. This method ensures that unnecessary bottleneck features are removed, and essential features are emphasized.

Despite the accuracy and computational performance superiority of the Tree-NET model compared with other approaches, several limitations and challenges in the implementation of the algorithm need to be addressed for it to reach its full potential.

One significant limitation in our comparative testing environment is the absence of pre-trained models for the novel Tree-NET approach. Established algorithms, such as U-NET and U-NET++, often benefit from extensive pre-training on large datasets, facilitating fine-tuning and evaluation. This disparity poses a challenge for a direct performance comparison. For instance, the U-NET model, fine-tuned using the ISIC-2018

dataset, achieved a Dice score of 0.8674 and an Intersection over Union (IoU) score of 0.8491. In contrast, our implementation of U-NET, without fine-tuning, yielded a Dice score of 0.8392 and an IoU score of 0.7719. Similarly, the U-NET++ model, fine-tuned with the same dataset, reached a Dice score of 0.8822 and an IoU score of 0.8651, whereas our non-fine-tuned U-NET++ achieved a Dice score of 0.8750 and an IoU score of 0.7250 [12].

These results underscore the performance gains attributable to fine-tuning with a large, relevant dataset. Consequently, it is reasonable to infer that our models, including Tree-NET, could achieve superior performance metrics if pre-trained models tailored to our approach were available. This limitation highlights the necessity for developing and utilizing pre-trained models to enhance the comparative evaluation of novel algorithms.

One challenge is the meticulous tuning of component parameters and layers within the Tree-NET model. Both the Encoder-Net and Bridge-Net components are designed to progressively shrink the input size to create a compact representation. However, this aggressive downsizing can lead to an overly small bottleneck in the Bridge-Net, potentially rendering many learning parameters ineffective. Striking the right balance in the downsizing process is crucial. If autoencoders are designed with less downsizing, the model may suffer from worse performance due to an inability to effectively capture and compress the essential features of the input data. Conversely, excessive downsizing can lead to information loss and a subsequent drop in segmentation accuracy. This delicate balance requires careful parameter tuning and extensive experimentation to optimize model performance without compromising the integrity of the learned features.

The inherent complexity of the Tree-NET algorithm presents another significant challenge. The model's architecture involves multiple interdependent components, including the Encoder-Net, Bridge-Net, and Decoder-Net, each contributing to the overall performance. Introducing changes or improvements in one component often necessitates corresponding adjustments in the other components to maintain coherence and functionality. This interdependency increases the difficulty of model design and optimization, as each modification must be carefully evaluated to ensure it does not adversely affect the overall system. Moreover, the complexity of the algorithm may lead

to longer training times and increased computational requirements, posing practical constraints in resource-limited settings.

Another challenge is utilizing various approaches and models as the Bridge-Net component of our Tree-NET architecture. Applying a range of algorithms, particularly larger models, is computationally intensive and time-consuming. In the ISIC 2018 Lesion Boundary Segmentation Challenge, models like Mask RCNN, ensembled models, GANs, and U-NET variants demonstrated the highest IoU scores [36-40]. Among these, the hybrid Mask RCNN and segmentation model, inspired by DeepLab and PSPNet, achieved the highest IoU score of 0.802. This score was slightly superior to the U-NET backboned Tree-NET model, which had an IoU score of 0.796. However, this comparison is not entirely equitable due to the additional Mask-RCNN boundary detection in the hybrid model. For a more accurate assessment, the Tree-NET model should incorporate a similar backbone and leverage the Mask-RCNN boundary detection.

In addition to these challenges, the Tree-NET model, like many deep learning approaches, requires a substantial amount of annotated data for training. The process of annotating medical images is both expensive and time-consuming, often necessitating the expertise of medical professionals. This requirement can be a significant barrier to the widespread adoption and deployment of the Tree-NET model, especially in domains where annotated data is scarce or difficult to obtain.

While the Tree-NET model demonstrates promising performance and accuracy in medical image segmentation, addressing these limitations and challenges is essential for its successful implementation and adoption. Creating standardized benchmarks, optimizing component parameters, managing algorithmic complexity, securing sufficient annotated data, and ensuring generalization across diverse datasets are key areas that require ongoing research and development. By tackling these challenges, the Tree-NET model can be further refined to provide reliable and efficient solutions in the field of medical image analysis.

Chapter 5

Conclusions and Future Prospects

5.1 Conclusions

In conclusion, the implementation of the Tree-NET model, enhanced by a self-tuning approach, has demonstrated substantial advancements in the field of medical image segmentation, particularly for polyp and skin cancer detection. This innovative approach has surpassed traditional methods, showcasing superior efficiency, accuracy, and robustness while exhibiting lower overfitting. By integrating this cutting-edge algorithm, significant improvements in segmentation performance have been achieved, all while maintaining a lower computational cost compared to conventional techniques. The Tree-NET model leverages the power of deep learning to effectively capture multi-scale contextual information, thereby enabling precise and efficient segmentation of polyps and skin cancer.

One of the standout features of this approach is its versatility. The Tree-NET model can be applied to a wide variety of image segmentation tasks, not just limited to medical imaging. This flexibility is largely due to the robustness of the Bridge-Net layer type. The architecture allows for any segmentation algorithm to be incorporated within the Bridge-Net framework, which can significantly reduce the overall computational cost of the algorithm. This adaptability makes the Tree-NET model a highly valuable tool for a range of applications beyond its initial use cases.

Moreover, the inclusion of bottleneck feature supervision within the Tree-NET model has proven to be a promising strategy for enhancing segmentation tasks. This method focuses on supervising the most informative features, resulting in unprecedented levels of performance and accuracy. By concentrating on critical features, the algorithm

ensures that the segmentation process is both effective and efficient, even in complex and varied medical imaging scenarios.

The potential of this approach extends to various fields where image segmentation is crucial. Whether in medical diagnostics, industrial inspection, or remote sensing, the Tree-NET model's advanced capabilities promise significant improvements. The model's ability to deliver high performance with reduced computational demands positions it as a leading solution for future developments in image segmentation.

As a result, bottleneck feature supervision and the overall architecture of the Tree-NET model stand out as groundbreaking strategies for enhancing segmentation tasks. This innovation represents a significant step forward in medical imaging and beyond, offering unprecedented levels of precision, efficiency, and adaptability. The ongoing refinement and application of these techniques will likely lead to further breakthroughs, solidifying their role in the advancement of image segmentation technology.

5.2 Societal Impact and Contribution to Global Sustainability

The proposed architecture for medical segmentation, with its potential application in tasks such as brain tumor and breast cancer detection, carries substantial societal impact and contributes significantly to global sustainability in multiple ways. Firstly, widespread adoption of this method across the medical sector could revolutionize disease detection and diagnosis, leading to improved patient outcomes and reduced healthcare burdens. By enabling early and accurate detection of diseases, including those beyond polyp and skin image segmentation, this technology empowers healthcare professionals with valuable tools for proactive intervention and treatment planning.

In terms of global sustainability, the utilization of advanced segmentation techniques in medical settings aligns with broader efforts to enhance healthcare efficiency and resource allocation. Early disease detection facilitated by such methods not only improves individual health outcomes but also contributes to the broader goal of

sustainable healthcare systems, reducing long-term healthcare costs and optimizing resource utilization.

Moreover, the impact of this technology extends beyond medical applications. The same segmentation principles that underpin polyp detection can be adapted for tasks in agriculture, environmental monitoring, and beyond. For instance, precise segmentation can aid in crop monitoring, pest detection, and land management practices, supporting sustainable agricultural practices and food security initiatives.

In the academic realm, the adoption of this model alongside other cutting-edge technologies fuels scientific progress and innovation. Collaborative research efforts leveraging advanced segmentation methods pave the way for new discoveries, insights, and publications, driving scientific advancements and knowledge dissemination.

In summary, the societal impact and contribution to global sustainability of this segmentation architecture are manifold. From enhancing healthcare outcomes and resource efficiency in medical settings to enabling innovation across diverse sectors like agriculture and academia, the broad applicability of this technology underscores its transformative potential in shaping a more sustainable and resilient future. Continued research and adoption of such technologies promise to catalyze positive societal change and scientific advancements on a global scale.

5.3 Future Prospects

The proposed architecture has promising future applications beyond polyp segmentation. It can be adapted for various tasks by using different backbone architectures instead of U-NET. Integrating alternative backbones like FCN, DeepLab, and their variants enhances performance and scalability, opening new opportunities in medical imaging, remote sensing, and computer vision. The pre-trained models also will be applied to the Tree-NET for all the three components of it. That could provide much better performance in terms of accuracy.

In the Self-tuning approach, we can use the frozen layers to limit the trainable parameters which could have a significant impact on the results. The unfrozen layers will be around where each component is combined where the significant harmonization could be captured.

In conclusion, exploring different backbones, using pre-trained models, improving the Self-tuning and applying the model on different segmentation tasks offers exciting possibilities for refining segmentation methods and expanding applications in deep learning and image analysis across diverse domains.

BIBLIOGRAPHY

[1] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-030-00889-5_1.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015. doi: 10.1007/978-3-319-24574-4_28.

[3] J. Ji, X. Lu, M. Luo, M. Yin, Q. Miao, and X. Liu, "Parallel Fully Convolutional Network for Semantic Segmentation," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2020.3042254.

[4] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. doi: 10.1109/CVPR.2019.00657.

[5] I. Despotović, B. Goossens, and W. Philips, "MRI segmentation of the human brain: Challenges, methods, and applications," 2015. doi: 10.1155/2015/450341.

[6] V. I. Butoi, J. J. Gonzalez Ortiz, T. Ma, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "UniverSeg: Universal Medical Image Segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2023. doi: 10.1109/ICCV51070.2023.01960.

[7] J. Tan, Y. Huo, Z. Liang, and L. Li, "Expert knowledge-infused deep learning for automatic lung nodule detection," *J Xray Sci Technol*, vol. 27, no. 1, 2019, doi: 10.3233/XST-180426.

[8] L. Song, K. Geoffrey, H. K.-E. S. with Applications, and undefined 2020, "Bottleneck feature supervised U-Net for pixel-wise liver and tumor segmentation," *Elsevier*, Accessed: Jan. 15, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417419308486?casa_token=KsyI8c51k6cAAAAA:U2yJ2nBpUkLoT9JysRRCzAeut5VOoZmLdClCaeVWJpBfmVZIC1kgLrXunapb96B66ssykvukiKq99A

[9] A. Lou, S. Guan, and M. Loew, "CaraNet: context axial reverse attention network for segmentation of small medical objects," *Journal of Medical Imaging*, vol. 10, no. 01, 2023, doi: 10.1117/1.jmi.10.1.014005.

[10] J. Zhang, Y. Zhang, Y. Jin, J. Xu, and X. Xu, "MDU-Net: multi-scale densely connected U-Net for biomedical image segmentation," *Health Inf Sci Syst*, vol. 11, no. 1, 2023, doi: 10.1007/s13755-022-00204-9.

[11] H. Zunair and A. Ben Hamza, "Sharp U-Net: Depthwise convolutional network for biomedical image segmentation," *Comput Biol Med*, vol. 136, 2021, doi: 10.1016/j.compbiomed.2021.104699.

[12] R. Azad *et al.*, "Medical Image Segmentation Review: The success of U-Net," *ArXiv*, vol. abs/2211.14830, 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:254043694

[13] B. Stewart and P. Kleihues, "World cancer report," 2003, Accessed: Jan. 15, 2024. [Online]. Available: https://www.env.go.jp/air/asbestos/commi_hhmd/03/ext01.pdf

[14] J. Bernal, J. Sánchez, and F. Vilariño, "Towards automatic polyp detection with a polyp appearance model," in *Pattern Recognition*, 2012. doi: 10.1016/j.patcog.2012.03.002.

[15] J. G. B. Puyal *et al.*, "Endoscopic Polyp Segmentation Using a Hybrid 2D/3D CNN," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020. doi: 10.1007/978-3-030-59725-2_29.

[16] M. Kavurmaci, "Skin cancer," in *Medical Nursing*, 2023. doi: 10.5206/uwomj.v85i2.2240.

[17] N. DurgaRao and Dr. G. Sudhavani, "A Survey on Skin Cancer Detection System," *Int J Eng Res Appl*, vol. 07, no. 06, 2017, doi: 10.9790/9622-0706055964.

[18] W. C. R. F. International, "Worldwide cancer data," World Cancer Research Fund International.

[19] Skin Cancer Foundation, "Skin Cancer Facts & Statistics - Skin Cancer Foundation," Skin Cancer Facts & Statistics - Skin Cancer Foundation.

[20] D. Vienneau *et al.*, "Effects of radon and UV exposure on skin cancer mortality in Switzerland," *Environ Health Perspect*, vol. 125, no. 6, 2017, doi: 10.1289/EHP825.

[21] K. P. Zhu *et al.*, "Gated Recurrent Units Based Neural Network For Tool Condition Monitoring," *Int J Mach Tools Manuf*, vol. 20, no. 2, 2017.

[22] S. Asgari Taghanaki, K. Abhishek, J. P. Cohen, J. Cohen-Adad, and G. Hamarneh, "Deep semantic segmentation of natural and medical images: a review," *Artif Intell Rev*, vol. 54, no. 1, 2021, doi: 10.1007/s10462-020-09854-1.

[23] E. Wu, K. Wu, R. Daneshjou, D. Ouyang, D. E. Ho, and J. Zou, "How medical AI devices are evaluated: limitations and recommendations from an analysis of FDA approvals," 2021. doi: 10.1038/s41591-021-01312-x.

[24] Q. Huang, J. Sun, H. Ding, X. Wang, and G. Wang, "Robust liver vessel extraction using 3D U-Net with variant dice loss function," *Comput Biol Med*, vol. 101, 2018, doi: 10.1016/j.compbiomed.2018.08.018.

[25] W. Chen, B. Liu, S. Peng, J. Sun, and X. Qiao, "S3D-UNET: Separable 3D U-Net for brain tumor segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019. doi: 10.1007/978-3-030-11726-9_32.

[26] M. Frid-Adar, A. Ben-Cohen, R. Amer, and H. Greenspan, "Improving the segmentation of anatomical structures in chest radiographs using U-net with an imagenet pre-trained encoder," in *Lecture Notes in Computer Science*

(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018. doi: 10.1007/978-3-030-00946-5_17.

[27] J. I. Orlando *et al.*, "U2-net: A Bayesian u-net model with epistemic uncertainty feedback for photoreceptor layer segmentation in pathological oct scans," in *Proceedings - International Symposium on Biomedical Imaging*, 2019. doi: 10.1109/ISBI.2019.8759581.

[28] Z. Zhong *et al.*, "3D fully convolutional networks for co-segmentation of tumors on PET-CT images," in *Proceedings - International Symposium on Biomedical Imaging*, 2018. doi: 10.1109/ISBI.2018.8363561.

[29] J. Dolz, C. Desrosiers, and I. Ben Ayed, "IVD-net: Intervertebral disc localization and segmentation in MRI with a multi-modal UNet," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11397 LNCS, pp. 130–143, 2019, doi: 10.1007/978-3-030-13736-6_11.

[30] Q. Jin, Z. Meng, C. Sun, H. Cui, and R. Su, "RA-UNet: A Hybrid Deep Attention-Aware Network to Extract Liver and Tumor in CT Scans," *Front Bioeng Biotechnol*, vol. 8, Dec. 2020, doi: 10.3389/FBIOE.2020.605132/FULL.

[31] X. Li, H. Chen, X. Qi, Q. Dou, C. W. Fu, and P. A. Heng, "H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes," *IEEE Trans Med Imaging*, vol. 37, no. 12, pp. 2663–2674, Dec. 2018, doi: 10.1109/TMI.2018.2845918.

[32] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and their applications in machine learning: a survey," *Artif Intell Rev*, vol. 57, no. 2, 2024, doi: 10.1007/s10462-023-10662-6.

[33] P. Li, Y. Pei, and J. Li, "A comprehensive survey on design and application of autoencoder in deep learning," 2023. doi: 10.1016/j.asoc.2023.110176.

[34] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, and F. Vilariño, "Polyp - CVC-ClinicDB," in *Computerized Medical Imaging and Graphics*, 2015.

[35] P. Tschandl, C. Rosendahl, and H. Kittler, "Data descriptor: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci Data*, vol. 5, 2018, doi: 10.1038/sdata.2018.161.

[36] C. Qian *et al.*, "A Two-Stage Method for Skin Lesion Analysis," *arXiv preprint arXiv:1809.03917*, 2018.

[37] Z. Al Nazi and T. A. Abir, "Automatic Skin Lesion Segmentation and Melanoma Detection: Transfer Learning Approach with U-Net and DCNN-SVM," 2020. doi: 10.1007/978-981-13-7564-4_32.

[38] Du Hao, Seok Jeon Young, Ng Dianwen, Yuan Ngiam Kee, and Feng Mengling, "Team HolidayBurned at ISIC CHALLENGE 2018," 2018, Accessed: Aug. 05, 2024. [Online]. Available: amazonaws.com/27353b72

[39] Y. Xue, L. Gong, W. Peng, X. Huang, and Y. Zheng, "Automatic Skin Lesion Analysis with Deep Networks," *ISIC 2018 Challenge*, 2018.

[40] Koohbanani Navid Alemi, Jahanifar Mostafa, Tajeddin Neda Zamani, and Gooya Ali, “Leveraging Transfer Learning for Segmenting Lesions and their Attributes in Dermoscopoy Images,” 2018.

CURRICULUM VITAE

2016 – 2021	B.Sc., Electrical and Electronics Engineering, Abdullah Gul University, Kayseri, TURKEY
2022 – Present	M.Sc., Electrical and Computer Engineering, Abdullah Gül University, Kayseri, TURKEY

SELECTED PUBLICATIONS AND PRESENTATIONS

There are no publications and presentations yet