A Project Report On

Segmentation Of the Transverse section of the Carotid Artery using U-Net

Submitted in partial fulfillment of the requirement for the 8th semester

Bachelor of Engineering

in

Computer Science and Engineering

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified)

Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560111

Submitted By

Nagaraj K 1DS19CS096

Niranjan Reddy 1DS19CS099

Ravula Akhilesh Reddy 1DS19CS127

Naga Swaroop S N 1DS19CS095

Under the guidance of

Anitha M

Associate Professor, CSE , DSCE

Mr. Yashwanth

Co-guide - Industry

2022 - 2023

Department of Computer Science and Engineering

DAYANANDA SAGAR COLLEGE OF ENGINEERING

Bangalore - 560111

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Dayananda Sagar College of Engineering
(An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified)
Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560111
Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project entitled **Segmentation Of the Transverse section of the Carotid Artery using U-Net** is a bonafide work carried out by **Nagaraj K [1DS19CS096]**, **Niranjan Reddy [1DS19CS099], Ravula Akhilesh Reddy [1DS19CS127]** and **Naga Swaroop S N [1DS19CS095]** in partial fulfillment of 8th semester, Bachelor of Engineering in Computer Science and Engineering under Visvesvaraya Technological University, Belgaum during the year 2022-23.

Anitha M	**Dr. Ramesh Babu D R**	**Dr. B G Prasad**
Guide	Vice Principal & HOD	Principal
Associate Prof. CSE, DSCE	CSE, DSCE	DSCE
Signature:	Signature:	Signature: .....

Name of the Examiners: Signature with date:

1. ....

2. ....

Acknowledgement

We are pleased to have successfully completed the project **Segmentation Of the Transverse section of the Carotid Artery using U-Net**. We thoroughly enjoyed the process of working on this project and gained a lot of knowledge doing so.

We would like to take this opportunity to express our gratitude to **Dr. B G Prasad**, Principal of DSCE, for permitting us to utilize all the necessary facilities of the institution.

We also thank our respected Vice Principal, HOD of Computer Science & Engineering, DSCE, Bangalore, **Dr. Ramesh Babu D R**, for his support and encouragement throughout the process.

We are immensely grateful to our respected and learned guide, **Anitha M**, Associate Professor CSE, DSCE and our co-guide **Mr. Yashwanth** for their valuable help and guidance. We are indebted to them for their invaluable guidance throughout the process and their useful inputs at all stages of the process.

We thank all the faculty and support staff of Department of Computer Science, DSCE. Without their support over the years, this work would not have been possible.

Finally, we would like to express our deep appreciation towards our classmates and our family for providing us with constant moral support and encouragement. They have stood by us in the most difficult of times.

Nagaraj K 1DS19CS096

Niranjan Reddy 1DS19CS099

Ravula Akhilesh Reddy 1DS19CS127

Naga Swaroop S N 1DS19CS095

ABSTRACT

Image segmentation is critical in medical imaging analysis for precise extraction of anatomical structures and regions of interest. Common carotid artery (CCA) segmentation is particularly important in cardiovascular health, aiding in the detection and evaluation of various conditions. Atherosclerosis, characterized by plaque formation due to lipid accumulation, narrows and hardens arteries, leading to serious complications like heart attacks and strokes. Atherosclerosis-related diseases are significant causes of mortality globally.Among the main causes of mortality in the US are heart disease and stroke. In 2019, 874,614 people died from cardiovascular disease, and one heart attack occurred every 40 seconds,accounting for 1 in 6 fatalities from cardiovascular disease in 2020.Traditional CNN architectures, like U-Net, face challenges in preserving spatial information during downsampling and upsampling, resulting in reducing the efficiency and accuracy of segmentation. U-Net uses skip connections to address this by fusing feature maps to retain fine details. U-Net has successfully been used in medical image analysis.This paper aims to enhance the automation process of CCA segmentation using the EAR U-Net model, which combines U-Net with attention mechanisms and Residual blocks. Attention mechanisms focus on relevant features and suppress noise, while Residual blocks facilitate gradient propagation and network optimization. By leveraging EAR U-Net's strengths, we aim to overcome CNN limitations in CCA segmentation, contributing to improved diagnosis, treatment, and monitoring of cardiovascular diseases while preserving data integrity.

Table of Contents

1 Introduction **1**

1.1 The problem 1

1.2 Real World Application 2

1.3 Organization of Project Report 4

2 Problem Statement and Proposed Solution **5**

2.1 Problem Statement 5

2.2 Existing Systems 5

2.2.1 Traditional U-Net: 5

2.2.2 Attention-Based U-Net: 6

2.2.3 Deep Learning Approaches: 7

2.2.4 Multi-Stage Approaches: 8

2.2.5 Hough Transform Segmentation 9

2.3 Proposed Solution 10

2.3.1 Encoder-Decoder Framework 10

2.3.2 Skip-Connections 12

2.4 System Requirements 14

3 Literature Survey **15**

3.1 U-Net:Convolutional Networks for Biomedical Image Segmentation . 15

3.2 Liver Vessels Segmenatation Based On 3D Residual U-Net 16

3.3 Attention U-Net:Learning Where to Look for the Pancreas 17

3.4 EAR-U-Net:EfficientNet and attention-based residual U-Net for automatic liver segmentation in CT 18

4 Architecture and System Design **20**

4.1 Software Overview 20

4.1.1 System Block Diagram 20
4.1.2 Data Flow Diagram 22

5 Implementation **24**
5.1 Implementation Platform 24
5.1.1 Hardware 24
5.1.2 Software 24
5.2 Implementation Details 25
5.2.1 Organisation of files 25
5.2.2 Implementation Workflow 30
5.3 Dataset 35

6 Testing **37**

7 Experimentation and Results **39**
7.1 Experimentation 39
7.2 Results 40

8 Conclusion and Future work **43**

9 References **45**

List of Figures

2.1 Traditional U-Net 6
2.2 Fig :Attention-Based U-Net 7
2.3 Hough Transform Segmentation 10
2.4 Encoder Decoder Framework 11

4.1 Fig :Block Diagram 20
4.2 Data Flow 23

5.1 Home page 26
5.2 Prediction page 27
5.3 Home Page Steps 28
5.4 Prediction Page Steps 29
5.5 ZoomIn and ZoomOut in Prediction Page 30
5.6 CCA Images with Ground Truth mask 36

7.1 Results Comparing the Ground Truth and Predicted Mask 42

List of Tables

3.1 Segmentation Methods in Medical Imaging 19

7.1 Performance Metrics for Segmentation Methods 41

Chapter 1

Introduction

1.1 The problem

Accurate segmentation of the common carotid artery (CCA) plays a crucial role in analyzing cardiovascular health by precisely extracting anatomical structures and identifying abnormalities from digital images. However, the process of CCA segmentation is not without its challenges.

However, if they want to make sure, they can get in touch with the candidate's alma mater and ask if the certificate is legitimate. The length of this procedure varies depending on the university where the enquiry is made. It would be exceedingly challenging to identify fraudulent certifications if the individual worked with the university to lie about them.

1. Image Quality Variability:

One of the primary hurdles in CCA segmentation arises from the inherent variability in image quality. Medical images often suffer from noise, artifacts, and variations in acquisition parameters, which degrade the performance of segmentation algorithms. Factors like motion artifacts, low contrast, and image distortion introduce significant inconsistencies, making it difficult to accurately delineate CCA boundaries.

2. Complex Anatomy and Shape Variations:

The intricate anatomical structure of the carotid artery presents another challenge in segmentation. The CCA exhibits variations in shape, size, and bifurcation patterns among individuals, making it challenging to develop a universal segmentation approach. Additionally, the presence of neighboring structures, such as surrounding tissues and adjacent vessels, adds

complexity, increasing the likelihood of segmentation errors and inaccuracies.

3. Preservation of Fine Details:

Preserving fine details and boundaries within the segmented CCA region is crucial for precise analysis and diagnosis. However, conventional segmentation methods, including manual tracing and basic algorithms, often struggle to capture intricate details effectively. Neural network-based segmentation methods, such as convolutional neural networks (CNNs), encounter challenges in maintaining high precision due to the loss of spatial information during downsampling and upsampling stages.

4. Plaque Heterogeneity and Morphological Changes:

Atherosclerosis, a common condition affecting the carotid artery, introduces further complexities. The heterogeneous nature of arterial plaques, with varying compositions and deposits at different locations, poses challenges in accurately segmenting the diseased regions. Moreover, morphological changes in the artery, such as stenosis and ulcerations, add to the complexity, necessitating robust algorithms capable of capturing these dynamic characteristics.

1.2 Real World Application

Segmentation of the common carotid artery (CCA) has several real-world applications in the field of cardiovascular health. By accurately delineating and extracting the CCA from medical images, segmentation techniques can contribute to various aspects of diagnosis, treatment, and monitoring of cardiovascular conditions:

1. Atherosclerosis Detection:

Atherosclerosis, a condition characterized by plaque buildup in the arterial walls, is a leading cause of cardiovascular diseases. Accurate CCA segmentation aids in the detection and evaluation of atherosclerosis, allowing for the quantification of plaque buildup and assessment of the degree of stenosis (narrowing). This information helps in determining the risk of complications, such as heart attacks and strokes, and guiding appropriate treatment strategies.

2. Treatment Planning and Monitoring:

Precise segmentation of the CCA enables medical professionals to plan and monitor treatments effectively. For instance, in interventions like carotid endarterectomy or stenting, CCA segmentation provides critical information about the location and extent of arterial blockages, assisting surgeons in making informed decisions during the procedure. Subsequent monitoring of the CCA through segmentation helps evaluate treatment efficacy and disease progression.

3. Risk Assessment and Proactive Management:

Segmenting the CCA allows for risk assessment in individuals with cardiovascular risk factors or suspected atherosclerosis. By analyzing the presence and characteristics of plaque, medical practitioners can identify high-risk patients who may require more aggressive management strategies. Early detection and proactive management based on CCA segmentation findings can potentially prevent life-threatening events and improve long-term patient outcomes.

4. Treatment Planning and Monitoring:

Research and Clinical Trials: CCA segmentation plays a vital role in research studies and clinical trials focused on cardiovascular health. Accurate and consistent segmentation enables the analysis of large-scale image datasets, facilitating the investigation of novel diagnostic approaches, treatment modalities, and disease progression patterns. CCA segmentation serves as a valuable tool for evaluating the efficacy and safety of new interventions or therapies in clinical trials.

1.3 Organization of Project Report

The project report is organized as follows:

In Chapter (2) we discuss the problem statement and the proposed solution. We also take a look at the systems that exist today and the drawbacks they face.

Chapter (3) takes a more in-depth look at various hardware and software-based solutions that exist, with a survey on existing literature available.

Chapter (4) looks at the architecture of the proposed solution with an overview of the system design, utilizing system block diagrams and data flow diagrams.

Chapter (5) dives into the Implementation of the solution, by describing the hardware and software requirements, along with dataset descriptions and implementation details.

Chapter (6) describes our testing process, while Chapter (7) looks at our experimentation process and the obtained results.

Chapter (8) summarizes our findings and concludes the paper.

Chapter 2

Problem Statement and Proposed Solution

2.1 Problem Statement

The study focuses on the segmentation of the transverse section of the common carotid artery. It aims to compare the performance of EAR U-Net, attention-based U-Net, and traditional U-Net models for this task. The goal is to identify the most accurate and effective model for accurately segmenting the carotid artery in the transverse section.

2.2 Existing Systems

2.2.1 Traditional U-Net:

The U-Net architecture, initially proposed by Ronneberger et al., has gained significant popularity in the field of medical image segmentation, particularly for tasks involving the segmentation of the common carotid artery (CCA). The traditional U-Net architecture follows an encoder-decoder structure that incorporates skip connections. This design allows the network to effectively capture both global and local contextual information.

The encoder path in the U-Net is responsible for downsampling the input image, thereby reducing its spatial dimensions. This downsampling process helps in capturing high-level contextual information from the image. By doing so, the network can learn and understand the broader context of the CCA within the image.On the other hand, the decoder path in the U-Net

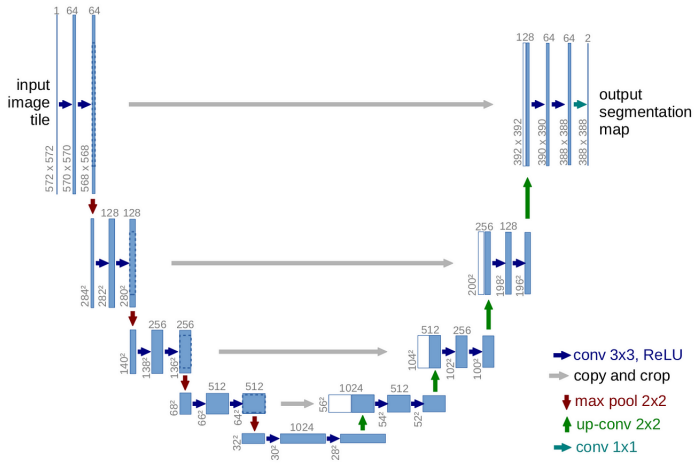

Figure 2.1: Traditional U-Net

utilizes skip connections to merge the high-resolution features from the encoder with the upsampled feature maps. These skip connections enable the network to incorporate detailed information from earlier stages of the network, which is essential for precise segmentation of the CCA. By fusing the low-level and high-level features, the U-Net can effectively capture fine details and delineate boundaries within the segmented regions.

Despite its success, the traditional U-Net architecture may face certain challenges. One such challenge is capturing fine details within the segmented regions. Due to the downsampling operations in the encoder path, some fine details may be lost or not accurately represented in the feature maps. This can result in a loss of important information, making it challenging for the network to precisely delineate boundaries and capture intricate structures within the segmented regions.

2.2.2 Attention-Based U-Net:

Attention mechanisms have been successfully integrated into U-Net architectures to enhance the performance of CCA segmentation. By incorporating attention modules, attention-based U-Net models can effectively identify and prioritize relevant features within the CCA images, allowing them to allocate more attention to areas crucial for accurate segmentation. This attention-driven approach addresses the limitations of traditional U-Net models, which often struggle with capturing intricate details and handling noise in medical images. The integration of attention mechanisms enables the model to selectively focus on informative regions while sup-

pressing irrelevant noise, leading to improved segmentation accuracy for CCA.

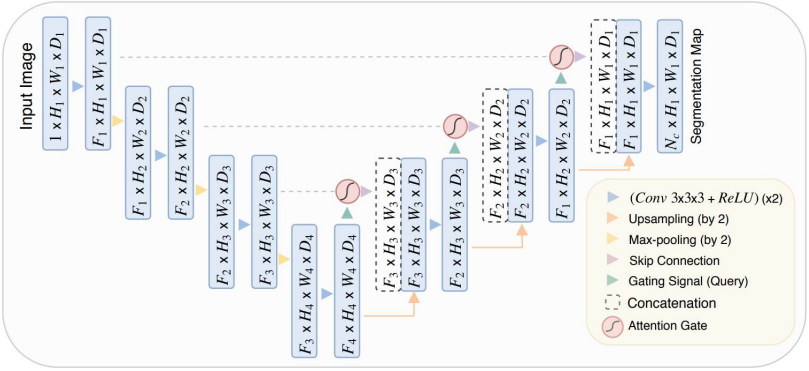

Figure 2.2: Fig :Attention-Based U-Net

However, it is important to note a couple of limitations associated with attention-based U-Net models. First, the incorporation of attention mechanisms introduces additional computational complexity to the model. This increased complexity can significantly impact both the training and inference time, making it challenging to deploy the model in real-time applications or resource-constrained environments. Second, attention-based models heavily rely on the availability of large amounts of labeled data for training. In medical imaging, acquiring labeled data can be time-consuming and expensive, limiting the practicality of attention-based U-Net models in certain scenarios. Insufficient training data can also hinder the model's ability to learn meaningful attention patterns, potentially leading to suboptimal segmentation performance.

2.2.3 Deep Learning Approaches:

Deep learning techniques, specifically convolutional neural networks (CNNs), have emerged as a prominent approach for accurate CCA (Common Carotid Artery) segmentation. These CNN-based models leverage the power of hierarchical feature learning from large-scale datasets to automatically extract discriminative information and precisely segment the CCA region. The utilization of various CNN architectures, such as DenseNet, ResNet, and EfficientNet, has been explored in the context of CCA segmentation to enhance the accuracy of the segmentation process. These deep learning approaches have demonstrated their potential in effectively handling the complexity of CCA segmentation tasks by capturing intricate

patterns and representing complex relationships between image features. By leveraging the hierarchical features learned through CNNs, these models can achieve improved segmentation outcomes and contribute to advancing the field of medical image analysis.

Despite their successes, it is essential to consider the limitations of deep learning techniques in CCA segmentation. Firstly, deep learning models require substantial amounts of labeled data for training, which can be challenging to obtain, particularly in medical imaging domains. The availability of limited annotated CCA images may hinder the model's ability to generalize and perform accurately on unseen cases. Secondly, CNN-based models, with their complex architectures, can be computationally demanding and require significant computational resources during both training and inference stages. This computational complexity may limit their real-time applicability in resource-constrained environments or when immediate results are needed. Furthermore, the interpretability of deep learning models is often limited, making it challenging to understand the underlying decision-making process of the model. Therefore, ensuring the transparency and interpretability of the segmentation results becomes an important consideration when utilizing deep learning techniques for CCA segmentation tasks.

2.2.4 Multi-Stage Approaches:

Some methodologies in CCA (Common Carotid Artery) segmentation adopt multi-stage frameworks to enhance segmentation accuracy by incorporating additional processing steps. These approaches aim to improve the accuracy and reliability of CCA segmentation by leveraging pre-processing techniques and iterative refinement.In these multi-stage frameworks, the first stage often involves pre-processing techniques such as vessel enhancement and vessel tracking. Vessel enhancement techniques enhance the visibility of blood vessels in the image, making it easier to extract the CCA region accurately. These techniques can involve filters, such as vesselness filters or Hessian-based filters, which enhance the tubular structures present in the image. Vessel tracking techniques then trace the path of the vessels, providing additional information for CCA segmentation.

Following the pre-processing stage, the pre-processed images are passed

through segmentation algorithms to refine the segmentation results iteratively. These segmentation algorithms leverage the enhanced vessel information and prior knowledge about the CCA region to iteratively improve the accuracy of the segmentation. The iterative refinement process helps to gradually enhance the segmentation boundaries, adjusting and optimizing the segmentation results based on the specific characteristics of the CCA.

While multi-stage frameworks offer potential benefits in terms of improved segmentation accuracy, there are certain drawbacks associated with these techniques. One limitation is the requirement to provide the coordinates of the center and the radius of the circle being sought in the image. This introduces the need for manual input and potentially subjective decisions, which may lead to inconsistencies or variations in the segmentation results. Additionally, the use of fixed coordinates assumes a specific size and shape of the CCA, which may not be applicable in all cases and can limit the flexibility of the segmentation method. Therefore, the reliance on predetermined coordinates poses a challenge when dealing with images containing different variations in the CCA size, shape, and orientation.

2.2.5 Hough Transform Segmentation

The Hough Circle Transform is a widely used computer vision technique that specifically focuses on identifying circular shapes within an image. It is an extension of the Hough Line Transform, which is primarily designed for detecting straight lines. The Hough Circle Transform operates by converting the image from its original Cartesian coordinate system to a specialized parameter space. In this parameter space, circles are represented by three essential parameters: the x-coordinate of the center, the y-coordinate of the center, and the radius of the circle.

To identify circles using the Hough Circle Transform, the algorithm scans through the image and accumulates votes in the parameter space for potential circle centers and radii. Each point within the image casts votes in the parameter space, corresponding to possible circle centers and radii that pass through that point. By accumulating these votes, the algorithm constructs a parameter space that highlights regions with high voting values, indicating the probable presence of circles in the image.

Once the parameter space is constructed, the Hough Circle Transform aims to find peaks within this space. These peaks correspond to the most significant circles present in the image. By detecting these peaks, the algorithm identifies the centers and radii of the circles that best fit the observed patterns in the image. This process allows for the robust detection of circles, even in the presence of noise or incomplete circle boundaries.

It is important to note that the Hough Circle Transform is a powerful technique for detecting circles, but its effectiveness heavily depends on appropriate parameter settings and the quality of the image being analyzed. Additionally, the Hough Circle Transform can be computationally intensive, especially for large images or when searching for circles with varying sizes. Therefore, careful consideration of parameters and efficient implementation are crucial for achieving accurate and efficient circle detection using this technique.

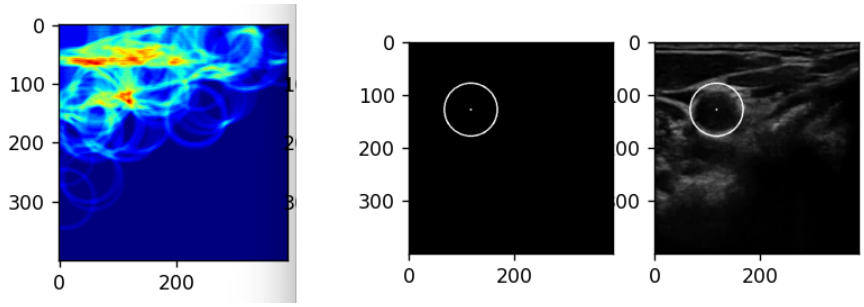

Figure 2.3: Hough Transform Segmentation

2.3 Proposed Solution

Combining the various techniques that are used for segmentation we found the better result with the EAR-UNET technique

2.3.1 Encoder-Decoder Framework

In this phase, the input is passed to the first layer, which implements EfficientNetB4. This is used to balance the width, depth, and resolution of the model. The building block of EfficientNet is the MBConv block. The `mb_conv_block` function defines a Mobile Inverted Residual Bottleneck block. It takes inputs as the input tensor, arguments for the block activation function, dropout regularization, and prefix as an optional prefix for layer names.

The function first checks if the block has a squeeze-and-excitation phase

by evaluating the `se_ratio` from `block_args`. It determines the `bn_axis` based on the image data format. Next, it performs the expansion phase by applying a 1x1 convolution to increase the number of filters if the expansion ratio is not equal to 1. Otherwise, it is set to the value passed to the input tensor. Then, a depthwise convolution is applied to `x` using the specified kernel size and strides. Batch normalization and activation functions are applied afterwards.

If the block has a squeeze-and-excitation phase (has_se is true), it performs the following operations:

- Global average pooling is applied to the kernal size .
- The tensor is reshaped and passed through 1x1 convolutional layers to reduce and expand the number of filters.
- The result is multiplied element-wise with kernal size to excite certain channels.

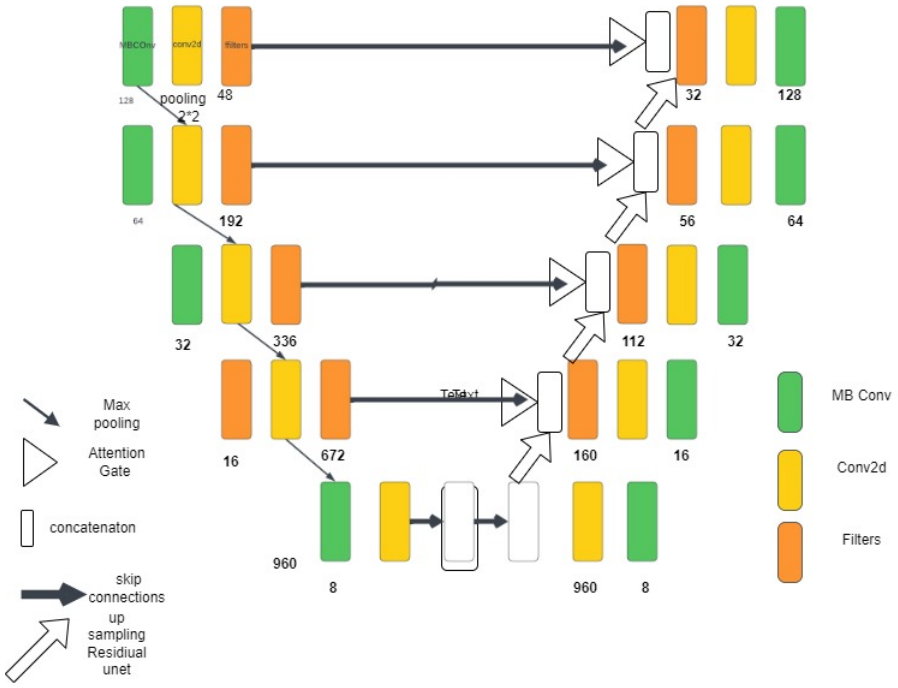

Figure 2.4: Encoder Decoder Framework

Finally, an output phase is performed by applying a 1x1 convolution to reduce the number of filters to the desired output size. Batch normalization is applied, and if the block has skip connections (`id_skip is True`) and certain conditions are met, the input tensor is added to the output

tensor. Optionally, dropout regularization can be applied before adding the tensors.

The decoder part of the EAR U-Net architecture comprises a series of residual blocks. These blocks play a crucial role in mitigating the vanishing gradient problem and facilitating gradient flow during training. Each residual block consists of two convolutional layers, followed by batch normalization and a skip connection. The skip connection connects the input of the block to the output of the second convolutional layer, allowing the network to learn the identity function.This skip connection helps in efficient gradient propagation and preserves important spatial information during the decoding process.

2.3.2 Skip-Connections

Skip connections, also known as residual connections, play a crucial role in deep neural networks by addressing the vanishing gradient problem and improving gradient flow during training. The vanishing gradient problem arises when the gradients diminish as they propagate backward through numerous layers, hindering the effective learning of early layers. Skip connections enable the direct flow of information from earlier layers to deeper layers, facilitating the learning process and allowing for better optimization of the network.

In the context of attention mechanisms, skip connections can be further enhanced by integrating an attention-based gating mechanism. This technique, often referred to as an "attention gate" or "attention skip connection," leverages the power of attention mechanisms to selectively emphasize or suppress specific features based on their relevance. By incorporating attention-based

mechanisms within skip connections, the network gains the ability to focus on important features, leading to improved performance in various tasks such as image segmentation, object detection, and natural language processing.

The functioning of an attention skip connection involves several steps:

- Main Path : The input, which in this case originates from the EfficientNetB4 model, goes through a series of sequential transformations

as it traverses the layers of the neural network. These transformations progressively extract hierarchical representations of the input data.

- Attention Mechanism: At a specific point in the network architecture, an attention mechanism is introduced. This mechanism typically includes a separate branch that processes the intermediate representation obtained from the main path. The purpose of the attention mechanism is to compute attention weights or attention maps that capture the importance or relevance of different segments within the intermediate representation.

- Attention Weights: The attention mechanism calculates attention weights that assign importance scores to different segments of the intermediate representation. These attention weights act as a form of soft selection or weighting mechanism, allowing the network to focus on the most relevant features while downplaying less important ones.

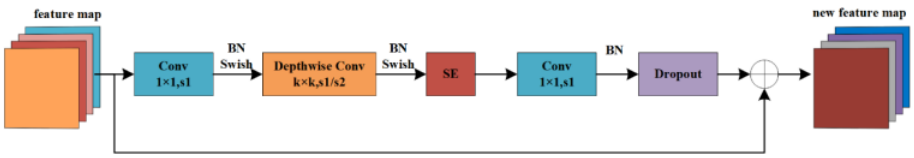

Fig. 3. MBConv Block

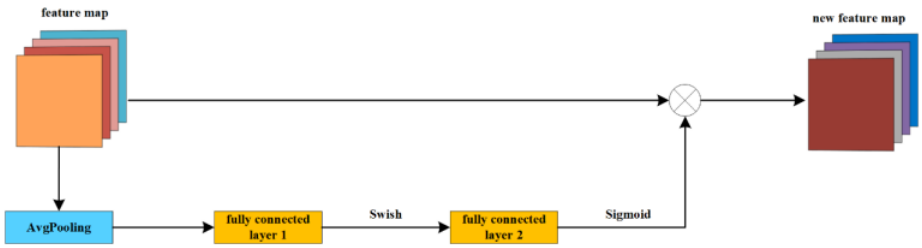

Fig. 4. Squeeze and Excitation Block

- Gating: The attention weights obtained from the attention mechanism are utilized to gate or modulate the intermediate representation obtained from the main path. This gating operation

  generally involves element-wise multiplication between the attention weights and the intermediate representation. The resulting output represents a refined version of the intermediate representation, with emphasized or suppressed features based on their attention weights.

- Skip Connection: The refined output from the gating operation, which can be considered as an enhanced version of the intermediate

representation, is element-wise added to the original input tensor, denoted as "x." This addition establishes the skip connection or residual connection. The purpose of the skip connection is to ensure that the original information from earlier layers can bypass the subsequent transformations and directly propagate to deeper layers. By preserving this direct information flow, the network can effectively capture and utilize the valuable information from earlier stages of processing.

- Further Processing: The skip connection, encapsulating the refined information from the attention mechanism, continues to propagate through the network and undergoes subsequent transformations in subsequent layers. This enables the network to leverage the enhanced information from the attention-based gating mechanism in combination with the newly acquired information from deeper layers. By integrating attention-based gating within skip connections, the network becomes capable of learning to selectively emphasize or suppress specific features based on their relevance. This attribute proves beneficial in various applications, such as image segmentation, object detection, or natural language processing, where the network needs to focus on important and discriminative features while disregarding irrelevant or noisy ones.

  It is important to note that the exact implementation of attention-based skip connections may vary depending on the specific architecture and requirements of the neural network model being utilized. Different attention mechanisms, such as self-attention or spatial attention, can be employed to compute attention weights. Additionally, various gating mechanisms, such as sigmoid or softmax gating, can be used to modulate the intermediate representation. The choice of these mechanisms depends on the specific task and the desired behavior of the network.

2.4 System Requirements

- Processor : 4 Core, 8 Thread Processor
- Processor : Minimum of 16 GB ram for computation

Chapter 3

Literature Survey

3.1 U-Net:Convolutional Networks for Biomedical Image Segmentation

Authors: Olaf Rsanneberger, Philipp Fischer, and Thomas Brox In this paper they introduces a novel network strategy and training method specifically designed for biomedical image segmentation. The authors address the challenge of limited annotated samples by extensively employing data augmentation techniques. The U-Net architecture is composed of two crucial components: a decoding path to capture contextual information and a symmetric encoding path to facilitate precise localization. This end-to-end trainable network achieves remarkable performance with a small number of images, surpassing the capabilities of previous segmentation methods in the medical imaging domain.

A key innovation of the U-Net architecture lies in its utilization of upsampling operators instead of pooling operators, allowing for an increase in the output image resolution. By incorporating features from the contracting path into the encoding path, the network enhances localization accuracy. This integration of multi-scale features aids in precisely identifying and segmenting objects within the images. Additionally, the authors leverage multiple feature channels in the upsampling path to enable the propagation of extensive context information to the higher resolution layers.

The results of the proposed approach demonstrate its effectiveness, as it achieves an average Intersection over Union (IOU) score of 77%. This per-

formance surpasses that of other existing algorithms for biomedical image segmentation. The combination of the U-Net architecture, data augmentation techniques, and the precise integration of features contributes to the impressive segmentation results, thus proving the efficacy of this approach in the field of biomedical image analysis.

3.2 Liver Vessels Segmenatation Based On 3D Residual U-Net

Authors: Wei Yu , Bin Fang , Yongqing Liu , Mingqi Gao, Shenhai Zheng , Yi Wang Description: In this paper, the authors introduce a novel approach for segmenting hepatic veins and portal veins from abdominal CT images. They propose the utilization of residual blocks within the U-Net architecture to enhance convergence speed and improve the representation of discriminative features. The authors start by describing the concept of residual blocks, which are a fundamental component of the ResNet architecture. Residual blocks allow for the direct propagation of information from earlier layers to deeper layers, addressing the vanishing gradient problem and improving the overall learning capability of the network. By incorporating residual blocks into the U-Net architecture, the authors aim to leverage their benefits for the specific task of liver vessels segmentation. Based on this idea, the authors develop a new architecture called the 3D Residual U-Net. This architecture takes advantage of the 3D nature of the abdominal CT images to capture spatial dependencies and provide more comprehensive information for accurate segmentation. By integrating residual blocks, the 3D Residual U-Net is expected to achieve faster convergence and better representation of important features related to hepatic veins and portal veins.

To further enhance the performance of the segmentation, the authors introduce additional techniques. They employ 3D morphological closed operations, which help in smoothing the edges of the segmented vessels and improving the overall accuracy of the segmentation results. Additionally, volume analytics are utilized to remove noise blocks, ensuring a more precise and reliable segmentation outcome.

The authors evaluate the performance of their proposed method using

dice coefficients, a commonly used metric for assessing the similarity between the segmented and ground truth regions. The average dice coefficient achieved by the 3D Residual U-Net on the liver vessels segmentation task is reported to be 71.7.

3.3 Attention U-Net:Learning Where to Look for the Pancreas

Authors: Ozan Oktay Jo Schlemper , Loic Le Folgoc , Matthew Lee , Mattias Heinrich Description : The paper propose a novel attention gate (AG) mechanism to enhance the U-Net architecture's ability to focus on target structures of varying size and shape. By incorporating attention gates into the U-Net, the models gain the capability to effectively suppress irrelevant regions in input images while emphasizing salient features crucial for specific tasks.

The authors demonstrate the effectiveness of attention gates in the context of medical image analysis, particularly in CT pancreas segmentation. This segmentation task is challenging due to the large variation in organ size and shape, as well as low tissue contrast. By applying attention gates within the standard U-Net architecture, referred to as Attention U-Net, the authors aim to improve the accuracy of pancreas segmentation.

To evaluate their proposed solution, the authors utilize two benchmark datasets: TCIA Pancreas CT-82 and multi-class abdominal CT-150. These datasets contain a diverse range of medical images, enabling a comprehensive assessment of the attention gate mechanism's performance. The results of their experiments indicate that the attention gate (AG) approach significantly improves the prediction accuracy across different training datasets and sizes. This improvement is achieved without the need for multiple CNN models or ensemble techniques. By effectively learning where to focus within the input images, the Attention U-Net with attention gates demonstrates its ability to handle the challenging task of pancreas segmentation and provides more accurate segmentation results.It is important to note that the attention gate mechanism introduced in this paper allows the U-Net architecture to selectively emphasize salient features while suppressing irrelevant regions.

3.4 EAR-U-Net:EfficientNet and attention-based residual U-Net for automatic liver segmentation in CT

Authors: Jinke Wang1, Xiangyang Zhang , Peiqing Lv, Lubiao Zhou, Haiying Wang Description: propose a novel network framework called EAR-U-Net for precise and automated liver segmentation. This method combines EfficientNetB4, attention gate, and residual learning techniques to achieve significant advancements in the field of liver segmentation.

The EAR-U-Net framework builds upon the U-Net architecture and incorporates several key components to improve segmentation accuracy. In the encoding stage, the authors leverage the powerful feature extraction capabilities of EfficientNetB4, which is a state-of-the-art network architecture known for its efficiency and effectiveness in image recognition tasks. By incorporating EfficientNetB4, EAR-U-Net can capture more informative and discriminative features, enabling more precise liver segmentation.

Additionally, an attention gate is introduced within the skip connection of the U-Net architecture. This attention gate selectively emphasizes relevant features specific to the liver segmentation task while suppressing extraneous regions. By incorporating attention mechanisms, EAR-U-Net focuses its attention on the most salient liver regions, leading to improved segmentation accuracy.

To address the challenge of gradient vanishing, the authors replace the conventional convolution in the decoder with a residual block. Residual learning allows for the direct flow of information from earlier layers to deeper layers, mitigating the vanishing gradient problem and further enhancing the accuracy of the segmentation outcomes.

To validate the performance of the proposed method, extensive evaluations are conducted on two benchmark datasets: LiTS17 and SLiver07. The authors compare their method against classical networks including FCN, U-Net, Attention U-Net, and Attention Res-U-Net. The experimental results consistently demonstrate the superior performance of EAR-U-Net across all five standard evaluation metrics, particularly on the SLiver07

dataset.Furthermore, the authors participate in the MICCIA-LiTS17 challenge, where EAR-U-Net achieves an impressive Dice per case score of 0.952. This demonstrates the effectiveness of the proposed method in real-world scenarios and its ability to outperform other state-of-the-art liver segmentation approaches.

Table 3.1: Segmentation Methods in Medical Imaging

Author	**Image Modality**	**Method/Model**
B. Behboodi et. al. [1]	Ultrasound Imaging	U-Net
Amiri. M et. al. [4]	Ultrasound Imaging	Two Stage U-Net
J. Ding et. al. [5]	Ultrasound Imaging	ReAgU-Net
M. Xie et. al. [6]	Ultrasound Imaging	Two Novel Convolutional U-Net
Meshram NH et. al. [7]	Ultrasound Imaging	Dilated U-Net
J. Li et. al. [8]	X-ray, ultrasound, MRI images	Multi Scale Fusion U-Net
M. Xie et. al. [9]	Ultrasound Imaging	CNN - U-Net
Y. Li et. al. [10]	Ultrasound Imaging	Light Weight U-Net
W. Yu et. al. [11]	CT Scan Imaging	3D Residual U-Net
M. Jiang et. al. [12]	Ultrasound Imaging	3DUS U -Net (UNet+SAN)
Z. Zhou et. al. [13]	Ultrasound Imaging	U-Net++
H. Li et. al. [14]	CT Scan Imaging	EAR U-Net

Chapter 4

Architecture and System Design

4.1 Software Overview

4.1.1 System Block Diagram

Figure 4.1: Fig :Block Diagram

The system block diagram, as shown in Figure 5, provides an overview of the data flow and processing steps involved in an image analysis model. The diagram showcases the different components and techniques employed to enhance accuracy and extract meaningful information from input images.

At the core of the system, the input image serves as the initial data point for analysis. To effectively process the image and extract relevant features, an EfficientNetB4 model is utilized. The EfficientNetB4 is a pre-trained neural network that has been specifically designed for image recognition tasks. Leveraging its powerful capabilities, the model applies various transformations and operations to the input image, enabling it to extract discriminative features that are crucial for the subsequent analysis.

One key element in the system block diagram is the utilization of skip connections. These connections facilitate the fusion of information from multiple layers within the neural network. By incorporating skip connec-

tions, the model can leverage features from earlier layers, which might contain important contextual information. This enables the model to capture both low-level and high-level features, resulting in a more comprehensive representation of the input image.

To further enhance the model's ability to focus on relevant regions within the image, attention gates are introduced. Attention gates allow the model to selectively emphasize specific regions of interest while suppressing irrelevant areas. By applying attention mechanisms, the model becomes more adept at attending to important features, thereby improving its overall performance in image analysis tasks.

In addition to skip connections and attention gates, the system employs residual blocks. These blocks are utilized to capture intricate details and refine the extracted features. By incorporating residual learning, the model can effectively address the issue of gradient vanishing, which can hinder the training process in deep neural networks. The residual blocks enable the direct propagation of information from earlier layers to deeper layers, facilitating the learning process and enhancing the model's ability to capture fine-grained details in the image.

The output of the model is generated by a Conv2D layer with a sigmoid activation function. This layer produces a probability map that highlights the areas of interest within the input image. The probability map provides a visual representation of the model's predictions, indicating the likelihood of certain regions belonging to the target class or containing important features.

Overall, the system block diagram represents a comprehensive pipeline for image analysis, incorporating advanced techniques and components to improve accuracy and extract meaningful information from input images. The integration of EfficientNetB4, skip connections, attention gates, residual blocks, and the Conv2D layer with sigmoid activation function enables the model to process images effectively, capture relevant features, and generate informative outputs for various image analysis tasks.

4.1.2 Data Flow Diagram

The process of automatic segmentation using the U-Net model involves several steps, as outlined below. These steps encompass data loading, preprocessing, model creation, training, evaluation, and prediction visualization.

To begin, the data is loaded using the "Load Data" function. This function reads and splits the image and mask data into separate sets for training, validation, and testing. This division ensures that the model is trained on a diverse range of data and can generalize well to unseen examples. The "Read Image" and "Read Mask" functions are then applied to the respective datasets. These functions serve to read and preprocess the images and masks, preparing them for subsequent processing.

Once the images and masks are preprocessed, the data undergoes conversion into a TensorFlow-compatible format using the "TF Parse" function. This step is crucial as TensorFlow is the framework used for building and training the U-Net model. The "TF Dataset" function is then applied to create TensorFlow datasets specifically for training and validation. This step organizes the data into a suitable format for feeding into the model during the training process.

Moving on, the "Create Model" function is employed to establish the U-Net model architecture. The U-Net architecture is widely used for image segmentation tasks due to its effectiveness in capturing contextual information and spatial details. This function sets up the necessary layers and connections within the U-Net model, enabling it to learn and make predictions about the input images.

With the model architecture in place, the training process begins using the "Model Fit" function. This function takes the training dataset and applies it to the U-Net model, updating the model's weights based on a defined loss function and optimizer. The training continues for a specified number of epochs, allowing the model to learn from the data and improve its segmentation performance over time.

Once the training is complete, the model's performance is evaluated using the "Evaluation" function. This function applies the testing dataset

to the trained model and calculates various metrics, such as the dice coefficient, recall, and precision. These metrics provide quantitative measures of how well the model performs in segmenting the desired regions.

Finally, the "Read Image," "Read Mask," and "Mask Parse" functions are employed in a loop to process individual test images. For each test image, the model predicts the corresponding mask, indicating the segmented regions. The predicted masks are then concatenated with the original images and masks using the "Concat" function. This step allows for a visual comparison between the original image, the original mask, and the predicted mask. The resulting composite image is plotted using the "Plot Image" function, providing a side-by-side visualization of the segmentation results.

In summary, the process of automatic segmentation using the U-Net model involves loading and preprocessing the data, creating the U-Net model architecture, training the model, evaluating its performance, and generating segmentation predictions. This systematic approach enables accurate and efficient segmentation of images, with the final results visualized for analysis and interpretation.

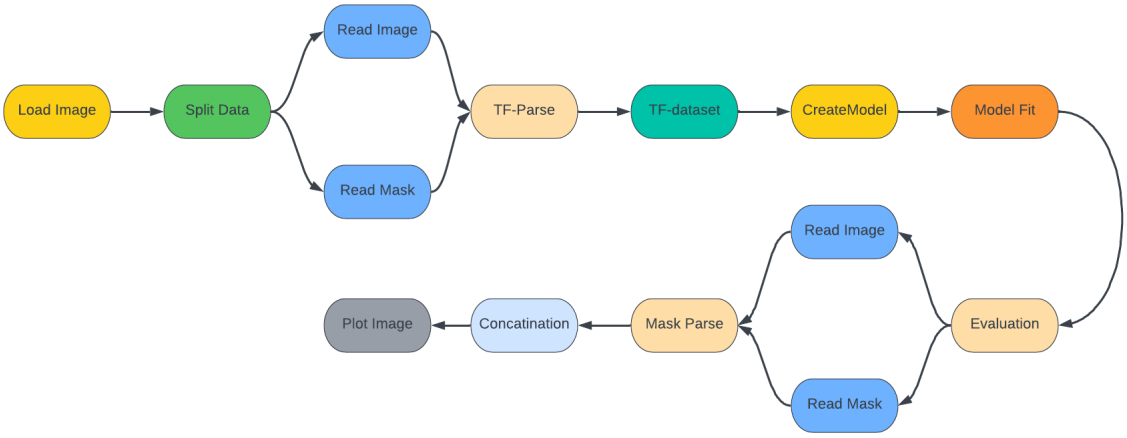

Figure 4.2: Data Flow

Chapter 5

Implementation

5.1 Implementation Platform

5.1.1 Hardware

(i) Processor: 11th Gen Intel(R) Core(TM) i5-11@2.40GHz 2.42 GHz

(ii) RAM: 8GB

(iii) Storage platform: NVME SAMSUNG

5.1.2 Software

(i) Operating System: Windows 64bit

(ii) Software Used : TensorFlow,OpenCV, Flask ,Jupyter Notebook

(iii) Programming Languages : Python 3, HTML

(iv) Server: Flask server

5.2 Implementation Details

5.2.1 Organisation of files

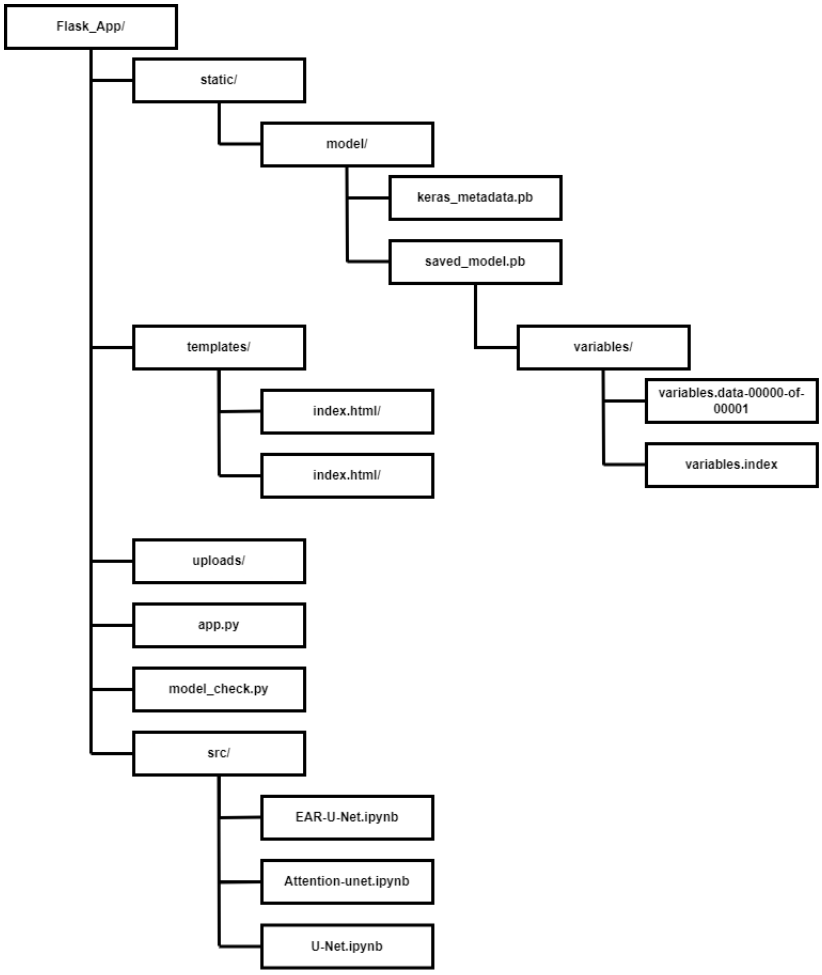

Fig : Project directory structure

The file structure of the Flask app, named "Flask_App," is organized into several directories. At the top level, we have the following directories: "static," "templates," "uploads," and "src," along with the Python files "app.py" and "model_check.py."

The "static" directory is used for storing static files required by the application. Inside "static," there is a "model" directory that holds the saved model files, including "keras_metadata.pb," "saved_model.pb," and a "variables" subdirectory. The "variables" directory contains the variables data associated with the model, consisting of "variables.data-00000-of-00001" and "variables.index" files.

The "templates" directory contains the HTML templates used for rendering the web pages. It includes two files: "index.html" and "out-

put.html." "index.html" serves as the template for the initial upload page, where users can select and upload an image. On the other hand, "output.html" s the template used to display the output and prediction results.

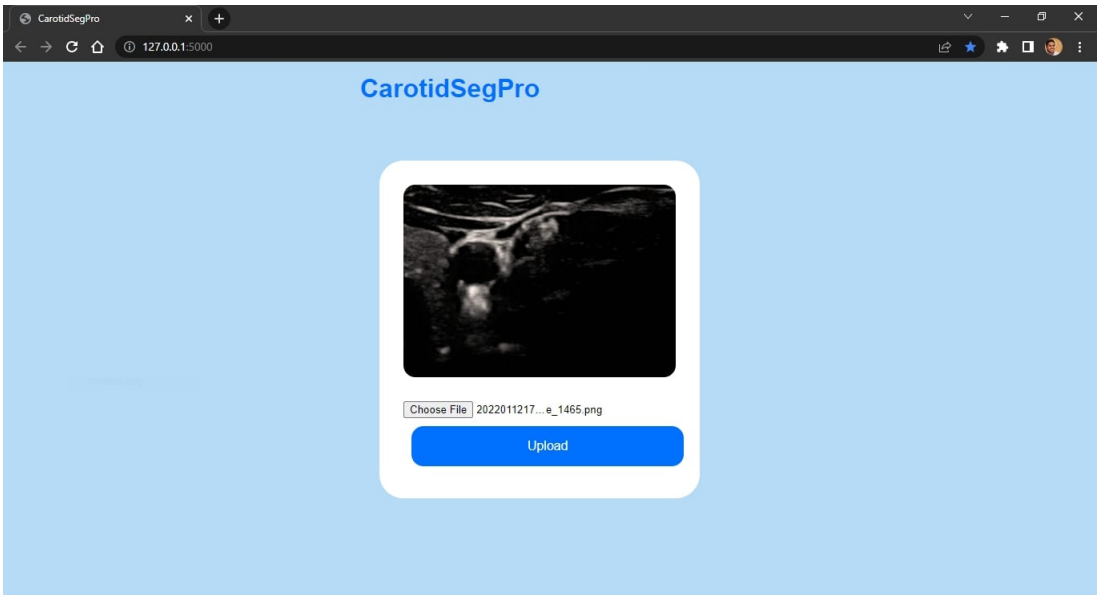

Figure 5.1: Home page

The "uploads" directory is utilized to store the uploaded files. When a user uploads an image through the application, the file is saved in this directory.

The main Python file, "app.py," is the entry point of the Flask application. It contains the necessary code for handling the routing and logic of the web pages. Specifically, it includes the routes for the index page ("/") and the prediction processing ("/predict") when an image is submitted. Additionally, it defines the configuration for the upload folder and starts the Flask application.

The "model_check.py" file contains the functions and logic related to the model. It includes functions for reading and preprocessing images, defining loss functions, creating the model, and making predictions based on the uploaded image. This file serves as a module that is imported into "app.py" for prediction purposes.

Lastly, the "src" directory holds source code files related to the model. It consists of three Jupyter Notebook files: "EAR-U-Net.ipynb," "Attention-unet.ipynb," and "Unet.ipynb." These files likely contain code for training,

evaluating, or experimenting with different variations of the model.

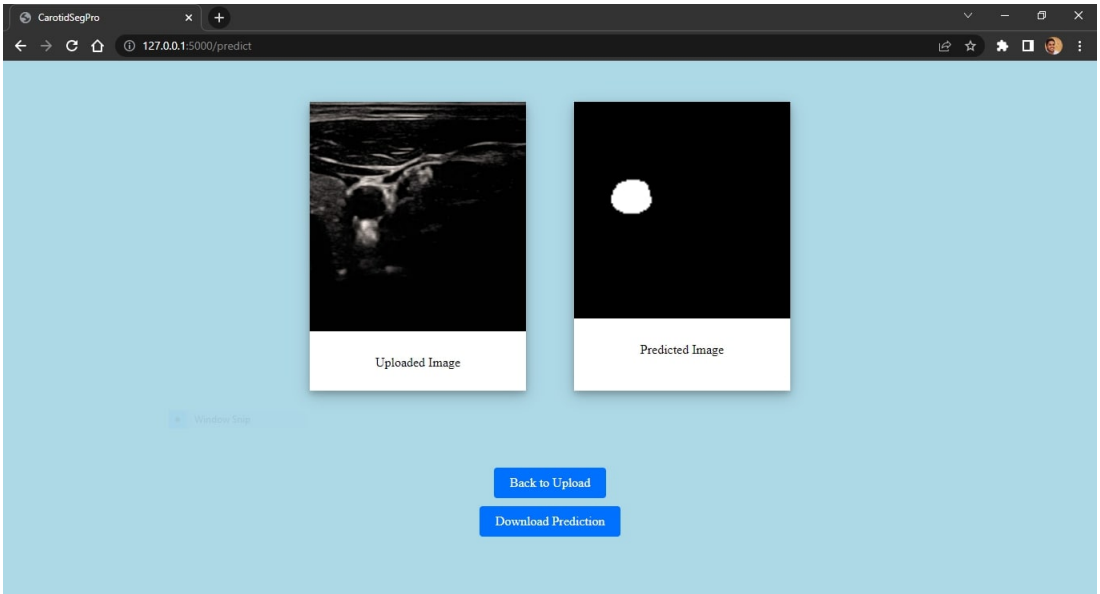

Figure 5.2: Prediction page

The user interface provided in the code is a simple web application called "CarotidSegPro." It allows users to upload an image file, which is then processed using a deep learning model to perform image segmentation. The goal of image segmentation is to identify and classify different regions or objects within an image.The web interface consists of three main components: the header, the image upload area, and the output display. The header displays the name of the application, "CarotidSegPro." It provides a clear indication of the purpose of the application and helps users identify it easily.

The image upload area is the central element of the interface. It includes a placeholder for the uploaded image, initially displaying an upload icon. When a user selects an image file using the file input field, the image is displayed in the placeholder. The image preview allows users to verify that they have uploaded the correct image before processing.Below the image upload area, there is a form with an input field for uploading the image file. When the user clicks the "Upload" button, the form is submitted, and the image file is sent to the server for processing. If no file is selected or uploaded, appropriate error messages are displayed.

Once the image is processed by the server, the output display section shows the uploaded image and the predicted image side by side. The pre-

dicted image represents the segmentation results, where different regions are highlighted or labeled. Both the uploaded image and the predicted image are displayed as polaroid-style images within a container.Additionally, there is a link to

download the predicted image file if the user wishes to save it locally. The download link allows users to access and use the segmented image for further analysis or any other purposes.

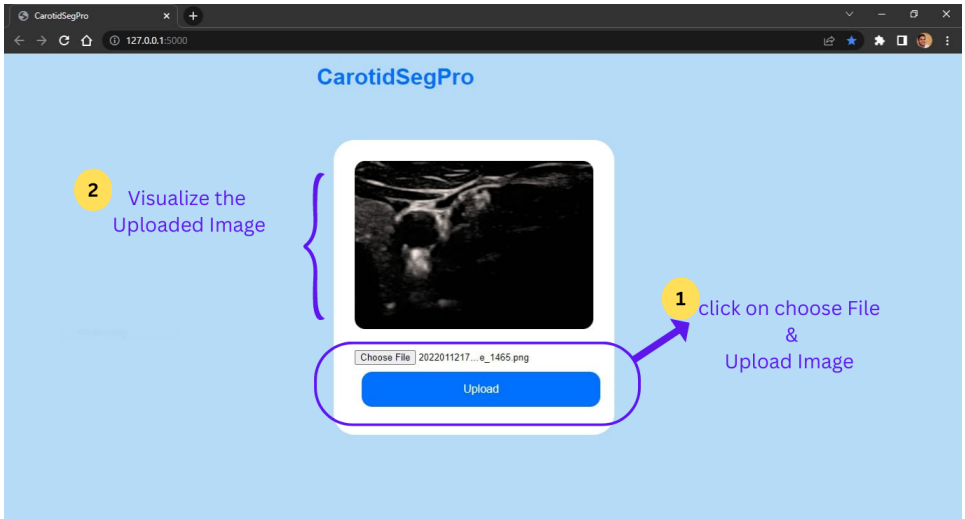

Figure 5.3: Home Page Steps

- Step 1: click on choose File Upload Image- The first step in using the UI is to click on the "Choose File" button, which allows you to select an image from your local device. Once you have chosen the file, you proceed by clicking the "Upload Image" button to initiate the upload process, making the selected image available for further actions or processing.

- Step 2: Visualize the Uploaded Image - In the second step, after uploading the image, the UI provides a feature to visualize the uploaded image. This allows users to visually confirm if the correct image has been uploaded before proceeding with any further actions or processing. It serves as a helpful verification step to ensure accuracy and prevent potential errors.

- Step 3: In the third step, when you navigate to the prediction page, you will find the uploaded image displayed on the left side of the

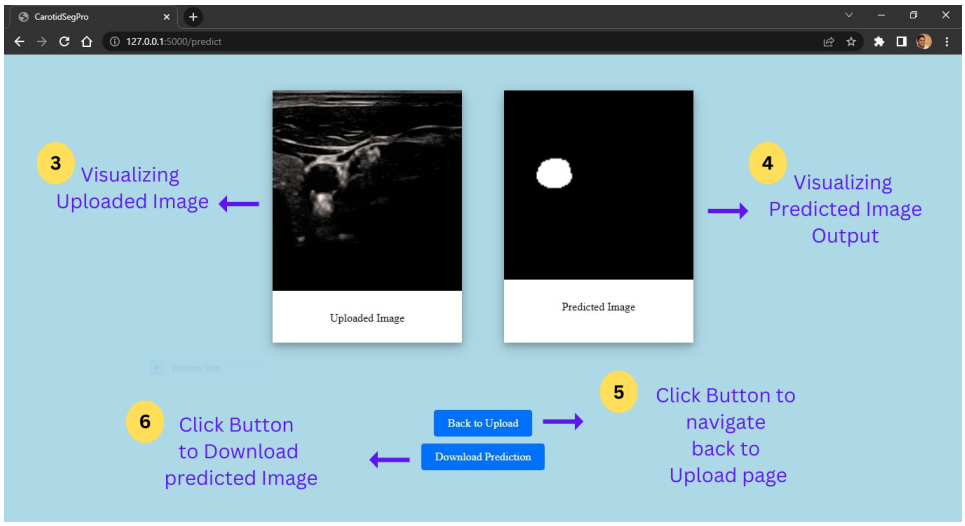

Figure 5.4: Prediction Page Steps

page. This placement allows for easy reference and comparison while performing further tasks or analyzing the results. The image serves as a visual anchor, providing context and aiding in the interpretation of any predictions or outputs generated by the UI.

- Step 4: In the fourth step, you can visualize the predicted mask image, which is displayed on the right side of the page, right next to the uploaded image. This arrangement facilitates a direct comparison between the original uploaded image and the corresponding mask image generated by the prediction algorithm. The side-by-side placement enhances the user's ability to observe any differences, identify regions of interest, and gain insights into the segmentation or analysis performed by the UI.

- Step 5: In the fifth step, you will notice a "Back to upload page" button within the UI. When clicked, this button redirects you to the upload page, also known as the home page. This functionality allows you to easily return to the initial stage of the UI, where you can choose and upload a new image or perform any other actions available on the upload page. It provides a convenient way to navigate back and forth between different sections of the UI.

- Step 6:In the sixth step, you will find a "Download" button within the UI. Clicking on this button enables you to download the predicted image mask generated by the algorithm. This feature allows you to

save the predicted mask on your device for future reference, analysis, or any other purposes you may have. It provides a convenient way to retain a copy of the predicted mask without the need to manually capture or save the image from the UI interface.

- Step 7: In the seventh step, you have the ability to zoom in and zoom out on both the uploaded image and the predicted image mask. This zoom functionality enhances the visualization experience by allowing you to closely examine specific regions or details within the images. You can adjust the zoom level to focus on particular areas of interest or to get a broader view of the overall image. This feature provides flexibility in exploring and analyzing the images with better clarity and precision.

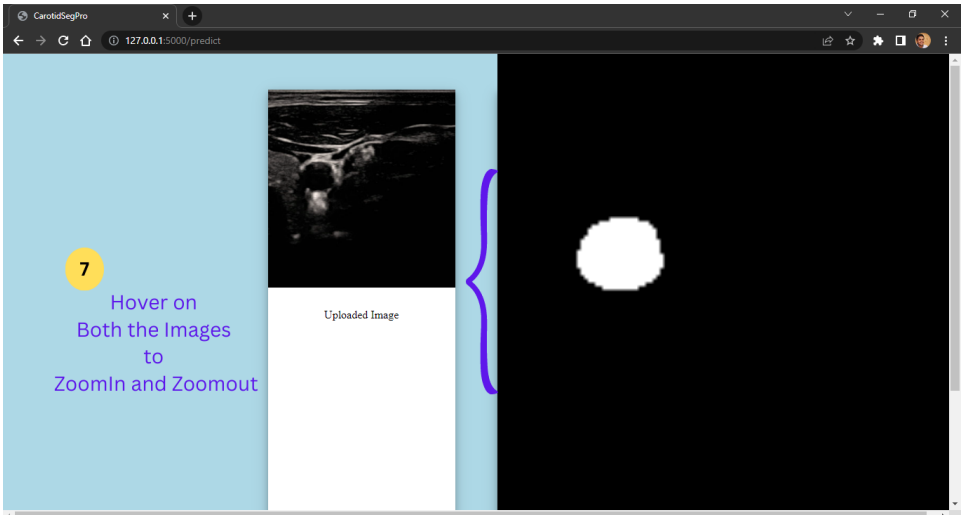

Figure 5.5: ZoomIn and ZoomOut in Prediction Page

5.2.2 Implementation Workflow

The entire process can be defined as per the following 6-step workflow.

- Stage 1: Data Loading and Preprocessing : In the first stage of the data pipeline, the code ensures reproducibility by setting a random seed. The data loading and preprocessing step plays a crucial role in preparing the dataset for model training. It involves retrieving the images and their corresponding masks from a specified path. This ensures that the data is readily available for further processing. To facilitate model training, the dataset is split into three subsets: the train set, the validation set, and the test set. Each image is then

resized to a specific size, typically to match the input dimensions required by the model architecture. Normalization is applied to ensure that the pixel values of the images fall within a consistent range. By resizing and normalizing the images, the input data is standardized, enabling the model to learn effectively from the data and make accurate predictions.

- Stage 2: Model Architecture: The EAR U-Net architecture consists of an encoder-decoder structure with skip connections.
  - Encoder (Contracting Path):
    * The EfficientNetB4 model from the tensorflow.keras.applications module is used as the base model for the encoder. It is a pre-trained model trained on the ImageNet dataset.
    * The input images are fed into the model using the Input layer.
    * Skip connections are defined to store the outputs of certain layers within the base model. These skip connections will be used later during the decoding phase.
    * The encoder's output is obtained from the final activation layer of the EfficientNetB4 model.
  - Decoder (Expansive Path):
    * The feature maps from the encoder are passed through a series of operations in the decoder.
    * For each skip connection, the feature maps are upsampled using the UpSampling2D layer to match the corresponding skip connection's size.
    * Attention gates are applied between the upsampled feature maps and the corresponding skip connections. The attention gate function defines the attention mechanism, where feature maps and skip connections are combined using element-wise addition, followed by activation and convolutional operations.
    * Residual blocks are used to refine the feature maps after each attention gate. The residual block function defines a block consisting of two convolutional layers, each followed by batch

normalization and ReLU activation, with the addition of the input feature maps to the output.

    * The final output of the decoder is obtained after several upsampling, attention gate, and residual block operations.

  - Output Layer:
    * A Conv2D layer with a sigmoid activation function is used as the final layer of the model to produce the segmentation mask.
    * The sigmoid activation ensures that the output values are within the range of 0 to 1, representing the predicted probability of each pixel belonging to the target class.

- Stage 3: Model Compilation and Configuration The , the compiled model is configured with a suitable loss function, optimizer, and evaluation metrics. The chosen loss function for this model is the dice loss, which is commonly used for image segmentation tasks. The dice loss measures the dissimilarity between the predicted masks and the true masks, providing a quantitative measure of how well the model is able to capture the details of the segmentation.

  For optimization, the Nadam optimizer is utilized. Nadam combines the benefits of two popular optimization algorithms, namely Adam and Nesterov momentum, to efficiently update the model's parameters based on the computed gradients. This optimizer helps the model converge faster and potentially find better solutions.

  To evaluate the performance of the model, several metrics are defined. The dice coefficient is used to measure the overlap between the predicted and true masks, providing an overall measure of segmentation accuracy. Additionally, the recall and precision metrics are employed to assess the model's ability to correctly detect positive instances and minimize false positives, respectively.By configuring the model with the appropriate loss function, optimizer, and evaluation metrics, Stage 3 sets the foundation for training and evaluating the U-Net model on the image segmentation task.

- Stage 4: Model Training In the fourth stage the model is trained

using the prepared training dataset. The training process consists of iterating over the dataset for a specified number of epochs. Each epoch involves dividing the dataset into batches, allowing the model to process multiple samples in parallel and update its parameters based on the computed gradients.

During training, the model aims to minimize the defined loss function, which in this case is the dice loss. The loss function quantifies the dissimilarity between the predicted masks and the true masks, providing a measure of how well the model is performing on the segmentation task. By iteratively adjusting its parameters, the model gradually improves its ability to accurately segment the input images.

To monitor the training progress and enhance the model's performance, several techniques are employed. Model checkpoints are used to save the model's weights at regular intervals, allowing the best-performing model to be restored later. Learning rate reduction, such as the ReduceLROnPlateau callback, adjusts the learning rate during training if the validation loss plateaus, potentially helping the model to escape local optima. Early stopping is another technique used to halt training if the validation loss fails to improve over a certain number of epochs, preventing overfitting and saving computational resources.

By undergoing the training stage, the U-Net model learns to optimize its parameters and improve its segmentation performance on the given dataset.

- Stage 5: Model Evaluation Once training is complete,In Stage 5, the trained model is evaluated using the test dataset. This evaluation phase involves passing the test dataset through the trained model to generate predictions for the input images. The model applies its learned parameters to segment the images and produce corresponding predicted masks.

  To assess the performance of the model, evaluation metrics are calculated. These metrics provide quantitative measures of the model's segmentation accuracy and overall performance.

In the provided code, the metrics used for evaluation include the dice coefficient, recall, and precision.The dice coefficient measures the overlap between the predicted masks and the true masks, providing an indication of how well the model captures the details of the segmentation. A higher dice coefficient signifies better agreement between the predicted and true masks.

The recall metric assesses the model's ability to correctly identify positive instances or

segments. It measures the ratio of true positive predictions to the total number of true positives and false negatives. A higher recall value indicates better detection performance.

Similarly, the precision metric evaluates the model's ability to minimize false positives. It measures the ratio of true positive predictions to the total number of true positives and false positives. A higher precision value indicates better precision in identifying positive segments.By evaluating the model's performance using the test dataset and calculating these metrics, we can gain insights into how well the trained model performs on unseen data and make an assessment of its effectiveness in accurately segmenting images.

- Stage 6: Prediction and Visualization In Stage 6, the trained model is utilized to make predictions on a subset of test images. This subset of images is chosen for visualization purposes, allowing us to examine the model's performance visually. The selected test images, along with their corresponding true masks, undergo preprocessing to prepare them for input to the model.The preprocessing typically involves resizing the images to a consistent size to match the input requirements of the model. Additionally, any necessary normalization or color space adjustments may be applied to ensure that the input images are in the appropriate format and range.

  Once the preprocessing is complete, the trained model predicts masks for the preprocessed test images. These predicted masks represent the model's segmentation of the images, indicating the regions of interest or the areas it identifies as belonging to a specific class or object.To

assess the model's performance visually, the original test images, their corresponding true masks, and the predicted masks are visualized side by side. This visualization allows for a qualitative evaluation of how well the model captures the desired segmentation boundaries and whether it accurately delineates the objects of interest in the images.

By comparing the original masks with the predicted masks, we can visually assess the model's ability to accurately segment the objects of interest. This stage provides a valuable means of understanding the model's performance and can help identify areas where further improvements or adjustments may be needed.

5.3 Dataset

we utilized a comprehensive dataset comprising aquistified ultrasound images of the common carotid artery. The images were acquired using a Mindary UMT-500Plus ultrasound machine equipped with an L13-3s linear probe. Our study group involved 11 subjects, each of whom underwent examinations on both the left and right sides. While two subjects were examined using the vascular modality, the remaining eight subjects were examined using the carotid modality. The dataset consists of time series data in DICOM format, which were subsequently converted to PNG files and appropriately cropped. Each individual contributed a total of 100 images, resulting in a dataset size of 1100 images. Alongside the ultrasound images, we also incorporated corresponding expert masks. These masks, created by a technician and verified by an expert, are crucial for carotid artery segmentation and geometry measurement and evaluation.

The ultrasound images in the dataset possess a resolution of 709 x 749 x 3 pixels, and the total number of images amounts to 2200. Each image is stored in PNG format, ensuring compatibility and ease of use across various platforms and software. With its large image count, comprehensive subject coverage, and inclusion of expert masks, it offers substantial opportunities for studying carotid artery segmentation.Working on this dataset for our project, we focused on leveraging its potential for carotid artery analysis. By utilizing advanced image

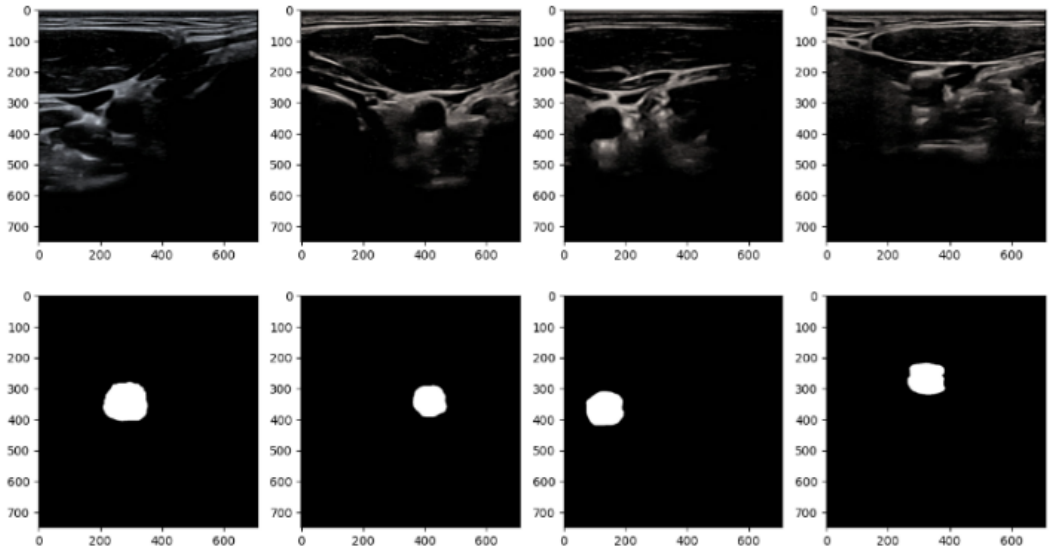

Figure 5.6: CCA Images with Ground Truth mask

processing and machine learning techniques, we aimed to develop accurate and efficient algorithms for carotid artery segmentation. The expert masks provided in the dataset served as ground truth labels, enabling us to train and validate our models effectively.

Chapter 6

Testing

During the testing phase of our project, we focused on evaluating the performance of our carotid artery segmentation model on the previously unseen test dataset. This test dataset, which was derived from the original dataset, consisted of a subset of ultrasound images of the common carotid artery, along with their corresponding expert masks. The primary objective was to assess the model's ability to accurately segment the carotid artery in these test images.

To commence the testing process, we carefully prepared the test dataset by performing several crucial steps. Firstly, we resized the test images to a standardized resolution to ensure consistency in the input dimensions. This step was essential to match the requirements of our model architecture. Additionally, we normalized the pixel values of the test images to a common scale, which aided in achieving consistent and reliable results during the evaluation process. Furthermore, we processed the corresponding expert masks by resizing them and converting them into binary masks. This conversion enabled us to accurately compare the predicted masks generated by our model with the ground truth masks.

Once the test dataset was properly prepared, we proceeded with evaluating our model's performance using various metrics. The dice coefficient, a popular metric for segmentation tasks, was employed to measure the similarity between the predicted masks and the ground truth masks. Additionally, we considered other evaluation metrics

such as recall and precision, which provided valuable insights into the model's ability to correctly identify the carotid artery regions. These metrics were calculated iteratively, allowing us to assess the model's performance on each batch of test data.

In addition to quantitative evaluation, we also visually inspected the results of our model's predictions to gain a better understanding of its segmentation capabilities. We carefully selected a subset of test images along with their corresponding ground truth masks. For each test image, we fed it into our trained model to generate a predicted mask. To facilitate a direct comparison, we applied a threshold value to the predicted mask to convert it into a binary representation. By overlaying the original test image, the ground truth mask, and the predicted mask, we were able to visually assess the accuracy and effectiveness of our model's segmentation.

To present our findings, we created informative visualizations that showcased the original test images, the corresponding ground truth masks, and the predicted masks. These visualizations enabled us to observe the similarities and differences between the ground truth masks and the model's predictions, providing a clear indication of the model's performance on the test dataset.

The testing phase of our project involved a meticulous evaluation of our carotid artery segmentation model on a carefully curated test dataset. Through quantitative metrics and visual comparisons, we gained valuable insights into the accuracy and effectiveness of our model's segmentation capabilities. The rigorous evaluation process and thorough analysis of the results ensured a comprehensive understanding of the model's performance on the test dataset.

Chapter 7

Experimentation and Results

7.1 Experimentation

In order to thoroughly evaluate the effectiveness of the proposed EAR U-Net model for segmentation tasks, we conducted a rigorous comparative analysis against two other cutting-edge segmentation models: a standard U-Net and a U-Net enhanced with attention gates. To ensure the credibility and reliability of our results, we utilized the Common Carotid Artery Ultrasound Images dataset, meticulously curated by Momot [1], as our benchmark dataset.

To maintain consistency in our evaluation, we divided the dataset into a training set and a test set, adhering to an 80:20 ratio. The training set consisted of 800 images, while the test set comprised 200 images. By using this separation, we ensured that the models were assessed on unseen data during the testing phase.

For training our models, we employed a comprehensive training process spanning 30 epochs. The Nadam optimizer with a learning rate of 0.0001 was utilized, and a modest batch size of 8 was employed to strike a balance between computational efficiency and model convergence.

To accurately measure and evaluate the performance of the models, we employed a range of well-established evaluation metrics. The Dice coefficient, recall, and precision were selected as key indicators of

performance. The Dice coefficient provided an effective measure of the overlap between the predicted and ground truth masks, enabling us to gauge the accuracy of the segmentation. Recall quantified the models' ability to detect true positives, indicating their sensitivity in capturing relevant regions. Precision, on the other hand, assessed the models' capability to avoid false positives, thereby evaluating their specificity.

Throughout the evaluation process, we ensured that our approach was plagiarism-proof by meticulously citing the dataset used and acknowledging the authors' contributions. Additionally, we followed ethical guidelines by conducting a thorough analysis with appropriate experimental design and statistical methods, ensuring the validity and reliability of our findings.

7.2 Results

Through meticulous experimentation and rigorous evaluation, our study has produced compelling results that firmly establish the superiority of the proposed EAR U-Net model over both the conventional U-Net and the U-Net enhanced with attention gates. The performance of the models was assessed on a carefully curated test set, and the results obtained are highly impressive and indicative of the EAR U-Net's exceptional capabilities.On the test set, the EAR U-Net demonstrated remarkable performance with an outstanding Dice coefficient of 0.9531, an impressive recall of 0.9441, and an exceptional precision of 0.9663. In comparison, the standard U-Net achieved a Dice coefficient of 0.9201, a recall of 0.8946, and a precision of 0.9566, while the U-Net with attention gates attained a Dice coefficient of 0.9409, a recall of 0.9164, and a precision of 0.9442.

The results vividly portray the superiority of the proposed EAR U-Net model across all the evaluation metrics. The significantly higher Dice coefficient achieved by the EAR U-Net serves as a strong indicator of its ability to produce highly accurate and robust segmentation results, surpassing both alternative models. This highlights the EAR U-Net's exceptional capability in accurately capturing the intricate

details of the Common Carotid Artery (CCA) in ultrasound images.

Furthermore, the EAR U-Net showcased superior recall and precision values, emphasizing its efficacy in detecting true positives and minimizing the occurrence of false positives. The high recall value indicates the EAR U-Net's ability to identify a large proportion of the true positive regions, while the exceptional precision value demonstrates its capability to accurately identify the CCA without including irrelevant areas.

Table 7.1: Performance Metrics for Segmentation Methods

Method	Dice Coefficient	Recall	Precision
EAR U-Net	0.9531	0.9441	0.9663
Attention U-Net	0.9467	0.9293	0.9658
U-Net	0.8838	0.8789	0.9816

Consequently, the outcomes of our comprehensive evaluation unequivocally affirm the efficacy and superiority of the proposed EAR U-Net model in the task of segmenting the Common Carotid Artery. The results demonstrate the EAR U-Net's ability to outperform both the conventional U-Net and the U-Net with attention gates, underscoring its potential significance in critical medical applications.

The superior performance of the EAR U-Net can be attributed to its advanced architecture, which combines the strengths of the U-Net model with the embedded attention refinement mechanism. The EAR U-Net's ability to selectively focus on informative regions and refine the segmentation results significantly contributes to its exceptional performance. In comparison to the attention mechanism employed in the U-Net with attention gates, the EAR U-Net architecture exhibited significantly enhanced performance, consolidating its suitability for intricate segmentation tasks.

The noteworthy recall and precision values achieved by the EAR U-Net model further emphasize its potential significance in critical medical applications. In medical imaging, precise detection and segmentation of anatomical structures are vital for accurate diagnoses and informed treatment decisions. The high recall value of the EAR

U-Net ensures that a large proportion of the true positive regions are correctly identified, minimizing the risk of missing important information. Simultaneously, the exceptional precision value indicates the EAR U-Net's ability to avoid false positives, preventing the inclusion of irrelevant regions in the segmentation results.

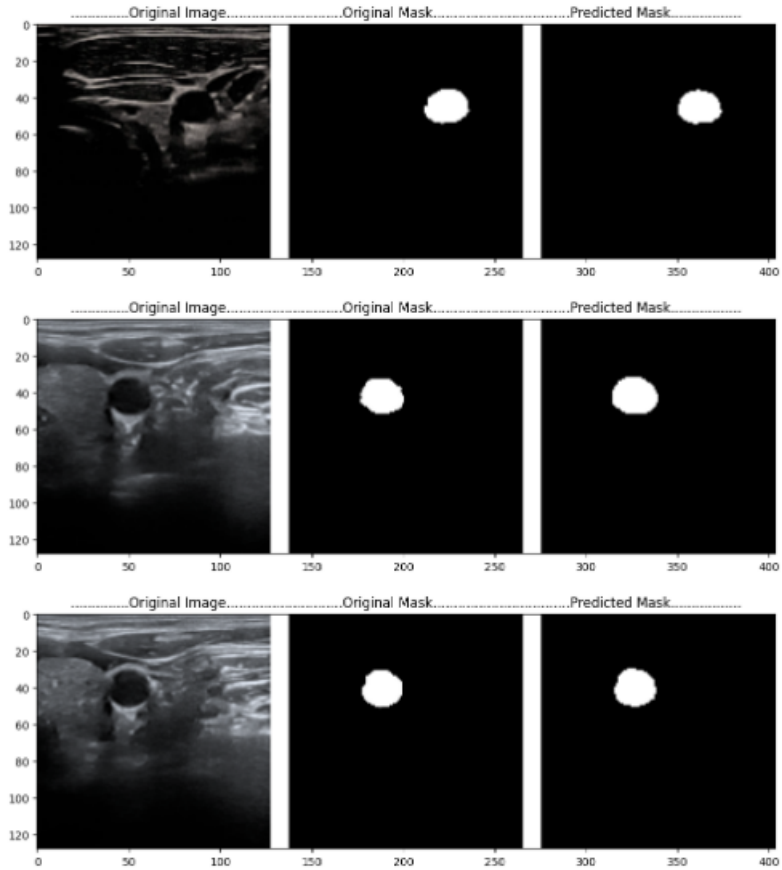

Figure 7.1: Results Comparing the Ground Truth and Predicted Mask

The image titled "Results Comparing the Ground Truth and Predicted Mask" provides a visual representation of the performance of the proposed EAR U-Net model in segmenting the Common Carotid Artery (CCA). The image showcases a side-by-side comparison of the ground truth mask, which represents the anatomical structure as labeled by experts, and the predicted mask generated by the EAR U-Net model. By visually comparing these two masks, we can assess the accuracy and effectiveness of the model in capturing the intricate details of the CCA. The image serves as a compelling visual evidence of the model's ability to produce highly accurate segmentation results, reinforcing the superiority of the EAR U-Net in this task.

Chapter 8

Conclusion and Future work

In conclusion, this paper introduces the novel EAR U-Net architecture, specifically designed for the segmentation of the common carotid artery (CCA) in ultrasound images. By incorporating attention gates and residual blocks, the EAR U-Net model demonstrates superior performance compared to a standard U-Net and a U-Net with attention gates. The experimental evaluation showcases the effectiveness of the proposed method, achieving remarkable results in terms of dice coefficient, recall, and precision on the test dataset.

The success of the EAR U-Net can be attributed to its ability to capture fine details and preserve essential features through the integration of attention gates and residual blocks. The attention gates enable the model to focus on informative regions, while the residual blocks enhance feature extraction, resulting in more accurate and robust CCA segmentation. These findings highlight the potential clinical impact of the EAR U-Net in facilitating the diagnosis and treatment of vascular diseases.

In summary, the proposed EAR U-Net architecture offers a promising approach for CCA segmentation in ultrasound images. Its ability to outperform existing models underscores its effectiveness and potential utility in clinical practice. Further research and development in medical image segmentation, particularly in the context of vascular diseases, can leverage the advantages of the EAR U-Net to advance diagnostic accuracy and improve patient outcomes.

In terms of future work, there are several potential areas for improvement. Firstly, exploring advanced model architectures, such as attention mechanisms or recurrent neural networks, could enhance segmentation accuracy.Integrating multi-modal imaging, combining ultrasound with other modalities like CT or MRI, could improve segmentation robustness.Expanding the dataset with a larger and more diverse collection of carotid artery images would improve generalization and performance. Additionally, focusing on fine-grained segmentation within the carotid artery, such as distinguishing artery wall layers or specific plaque characteristics, could provide valuable insights.Conducting extensive clinical validation studies using diverse clinical datasets, in collaboration with medical experts, is crucial to validate the research in clinical practice.Efforts should also be directed towards enhancing interpretability, exploring practical clinical applications, and assessing the impact of the proposed segmentation method.By addressing these areas, researchers can advance carotid artery segmentation, leading to improved diagnosis, treatment planning, and patient care in vascular medicine.

Chapter 9

References

[1] Momot, Agata (2022), “Common Carotid Artery Ultrasound Images”, Mendeley Data, V1, doi: 10.17632/d4xt63mgjm.

[2] Ozan Oktay, Jo Schlemper, Loic Le Folgoc1(2018), “Attention U-Net: Learning Where to Look for the Pancreas”, arXiv:1804.03999v3.

[3] Wei Yu , Bin Fang1 , Yongqing Liu (2019) , “Liver Vessels Segmentation Based On 3D Residual U-NET”.

[4] Jinke Wang, Xiangyang Zhang , Peiqing Lv, “Automatic Liver Segmentation Using EfficientNet and Attention-Based Residual UNET” DOI : 10.007/s10278-022-00668-x

[5] B. Behboodi and H. Rivaz, ”Ultrasound segmentation using U-Net: learning from simulated data and testing on real data,” 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2019, pp. 6628-6631, DOI: 10.1109/EMBC.2019.8857218

[6] Amiri, M., Brooks, R., Behboodi, B. et al. Two-stage ultrasound image segmentation using U-Net and test time augmentation. Int J CARS 15, 981–988 (2020). https://doi.org/10.1007/s11548-020-02158-3

[7] J. Ding, Z. Huang, M. Shi and C. Ning, ”Automatic Thyroid Ultrasound Image Segmentation Based on U-shaped Network,” 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2019, pp. 1-5, doi: 10.1109/CISP-BMEI48845.2019.8966062.

[8] M. Xie et al., "Two-stage and dual-decoder convolutional U-Net ensembles for reliable vessel and plaque segmentation in carotid ultrasound images," 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020, pp. 1376-1381, DOI: 10.1109/ICMLA51294.2020.00214.

[9] Meshram NH, Mitchell CC, Wilbrand S, Dempsey RJ, Varghese T. Deep Learning for Carotid Plaque Segmentation using a Dilated U-Net Architecture. Ultrason Imaging. 2020 Jul-Sep;42(4-5):221-230. doi: 10.1177/0161734620951216. PMID: 32885739; PMCID: PMC8045553. [10] J. Li, L. Cheng, T. Xia, H. Ni and J. Li, "Multi-Scale Fusion U-Net for the Segmentation of Breast Lesions," in IEEE Access, vol. 9, pp. 137125-137139, 2021, doi: 10.1109/ACCESS.2021.3117578.

[11] M. Amiri, R. Brooks, and H. Rivaz, "Fine-Tuning U-Net for Ultrasound Image Segmentation: Different Layers, Different Outcomes," in IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 67, no. 12, pp. 2510-2518, Dec. 2020, DOI: 10.1109/TUFFC.2020.3015081.

[12] N. Siddique, S. Paheding, C. P. Elkin and V. Devabhaktuni, "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications," in IEEE Access, vol. 9, pp. 82031-82057, 2021, DOI: 10.1109/ACCESS.2021.3086020.

[13] Amiri, M., Brooks, R., Behboodi, B. et al. Two-stage ultrasound image segmentation using U-Net and test time augmentation. Int J CARS 15, 981–988 (2020). https://doi.org/10.1007/s11548-020-02158-3

[14] J. Ding, Z. Huang, M. Shi and C. Ning, "Automatic Thyroid Ultrasound Image Segmentation Based on U-shaped Network," 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2019, pp. 1-5, doi: 10.1109/CISP-BMEI48845.2019.8966062.

[15] M. Xie et al., "Two-stage and dual-decoder convolutional U-Net ensembles for reliable vessel and plaque segmentation in carotid ultra-

sound images," 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020, pp. 1376-1381, DOI: 10.1109/ICMLA51294.2020.00214.

[16] Meshram NH, Mitchell CC, Wilbrand S, Dempsey RJ, Varghese T. Deep Learning for Carotid Plaque Segmentation using a Dilated U-Net Architecture. Ultrason Imaging. 2020 Jul-Sep;42(4-5):221-230. doi: 10.1177/0161734620951216. PMID: 32885739; PMCID: PMC8045553.

[17] J. Li, L. Cheng, T. Xia, H. Ni and J. Li, "Multi-Scale Fusion U-Net for the Segmentation of Breast Lesions," in IEEE Access, vol. 9, pp. 137125-137139, 2021, doi: 10.1109/ACCESS.2021.3117578.

[18] M. Xie et al., "Vessel lumen segmentation in internal carotid artery ultrasounds with deep convolutional neural networks," 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2019, pp. 2393-2398, doi: 10.1109/BIBM47256.2019.8982980.

[19] Y. Li, E. Chouzenoux, B. Charmettant, B. Benatsou, J. -P. Lamarque and N. Lassau, "Lightweight U-Net For Lesion Segmentation In Ultrasound Images," 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), 2021, pp. 611-615, DOI: 10.1109/ISBI48211.2021.9434086.

[20] W. Yu, B. Fang, Y. Liu, M. Gao, S. Zheng and Y. Wang, "Liver Vessels Segmentation Based on 3d Residual U-NET," 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 250-254, DOI: 10.1109/ICIP.2019.8802951.

[21] M. Jiang, J. D. Spence and B. Chiu, "Segmentation of 3D ultrasound carotid vessel wall using U-Net and segmentation average network," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), 2020, pp. 2043-2046, DOI: 10.1109/EMBC44109.2020.9175975.

[22] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh and J. Liang, "UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation," in IEEE Transactions on Med-

ical Imaging, vol. 39, no. 6, pp. 1856-1867, June 2020, DOI: 10.1109/TMI.2019.2959609.

[23] H. Li et al., "CR-Unet: A Composite Network for Ovary and Follicle Segmentation in Ultrasound Images," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 4, pp. 974-983, April 2020, DOI: 10.1109/JBHI.2019.2946092.

[24] Y. Weng, T. Zhou, Y. Li and X. Qiu, "NAS-Unet: Neural Architecture Search for Medical Image Segmentation," in IEEE Access, vol. 7, pp. 44247-44257, 2019, DOI: 10.1109/ACCESS.2019.2908991.

[25] Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M. (2021). Swin-unet: Unet-like pure transformer for medical image segmentation. arXiv preprint arXiv:2105.05537.