Learning Deep Discriminative Features for Retinal Vessel Segmentation

Xinxu Wei

Integrated Program in Neuroscience

McGill University, Montreal

April, 2023

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Master of Neuroscience

Abstract

Most existing deep learning based methods for vessel segmentation neglect two important aspects of retinal vessels: the orientation information of vessels and the contextual information of the whole fundus region. In this paper, we propose a robust orientation and context entangled network (OCE-Net), which can extract complex orientation and context information from blood vessels. To achieve complex orientation-aware convolution, we propose a dynamic complex orientation-aware convolution (DCOA Conv) to extract complex vessels with multiple orientations for improving vessel continuity. To simultaneously capture the global context information and emphasize the important local information, we propose a global and local fusion module (GLFM) to simultaneously model the long-range dependency of vessels and give sufficient attention to local thin vessels. A novel orientation and context entangled nonlocal (OCE-NL) module is also proposed to entangle the orientation and context information together. In addition, an unbalanced attention refining module (UARM) is proposed to deal with the unbalanced pixel numbers of the background and thick and thin vessels. Additionally, retinal vessel segmentation is a challenging task due to the need to capture global context information and ensure continuity of the vessels. To address these challenges, we also propose a novel Graph Capsule Convolution Network (GCC-UNet). This approach integrates capsule convolution with CNN to extract both local features and global context information, and we develop a Graph Capsule Convolution (GC Conv) operator to more effectively capture global context features. Furthermore, we design a Selective Graph Attention Fusion (SGAF) module to fuse local and global features. For improving the continuity of vessels,

we propose a Bottleneck Graph Attention (BGA) module, consisting of a Channel-wise Graph Attention (CGA) module and a Spatial Graph Attention (SGA) module. To fuse multi-scale features, we propose a Multi-Scale Graph Fusion (MSGF) module. Extensive experiments were performed on several commonly used datasets (DRIVE, STARE, and CHASEDB1) and some more challenging datasets (AV-WIDE, UoA-DR, RFMiD, and UK Biobank). The ablation study results show the proposed method's good performance in maintaining the continuity of thin vessels, and the comparative experimental results show the good performance of our OCE-Net and GCC-UNet in retinal vessel segmentation. Thus, the proposed frameworks can effectively carry out retinal vessel segmentation.

Abrégé

La plupart des méthodes existantes basées sur l'apprentissage en profondeur pour la segmentation des vaisseaux négligent deux aspects importants des vaisseaux rétiniens : l'orientation des vaisseaux et l'information contextuelle de l'ensemble de la région du fond d'œil. Dans cet article, nous proposons un réseau robuste entrelacé d'orientation et de contexte (OCE-Net), qui peut extraire des informations complexes sur l'orientation et le contexte des vaisseaux sanguins. Pour réaliser une convolution complexe prenant en compte l'orientation, nous proposons une convolution dynamique complexe prenant en compte l'orientation (DCOA Conv) pour extraire des vaisseaux complexes avec de multiples orientations afin d'améliorer la continuité des vaisseaux. Pour capturer simultanément l'information contextuelle globale et mettre l'accent sur l'information locale importante, nous proposons un module de fusion global et local (GLFM) pour modéliser simultanément la dépendance à long terme des vaisseaux et accorder une attention suffisante aux vaisseaux fins locaux. Un nouveau module non local entrelacé d'orientation et de contexte (OCE-NL) est également proposé pour entrelacer l'orientation et l'information contextuelle. De plus, un module de raffinement d'attention non équilibré (UARM) est proposé pour traiter les nombres de pixels déséquilibrés du fond et des vaisseaux épais et fins.

De plus, la segmentation des vaisseaux rétiniens est une tâche difficile en raison de la nécessité de capturer l'information contextuelle globale et d'assurer la continuité des vaisseaux. Pour relever ces défis, nous proposons également un nouveau réseau de convolution de capsule de graphe (GCC-UNet). Cette approche intègre la convolution de

capsule avec CNN pour extraire à la fois des caractéristiques locales et des informations contextuelles globales, et nous développons un opérateur de convolution de capsule de graphe (GC Conv) pour capturer plus efficacement les caractéristiques contextuelles globales. De plus, nous concevons un module de fusion d'attention sélective de graphe (SGAF) pour fusionner les caractéristiques locales et globales. Pour améliorer la continuité des vaisseaux, nous proposons un module d'attention de graphe d'étranglement (BGA), composé d'un module d'attention de graphe selon les canaux (CGA) et d'un module d'attention de graphe spatial (SGA). Pour fusionner les caractéristiques multi-échelles, nous proposons un module de fusion de graphe multi-échelle (MSGF).

Des expériences approfondies ont été menées sur plusieurs ensembles de données couramment utilisés (DRIVE, STARE et CHASEDB1) ainsi que sur certains ensembles de données plus difficiles (AV-WIDE, UoA-DR, RFMiD et UK Biobank). Les résultats de l'étude d'ablation montrent la bonne performance de la méthode proposée pour maintenir la continuité des vaisseaux fins, et les résultats expérimentaux comparatifs montrent la bonne performance de notre OCE-Net et GCC-UNet dans la segmentation des vaisseaux rétiniens.

Acknowledgements

In my master's study, I would like to sincerely thank my UESTC supervisor Prof. Yongjie Li and my McGill supervisor Prof. Danilo Bzdok for their guidance and enlightenment in my research journey. I am grateful for their support and guidance in my thesis. I also appreciate the help from my McGill mentor Prof. Edward Ruthazer and Vaibhav Sharma at McGill University, as well as Zilong Wang, Enning Yang, Liam Hodgson, Jakub Kopal, and others for their support and assistance. I am also grateful to my committee members Prof. Boris Bernhardt and Prof. Bratislav Misic.

I would like to express my gratitude to my dear friends, Xi Lin, Yanlin Huang, Haohan Bai, Shisen Wang, Hualong Han, Ninghao Chen, and others, as well as my labmates, Fuya Luo, Yijun Cao, Shixuan Zhao, Wangwang Yu, Peng Peng, Wei Huang, Yi Shi, Teng Qiu, Zhidan Li, Jingran Cui, and Zhenxiang Chen. Their company and support have given me sincere friendship and valuable qualities to grow and improve. I am also grateful to my parents for their attentive education and selfless guidance over the past 20 years. They rescued me from countless mistakes and difficulties and spent significant resources to raise me into a graduate student at a prestigious university. I would like to thank my elementary school teachers for laying a solid foundation for my growth, my middle school teachers for introducing me to the wonderful world of mathematics and foreign culture, and especially my high school teachers for their rigorous demands and valuable qualities of hard work, love of learning, patriotism, and discipline. They greatly enhanced my learning ability and instilled in me the pursuit of excellence.

Contribution of Authors

Xinxu Wei completed all the charpters in this thesis. The methods in this thesis are proposed by Xinxu Wei. The conceptualization, writing, and validation are finished by Xinxu Wei.

Prof. Yongjie Li and Prof. Danilo Bzdok are responsible for supervision of the retinal fundus project and review the manuscript of this thesis.

Table of Contents

Chapter 1

Introduction

Retinal vessel segmentation is helpful for ophthalmologists to diagnose eye-related diseases such as glaucoma, hypertension, diabetic retinopathy (DR), and arteriosclerosis [50] because changes in the vessel morphology and structure can be a symptom of a pathological condition. However, manual segmentation of retinal blood vessels is time-consuming and laborious because the structure of vessels is complicated. In addition, manual labeling is also error-prone because there are numerous capillaries throughout whole fundus images, which are narrow and have low local contrast against the fundus background. In addition, these thin vessels are easily mislabeled or missed during manual annotation. Therefore, automated fundus vessel segmentation [57] [53] [55] is meaningful and necessary for ophthalmologists to achieve a more accurate and rapid diagnosis of ophthalmic diseases.

However, vessel segmentation of fundus images is a challenging task. The blood vessels have complex geometric structures, and arteries and veins usually have different widths. In the whole vascular system, different vascular branches have different orientations, the capillaries are usually small, and it is difficult to seperate thin vessels from the fundus background, which is easy to be ignored. In addition, fundus images also contain various lesion areas, and the characteristics of some lesions are similar to those of blood vessels, so these vessel-like lesions are easily wrongly segmented as blood vessels.

To overcome these challenging problems, researchers have proposed many traditional vessel segmentation methods [8] [63] [48] and achieved good results. These traditional methods usually use some manually designed filters [8] [75] to extract vascular features, and some machine learning-based methods use classifiers [64] to classify each pixel of the fundus image.

In recent years, deep learning [74] [29] has been widely used in the area of medical image processing [15] [3] and has achieved great success. In particular, many efficient segmentation networks have been proposed [55] [51] [38], the most well-known one being UNet [71]. UNet is widely used in medical image segmentation, such as vascular segmentation [38], lesion area segmentation [43], and organ and tissue segmentation [20].

To improve the accuracy of vascular segmentation, most previously developed deep-learning-based fundus vessel segmentation methods attempt to increase the depth and width of the networks, and expand the receptive field by stacking numerous local convolution kernels. However, most of these methods pay little attention to two important information of fundus vessels: the orientation and context information, which are greatly important for accurate vessel segmentation.

As for the orientation information, unlike the instance segmentation in natural images, the blood vessels in fundus images have extremely complex orientation information owing to numerous furcations and branches. Capturing this complex orientation information is helpful to improve the accuracy of vessel segmentation and the continuity of thin vessels. In terms of the context information, in the whole vasculature, the overall skeleton of blood vessels presents certain distribution patterns, for example, the symmetry and relative position, orientation, and shape of each blood vessel branch relative to the whole vascular skeleton. In addition, some vessels are occluded by lesions, which makes them difficult to detect. Capturing global context information can allow the network to learn the distribution of blood vessels from a holistic perspective, which can alleviate the problem of occlusion. This is the motivation of the work of OCE-Net.

The main contributions of our OCE-Net are as follows.

- An orientation and context entangled network (OCE-Net) is proposed for retinal vessel segmentation by simultaneously capturing the orientation and context information of blood vessels;
- A dynamic complex orientation-aware convolution (DCOA Conv) is proposed to capture complex vessels with multiple orientations;
- A novel global and local fusion module (GLFM) is proposed to simultaneously model the global long-range dependencies and focus attention on local thin vessels.
- An orientation and context entangled nonlocal (OCE-DNL) block is proposed to entangle the orientation and context information by introducing correlation into a vanilla nonlocal operation.
- An unbalanced attention refining module (UARM) is designed to refine the output features and cope with the unbalanced problem among the background, thick, and thin vessels.
- The results of extensival experiments on multiple widely used datasets show our method's superior performance to other recent methods.

In addition, due to the intricate and intricate nature of the vascular system, pixel-level segmentation of vessels is prone to errors, especially in the presence of numerous thin vessels and capillaries. To overcome these challenges, there is a pressing need to develop automated computational methods that are both efficient and accurate in retinal vessel segmentation. Such methods would greatly improve the efficiency and effectiveness of retinal image analysis, which is critical for the diagnosis and management of various retinal diseases.

Developing accurate retinal vessel segmentation methods is a challenging task due to the high similarity between blood vessels and the background in retinal fundus scans [1]. Even experienced ophthalmological experts struggle to accurately annotate blood vessels from the background, and some lesions can be easily confused with vessels. However,

computer-aided methods [89] have made significant progress in this field, with practical algorithms and models that can assist doctors in making accurate diagnoses, developing effective treatments, and providing better medical care. Numerous computational methods have been proposed for retinal vessel segmentation [21] [99] [90], with promising performance. Traditional methods [31] [8] [107] [2] leverage pre-designed features and filters [75] to segment vessels from the complex fundus background, while machine learning-based models treat vessel segmentation as a pixel-level classification task [63], applying classifiers such as SVM [70], KNN, and Random Forest to each pixel. Recently, deep learning-based vessel segmentation methods [90] [18] have achieved state-of-the-art performance. However, deep learning still faces two major challenges: effectively capturing global context information and enhancing the continuity of blood vessels, especially capillaries. Many previous methods [85] [90] [94] have tried to capture global context information for medical image segmentation, as global information helps model long-range dependencies in holistic medical images. To improve vessel connectivity, previous methods propose effective loss functions [99] to constrain vessel connectivity and visual attention mechanisms [49] [90] to allocate more attention to thin vessels rather than the background and thick vessels.

To address these challenges, we propose a well-designed Graph Capsule Convolution Network (GCC-UNet) for retinal vessel segmentation. To our best knowledge, this is the first work to unify graph networks, capsule networks, and vanilla convolutional networks in a framework for medical image segmentation applications.

We highlight the contributions of the work GCC-UNet as follow:

- An novel Graph Capsule Convolution UNet (GCC-UNet) is proposed for retinal vessel segmentation by simultaneously capturing local features and global context features.
- A novel Graph Capsule Convolution (GC-Conv) is proposed to improve the existing capsule convolution by introducing graph reasoning into capsule.

- A novel Selective Graph Attention Fusion (SGAF) module is developed to fuse local and global context features.
- A Bottleneck Graph Attention (SGA) module, which consists of a Channel-wise Graph Attention (CGA) and a Spatial Graph Attention (SGA), is proposed to enhance the continuity of vessels.
- A Multi-Scale Graph Fusion (MSGF) module is designed to fuse multi-scale features.
- Extensive experiments on multiple widely used datasets show that our method outperforms many other recent methods and achieves state-of-the-art performance.

Chapter 2

Related Works

2.1 Deep Learning and Convolution Neural Networks

Deep learning is a subfield of machine learning that utilizes artificial neural networks with multiple layers to extract features and make predictions from data [46]. The key advantage of deep learning over traditional machine learning algorithms is its ability to automatically learn high-level features from raw data, without requiring manual feature engineering. This makes it particularly useful for tasks such as image and speech recognition, natural language processing, and computer vision.

Convolutional neural networks (CNNs) are a type of deep neural network specifically designed for image recognition tasks [36]. They have revolutionized the field of artificial intelligence (AI) and machine learning (ML) by automatically learning and extracting meaningful patterns and features from large amounts of data [29]. This has led to remarkable breakthroughs in various domains, including computer vision, natural language processing, and medical imaging [90].

One of the key advantages of CNNs is their ability to perform feature extraction automatically, without the need for manual feature engineering. They achieve this through the use of convolutional layers, which apply learnable filters to input images, capturing

specific visual patterns such as edges, corners, and textures. The resulting feature maps represent the presence of these patterns in the input image.

CNNs also incorporate non-linear activation functions, such as the Rectified Linear Unit (ReLU), which introduce non-linearity and enable the network to learn more complex representations of the input. Additionally, pooling layers are used to reduce the dimensionality of the feature maps, aiding in reducing sensitivity to small shifts and distortions in the input while also reducing computational complexity.

After several rounds of convolution, activation, and pooling layers, the output of the last layer is flattened into a vector and fed into a fully connected layer, which performs a classification or regression task. The fully connected layer consists of neurons that compute a weighted sum of inputs and pass the result through a non-linear activation function, such as the softmax function for classification tasks.

The history of CNNs dates back to the 1980s, with Fukushima's neocognitron introducing the concept of convolutional layers [23]. In the 1990s, LeNet architecture, developed by LeCun and colleagues, achieved success in character recognition tasks and introduced innovations such as backpropagation, max-pooling, and local contrast normalization [47].

In the 2000s, the field experienced a resurgence with the Deep Convolutional Neural Network (DCNN) developed by Hinton and colleagues, which introduced rectified linear units (ReLU), dropout regularization, and stacked convolutional layers [59]. The mid-2010s witnessed the popularity of CNNs due to their success in competitions like the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Influential works during this period include GoogLeNet, which introduced inception modules [80], and ResNet, which introduced residual connections [29].

Recent advancements include the MobileNet architecture, which reduces computational cost with depthwise separable convolutions [13], and the EfficientNet architecture, which achieves state-of-the-art performance with efficient scaling [**?**]. These developments have significantly improved the performance and efficiency of CNNs.

CNNs have achieved remarkable success in image recognition tasks such as object detection, segmentation, and classification [89, 90, 93]. They have surpassed human-level performance in some benchmarks due to their ability to automatically learn high-level features and capture increasingly complex representations of the input

2.2 Traditional Vessel Segmentation Methods

Many traditional methods have been proposed for vessel segmentation. Some researchers have attempted to leverage orientation-aware filters to capture the orientation features of vessels. For example, Zhang et al. [107] detected retinal blood vessels by a matched filter with the first-order derivative of Gaussian (FDOG), which leverages the orientation selectivity of a matched filter. Soares et al. [75] adopted a multi-scale two-dimensional (2D) Gabor wavelet transform to extract features and used a Bayesian classifier to classify each pixel as a vessel or non-vessel; this was the first work to introduce a Gabor filter into vessel detection. Azzopardi et al. [2] proposed trainable COSFIRE filters to achieve orientation selectivity. All these works' proposed approaches take the orientation into consideration; however, none of them can handle more than one orientation at a time. Inspired by these works, we propose a module that can capture multiple orientations of vessels.

Other traditional methods have been established; for example, Xie proposed HED [96], an effective edge detection algorithm that can be introduced to conduct vessel segmentation. Taking vessel segmentation as the line detection task, Ricci proposed LineDet [70], a vessel segmentation method that uses line operators and support vector machine (SVM) for classification. Nguyen proposed a method based on LineDet, MS-LineDet [60], which applies line detectors at varying scales and changes the length of the basic line operators. These line-detection-based methods have the advantage of improving vessel connectivity, but they always misdetect other tissues as blood vessels.

In general, these traditional methods leverage the intrinsic characteristics of vessels and achieve good performance. However, the features extracted by these traditional methods lack discriminability, so they always fail to distinguish capillaries from the fundus background, which results in failure in the detection of thin vessels.

2.3 Deep Learning Vessel Segmentation Methods

Deep learning methods for retinal vessel segmentation have generally outperformed traditional methods when successfully trained on large-scale datasets with manual labels. This has been demonstrated by many groundbreaking works. For example, Maninis proposed DRIU [55], which uses a basic network and two specialized layers to perform blood vessel and optic disc segmentation; this was a pioneering work introducing deep learning into retinal vessel segmentation. Fu proposed DeepVessel [21], which views vessel segmentation as a boundary detection task and uses the conditional random rield (CRF) to capture long-range interactions between pixels, making it the first method to take context information into consideration. Son proposed V-GAN [76], which introduces generative adversarial networks (GANs) into vessel segmentation to extract clear and sharp vessels with less false positives, making it the first method to introduce GAN into retinal vessel segmentation. Shin proposed VGN [73], which incorporates a graph convolutional network into a CNN to learn the graphical connectivity of vessels.

In addition, some researchers have aimed to improve the plain UNet by introducing some novel modules. For example, Oktay proposed Attention UNet [61], which introduces an attention gate into UNet for better feature learning and integration with channel attention; in contrast, Guo proposed SA-UNet [26], which introduces spatial attention into UNet. Jin designed DUNet [38], which introduces deformable convolution into UNet to adaptively fit the shape of vessels; however, the deformable convolution is computationally intensive. Inspired by DUNet, we propose a module (DCOA Conv) that can fit the orientation of vessels in this work.

Other works have been dedicated to segmenting thick and thin vessel separately. For example, Yan proposed JL-UNet [99], which features a segment-level loss to focus more on the thickness consistency of thin vessels. The three-stage model proposed by Yan et al. [100] segments thick and thin vessels separately for addressing the imbalance problem between them. Inspired by these methods, we propose a module (UARM) that can focus attention separately on thick and thin vessels.

Moreover, some researchers have aimed to capture multi-scale features of vessels. For example, to deal with the varying widths and directions of vessel structures, Oliveira proposed SWT-FCN [62], which acts as a multi-scale fully convolutional neural network by combining the multi-scale analysis and using the stationary wavelet transform. Guo designed BTS-DSN [28], a multi-scale deeply supervised network with short connections for vessel segmentation. Wang's CTF-Net [86] is a coarse-to-fine SegNet for preserving multi-scale feature information. CC-Net [18] is a cross-connected convolutional neural network designed to better learn features. Inspired by these works, we propose OCE-DNL, which can leverage the multi-scale features by fusing them together.

In addition, some scholars have tried to improve the connectivity of vessels. For example, DeepDyn, proposed by Khanal [40], is a stochastic training scheme for deep neural networks to balance precision and recall. To improve the continuity of thin vessels, Tan proposed SkelCon [81], a lightweight network that introduces the skeletal prior and contrastive loss during training.

Although the above works' proposed methods can greatly improve retinal vessel segmentation performance, they all ignore the multiple orientations of vessels and the context information of whole fundus images, which are important for maintaining the continuity and connectivity of vessels.

2.4 Visual Attention Mechanism

Visual attention is a cognitive mechanism that enables human beings to focus on a subset of relevant information in a given scene while ignoring irrelevant information. The study of visual attention [35] has attracted a lot of attention in computer vision [27] and artificial intelligence [82] [83] because it is an essential aspect of human visual perception that is crucial for tasks such as object recognition [93], scene understanding [108], and medical imaging [89]. Deep learning models have been used to model the visual attention mechanism and have shown promising results in various computer vision tasks [91] [27]. In this article, we will review some of the recent works related to deep learning and visual attention mechanisms.

The visual attention mechanism is a complex process that involves the selection of relevant visual features from a given scene while filtering out irrelevant features [27]. In the context of deep learning, visual attention mechanisms refer to the ability of neural networks to selectively focus on relevant parts of the input image while suppressing irrelevant parts. One of the most popular visual attention mechanisms is the spatial attention mechanism [93], which involves the use of convolutional neural networks (CNNs) to learn spatially varying attention masks that highlight relevant parts of the input image.

Deep learning attention mechanisms have been widely studied and applied in various fields, such as natural language processing [82], computer vision [27], and speech recognition. The attention mechanism allows the model to focus on important information, improving the performance of the model in tasks such as classification and generation. One popular attention mechanism is the self-attention mechanism [82] proposed in the Transformer model, which calculates the attention weights based on the relationships between all input tokens. This mechanism has been successfully applied in various natural language processing tasks, such as machine translation and text summarization. Another attention mechanism is the visual attention mechanism, which is commonly used in computer vision tasks. This mechanism allows the model to selectively focus on cer-

tain parts of the image, improving the accuracy of object recognition and low-level image tasks [92]. One example is the Spatial Transformer Network, which uses attention to transform the input image before feeding it into a convolutional neural network. There have also been works on incorporating attention mechanisms into generative models, such as the attention-based Generative Adversarial Networks (GANs) [25]. This approach allows the generator to focus on certain parts of the input, improving the quality of generated images.

In conclusion, deep learning models have shown promising results in modeling the visual attention mechanism and have been successfully applied to various computer vision tasks. Recent works [33] have focused on developing new models and algorithms that can improve the performance of these tasks further. The development of these models and algorithms is expected to have significant implications for the field of computer vision and artificial intelligence.

2.5 Non-local and Self-Attention Mechanism

Non-local (NL) [87] and Self-Attention (SA) [82] mechanisms have become important techniques in deep learning for modeling dependencies between input elements and selectively attending to important information. The non-local operation was first proposed by Wang et al. in their 2018 paper "Non-local Neural Networks" [87] for modeling long-range dependencies between features in computer vision tasks. The self-attention mechanism was first introduced in the Transformer model by Vaswani et al. in their 2017 paper "Attention is All You Need" [82] for modeling contextual dependencies between elements in natural language processing tasks.

Several works have combined non-local and self-attention mechanisms to improve the performance of deep learning models [69] [90]. For example, the Non-local Neural Networks paper introduced a non-local block that can be inserted into any convolutional neural network. The Dual Attention Network proposed by Fu et al. in their 2019 paper

"Dual Attention Network for Scene Segmentation" [22] consists of two attention modules: a spatial attention module that attends to different spatial locations in the input image, and a channel attention module that attends to different feature channels.

These attention mechanisms have led to significant improvements in various fields such as natural language processing, computer vision, and speech recognition. Future research will likely explore new ways of combining these mechanisms to further improve the accuracy of deep learning models.

Non-local (NL) [87] and Self-Attention (SA) [82] were proposed to model the long-range dependencies without the distance constraints of pixels. Both of them calculate the affinities between the key and query vectors obtained from the same features for capturing the self-correlated attention map. The nonlocal neural network [87] computes the response at a position by calculating the weighted sum of the features at all positions in images, inspired by NL's application in image denoising [5].

Some authors have attempted to improve the performance of NL. For example, Cao proposed GCNet [6], which finds the relationship of a nonlocal and SE block [32] and combines them to further improve NL. Yin's disentangled nonlocal (DNL) [104] uses a whitening operator to disentangle the vanilla NL into pairwise and unary branches to better learn the within-region clues and salient boundaries. Both of the above works provide insights into NL and proposed methods that improve NL. The SA mechanism [82] proposed in Transformer [82] can capture the global context information of sequence embeddings in natural language processing (NLP).

However, nonlocal is computationally expensive. To reduce the computational cost of NL, Huang proposed CCnet [33] to reduce the computation complexity of self-attention. In addition, to further extend the ability of SA, Bello et al.'s method [4] combines the vanilla convolution and self-attention to obtain better performance. Furthermore, Ramachandran et al. [69] found that self-attention could serve as an effective standalone layer, which was an interesting discovery. Fu proposed DAN [22], which introduces self-attention into scene segmentation for adaptively integrating local features with their

global dependencies, considering that local and global features are equally important. Their motivation aligned with ours.

In this paper, we adopt nonlocal and self-attention for capturing context information of fundus images and extend the capacity of nonlocal by introducing cross-correlation mechanism into the vanilla nonlocal.

2.6 Variants of Convolution Operator

Convolution is a basic operator in a CNN for extracting deep representative features [80] [74]. Based on vanilla convolutions, a number of efficient variants of the convolution operator have been designed to extend the representation ability of the vanilla convolution.

For instance, some researchers have aimed to equip plain convolution operators with the ability to learn the shape of objects. For example, Jeon proposed Active Convolution [37], which adaptively learns the shape of convolution during training. Dai's deformable convolution [14] [111] learns the offsets of the kernel shapes to fit the object shapes in ROI. Chen proposed dynamic region-aware convolution [9], which learns to apply each convolution kernel on a single patch region to handle the complex and variable spatial distribution.

Others have tried to endow the plain convolution operators with ability of learning the orientation information of objects. For example, Luan proposed Gabor convolution [54], which introduces Gabor filters with different orientations into vanilla convolution, but it can only learn a single orientation per channel and cannot capture multiple orientations.

In addition, some researchers have tried to endow plain convolution operators with the ability to integrating several kernels. For example, CondConv was first proposed to integrate several convolution kernels, and it [101] learns specialized convolutional kernels for each example by introducing conditional parameters. Dynamic Convolution was further improved by introducing an attention mechanism to learn the weights and the biases of each vanilla convolution and to integrate them into a single convolution kernel [10].

All of the above works' proposed convolution operators extend the capacities of vanilla convolution; however, most of them do not take the orientation of objects into consideration, and some of them [54] can only encode simple orientation information. In this paper, we propose a novel convolution that can capture complex orientations to fully extract the features of fundus vessels for improving the continuity of thin vessels.

2.7 Graph Neural Networks

Graph Neural Networks (GNNs) [110] are a class of deep learning models that operate on graphs or networks. Unlike traditional deep learning models that operate on structured data such as images and text, GNNs are designed to handle non-Euclidean data structures such as graphs and networks, which are common in many real-world applications such as social networks, molecular structures, and recommendation systems.

The basic idea behind GNNs is to learn representations of nodes and edges in a graph that capture the structure and relationships between them. These representations are then used to perform various tasks such as node classification, link prediction, and graph classification. The key challenge in designing GNNs is to develop models that can effectively capture the complex dependencies and interactions between nodes and edges in a graph.

There are several types of GNN architectures, including spectral-based methods and spatial-based methods. Spectral-based methods, such as Graph Convolutional Networks (GCNs) [42], operate in the Fourier domain and apply convolutional operations to graph signals. Spatial-based methods, such as Graph Attention Networks (GATs) [83], operate in the spatial domain and use attention mechanisms to selectively attend to different parts of the graph.

In GCNs, the graph convolution operation is defined as a linear combination of the node features of each node and its neighbors in the graph. The weights of the linear combination are learned during training, and the resulting feature vectors are passed through non-linear activation functions to generate node embeddings. The process is repeated

for multiple layers, with each layer learning increasingly complex representations of the nodes in the graph.

In GATs, attention mechanisms are used to weight the contributions of each neighbor node to the representation of a target node. The attention weights are computed using a learned function of the node features and are used to generate a weighted sum of the neighbor node features. The resulting representation is then passed through non-linear activation functions to generate node embeddings.

GNNs have been applied to a wide range of applications, including recommendation systems, drug discovery, social network analysis, and computer vision. They have been shown to outperform traditional methods in many tasks, especially when the data is represented as a graph or network.

Graph neural networks (GNNs) [110] [95], especially graph convolutional networks (GCNs) [109] [42], have become popular in various research fields such as computer vision [108] [103]. Kipf [42] first proposed GCNs for achieving the classification of non-Euclidean data, and since then, many applications have emerged in image recognition [103], segmentation [108], medical image analysis [58], visual attention [103], and text classification [102]. Although there have been numerous works [83] [105] [88] aimed at improving and extending the capabilities of GNNs, few studies have applied graph networks to vessel segmentation [73]. To address this gap, we propose the development of GCN modules and their introduction into retinal vessel segmentation to improve vessel continuity.

In conclusion, Graph Neural Networks (GNNs) are a powerful class of deep learning models that operate on graphs or networks. They are designed to handle non-Euclidean data structures and can effectively capture the complex dependencies and interactions between nodes and edges in a graph. GNNs have been applied to a wide range of applications and have shown to outperform traditional methods in many tasks. As the field of graph neural networks continues to evolve, it is likely that new architectures and methods will be developed to further improve the accuracy and efficiency of GNNs.

Our work aims to extend the application of GNNs to retinal vessel segmentation by explicitly developing GCN modules.

2.8 Capsule Neural Networks

The concept of capsules was first proposed by Hinton to address the intrinsic limitations of CNNs in capturing global contextual information due to limited kernel sizes and receptive fields, as well as their lack of equivalence ability [72]. Sabour [72] subsequently developed a capsule network that employs dynamic routing between capsule units to model part-whole relationships and enhance equivalence ability. Since then, various studies [12] [30] have aimed to improve the performance of capsule networks, such as integrating graph networks with capsule networks [97] [84], improving computational efficiency with DeformCaps [44], and enhancing the routing algorithm with EM-Routing [30], attentive routing [12], and self-attention routing [56]. Capsule networks have been applied in various fields, including object detection [44], biomedical image segmentation [45], and image classification [16].

Capsule Neural Network (CapsNet) is a novel neural network architecture inspired by the human visual system. Unlike traditional Convolutional Neural Networks (CNNs), CapsNet uses capsules as the basic building blocks of the network, which enhances the network's ability to capture object pose and spatial relationships between objects.

A capsule is a group of neurons that represents a particular entity, such as an object or a part of an object. Each capsule outputs a vector, which represents the probability of the entity's existence and its various properties, such as its pose, size, and orientation. These vectors are called activation vectors, and they are computed based on the input data and the weight matrix of the capsule.

CapsNet consists of multiple layers of capsules, with each layer having a specific role in the network's function [72]. The first layer of capsules is called the primary capsules, which extract basic features from the input data, such as edges and corners. The primary

capsules then pass their activation vectors to the next layer of capsules, called the routing capsules. The routing capsules use a dynamic routing algorithm to combine the activation vectors of the primary capsules into higher-level capsules, which represent more complex entities, such as objects or parts of objects. The dynamic routing algorithm ensures that the higher-level capsules receive input only from primary capsules that agree on their existence and pose, which helps to reduce the effects of occlusion and ambiguity in the input data.

CapsNet has several advantages over traditional CNNs. First, it can capture spatial relationships between objects more accurately than CNNs, which are limited by their local receptive fields. Second, it can handle occlusion and deformation of objects more robustly, because the activation vectors of the capsules are invariant to these changes. Third, it can generate more interpretable results, because the activation vectors of the capsules represent the properties of the entities they represent.

CapsNet has been applied to various tasks, such as image classification, object detection, and image generation, and has achieved state-of-the-art performance on several benchmarks. However, CapsNet also has some limitations. It requires more training data and computational resources than traditional CNNs, because it has more parameters and more complex computations. It is also less well-understood than CNNs, because it is a relatively new architecture with fewer theoretical analyses and empirical studies.

We firstly introduce capsule network into retinal vessel segmentation by leveraging its power of modelling global context,and improve the capability of capsule conv by introducing graph reasoning. LaLonde et al. [45] proposed a capsule UNet for object segmentation and biomedical image segmentation, while Hoogi et al. integrated self-attention mechanism with capsules for object classification. Nguyen developed ResCap for medical image segmentation by adopting residual connections among the capsules to preserve pose information. In conclusion, Capsule Neural Network is a promising neural network architecture that can enhance the network's ability to capture object pose and spatial relationships between objects. It has several advantages over traditional CNNs, such as better

spatial modeling, robustness to occlusion and deformation, and interpretability. However, it also has some limitations, such as higher data and computational requirements, and less theoretical understanding. Capsule networks have emerged as a powerful alternative to traditional CNNs for tasks requiring capturing global context information and preserving pose information. With the continued development of capsule networks and their applications, we can expect more breakthroughs in this field.

Chapter 3

Methdology

3.1 Methodology of the Proposed OCE-Net

3.1.1 Overall Architecture of OCE-Net

The network architecture of OCE-Net is shown in Fig. 3.7. The backbone of OCE-Net is the vanilla UNet [71], which is widely used in medical image segmentation. In the down-sampling stage, we use the proposed DCOA Conv to extract the multiple orientation features of vessels and employ the plain convolution to extract the plain features of the fundus images, and then a Selective Attention Fusion Module (SAFM) is utilized to fuse these two kinds of features (the reason is explained in Section. 4.2.5). A GLFM is proposed to play the role of an attention gate of UNet for simultaneously capturing the global and local features of vessels. In the up-sampling stage, the extracted orientation-aware features are used as the prior guidance for improving the continuity. At the end of the network, a multi-Scale fusion module (MSFM) is introduced with an OCE-DNL to entangle the orientation and context information together. Finally, a UARM is proposed to refine the output feature by focusing more attention on thin vessels.

3.1.2 Dynamic Complex Orientation-Aware Convolution

Orientation of tissues is an important feature for medical image segmentation [75] [11]. To capture the complex orientation information of the blood vessels, we propose DCOA Conv to extract the features of vessels with multiple orientations within the same receptive field. As shown in Fig. 3.1, in the proposed DCOA Conv, the oriented Gabor filters are generated from a pre-designed Gabor Filter Bank. The 2D Gabor function is defined in literature [106].

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = exp(-\frac{x'^2 + \lambda^2 y'^2}{2\sigma^2})exp(i(2\pi\frac{x'}{\lambda}) + \psi) \tag{3.1}$$

where $x' = xcos(\theta) + ysin(\theta)$ and $y' = -xsin(\theta) + ycos(\theta)$. x and y are the horizontal and vertical coordinates of pixels, respectively. λ indicates the wavelength of the Gabor filter, and it is set to $1/\sqrt{2}$. θ represents the orientation of a filter. ψ is the phase offset, which is set to 0. σ denotes the standard deviation, and it is set to 1. γ is the spatial aspect ratio and it is set to 1. As shown in Fig. 3.1, by setting different values for θ, we can obtain filters with different orientations can be obtained. There are filters with eight orientations, so the values of $theta$ are set to 2*π / i, (i=1,2,...7,8). Then, these kernels are multiplied with eight vanilla convolution kernels to assign the filter kernels with orientation preference to the vanilla convolution whose kernel has no orientation selectivity. A batch normalization layer [34] is used here for normalizing the weights of the Gabor kernels. Following the work in [10], we use an attention module to learn the weight coefficients of each oriented kernel for selecting the useful convolution operators, and the attention coefficients are computed following the design in [10]. Note that the selection depends on the orientation of the vessels in the receptive field. If there is no vessel along the orientation of the receptive field, the weight coefficient of this oriented convolution is set to 0 by the attention module, and it is not integrated into the final convolution kernel with multiple orientations.

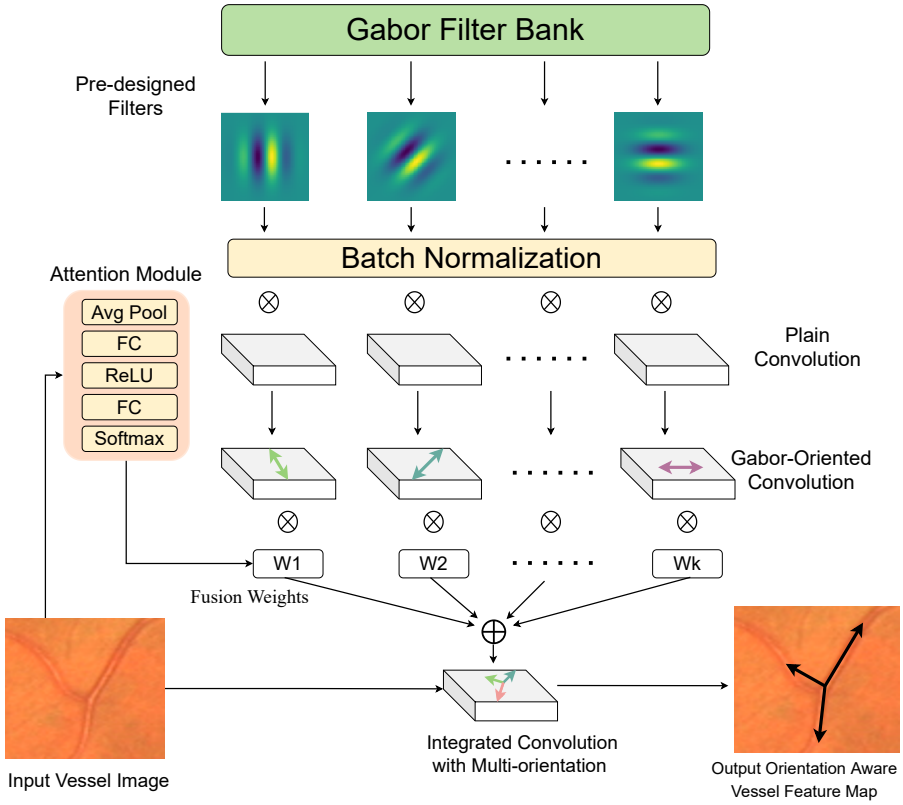

Figure 3.1: The proposed DCOA Conv.

Then, the selected convolution kernels are integrated together to form a single convolution operator, which has a kernel with multiple selective orientations. The DCOA Conv is written as

$$DCOA = \sum_{i=1}^{8} w_i(K_i \otimes BN(G(\theta_i))) \quad (i = 1, 2, 3, ...8) \tag{3.2}$$

where $G(\theta_i)$ represents the composite Gabor kernels. BN denotes the batch normalization layer. K_i is the plain convolution kernel without orientation. $\otimes$ is the multiplication operator. w_i denotes the weight coefficient learned by the attention module. Our experiments showed that choosing eight ($i = 8$) orientations could encode the orientations of all blood vessels well. This composite convolution kernel has the ability to capture the vessels with complex multiple orientations in a single receptive field. It is dynamic because the orientations of the final composite convolution kernel can be dynamically adjusted

by learning to set different weight coefficients for different oriented convolution kernels based on the orientations of specific vessels.

As shown in Fig. 3.7, a DCOA block now can be obtained by stacking DCOA Conv, batch normalization (BN), and the ReLU function together for extracting orientation features F_o:

$$F_o = ReLU(BN(DCOA(F_{in}))) \tag{3.3}$$

where F_{in} means the input features. From Fig. 3.7, the basic block can be obtained by directly replacing the DCOA Conv in DCOA block with the vanilla convolution to extract the plain features without an orientation preference. The kernel sizes of both vanilla convolution and DCOA Conv are set to 3x3 in their own blocks.

The DCOA Conv in DCOA block is used to capture multiple orientations of vessels; however, DCOA Conv only focuses on the vessels along one or several orientations and ignores other important features that are not located at these particular orientations, such as the fundus background and other nonvascular tissues, which are not oriented but also helpful for identifying blood vessels when serving as the important negative samples. Therefore, the best solution is to capture both the plain features and the orientation features extracted by the basic blocks and the DCOA blocks described above. In other words, the extracted orientation features should be viewed as the auxiliary prior, which is used to guide the segmentation of blood vessels.

As shown in Fig. 3.2, inspired by [52], we introduce an SAFM to fuse the plain and orientation features, which are extracted by the basic block and the proposed DCOA block. The channel attention mechanism is used to select useful channels of both kinds of features before fusion because there are some redundant channels in both the plain and orientation features of fundus images, which contain much noise and artifacts. The spatial attention [68] is used to focus more attention on the useful features in both branches because tissues like exudates, hemorrhages, and maculas have similar features with blood vessels, which may mislead the network to identify them as blood vessels.

The spatial attention SPA is defined as

$$F_{spa} = SPA(F_{in}) = F_{in} * \delta(Conv(ReLU(Conv(F_{in})))) \tag{3.4}$$

where F_{in} and F_{spa} are the input and output features of spatial attention, respectively. $Conv$ means the convolution operator has a kernel of 3x3 size. $ReLU$ means the ReLU function. δ denotes the sigmoid function.

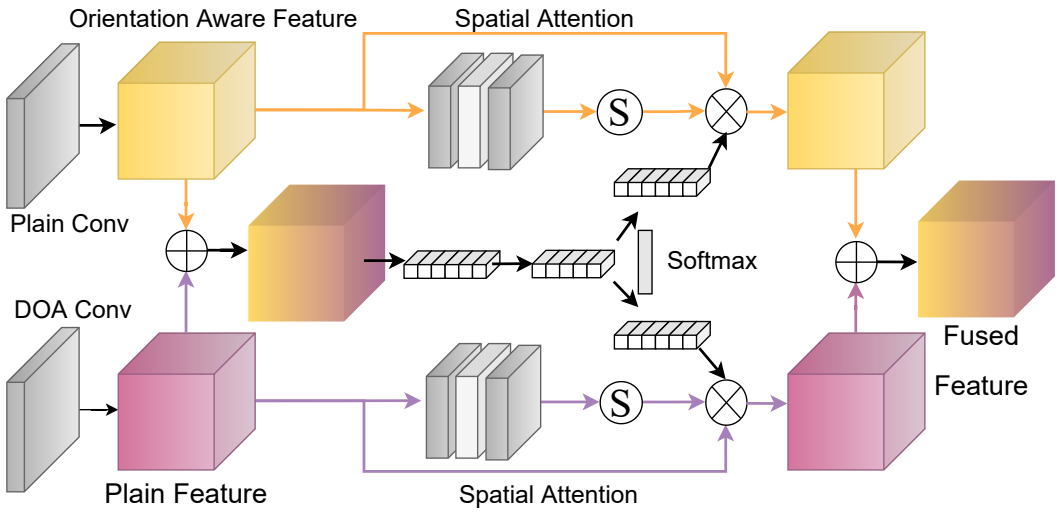

Figure 3.2: The schematic of the Selective Attention Fusion Module (SAFM). The plain features and orientation features are fused with the help of channel-wise selection and spatial-wise attention. This SAFM is inspired by the the framework of selective kernel network (SKNet) in [52].

3.1.3 Global and Local Fusion Module

Convolution with the kernel of 3x3 size has a local receptive field and has been proved very important for many computer vision tasks. But such convolution can not capture global contextual information of the whole input image, which is also very helpful for image segmentation [87] [22]. Retinal vessels have abundant context information [85], for example, the complex furcations and branches of blood vessels have their relative positions and sizes against the entire vascular system, which present a kind of symmetry

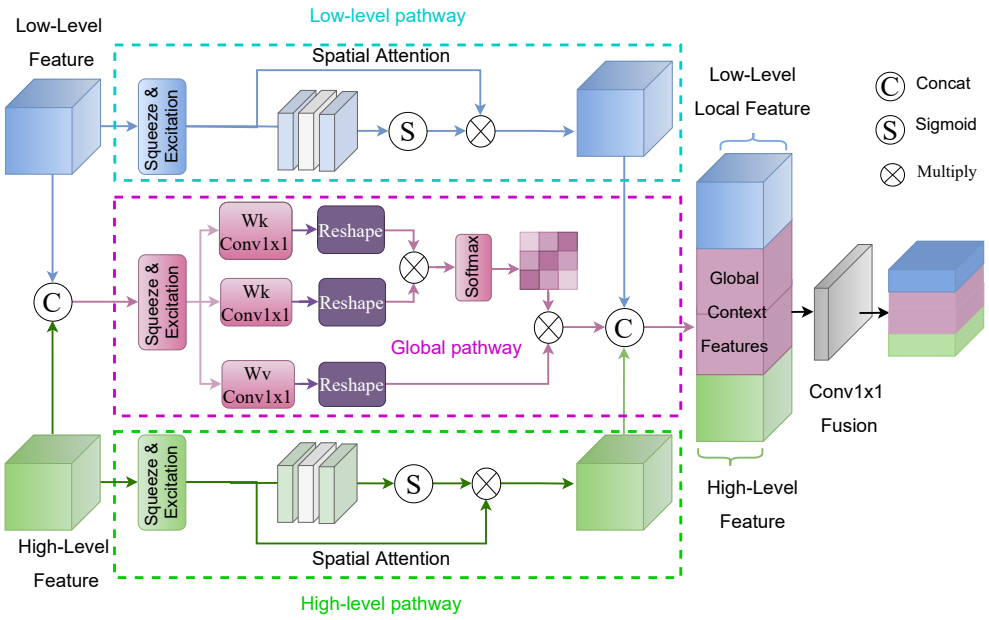

Figure 3.3: The proposed Global and Local Fusion Module (GLFM).

from the view of the whole retinal fundus image. Furthermore, some local lesion areas may occlude the blood vessels, and the global context information can help reconstruct the occluded vessels and improve the continuity of vessels from a holistic view. In addition, thin vessels contain rich local detail information and has a more complex orientation diversity. In a word, capturing global context and local detail information are equally important for vessel segmentation. A Global and Local Fusion Module (GLFM) is proposed herein to achieve this goal.

As shown in Fig. 3.3, GLFM is composed of three pathways, i.e., low-level, high-level and global pathways.

The squeeze-and-excitation operations $SE(.)$ [32] are used here for channel-wise integration because there is noise and other interference in the retinal fundus image. In both the low-level and high-level pathways, spatial attention $SPA(.)$ [68] with a 3x3 local convolution is used for focusing more attention on some local areas and vessel tissues. In addition, in the global pathway, both the low-level F_L and high-level F_H features are concatenated via $Concat(.)$ and used to model the global long-range dependencies via

self-attention $SA(.)$ mechanism, then the global context features F_G are obtained.

$$\begin{aligned} F_{LL} &= SPA(SE(F_L)) \\ F_{HH} &= SPA(SE(F_H)) \\ F_G &= SA(SE(Concat(F_H, F_L))) \end{aligned} \tag{3.5}$$

where F_{LL}, F_{HH} and F_G denote the low-level local features, high-level features and global context features, respectively.

Finally, these three kinds of features are concatenated via $Concat(.)$ and fused via a 1x1 convolution $Conv(.)$ to contain the output features of F_{glfm} which simultaneously captures the global context information and the local detail information.

$$F_{glfm} = Conv(Concat(F_{LL}, F_{HL}, F_G)) \tag{3.6}$$

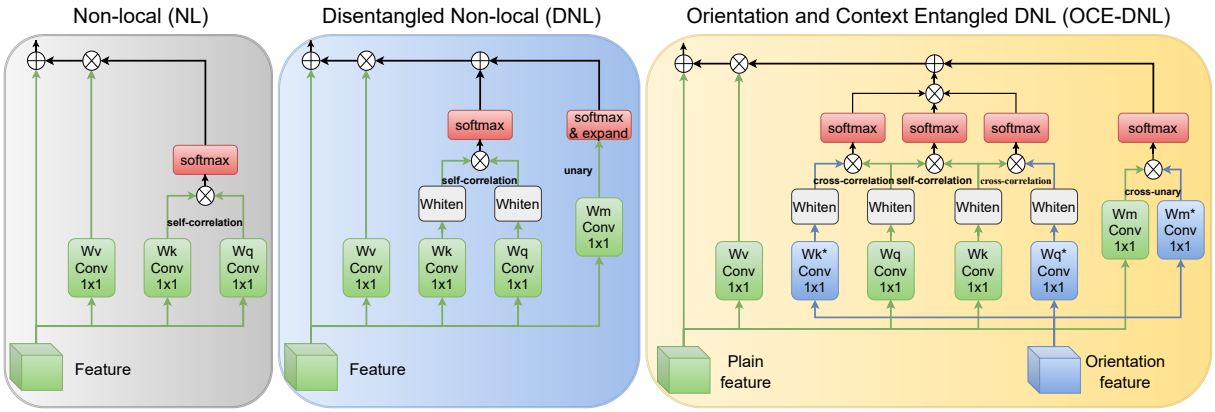

Figure 3.4: The schematic of nonlocal (NL) [87], Disentangled nonlocal (DNL) [104] and the proposed Orientation and Context Entangled nonlocal (OCE-DNL). Due to space limitation, OCE-NL is not presented here. 'Whiten' here means whitening operator. Note that the functions of whiten operator in DNL are to: 1. reduce the correlation between features. 2. make features have the same variance. More details about the whitening operator can be found in [104].

3.1.4 Orientation and Context Entangled NL/DNL

In order to fully mine the association between plain and orientation features, inspired by Disentangled nonlocal (DNL) [104], an Orientation and Context Entangled DNL (OCE-DNL) is proposed to entangle the orientation and context information together, in which orientation information is used to guide MSFM to better learn context information and discriminate blood vessels from the plain features. The NL and DNL are defined as

$$\begin{aligned} X_{out}^{NL}(x) &= X_{in}(x) + W_{NL}(x_i, x_j) \cdot V(X_{in}(x)) \\ X_{out}^{DNL}(x) &= X_{in}(x) + W_{DNL}(x_i, x_j) \cdot V(X_{in}(x)) \end{aligned} \tag{3.7}$$

where $X_{in}(x)$ means the input features and x denotes the position of pixel in the features. V means the value vector of NL. W_{NL} and W_{DNL} represent the self-correlation weights (self-attention map) yielded by computing the affinities between the Query and Key vectors, respectively, which are defined as

$$\begin{aligned} W_{NL}(x_i, x_j) &= Q_i^T \cdot K_j \\ W_{DNL}(x_i, x_j) &= (Q_i - \mu_Q)^T \cdot (K_j - \mu_K) + \mu_Q \cdot k_j \end{aligned} \tag{3.8}$$

where Q_i and K_j denote respectively the query and key vectors in NL. $(.)^T$ means the transposition operator. μ_Q and μ_K denote their mean values calculated by a whitening operation in DNL. x_i and x_j represent two different pixels in the input features.

We propose a novel nonlocal operator to explore the potential association between two related features by calculating their cross-correlation. To our best knowledge, this is the first attempt to introduce cross-correlation (cross-attention) into the nonlocal that was originally designed to capture only self-correlation or self-attention. The proposed

cross-correlation based entanglement is defined as

$$W_{OCE-NL}(x_i, x_j; y_i, y_j) = (Q_i^T \cdot K_j) \cdot (\tilde{Q_i}^T \cdot K_j) \cdot (Q_i^T \cdot \tilde{K_j}) \tag{3.9}$$

$$\begin{aligned} W_{OCE-DNL}(x_i, x_j; y_i, y_j) &= [(Q_i - \mu_Q)^T \cdot (K_j - \mu_K)] \\ &\cdot [(\tilde{Q_i} - \tilde{\mu_Q})^T \cdot (K_j - \mu_K)] \cdot [(Q_i - \mu_Q)^T \cdot (\tilde{K_j} - \tilde{\mu_K})] \\ &+ (\mu_Q \cdot k_j) \cdot (\tilde{\mu_Q} \cdot \tilde{k_j}) \end{aligned} \tag{3.10}$$

where Q_i and K_i denote respectively the query and key vectors of the plain features. $\tilde{Q_i}$ and $\tilde{K_i}$ denote respectively the query and key vectors of the orientation features. μ_Q and μ_K denote respectively the mean values of plain features which are calculated by a whitening operation [104]. $\tilde{\mu_Q}$ and $\tilde{\mu_K}$ denote the mean values of orientation features calculated by a whitening operation. x_i and x_j represent two different pixels in the plain features and y_i and y_j represent two different pixels in the orientation features. Note that the blue parts of Eq. 3.9 and 3.10 represent cross-correlation calculations.

As shown in Fig. 3.4, the multi-scale plain features outputted by the network and the multi-scale orientation features extracted in the down-sampling stage are decoupled into query and key vectors by 1x1 convolutions. As for the original nonlocal [87], only the self-correlation between the query and key of the plain feature is computed to obtain a self-attention map by multiplying them, and this self-attention map can model long-range dependencies and capture global context information. In order to calculate the cross-correlation between two different features, the respective query and key vectors for the two features are multiplied separately and the cross-attention maps can be obtained. Then the self-attention map is used to capture the context information in the plain features and the cross-attention map is used to model the global relationship between the plain features and the prior orientation features. Through computing cross-attention and applying cross-attention maps to the plain features, the context and orientation informa-

tion can be entangled together into the final output features, which are ultimately used to predict the vessels from the fundus background.

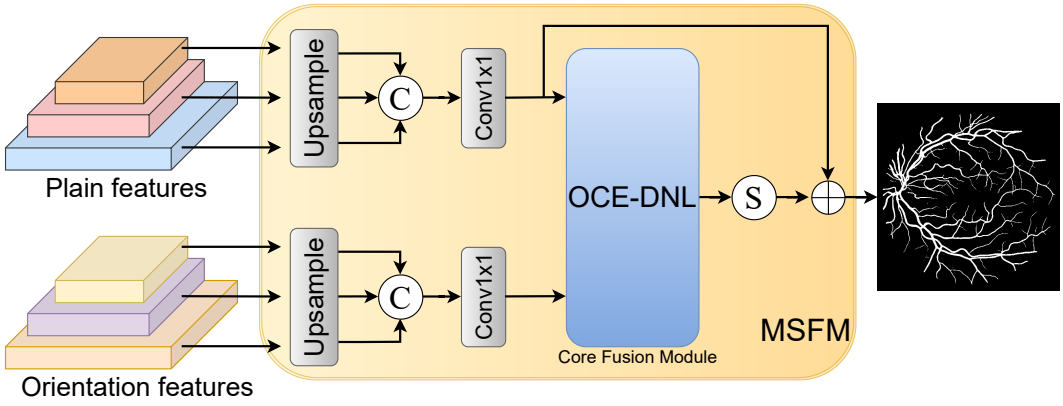

Figure 3.5: The proposed Multi-Scale Fusion Module (MSFM).

Leveraging multi-scale features is helpful for better vessel segmentation. Inspired by [94], we design a Multi-Scale Fusion Module (MSFM) to fuse multi-scale features including multi-scale plain features $F_i \in \mathbb{R}^{\frac{C}{r} \times H \times W}, (i, r = 1, 2, 3)$ and multi-scale orientation features $F_i^o \in \mathbb{R}^{\frac{C}{r} \times H \times W}, (i, r = 1, 2, 3)$, as shown in Fig. 3.5. The features at different scales are unified through up-sampling $Up_i(.)(i = 1, 2, 3)$ and concatenated in the channel dimension. Then these features are fused and dimensionally reduced through an 1x1 convolution, yielding the plain input feature F_{in} and the orientation input features F_{in}^o for MSFM as follows

$$F_{in} = Conv(Concat(Up_1(F_1), Up_2(F_2), Up_3(F_3)))$$
$$F_{in}^o = Conv(Concat(Up_1(F_1^o), Up_2(F_2^o), Up_3(F_3^o))) \tag{3.11}$$

The core fusion module of MSFM is the proposed OCE-DNL, which is used to entangle the orientation and context information together by computing cross-correlation between the plain and the prior orientation features. And the output feature F_{msfm} after fusion

can be obtained by entangling the context and orientation information as follows

$$F_{msfm} = \delta(OCEDNL(F_{in}, F_{in}^{o})) + F_{in} \tag{3.12}$$

where δ means the sigmoid function and $OCEDNL$ denotes the proposed OCE-DNL module.

3.1.5 Unbalanced Attention Refining Module

There are two kinds of unbalance in fundus images. One is the serious unbalance between the pixel numbers of blood vessels and the fundus background. The fundus background occupies the majority of pixels, while the blood vessels only take up a small proportion of total pixels. The other is the unbalance between the numbers of thick and thin vessels [100]. Thick vessels are generally large in width and hence occupy the majority of the blood vessels, while the width of thin vessels is usually 3-5 pixels. These unbalances make blood vessels difficult to be detected and identified from the background, and also make thin vessels harder to be detected due to their less prominent features. In order to deal with these unbalances, previous deep learning methods [40] usually design class-balanced loss or introduce weighted coefficients into the pixel-wise loss functions for imposing more penalties on thin vessels [99]. Here we propose a novel approach to tackle these unbalances from the perspective of visual attention mechanism.

As shown in Fig. 3.6, a novel Unbalanced Attention Refining Module (UARM) works to focus more visual attention on the vessels, especially the thin vessels. The proposed UARM is applied at the end of the network to refine the final output features of the network. Instead of directly applying spatial attention to the final output feature (the input feature $F_{in} \in \mathbb{R}^{C \times H \times W}$ of UARM), we first use a 1x1 convolution to reduce the dimension of the feature from 32 to 1 and use the softmax function to obtain a probability map $p(F_{in})$,

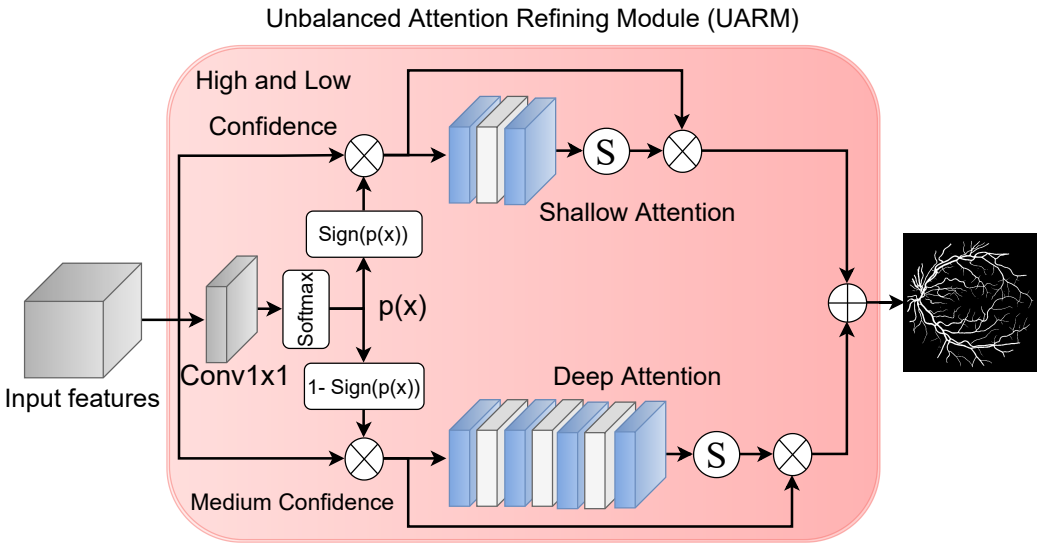

Figure 3.6: The proposed Unbalanced Attention Refining Module (UARM). The blue blocks represent the Conv3x3 and the gray blocks denote the ReLU function. The deeper attention module is adopted for mining hard sample. Note that the output channel number of the last convolutional block in both deep and shallow paths is 1 for calculating the probability map.

which describes the probability that each pixel belongs to a vessel, ranging from 0 to 1.

$$p(F_{in}) = \mathit{Softmax}(Conv(F_{in})) \tag{3.13}$$

Then, a pre-defined Sign function $Sign(.)$ is used to separate the probability values into three intervals, corresponding to three different regions in the image. In other words, we divide the image into three different regions of high confidence, medium confidence and low confidence according to the probability of pixels belonging to vessels. The $Sign(.)$ function is defined as

$$Sign(x) = \begin{cases} 1 & 0 \leq x < 0.4 \\ 0 & 0.4 \leq x < 0.7 \\ 1 & 0.7 \leq x < 1.0 \end{cases} \tag{3.14}$$

where x means the probability of each pixel in the probability map $p(F_{in})$.

By setting two thresholds between 0 and 1 (these two thresholds are experimently set to 0.4 and 0.7), we can separate the features F_2 with medium confidence regions. And high and low confidence regions are combined together and separated into features F_1 as well. F_1 and F_2 are fomulated as

$$F_1 = F_{in} \otimes Sign(pF_{in})), \quad F_2 = F_{in} \otimes (1 - Sign(p(F_{in})) \tag{3.15}$$

We found that the network usually pays more attention to the thick vessels and tends to ignore the thin vessels, that is, less attention is allocated to the ambiguous areas with medium confidence, where thin vessels are always located. Therefore, for such ambiguous regions, a deeper attention module $Att_d(.)$ with stronger discrimination ability is solely applied to gain more attention. For the regions with high and low confidence, a shallow attention module $Att_s(.)$ is used because thick vessels (usually composed of pixels with high probability) and the fundus background (usually composed of pixels with low probability) are highly discriminable. The process of UARM is defined as

$$F_{uarm} = UARM(F_{in}) = Att_s(F_1) + Att_d(F_2) \tag{3.16}$$

where F_{uarm} denotes the output features of UARM.

This unbalanced, biased approach for applying attention allows the network to better focus on uncertain areas that need more attention.

3.1.6 Loss Function of OCE-Net

A Cross-entropy loss $\mathcal{L}_{CE}$ is adopted as the loss function of our OCE-Net for vessel segmentation, which is defined as

$$\mathcal{L}_{CE}(p, q) = -\sum_{k=1}^{N} p_k * log(q_k) \tag{3.17}$$

3.2 Methodology of the Proposed GCC-UNet

3.2.1 Overall Architecture of GCC-UNet

The GCC-UNet architecture, depicted in Figure 3.7, utilizes the U-Net [71] as its backbone. In the downsampling phase, a Local Feature Extractor (Local FE), a Global Feature Extractor (Global FE), and a Selective Graph Attention Fusion (SGAF) module are introduced to merge the local features extracted by a plain CNN with the global context features extracted by a Capsule Neural Network. To serve as a global feature extractor, we propose a Graph Capsule Convolution (GC Conv) operator, which replaces the vanilla capsule convolution operator. The Bottleneck Graph Attention (BGA) module is inserted in the bottleneck to enhance vessel continuity by modeling the connectivity of vessel nodes flowing on the graph. In the upsampling phase, the extracted global context features are passed directly to the upsampling builder to reduce computational costs. The SGAF is then used to fuse the global features with the upsampled local features. Finally, we propose a Multi-Scale Graph Fusion (MSGF) module to leverage the features from different stages of the U-Net.

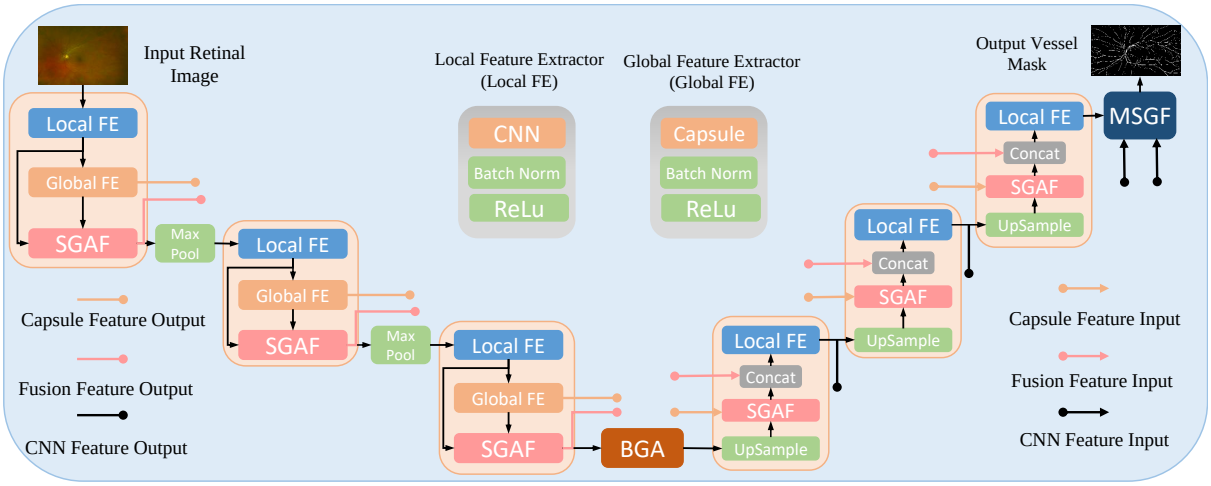

Figure 3.7: The network architecture of the proposed GCC-UNet.

3.2.2 Graph Capsule Convolution

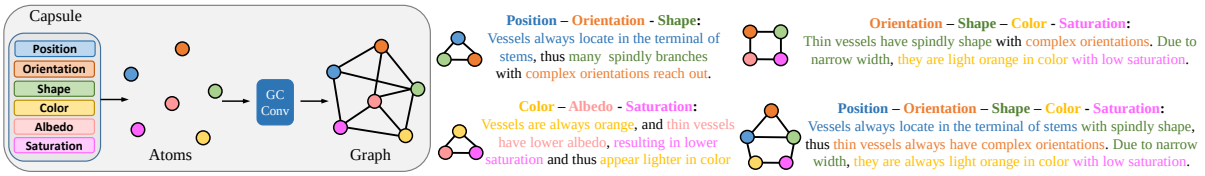

Figure 3.8: The schematic of relationships among atoms in capsules. We inreoduce graph into capsules to model the relationships among atoms.

The capsule convolutional network is known for its inherent capability to learn spatial correlations between objects, which makes it efficient in identifying multiple objects in an image, even when they overlap significantly. In contrast to traditional CNNs that use scalar elements, the capsule NN employs vectors as its basic components. Each capsule contains a vector that can capture various intrinsic characteristics of an object, such as its pose (position, size, orientation, shape), deformation, color, saturation, and so on, thereby forming meaningful relationships between parts and wholes and providing global context information. The length of each vector represents the estimated probability of the object's presence in the image. Compared to CNN, capsule NN has an advantage in capturing local detailed features, but it has a poorer ability to model translation invariance and part-to-whole relationships.

In Fig. 3.9, the input feature extracted by plain CNN convolution is transformed into primary capsules, which represent low-level entities in this layer. Dynamic routing [72] is then employed to route the low-level capsules to the high-level ones, capturing the part-to-whole relationships. Dynamic routing can be seen as a transfer matrix with attention weights that focuses on important capsules and vectors while ignoring unimportant ones. However, the dynamic routing proposed in [72] fails to model the correlations among different capsules and atoms in the capsules. To address this, we introduce graph reasoning into the dynamic routing process, as depicted in Fig. 3.9, to model these correlations. By doing so, we are able to adequately model the relationships among the channels, capsules, and atoms dimensions.

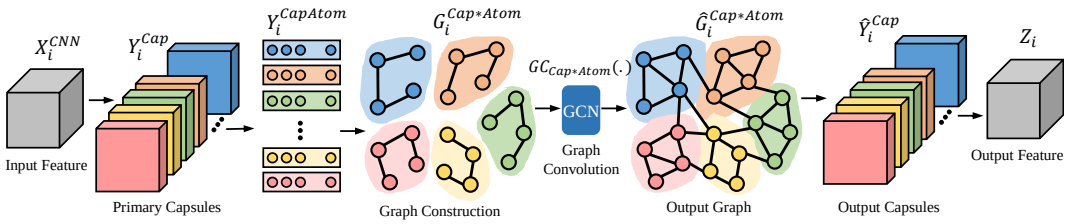

Figure 3.9: The proposed Graph Capsule Convolution (GC Conv).

As shown in Fig. 3.9, the input CNN feature X_i^{CNN} has a shape of $[B, C, H, W]$. This feature is transformed into primary capsules Y_i^{Cap}, which have a shape of $[B, H, W, k^2, C, L, V]$, where C, L, and V represent the number of channels, capsules, and atoms in each capsule. The channel dimension is split from the features to obtain independent features $Y_i^{Channel}$ with a shape of $[B, H, W, k^2, C, 1, 1]$, which are independent of the capsules and atoms dimensions. Similarly, the capsules and atoms dimensions are split to obtain independent features $Y_i^{CapAtom}$ with a shape of $[B, H, W, k^2, 1, L, V]$. By multiplying the channels of capsules and atoms dimensions, we obtain a feature $Y_i^{Cap*Atom}$ with a shape of $[B, H, W, k^2, 1, L * V]$. We then use average pooling to remove the H, W, and K^2 dimensions and construct a graph $G_i^{Channel}$ along the channel dimension C for $Y^{Channel}i$. Similarly, we construct a graph $G_i^{Cap*Atom}$ along the $L * V$ dimension for $Y_i^{Cap*Atom}$. We apply a graph convolution $GC_{Channel}(.)$ on $G_i^{Channel}$ to obtain the output graph feature $\widehat{G}_i^{Channel}$. We also apply a graph convolution $GC_{Cap*Atom}(.)$ on $G_i^{Cap*Atom}$ to obtain the output graph feature $\widehat{G}_i^{Cap*Atom}$. Finally, we integrate $\widehat{G}_i^{Channel}$ and $\widehat{G}_i^{Cap*Atom}$ using addition and expansion operators, and transfer them into capsule features $\widehat{Y}_i^{Cap}$ to obtain the output feature Z_i.

3.2.3 Selective Graph Attention Fusion Module

It is crucial for models to be able to incorporate global context information, given the variations in scale, orientation, and partial occlusions of fundus vessels. However, capsule neural networks have limitations in learning critical local features. As a result, the optimal approach would be to integrate capsule convolution with plain CNN models to

enable the model to learn both local and global features. To attain the best fusion performance, we present a novel Selective Graph Attention Fusion (SGAF) module. This module exploits the graph structure to model the inner-channel relationships of both the local and global features while simultaneously learning the inter-channel correlations between them.

In Fig. 3.10, we have two types of input features: local features X_i^{Local} obtained through plain CNN convolution, and global context features X_i^{Global} obtained through capsule convolution. Then we add X_i^{Local} and X_i^{Global} to obtain the fusion feature X_i^{Fusion}. We then apply three independent Average Pooling operators to eliminate spatial dimensions, preserving only the channel dimension. After pooling, we construct graphs along the channel dimension of the three features, resulting in four independent graphs: G_i^{Local}, G_i^{Global}, $G_i^{Fusion-Local}$ and $G_i^{Fusion-Global}$. Note that we construct two graphs from X_i^{Fusion}, resulting in two individual graphs $G_i^{Fusion-Local}$ and $G_i^{Fusion-Global}$. These two graphs provide shared fusion information for both the local feature X_i^{Local} and global feature X_i^{Global}. We assume that the two graphs should contain different topological structures of channels from X_i^{Local} and X_i^{Global} after learning and reweighting of graph convolution operators.

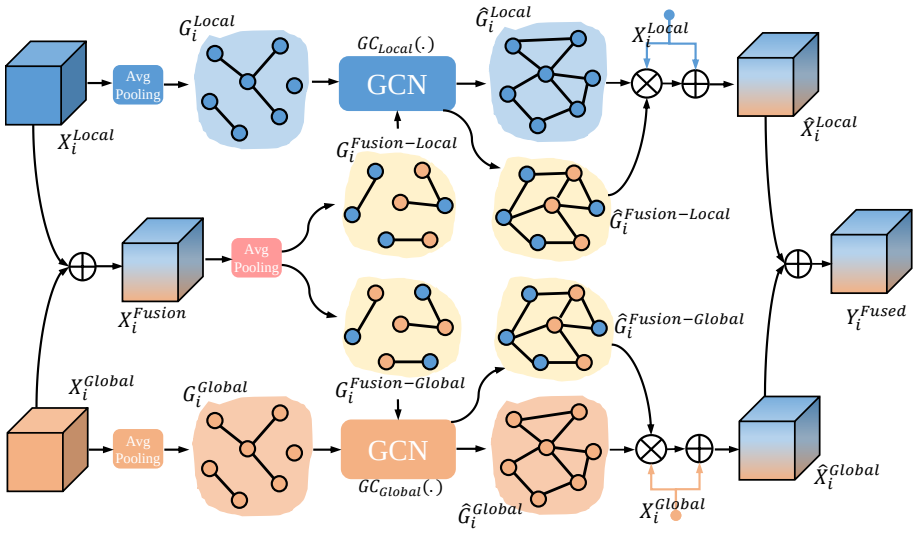

Figure 3.10: The architecture of the proposed Selective Graph Attention Fusion (SGAF) module.

The graph represents each channel of the feature as a node. To learn the connectivity and relationships among nodes (channels), we apply only two graph convolution operators on the four constructed graphs: $GC_{Local}(.)$ for G_i^{Local} and $G_i^{Fusion-Local}$, and $GC_{Global}(.)$ for G_i^{Global} and $G_i^{Fusion-Global}$. By using shared graph convolution, the local or global graphs can share nodes and connectivity information with the fusion graphs, resulting in better connectivity weight adjustment, more informative representation flow on the graph, and reduced computational cost and parameters. After applying graph convolu-

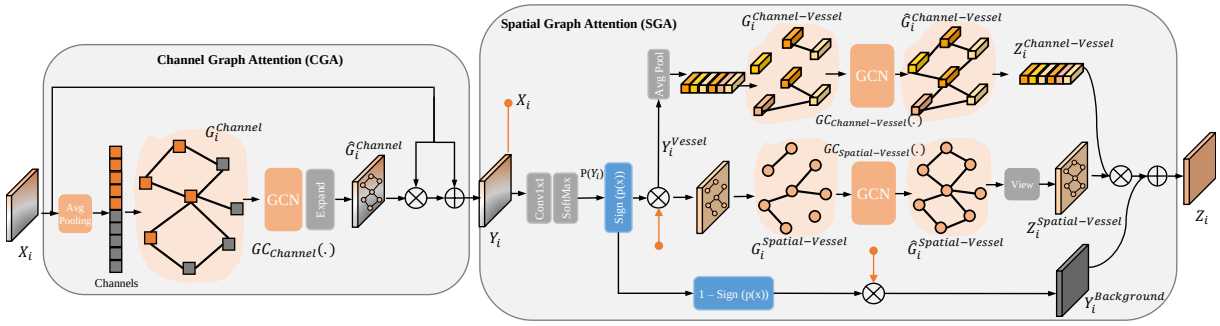

Figure 3.11: The architecture of the proposed Bottleneck Graph Attention (BGA) module.

tion, we obtain four output graphs: $\widehat{G}_i^{Local}$, $\widehat{G}_i^{Fusion-Local}$, $\widehat{G}_i^{Global}$, and $\widehat{G}_i^{Fusion-Global}$. We then apply $\widehat{G}_i^{Local}$ and $\widehat{G}_i^{Global}$ on the input features X_i^{Local} and X_i^{Global} using multiplication and addition operators, respectively, which can be viewed as a kind of self-attention because the graph attention weights generated from the input features are applied back on the channels of original input features. At the same time, $\widehat{G}_i^{Fusion-Local}$ and $\widehat{G}_i^{Fusion-Global}$ are applied on the input features X_i^{Local} and X_i^{Global} using multiplication operators. The resulting refined output features are denoted as $\widehat{X}_i^{Local}$ and $\widehat{X}_i^{Global}$. Finally, we add $\widehat{X}_i^{Local}$ and $\widehat{X}_i^{Global}$ together to obtain the fused feature Y_i^{Fused}.

3.2.4 Bottleneck Graph Attention Module

To improve vessel continuity, particularly for thin vessels, we propose a novel Bottleneck Graph Attention (BGA) module comprising of Channel-wise Graph Attention (CGA) and Spatial Graph Attention (SGA). The input features X_i are first fed into CGA, where an Av-

erage Pooling operator is used to extract channel-only features, transforming the feature shape from $[B, C, H, W]$ to $[B, C, 1, 1]$. A graph $G_i^{Channel}$ is constructed along the channel dimension, where each node represents a channel of features and edges connectivity between nodes indicates their relationship. A graph convolution operator $GC_{Channel}(.)$ is applied to $G_i^{Channel}$, producing an output graph $\widehat{G}_i^{Channel}$ with re-weighted connectivity and re-modelled channel relationships. The refined graph $\widehat{G}_i^{Channel}$ is then expanded along the spatial dimensions and recovered to $[B, C, H, W]$. The refined feature and graph representation $\widehat{G}_i^{Channel}$ are fused with the input feature Xi through multiplication and addition, generating the output feature Y_i. The CGA module enables the representation of channel dependencies as a graph and captures the relationships among channels.

In the SGA module, the input feature is Y_i, and a feature selector is proposed to extract vessels from the fundus background. The feature selector applies a conv1x1 $Conv(.)$ operator to reduce the dimension of Y_i and Softmax function $Softmax(.)$ to calculate a probability map $p(Y_i)$, which contains information about the probability that each pixel belongs to a vessel, ranging from 0 to 1.

$$p(Y_i) = \textit{Softmax}(Conv(Y_i)) \tag{3.18}$$

A pre-defined piecewise function called the Sign function $Sign(.)$ is then applied to separate the probability values into two intervals. Specifically, a threshold of 0.4 was set in our experiments, indicating that pixels with probability values greater than 0.4 correspond to blood vessel pixels, while those with values less than 0.4 correspond to background pixelse. This allows for effective separation of vessel regions from the background. The $Sign(.)$ function is defined as

$$Sign(x) = \begin{cases} 1 & x > 0.4 \qquad (Vessel) \\ 0 & x < 0.4 \qquad (Background) \end{cases} \tag{3.19}$$

where x means the probability of each pixel in the probability map $p(Y_i)$. Using the Sign function, we can obtain the vessel features Y_i^{Vessel} and background features $Y_i^{Background}$ separately from the input features Y_i based on their probability values. To improve the continuity of vessels, we perform two individual operations. The first operation involves constructing a graph $G_i^{Spatial-Vessel}$ for the vessel-only features based on their spatial distribution. Nodes and edges in the graph represent the vessels and their connectivity, respectively. We then apply a graph convolution $GC_{Spatial-Vessel}(.)$ on the graph $G_i^{Spatial-Vessel}$ to learn information about the nodes and edges connectivity, aiming to improve the continuity of vessels without interference from the background, especially noise and other issues in the background. This yields the output features of vessels $Z_i^{Spatial-Vessel}$. In addition to improving vascular continuity in spatial distribution, we also enhance semantic consistency. To achieve this, we use an average pooling operator to extract channel information of vessels, and construct a graph $G_i^{Channel-Vessel}$ for these channels. We then apply a graph convolution operator $GC_{Channel-Vessel}(.)$ to learn the graph representation of channels, yielding the output features of vessels $Z_i^{Channel-Vessel}$ after viewing operation. Finally, we multiply $Z_i^{Spatial-Vessel}$ and $Z_i^{Channel-Vessel}$, add $Y_j^{Background}$, and obtain refined features Z_i whose vascular continuity has been enhanced.

3.2.5 Multi-Scale Graph Fusion Module

To integrate the multi-scale features extracted from different stages of the UNet, we propose a Multi-Scale Graph Fusion module, which is depicted in Fig. 3.12. The input features X_i^a, X_i^b and X_i^c are obtained from different upsampling stages of the UNet. Firstly, we apply upsampling and conv1x1 operators on X_i^b and X_i^c to reshape their spatial and channel dimensions to match those of X_i^a. Subsequently, we apply Average Pooling operators on these features to reduce their dimensionality and preserve only channel-wise information. Then, we construct three independent graphs, G_i^a, G_i^b and G_i^c, for these features along the channel-wise dimension.

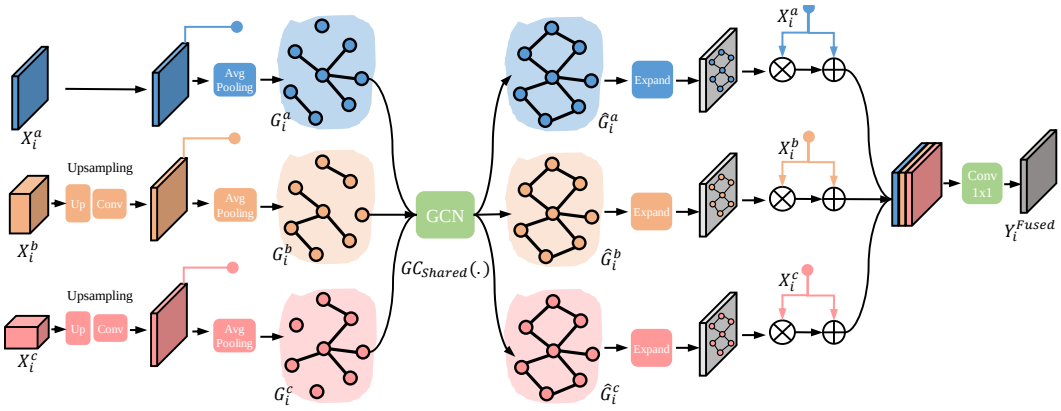

Figure 3.12: The proposed Multi-Scale Graph Fusion (MSGF) module.

Instead of adopting three individual graph convolution on these three independent graphs, we use only a single shared graph convolution $GC_{Shared}(.)$ to conduct convolutional process on G_i^a, G_i^b and G_i^c, because we assume that graphs constructed from different scales with the same input feature are supposed to have similar graph pattern and nodes connectivity. Adopting shared graph convolution can simultaneously capture the topological structure representations of G_i^a, G_i^b and G_i^c, and adjust the connectivity of nodes on the graph by taking other graphs into the consideration, so that the information can propogate and flow on the graphs constructed from multi-scale features. After applying graph convolution operators, we obtain three independent graph representations $\widehat{G}_i^a$, $\widehat{G}_i^b$ and $\widehat{G}_i^c$. And then these ouput channel-wise graphs are expanded spatially and applied directly on each input feature X_i^a, X_i^b and X_i^c, respectively, obtaining three refined features $\widehat{X}_i^a$, $\widehat{X}_i^b$ and $\widehat{X}_i^c$. Finally, we concatenate the three refined features and adopt a conv1x1 operator to reduce the dimension and generate the output fused feature Y_i^{Fused}.

3.2.6 Loss Function of GCC-UNet

The Cross Entropy (CE) loss $\mathcal{L}_{CE}$ is adopted as the loss function of our GCC-UNet, which is defined as

$$\mathcal{L}_{CE}(p,q) = -\sum_{k=1}^{N} p_k * log(q_k) \tag{3.20}$$

Chapter 4

Discussion of All the Findings

4.1 Datasets and Materials

4.1.1 Retinal Fundus Datasets

Our model was trained on three widely used datasets: DRIVE [77], STARE [31], and CHASEDB1 [19].

The DRIVE dataset contains 40 pairs of retinal fundus images with their corresponding labels, which were manually delineated by two human observers, and the labels of the first observer are usually used as the ground truth. The size of each fundus image in DRIVE is 565 $\times$ 584 pixels, and the training and test sets are nonoverlapping with each other; each contains 20 pairs of images.

The STARE dataset consists of 20 fundus images with their corresponding manual labels annotated by two human experts. The resolution of each image is 700 $\times$ 605 pixels. Generally, the first 10 images and their labels are used as training set, and the other 10 images are used as test set.

The CHASEDB1 dataset contains 28 fundus images and their labels, with a resolution of 999 $\times$ 960 pixels. The first 20 images are usually used as training set and the other eight images are considered as the test set.

To evaluate the generalization performance of our model, we also tested OCE-Net on some challenging datasets, including AV-WIDE [17], UoA-DR [7], RFMiD [65], and UK Biobank [78].

The AV-WIDE dataset contains 30 wide-FOV color images, and the arteries and veins were annotated separately for artery-vein classification. The resolutions of images vary, but most of them are around 1300 $\times$ 800 pixels. The vessels are usually thin in AV-WIDE. The UoA-DR dataset consists of 200 images with a resolution of 2124 $\times$ 2056 pixels, which were collected by the University of Auckland. The RFMiD dataset contains 3200 fundus images. Of these, 1920 images of them are allocated to the training set, 640 images are in the validation set, and the remaining 640 images are in the test set. The fundus images were captured by three different fundus cameras. The sizes of images vary, having the resolutions of 4288 $\times$ 2848 (277 images), 2048 $\times$ 1536 (150 images), and 2144 $\times$ 1424 (1493 images), respectively. The UKBB dataset contains 100K fundus images with the size of 2048 $\times$ 1536 pixels.

Note that instead of retraining the model on these datasets, we just tested on these challenging sets using the models already trained on DRIVE [77].

4.1.2 Evaluation Metrics

We evaluated our model with some frequently used metrics, including the F1 score (F1), accuracy (Acc), sensitivity (SE), specificity (SP), and area under the ROC curve (AUC), which are defined as

$$\begin{aligned} SE = Recall = \frac{TP}{TP + FN} \qquad SP = \frac{TN}{TN + FP} \\ F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad Acc = \frac{TP + TN}{TP + TN + FP + FN} \end{aligned} \tag{4.1}$$

where TP, TN, FP, and FN represent the numbers of true positive, true negative, false positive, and false negative pixels, respectively. In addition, we also adopted some im-

proved metrics proposed by Gegundez et al. [24], including connectivity (C), overlapping area (A), consistency of vessel length (L), and the overall metric (F). The overall metric (F) is defined as

$$F = C \times A \times L \tag{4.2}$$

Moreover, Yan et al. [98] proposed some other novel metrics, including rSE, rSP, and rAcc, for improving the corresponding SE, SP, and Acc metrics, respectively. We also used these newly designed metrics to evaluate our model and compare it with other methods. More details about these redefined metrics can be found in [98]. Additionally, the Matthews correlation coefficient (Mcc) [39] was also used to evaluate our model.

4.2 Experiments of OCE-Net

4.2.1 Implementation details

We built our model using the PyTorch framework [66]. The model was trained on a TITAN XP GPU with 12 G memory. The Adam optimizer [41] and the cosine annealing learning rate (LR) were adopted during the training. Before training, the fundus images were converted from RGB to and grayscale, then Gamma correction and CLAHE [67] were applied to enhance the lightness and contrast of the grayscale fundus images. The images were randomly cropped into patches with a size of 48 $\times$ 48 pixels owing to the limitation of GPU memory, and the number of cropped patches was set to 15,000. The batch size was set to 32, and the total epoch was set to 50. The early stopping strategy was adopted, and the epoch was set to 8.

4.2.2 Overall comparison with other methods

To evaluate the performance of our method and demonstrate its superiority, we conducted extensive quantitative and qualitative experiments.

As shown in Table 4.15, 4.16, and 4.17, we compared our method with numerous state-of-the-art methods on the DRIVE [77], STARE [31], and CHASEDB1 [19] datasets. In terms of the commonly used indicators, our method outperforms most of previous state-of-the-art methods, especially on DRIVE.

Among these compared methods, BTS-DSN [28] neither focuses more attention on local thin vessels and nor captures global context information; therefore, it exhibits poor performance on both SE and SP. CSU-Net [85] has a context path to capture the global context of images, but neglects to emphasize local details of thin vessel; therefore, it shows good performance on SP but relatively poor performance in SE. CTF-Net [86] has a specially designed Fine segNet to deal with local thin vessels, but neglects the global context of the whole vascular system; therefore, it shows high indicators on SE but low indicators on SP. Unlike these methods, our OCE-Net captures both global and local information of vessel as well as focuses more attention on thin vessels, so OCE-Net shows promising performance on both SE and SP.

As shown in Table 4.18, in terms of some newly redefined metrics, our method also outperforms many other recent methods.

SkelCon [81] adopts the contrastive learning (CL) strategy to better fit the shape of vessel for improving the connectivity of thin vessels; therefore, it shows better performance on vessel connectivity. However, it does not capture global context information of vessels or focus more attention on thin vessels, so it exhibits poor performance on rSE and rSP indicators. In comparison, as it takes both local and global information into consideration, our OCE-Net exhibits better performance than SkelCon on both rSE and rSP.

Note that in Table 4.17, on the CHASEDB1 dataset, CTF-Net [86] shows a better F1 score and SP than our OCE-Net, which is mainly because the fundus images in CHASEDB1 are different from those in the DRIVE and STARE datasets. As shown in Fig. 4.9, there are almost no thin vessels in the images from CHASEDB1, but primarily thick vessels. In addition, the orientations of the blood vessels in CHASEDB1 are also less complicated. However, our newly designed modules for OCE-Net are mainly used to deal with thin

Table 4.1: Experiments conducted on DRIVE for performance comparison of segmenting thin vessels (Unit: %). We calculated the P-value of T-test between our method and other methods.

Method	Connectivity (C)	Overlapping Aera (A)	Consistency (L)
UNet	91.34 (<0.01)	85.72 (<0.01)	78.66 (<0.01)
Attention UNet	92.03 (<0.01)	87.89 (<0.01)	80.14 (<0.01)
Dense UNet	92.27 (<0.01)	88.02 (<0.05)	80.36 (<0.01)
OCE-Net	92.45	88.23	80.68

vessels with multiple orientations and contexts. Therefore, our OCE-Net shows poorer performance on CHASEDB1 than on DRIVE and STARE. In addition, in terms of some novel indicators in Table 4.18, SkelCon [81] exhibits better performance on Connectivity (C) and Consistency (L) because SkelCon has specially designed modules to improve the connectivity and consistency of vessels.

As shown in Fig. 4.8, on these three widely used datasets, our method exhibits better visual segmentation results than the previous methods. Compared with other methods in Fig. 4.9, our method can effectively segment the thin blood vessels. In contrast, many other methods cannot segment thin vessels well.

In addition, we also conducted a quantitative comparison of detecting thin vessels on DRIVE. We used morphological image processing to separate out thin vessels. We chose three novel indicators, that is, Connectivity (C), Overlapping Area (A), and Consistency (L), proposed in [98]. Because thin vessels are usually small, it is not suitable to evaluate them with ACC and AUC. As shown in Table. 4.1, our OCE-Net exhibits better performance in detecting thin vessels than other methods.

4.2.3 Comparison and ablation study of individual modules

Comparison between the proposed GLFM and other attention gates

The proposed GLFM can be viewed as an alternative of attention gates of U-shaped networks. We compared our GLFM with other attention blocks and self-attention blocks by

Table 4.2: Quantitative comparison with other state-of-the-art methods on **DRIVE**. We calculated the P-value of T-test between our method and other methods. Red: the best, Blue: the second best.

Method	Year	F1	Se	Sp	Acc	AUC
2nd observer [77]	2004	N.A	77.60(<0.01)	97.24(<0.01)	94.72(<0.01)	N.A
HED [96]	2016	80.89(<0.01)	76.27(<0.01)	98.01(<0.01)	95.24(<0.01)	97.58(<0.01)
DeepVessel [21]	2016	N.A	76.12(<0.01)	97.68(<0.01)	95.23(<0.01)	97.52(<0.01)
Orlando et al. [64]	2017	N.A	78.97(<0.01)	96.84(<0.01)	94.54(<0.01)	95.06(<0.01)
JL-UNet [99]	2018	81.02(<0.02)	76.53(<0.01)	98.18(<0.01)	95.42(<0.03)	97.52(<0.01)
CC-Net [18]	2018	N.A	76.25(<0.01)	98.09(<0.01)	95.28(<0.01)	96.78(<0.01)
Att UNet [61]	2018	82.32(<0.01)	79.46(<0.01)	97.89(<0.01)	95.64(<0.01)	97.99(<0.01)
Dense UNet [51]	2018	82.79(<0.01)	79.85(<0.01)	98.05(<0.01)	95.73(<0.01)	98.10(<0.01)
Yan et al. [100]	2019	N.A	76.31(<0.01)	98.20(<0.01)	95.33(<0.01)	97.50(<0.01)
BTS-DSN [28]	2019	82.08(<0.02)	78.00(<0.01)	98.06(<0.01)	95.51(<0.01)	97.96(<0.01)
DUNet [38]	2019	82.49(<0.01)	79.84(<0.01)	98.03(<0.01)	95.75(<0.04)	98.11(<0.01)
CTF-Net [86]	2020	82.41(<0.01)	78.49(<0.01)	98.13(<0.01)	95.67(<0.01)	97.88(<0.01)
CSU-Net [85]	2021	82.51(<0.01)	80.71(<0.01)	97.82(<0.01)	95.65(<0.01)	98.01(<0.01)
OCE-Net (Ours)	2022	83.02	80.18	98.26	95.81	98.21

Table 4.3: Quantitative comparison with other state-of-the-art methods on **STARE** dataset. We calculated the P-value of T-test between our method and other methods.

Method	Year	F1	Se	Sp	Acc	AUC
HED [96]	2016	82.68(<0.01)	80.76(<0.01)	98.22(<0.01)	96.41(<0.01)	98.24(<0.01)
Orlando et al. [64]	2017	N.A	76.80(<0.01)	97.38(<0.01)	95.19(<0.01)	95.70(<0.01)
JL-UNet [99]	2018	N.A	75.81(<0.01)	98.46(<0.01)	96.12(<0.01)	98.01(<0.01)
Att UNet [61]	2018	81.36(<0.01)	80.67(<0.01)	98.16(<0.01)	96.32(<0.01)	98.33(<0.01)
CC-Net [18]	2018	N.A	77.09(<0.01)	98.48(<0.01)	96.33(<0.01)	97.00(<0.01)
Dense UNet [51]	2018	82.32(<0.01)	78.59(<0.01)	98.42(<0.01)	96.44(<0.25)	98.47(<0.01)
Yan et al. [100]	2019	N.A	77.35(<0.01)	98.57(<0.01)	96.38(<0.01)	98.33(<0.01)
BTS-DSN [28]	2019	83.62(<0.01)	82.01(<0.01)	98.28(<0.01)	96.60(<0.01)	98.72(<0.01)
DUNet [38]	2019	82.30(<0.01)	78.92(<0.01)	98.16(<0.01)	96.34(<0.01)	98.43(<0.01)
OCE-Net (Ours)	2022	83.41	80.12	98.65	96.72	98.76

individually adding different attention gates to the baseline UNet. Among the methods listed in Table. 4.6, the original attention gate in [61] and CBAM block [93] can be viewed as the modules that only carry out 'local' attention. The NL, DNL, self-attention blocks can be seen as the methods that only apply 'global' attention. In contrast, the proposed GLFM not only exerts 'local' attention, but also applies 'global' attention.

From Table 4.6, it can be seen that the proposed GLFM outperforms all the 'local'-only and 'global'-only attention modules. This demonstrates that it is important to capture both of the global context information and the local details of vessels because global con-

Table 4.4: Quantitative comparison with other state-of-the-art methods on **CHASEDB1** dataset. We calculated the P-value of T-test between our method and other methods.

Method	Year	F1	Se	Sp	Acc	AUC
2nd observer [77]	2004	N.A	81.05(<0.01)	97.11(<0.01)	95.45(<0.01)	N.A
HED [96]	2016	78.15(<0.01)	75.16(<0.01)	98.05(<0.01)	95.97(<0.01)	97.96(<0.01)
DeepVessel [21]	2016	N.A	74.12(<0.01)	97.01(<0.01)	96.09(<0.01)	97.90(<0.01)
Orlando et al. [64]	2017	N.A	75.65(<0.01)	96.55(<0.01)	94.67(<0.01)	94.78(<0.01)
JL-UNet [99]	2018	N.A	76.33(<0.01)	98.09(<0.01)	96.10(<0.01)	97.81(<0.01)
Att UNet [61]	2018	80.12(<0.01)	80.10(<0.01)	98.04(<0.01)	96.42(<0.01)	98.40(<0.01)
Dense UNet [51]	2018	79.01(<0.01)	78.93(<0.01)	97.92(<0.01)	96.11(<0.01)	98.35(<0.01)
Yan et al. [100]	2019	N.A	76.41(<0.01)	98.06(<0.01)	96.07(<0.01)	97.76(<0.01)
BTS-DSN [28]	2019	79.83(<0.01)	78.88(<0.01)	98.01(<0.01)	96.27(<0.01)	98.40(<0.01)
DUNet [38]	2019	79.32(<0.01)	77.35(<0.01)	98.01(<0.01)	96.18(<0.01)	98.39(<0.01)
CTF-Net [86]	2019	82.20(<0.01)	79.48(<0.01)	98.42(<0.01)	96.48(<0.01)	98.47(<0.01)
OCE-Net (Ours)	2022	81.96	81.38	98.24	96.78	98.72

Table 4.5: Quantitative comparison of several newly proposed metrics [98] between our method and other methods on **DRIVE** dataset. Note that the model of SA-UNet [26] was reproduced by us in order to eliminate the inconsistency of indicators in different papers [38] [26]. We calculated the P-value of T-test between our method and other methods.

Method	F	C	A	L	rSe	rSp	rAcc	Mcc
2nd observer	83.75	100	93.98	89.06	85.84	99.19	95.74	76.00
HED [96]	80.09(<0.01)	99.75(<0.01)	90.06(<0.01)	89.11(<0.01)	71.57(<0.01)	95.11(<0.01)	89.08(<0.01)	66.00(<0.01)
DRIU [55]	80.43(<0.35)	99.56(<0.01)	91.52(<0.01)	88.23(<0.08)	82.36(<0.01)	96.85(<0.03)	93.13(<0.01)	71.61(<0.01)
DeepVessel [21]	61.74(<0.01)	99.60(<0.01)	84.23(<0.01)	73.38(<0.01)	54.93(<0.01)	99.78(<0.01)	88.32(<0.01)	73.34(<0.01)
V-GAN [76]	84.82(<0.01)	99.64(<0.01)	94.69(<0.01)	89.84(<0.01)	80.77(<0.01)	99.63(<0.19)	94.76(<0.01)	80.24(<0.07)
JL-UNet [99]	81.06(<0.01)	99.61(<0.01)	93.08(<0.01)	87.35(<0.01)	76.11(<0.01)	99.57(<0.25)	93.53(<0.01)	78.98(<0.01)
SWT-FCN [62]	83.92(<0.01)	99.73(<0.01)	94.36(<0.18)	89.11(<0.01)	79.63(<0.01)	99.64(<0.01)	94.48(<0.01)	80.53(<0.01)
DeepDyn [40]	84.53(<0.01)	90.70(<0.01)	94.58(<0.01)	89.61(<0.01)	81.52(<0.01)	99.44(<0.01)	94.82(<0.03)	80.02(<0.01)
DAP [79]	82.55(<0.01)	99.72(<0.01)	93.74(<0.01)	88.24(<0.01)	78.57(<0.01)	99.57(<0.15)	94.15(<0.01)	79.00(<0.01)
DRIS-GP [11]	84.94(<0.20)	99.68(<0.01)	94.91(<0.01)	89.74(<0.02)	80.22(<0.40)	99.64(<0.01)	94.66(<0.01)	81.84(<0.01)
SA-UNet [26]	83.19(<0.01)	99.61(<0.01)	93.96(<0.01)	88.89(<0.01)	80.01(<0.01)	99.32(<0.01)	94.53(<0.01)	79.67(<0.01)
SkelCon [81]	85.30(<0.01)	99.85(<0.01)	93.58(<0.01)	91.26(<0.01)	83.23(<0.01)	98.59(<0.01)	94.61(<0.01)	80.30(<0.01)
OCE-Net	85.83	99.80	95.07	90.43	83.83	99.29	95.30	80.40

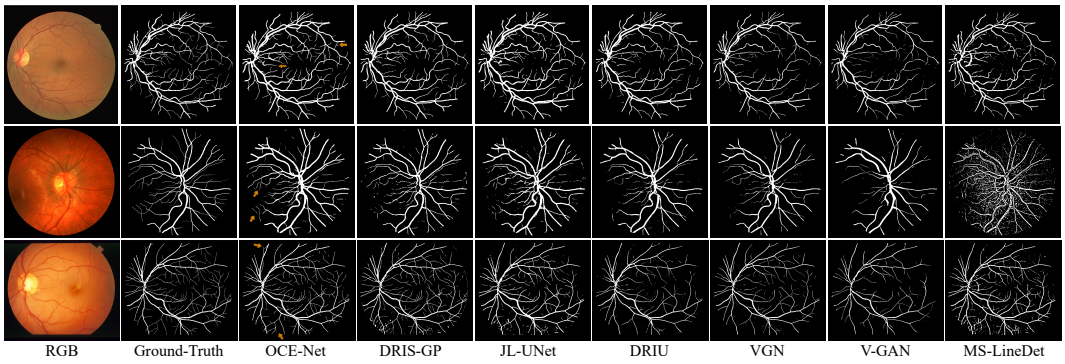

Figure 4.1: Visual comparison with other state-of-the-art methods on DRIVE, CHASEDB1, and STARE datasets, from top to bottom rows. The orange arrows indicate some details of segmentation. Please zoom in for a better view.

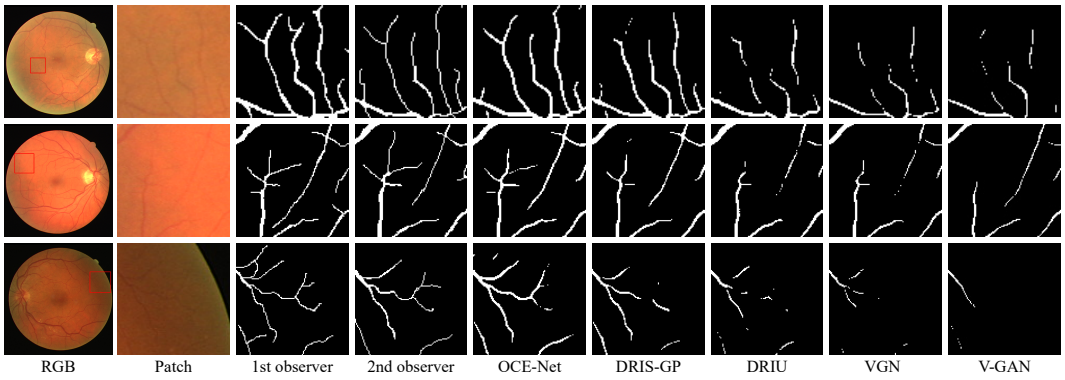

Figure 4.2: Visual comparison with other state-of-the-art methods for segmenting thin vessels.

text represents the overall structure information of blood vessels, while the local details focus more on the thin vessels. Without local attention, thin vessels will be missed easily.

Table 4.6: Comparison between the proposed GLFM and other prevalent modules serving as attention gates on DRIVE. We calculated the P-value of T-test.

Method	F1	Se	Sp	Acc	AUC
Baseline (UNet) [71]	82.11(<0.01)	79.48(<0.01)	97.94(<0.01)	95.59(<0.01)	97.85(<0.01)
+ Attention Gate [61]	82.33(<0.02)	79.12(<0.01)	98.09(<0.01)	95.68(<0.01)	98.02(<0.01)
+ CBAM [93]	82.21(<0.01)	79.26(<0.01)	98.01(<0.01)	95.64(<0.01)	97.92(<0.01)
+ nonlocal [87]	82.57(<0.01)	79.88(<0.01)	98.04(<0.25)	95.73(<0.01)	98.10(<0.01)
+ DNL [104]	82.37(<0.01)	78.99(<0.14)	98.02(<0.01)	95.72(<0.01)	98.10(<0.01)
+ Self-Attention [82]	82.66(<0.01)	79.32(<0.01)	98.12(<0.01)	95.73(<0.05)	98.12(<0.01)
+ **GLFM (Proposed)**	83.00	80.90	97.92	95.76	98.16

Comparison between the proposed OCE-NL/OCE-DNL and other modules serving as core fusion module in MSFM

The proposed OCE-NL and OCE-DNL act as the core modules of the MSFM, which are used to entangle the context and orientation information. To prove the superiority of this entanglement mechanism, we conducted a comparison by directly replacing OCE-NL (OCE-DNL) with other prevalent modules for fair comparison, including vanilla NL, DNL, and self-attention modules. The results shown in Table 4.7 demonstrate that entangling orientation features with the output features of the network can effectively improve the reconstruction of vessels compared with vanilla NL and DNL. This is because orientation features provide more concentration on blood vessels via the oriented constraints of DCOA Conv, which can act as a kind of auxiliary prior information and help the network reconstruct vessels better.

In Table 4.7, 'Addition + DNL' means that the plain and orientation features are added together by an element-wise addition operation and then fed into DNL for fusion. 'Concat + DNL' means that the plain and orientation features are concatenated together, and a Conv1x1 operator is then applied to the features for dimension reduction and then fed into DNL for fusion. The addition, concatenation, and our proposed cross-correlated entanglement can be regarded as three independent methods for feature fusion. Compared with DNL without using orientation prior features, both 'Addition + DNL' and 'Concat + DNL' (both the plain and orientation features involved) show a significant improvement over the 'plain features only' method, which indicates that orientation features can help segment vessels better. Among the 'both plain and orientation features involved' methods, our proposed entanglement is the best way to integrate the plain and orientation features together.

Table 4.7: Comparison between the proposed OCE-NL/OCE-DNL and other modules serving as core fusion block in MSFM on STARE. We calculated the P-value of T-test.

Method	F1	Se	Sp	Acc	AUC
Baseline (UNet) [71]	80.87(<0.01)	74.82(<0.01)	98.78(<0.01)	96.24(<0.01)	98.17(<0.01)
+ Self-Attention [82]	82.22(<0.01)	77.77(<0.01)	98.64(<0.01)	96.43(<0.20)	98.40(<0.01)
+ nonlocal (NL) [87]	81.06(<0.01)	75.55(<0.01)	98.87(<0.01)	96.40(<0.01)	98.46(<0.01)
+ DNL [104]	80.89(<0.01)	75.36(<0.01)	98.96(<0.01)	96.36(<0.01)	98.49(<0.01)
Addition + DNL	82.06(<0.01)	76.21(<0.01)	98.86(<0.01)	96.46(<0.01)	98.51(<0.01)
Concat + DNL	82.32(<0.02)	75.29(<0.01)	98.89(<0.01)	96.54(<0.03)	98.53(<0.01)
+ **OCE-NL (Proposed)**	81.95	75.17	99.02	96.49	98.57
+ **OCE-DNL (Proposed)**	82.37	75.65	99.05	96.56	98.58

Comparison between the proposed DCOA Conv and other convolutions

We compared the proposed DCOA Conv with other prevalent variants of convolution. Note that we directly replaced all the vanilla convolution in the UNet with different convolution variants during testing for fair comparison.

In Table 4.8, 'Gabor Conv (4)/(8)' and 'DCOA Conv (4)/(8)' mean that filters with 4 or 8 different orientations were adopted. 'Dynamic Conv (4)/(8)' means that 4 or 8 kernels were used.

As shown in Table 4.8, the proposed DCOA Conv outperforms other convolution operators when the number of orientation is 8. Note that Gabor Conv has a side effect on vessel segmentation when inserted into the UNet baseline because Gabor Conv can only encode a single orientation per channel, which cannot capture the complex vessels with various orientations. This restriction limits the network to learning the complex orientation characteristics of blood vessels, resulting in poor performance.

Dynamic Conv (4) and Dynamic Conv (8) have similar effects because Dynamic Conv improves the performance by adding more kernels; however, these kernels do not have orientation selectivity. When the number of kernels is 4 in Dynamic Conv, the feature extraction capability of network is at its peak; therefore, increasing the number of kernels to 8 cannot improve the feature extraction capability.

In comparison, the proposed DCOA Conv can capture complex multiple orientations, which succeeds in overcoming the disadvantage of Gabor Conv. Our experiments demon-

strate that eight orientations are enough for our model to work well on the DRIVE dataset, and adding more orientations does not contribute much. In particular, the vascular orientation in the CHASEDB1 dataset is much simpler than that in the DRIVE dataset, with fewer thin vessels and less complexity in orientations; therefore, four orientations can well encode the orientation information for the CHASEDB1 dataset.

Deformable Conv improves the performance by fitting the kernel's shapes to vessels (by learning the varying shapes of vessels). In comparison, our proposed DCOA Conv learns different orientations of vessels by integrating multiple oriented kernels together. These two methods improve the accuracy of vascular segmentation from two different starting points.

Table 4.8: Comparison and abaltion study between the proposed DCOA Conv and other prevalent convolution operators on DRIVE. We calculated the P-value of T-test.

Method	F1	Se	Sp	Acc	AUC
Baseline (UNet) [71]	82.11(<0.01)	79.48(<0.01)	97.94(<0.01)	95.59(<0.01)	97.85(<0.01)
+ Gabor Conv (4) [54]	81.43(<0.02)	78.85(<0.01)	97.89(<0.01)	95.48(<0.01)	97.73(<0.01)
+ Gabor Conv (8) [54]	81.78(<0.01)	77.99(<0.01)	98.12(<0.04)	95.52(<0.01)	97.82(<0.01)
+ DR Conv [9]	82.11(<0.01)	79.29(<0.01)	98.03(<0.01)	95.63(<0.01)	97.93(<0.01)
+ Cond Conv [101]	81.94(<0.01)	80.24(<0.01)	97.86(<0.01)	95.62(<0.01)	97.94(<0.01)
+ Dynamic Conv (4) [10]	82.22(<0.01)	80.15(<0.01)	98.00(<0.03)	95.69(<0.01)	98.08(<0.01)
+ Dynamic Conv (8) [10]	82.31(<0.02)	79.55(<0.01)	98.03(<0.01)	95.69(<0.01)	98.08(<0.01)
+ Deformable Conv V1 [14]	82.42(<0.01)	80.42(<0.01)	97.93(<0.01)	95.70(<0.01)	98.11(<0.01)
+ Deformable Conv V2 [111]	82.26(<0.01)	79.82(<0.01)	98.00(<0.01)	95.69(<0.01)	98.10(<0.01)
+ **DCOA Conv (4)**	82.28	79.67	98.01	95.64	98.03
+ **DCOA Conv (8)**	82.69	80.50	97.97	95.74	98.13

Why must we fuse the plain and the orientation features together via SAFM?

As shown in Fig. 3.7, we fused the plain features extracted by basic blocks and the orientation features extracted by DCOA blocks together in the network via SAFM. Why is this fusion essential? To answer this important question, we conducted experiments to demonstrate that the fusion is essential to involve the orientation information into the network.

As shown in Table 4.9, when the DCOA block and SAFM are removed from the OCE-Net, corresponding to the 'plain only' mode, the orientation information of vessels is

Table 4.9: Experiments conducted on CHASEDB1 for explaining for the necessity of SAFM. The underlined results indicate significant declined performance. We calculated the P-value of T-test.

Method	F1	Se	Sp	Acc	AUC
Plain only	81.42(<0.01)	83.19(<0.01)	97.97(<0.01)	96.56(<0.01)	98.62(<0.01)
Orientation only (DCOA)	81.01(<0.01)	82.74(<0.01)	97.84(<0.01)	96.47(<0.01)	98.48(<0.01)
Fusion by Conv1x1	71.76(<0.01)	60.76(<0.01)	98.96(<0.01)	94.97(<0.01)	98.61(<0.01)
Fusion by SAFM	81.59	82.25	98.11	96.66	98.67

not captured well. When we directly replace the plain conv with the proposed DCOA Conv, corresponding to the 'orientation only' mode, we only use the DCOA Conv to extract orientation features. As we can see in Table 4.9, the results of 'orientation only' mode are significantly lower than those of the 'plain only' mode, which demonstrates that segmenting vessels with only orientation features fails to achieve promising results and even has side effects on the context-aware modules (GLFM, OCE-DNL). This is because except for the orientation features, other features such as the fundus background and other tissues can serve as negative samples against the blood vessels (positive samples) and also play an important role in accurately detecting vessels. Therefore, orientation features should be regarded as a kind of auxiliary prior guidance, used to help the plain features in better reconstruction of vessels.

Considering that there are some redundant channels in both features and there is correlation between the channels of the two features, direct fusion using 1x1 convolution cannot achieve the best performance. Furthermore, when we replace SAFM with Conv1x1 (the most commonly used fusion module) as the fusion module, the segmentation performance is seriously degraded, as indicated by the the underlined results in Table. 4.9. In comparison, adopting SAFM as fusion module in the network instead of Conv1x1 produces a distinct performance improvement.

Comparison between the proposed UARM and the vanilla spatial attention as the refining module

At the end of the network, the proposed UARM is adopted to refine the final output features by dealing with the unbalance problem among the fundus background and the thick and thin vessels. The vanilla spatial attention was used here as the competitor to prove the effectiveness of our UARM. As shown in Table 4.10, the vanilla spatial attention shows an improvement, and the proposed UARM is better than the vanilla spatial attention. In the UNet baseline, the attention of the network focuses more on the thick vessels, which have more salient features, and the thin vessels are usually neglected, resulting in low segmentation accuracy. As shown in Fig. 4.3, UARM can allocate more attention to distinguish thin vessels and obtain a significant promotion.

Table 4.10: Comparison between the proposed UARM and the vanilla spatial attention serving as the final refining module on DRIVE.

Method	F1	Se	Sp	Acc	AUC
Baseline (UNet) [71]	82.11	79.48	97.94	95.59	97.85
+ Spatial Attention [68]	82.19	79.47	98.01	95.66	98.04
+ **UARM (Proposed)**	82.61	80.02	98.03	95.73	98.10

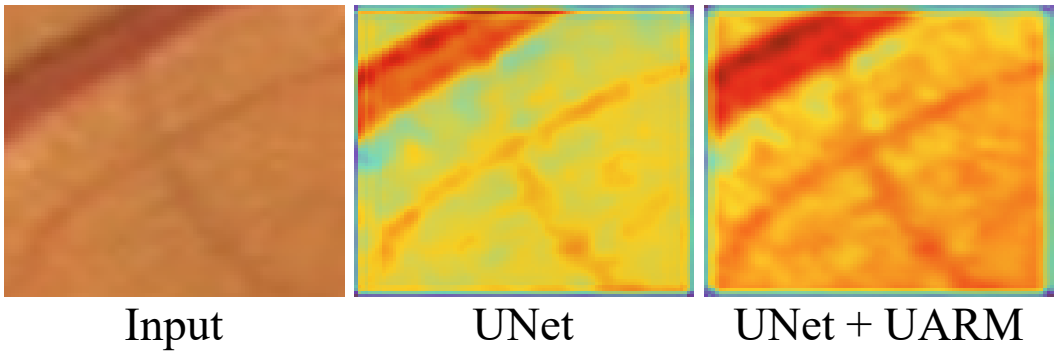

Figure 4.3: Ablation study of UARM in terms of a visual attention map. More attention is paid to the thin vessels.

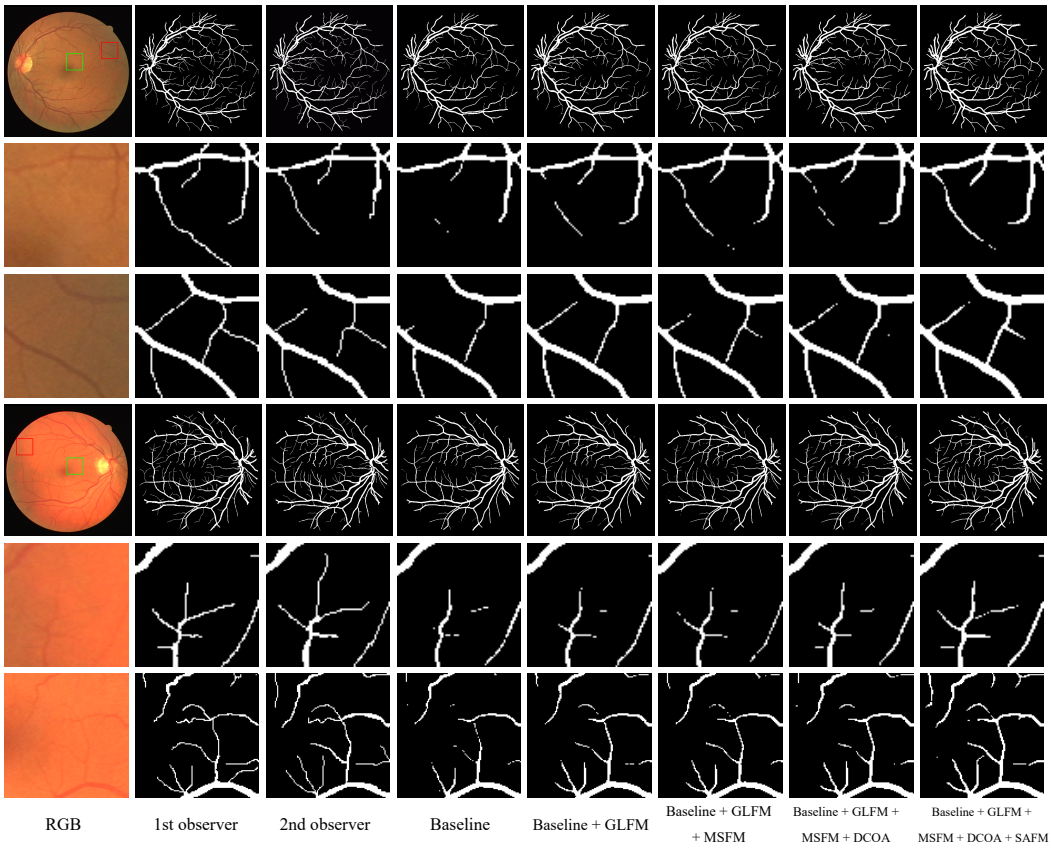

Figure 4.4: Visual ablation study of each proposed module. Notice that in the second-to-last column, when we directly replace plain conv with DCOA Conv, the visual effect is worse. However, as shown in the last column, when we use SAFM to fuse the plain and orientation features, an improvement can be observed. The corresponding quantitative results are listed in Table 4.11.

Table 4.11: Overall ablation study for each proposed module on CHASEDB1. The underlined results indicate significant declined performance.

Method	F1	Se	Sp	Acc	AUC
Baseline (UNet)	79.22	79.10	97.81	96.30	98.22
+ DCOA Conv	80.71	79.38	98.11	96.45	98.47
+ GLFM	80.61	79.93	98.09	96.47	98.48
+ GLFM + OCE-DNL	81.33	82.03	97.98	96.54	98.58
+ GLFM + OCE-DNL + DCOA Conv	81.01	81.24	97.92	96.50	98.48
+ GLFM + OCE-DNL + DCOA Conv + SAFM	81.51	82.17	98.08	96.61	98.63
+ GLFM + OCE-DNL + Deformable Conv	80.31	79.41	98.32	96.51	98.49
+ GLFM + OCE-DNL + Deformable Conv + SAFM	81.29	81.86	98.12	96.60	98.62
+ GLFM + OCE-DNL + DCOA Conv + SAFM + UARM	**81.59**	**82.25**	**98.11**	**96.66**	**98.67**

4.2.4 Overall ablation study for each proposed modules

As shown in Table 4.11, we conducted an overall ablation study for each of the proposed modules on CHASEDB1. Note that directly replacing the plain convolution with DCOA Conv makes the model worse and yields a dramatic performance reduction, as indicated by the underlined results in Table 4.11. Only with SAFM can plain features and orientation features be well fused. Note that the model referred to here is the model equipped with the context-aware modules, such as GLFM and OCE-DNL. Directly replacing the plain conv with the DCOA Conv in the vanilla UNet can actually produce an improvement, which indicates that there is an unexpected conflict between the vessel-aware conv and context-aware modules. This conflict is be discussed in Section 4.2.5.

Note that directly replacing the plain convolution with Deformable Conv [14] [111] unexpectedly makes the performance (of the model with context-aware modules) worse as well. The reason is similar to that of DCOA Conv, mentioned before, because the Deformable Conv reshapes the convolution kernels (DCOA Conv also changes the shape of convolution kernels) to fit the vessels and inevitably neglects other important features like the fundus background and other tissues. These features can serve as important negative samples to help better detect blood vessels. When SAFM is adopted to fuse the plain features and the features extracted by deformable convolution together, a promotion on segmentation can be observed that is similar to that when fusing the plain and orientation features together before. The result emphasizes that it is essential to integrate plain features and the vessel-aware features (shape-aware or orientation-aware) together.

4.2.5 The conflict between the vessel-aware conv and the context-aware modules

We reassert that directly inserting DCOA Conv or Deformable Conv into a vanilla UNet without the context-aware modules (GLFM, OCE-DNL) can produce gain an improvement, but the performance of context-aware modules is be degraded when they work

together with vessel-aware convolution. In other words, there is a conflict between the vessel-aware conv and the context-aware modules. The reasons are as follows. The vessel-aware conv extracts the local features with special orientations (DCOA Conv) or shapes (Deformable Conv); however, other features (background and nonvascular tissues) cannot be extracted and encoded in these vessel-aware features. In contrast, context-aware modules work to capture the global context information by computing all the features in the fundus images, including the vascular and nonvascular features, which is why the 'orientation-only' mode in Table 4.9 shows a relatively worse performance. This demonstrates again that both the plain and the orientation features are equally important. The function of the proposed SAFM is to tackle the conflict and make the vessel-aware conv compatible with those context-aware modules. Our OCE-DNL is also a novel approach to entangling the vessel-aware features into context information.

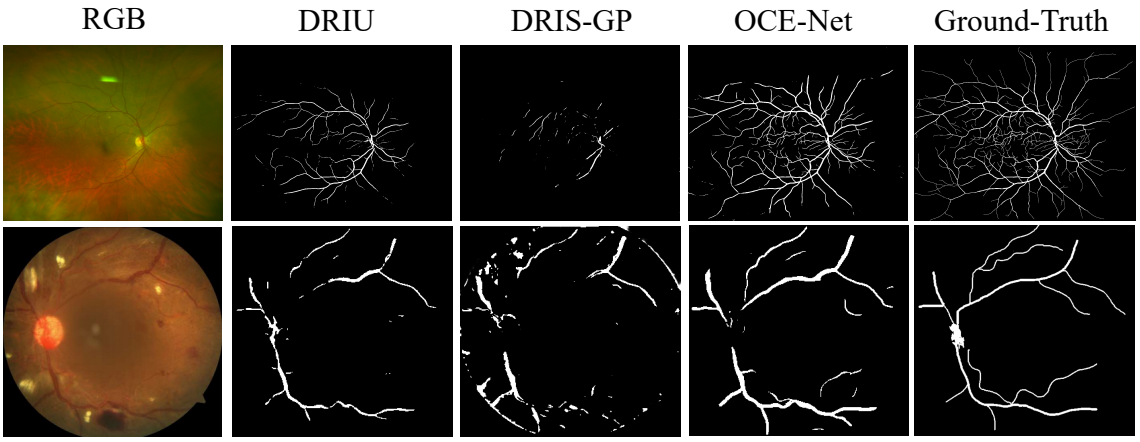

Figure 4.5: Visual comparison on the AV-WIDE (the first row) and UoA-DR (the second row) datasets.

4.2.6 Comparison with other methods on other challenging test sets

To evaluate the generalization and robustness of OCE-Net, we tested our model on several challenging test sets, including AV-WIDE [17], UoA-DR [7], RFMiD [65], and UK Biobank [78]. Note that all the models involved in comparison were trained on DRIVE

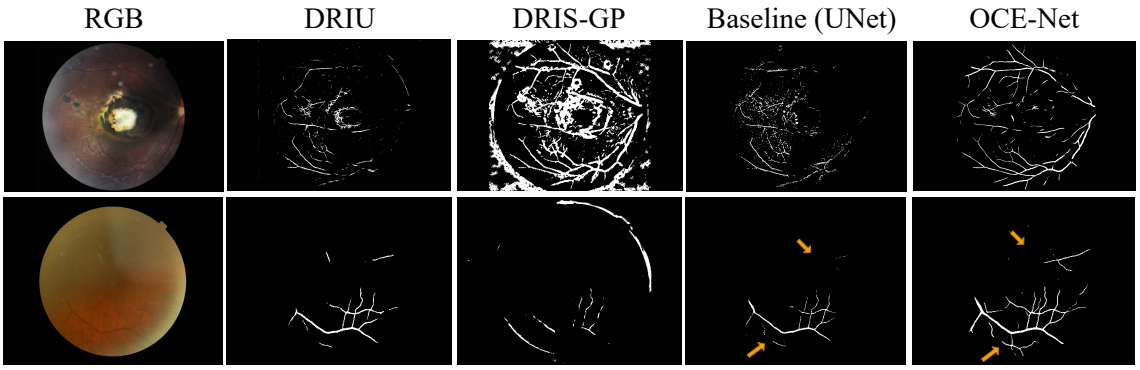

Figure 4.6: Visual comparison on the RFMiD (the first row) and UK Biobank (the second row) datasets.

instead of retraining them from scratch on these challenging datasets. For fair comparison, all the models involved were trained and tested with grayscale fundus images rather than RGB images.

As shown in Fig. 4.5 and 4.6, our method outperforms other methods, including DRIU [55] and DRIS-GP [11], and shows a significant improvement over the baseline UNet [71]. Note that in the test on the UK Biobank [78] in the second row of Fig. 4.6; the orange arrows indicate the area where the thin vessels are severely obscured by the opacity, which are difficult to detect, even with human eyes. In contrast, our method can detect these blurred and occluded vessels well, which demonstrates the strong power of the proposed OCE-Net.

Table 4.12: Cross validation across the DRIVE and STARE datasets.

Test set	Method	Se	Sp	Acc	AUC
DRIVE (trained on STARE)	CC-Net [18]	72.17	98.20	94.86	93.27
	BTS-DSN [28]	72.92	98.15	95.02	97.09
	OCE-Net	75.36	98.62	94.65	97.32
STARE (trained on DRIVE)	CC-Net [18]	74.99	97.98	95.63	96.21
	BTS-DSN [28]	71.88	98.16	95.48	94.86
	OCE-Net	75.06	98.28	95.53	96.23

4.2.7 Cross validation

To test the generalization ability, we conducted cross-validation experiments on DRIVE and STARE and compared our method with other methods. As shown in Table 4.12, compared with other methods, our method achieves promising performance in terms of Se, Sp, and AUC, and comparable performance in terms of Acc. This demonstrates the relatively better generalization ability of the proposed model.

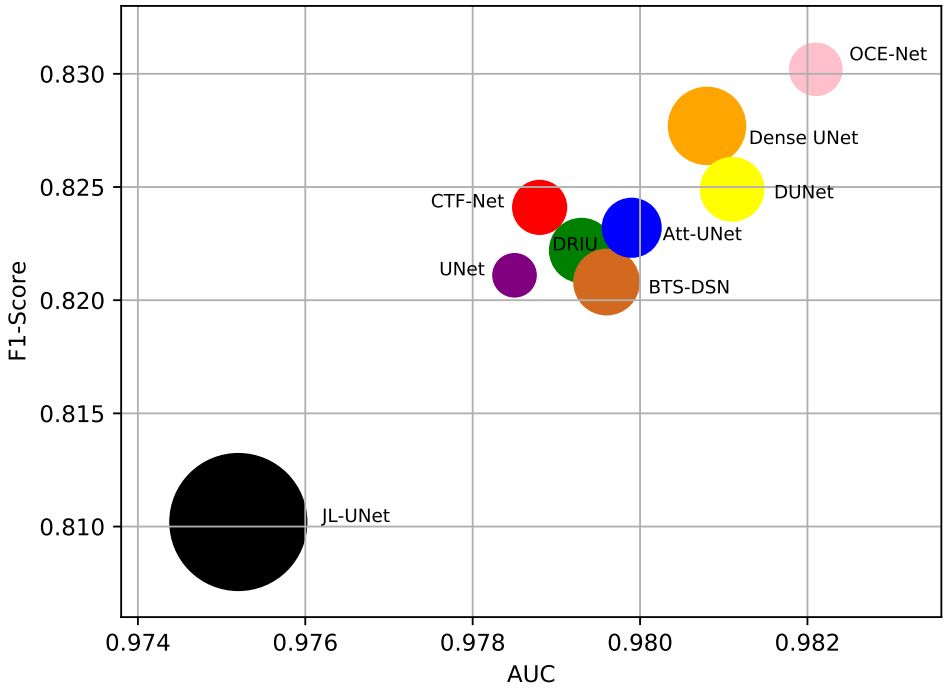

Figure 4.7: Comparison of model parameters. Note that the size of the circle indicates the number of model parameters. F1 score and AUC were used to evaluate the performance of models.

4.2.8 Comparison of Parameters, Flops and Speeds

As shown in Fig. 4.7 and Table 4.13, we compared our OCE-Net with other state-of-the-art methods in terms of model parameters, F1 score, and AUC. Our OCE-Net uses plain

Table 4.13: Comparison of Parameters (Unit: M) and Flops (Unit: G) between different methods on DRIVE.

Method	UNet	Att UNet	Dense UNet	DUNet	OCE-Net
Params (M)	3.4	7.1	11.0	7.4	6.3
Flops (G)	0.07	0.10	0.68	0.23	0.21

Table 4.14: Comparison of the training and inference time between different methods on DRIVE. Note that we only calculate the time cost for one epoch (Unit: Sec/Epoch).

Method	UNet	Att UNet	Dense UNet	DUNet	OCE-Net	w/o GLFM
Training Time	43	91	138	196	186	97
Inference Time	7	17	29	125	83	36

UNet as the backbone. The parameters of the UNet model are about 3.4 MB. Our OCE-Net has about 6.3 MB parameters after adding all the proposed modules (DCOA Conv, GLFM, OCE-DNL and UARM) into the UNet backbone. In contrast, JL-UNet [99] is a large model with about 33 MB parameters; however, it exhibits the worst performance. CTF-Net [86], DRIU [55], Attention UNet [61], and BTS-DSN [28] have similar numbers of parameters (around 7MB) and similar performance. In addition, Dense UNet has about 11 MB parameters, and Deformable UNet (DUNet) has 7.4 MB parameters, but both of them show less worse performance than our OCE-Net. Our OCE-Net achieves the best performance in terms of the F1 score and AUC, without introducing too much parameters.

As shown in Table. 4.14, we compared the training and inference speeds between different UNet-based methods. Note that all the models were trained and tested on the DRIVE dataset with the same parameter settings, including the batch size and the number of cropped samples. UNet is the most lightweight model, which achieves the fastest speed. All the other models are much slower than the backbone UNet. In our model, the most time-consuming module is GLFM because it contains self-attention processing, which requires a lot of computation to calculate global dependencies. When we remove the GLFM from the OCE-Net, there is a significant decline on the time cost.

4.3 Experiments of GCC-UNet

4.3.1 Implementation details

The GCC-UNet model was implemented using the PyTorch framework [66] and trained on a TITAN XP GPU. During the training process, we used the Adam optimizer. Prior to training, the RGB fundus images were converted into grayscale. Then, the images were subjected to random cropping to generate 48×48 patches, and a total of 15,000 patches were generated. A batch size of 32 was used and the total number of training epochs was set to 60.

4.3.2 Overall comparison with other methods

We conducted comprehensive comparison experiments to demonstrate the exceptional performance of our proposed GCC-UNet model. The results, presented in Tables 4.15, 4.16, and 4.17, show that our method outperforms numerous state-of-the-art methods on the DRIVE [77], STARE [31], and CHASEDB1 [19] datasets, based on both traditional and advanced metrics in Table 4.18. These findings highlight the superiority of our approach compared to previous methods. Furthermore, as shown in Figures 4.8 and 4.9, our method also exhibits superior visual performance compared to other methods, particularly on thin vessels. These results provide further evidence of the effectiveness of our approach and its ability to capture global context and improve the continuity of vessels.

4.3.3 Comparison and ablation study of individual module

Comparison and ablation analysis between the proposed GC Conv and plain Capsule Conv

To enhance the vanilla capsule convolution [72], we propose the Graph Capsule Convolution (GC Conv) to model the interdependencies among channels, capsules, and even atoms. In our experiment, we replaced the vanilla convolution operators with both the

Table 4.15: Quantitative comparison with other state-of-the-art methods on **DRIVE**. Red: the best, Blue: the second best.

Method	F1	Se	Sp	Acc	AUC
2nd observer [77]	N.A	77.60	97.24	94.72	N.A
HED [96]	80.89	76.27	98.01	95.24	97.58
DeepVessel [21]	N.A	76.12	97.68	95.23	97.52
Orlando et al. [64]	N.A	78.97	96.84	94.54	95.06
JL-UNet [99]	N.A	76.53	98.18	95.42	97.52
CC-Net [18]	N.A	76.25	98.09	95.28	96.78
Yan et al. [100]	N.A	76.31	98.20	95.33	97.50
BTS-DSN [28]	82.08	78.00	98.06	95.51	97.96
CTF-Net [86]	82.41	78.49	98.13	95.67	97.88
CSU-Net [85]	82.51	80.71	97.82	95.65	98.01
GCC-UNet (Ours)	82.71	80.12	98.16	95.72	98.10

Table 4.16: Quantitative comparison with other methods on **STARE**.

Method	F1	Se	Sp	Acc	AUC
HED [96]	82.68	80.76	98.22	96.41	98.24
Orlando et al. [64]	N.A	76.80	97.38	95.19	95.70
JL-UNet [99]	N.A	75.81	98.46	96.12	98.01
Att UNet [61]	81.36	80.67	98.16	96.32	98.33
CC-Net [18]	N.A	77.09	98.48	96.33	97.00
Yan et al. [100]	N.A	77.35	98.57	96.38	98.33
GCC-UNet (Ours)	82.82	78.06	98.77	96.58	98.56

Table 4.17: Quantitative comparison with other methods on **CHASEDB1**.

Method	F1	Se	Sp	Acc	AUC
2nd observer [77]	N.A	81.05	97.11	95.45	N.A
HED [96]	78.15	75.16	98.05	95.97	97.96
DeepVessel [21]	N.A	74.12	97.01	96.09	97.90
Orlando et al. [64]	N.A	75.65	96.55	94.67	94.78
JL-UNet [99]	N.A	76.33	98.09	96.10	97.81
Yan et al. [100]	N.A	76.41	98.06	96.07	97.76
BTS-DSN [28]	79.83	78.88	98.01	96.27	98.40
GCC-UNet (Ours)	80.86	81.23	98.15	96.59	98.50

Table 4.18: Quantitative comparison with other methods in terms of metrics in [98] on **DRIVE** dataset.

Method	F	C	A	L	rSe	rSp	rAcc	Mcc
2nd observer	83.75	100	93.98	89.06	85.84	99.19	95.74	76.00
HED [96]	80.09	99.75	90.06	89.11	71.57	95.11	89.08	66.00
DRIU [55]	80.43	99.56	91.52	88.23	82.36	96.85	93.13	71.61
DeepVessel [21]	61.74	99.60	84.23	73.38	54.93	99.78	88.32	73.34
V-GAN [76]	84.82	99.64	94.69	89.84	80.77	99.63	94.76	80.24
JL-UNet [99]	81.06	99.61	93.08	87.35	76.11	99.57	93.53	78.98
SWT-FCN [62]	83.92	99.73	94.36	89.11	79.63	99.64	94.48	80.53
DeepDyn [40]	84.53	90.70	94.58	89.61	81.52	99.44	94.82	80.02
DAP [79]	82.55	99.72	93.74	88.24	78.57	99.57	94.15	79.00
DRIS-GP [11]	84.94	99.68	94.91	89.74	80.22	99.64	94.66	81.84
GCC-UNet	85.83	99.75	95.10	90.46	82.60	99.17	95.06	80.27

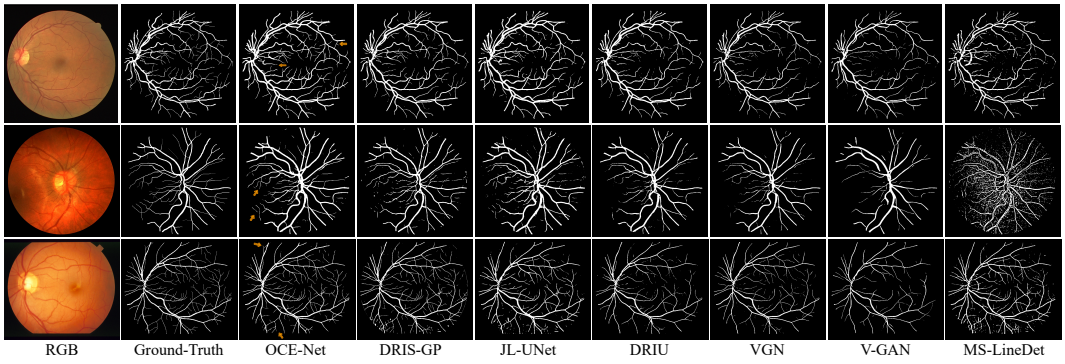

Figure 4.8: Visual comparison with other state-of-the-art methods on DRIVE, CHASEDB1 and STARE datasets.

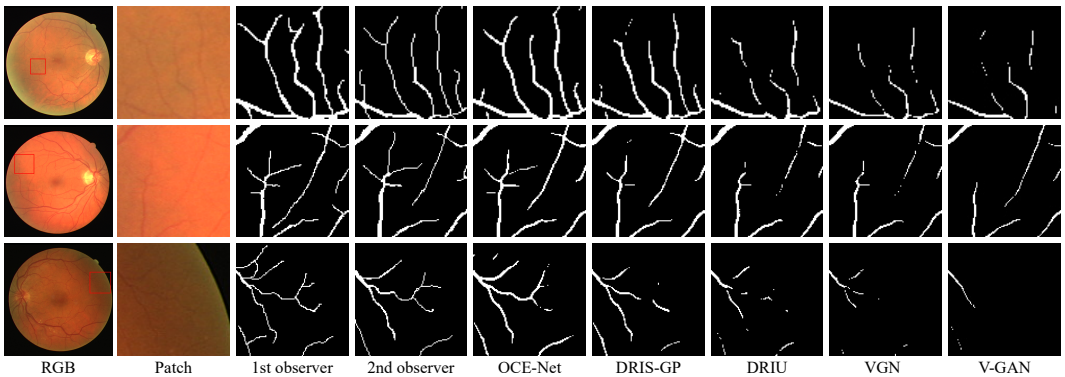

Figure 4.9: Visual comparison with other methods in terms of thin vessels.

Table 4.19: Comaprison between the vanilla Capsule Conv (Cap Conv) in [72] and our Graph Capsule Conv (GC Conv) on DRIVE.

Method	F1	Se	Sp	Acc	AUC
Baseline (UNet) [71]	81.76	78.36	98.03	95.56	97.86
+ Capsule Conv [72]	81.19	78.07	98.12	95.53	97.81
+ **GC Conv (Proposed)**	82.01	78.12	98.18	95.63	97.93

capsule convolution (Cap Conv) [72] and our proposed GC Conv in the U-Net architecture. As shown in Table. 4.19, the performance deteriorated after replacing the vanilla conv with Cap Conv, while our GC Conv yielded significant improvement. This can be attributed to the fact that capsule convolution only captures global features such as relative position, orientation, and color of vessels, without explicitly modeling the relationships among these global characteristics. For instance, capillaries usually have a lighter color and are located at the terminal branches (position), with a more complex orientation. However, our GC Conv can model the relationship between characteristics and implicitly learn the correlations on a graph.

Comparison and ablation analysis between the proposed SGAF and other fusion modules

We performed a comprehensive set of experiments to evaluate the effectiveness of our proposed SGAF module compared to other fusion modules. Table 4.20 shows the performance of fusing local features with different types of global features extracted by the vanilla Capsule Convolution (Cap Conv) with dynamic routing [72] and our proposed Graph Capsule Convolution (GC Conv) with graph dynamic routing. In addition to SGAF, we also evaluated vanilla Conv1x1 [74] and Selective Kernel Attention (SK) [52] as fusion modules.

As seen in Table 4.20, Conv1x1 was not effective in fusing local and global features, as it could not differentiate useful channels in these two types of features. SK Attention, on the other hand, was effective in fusing features extracted using different mechanisms,

achieving promising performance. However, our SGAF outperformed SK with a significant margin. Furthermore, our proposed GC Conv significantly outperformed Cap Conv in terms of extracting global contextual features with the same fusion module.

We also evaluated different modes, including the series mode and parallel mode shown in Fig. 4.10, for the combination of CNN Conv and Capsule Conv. Our experiments showed that the series mode performed better than the parallel mode. As Cap Conv and our GC Conv cannot directly extract global information from the raw image, the optimal approach is to first extract features using vanilla CNN convolutions, then use capsule convolutions to further extract global contextual information from the CNN features, and finally fuse the local and global features through skip connections.

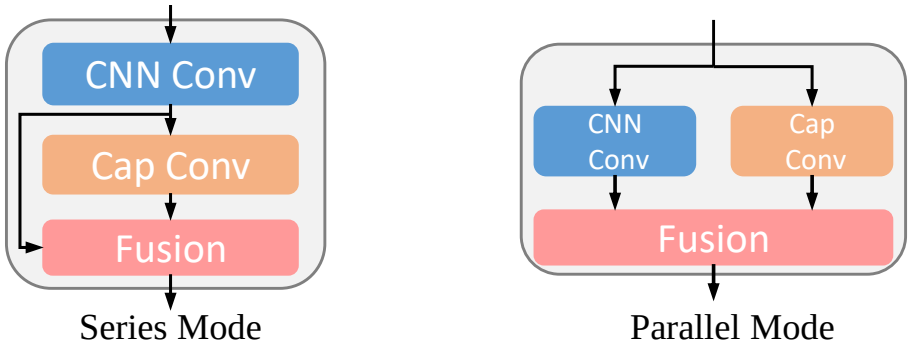

Figure 4.10: Different modes for the combination of CNN Conv and Capsule Conv. The experiment shows that the series model can achieve better performance.

Comparison and ablation analysis between our proposed BGA and other attention modules in the bottleneck

We conduct experiments for comparing our proposed Bottleneck Graph Attention module with other well-known attention modules. As shown in Table. 4.21, our BGA outperforms many other attention module, such as SE [32], CBAM [93], Non-Local [87], and Self-Attention [82]. The proposed Channel Graph Attention (CGA) and Spatial Attention (SGA) can also achieve promising performance. Because our BGA can model the relationships among channels via CGA by constrcting and learning graph representation. In

Table 4.20: Comaprison between our SGAF and other fusion modules (Conv1x1 [74], SK Attention [52]) on DRIVE.

Method	F1	Se	Sp	Acc	AUC
Baseline (UNet) [71]	81.76	78.36	98.03	95.56	97.86
Baseline (Capsule UNet) [72]	81.19	78.07	98.12	95.53	97.81
CAPSULE / FUSION	F1	Se	Sp	Acc	AUC
Cap Conv [72] / Conv1x1 [74]	81.42	77.96	97.79	95.52	97.82
Cap Conv [72] / SK [52]	82.12	78.16	97.76	95.65	98.01
Cap Conv [72] / **SGAF**	82.30	78.98	98.18	95.67	98.04
GC Conv / Conv1x1 [74]	81.82	78.53	97.92	95.59	97.87
GC Conv / SK [52]	82.23	78.86	97.91	95.68	98.03
GC Conv / SGAF (Parallel)	81.75	79.36	98.05	95.64	97.99
GC Conv / SGAF (Series)	82.42	79.45	98.11	95.70	98.07

addition, our BGA can leverage SGA to split the vessel out of the background and construct graph to improve the continuity of vessels by learning the connectivity among the nodes of vessels.

Table 4.21: Comaprison and ablation study of the proposed BGA and other attention modules on DRIVE.

Method	F1	Se	Sp	Acc	AUC
Baseline (UNet) [71]	81.76	78.36	98.03	95.56	97.86
+ SE [32]	81.81	79.03	97.77	95.60	97.90
+ CBAM [93]	81.06	78.85	97.87	96.61	97.89
+ Non-Local [87]	81.75	78.98	97.76	95.61	97.91
+ Self-Attention [82]	82.03	79.45	97.96	95.64	97.93
+ CGA (Proposed)	82.18	79.32	98.00	95.64	97.93
+ SGA (Proposed)	82.11	79.89	97.95	95.64	97.93
+ BGA (Proposed)	82.25	79.65	98.05	95.67	97.94

Comparison and ablation analysis between our proposed MSGF and other multi-scale fusion modules

We performed experiments to evaluate the performance of our proposed Multi-Scale Graph Fusion (MSGF) module in comparison to other multi-scale fusion modules, includ-

ing vanilla Conv1x1 and the fusion module proposed in [94]. Additionally, we compared the different modes of our MSGF module, namely Individual (applying three individual graph convolutions to three different scales of input graphs), Concat (concatenating the graphs of three input features and feeding it into a single graph convolution), and Shared (feeding the graphs of input features into a shared graph convolution).

As shown in Table. 4.22, all three proposed modes of MSGF outperformed other fusion modules. Among these three modes, the Shared mode achieved the best performance with fewer parameters and computational costs. This is because feeding the constructed graphs from three different scales into a single graph convolution allows the convolution operator to learn all the information and features of the three graphs at once. Moreover, since these different scales of features come from the same fundus image, we assume that they share the same graph pattern. Applying a single graph convolution aligns the graph representation among these features, and the graphs of these three scales can achieve the effect of information complementation on one shared graph convolution.

Table 4.22: Ablation study of the proposed MSGF on DRIVE.

Method	F1	Se	Sp	Acc	AUC
Baseline (UNet) [71]	81.76	78.36	98.03	95.56	97.86
+ Fusion via Conv1x1 [74]	81.69	78.88	97.89	96.59	97.89
+ Fusion module in [94]	81.86	78.54	97.95	96.62	97.91
+ MSGF (Individual)	82.03	78.84	98.02	96.64	97.93
+ MSGF (Concat)	81.95	79.14	97.98	96.64	97.93
+ MSGF (Shared)	82.15	79.23	98.08	95.68	97.94

4.3.4 Comparison study on challenging test sets

To assess the generalization ability of our GCC-UNet model, we conducted experiments on challenging datasets, including AV-WIDE [17], UoA-DR [7], and UK Biobank [78]. All the models used for comparison were trained from scratch on the DRIVE dataset. Our experimental results, presented in Fig. 4.11 and 4.12, demonstrate that GCC-UNet outper-

forms the state-of-the-art methods DRIU and DRIS-GP, and significantly improves upon the baseline UNet. In the UK Biobank test, as shown in the second row of Fig. 4.12, orange arrows indicate regions where thin vessels are obscured by opacities, making them challenging to detect even for human experts. However, our GCC-UNet model successfully detected these blurry and occluded vessels, indicating its superior performance.

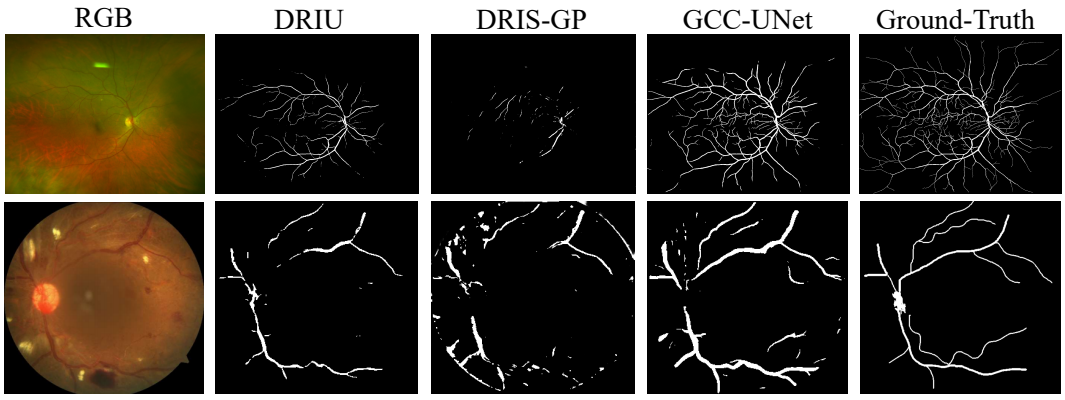

Figure 4.11: Visual comparison on the AV-WIDE and UoA-DR datasets.

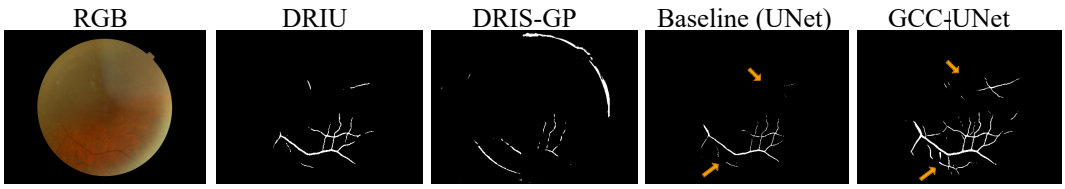

Figure 4.12: Visual comparison on the UK Biobank dataset.

Table 4.23: Cross-training validation on DRIVE and STARE.

Test set	Method	Se	Sp	Acc	AUC
DRIVE (Trained on STARE)	CC-Net [18]	72.17	98.20	94.86	93.27
	GCC-UNet	71.06	97.96	94.92	97.03
STARE (Trained on DRIVE)	CC-Net [18]	74.99	97.98	95.63	96.21
	GCC-UNet	75.42	98.23	95.53	96.98

4.3.5 Cross validation

To evaluate the generalization ability of our method, we conducted cross-validation experiments on two different datasets, DRIVE and STARE, and compared our results with other state-of-the-art methods. Our method achieved competitive performance, as shown in Table. 4.23. However, we also acknowledge that our method has a drawback, namely a relatively lower accuracy compared to some other methods.

Chapter 5

Conclusion and Summary

5.1 Discussion of OCE-Net

We quantitatively and qualitatively compared OCE-Net with many other state-of-the-art methods. We reproduced the network models of many UNet-based methods and re-trained them from scratch on our codes, such as Attention UNet and Dense UNet. Some methods provide original codes and segmentation results, such as JL-UNet and SkelCons, so we used their segmentation results for the comparison. However, some methods did not provide codes and results, such as DeepVessel, CC-Net, BTS-DSN, and CSU-Net, so we could only compare our methods with the data in their papers. In general, we believe that the comparison results proved the advantage of our work. However, our method also had certain limitations. For example, when fundus images contained various lesions, the continuity and connectivity of the extracted vessels were poor. In addition, when the lesion area contained many tissues whose textures were similar to those of the thin vessels, our method tended to detect these tissues as continuous intact vessels while attempting to detect all the potential blood vessels of various orientations. As an important future work, we plan to build a graph model to perceive the entire vascular skeleton to better capture the context information and improve vascular connectivity while suppressing the nonvascular tissues as much as possible.

5.2 Discussion of GCC-UNet

Our proposed GCC-UNet has demonstrated promising performance on retinal vessel segmentation by effectively capturing global context, local and global fusion, and improving the continuity of vessels. Notably, our model has a relatively small number of parameters (only 5.48M). However, there are still inherent weaknesses in our approach. As pointed out in [72], the introduction of capsule convolution enables the capturing of global context, but it significantly increases the computational cost of the model, resulting in slower inference speed. This issue is intrinsic to the capsule convolution itself. Although our GC Conv significantly improves the effect of capsule convolution, it does not reduce the computational cost or increase the inference speed. In the future, researchers could focus on developing methods to improve the inference speed of capsule convolution. Additionally, transformers [82] could be a promising complement to our model, given their ability to capture long-range global context.

5.3 Conclusion

In this paper, a novel model called OCE-Net was proposed to simultaneously capture the orientation and context information of blood vessels as well as entangle them together. To this end, a novel convolution operator was designed to fit the vessels with multiple orientations, and the experimental results indicate that feature extraction along more specific orientations can improve vascular continuity and connectivity. In addition, a global and local fusion module was constructed to leverage both of the context and detail information of vessels because context information could help the network perceive the whole vascular skeleton and deal with occlusion well. Moreover, a novel entanglement mechanism was developed to entangle the context and the orientation information by introducing cross correlation into the vanilla nonlocal. Finally, to deal with the unbalance among the fundus background and thick and thin vessels, a novel attention module was proposed to refine the results by allocating more attention to the regions where the ves-

sels had low discriminability. Thus, the proposed framework could effectively carry out retinal vessel segmentation.

In addition, in this paper, we also propose a novel approach for retinal vessel segmentation using a global and local fusion UNet, which integrates vanilla, graph, and capsule convolutions. Our approach represents the first attempt to unify these different convolution types. Specifically, we use capsule convolution to capture global contextual information, and graph convolution to model vessel connectivity and improve continuity. Our GC Conv enhances vanilla capsule convolution, while our SGAF can fuse features from various domains (CNN, Graph, and Capsule). Additionally, our BGA improves vessel continuity using a divide-and-conquer strategy, and our MSGF can handle multi-scale feature fusion. Importantly, these modules can be applied to various applications beyond vessel segmentation, including MRI tumor segmentation, geometric modeling of medical images, and even semantic or instance segmentation.

Chapter 6

Copyright

Signature: [Xinxu Wei]

Date: 2023.4.7

Bibliography

[1] AKBAR, S., SHARIF, M., AKRAM, M. U., SABA, T., MAHMOOD, T., AND KOLIVAND, M. Automated techniques for blood vessels segmentation through fundus retinal images: A review. *Microscopy research and technique 82*, 2 (2019), 153–170.

[2] AZZOPARDI, G., STRISCIUGLIO, N., VENTO, M., AND PETKOV, N. Trainable cosfire filters for vessel delineation with application to retinal images. *Medical Image Analysis 19*, 1 (2015), 46–57.

[3] BADAR, M., HARIS, M., AND FATIMA, A. Application of deep learning for retinal image analysis: A review. *Computer Science Review 35* (2020), 100203.

[4] BELLO, I., ZOPH, B., VASWANI, A., SHLENS, J., AND LE, Q. V. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 3286–3295.

[5] BUADES, A., COLL, B., AND MOREL, J.-M. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), vol. 2, IEEE, pp. 60–65.

[6] CAO, Y., XU, J., LIN, S., WEI, F., AND HU, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (2019), pp. 0–0.

[7] CHALAKKAL, R. J., ABDULLA, W. H., AND SINUMOL, S. Comparative analysis of university of auckland diabetic retinopathy database. In *Proceedings of the 9th International Conference on Signal Processing Systems* (2017), pp. 235–239.

[8] CHAUDHURI, S., CHATTERJEE, S., KATZ, N., NELSON, M., AND GOLDBAUM, M. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on Medical Imaging 8*, 3 (1989), 263–269.

[9] CHEN, J., WANG, X., GUO, Z., ZHANG, X., AND SUN, J. Dynamic region-aware convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 8064–8073.

[10] CHEN, Y., DAI, X., LIU, M., CHEN, D., YUAN, L., AND LIU, Z. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 11030–11039.

[11] CHERUKURI, V., BG, V. K., BALA, R., AND MONGA, V. Deep retinal image segmentation with regularization under geometric priors. *IEEE Transactions on Image Processing 29* (2019), 2552–2567.

[12] CHOI, J., SEO, H., IM, S., AND KANG, M. Attention routing between capsules. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (2019), pp. 0–0.

[13] CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 1251–1258.

[14] DAI, J., QI, H., XIONG, Y., LI, Y., ZHANG, G., HU, H., AND WEI, Y. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 764–773.

[15] DOS SANTOS FERREIRA, M. V., DE CARVALHO FILHO, A. O., DE SOUSA, A. D., SILVA, A. C., AND GATTASS, M. Convolutional neural network and texture descriptor-based automatic detection and diagnosis of glaucoma. *Expert Systems with Applications 110* (2018), 250–263.

[16] DU, Y., ZHAO, X., HE, M., AND GUO, W. A novel capsule based hybrid neural network for sentiment classification. *IEEE Access 7* (2019), 39321–39328.

[17] ESTRADA, R., ALLINGHAM, M. J., METTU, P. S., COUSINS, S. W., TOMASI, C., AND FARSIU, S. Retinal artery-vein classification via topology estimation. *IEEE Transactions on Medical Imaging 34*, 12 (2015), 2518–2534.

[18] FENG, S., ZHUO, Z., PAN, D., AND TIAN, Q. Ccnet: A cross-connected convolutional network for segmenting retinal vessels using multi-scale features. *Neurocomputing 392* (2020), 268–276.

[19] FRAZ, M. M., REMAGNINO, P., HOPPE, A., UYYANONVARA, B., RUDNICKA, A. R., OWEN, C. G., AND BARMAN, S. A. An ensemble classification-based approach applied to retinal blood vessel segmentation. *IEEE Transactions on Biomedical Engineering 59*, 9 (2012), 2538–2548.

[20] FU, H., CHENG, J., XU, Y., WONG, D. W. K., LIU, J., AND CAO, X. Joint optic disc and cup segmentation based on multi-label deep network and polar transformation. *IEEE Transactions on Medical Imaging 37*, 7 (2018), 1597–1605.

[21] FU, H., XU, Y., LIN, S., KEE WONG, D. W., AND LIU, J. Deepvessel: Retinal vessel segmentation via deep learning and conditional random field. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2016), Springer, pp. 132–139.

[22] FU, J., LIU, J., TIAN, H., LI, Y., BAO, Y., FANG, Z., AND LU, H. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 3146–3154.

[23] FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics 36*, 4 (1980), 193–202.

[24] GEGÚNDEZ-ARIAS, M. E., AQUINO, A., BRAVO, J. M., AND MARÍN, D. A function for quality evaluation of retinal vessel segmentations. *IEEE Transactions on Medical Imaging 31*, 2 (2011), 231–239.

[25] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A., AND BENGIO, Y. Generative adversarial networks. *Communications of the ACM 63*, 11 (2020), 139–144.

[26] GUO, C., SZEMENYEI, M., YI, Y., WANG, W., CHEN, B., AND FAN, C. Sa-unet: Spatial attention u-net for retinal vessel segmentation. In *2020 25th international Conference on Pattern Recognition (ICPR)* (2021), IEEE, pp. 1236–1242.

[27] GUO, M.-H., XU, T.-X., LIU, J.-J., LIU, Z.-N., JIANG, P.-T., MU, T.-J., ZHANG, S.-H., MARTIN, R. R., CHENG, M.-M., AND HU, S.-M. Attention mechanisms in computer vision: A survey. *Computational Visual Media 8*, 3 (2022), 331–368.

[28] GUO, S., WANG, K., KANG, H., ZHANG, Y., GAO, Y., AND LI, T. Bts-dsn: Deeply supervised neural network with short connections for retinal vessel segmentation. *International journal of medical informatics 126* (2019), 105–113.

[29] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.

[30] HINTON, G. E., SABOUR, S., AND FROSST, N. Matrix capsules with em routing. In *International Conference on Learning Representations* (2018).

[31] HOOVER, A., KOUZNETSOVA, V., AND GOLDBAUM, M. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging 19*, 3 (2000), 203–210.

[32] HU, J., SHEN, L., AND SUN, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 7132–7141.

[33] HUANG, Z., WANG, X., HUANG, L., HUANG, C., WEI, Y., AND LIU, W. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 603–612.

[34] IOFFE, S., AND SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning* (2015), PMLR, pp. 448–456.

[35] ITTI, L., AND KOCH, C. Computational modelling of visual attention. *Nature Reviews Neuroscience 2*, 3 (2001), 194–203.

[36] JAIN, A. K., MAO, J., AND MOHIUDDIN, K. M. Artificial neural networks: A tutorial. *Computer 29*, 3 (1996), 31–44.

[37] JEON, Y., AND KIM, J. Active convolution: Learning the shape of convolution for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 4201–4209.

[38] JIN, Q., MENG, Z., PHAM, T. D., CHEN, Q., WEI, L., AND SU, R. Dunet: A deformable network for retinal vessel segmentation. *Knowledge-Based Systems 178* (2019), 149–162.

[39] KHAN, K. B., SIDDIQUE, M. S., AHMAD, M., AND MAZZARA, M. A hybrid unsupervised approach for retinal vessel segmentation. *BioMed Research International 2020* (2020).

[40] KHANAL, A., AND ESTRADA, R. Dynamic deep networks for retinal vessel segmentation. *Frontiers in Computer Science* (2020), 35.

[41] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980* (2014).

[42] KIPF, T. N., AND WELLING, M. Semi-supervised classification with graph convolutional networks. *ArXiv Preprint ArXiv:1609.02907* (2016).

[43] KOU, C., LI, W., YU, Z., AND YUAN, L. An enhanced residual u-net for microaneurysms and exudates segmentation in fundus images. *IEEE Access 8* (2020), 185514–185525.

[44] LALONDE, R., KHOSRAVAN, N., AND BAGCI, U. Deformable capsules for object detection. *ArXiv Preprint ArXiv:2104.05031* (2021).

[45] LALONDE, R., XU, Z., IRMAKCI, I., JAIN, S., AND BAGCI, U. Capsules for biomedical image segmentation. *Medical Image Analysis 68* (2021), 101889.

[46] LECUN, Y., BENGIO, Y., AND HINTON, G. Deep learning. *nature 521*, 7553 (2015), 436–444.

[47] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[48] LI, Q., FENG, B., XIE, L., LIANG, P., ZHANG, H., AND WANG, T. A cross-modality learning approach for vessel segmentation in retinal images. *IEEE Transactions on Medical Imaging 35*, 1 (2015), 109–118.

[49] LI, R., LI, M., LI, J., AND ZHOU, Y. Connection sensitive attention u-net for accurate retinal vessel segmentation. *ArXiv Preprint ArXiv:1903.05558* (2019).

[50] LI, T., BO, W., HU, C., KANG, H., LIU, H., WANG, K., AND FU, H. Applications of deep learning in fundus images: A review. *Medical Image Analysis 69* (2021), 101971.

[51] LI, X., CHEN, H., QI, X., DOU, Q., FU, C.-W., AND HENG, P.-A. H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE Transactions on Medical Imaging 37*, 12 (2018), 2663–2674.

[52] LI, X., WANG, W., HU, X., AND YANG, J. Selective kernel networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 510–519.

[53] LISKOWSKI, P., AND KRAWIEC, K. Segmenting retinal blood vessels with deep neural networks. *IEEE Transactions on Medical Imaging 35*, 11 (2016), 2369–2380.

[54] LUAN, S., CHEN, C., ZHANG, B., HAN, J., AND LIU, J. Gabor convolutional networks. *IEEE Transactions on Image Processing 27*, 9 (2018), 4357–4366.

[55] MANINIS, K.-K., PONT-TUSET, J., ARBELÁEZ, P., AND GOOL, L. V. Deep retinal image understanding. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2016), Springer, pp. 140–148.

[56] MAZZIA, V., SALVETTI, F., AND CHIABERGE, M. Efficient-capsnet: Capsule network with self-attention routing. *Scientific Reports 11*, 1 (2021), 14634.

[57] MENDONCA, A. M., AND CAMPILHO, A. Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *IEEE Transactions on Medical Imaging 25*, 9 (2006), 1200–1213.

[58] MENG, Y., ZHANG, H., GAO, D., ZHAO, Y., YANG, X., QIAN, X., HUANG, X., AND ZHENG, Y. Bi-gcn: boundary-aware input-dependent graph convolution network for biomedical image segmentation. *ArXiv Preprint ArXiv:2110.14775* (2021).

[59] NAIR, V., AND HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international Conference on Machine Learning (ICML-10)* (2010), pp. 807–814.

[60] NGUYEN, U. T., BHUIYAN, A., PARK, L. A., AND RAMAMOHANARAO, K. An effective retinal blood vessel segmentation method using multi-scale line detection. *Pattern Recognition 46*, 3 (2013), 703–715.

[61] OKTAY, O., SCHLEMPER, J., FOLGOC, L. L., LEE, M., HEINRICH, M., MISAWA, K., MORI, K., MCDONAGH, S., HAMMERLA, N. Y., KAINZ, B., ET AL. Attention u-net: Learning where to look for the pancreas. *ArXiv Preprint ArXiv:1804.03999* (2018).

[62] OLIVEIRA, A., PEREIRA, S., AND SILVA, C. A. Retinal vessel segmentation based on fully convolutional neural networks. *Expert Systems with Applications 112* (2018), 229–242.

[63] ORLANDO, J. I., AND BLASCHKO, M. Learning fully-connected crfs for blood vessel segmentation in retinal images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2014), Springer, pp. 634–641.

[64] ORLANDO, J. I., PROKOFYEVA, E., AND BLASCHKO, M. B. A discriminatively trained fully connected conditional random field model for blood vessel segmentation in fundus images. *IEEE Transactions on Biomedical Engineering 64*, 1 (2016), 16–27.

[65] PACHADE, S., PORWAL, P., THULKAR, D., KOKARE, M., DESHMUKH, G., SAHASRABUDDHE, V., GIANCARDO, L., QUELLEC, G., AND MÉRIAUDEAU, F. Retinal fundus multi-disease image dataset (rfmid): a dataset for multi-disease detection research. *Data 6*, 2 (2021), 14.

[66] PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., ET AL. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems 32* (2019).

[67] PIZER, S. M., AMBURN, E. P., AUSTIN, J. D., CROMARTIE, R., GESELOWITZ, A., GREER, T., TER HAAR ROMENY, B., ZIMMERMAN, J. B., AND ZUIDERVELD, K.

Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing 39*, 3 (1987), 355–368.

[68] QIN, X., WANG, Z., BAI, Y., XIE, X., AND JIA, H. Ffa-net: Feature fusion attention network for single image dehazing. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 11908–11915.

[69] RAMACHANDRAN, P., PARMAR, N., VASWANI, A., BELLO, I., LEVSKAYA, A., AND SHLENS, J. Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems 32* (2019).

[70] RICCI, E., AND PERFETTI, R. Retinal blood vessel segmentation using line operators and support vector classification. *IEEE Transactions on Medical Imaging 26*, 10 (2007), 1357–1365.

[71] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2015), Springer, pp. 234–241.

[72] SABOUR, S., FROSST, N., AND HINTON, G. E. Dynamic routing between capsules. *Advances in Neural Information Processing Systems 30* (2017).

[73] SHIN, S. Y., LEE, S., YUN, I. D., AND LEE, K. M. Deep vessel segmentation by learning graphical connectivity. *Medical Image Analysis 58* (2019), 101556.

[74] SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556* (2014).

[75] SOARES, J. V., LEANDRO, J. J., CESAR, R. M., JELINEK, H. F., AND CREE, M. J. Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification. *IEEE Transactions on Medical Imaging 25*, 9 (2006), 1214–1222.

[76] SON, J., PARK, S. J., AND JUNG, K.-H. Retinal vessel segmentation in fundoscopic images with generative adversarial networks. *ArXiv Preprint ArXiv:1706.09318* (2017).

[77] STAAL, J., ABRÀMOFF, M. D., NIEMEIJER, M., VIERGEVER, M. A., AND VAN GINNEKEN, B. Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging 23*, 4 (2004), 501–509.

[78] SUDLOW, C., GALLACHER, J., ALLEN, N., BERAL, V., BURTON, P., DANESH, J., DOWNEY, P., ELLIOTT, P., GREEN, J., LANDRAY, M., ET AL. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Medicine 12*, 3 (2015), e1001779.

[79] SUN, X., FANG, H., YANG, Y., ZHU, D., WANG, L., LIU, J., AND XU, Y. Robust retinal vessel segmentation from a data augmentation perspective. In *International Workshop on Ophthalmic Medical Image Analysis* (2021), Springer, pp. 189–198.

[80] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1–9.

[81] TAN, Y., YANG, K.-F., ZHAO, S.-X., AND LI, Y.-J. Retinal vessel segmentation with skeletal prior and contrastive loss. *IEEE Transactions on Medical Imaging* (2022).

[82] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. Attention is all you need. *Advances in Neural Information Processing Systems 30* (2017).

[83] VELIČKOVIĆ, P., CUCURULL, G., CASANOVA, A., ROMERO, A., LIO, P., AND BENGIO, Y. Graph attention networks. *ArXiv Preprint ArXiv:1710.10903* (2017).

[84] VERMA, S., AND ZHANG, Z.-L. Graph capsule convolutional neural networks. *ArXiv Preprint ArXiv:1805.08090* (2018).

[85] WANG, B., WANG, S., QIU, S., WEI, W., WANG, H., AND HE, H. Csu-net: A context spatial u-net for accurate blood vessel segmentation in fundus images. *IEEE Journal of Biomedical and Health Informatics 25*, 4 (2020), 1128–1138.

[86] WANG, K., ZHANG, X., HUANG, S., WANG, Q., AND CHEN, F. Ctf-net: Retinal vessel segmentation via deep coarse-to-fine supervision network. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)* (2020), IEEE, pp. 1237–1241.

[87] WANG, X., GIRSHICK, R., GUPTA, A., AND HE, K. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 7794–7803.

[88] WANG, X., ZHU, M., BO, D., CUI, P., SHI, C., AND PEI, J. Am-gcn: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), pp. 1243–1253.

[89] WEI, X., NIU, X., ZHANG, X., AND LI, Y. Deep pneumonia: Attention-based contrastive learning for class-imbalanced pneumonia lesion recognition in chest x-rays. In *2022 IEEE International Conference on Big Data (Big Data)* (2022), IEEE, pp. 5361–5369.

[90] WEI, X., YANG, K., BZDOK, D., AND LI, Y. Orientation and context entangled network for retinal vessel segmentation. *Expert Systems with Applications 217* (2023), 119443.

[91] WEI, X., ZHANG, X., AND LI, Y. Sarn: A lightweight stacked attention residual network for low-light image enhancement. In *2021 6th International Conference on Robotics and Automation Engineering (ICRAE)* (2021), IEEE, pp. 275–279.

[92] WEI, X., ZHANG, X., AND LI, Y. Tsn-ca: A two-stage network with channel attention for low-light image enhancement. In *Artificial Neural Networks and Machine Learning–ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, September 6–9, 2022, Proceedings, Part III* (2022), Springer, pp. 286–298.

[93] WOO, S., PARK, J., LEE, J.-Y., AND KWEON, I. S. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on computer vision (ECCV)* (2018), pp. 3–19.

[94] WU, H., LIU, J., WANG, W., WEN, Z., AND QIN, J. Region-aware global context modeling for automatic nerve segmentation from ultrasound images. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 2907–2915.

[95] WU, Z., PAN, S., CHEN, F., LONG, G., ZHANG, C., AND PHILIP, S. Y. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems 32*, 1 (2020), 4–24.

[96] XIE, S., AND TU, Z. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1395–1403.

[97] XINYI, Z., AND CHEN, L. Capsule graph neural network. In *International Conference on Learning Representations* (2019).

[98] YAN, Z., YANG, X., AND CHENG, K.-T. A skeletal similarity metric for quality evaluation of retinal vessel segmentation. *IEEE Transactions on Medical Imaging 37*, 4 (2017), 1045–1057.

[99] YAN, Z., YANG, X., AND CHENG, K.-T. Joint segment-level and pixel-wise losses for deep learning based retinal vessel segmentation. *IEEE Transactions on Biomedical Engineering 65*, 9 (2018), 1912–1923.

[100] YAN, Z., YANG, X., AND CHENG, K.-T. A three-stage deep learning model for accurate retinal vessel segmentation. *IEEE journal of Biomedical and Health Informatics 23*, 4 (2018), 1427–1436.

[101] YANG, B., BENDER, G., LE, Q. V., AND NGIAM, J. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in Neural Information Processing Systems 32* (2019).

[102] YAO, L., MAO, C., AND LUO, Y. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33, pp. 7370–7377.

[103] YE, J., HE, J., PENG, X., WU, W., AND QIAO, Y. Attention-driven dynamic graph convolutional network for multi-label image recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16* (2020), Springer, pp. 649–665.

[104] YIN, M., YAO, Z., CAO, Y., LI, X., ZHANG, Z., LIN, S., AND HU, H. Disentangled non-local neural networks. In *European Conference on Computer Vision* (2020), Springer, pp. 191–207.

[105] YOU, J., YING, R., AND LESKOVEC, J. Position-aware graph neural networks. In *International Conference on Machine Learning* (2019), PMLR, pp. 7134–7143.

[106] YUAN, Y., WANG, L.-N., ZHONG, G., GAO, W., JIAO, W., DONG, J., SHEN, B., XIA, D., AND XIANG, W. Adaptive gabor convolutional networks. *Pattern Recognition 124* (2022), 108495.

[107] ZHANG, B., ZHANG, L., ZHANG, L., AND KARRAY, F. Retinal vessel extraction by matched filter with first-order derivative of gaussian. *Computers in Biology and Medicine 40*, 4 (2010), 438–445.

[108] ZHANG, L., LI, X., ARNAB, A., YANG, K., TONG, Y., AND TORR, P. H. Dual graph convolutional network for semantic segmentation. *ArXiv Preprint ArXiv:1909.06121* (2019).

[109] ZHANG, S., TONG, H., XU, J., AND MACIEJEWSKI, R. Graph convolutional networks: a comprehensive review. *Computational Social Networks 6*, 1 (2019), 1–23.

[110] ZHOU, J., CUI, G., HU, S., ZHANG, Z., YANG, C., LIU, Z., WANG, L., LI, C., AND SUN, M. Graph neural networks: A review of methods and applications. *AI open 1* (2020), 57–81.

[111] ZHU, X., HU, H., LIN, S., AND DAI, J. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 9308–9316.