
# Development of a 3D Reconstruction Technique for the Retinal Surface from Monocular Fundus Photography

**Anghong Du**
Lancaster University

A dissertation submitted for the degree of
*MSc by Research in Computer Science*

February, 2024

# Declaration

I declare that the work presented in this dissertation is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. Regarding the electronically submitted work, I consent to this being stored electronically and copied for assessment purposes, including the School's use of plagiarism detection systems in order to check the integrity of assessed work. I agree to my dissertation being placed in the public domain, with my name explicitly included as the author of the work.

Anghong Du

**Development of a 3D Reconstruction Technique for the Retinal Surface from Monocular Fundus Photography**

Anghong Du
Lancaster University

A dissertation submitted for the degree of *MSc by Research in Computer Science*.

February, 2024

## Abstract

Fundus images play a key role in clinical diagnosis and are especially critical for the diagnosis of retinal diseases. However, current fundus images are usually two-dimensional and lack three-dimensional depth information, which poses a limitation for physicians to fully understand patients' ocular diseases. For diseases that require depth information on the fundus surface such as glaucoma [MacCormick et al., 2019], commonly used diagnostic methods such as fundus OCT often do not provide sufficient 3D information, and these methods do not include background parameters regarding fundus photography information. In addition, since the fundus is located inside the eye, acquiring its corresponding 3D image is often not easy, especially when using a mobile camera like the remidio.

To address this challenge, this study worked on developing a method that can estimate 3D surface contours from monocular fundus images to provide more information about the surface structure of the eye. We created a dataset containing fundus OCT images and their corresponding 3D truth values, named 3D-CSCR.Based on this dataset, we developed a method capable of constructing corresponding 3D models from monocular fundus images and constructed an average template of the fundus 3D model to provide generic structural features.

The results of our study show that our method has made significant progress in providing depth information, which provides ophthalmologists with more comprehensive image information and thus contributes to a more accurate diagnosis of ocular diseases, especially diseases like glaucoma, which require depth information. In addition, our study provides new perspectives for improving ophthalmic medical diagnosis and lays a solid foundation for research and development in the field of 3D reconstruction based on monocular fundus images.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Scientific and technological progress can often affect traditional medical diagnosis methods. In recent years, the improved performance of artificial intelligence algorithms, especially the application of machine learning [Hosny et al., 2018] technologies such as convolutional neural networks to variational autoencoders in the field of medical image analysis, has promoted the rapid development of automated disease diagnosis. As a branch of computer vision technology, medical 3D reconstruction technology can provide rich and clear 3D images for medical image processing, thus playing a vital role in the diagnosis of diseases.

However, due to the small size of biomedical images, the high levels of noise, and the difficulty of obtaining samples, the 3D reconstruction of biomedical images is more difficult than the reconstruction of human hands and the reconstruction of unmanned driving scenes. For example, for hand reconstruction, there are many clean and high-precision hand datasets containing 3D information, such as DART, FreiHAND and HO3D, and the MANO [Romero, Tzionas, and Black, 2017] parameterised model can be applied to pre-estimate a reasonable and accurate initial hand pose. For many types of medical imaging, including those for retina surface OCT, lung cancer, and glaucoma, do not have a prior 3D model, and most pre-trained network parameters such as ImageNet and VggNet do not include these types of images, which adds a lot of resistance to the 3D reconstruction of biomedical images.

Many disease recognition projects based on biomedical images, such as [Yu-Qian et al., 2006][Song et al., 2017][Oktay et al., 2018], have focused on using segmentation networks like U-Net or Mask R-CNN for disease identification and segmentation. Compared with the fields of medical image segmentation and classification, which have made significant progress, research on generating 3D models based on biomedical images is relatively limited. Although some research studies, such as [Y. Wang, Zhong, and Hua, 2019] propose, have successfully used 3D/4D-CT projection or X-ray images to obtain reliable 3D information and construct accurate 3D models, the associated costs are prohibitively high in clinical settings. It is unrealistic for developing countries like China or Africa to be able to afford such expenses for clinical trials.

In addition, machine learning-based clinical apps like PupilScreen [Mariakakis et al.,

2017] have been successful in areas where tech services are relatively scarce, such as Africa. The app enables an initial diagnosis of eye health by capturing images of the eye using a mobile phone camera and relying on computer vision and machine learning algorithms to analyse these images. This example inspired me to think about how machine learning can lead to technological advances in clinical settings, especially in developing countries. In these places, reducing the cost of 3D reconstruction becomes a crucial factor.

By studying research papers related to 3D reconstruction of the hand, such as the MANO parametric model of the hand and the $S^2$HAND paper published by Y. Chen et al., 2021 on generating a 3D hand model from a monocular hand image by leveraging the consistency between 2D and 3D representations, we can analyze and conclude from them that the research o n $S^2$HAND has demonstrated that implementing 3D modeling of the hand requires only a single RGB captured image of the hand. This marks a significant cost reduction compared to traditional 3D reconstruction techniques, which typically involve using a depth camera to capture a depth map and then manually modeling the hand in 3D. This finding further supports our research view that if depth values can be extracted from monocular medical images and utilized for 3D modeling, it will significantly reduce the cost associated with traditional methods. In addition, some diseases, such as glaucoma, which was analysed in the MacCormick et al., 2019 paper, may also lead to changes in the ratio of the optic cup to the optic disc of the eye. The ability to quickly construct 3D models based on OCT images during clinical diagnosis would help to easily identify the presence of these diseases. This further emphasises the importance and potential application areas of our research.

## 1.2 Project Aim

Based on the above situation, this study aims to develop a 3D reconstruction method based on monocular fundus images through machine learning. This method processes biomedical image data with a small amount of noise and extracts 3D features of the depth value of the fundus surface. It will use Computer vision technology generates 3D models to promote the development of automated disease diagnosis technology.

## 1.3 Objectives

In order to achieve our research goals, we need to proceed step by step with the analysis:

- Firstly, we need to find an applicable fundus image dataset. This dataset should contain clear fundus OCT images to ensure image quality and feature clarity. This step is to ensure that we can extract valid 3D information from the images in the subsequent experiments, and the fundus OCT images can provide us with the necessary fundus features to ensure that we can restore the corresponding 3D models in the subsequent applications. At the same time, OCT images are not as expensive as depth cameras or Lidar 3D inspection images [Qian et al., 2020], which ensures the clinical usability

of our study and reduces the cost of 3D reconstruction, in line with our expectations for the results of our study.

- Next, we need to determine whether the dataset already contains the 3D depth value information required for our experiments. If the dataset does not contain this information, we will need to integrate the image feature information in the dataset and construct the corresponding 3D model to infer the corresponding 3D depth value information. This step provides the 3D truth-value labels for our subsequent neural network training and is an integral part of our research.

- After completing the preparation of the dataset, we need to construct a machine learning neural network suitable for this dataset. The goal of our research is to enable the network to extract valid 3D depth value information from monocular fundus OCT images and use this information to restore the corresponding 3D model. The most critical part in this network is to train it to have the ability to learn to recover 3D structures from 2D images. Unlike traditional 2D image detection and classification, our deep learning network needs to not only extract the feature details of 2D images, but also learn the structural features of the fundus surface in 3D space.

- Finally, we will conduct extensive testing and evaluation of the trained model. We will use different evaluation criteria to judge the performance of the model, as well as to analyse the reasons for the poor performance and to find the direction of improvement, and finally to achieve the results we are satisfied with.

## 1.4 Current Approach

Firstly, we constructed a fundus image dataset containing 3D depth value information. We obtained clean fundus OCT images by eliminating the black circular localisation lines and the green scanning area noise caused by the OCT device shots.

To extract effective 3D depth information from the images, we combined the fundus OCT top view and its corresponding slices, constructed a world coordinate system, matched the specific coordinate positions in the fundus OCT top view with the OCT slice images, and filled in the blank areas between the slice images using Gaussian interpolation. This process restored the complete 3D model of the fundus and preserved the corresponding 3D depth information.

After completing the dataset preparation, we proceeded to construct a machine learning neural network suitable for this dataset. We chose the U-Net network based on the image segmentation task and modified it for the regression task. We also added a dropout layer between the Decoder module and the Encoder module in the U-Net network to prevent overfitting problems, especially when the dataset is small. Additionally, we proposed a composite loss function that combines the Mean Square Error (MSE) and the SmoothL1 loss function. The MSE was used for pixel-level image variance, while incorporating the SSIM to consider the structural and perceptual quality of the image. This composite loss function improved the network's ability to perceive changes in image structure while retaining the

ability to compare pixel-level image differences, thus effectively enhancing the adaptability of the U-Net network model to our research goals.

Finally, we performed extensive testing and evaluation of the trained U-Net model using MSE to assess the prediction of 3D depth values. The results showed that our U-Net network model was able to extract the feature details of fundus OCT images well and learned the structural features of the 3D model to some extent. Additionally, we identified a direction for further in-depth research, aiming to improve the network model's ability to learn 3D structural features. Referring to the MANO hand parametric model, we abstracted the fundus 3D model into a digitized parametric model and constructed an average template of the fundus 3D model. This template was input into the model during network training to improve the structural features.

## 1.5 Report Overview

The rest of this thesis is organised as follows. Chapter 2 presents the background and literature review. In Chapter 3, the development of the new 3D retinal surface dataset is presented. The 3D ground truth is constructed in Chapter 4. The first model for 3D reconstruction is presented in Chapter 5. The template-based model is developed in Chapter 6. This work is discussed and concluded in Chapters 7 and 8

# Chapter 2

# Background and Literature Review

Over the past two decades, computer vision has revolutionised the field of medical diagnosis by enabling significant advancements in imaging technology. The emergence of advanced medical imaging techniques, such as 3D imaging [Schwab et al., 2017], has greatly enriched the visual information available to surgeons and physicians, enabling them to make more accurate diagnoses and develop improved treatment strategies.

One of the key areas where 3D imaging has had a transformative impact is in medical diagnosis. Although the 3D information of biomedical images is hard to obtain, it is undeniable that machine learning of biomedical images can play a great role in clinical diagnosis. By visualising anatomical structures in three dimensions, healthcare providers can gain a more comprehensive understanding of complex diseases and their progression. This enables them to plan treatments more effectively, perform surgical interventions with greater precision, and deliver personalised care tailored to each patient's unique needs. Such as, in order to promote hematopathologists to study megakaryocytes in bone marrow trephine biopsy, Song et al., 2017 proposed a new framework based on supervised machine learning, using colour and texture features to effectively delineate megakaryocyte nuclei and cytoplasm in digital images of bone marrow trephine biopsy. Oktay et al., 2018 proposed a novel attention gate (AG) model for medical imaging, using AG to train the U-Net model, which implicitly learns to suppress irrelevant areas in the input image while highlighting salient areas features useful for specific tasks. Using this method to locate the pancreas has greatly improved performance compared with the cascaded multi-model CNN method [S. Liu et al., 2022]. The availability of 3D imaging [Singh et al., 2020] has significantly improved the accuracy and efficacy of medical diagnoses, leading to better patient outcomes.

## 2.1 Application of 3D reconstruction in the field of computer vision

In the field of computer vision, 3D reconstruction technology is a compelling field, which aims to obtain the geometry of objects from a collection of images. Its applications span many fields such as robot navigation [M. Li et al., 2019], object recognition and scene

understanding [Handa et al., 2016], 3D modeling and animation, industrial control, and medical diagnosis [Zhao and L. Wang, 2019][Borse, S. Patil, and B. Patil, 2013]. The core goal of this technology is to convert information from single-view or multi-view two-dimensional images or point cloud data into 3D models containing three-dimensional structural and geometric information [Ram et al., 2022]. 3D reconstruction is a complex process involving multiple technical fields. Its core technology roughly includes the following three aspects:

- Visual geometry is the basis of 3D reconstruction, which studies camera projections and geometric transformations. By calculating feature points, camera parameters and geometric relationships in the image, the three-dimensional shape and position of the object can be reconstructed.
- Deep Learning: Deep learning methods, especially Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN), have achieved tremendous breakthroughs. They can be used for tasks such as extracting features from images, performing image segmentation, and generating realistic 3D models.
- Point cloud processing: For point cloud-based 3D reconstruction, point cloud processing technology is the key. This includes filtering, registration, segmentation and fitting of point clouds to obtain accurate 3D models.

In addition to division based on application fields, 3D reconstruction can also be classified and applied according to the type of reconstructed objects. These include human body reconstruction SMPL [Loper et al., 2015], scene reconstruction CAM [Facil et al., 2019], object reconstruction SMR [Hu et al., 2021], human hand reconstruction MANO [Romero, Tzionas, and Black, 2017], and medical image 3D model construction, etc. These different types of application fields reflect the diversity and wide application of 3D reconstruction technology. In the field of biomedicine, 3D reconstruction technology plays a vital role in medical image processing [J.-J. Wang et al., 2021], pathological analysis, surgical planning, etc. Compared with the application of 2D images in the medical field, common surgical procedures use X-rays as a reference [Ham, Wesley, and Hendra, 2019] for doctors to perform operations based on specific conditions. However, some important features often cannot be well visualized in 2D images [Yao et al., 2003]. Besides, medical diagnosis based on 2D images also depends on the accuracy of the image, including the number of 2D views, noise in the image, and image distortion. These requirements further limit the medical diagnosis effect of 2D views in some cases.

As 3D reconstruction technology in the biomedical field gradually matures, the above-mentioned problems have gradually been solved. 3D reconstruction technology provides doctors and researchers with comprehensive and accurate vision by converting 2D medical images into 3D models with spatial information. Display and analysis tools. This allows them to examine medical images from multiple angles, gain a deeper understanding of anatomy and pathology, and make informed decisions about patient care. [Pichat et al., 2018] introduces 3D reconstruction methods for 3D histology to overcome the limitations

of single-section studies in a dimensional range. [Guedri, Malek, and Belmabrouk, 2015] used fractal interpolation to 3D reconstruct human retinal blood vessels. [Sumijan et al., 2017] in their work introduced a method to calculate volume Hemorrhage Brainon CT-Scan Image and 3D Reconstruction. These examples illustrate the expanding promise of 3D reconstruction in biomedicine, demonstrating its versatility and transformative potential. As this technology continues to develop, it is expected to further enhance the capabilities of medical professionals and researchers seeking to improve healthcare and gain a deeper understanding of complex biological systems.

## 2.2 Methods to achieve 3D reconstruction

Traditional 3D reconstruction methods of biomedical images mainly rely on geometric models and traditional computer vision technology. These methods reconstruct 3D models by extracting geometric features and edge information from medical images. For example, [Kar et al., 2015] proposed a class-specific 3D shape model learning method based on object contours, capable of capturing intra-class shape variations from a single image. In addition, the voxelization method divides medical images into voxel grids and uses the relationship between pixel gray values and adjacent pixels for reconstruction. Another approach proposed by [Huang, H. Wang, and Koltun, 2015] is to use a global network to establish dense pixel-level correspondence between natural and rendered images, which is then used for joint image segmentation and 3D model building. Furthermore, [Widya et al., 2019] suggested that medical images can be reconstructed from videos by learning structure from motion (SfM). The 3D model generated by SfM can provide better 3D visualization and provide more details for doctors to diagnose. However, traditional methods have certain limitations when dealing with complex scenes, noise, and blurred images. In the field of biomedical images, it is more difficult to obtain clear, high-quality and sufficient images of medical cases than large objects (such as human bodies, buildings, scenes).

In recent years, deep learning techniques have made remarkable advancements in the field of 3D reconstruction, particularly in biomedical imaging. Deep learning methods leverage neural network models to achieve more accurate and efficient 3D reconstruction of biomedical images. These methods can learn meaningful feature representations from images and directly reconstruct 3D structures through end-to-end training processes. Convolutional neural networks (CNNs) [Aghasi et al., 2017] excel in extracting image features, while [McCann, Unser, et al., 2019] proposed imaging system models that use forward models and sparsity-based regularisation to solve reconstruction problems. Generative adversarial networks (GANs) offer advantages in improving image quality and capturing fine details. Additionally, methods based on variational autoencoders (VAEs) and attention mechanisms have emerged, aiming to enhance computational efficiency and image quality while maintaining accuracy.

Although modern methods have made significant progress in achieving 3D reconstruction, 3D reconstruction in the biomedical field still presents some challenges. Biomedical images, such as optical coherence tomography (OCT) images, are often affected by various

interference factors, such as positioning lines and time recording noise, which makes it difficult to obtain a large number of clean medical images. Additionally, with CT (computed tomography) images, there is a potential risk of radiation exposure for the patient due to their use of X-rays. Especially for children, pregnant women, and patients who require multiple scans, radiation dose can be a concern. Because CT images have high resolution in displaying hard tissues and relatively low resolution in displaying soft tissues, they are often used for 3D reconstruction of non-soft tissues (such as teeth and bones) [Heo and Chae, 2004]. MRI (magnetic resonance images) is more expensive than OCT images and has a longer scan cycle, it is not suitable for medical image applications that require instant imaging such as eyeball scanning. However, in terms of 3D reconstruction, MRI-based biomedical images have shown outstanding performance good effect. [Gnonnou and Smaoui, 2014] proposed a 3D reconstruction method based on breast cancer MRI images, which effectively solved the problem of detecting such cancers and the problem of 3D reconstruction of MRI images for breast cancer detection.

## 2.3 3D reconstruction based on monocular images

Compared with multi-view 3D construction, the 3D reconstruction method based on single view has wider availability of image resources. However, extracting information from monocular images and predicting 3D models are also more challenging tasks [Tian et al., 2023], because a single view can only express limited feature information [Toppe, Nieuwenhuis, and Cremers, 2013]. In addition, occlusion often occurs in single-view images [Y. J. Lee et al., 2012], which means that some parts of an object may be occluded by other objects or structures, resulting in incomplete image information. Unlike the multi-view approach, the single view problem cannot be solved simply by comparing multiple views together.

The initial single-view three-dimensional reconstruction work mainly restored the object shape through mathematical perspective changes [Lavest, Rives, and Dhome, 1993]. After that, [Sun et al., 2018] proposed to simultaneously recover 3D shape and surface color from a single image, namely "color 3D reconstruction". With the rapid development of deep learning, single-view 3D reconstruction methods based on voxel, point cloud and grid representation have been applied [Shin, Fowlkes, and Hoiem, 2018], achieving better results than before.

After this, a number of excellent results on single-view based 3D modelling have gradually emerged in the field of biomedical imaging, e.g., [Y. Wang, Zhong, and Hua, 2019] proposed DeepOrganNet, a method that can generate and visualise fully high-fidelity 3D / 4D organ geometry models in real-time from single-view medical images with complex backgrounds. Unlike conventional 3D / 4D medical image reconstruction that requires nearly hundreds of projections, the method learns to smooth the model through the DeepOrganNet framework, exploits information-rich latent descriptors extracted from the input 2D image, and generates high-quality and high-fidelity streaming meshes for the 3D/4D lung model, significantly shortening the surgical time for dynamic visualisations and

significantly reducing the human subjects' imaging dose. After that, [Loper et al., 2015] proposed a skeleton-driven parametric human body model SMPL. The model is learned based on a large amount of human body data and uses a vertex-based additive method to control the shape changes of the 3D mesh. The SMPL model is controlled by a shape component and a pose component, enabling it to effectively represent the 3D form of the human body and provide a template mesh for single-view 3D reconstruction. Similarly, the MANO [Romero, Tzionas, and Black, 2017] hand parametric model is also based on the SMPL hand topology. The MANO model is similar to SMPL. It includes a template mesh, a dynamics tree (a tree structure for finger movement), parameters that control the shape and posture of the hand, as well as structural parameters such as joint regression matrices and skin weights. All these parameters provide strong support for single-view-based 3D hand reconstruction. In addition, there are some other related research, such as SMR based on 2D-3D loss consistency [Hu et al., 2021], and DD-Net [Guan et al., 2021], which is specifically designed for 3D sparse and limited-view photoacoustic tomography (PAT) image reconstruction. By incorporating dense connections and dilated convolutions into the U-Net architecture, DD-Net enhances information flow and expands the effective receptive field without sacrificing resolution coverage. Similarly,[Gunduzalp et al., 2021] proposed a reconstruction technique that leverages deep learning and adopts a 3D U-Net architecture (similar to U-Net) to denoise sparse or noisy signal projections in the image domain. Furthermore, in [Seok et al., 2021] paper, a U-Net based deep learning architecture was used to generate personalized 3D models for patients undergoing thyroid surgery (after obtaining appropriate informed consent).etc.These studies have achieved satisfactory results.

Medical imaging plays a key role in clinical diagnosis and promotes advancements in the field of clinical medicine. Our research is to achieve three-dimensional reconstruction based on monocular fundus OCT images without providing depth information. These fundus OCT images will be converted into accurate 3D models to promote the development of automated disease diagnosis. Especially in the diagnosis of complex fundus and retinal diseases, the use of two-dimensional fundus OCT images and their corresponding visualized 3D images allows medical professionals to better understand the complex structure of the fundus retinal surface, thereby improving the level of medical diagnosis.

# Chapter 3

# Building a New 3D Retinal Surface Dataset

## 3.1 Requirements and Search of Suitable Datasets

### 3.1.1 Requirements

Our research aims to achieve 3D reconstruction based on monocular fundus images, and we have the following data requirements for our experiments:

- Requirements for Fundus Images: We require a dataset containing multiple fundus images, which should be from different patients or sources, and try to ensure that the dataset covers different kinds of fundus diseases and health states so that the model can work effectively in multiple situations. The size and quality of the dataset is also important for training the deep learning model, so we need to collect as many fundus images as possible and filter the data to remove images with obvious anomalies and errors, and make sure that the structural regions of the fundus are clear enough to help in the extraction of features from the surface of the fundus.

- Depth value labels: For each fundus image, we need to have corresponding depth value labels. These labels contain the specific depth values of the fundus surface structures, expressed in pixel units from the world coordinate system. This requires us to extract the depth information in the fundus image by means of data preprocessing to construct the corresponding mask labels.

- Training and Testing Sets: The division of training and testing sets is also very important for model training. The training set will be used to train the regression network model while the test set will be used to evaluate the performance of the model. When dividing the fundus image dataset into a training set and a test set, make sure that there is no data overlap between the two to ensure the generalisation ability of the model and avoid overfitting during training.

### 3.1.2 Data Search

We performed a search for datasets, including the Messidor dataset that facilitates computer-aided diagnosis of diabetic retinopathy, the Diabetic Retinal Morphology dataset provided by the Kaggle competition, the DRISHTI-GS1 dataset from the prospective study of glaucoma conducted at the Aravind Eye Hospital in India, and datasets from other domains such as the 3D reconstruction of the hand for the FreiHAND and HO3D datasets, etc.

We found that although deep learning has been widely used in the field of biomedical imaging with significant achievements in different disease domains, most of the applications are still focused on disease classification or disease segmentation [He et al., 2021] of 2D images [Aggarwal et al., 2011] via neural networks. Even though we have been able to extract features and construct 3D models from monocular images in dealing with large objects, such as faces [Amin and Gillies, 2007], hands [Loper et al., 2015], and cars [Lequellec and Lerasle, 2000], no in-depth research results have been achieved in the field of monocular image-based 3D reconstruction of tiny objects, such as the fundus retina. Although many datasets exist that provide excellent OCT fundus images and their labels, they lack the corresponding labels with information about the details of the fundus surface, and therefore we are unable to construct the corresponding 3D models.

Based on the above, we can draw the following conclusions:

(i) Although several good quality fundus datasets are currently available, these datasets do not contain 3D information on fundus OCT, which is crucial in our study.

(ii) Relevant datasets that we currently have include the DR-OCT dataset and the CSCR dataset. Of particular note, the CSCR dataset not only contains OCT top-view images of the fundus surface, but also provides the corresponding sliced OCT images. This provides the necessary foundation for us to combine the existing 2D dataset to construct a dataset containing 3D depth information of the fundus surface.

Therefore, (iii) we plan to create a new dataset applicable to our study based on the existing dataset, which will contain information about the 3D depth values of the fundus surface. This will provide more comprehensive data to support our study for 3D reconstruction based on monocular fundus images.

## 3.2 Introduction to Existing Datasets

The dataset we plan to create, named CSCR_3D, will contain the following elements: a fundus OCT top view image, a slice OCT image corresponding to the feature location, and key 3D ground truth mask labels. Our goal is to use this dataset to train the neural network by feeding the fundus OCT top-view image into the neural network and providing the corresponding 3D ground truth at the same time, in order to help the neural network to extract the deep features and predict the feature details of the fundus surface. The construction of this dataset relies on the CSCR dataset and the DR-OCT dataset that we will introduce next.

### 3.2.1 CSCR Dataset

The dataset We currently have is from the Lancaster ICVL group, called CSCR (center serous chorioretinopathy), and contains fundus images of the left and right eyes of 77 subjects (some subjects had only a left or right eye). This dataset includes two types of EDI (As shown in Figure 3.2) images and OCT (As shown in Figure 3.1) images. Because the OCT image can quickly scan the retina within milliseconds, it can eliminate as much as possible the image capture error caused by the experimenter's eye rotation and other problems, and the accuracy is high. At the same time, OCT can image different layers of fundus tissue, with better spatial resolution and contrast. EDI images can provide deeper retinal imaging, but what we want to achieve here is to build a corresponding 3D model based on the top view of the monocular fundus OCT image, so we are only interested in the details of the surface layer of the fundus, so we decided to use OCT images for the production of the dataset.

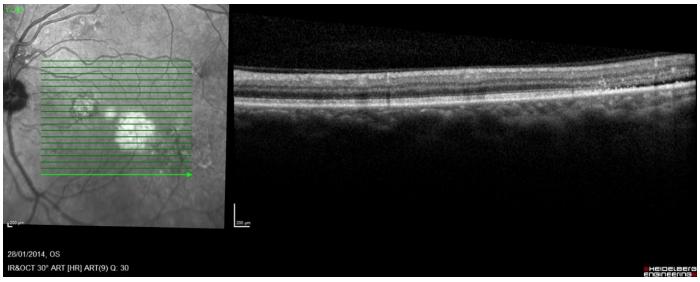


Figure 3.1: Subject A's left eye OCT image and partial slice images

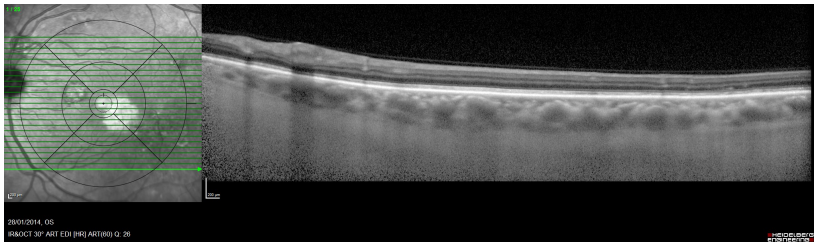


Figure 3.2: Subject A's left eye EDI image and partial slice images

The data in the CSCR dataset consists of left and right parts, where the left part is the fundus OCT top-view scanning image, which contains the depth features of the fundus surface, and the observation also shows the auxiliary black localisation lines left by the OCT scanning, as well as the highlighted green lines in the scanning area, where each position of the green lines with arrows corresponds to the right part of the sliced OCT image, which is the key for us to construct the 3d depth information of the fundus surface.

### 3.2.2 DR-OCT dataset

In addition to the CSCR dataset that we have, we also draw on a comprehensive open access dataset OCTID, the Optical Coherence Tomography Image Database which contains more than 500 high-resolution OCT images that are categorised into different pathological conditions. We selected the portion of these images that contains diabetic images and normal images to form the DR-OCT dataset(diabitc retinopathy, shown in Figure 3.3), and we chose to use this dataset because it provides the same OCT images as our CSCR dataset, which can show the deep layers of the retina, which is very important for us to obtain the information about the depth of the retinal surface, and it also provides the ground truth(As shown in Figure 3.4) corresponding to the OCT images. contour, which is not included in our CSCR dataset, with the help of DR-OCT dataset we can construct the mask labels corresponding to the sliced OCT images in the CSCR dataset.

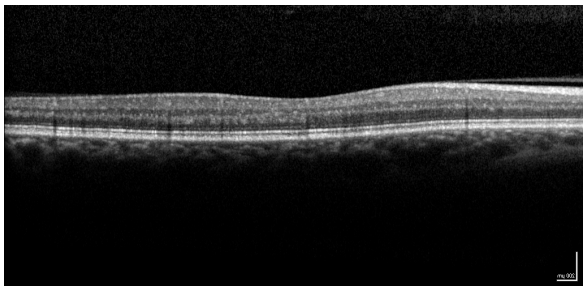

Figure 3.3: DR-OCT_data

Figure 3.4: DR-OCT_data mask

## 3.3 Motivation for pre-processing fundus images

Fundus OCT images in the CSCR dataset contain noise such as black circular lines and green lines in addition to fundus surface depth information.

The black circular lines in fundus OCT (Optical Coherence Tomography) images are usually due to the optical paths and interference patterns of the OCT scanning equipment. These black circular lines are actually reference lines or markers for the OCT scan, which

are used to help the doctor or researcher to identify and locate and correct the scanning of the different structures of the fundus in the image, especially to reason about the position of the various layers, which is very important for obtaining high-quality OCT images of the fundus. Therefore, we were unable to directly acquire fundus images that did not contain black circular localisation lines.

The area covered by the green line usually indicates an image processing or labelling method. In fundus OCT images, the green lines serve to mark the location of the corresponding slice image of the OCT as well as the size of the region. Each green line represents a specific location of a slice image in the fundus OCT image, while the green line with an arrow indicates the exact location of the current slice image. These green lines play a key role in positioning and aligning the slice images to ensure the correct position of the slice image in the overall fundus OCT image.

Our aim is to input the fundus OCT images from the CSCR dataset into the U-Net model to train it for regression prediction, but the black localisation lines and green line noise contained in the images can adversely affect the training of the model, which is the reason why they should be removed:

- Noise interference: the black localisation lines and green lines do not contain useful data about the fundus structure or depth information, but are interference due to the workings of the OCT scanning equipment. During model training, these noises may be incorrectly seen as part of the structure, leading to misleading results from the model.

- Reduced computational complexity: retaining these noises increases the complexity of image processing and analysis, and pre-processing of the data should also take into account whether it will have an effect on the noise that will lead to poorer model results. Removing these lines can simplify the image pre-processing process and improve processing efficiency.

- Improve model performance: removing the noise can reduce the interference of the model and make it more focused on learning the real features of the fundus structure, thus improving the performance and accuracy of the model.

In conclusion, removing the black localisation lines and the green lines can improve the robustness of the model, allowing it to better adapt and extract features from the fundus OCT image, and construct a better 3d fundus surface model.

## 3.4 Remove the noise interference of the black ring positioning line

It can be seen from the image that the initial data set we obtained at the beginning is full of noise interference such as green lines and black circles on the OCT image because OCT and EDI images need to be positioned and scanned, so we need to remove these noises.

### 3.4.1 Try to set the threshold to remove the black positioning line

At the beginning, we started to solve the black circle, because the RGB value of black is close to 0, so We want to clean up the noise by extracting the value of all pixels, looking for a threshold, and removing pixels below this threshold The purpose, but in fact, if the noise is removed in this way, the black circle noise can be removed perfectly, but at the same time, a lot of image details will be lost, which is unacceptable to us.

### 3.4.2 Corrosion operation

In our specific case, the main goal is to eliminate the black positioning lines that appear in binary images. These lines often prove detrimental to subsequent image processing tasks and can significantly affect the accuracy of our analysis.

We then decided to look deeper into whether the corrosion operation [Ahuja and Shukla, 2018], commonly known as etching, might be able to solve our problem. This operation often plays a key role in image processing. It does this by acting on every pixel in the binary image. It uses a small structural element (usually a square or circular kernel) to scan the image and replace the value of each pixel with the minimum value in its local neighborhood.

The choice of whether to use square or circular kernels in a task often depends on the specific image processing task and application scenario. Square kernels are often used when it is necessary to emphasize a uniform impact on specific areas in the image. For example, when removing small-sized noise points or smoothing image edges, square kernels can provide a more uniform effect. The circular kernel is mostly used to process images involving arcs and curves, because they are closer to the image in shape, and are more suitable for situations where the characteristics of arcs, curves or circular objects in the image need to be preserved. They can be more Captures the outlines of these shapes nicely.

The result of the erosion operation is to erode or reduce the boundaries and features of objects in the image, effectively compressing scattered pixel areas and eliminating unwanted artifacts such as noise points, fragmented lines, and isolated points. From a theoretical perspective, erosion operations help improve the connectivity of an image, remove unnecessary details, and reduce the size of objects in the image. Therefore, we hope to obtain a clean fundus image by removing the black positioning ring that is incompatible with the surrounding area through an etching operation.

By applying erosion, the aim was to erode the boundaries of these lines, effectively eliminating them from the image and retaining only the essential features intact. However, upon careful consideration of the specific characteristics of the lines and the resulting consequences, we ultimately concluded that erosion was not well-suited (As shown in Figure 3.5) for achieving our intended goals.

Therefore, it is imperative for us to explore alternative methods that can effectively remove the black positioning lines while simultaneously preserving the crucial features present in the image. By doing so, we can ensure the accurate and reliable analysis required for the successful completion of our graduation thesis.

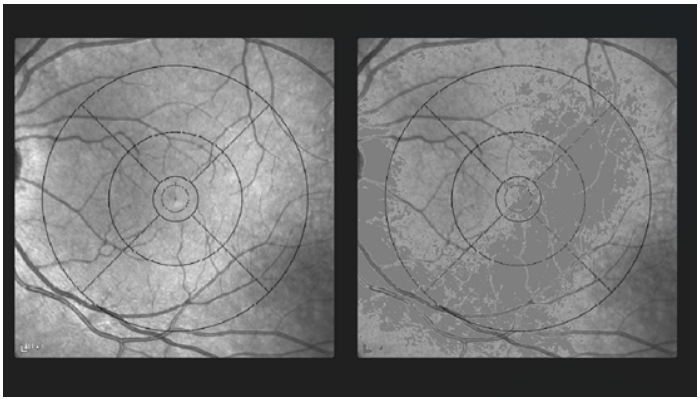

Figure 3.5: corrosion operation

### 3.4.3 Retrieve the black ring

Because the black circle noise is a regular pattern, we thought of extracting the black positioning line shape by retrieving straight lines and circles. we first convert the image to grayscale and then perform a binarization operation on it (the object of the operation must be a grayscale image). We then retrieved the circles in the image through the Hough transform, but in the end only the largest circle was successfully retrieved and covered with white (As shown in Figure 3.6). While this method provides some degree of noise reduction by eliminating the largest circles, it fails to account for the remaining smaller circles and any potential irregularities in the image. Therefore, further exploration and refinement of alternative techniques are necessary to effectively address these challenges and achieve the desired level of noise cancellation in our study.

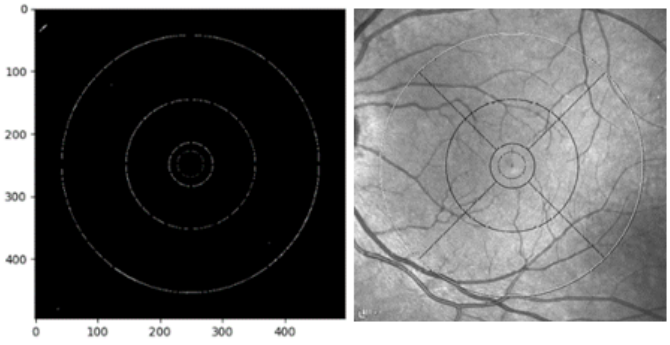


Figure 3.6: retrieve black circle

Inspired by the previous steps, we chose not to retract and cover the rings one by one, but to choose a suitable RGB threshold, make a mask image, and then directly use the Navier-Stokes algorithm to interpolate the mask image, and finally succeeded in getting a relatively good result.

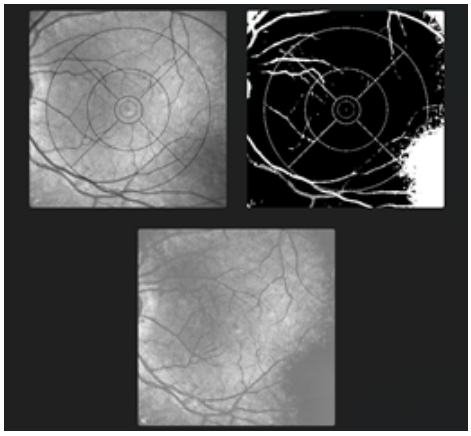

Figure 3.7: The result after removing circles and positioning lines through RGB threshold (from left to right and from top to bottom are the original OCT image, the mask obtained through RGB thresholding, and the OCT image obtained after removing noise through mask)

However, although the result of this method is to successfully remove the noise and reduce the loss of image details (As shown in Figure 3.7), even such a loss is still unacceptable. At this point, we began to think about how to accurately locate the position of the ring in each image.

Although there is no way to directly remove the rings, we can use the cv2.HoughCircles() function to obtain useful information about the position of the center of the circle and the approximate radius of each ring. Here we wrote a detect_ring code to extract the center and radius information, and use this information to make a ring mask template.

We obtain the center and radius data of the ring through the circle retrieval algorithm of Hough transform, and obtain the parameters rho and theta of the straight line through the straight line retrieval algorithm of Hough transform. rho represents the distance from the origin to the straight vertical line, and theta represents the angle between the vertical line and the x-axis. Through the obtained straight line information and its intersection on each circle, the position information of four intersection points (top_left, top_right, bott_left, bott_right) is obtained.

### 3.4.4 Mask for making black circle positioning lines

In order to create a template that can effectively cover the black ring positioning line, we conducted a decomposition of the black ring noise into its constituent parts. We identified four rings, four diagonal lines, a straight line on the first ring, a cross line at the center of the circle, a line on the left side of the fourth ring, and a line on the right side of the fourth ring. Based on the obtained center position, four intersection positions, and four ring radius information, we constructed a mask template.

Since we only want to remove the black ring positioning line, we need to expand the

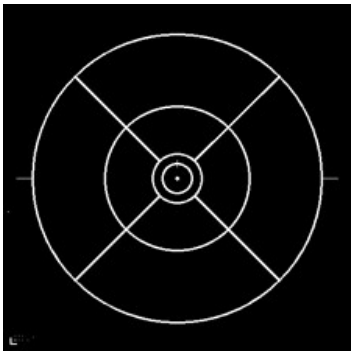

Figure 3.8: mask template

mask to the size of the original image and ensure that only the position of the positioning line is marked.

Figure 3.9: extend mask template

This mask template (As shown in Figure 3.8 and Figure 3.9) was then used as the mask parameter in the cv2.inpaint() function to remove the noise from the original image. The cv2.inpaint() function works by filling in the masked regions with plausible image content, based on the surrounding pixels. By applying this technique, we were able to successfully remove the black ring positioning line while preserving a significant amount of image details (As shown in Figure 3.10 and Figure 3.11).

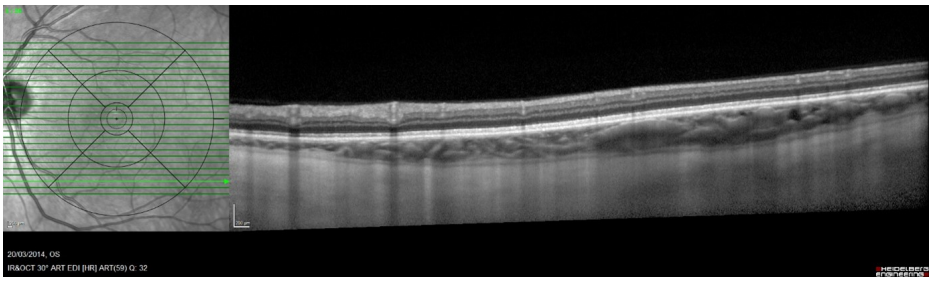


Figure 3.10: original fundus image

The use of the mask template in the inpainting process yielded excellent results, significantly improving the overall quality of the image and reducing the impact of the

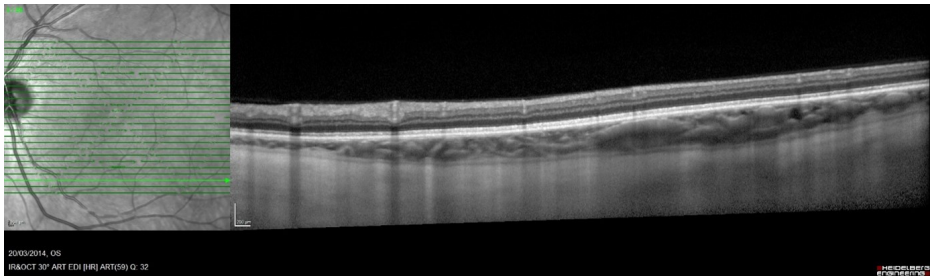


Figure 3.11: the result after processing using the mask template

noise. This approach allowed us to obtain a cleaner and more accurate representation of the desired features in the image.

## 3.5 Remove green line noise interference

It is relatively easy to remove the noise of the green line, because if the values of the three channels of RGB are the same, the pixel will display gray. The value of the three channels of RGB determines the brightness . If the values of the three channels of RGB are not the same, the pixel will display Various colors, so what we have to do is to traverse the values of all pixels in the image, replace all the pixels that are not gray with white (255, 2 55, 255), and then use the cv2.inpaint() function Just do the interpolation. Result shown like Figure 3.12.

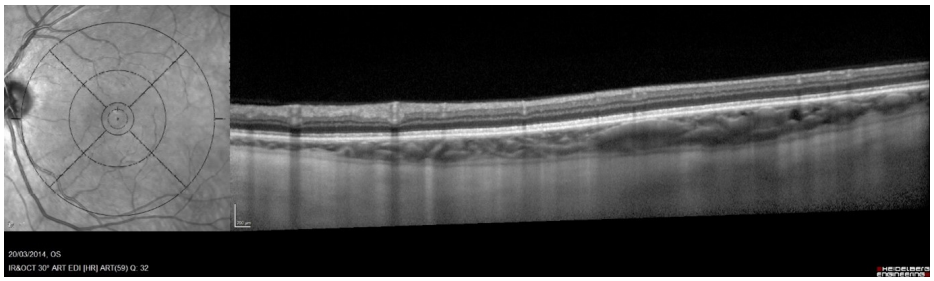


Figure 3.12: Remove the green line noise result

## 3.6 Initially obtain one clean OCT fundus image

By employing the mask template method to eliminate the black circle noise and the technique of removing the green line noise, we are able to obtain a pristine OCT fundus image when these two methods are combined. However, it is crucial to follow a specific order of noise removal, giving priority to eliminating the black circle noise before tackling the green line noise. This sequence is essential because of the subsequent image inpainting process performed using the cv2.inpaint() function.

Once we remove the noise from the image, the cv2.inpaint() function works by interpolating and repairing the affected areas. Consequently, the original positions of the noise are filled with other pixels. If we were to remove the green line noise first, numerous black pixels would remain in the image after inpainting. However, by eliminating the black circle noise first, the interpolated pixels will predominantly consist of green, allowing for their simultaneous removal when addressing the subsequent green line noise. Results shown like Figure 3.13 and Figure 3.14.

Following this sequential approach ensures that the resulting image is free from both the black circle noise and the green line noise. It leads to a more accurate and visually appealing representation of the OCT fundus image, providing a clearer view of the underlying features of interest.

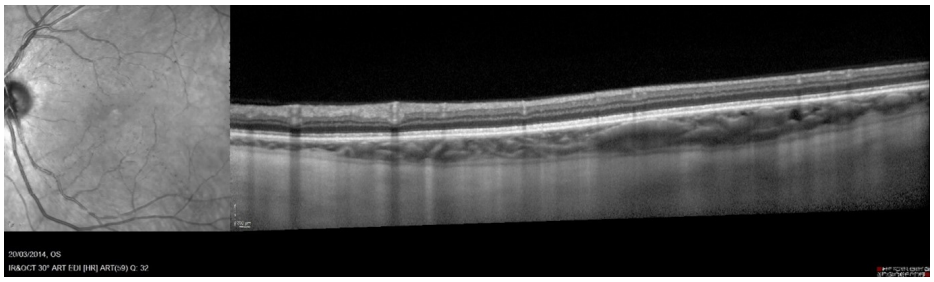

Figure 3.13: Removing in the wrong order

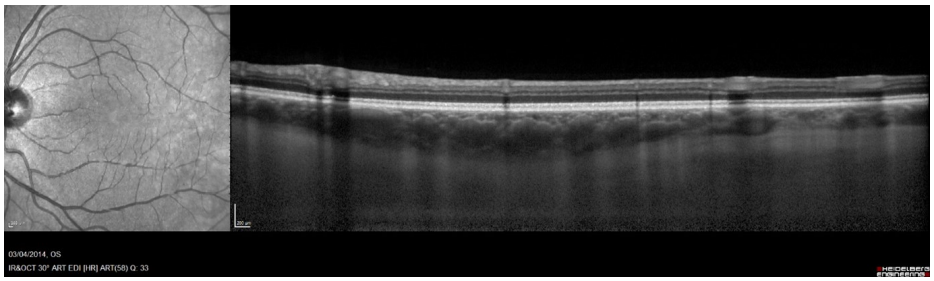

Figure 3.14: Removing in the correct order

## 3.7 Automatically process all images and obtain corresponding OCT fundus images

Clean OCT fundus images obtained by data preprocessing proved the feasibility of the above method, but there were limitations that could not be addressed by this template. The corresponding template produced from a specific image is not compatible with other images. If there are other images with inconsistent ring sizes, inconsistent centers, or different ranges of green lines, there will still be a lot of noise using a fixed template that will not achieve the results we require.

In order to make this method work for all images, we tried to construct methods that automatically find position information, make mask templates for the corresponding images, and automatically remove noise. To realize the automatic search function, combined with the conditions required for constructing mask templates before, we first need to know the position of the corresponding black localization line in the image for different OCT images.

### 3.7.1 Backtracking algorithm to solve for Template center point coordinates

Since we are solving a graphical and decision-making problem, and since our noisy templates are also used to obtain the coordinates of the intersections of straight lines and circles through the Hough transform and the retrieval of circles in bresenham's algorithm, the first thing that comes to mind is to simulate the trajectory of the movements of the MASK centers through a backtracking algorithm to find the corresponding MASK centers for the new image.

First, we selected the template center point and moved it in four directions, up, down, left, and right, by adding and subtracting along the x and y coordinate directions. After each move, we performed an elimination operation on the template to assess its effect on the OCT image. Specifically, we first calculated the initial pixel mean and subsequently, after each action, the pixel mean was calculated again and compared to the initial value. Since black pixels have an RGB value of 0, the RGB value of pixels filled using Gaussian interpolation must increase after eliminating black pixels. Therefore, when the average value of the pixels of the OCT image rises, it can be inferred that the action in that direction is effective.

However, in subsequent implementations, we noticed that the results were not as expected. Through breakpoint debugging, we found that in the backtracking algorithm, the center point was backtracking when retrieving surrounding pixels. Specifically, after acting upward once, the center point would retrieve the other three directions around it. However, if satisfactory results were not obtained in these directions, the center point would fall back to its previous position and then repeat the process over and over again. Meanwhile, even if we succeeded in finding a new image corresponding to the mask center, the sizes of the four circles may vary from image to image, i.e., the problem of scaling the sizes of the circles from one image to another.

### 3.7.2 Traversal alternative backtracking algorithm to find the center point of the black marked line

Based on a detailed observation of the dataset, we found that there is limited variation in the distance between the centers of the localization lines taken by the same instrument. In order to overcome the backtracking problem of the backtracking algorithm in finding the centroid, we decided to adopt a more effective strategy, i.e., using the average value of the pixels in the surrounding 20x20 range for locating the center of the locus line, and sacrificing the space complexity in exchange for the reduction of the time complexity. This

method has achieved remarkable results in practical applications, however, we also realize that it has certain limitations. In particular, when facing images with large differences in shooting angles, the traversal range may need to be adjusted accordingly, and we note that the adjustment of the traversal range may lead to an increase in the space complexity, thus requiring a trade-off between time complexity and space complexity.

### 3.7.3 Automatically find the size change of the black circle and its position.

With the center of the black localization line determined, we consider the radius of the rings as a variable parameter. By varying the radius size of the rings one by one, we observe how the change in the radius of each ring correlates with the change in the average value of the image pixels. The goal of this process is to find the appropriate radius size such that there is a clear correspondence between changes in the radius of the rings and changes in the average value of the image pixels.

By systematically repeating the above operation for each circle, we can gradually construct a mask template in which the radius of each circle is precisely adjusted to reflect the exact characteristics of the black localization lines of the new image. Eventually, the mask template we obtain will be able to exactly match the black localization lines in the new image, thus providing us with a precise and effective tool to accomplish noise removal.

### 3.7.4 Loss Determination Criteria for Finding Algorithms

Let the mask template adjust its position and size according to the position of the black circle in the image, which reminds me of the application of loss value in deep learning of neural network. However, our previous method of using the global pixel average of the image is not only very slow in terms of time, but also takes up a lot of unnecessary space resources in terms of space complexity, and cannot automatically find the position of the black positioning line for noise elimination. Here we need to define the range of the average value calculated in this algorithm. First, define a variable as the average value, which is the average value of all pixels in the black circle on the mask image, because the value range of the RGB value of the pixel is [0,255 ] , and the RGB value of black is (0, 0, 0), so the closer the mean value is to 0, the closer these pixels are to black, that is, the closer to the circle position.

The implementation is as follows. First, we define a standard mask template. 1. Get the RGB values of the pixels on the ring by center position and radius. 2. through the Bresenham algorithm, take the position information of the four intersection points (top_left, top_right, bottom_left, bottom_right), backward calculate the position of the pixels on the four straight lines, and obtain the corresponding RGB values. 3. Obtain the RGB values of the center pixel point Cross, the upper line of the first circle, and the left and right sides of the fourth circle. The RGB average thus obtained represents the proximity of the template to the black localization line in the image, which is much less space-complex than the global average of the image used previously, while the improvement in computational speed is

also significant. 4. Define the optimization class. This class contains the find_center and find_circle functions. By traversing the surrounding locations, traverse the size of nearby circles to get different averages. Find the location information of the center of the circle, the radius information of the circle, and the location information of the four intersections when the mean is smallest. This is the closest mask template to the black circle on the image. 6. By constantly adjusting the mean value, we obtain a different mask template suitable for each image, and then using our previous method of eliminating the black circle noise and the green line noise, we can automatically obtain a clean fundus from the initial dataset of OCT images , and there is virtually no loss of image detail information (the loss of detail information when interpolating and fitting via inpaint is unavoidable)

# Chapter 4

# Constructing 3D ground truth based on CSCR dataset

After performing a series of data preprocessing operations on the CSCR data set by clearing the black positioning lines and green scanning areas, we obtained the clean fundus OCT top view of the fundus OCT and its corresponding slice images from the original OCT image (As shown in Figure 4.1).

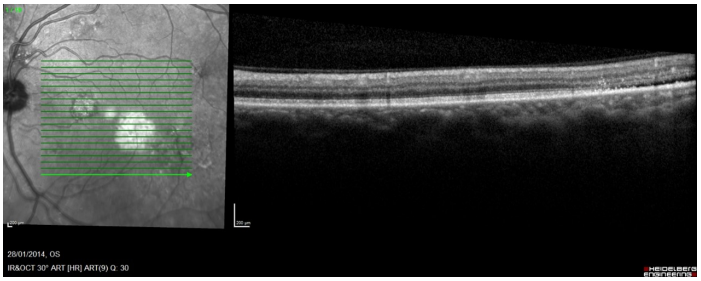


Figure 4.1: Original OCT fundus image

Now what we need to do is to integrate the information through the limited data in hand and deduce the 3d data information from it. From the top view of the fundus OCT we can see that there are many features that can be extracted such as texture features, color features, shape features, etc., but our purpose is to build a retinal surface model, so we only need the distance of the retinal fundus from the camera view.

## 4.1 Automatically Generating retinal OCT slice image mask labels

To get the details of the surface of the fundus image, we need to know the distance and depth of the surface of the fundus from the camera, because we uniformly take the top view from the same perspective, so we need to get the distance from the surface of the fundus

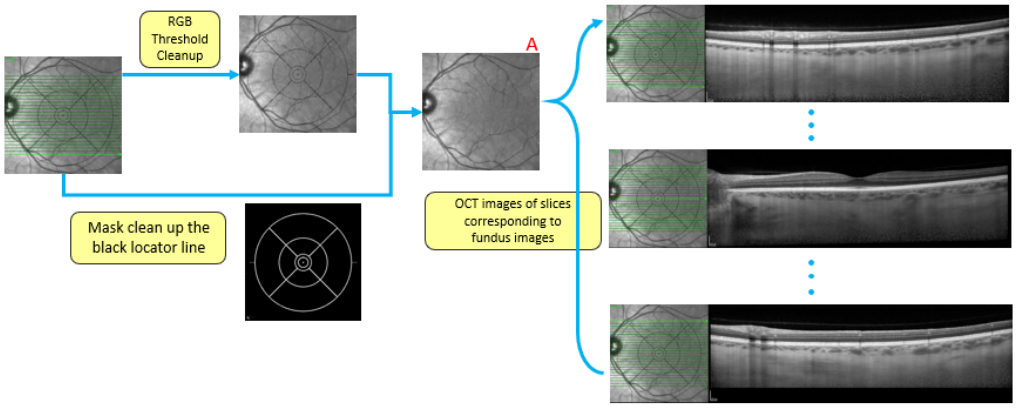


Figure 4.2: Data pre-processing process

to the top of the image, which we call the fundus depth value. Data pre-processing process shown in Figure 4.2.

Then we need to construct the corresponding coordinate system according to the OCT slice image to specify the distance of the camera viewpoint from the fundus surface, and we selected the distance of the fundus surface from the coordinate axis x of the fundus OCT slice image as the fundus depth value. As shown in Figure 4.3.

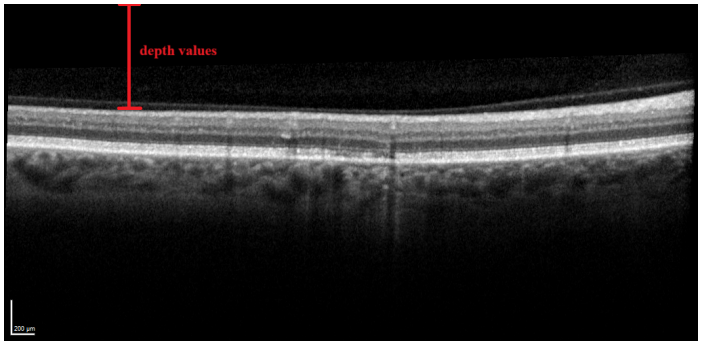


Figure 4.3: Definition of fundus depth value

### 4.1.1 Determining the training dataset DR-OCT for neural network segmentation models

In order to get the coordinate distance corresponding to the fundus depth value, we need to obtain the precise coordinate position of each pixel of the fundus surface, but it is known that our image shape is (1024, 496), which means that we need to artificially find the coordinates of the 1024 pixel points for each sliced image, and the number of sliced images corresponding to each fundus OCT top-view image is 19, which is a huge project.

In addition to identifying and confirming the coordinates pixel by pixel, we can also use professional labeling tools such as LabelMe to label the surface of the sliced OCT images, and then identify and record them by code. However, this method still has a big labor cost problem, and it is still a tedious and huge project to annotate nearly 600 images one by one.

For the case of too many images, manual annotation labor cost is too large, based on the current computer vision field of mature image segmentation and recognition technology, we think of using our computer vision experimental group another group of similar dataset DR-OCT_dataset, the images in the dataset are the same as the fundus slice OCT images, and contains a complete mask image and the corresponding data, we use this dataset and its annotations as the training and testing sets for the neural network, so that the trained neural network can automatically annotate the fundus slice OCT images in our CSCR dataset.

### 4.1.2 Segmentation of train and validation set of DR-OCT dataset

We began by randomly permuting the identifiers of the DR-OCT dataset using the np.random.permutation function. This step was taken to introduce a certain level of randomness into the dataset division, thereby enhancing its diversity. Subsequently, we divided the randomly shuffled sequences into an 80% training set and a 20% validation set. These sets were used for training the deep learning model, evaluating its performance, and checking for potential issues such as overfitting.

After segmentation training of the U-Net network using the DR-OCT dataset, we evaluate the model performance based on Loss, Accuracy and IoU score evaluation criteria. The three images below show the numerical changes of these criteria during the training process.

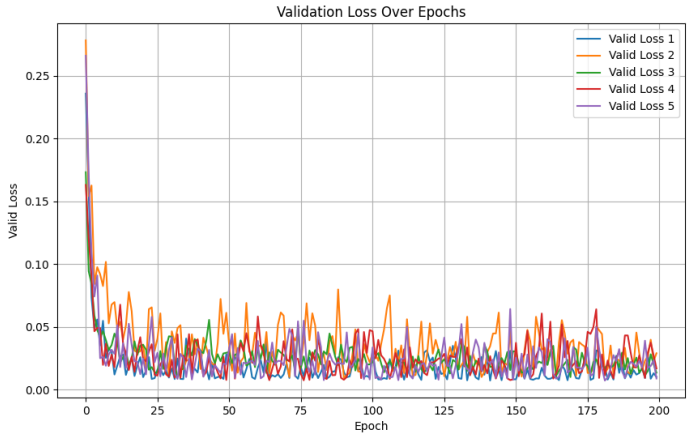


Figure 4.4: Validation Loss Over Epochs comparison (The numbers in the upper right corner represent 5 datasets split in different random ways. Each Valid Loss represents a validation curve obtained by training using a type of dataset.)

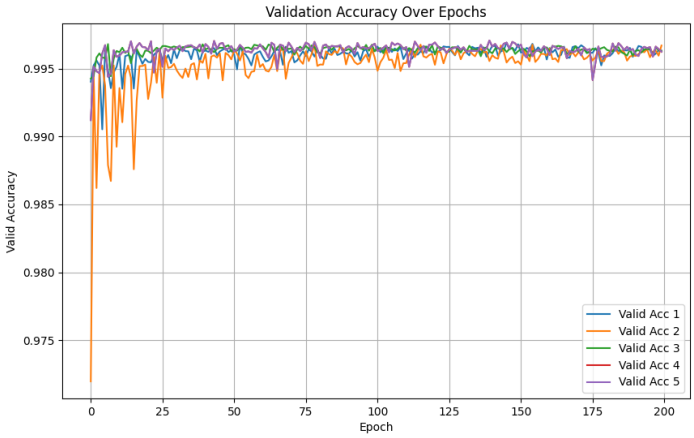


Figure 4.5: Validation Accuracy Over Epochs comparison (The numbers in the the lower right corner represent 5 datasets split in different random ways. Each Valid Acc represents an accuracy curve obtained by training using a type of dataset.)

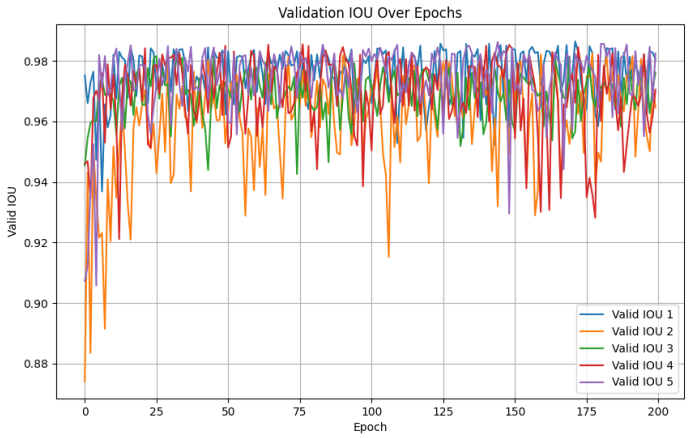


Figure 4.6: Validation IOU scores Over Epochs comparison (The numbers in the the lower right corner represent 5 datasets split in different random ways. Each Valid IOU represents a IOU scores curve obtained by training using a type of dataset.)

We conducted this dataset division process five times, and each resulting dataset was fed into the segmentation model for performance evaluation. Upon comparing our data curves, we observed that while valid_log5 had the highest accuracy, log1 exhibited the most substantial decline in the Loss curve and achieved the highest IOU scores. Both of these datasets achieved a final accuracy of over 99.5%. Consequently, we selected the dataset generated from the segmentation ratio of log1 as the ultimate dataset. This dataset was then used for comprehensive training of the segmentation network model, and we saved the best-weighted parameters.

### 4.1.3 U-Net image segmentation network model construction

U-Net, proposed in 2015, is a deep learning semantic segmentation model whose underlying architecture still continues the core idea of full convolutional neural networks. It was initially created to address the semantic segmentation challenges in the field of medical imaging, especially for applications in areas such as organ segmentation and lesion detection in medical imaging. In recent years, U-Net has gained wide application in the field of medical image analysis.

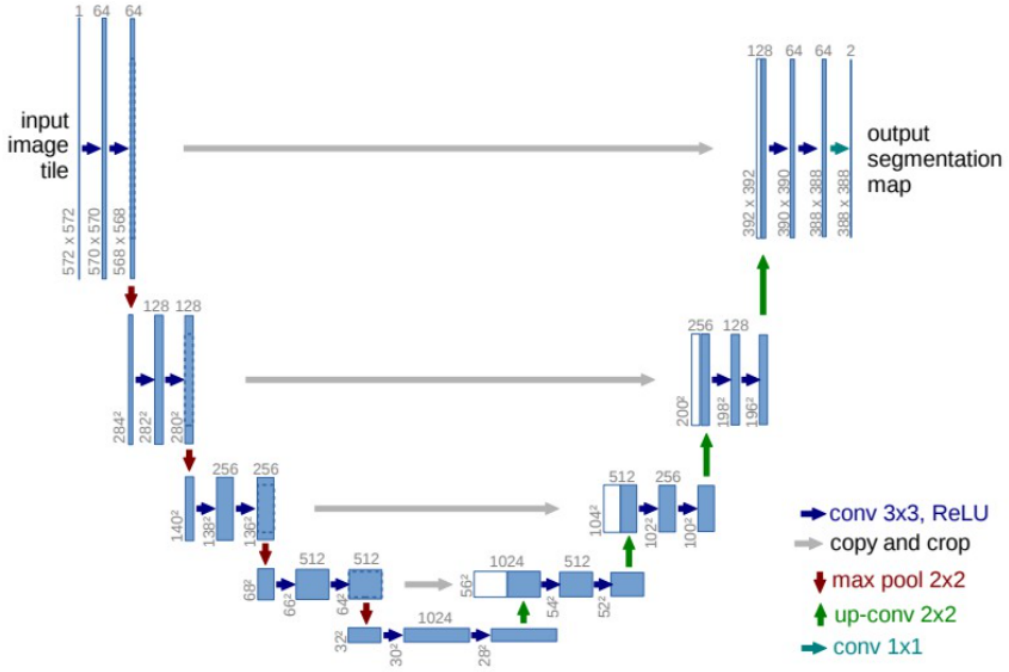


Figure 4.7: U-Net network model (the left part is the Encoder module, the right part is the Decoder module) Image source: U-Net model, Website URL: https://blog.csdn.net/weixin_44969144/article/details/126153665

We selected U-Net (model shown in Figure 4.7) as a model for OCT slice image segmentation based on the following considerations: first, its unique Encoder-Decoder

structure can tightly fuse the shallow features and deep features of the input image, restore the low-resolution image containing high-level abstract features to a high resolution, and then, by fusing with the high-resolution image of the low-level features (through the Then, by fusing with the high-resolution image of low-level features (through concatenation operation), a feature map with complex semantic information is generated.

Another significant advantage is U-Net's pixel-level classification, where the output is classified for each pixel point. This design naturally fits our need for localization in fundus OCT slice images, which can accurately label and localize pixel coordinates on the surface of fundus OCT slice images.

Given the excellent performance of U-Net in the field of image segmentation, we decided to adopt SMP (Segmentation Models) segmentation model in PyTorch framework to build an efficient U-Net image segmentation neural network framework.

In this framework, we chose SE-ResNeXt ('se_resnext50_32x4d') as the component of Encoder, an image classification network based on the ResNeXt architecture, which incorporates multi-dimensional grouped convolutions and is thus capable of capturing a rich variety of features (As shown in Figure 4.8). In the network module, it retains the Residual structure proposed based on ResNet networks. This structure effectively solves the gradient vanishing problem by connecting the inputs directly to the middle or end layers of the network through shortcut connections.

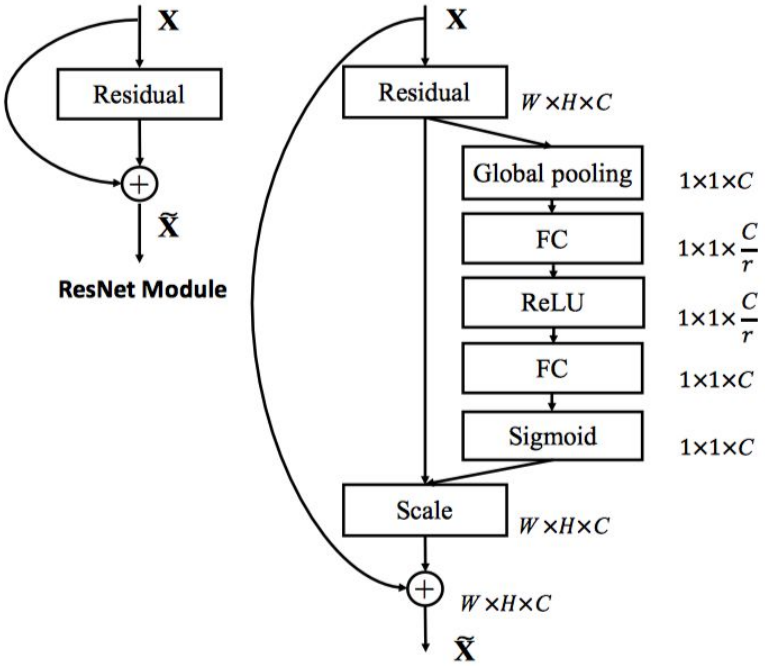


Figure 4.8: Comparison of Residual modules between ResNet and SE-ResNeXt

In addition to this, SE-ResNeXt introduces the Squeeze-and-Excitation (SE) module,

which is a channel attention mechanism. This module enables the network to intelligently adjust the weights of channel features by globally pooling the feature graph of each channel and then learning the channel weights with the help of a series of fully connected layers. Such a mechanism allows the network to adaptively highlight task-relevant features while suppressing responses to task-irrelevant features.

### 4.1.4 Transfer Learning

Following the completion of the U-Net neural network structure, our decision to initialize the model with pre-trained weights from 'ImageNet' [Deng et al., 2009] marks a pivotal step. These pre-trained weights are obtained through U-Net's prior training on the extensive ImageNet dataset, a colossal collection of images encompassing millions of visuals across over a thousand categories. Through this pre-training process, the U-Net model effectively learns to extract features from a vast array of images, acquiring the ability to capture intricate details about each individual category. This collection of parameters can be conceptualized as an abstract representation of features derived from a diverse array of images.

The comprehensive nature of the ImageNet dataset, coupled with its remarkable diversity, allows for the seamless adaptation of these pre-trained parameters to diverse visual tasks, mirroring our current demand for the task of annotating sliced OCT images. Notably, even in the presence of a relatively modest dataset, these parameters exhibit the capacity to yield commendable performance. This approach, founded on the principles of transfer learning, not only obviates the need to initiate the training of extensive models from scratch but also translates into substantial savings in terms of time and computational resources.

### 4.1.5 Image segmentation evaluation metrics

In image segmentation tasks, the choice of evaluation metrics that quantify the performance of the U-Net model in segmenting the surface of OCT sliced images is crucial. Here we have chosen two image segmentation evaluation metrics: Intersection over Union (IOU) and Accuracy.

IOU (Intersection over Union) is one of the important metrics to measure the performance of segmentation model. It evaluates the accuracy of segmentation by calculating the ratio of intersection and concatenation between the predicted results and the real mask annotations in the segmentation task. The specific calculation formula is as follows eq4.1:

$$IOU = \frac{A \cap B}{A \cup B} \tag{4.1}$$

The intersection of A and B here is the overlap between the model prediction and the real annotation, while the concatenation of A and B refers to their merged parts.The value of IOU ranges from 0 to 1 (As shown in Figure 4.9), and a higher score means a more accurate segmentation range of the surface of the sliced OCT image, based on which it is an intuitive measure of the model's segmentation boundary precision and segmentation region

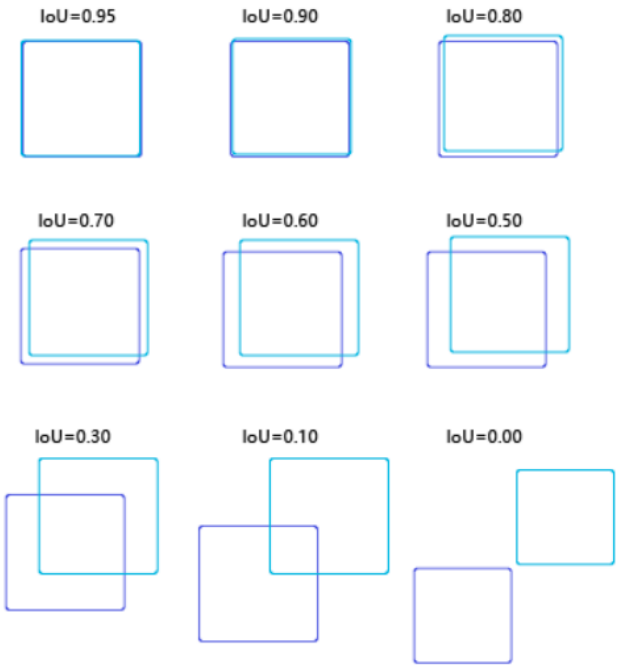


Figure 4.9: Comparison of different IOU scores

accuracy, reflecting the effectiveness of the model training. The score is higher than 1, which means that the segmentation range of the sliced OCT image is more accurate.

Accuracy, on the other hand, is another common evaluation metric that calculates the number of correctly predicted pixels as a proportion of the total number of pixels, and is usually used in image classification tasks. However, it can also be used in image segmentation tasks, especially since our image segmentation task belongs to binary segmentation (dividing an image into target and background). In our OCT slice image segmentation task, the accuracy is calculated by determining the number of correctly predicted pixels and then comparing it to the total number of pixels. By examining the number of pixels that the model correctly segmented the surface region of the fundus as a proportion of the overall image pixels, we are able to accurately reflect the accuracy of the model's prediction results, leading to a more in-depth understanding of the model's performance and training effectiveness.

### 4.1.6 Data Augmentation

In order to increase the diversity of the training data, improve the generalization ability of the deep learning model, and effectively avoid the overfitting problem, we employ a series of image enhancement operations. These operations subject the original images to multiple transformations to generate more diverse training data. This enhancement process was designed to target the specific needs of fundus OCT slice images and was implemented through the albumentations library.

We used the following multiple ways to perform randomized data enhancement on the data, with each data enhancement method deciding whether or not to perform this operation according to the corresponding probability at the time of invocation.

- horizontal flip:

Each batch calls the training set with a 50% probability to flip the image horizontally to increase the diversity of the data.

- Scale, Rotation and Translation Transformations:

Scale, rotation and translation transformations are performed on the image to simulate fundus OCT images at different angles and positions.

- fill image:

Fills the image according to the specified minimum height of 320 and width of 320.

- Random Crop:

Randomly crops the image at a size of 320x320 to keep the image size consistent while expanding the training set.

- Gaussian Noise:

Each batch calls the training set with a 20% probability to add Gaussian noise to the image, introducing random, Gaussian-distributed [Goodman, 1963] brightness variations in the image. Due to the random nature of the Gaussian distribution, each pixel varies differently, thus creating a visual noise in the image. Adding this random interference can help improve the robustness and performance of the algorithmic model by simulating real-world noise situations, thus allowing the model to better adapt to images in various environments.

- Perspective Transformation:

Each batch calls the training set with a 50% probability of perspective transformation of the image, which guides the parallel lines of the image to the same vanishing point, simulates the sense of perspective near and far, and converts the camera viewpoint to a different viewpoint, thus producing a perspective effect, increasing the diversity of the training data, and improving the adaptability of the deep learning model to different imaging conditions and angles.

In addition to the conventional data enhancement methods mentioned above, we further constructed an integration of the three data enhancement methods through the "OneOf" operation, so that there is a 90% probability of randomly selecting a method from this integrated package during the data enhancement process of each batch, in order to increase the Randomness and diversity of data enhancement. This integrated data enhancement operation includes the following three methods:

- a. Contrast, Brightness and Gamma Transform:

By randomly selecting an operation, we apply it to the image to increase the change in brightness and contrast of the image.

- b. Sharpening, Blurring and Motion Blur:

By randomly selecting an operation, we simulate changes in the sharpness and blurring of an image as a way of introducing varying degrees of visual effects.

- c. Contrast and Hue Saturation Transformations:

By randomly selecting an operation, we apply it to the image to increase the color saturation and contrast variations of the image, in order to enrich the visual characteristics of the image.

### 4.1.7 Obtain the Mask label corresponding to the CSCR dataset image

Based on the DR-OCT_ dataset and with the help of the U-Net image segmentation network constructed using the SMP framework, we successfully obtained the optimised weight parameter best_model_eyes1.pth after 200 epochs of training.During the training process of each epoch, we continuously monitored and recorded the evaluation stage of the validation metrics, including the results of valid_loss, valid_acc, and valid_iou, and stored these data in a csv file named valid_logs1.

From the detailed analysis of the data in valid_logs1.csv, it is obvious that the U-Net network model performs well in terms of training effectiveness. Its performance on the validation set shows a gradual decrease in the loss value, which implies the gradual improvement of the model's learning ability and fitting ability. In addition, in terms of accuracy, after 25 epochs of training, the accuracy of the model tends to be stable and close to the level of 99.6%, which proves the stability of the model in classifying unseen data. Correspondingly, the IOU scores fluctuate between 96% and 98%, highlighting the model's high adaptability to the segmentation task.

Together, these positive performances demonstrate the successful training of our designed U-Net model on the DR-OCT dataset. It is able to almost perfectly separate the fundus surface from the surrounding background region in OCT slice images effectively, achieving excellent image segmentation and annotation results. This series of performances confirms the effectiveness of the optimal weighting parameters, which can be migrated to our subsequent CSCR dataset for image segmentation of the fundus surface, thus further enriching our analyses and experimental results!

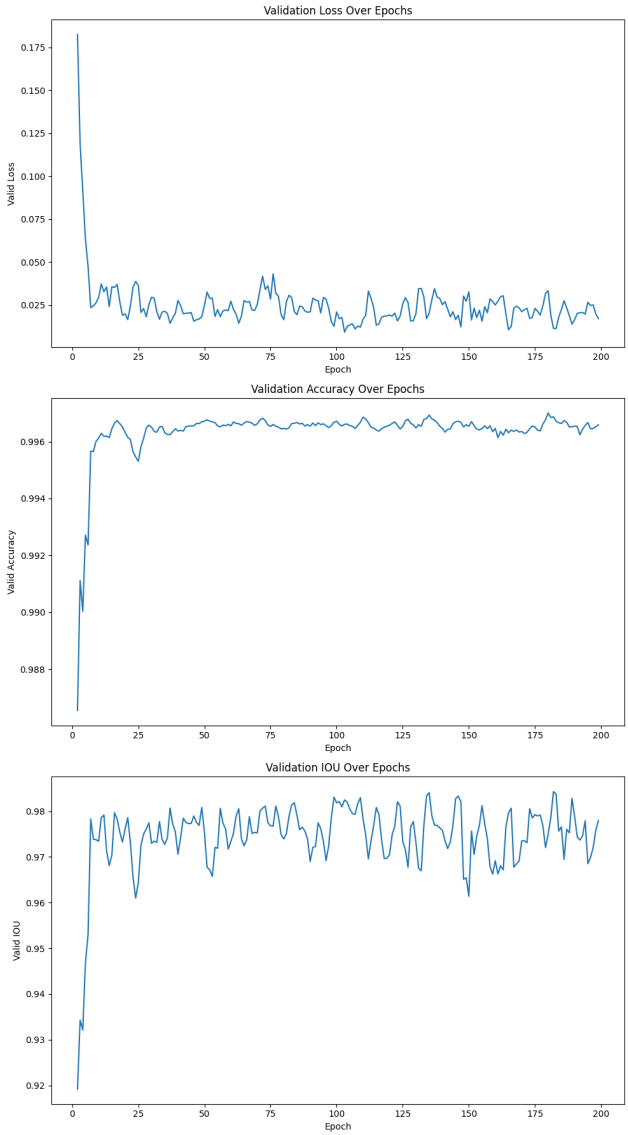


Figure 4.10: Validation Metrics Over Epochs

We replaced the 'imagenet' pre-trained parameters with the meticulously optimized weight parameters we obtained through our training regimen. Employing this refined set of weights, we employed the U-Net network architecture to conduct precise image segmentation on the CSCR dataset. The outcome was a set of meticulously generated masks, each intricately aligned with its respective image. These masks were then methodically stored within the designated 'mask' directory, meticulously serving as the precise labels for the delineated fundus surface within our CSCR dataset. Result shown like Figure 4.11.

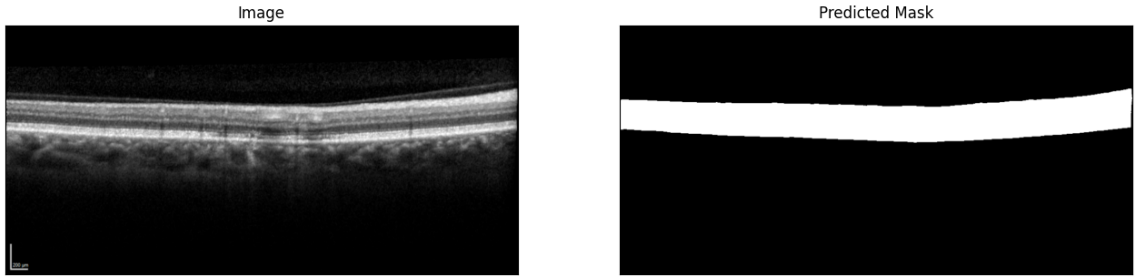


Figure 4.11: OCT fundus slice image and its corresponding Mask Label

### 4.1.8 Manual removal of abnormal labels using LabelMe software

Although the optimal weighting parameters of the U-Net network model in the DR-OCT dataset succeeded in distinguishing the fundus from the background with an accuracy of more than 99%, there is still a certain degree of error rate. To ensure the rigour of the research results, we manually checked the mask labels of 674 images automatically segmented by the U-Net network model and saved the identified segmentation error labels in the "error" folder. According to our statistical analysis, the accuracy of the best weighting parameter in the DR-OCT dataset drops to 87% for the segmentation annotation task on the CSCR dataset.

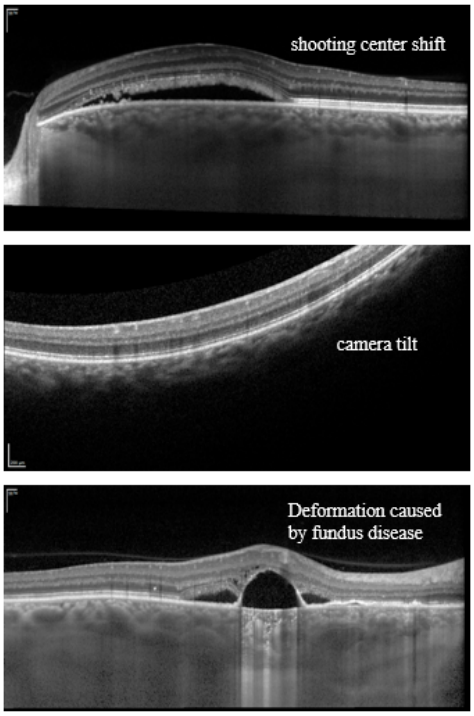


Figure 4.12: CSCR dataset image problems

We conducted an in-depth analysis of the original images corresponding to these mislabelled images (As shown in Figure 4.12) and concluded that the CSCR dataset contains some fundus images of patients with ophthalmic diseases as well as incorrectly photographed images, which are missing in the DR-OCT dataset (even though a few ophthalmic disease slices are also included in the DR-OCT dataset), and that such a lack cannot be compensated for by data preprocessing and data enhancement operations.

To address this problem, the first task was to exclude error images with tilt and translation problems. Unlike the fundus images that appeared to feature disease, these mislabelled images could no longer be positionally aligned with the fundus top-down OCT images, and even if we were able to acquire the depth information, we were unable to map them to the correct position in the world coordinate system. We then used LabelMe software to manually annotate the remaining mislabelled images to obtain complete annotation information. This step of annotation using the U-net segmentation model not only significantly reduces the cost of manual annotation, but also provides us with high-quality annotated data. Based on these labelling data, we can obtain pixel-level depth value information for each OCT image corresponding to the slice image, which provides an accurate and reliable basis for our study. Result shown in Figure 4.13.

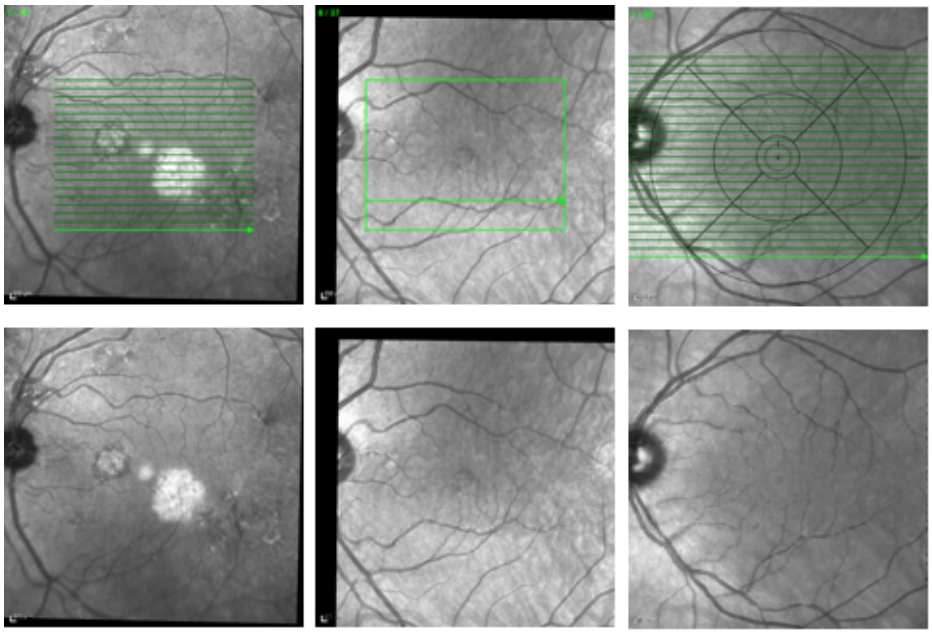

Figure 4.13: Data preprocessing effect display

## 4.2 Constructing World Coordinate System to extract depth information

In the process of constructing the fundus surface model, our key task is to obtain the depth values of the fundus surface corresponding to the ground truth and clearly define the size range of the model. Based on the analysis of the top view of the fundus OCT, the range of the fundus OCT slices corresponds to the size of the green area labelled in the top view of the fundus OCT, which in turn defines the range of the features from which we can extract the depth values. We determined the pixel positions of the four directional corners of the green localisation line by locating the left, right, and upper and lower polar ranges of pixels with green RGB values in the image by their RGB values, which ultimately gave us the green localisation line region size, i.e., the size of the 3D ground truth construction range. As shown in Figure 4.14.

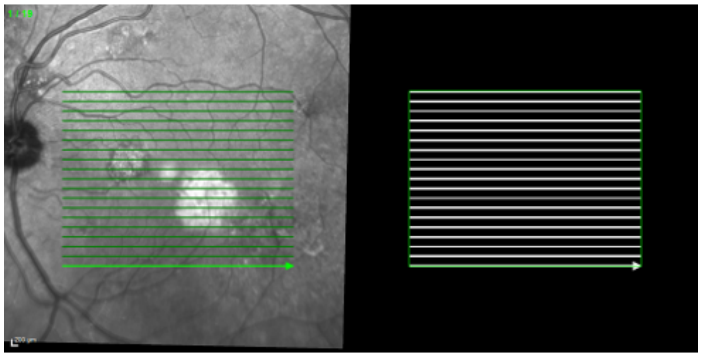

Figure 4.14: Depth value feature extraction range

### 4.2.1 Automated acquisition of depth value information for each slice image

After determining the size of the 3D ground truth range, our next step is to obtain depth information corresponding to each individual slice image. Defining the distance of fundus surface pixels from the camera's viewpoint as depth values is the foundation of our approach. However, manually extracting the positions of fundus surface pixels is a monumental task. To address this challenge, we leverage the distinguishing feature in the fundus OCT image masks where the labels have different RGB values from the background.

We achieve this by systematically traversing each column of the label pixels and identifying the minimum y-coordinate where the label pixels are present.This process allows us to determine the distance from the top of the image to the fundus surface in the mask label and obtain the corresponding fundus surface depth value for each fundus slice image.

### 4.2.2 Construct a World Coordinate System to map fundus slice images

Intermittent depth value information is meaningless for neural network learning, and we need to correspond the fundus OCT top view and its corresponding all slice images. For this purpose, we established a world coordinate system, mapped all slice images separately according to the position of the green localisation line in the fundus top-view OCT image in the same coordinate system, integrated the depth value information of all images, and constructed the corresponding 3D model of the fundus.

In order to integrate two different-sized images, namely the fundus OCT top-view and its corresponding set of fundus OCT slice images, into the same global coordinate system, it is essential to ensure that their pixels can be aligned seamlessly. As shown in Figure 4.15.

The fundus OCT slice images have dimensions of 1024x496 pixels, whereas the effective area of the fundus OCT top view measures 332x251 pixels. Therefore, we employed an interpolation method to process the fundus OCT top view, interpolating its original 332 pixels into 1024 smaller pixels. This enables us to accurately map the 1024 pixels of the OCT slice images to the fundus OCT top view, facilitating the seamless integration of these two differently sized images within the same global coordinate system.

After completing the mapping of a single slice image, we also need to consider that one fundus OCT top view often corresponds to multiple slice images at different positions. In our scenario with 19 corresponding slice OCT images, our task is to sequentially map these 19 images to the positions corresponding to the green positioning lines in the original image. This process requires us to determine the position coordinates of each positioning line. Based on the analysis of the original OCT image, we have observed that the interval between each green positioning line remains constant.Therefore, we perform an equal division operation on the range of $y \in [0, 250]$, dividing it into 19 segments, each of which corresponds to the position of a slice OCT image. The process is shown in the Figure 4.16.

At this point, we have been able to reconstruct the 3D model using the positional information from the original fundus OCT top-view and the depth values obtained from the OCT slice images. It turns out that the effect is good. However, there is still a need for further refinement. While the top view looks promising, from a lateral perspective, it remains evident that it consists of 19 slice images spaced apart, rather than providing continuous depth information. Unfortunately, we cannot extract depth values from the top-view within this interval.Therefore, we must resort to Gaussian interpolation to enable us to establish continuous depth information for the entire area.

Through the above-mentioned method of mapping images from the horizontal and vertical directions, we have successfully mapped the original fundus OCT top view and its corresponding multiple slice images to the same world coordinate system.But what needs to be noted here is that all operations we do on the sliced OCT image must be performed on the mask again to ensure that the mask is also in the same world coordinate system and that the fundus OCT image always corresponds to its mask.

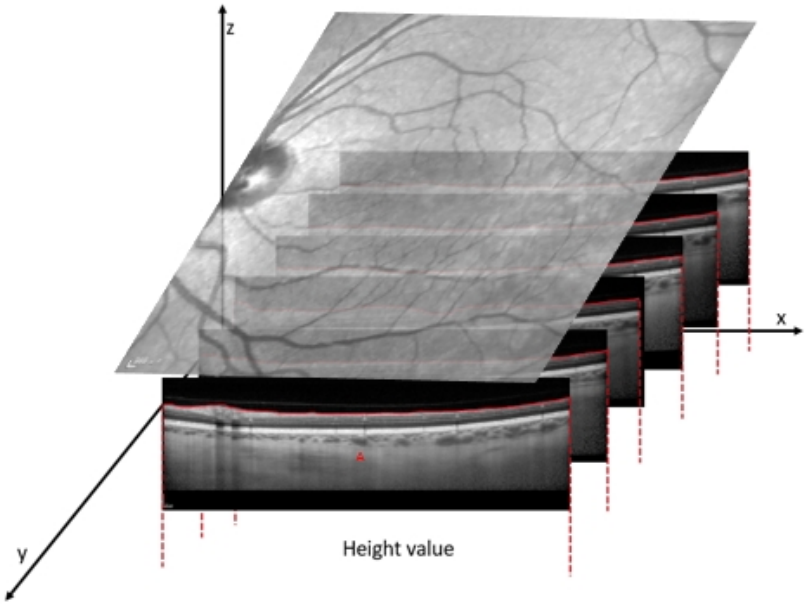


Figure 4.15: The correspondence between fundus and slices in 3D

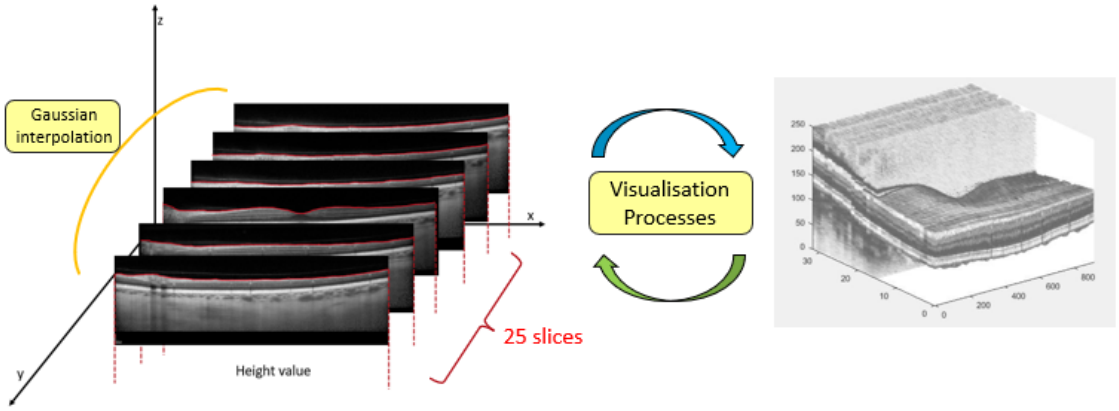


Figure 4.16: Visual display of world coordinate system construction

## 4.3 Processing and Storage of 3D ground truth

At this stage, we have successfully constructed the 3D ground truth of the fundus OCT images, as illustrated in Figure 4.17, demonstrating excellent results. However, it is important to address the fact that not all pixels in the mask images contain accurate pixel depth values ranging from 0 to 1024. Some of these pixels deviate from the expected features of the fundus structure, and such pixel points need to be excluded from our analytical framework as they affect the interpolation of the surrounding pixel points when performing Gaussian interpolation.

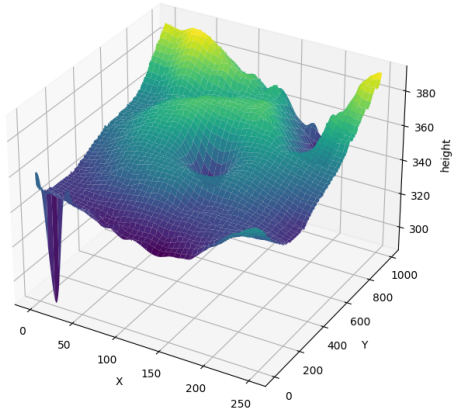


Figure 4.17: 3d ground truth without processing outlier pixel values

In the presence of such abnormal pixels, we perform a column-wise search to identify pixels with depth values significantly different from the majority of pixels. Subsequently, we remove the entire column of pixels containing these points. This step produces the final model, effectively creating a 3D model of the fundus surface.

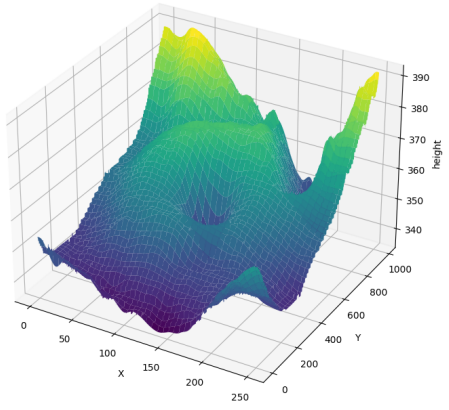


Figure 4.18: The final constructed 3D ground truth

After obtaining the 3D ground truth corresponding to the fundus OCT images respectively, we need to save them as labels suitable for neural networks. Since the original images are 2D images, we consider saving the 3D ground truth as a 2D image as well. However, unlike npy arrays, PNG images usually have a value range between [0, 255]. Therefore, we need to normalise the depth values stored in the npy array to the range of the PNG image in order to feed them into the neural network for training along with the training data. We used two normalisation algorithms:

$$y = 255 \cdot \frac{y - min(y)}{max(y) - min(y)} \tag{4.2}$$

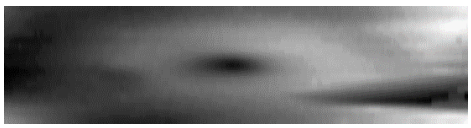

Figure 4.19: 3D Ground Truth saved using normalisation algorithm Eq (4.2)

The normalisation algorithm Eq (4.2) is based on the minimum and maximum values of the dataset, which ensures that the range of depth values is between [0, 255], and is suitable for cases where different depth values need to be mapped to the standard RGB colour range for maintaining data consistency. However, if there are outliers in the dataset, such as noise or outliers in the depth values, it may result in the range of depth values being stretched by the outliers, making the normalised depth image not obvious enough to distinguish between different depth levels, which may lead to the loss of details in some depth levels.

$$y = \left(\frac{y}{495}\right) \cdot 255 \tag{4.3}$$

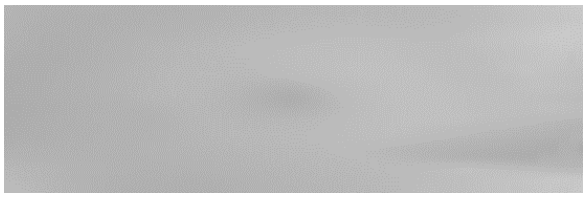

Figure 4.20: 3D Ground Truth saved using normalisation algorithm Eq (4.3)

The normalisation algorithm Eq (4.3) scales depth values directly and proportionally to the range [0, 255], which is simple and fast. The reason we choose to use a factor of 495 for scaling is that the depth value of the 3D model is in the range [0, 495]. By applying the scaling operation $\frac{y}{495}$, we can map the numerical range of depth values to [0, 1]. This step normalizes the range of depth values to better adapt to the normalization process. The subsequent multiplication operation 255 proportionally maps the normalized depth range into the standard range [0, 255] of pixel RGB values. Effective in preserving depth level detail for datasets with relatively uniform depth value distribution.

Given that our 3D ground truth dataset has a relatively uniform distribution of depth values and the outliers have been removed and filtered in advance by means of data preprocessing, we choose to use the normalisation algorithm Eq (4.3) as a simple but effective preservation method. Before inputting the model, we will use the inverse operation of the algorithm to input the real depth value information into the model for training.

# Chapter 5

# Regression-based 3D Surface Estimation from Monocular Images

## 5.1 Motivation for U-Net regression model construction

Achieving 3D reconstruction based on monocular images is currently one of the important research directions in the field of computer vision and has achieved remarkable results in several fields, such as hand reconstruction. The core problem in this direction is how to extract multi-dimensional feature information including colour, texture, size, etc. from a single image for 3D reconstruction of objects. Unlike traditional 2D image feature extraction, achieving 3D reconstruction of objects requires the extraction and learning of 3D structural features. Therefore, successful models not only need to ensure that their predicted values are close to those of the real labels, but also that their predicted 3D models are similar to the real 3D Ground Truth in terms of structural features.

Our research goal is to allow the neural network to extract the depth information features of the fundus surface from the monocular fundus OCT top view, and to construct a 3D fundus surface model based on the predicted values, which requires that we need to predict the depth value of each pixel, and U-Net, as a deep convolutional neural network architecture for pixel-level image segmentation and semantic segmentation tasks, is an ideal model to achieve our research goal . There are several reasons for this:

- Pixel-level feature extraction: Its encoder-decoder architecture is well suited for processing fundus surface OCT medical images because it enables pixel-level depth information ingestion through multilevel feature extraction, where there is a lot of tiny structural and detail information in the fundus surface images, and the depth information extracted by the U-Net is very important for performing 3D reconstruction.

- Up-sampling operation of decoder: The decoder part of U-Net includes up-sampling operation which helps to reduce the feature maps extracted in the encoder to the same resolution as the input image. This is important for the 3D reconstruction task

because we need to reduce the feature information in the 2D image to 3D information in equal proportion.

- Previous successes: U-Net has achieved extensive success in the field of medical image analysis for implementing 3D reconstruction based on monocular images, such as mannequins, faces, and hand models. These research results provide strong support for the performance of U-Net in processing fundus images
- Trainability: Compared with other network structures (e.g., VGG19, ResXnet, etc.), the structure of U-Net is relatively simple and easy to be restructured and modified. This means that we can adapt the U-Net image segmentation model to depth-valued predictive regression model according to the specific fundus surface feature extraction and 3D model reconstruction needs.

## 5.2 Modify the U-Net image segmentation model to a regression model

U-Net models are commonly used for image segmentation tasks. In our study, we first trained the U-Net segmentation model using the DR-OCT dataset to obtain the corresponding mask labels for the CSCR dataset. However, achieving our research goal requires changing the U-Net network from a common image segmentation task to a regression task capable of performing prediction of depth values. To achieve this goal, we need to modify the network structure of U-Net. This requires an in-depth understanding of the difference between segmentation tasks and regression tasks, an understanding of how U-Net networks work, and how to adapt the model structure of U-Net networks accordingly.

### 5.2.1 Analysis of the working principle of U-Net network

When obtaining the mask labels corresponding to the CSCR data set, we introduced the special components of the U-Net network structure and learned that it can rely on its unique Encoder-Decoder structure to generate high-level feature information with complex semantics.

The encoder part of the U-Net network structure mainly implements the capture of input image context information. It is composed of multiple convolutional layers and pooling layers stacked. The convolutional layer is used to extract input image features, and the pooling layer is used to Reduce the resolution of the feature map obtained by the convolutional layer. The encoder reduces the details in the image by mapping the input image to a low-resolution feature representation, while extracting different levels of feature information. Each layer of the encoder can be regarded as an abstract representation of features, where low-level features include local information such as edges and textures, while high-level features include more abstract semantic information. These feature abstraction capabilities enable U-Net to capture the contextual information of an image, that is, the pixel values and features surrounding each pixel.

The decoder part of the U-Net network structure mainly maps the low-resolution feature map output by the encoder back to the image segmentation result. It consists of multiple upsampling operations and convolutional layers, and can gradually restore the segmentation result layer by layer. spatial resolution, and finally restore the low-resolution feature map to the resolution of the original input image. As shown in Figure 5.1.

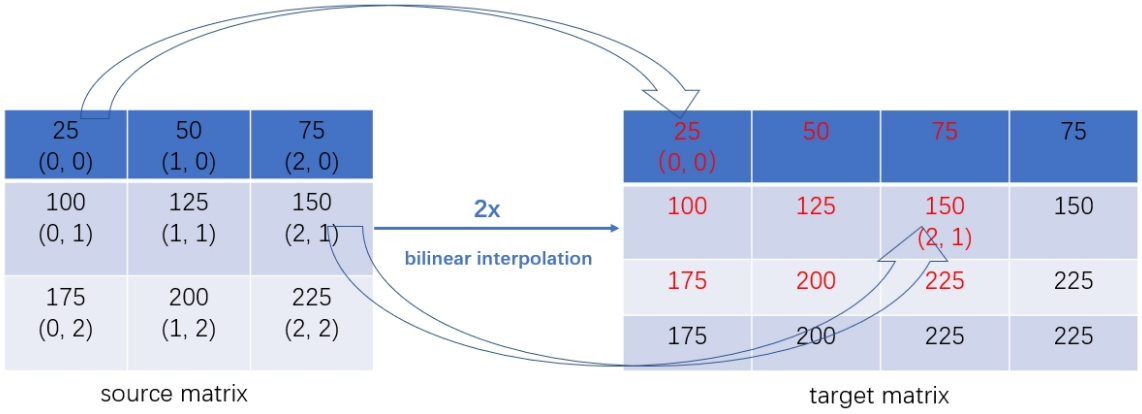


Figure 5.1: Bilinear interpolation upsampling restores image resolution

It should be especially noted that the U-Net network is not a simple input-output mode, but after each layer of the encoder ends, the feature map output by the layer is copied to the corresponding layer of the decoder to establish a skip connection or called U-shaped connection. This means that the decoder can use high-resolution feature information from shallow layers to compensate for and restore image details lost due to downsampling while performing upsampling and feature recovery. This direct connection method (As shown in Figure 5.2) helps It alleviates the problems of information loss and gradient disappearance, and provides a better information flow and feature reconstruction mechanism, allowing the network to more accurately retain details and contextual information when performing image segmentation. This structure has achieved good results in fields such as medical image segmentation that pays attention to image details.

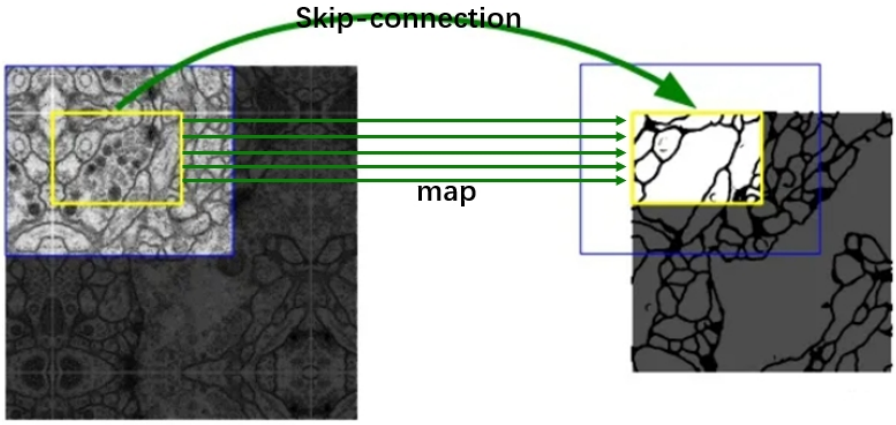


Figure 5.2: Skip-Connection implementation process

The high-resolution feature map with complex semantics output after the complete U-Net network will be input into the softmax function to convert each pixel of the feature map into a category probability pixel by pixel, resulting in a segmentation with the same size as the input image Probability map is then segmented according to different category thresholds, and finally completes the image segmentation task.

### 5.2.2 The difference between segmentation tasks and regression tasks

The main difference between U-Net in image segmentation tasks and depth value prediction regression tasks is the task type and output. Here are the key differences between them:

- Task type:

  In the image segmentation task, the goal of U-Net is to assign each pixel in the input image to a specific category or label, so its final output is a pixel-level mask label, where each pixel corresponds to the input image A pixel position in a pixel and is assigned to a category or label. Usually, we use these labels to represent different objects or parts of objects to achieve pixel-level semantic segmentation. The mask label of the CSCR dataset is also obtained by segmenting the OCT image of the fundus slice by distinguishing the difference between the background and the fundus label.

  In the depth value prediction regression task, the depth value prediction task we want to implement is more focused on the regression problem, which requires the U-Net network to estimate the depth or distance information of each pixel in the fundus surface OCT image, rather than dividing it into a certain specific categories or tags. This requires us to modify the final output type of the U-Net network from outputting a mask label with the same spatial resolution as the input image to a depth value map with the same spatial resolution as the input image, where each pixel contains a depth value.

- Loss function:

  In image segmentation tasks, classification loss functions like cross-entropy are often used to compare the difference between the model output and the ground truth.

  In the depth value prediction regression task, a regression loss function, such as mean square error (MSE)[Allen, 1971], is usually used to measure the error between the depth estimate of the model and the actual depth to measure the similarity between the model output and the 3D ground truth.

- The focus of feature extraction is different:

  The image segmentation task focuses more on the boundaries and shapes of different objects or object parts in the image, while the regression task requires more accurate prediction of the depth information of different pixel positions in the global OCT image of the fundus surface, rather than individual areas.

These differences highlight the adaptability of U-Net to different task contexts and the variations in output types. They also represent the key factors in how we successfully modified the U-Net network structure from a segmentation task to suit the regression task of predicting depth values.

### 5.2.3 U-Net network structure modification

Based on our understanding of the U-Net architecture and the differences between segmentation and regression tasks, we have identified the specific modifications required to adapt U-Net for our purposes.

Firstly, it can be observed from the source code that in a typical U-Net architecture, the common practice is to employ a 1x1 convolution operation with a number of output channels equal to the number of classes in the classification task. Consequently, each channel represents a distinct class, and each pixel within these channels contains the predicted probability for its corresponding class. In order to transform the output from predicting class probabilities to predicting depth values, we need to modify the number of output channels to be 1. This implies that the final output will no longer be a multi-channel probability distribution for different classes but a single-channel numeric value representing the regression prediction for each pixel, such as the desired retinal depth values in our research.

Furthermore, in multi-class classification tasks, it is customary to apply a softmax activation function to the final output. This function maps the model's output to a probability distribution over each class, ensuring that the probabilities sum to 1. However, in regression tasks, we no longer require this transformation into a probability distribution. Instead, we aim to directly obtain predicted depth values. (As shown in Figure 5.3) Therefore, we need to replace the softmax activation function with a linear activation function (also known as the identity activation function) or use no activation function at all. This allows us to obtain the final output of the U-Net network as the predicted depth values without any further transformations.

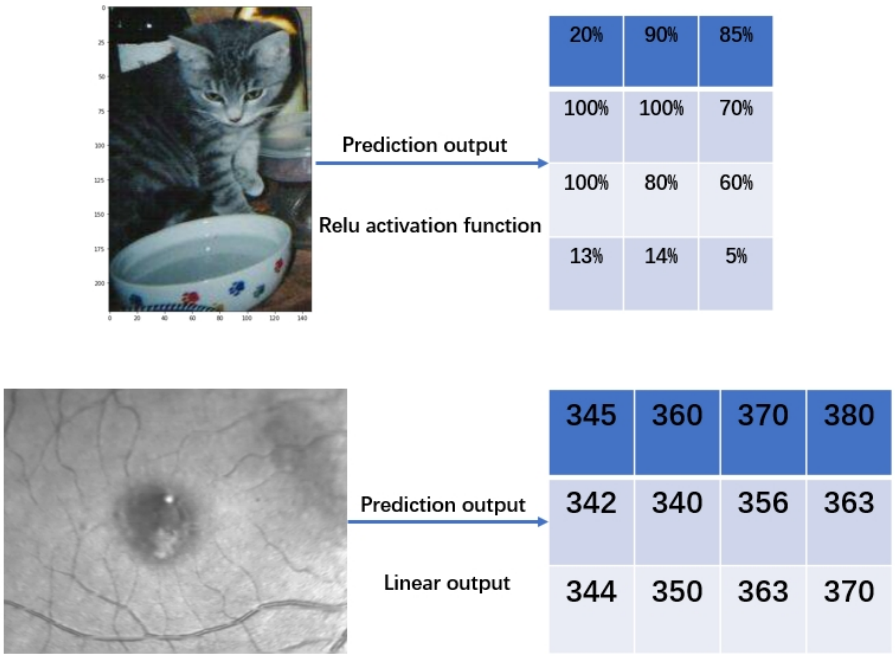


Figure 5.3: Comparison of classification task and regression task output

After completing the modification to the U-Net network model, we still need to pay attention to the training process of the model. Typically, in classification tasks, we usually use cross-entropy as the classification loss function to measure the difference between the model output and the true label.

The core idea of cross-entropy loss is to measure the difference between two probability distributions. In a classification task, we have two probability distributions:

The output probability distribution of the model: This is generated by the neural network model and is usually represented as P(y—x), where y represents the category and x represents the input sample.

Distribution of real labels: This is a one-hot encoding vector representing the real categories. In this vector, only the element corresponding to the true category is 1, and the remaining elements are 0.

The cross-entropy loss function is calculated as follows:

$$H(y, p) = -\sum_{i}(y_i \log(p_i)) \tag{5.1}$$

Where: H(y, p) is the cross-entropy loss. $y_i$ is the i-th element of the real label (one-hot encoding), which takes a value of 0 or 1. $p_i$ is the i-th element of the model's output probability distribution, representing the probability that the sample belongs to category i.

When optimizing a neural network model, the cross-entropy loss can be viewed as the relative entropy between the true distribution and the model output distribution, that is,

the model's estimation error of the true distribution's uncertainty. Therefore, minimizing the cross-entropy loss can make the model closer to the real distribution, thereby improving the accuracy of classification.

What we want to achieve is to modify the classification task to a regression task, and after converting U-Net into a regression task of depth value prediction, our training data will contain real depth value information. Therefore, our focus is no longer on the difference in probability distribution, but on the difference between the output predicted value and the real value. Here we need to change the loss function used during training to the mean square error function.

The mean square error loss is calculated as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{5.2}$$

in MSE: $y_i$ is the ith element of the real value. $\hat{y}_i$ is the ith element of the model's predicted value. n is the sample size.

The MSE loss function squares the difference between the real value and the model's predicted value, and then averages it, which can be regarded as a secondary measure of the difference. This also means that the loss function is very sensitive to outliers, and the predicted value is different from the true value. The larger the difference between , there will be a higher loss value, and for the sample with a smaller difference, the loss value will be lower.

## 5.3 Dataset preprocessing and augmentation

Due to limitations of currently available dataset sizes, our comprehensive CSCR_3D dataset contains only 32 retinal optical coherence tomography (OCT) images and their corresponding 3D ground truth labels. Given the relatively small size of this dataset, training a neural network on it carries a significant risk of overfitting. Therefore, in order to meet the training requirements of neural networks, we must adopt data preprocessing techniques to effectively increase the dataset size.

The dataset expansion process aims to increase the number of available data instances while adhering to sound academic principles. This data augmentation is critical to enhance the model's generalization ability and reduce the potential for overfitting.

### 5.3.1 Flip Expansion

Data flip augmentation is a classic data preprocessing technology that is widely used in the field of image processing. In our study, we adopted three common flipping operations, including vertical flipping, horizontal flipping, and 180-degree flipping, to expand the size of our original fundus OCT image data set by four times its original size.

They both aim to create variants of the original image to increase the diversity of the dataset. The following is a step-by-step analysis and rationale for these operations:

- Horizontal Flip: Also known as left-right flip. In horizontal flipping, the fundus OCT image is mirror flipped along the vertical central axis (usually the center line of the image). The implementation principle is to reverse the position of each pixel in the image, that is, copy the pixel data of the i-th row in the original image to the i-th row in the new image, but in reverse order.
- Vertical flip (Vertical Flip): Vertical flip is to flip the image along the horizontal axis, also known as flip up and down. The implementation is the same as horizontal flipping. The difference is that vertical flipping is the reverse sorting of the pixel data in column i in the original image.

Both of the above mirror flips can reduce the model's dependence on the location of specific structures in the image. For the fundus images we studied, artificially making some changes in the position of the fundus structures can improve the generalization ability of the model.

- 180 degree flip: This operation is to perform an angular rotation on the image, also known as inversion. 180-degree flip rearranges the pixel data in the image so that the image is rotated 180 degrees, and the effect is equivalent to first flipping horizontally and then flipping vertically.

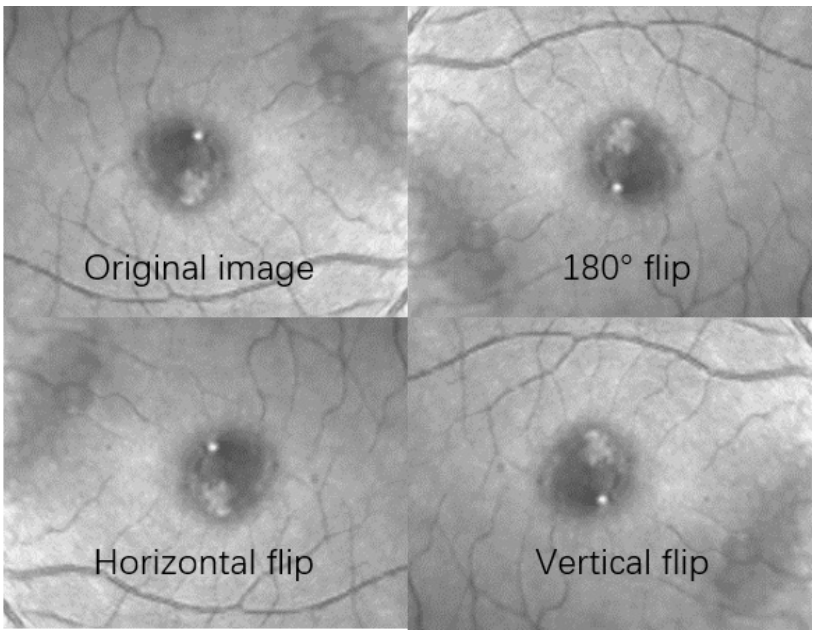


Figure 5.4: Original image flip examples

The purpose of these flipping operations (Results shown in Figure 5.4) is to increase the diversity of the dataset, allowing the neural network to be exposed to images of different

perspectives and characteristics. These transformations are usually reversible, so they do not change the semantic content of the image, but help the network generalize better to different situations.

### 5.3.2 Crop Expansion

Although we have obtained 128 training data images through flip augmentation, this is still not enough to meet the amount of data required to fully train the model. Therefore, we need to employ additional data preprocessing operations to further expand the size of the dataset. Our goal is to enable the model to predict the depth value of the entire fundus OCT image, so we not only focus on specific local areas, but also on the characteristics of the overall image. This is the basis on which we can further expand the data set through clipping operations.

In order to ensure that the image data corresponds to the label, all our image cropping operations must be applied to the label image. First of all, we already know that the size of a fundus surface OCT image is 1024x332. Here we want to crop it into 5 images of equal size, and ensure that the original image can be perfectly restored after these 5 images are spliced according to the cropping position. Therefore, we use sliding window cropping or overlap cropping.

The specific implementation steps are as follows:

- Starting from the upper left corner (0,0) of the original image, crop out the first 332x332 sub-image.
- When cropping the second image, we do not start cropping from 332, but slide the upper left corner of the cropping window to the right by a certain number of pixels to create an overlap, and then crop again.
- And so on, cropping out all required sub-images in turn. Each time we slide the window, make sure there is a portion of overlap so that continuous image information is captured.

This approach ensures image continuity and allows us to crop out any number of sub-images without losing information. However, attention should be paid to determining the sliding step size and cropping window size to avoid the sliding window not covering the entire original image after cropping is completed, resulting in the loss of the second half of the data.

Now we know that the size of the cropping window is 332x332, and the calculation formula of the sliding step is:

$$Stepsize = \frac{Originalimagesize - Cropsize}{Cropnumber - 1}$$

In short, through this series of data preprocessing operations, we successfully expanded the size of the data set and provided more samples for the training of the deep learning model, which is expected to improve the performance and robustness of the model. This is crucial for our depth value prediction regression task.

## 5.4 Model training process

### 5.4.1 Add a dropout neural network layer after the Encoder module

Given that the data set we have is relatively small and the U-Net network has powerful feature extraction capabilities, this can easily lead to overfitting problems. In the case of overfitting, the model will pay too much attention to the specific details and noise in the training data and ignore the general features, and our research goal is to predict the global image depth value. Therefore, we adopted a commonly used regularization technique, which is to introduce a Dropout neural network [Srivastava et al., 2014] layer after the Encoder module of U-Net to prevent the model from paying too much attention to specific parts of the data, effectively solving the problem of U-Net network in small data sets. Deep feature aggregation problem.

Dropout is a regularization method that randomly turns off neurons during the training process (The operating principle is shown in the Figure 5.5), the left half of the figure represents the process of feature transfer in the neural network layers, and the right half represents the process of feature transfer between layers after using dropout. It can be seen that in the process of feature transfer between layers, there are some units containing features are hidden. The dropout operation randomly hides some feature information, thereby reducing the cooperation between neurons and helping reduce the model's dependence on specific training samples. It should be noted here that the hidden units are random during each training process, which ensures that we will not permanently lose some features.

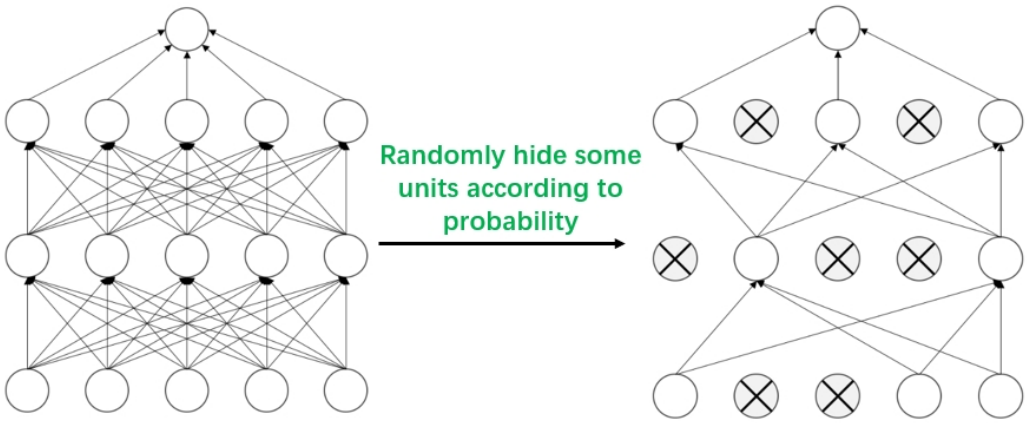


Figure 5.5: The working process of the dropout layer. Source: adopted from Website URL: https://blog.csdn.net/upupyon996deqing/article/details/124840237

### 5.4.2 Parameter settings

After completing the modification of the U-Net regression network model, we need to start setting the parameters of the network training. This is a crucial step in the deep learning model training process.

Initially, the dataset is partitioned into a training set and a test set, utilizing 80% and 20% of the data, respectively. To ensure the exclusive presence of a patient's data (comprising multiple augmented images) in either the training or test set, we have adopted a careful grouping strategy. Each patient's images are identified and grouped accordingly. Subsequently, a random selection process allocates 80% of patient groups to construct the training set, leaving the remaining 20% for the test set. This ensures that all images pertaining to a specific patient are exclusively present in either the training or test set, mitigating mutual interference. This partitioning strategy preserves dataset independence, enhancing the accuracy and generalization of the model. Following this, data augmentation is applied to the grouped dataset, ensuring augmented images do not concurrently appear in both the training and test sets.

Then we need to select an appropriate loss function. Here we use the mean square error as the loss function to measure the difference between the model's predicted depth value and the true value. During the training process, by minimizing the MSE loss, we can make the model better fit the training data, thereby improving the accuracy of depth value prediction.

Next, we need to choose a suitable optimizer. The optimizer is responsible for updating the weights and biases of the model to minimize the loss function. Commonly used optimizers include stochastic gradient descent (SGD), Adam, RMSprop, etc. When choosing an optimizer, you need to consider factors such as the complexity of the model, the size of the data set, and the speed of convergence during training. We finally chose Adam as an optimizer for adjusting weights. It has the characteristics of adaptive learning rate and can converge to the local minimum faster. Here, for the learning rate adjustment strategy during the training process, we use the optim.lr_scheduler.ReduceLROnPlateau scheduler, which is used to dynamically adjust the learning rate of the Adam optimizer based on the performance of the model. The parameters set are as follows:

The parameter 'min' of the scheduler means that it monitors the minimum value of the loss function.

- 'patience'=3 indicates that when the monitoring indicator does not improve within 3 consecutive epochs, the learning rate should be adjusted. If the loss function does not decrease within 3 consecutive epochs, then the learning rate will be reduced.
- 'factor'=0.1: This parameter indicates that when the learning rate needs to be adjusted, the learning rate is multiplied by this factor. Here, if the monitoring metric does not improve, the learning rate will become 0.1 times the current learning rate, which is reduced to 1/10 of the original.
- 'min_lr'=0.00001: This parameter represents the minimum value of the learning rate. If the learning rate is lowered than this value, it will no longer decrease.
- 'cooldown'=1: This is a scheduling parameter that indicates the number of periods to wait after the learning rate is adjusted to avoid continuous triggering of learning rate adjustments due to rapid loss function decline in the early stages of model training. Here, wait for one epoch before triggering the learning rate adjustment again.

After adjusting the module that monitors the loss function of the model, if the loss function does not improve for 3 consecutive periods, the learning rate is reduced to 1/10 of the current learning rate until the learning rate is reduced to the specified minimum value. This adjustment method adjusts the learning rate more finely than the traditional method of decreasing the learning rate with each epoch. It is more adaptable to various situations in the decrease of the loss function and can improve the performance and generalization ability of the model.

Finally, we need to define evaluation metrics to measure the performance of our modified U-Net regression model on the task of predicting depth values. In regression tasks, various evaluation indicators are usually used to measure the performance of the model, including mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), etc. For our task, we finally chose mean square error (MSE) as the main evaluation metric. The main reason is that it is relatively insensitive to outliers. This means that if there are some depth value outliers in the data, MSE will reduce their impact on the evaluation results, making the model more robust. Secondly, MSE provides a secondary measure of the depth value prediction error, which can more clearly reflect the difference between the predicted value and the true value.

Once we define these training parameters and evaluation metrics, we can start training the U-Net regression model. During the training process, the model gradually adjusts the weights and biases based on feedback from the loss function to minimize the loss and improve the accuracy of depth value prediction. As training progresses, we monitor changes in evaluation metrics to evaluate the performance of the model and make hyperparameter adjustments if necessary. It should be noted that this training process usually requires a certain amount of time and computing resources, and the selection of hyperparameters will have an important impact on the training effect. Therefore, in subsequent experiments, we also need to conduct some ablation experiments and comparisons of hyperparameter combinations to determine the best model configuration.

By carefully setting training parameters and appropriate data augmentation, we can expect to obtain a high-quality depth value prediction model to provide strong support for fundus OCT image analysis tasks.

### 5.4.3 Ablation Experiment

In order to achieve optimal depth value prediction, we must have a deep understanding of the impact of each component in network training and their weights on overall performance. This requires continuous trial and analysis to determine the most appropriate configuration of network components and weights. To achieve this goal, we employed an ablation experimental approach.

Ablation experiments are an important method in scientific research and experimental studies, the main purpose of which is to understand their impact on model performance by gradually excluding or reducing components, parameters, or characteristics of the model. In our study, ablation experiments can be applied in the following three aspects:

- Component analysis: We can use ablation experiments to gain insights into the

contribution of individual components or modules in the model to overall performance. By gradually disabling or adjusting certain components, such as different layers in a neural network or features in a model, we are able to identify which components are critical to the task and which contribute positively to performance improvements. This analysis is very useful for the dropout layer we added independently, because based on the theoretical analysis of the data set, we believe that the dropout layer can reduce the overfitting problem caused by the small size of the data set. However, to verify its effectiveness, it needs to be verified through ablation experiments.

- Parameter impact: Our U-Net network and loss function usually contain many adjustable parameters. Through ablation experiments, we can evaluate the impact of each parameter on model performance, determine which parameters are most sensitive to performance, and which parameters can be adjusted to further improve the model effect.

- Hyperparameter tuning: Ablation experiments help us determine the optimal hyperparameter configuration for the model. By progressively eliminating different hyperparameter settings, we can find the most efficient combination of hyperparameters for a specific task. In our study, ablation experiments are also used to adjust the weight ratio of each part of the composite loss function. We plan to add other loss functions in subsequent studies to build more complex composite functions, and the weight ratio between these functions will be precisely regulated through ablation experiments.

But ablation experiments also have their corresponding limitations, because ablation experiments often involve removing components or parameters from complex models to observe their impact on performance. This simplification may not fully reflect the situation in real applications, where individual components and parameters may interact with each other rather than simply add up. Therefore, the results of ablation experiments may sometimes be too idealistic.

Beyond this, ablation experiments typically focus on the effects of one or a few specific factors. However, in practical problems, many factors may interact, making it difficult to fully understand through a single ablation experiment. This can lead to a poor understanding of overall system behavior.

Considering time and computing costs, conducting large-scale ablation experiments may require a large amount of computing resources and time. Stepwise exclusion of components or parameters may require training and evaluating the model multiple times, which may increase research costs.

Despite these limitations, ablation experiments are still a powerful tool in our deep learning research, which can help us deeply understand the behavior and performance of the model and optimize model design. Therefore, it is necessary to conduct ablation experiments to improve the effect of the network model. After all, everything is ultimately The quality of the method must be determined based on the effect of the network model.

## 5.5 Result analysis

### 5.5.1 Training Program

After completing the construction of the U-Net model and the design of the ablation experimental plan (As shown in Figure 5.6), we began to formally train the network to predict depth values. We have designed the following plans for ablation experiments:

- A - No dropout layer is added, and a separate MSE loss function is used for network training.
- B - No dropout layer is added, and a separate SML1 loss function is used for network training.
- C - Add a dropout layer and use a separate MSE loss function for network training.
- D - Add a dropout layer and use a separate SML1 loss function for network training.
- E - Add a dropout layer and use MSE and SML1 composite functions according to the weight ratio for network training (here you need to test the weight ratio multiple times to find the best ratio).

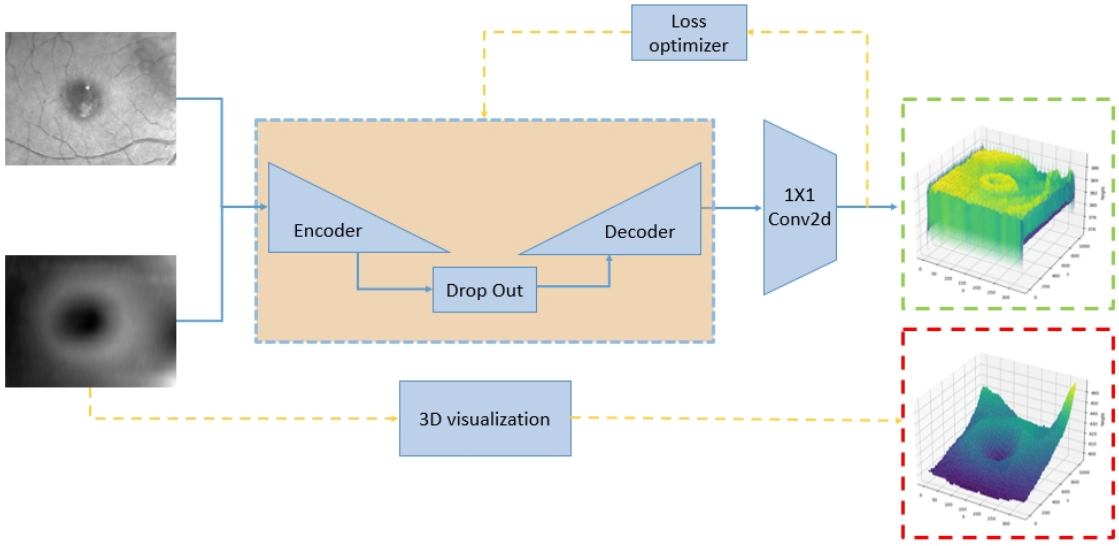


Figure 5.6: The whole process of U-Net regression model training

The reason why we chose to introduce the SML1 loss function is mainly to supplement the shortcomings of the MSE (Mean Square Error) loss function, especially in the regression depth value prediction task. MSE is sensitive to errors, which means that larger errors have a significant impact on the loss. If the dataset contains errors, MSE may cause poor model

performance. In contrast, SML1 Loss is less sensitive to errors because it penalizes a linear penalty on large errors rather than a quadratic penalty. It provides smoother gradients during optimization and is more robust in situations where the data contains errors or extreme values.

$$SML1Loss = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{\max(|y_i|, \tau)} \tag{5.3}$$

In SML1 mathematical formula:

- N is the total number of samples, which refers to the total number of pixels in the OCT image.
- $|y_i - \hat{y}_i|$ represents the absolute error between the true value and the predicted value.
- $\max(|y_i|, \tau)$ is used to calculate the scale factor, where $\tau$ is a small positive number, often used to prevent the denominator from being zero.
- The key characteristic of the SML1 loss function is that its numerator contains the absolute error, while the denominator includes a scaling factor $\max(|y_i|, \tau)$. This scaling factor makes the SML1 loss function insensitive to the scaling of the data. When there is a significant difference between the true value $|y_i|$ and the predicted value $|\hat{y}_i|$, the absolute error in the numerator dominates, while the influence of the scaling factor decreases when the differences are small. This property makes the SML1 loss function perform well in the presence of outliers or data with different scales, as it is not overly sensitive to large errors.

$$Smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if|x| < 1 \\ |x| - 0.5 & otherwise, \end{cases}$$

Figure 5.7: Mathematical explanation of smooth loss function

First, the SML1 loss function is scale invariant, which means it is not affected by data scaling. In contrast, the MSE loss function penalizes large error terms more heavily because it includes a squaring operation, which makes it very sensitive to outliers or outliers. SML1 reduces the impact of large error terms on the loss by using absolute values instead of squared differences to measure errors. Therefore, SML1 performs better on data sets with outliers because it is not too disturbed by extreme values.

Second, SML1's absolute value operation generally results in a smoother loss function surface, relative to the squared term of MSE, which helps to find the global minimum more easily, thereby enhancing the stability of training. MSE may introduce multiple local

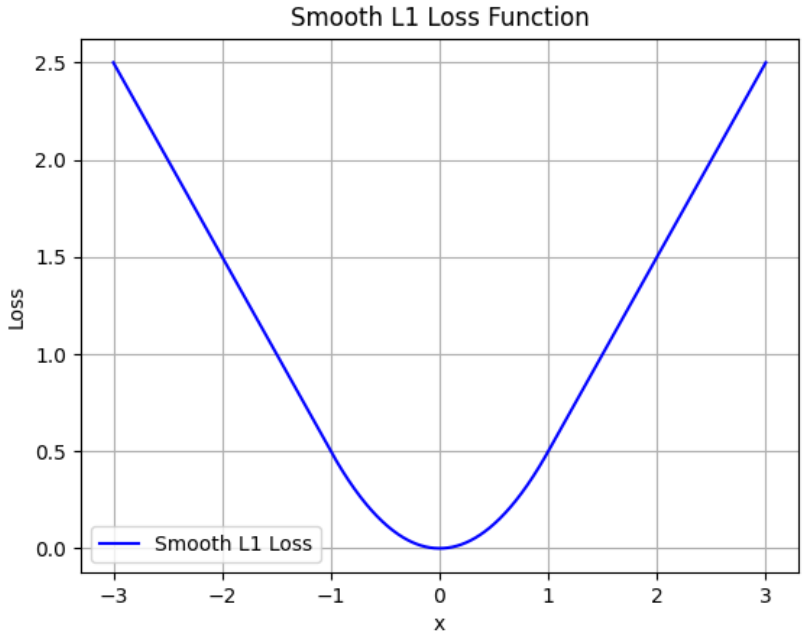


Figure 5.8: Smooth loss function curve

minima in the loss function surface, leading to instability in training, while SML1 makes it easier to avoid this situation.

Furthermore, the MSE loss function assumes that the error follows a Gaussian distribution, which may not be appropriate in some cases, especially when the error distribution does not satisfy normality. In contrast, SML1 makes looser assumptions about the error distribution and is therefore more suitable for different types of data.

To summarize, the SML1 loss function is generally more robust in regression tasks, especially when outliers are present or scale invariance is required. It complements the MSE loss function and helps improve the performance and stability of the model. Therefore, we choose to introduce the SML1 loss function into our ablation experiments in order to obtain better depth value prediction effects.

### 5.5.2 Experimental Analysis of single loss function

After conducting the first round of training according to Plan A (3D model shown in Figure 5.9), we have made certain progress. This is a positive sign for such a small dataset. The regression U-Net network showed excellent ability to extract some depth value information from a single fundus image, and the difference between the predicted value and the true value was only 88.98 (the difference in the first epoch was 323892462) . This indicates that the model has learned to predict depth information from fundus images to some extent.

However, when we further observe and visualize the predicted 3D model, we find that

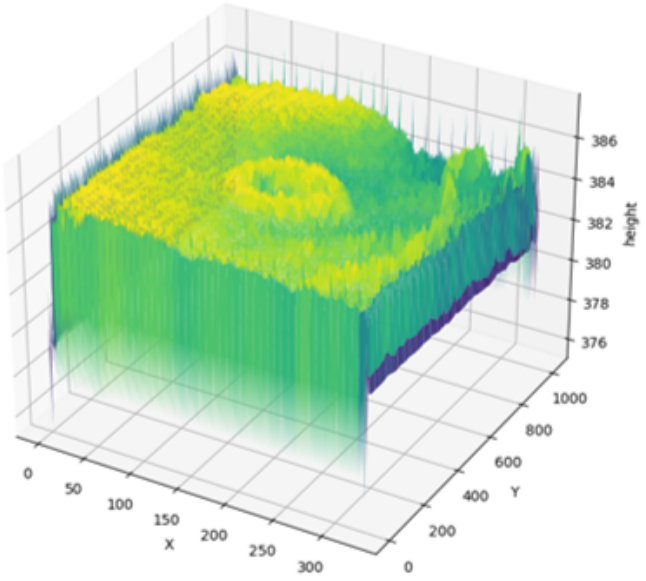


Figure 5.9: Plan A predicts fundus 3D model

the similarity between the predicted model and the real 3D model does not seem to be as close as it should be with a difference value of 88.98. We believe that the main reason for this problem is that the U-Net model over-learns the significantly changing feature information in the center, but ignores the detailed feature information in the edges and surrounding areas. This situation may be caused by the small size of the data set and limitations of the U-Net architecture or problems with network parameter settings. In U-Net, the information transfer between encoder and decoder is usually implemented through skip connections, but they may not be enough to effectively capture global and local features. In addition, factors such as the depth of the network, the number of feature maps, and regularization may affect the performance of the model.

This analysis highlights the challenges of training deep learning models on small-scale datasets, while also highlighting the importance of further research and tuning to improve model performance. This also highlights the complexity and challenge of the medical image analysis task we studied, namely fundus OCT image analysis. Through continuous experimentation, analysis, and improvement, we can gradually improve the model to achieve more accurate depth value predictions.

Compared with plan A, the training effect of plan B is very close. The observation that solution B using the SML1 loss function has a predicted value error of 81.72 suggests that in this particular task, the SML1 loss function and the MSE loss function perform almost identically when used alone.

However, this also inspires us to construct a composite loss function by reasonably configuring the weight ratio of the SML1 loss function and the MSE loss function to further

improve the performance of the model. This composite function can comprehensively utilize the scale invariance of the SML1 loss function and the squared difference characteristics of the MSE loss function to better capture the key features of the depth value prediction task.

This finding inspired our subsequent research directions, especially on how to determine the optimal weight ratio between SML1 and MSE, and how to effectively combine them into a composite loss function. This is also closely related to the ablation experiments we mentioned before, because through ablation experiments, we can evaluate the performance of the model under different weight ratios and find the best combination to achieve more accurate depth value prediction.

In subsequent experiments C and D, we added the dropout layer to the training process of the MSE and SML1 loss functions respectively. The results showed that the predictive value of the MSE loss function dropped to 54.54, while the predictive value of the SML1 loss function increased to 93.84. Based on further analysis, we think this phenomenon may be due to the impact of the data distribution of our data set, because the MSE and SML1 loss functions have different focuses. Although adding the dropout layer helps alleviate the overfitting problem, it still It is inevitable that detailed information will gradually be lost as the number of neural network layers progresses. The lack of this information may cause individual larger error terms, but there will not be a large number of outliers. The MSE loss function has a relatively larger penalty for large error terms than SML1, which makes it more robust in the presence of individual outliers or noisy data. SML1 is not sensitive to large error terms, so when the data distribution of our data set is relatively even without a large number of outliers, it may not perform as well as MSE in certain situations.

In addition, the hyperparameter settings of the dropout layer itself will also have a certain impact on the training effect of the neural network model, including the dropout dropout rate and the settings of other training parameters. A dropout rate that is too large may also lead to large changes in the data distribution of the data set, which in turn affects the adaptability of the loss function to the data distribution. This requires trying different hyperparameter configurations to find the best combination. This is true for SML1 and The MSE loss function is applicable.

In the preceding discussion, we primarily explored network training strategies based on individual loss functions. However, in Experiment E, we ventured into a more challenging approach by combining the MSE (Mean Squared Error) and SML1 (Scale-Invariant Mean Absolute Error) loss functions into a composite loss function. This endeavor was motivated by our recognition of the limitations of single loss functions in specific scenarios.In Experiments A and B, different loss functions exhibited similar performance when handling data distributions of the same nature. However, in Experiments C and D, different loss functions demonstrated varying performances when dealing with different data distributions and task characteristics. This led us to the insight that combining these loss functions could be a promising strategy to achieve favorable outcomes in different aspects.

Our objective was to optimize the composite loss function by adjusting the weight ratios between MSE and SML1, enabling it to harness the strengths of both loss functions. This approach aimed to further enhance the training efficiency and performance of the neural network model.

### 5.5.3 Experiments to find the weight ratio of the composite loss function

Ensemble methods are widely used in neural network training, such as random forest, which is a typical method based on ensemble learning. It improves the performance of the model by combining the prediction results of multiple decision trees. Similarly, loss functions in deep learning can also combine different loss functions within a single model through composite integration to guide the model to learn different aspects of knowledge or optimize different goals. The main loss function compound methods are as follows:

- Weighted combination of loss functions:

  In neural networks, multiple loss functions can be combined according to certain weights to form a composite loss function. In our research, we combine the MSE and SML1 loss functions. MSE focuses on the regression task of continuous data. By appropriately adjusting the weights of the SML1 loss function, we can balance the influence of different loss functions in multi-task learning and improve the multi-tasking ability of the model. Feature.

- Adversarial loss function:

  In adversarial training, two competing loss functions are usually used. For example, the generator and discriminator in generative adversarial networks (GANs) use different loss functions. The generator seeks to fool the discriminator, and the discriminator seeks to accurately classify real and fake data. This competitive dynamic often results in the generator generating more realistic data. This combination of adversarial loss functions can also be used as a compound loss in deep learning.

- Loss function adaptation:

  Some methods adapt to different characteristics of the data by dynamically adjusting the weight or shape of the loss function. This kind of adaptation can also be regarded as a collection of multiple characteristics. For example, Focal Loss performs well in target detection. It reduces the weight of easy-to-classify samples and increases the importance of difficult-to-classify samples when the samples are imbalanced.

These examples illustrate that in neural network training, the combination of loss functions can be used to take full advantage of different loss functions and improve model performance and robustness. The weighted composite loss function used in our study uses the weighted composite formula as follows:

$$\arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left( \alpha (y_i - \hat{y}_i)^2 + \beta \frac{(2\mu_{y_i}\mu_{\hat{y}_i} + c_1)(2\sigma_{y_i}\sigma_{\hat{y}_i} + c_2)}{(\mu_{y_i}^2 + \mu_{\hat{y}_i}^2 + c_1)(\sigma_{y_i}^2 + \sigma_{\hat{y}_i}^2 + c_2)} \right) \tag{5.4}$$

$N$ is the number of training images; $y_i$ is the ground truth depth value; $\hat{y}_i$ is the predicted depth value; $\mu_x$ denotes the mean of $x$; $\sigma_x$ denotes the standard deviation of $x$; $c_1$, $c_2$, $\alpha$, $\beta$ are tunable parameters.

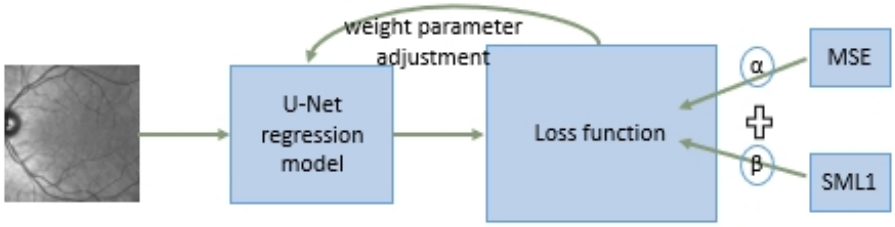


Figure 5.10: Composite loss function training process

In Experiment E, we chose to use the simplest weighted combination method to control the weights of different loss functions by adjusting the values of $\alpha$ and $\beta$ so that they have different effects on the model during training. In our research, we have also considered using cascade or parallel network methods or neural network adaptive weight learning methods to regulate the proportion of $\alpha$ and $\beta$ parameters, but these two methods each have their own drawbacks.

For the cascade or parallel network approach, it requires designing two independent neural networks, one using the MSE loss function and the other using the SML1 loss function. Their outputs can then be cascaded or paralleled together to form the final prediction. Although this method is theoretically very flexible and can better handle the output of different loss functions, it is not applicable in our specific task. Since we only have one data set and one target task, simply training two networks would double the computational cost with about the same effect as using two loss functions alone, so we decided not to use this approach.

For the neural network adaptive weight learning method, it is a very powerful weight learning technique that can learn dynamic weights through the neural network to adaptively adjust the weights of the two loss functions. This can be achieved by designing additional network layers so that the two weight parameters $\alpha$ and $\beta$ become hyperparameters of the network layer and combining them with the characteristics of the input data to achieve adaptive adjustment. However, this method of autonomously regulating weights will lead to increased computational costs and often requires more training data to ensure that the network finds the corresponding weight balance. This method may not be as good as when data is scarce or very noisy. Simple fixed weights work. And this will make the model's decision-making process more opaque and reduce the interpretability of the model.

Therefore, after considering both methods, we finally chose the simplest weighted combination method to control the weight of the loss function by manually adjusting the values of $\alpha$ and $\beta$ (As shown in Figure 5.10). Although this approach is relatively more intuitive and easier to implement, in some cases more experimentation and tuning may be required to find the optimal weight settings. This decision was made after weighing factors such as computational cost, model interpretability and mission requirements.

In our specific implementation plan, we chose to use the MSE loss function as the main loss function and the SML1 loss function as the auxiliary loss function to ensure the stability

of model training and reduce the impact of outliers on the overall composite loss function. Influence. We divided the experiment into 40 groups, in which the value of $\alpha$ was fixedly set to 1, while the value of $\beta$ gradually increased from 0 to 2 with a step size of 0.05. Each set of experiments was conducted for 100 rounds of training, and 3 sets of training weight parameters with the best results were selected. We used 3D visualization methods to compare and observe the training effects of different parameter combinations.

However, over many epochs of training, we noticed some particularly large outliers, with values exceeding 10,000. The occurrence of these outliers will cause the entire network to completely collapse in this round of training. Therefore, to deal with this situation, we introduce a threshold, i.e. (0, 496), every time we obtain a prediction value to avoid occasional outliers from ruining the entire training process. The purpose of this strategy is to ensure that the model is robust to outliers during the training process and prevent them from having too great an impact on the overall training. The implementation of this strategy helps to improve the stability of the model and ensure that the model can better adapt to different parameter combinations.

The data tables and curves we finally obtained from training are as Table 5.1 and Figure 5.11.

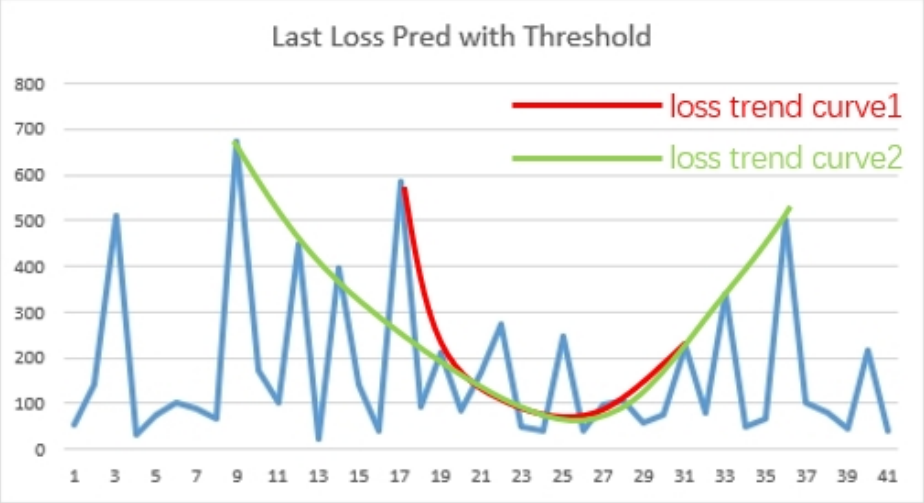


Figure 5.11: Comparison curve of loss prediction value with different weight ratios

According to the comparative analysis of the table and the curve (As shown in Figure 5.12), we can observe that the final loss prediction value shows a standard gradient downward trend, but starts to rise when approaching the convergence point. By carefully observing the loss trend curve 1 and curve 2, we can initially determine that the optimal weight parameter $\beta$ should be between 1.245 and 1.275. In order to determine the optimal value more precisely, we further narrowed the step size, setting it to 0.005, and conducted a series of experiments to find the most suitable $\beta$ weight parameter.

Finally, we determined that the combination with $\alpha$ of 1 and $\beta$ of 1.25 achieved the best results in the 51st round of training, with a predicted value of 29.703. We saved the weight

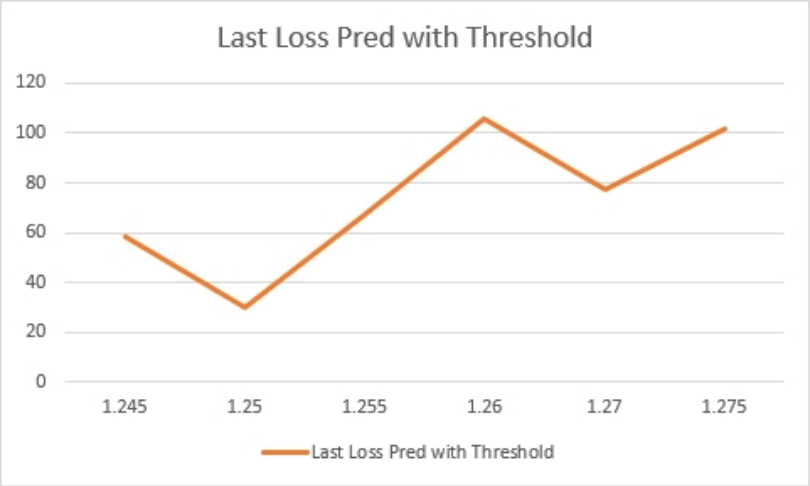


Figure 5.12: Comparison curve2 of loss prediction value with different weight ratios

parameters this time as "Best_checkpoint_epoch51.pth". This set of weight parameters was gradually adjusted through multiple rounds of ablation experiments. After repeated manual adjustments to the weight ratio of the $\alpha$ and $\beta$ parameters, we successfully reduced the loss to 29.08, which was a significant decrease compared to the original 81. This shows that our model is closer to the real image label in depth value prediction, which further validates the effectiveness of our method.

At the same time, the following is the final table of our ablation experiment, showing the predicted values obtained by different U-Net module combinations, as well as the finally selected best parameter combination on Table 5.3.

This table gradually shows the entire process of our ablation experiment. It can be seen that as we gradually modify the network structure and loss function with scientific rigor, the final predicted loss value has been steadily declining. This marks that our model has undergone a series of fine optimizations and adjustments in the depth value prediction task of monocular images, continuously approaching the relationship between real images and 3D scenes. Through this series of ablation experiments, we gradually locked in the best parameter combination, and generated the corresponding 3D visualization model through the best weight combination:

### 5.5.4 Analysis and Conclusion

So far, we have achieved certain research results. Observing the 3D model predicted from monocular fundus images obtained by our E-scheme (As shown in Figure 5.13), we can see that the optimal weight ratio trained U-Net model performs well in extracting central salient features and partially extracts the 3D ground truth center left Half of the marginal information shows a trend of high, low, and then higher (As shown in Figure 5.14). However, the feature extraction effect for the right half still needs to be improved, which is a direction

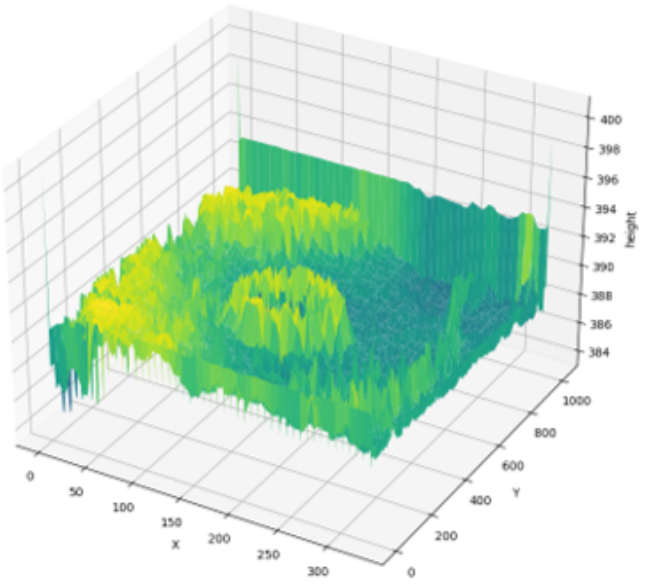


Figure 5.13: Plan E predicts fundus 3D model

that requires further research and optimization.

Combining the predicted 3D models obtained by Plan A and Plan E, we can find the inclined plane trend with the 3D ground truth. The predicted 3D models of both are still in the horizontal plane, which means that even the improved U-Net model is still It is impossible to learn the structural features of 3D ground truth from the monocular fundus image, and it is still only a simple extraction of 2D feature information. This shows that although our U-Net model has achieved certain improvements in extracting 2D feature information, requiring it to learn 3D structural features from monocular fundus images is still a very challenging task.

As for why the verified MSE prediction depth value error has been greatly reduced, our analysis is because the predicted 3D depth value is close to the 3D ground truth depth value in the overall trend (As shown in Figure 5.15). However, the part with relatively poor feature extraction ability is closer to the 3D The average depth value of the ground truth, so its error should be seen as reduced relative to the average depth value of the label, but this does not mean that it has learned the 3D structured features of the label.

In this stage of research, we have made a series of progress in the depth value prediction task, but we also face some challenges. Our method utilizes the design and optimization of the deep learning model U-Net network and composite loss function to provide a promising approach for 3D depth value prediction of monocular fundus images. However, we also identify directions for future research and improvements. We need to explore more deeply the common structural features in fundus images and input these features into the U-Net model together with monocular fundus images. This will enable the model to combine structural features during feature extraction to better learn and predict 3D models, making

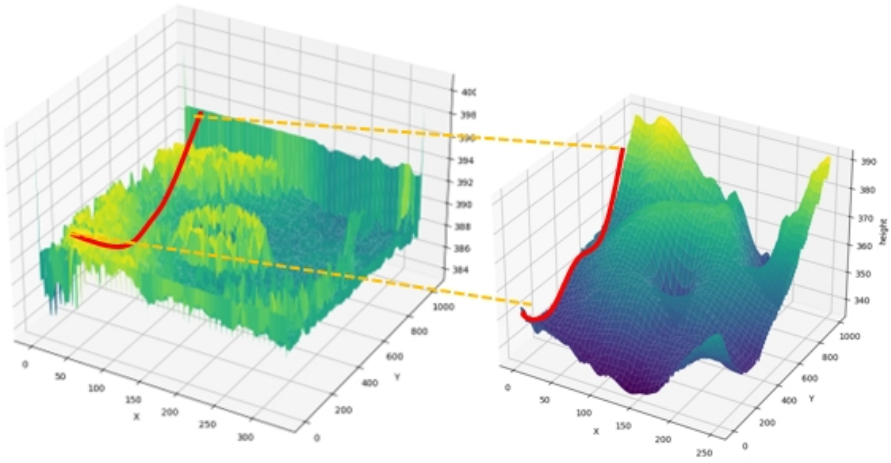

Figure 5.14: Edge feature learning

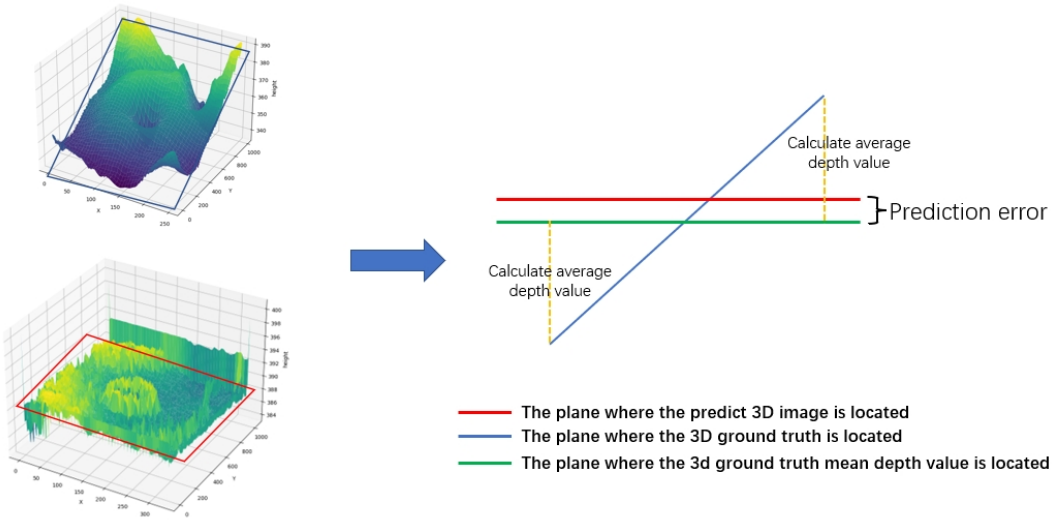


Figure 5.15: Explanation of reduced error in predicted depth values

them closer to 3D ground truth. This improvement is expected to improve the performance of U-Net networks in 3D model prediction.

In summary, our research provides a promising starting point for the field of 3D models for predicting depth values from monocular fundus images, and also points out the direction of future research aimed at further improving the prediction effect.

Table 5.1: Comparison table of loss prediction values for different weight ratios

| Step(0.05) | Epoch | Last Loss Pred with Threshold |
|---|---|---|
| 1 | 71 | 54.54070282 |
| 2 | 68 | 140.0891266 |
| 3 | 64 | 510.9633789 |
| 4 | 7 | 33.11423874 |
| 5 | 7 | 76.83986664 |
| 6 | 31 | 103.9022903 |
| 7 | 5 | 87.6574173 |
| 8 | 6 | 65.116539 |
| 9 | 24 | 672.6921387 |
| 10 | 70 | 172.3680115 |
| 11 | 4 | 102.9262238 |
| 12 | 36 | 449.5148926 |
| 13 | 3 | 23.71237946 |
| 14 | 15 | 394.0952759 |
| 15 | 3 | 142.9620819 |
| 16 | 3 | 42.38488007 |
| 17 | 15 | 584.9703979 |
| 18 | 100 | 95.14992523 |
| 19 | 4 | 213.114212 |
| 20 | 7 | 85.83585358 |
| 21 | 1 | 168.0022278 |
| 22 | 92 | 271.7408447 |
| 23 | 17 | 49.78013611 |
| 24 | 5 | 41.10489655 |
| 25 | 1 | 247.4756622 |
| 26 | 5 | 40.82254028 |
| 27 | 67 | 96.008461 |
| 28 | 94 | 105.7719727 |
| 29 | 5 | 56.02663422 |
| 30 | 51 | 74.1809845 |
| 31 | 13 | 227.0072937 |
| 32 | 4 | 81.3144989 |
| 33 | 65 | 338.4442444 |
| 34 | 2 | 48.83658218 |
| 35 | 50 | 66.39360809 |
| 36 | 3 | 499.8345032 |
| 37 | 87 | 100.7852707 |
| 38 | 5 | 79.73686981 |
| 39 | 7 | 45.86663437 |
| 40 | 96 | 215.1383057 |
| 41 | 5 | 41.15210724 |

Table 5.2: Test for optimal weight ratio of $\beta$ parameters

| $\beta$ | Epoch | Last Loss Pred with Threshold |
|---|---|---|
| 1.245 | 24 | 58.573 |
| 1.25 | 51 | 29.704 |
| 1.255 | 28 | 66.924 |
| 1.26 | 15 | 105.943 |
| 1.27 | 21 | 77.472 |
| 1.275 | 23 | 101.837 |

Table 5.3: Final results of Ablation Experiment

| Exp. | Setup | MSE (validation) ↓ |
|---|---|---|
| A | U-Net + Loss_MSE | 88.98 |
| B | U-Net + Loss_SML1 | 81.72 |
| C | U-Net + Dropout + Loss_MSE | 54.54 |
| D | U-Net + Dropout + Loss_SML1 | 93.84 |
| E | U-Net + Dropout + ($\alpha$*Loss_MSE + $\beta$*Loss_SML1) | 29.71 |

# Chapter 6

# Template-based 3D Surface Estimation from Monocular Images

## 6.1 Motivation for building a generic template for fundus 3d modelling

Based on our previous research and related results in the field of 3D reconstruction, we can conclude the following: unlike traditional 2D image feature extraction, realising the 3D reconstruction task for monocular fundus images requires not only extracting 2D depth-valued feature information, but also learning and capturing real-world 3D structural features [Jebara, Azarbayejani, and Pentland, 1999].

3D model structural features usually refer to information such as the 3D shape, geometric structure, and topological relationships of the target object or scene. These features include, but are not limited to, the object's surface shape, pose, edges, curvature, surface normals, volume, corner points, planes, bumps, etc. In our study of monocular fundus images, based on 3D truth analysis, it can be obtained that the 3D structural features of the fundus may involve the shape of the eyeball, the hierarchical structure of the retina, the distribution of the vascular network, the optic cups and discs of the retina, and the anomalous protruding lesions caused by fundus diseases. By capturing and learning the regular features (i.e., the feature regions that are present in every eyeball) from these features, we can more accurately restore the 3D scene corresponding to the fundus image and achieve a more accurate 3D reconstruction, instead of stacking the feature information in a 2D horizontal plane.

The research direction of 3D model reduction based on monocular images has made remarkable progress since it was proposed. A parametric 3D model of the human body was proposed in the SMPL(Loper et al., 2015) paper, which greatly advanced the research in this area by modelling shape and pose on top of the base human model. The SMPL models the human body as a base model of the human form and then captures the features of different human bodies through deformation. This deformation is achieved through PCA (Principal Component Analysis), which results in low-dimensional parameters that portray the shape, often referred to as shape parameters. Meanwhile, the SMPL model uses the

concept of a kinematic tree to represent the pose of the human body, where the rotational relationship between each joint point and its parent node can be represented by 3D vectors. These local rotation vectors ultimately constitute the pose parameters of the SMPL model. The uniqueness of this approach is that it can accurately simulate the muscle stretching and contraction of the human body in different poses, thus avoiding the problem of surface distortion during motion and providing a more accurate morphological description for human modelling.

In fact, although the SMPL model has been very successful in modelling the human body, it has some limitations in accurately modelling detailed models. This limitation is mainly manifested in the portrayal of human hand structures and gestures. The relatively small number of pixels occupied by the hand in the whole human body image, especially in full or half body images, makes it difficult for SMPL to accurately distinguish hand movements and details.

To overcome this limitation, Romero, Tzionas, and Black, 2017 proposed the MANO hand parametric model in 2017. This model is dedicated to fine-grained hand modelling, it contains 78 vertices, 1538 faces and constructs a complete hand skeleton, which can also be referred to as a forward dynamics tree, based on 16 keypoints as well as 5 points from the vertices at the fingertips of the fingers. As shown in Figure 6.1.

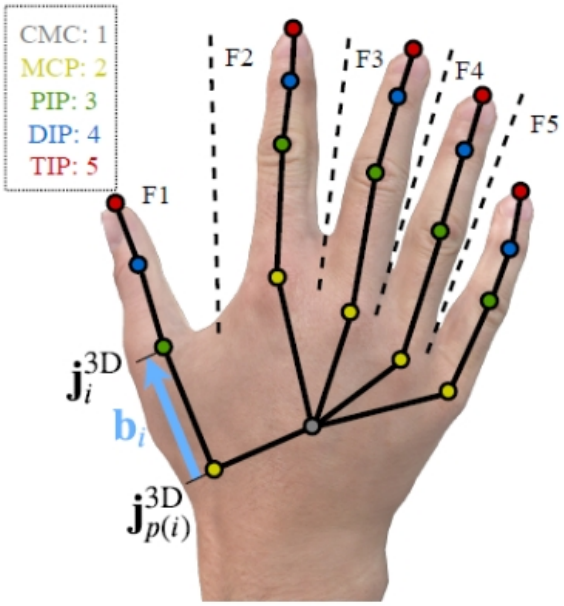


Figure 6.1: Joint skeleton structure. Source: Web URL: https://blog.csdn.net/g11d111/article/details/115539407

The role of the MANO model is equivalent to introducing an intermediate layer in the process of extracting the 3D pose from the input image, which acts as a transition representation or a powerful layer of a priori information. This allows the model to better handle occluded and low-resolution images, thus improving the ability to accurately model

hand structures and gestures.The introduction of the MANO model has greatly contributed to the field of 3D modelling of hand structures and gestures, allowing us to better reconstruct the complex movements and details of the hand, providing strong support for applications in areas such as hand pose recognition, gesture control, and virtual reality.

With the significant influence of MANO model in the field of hand 3D model construction, the number of MANO-based hand applications has been increasing, among which the $S^2$HAND 3D hand reconstruction network is in line with our research direction.

Due to the varied hand configurations and depth ambiguity, in order to reliably reconstruct 3D hands from monocular images, most state-of-the-art methods rely heavily on 3D annotations during the training phase, but the cost of obtaining 3D annotations is high. Y. Chen et al., 2021 proposes a self-supervised 3D hand reconstruction network, $S^2$HAND, that is capable of jointly estimating poses, shapes, textures, and camera viewpoints in order to alleviate the network's training dependence on 3d annotated training data. dependence. The model is stripped down from the MANO parametric hand model, which obtains geometric cues from the input image via easily accessible 2D detection keypoints, and exploits the consistency between 2D and 3D representations by proposing a series of novel 2D-3D losses to rationalise the output of the neural network. This result demonstrates the feasibility of the MANO hand parametric model for monocular image reconstruction tasks and further strengthens the exploration of parametric models of the fundus in our research.

All of these findings demonstrate the feasibility of constructing a generalised template model in the field of medical imaging, and in our subsequent research we hope to draw on the idea of the MANO model to abstract realistic 3-dimensional objects into parametrically representative mathematical models by constructing a parametric model of the fundus to provide generic template fundus structural features to the network model. By capturing these generic features, we can better perform 3D reconstruction of monocular fundus images without relying on a large amount of training data and individualised models, thus improving the prediction of the models.

## 6.2 Parameter definition of fundus parametric model template

In order to abstract real-world objects into mathematical models composed of parameters, we first need to find the ubiquitous characteristics or commonalities of these objects. The MANO model is built based on an initialized hand model. This model uses changes in 21 skeletal points to control the deformation of the entire hand model. This means that by tracking the changes of these 21 key points in the three-dimensional space, the changes of the entire hand model in the 3D space can be effectively represented. Therefore, our goal is to find common feature points between multiple fundus 3D models and use them as reference standards for fundus 3D model changes.

By finding and capturing these common feature points, we can build a universal fundus 3D model parametric model. This model will be able to reflect the changes between different fundus 3D models while retaining their common features. The establishment of this abstract

model will provide a promising method for 3D reconstruction of monocular fundus images without relying on a large amount of individualized training data, thus improving the prediction effect and generalization ability of the model. 3D ground truth model shown in Figure 6.2.

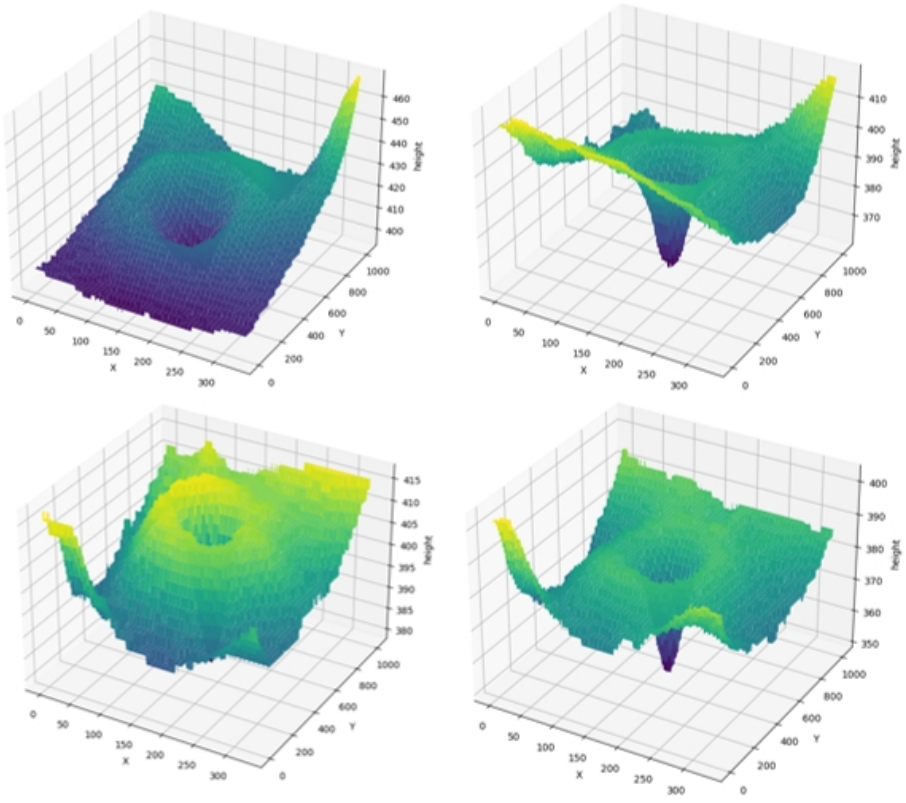


Figure 6.2: 3D ground truth comparison

While analysing different 3D ground truth fundus models, we noticed significant differences between each fundus model, which are less similar in their overall structure and have multiple points that may be used as representative features, such as textural features of the fundus surface, trends in the concavity and convexity of the fundus, and the range of variation in fundus depth values. However, our goal was to find feature points that are largely invariant across different fundus 3D models, which can be used to determine a fundus 3D model.

Further observations showed that although the structures of the fundus 3D models varied from subject to subject, they collectively showed a tendency to convex from the periphery to the centre and then concave to a single point. Although the size of the central elliptical ring region of the fundus and the plane in which it is located change with different fundus 3d models of the fundus, and the centre point of the fundus concavity also changes, based on the plane in which the ellipse is located and the coordinates of the centre of the fundus,

we can roughly determine the position of the fundus 3d model in 3D space. Therefore, we chose the elliptical circle and the centre of the fundus as representative features of the entire 3D model of the fundus and abstracted them to be represented as parameters.

This parametric approach allows us to represent the key features of the fundus 3D model in a more accurate way without considering the complexity of the whole model. We can achieve adaptability of the generic fundus structure template by adjusting the parameters so that it can be adapted to different fundus structures without having to rely on large amounts of individualised training data. Another important advantage of such a parameterised model is that it ensures that the U-Net network does not learn the features of the validation set labels while learning the fundus structure. This separated feature representation helps to improve the generalisation performance of the model, which leads to better adaptation to new fundus images and reduces the risk of overfitting.

## 6.3 Calculate the coordinates of the centre point of the fundus recess

The key to finding the depression centre point of each fundus 3d model is how to determine its coordinate position. Because not every depression centre point is exactly the position of the image centre, so we have to judge its coordinates according to the changing trend of the fundus model.

According to the analysis of the image trend theory, all the fundus images follow the changing trend of protruding from the outside to the inside and then concave to a point, which means that the trend around the concave centre point is changing very drastically, and the gradient of the pixel point expresses the changing trend around the pixel value, so by detecting the change of the gradient, we can find the position with the fastest increase or decrease of the gradient, and it is the coordinates of the concave centre point that we want to determine. As shown in Figure 6.3.

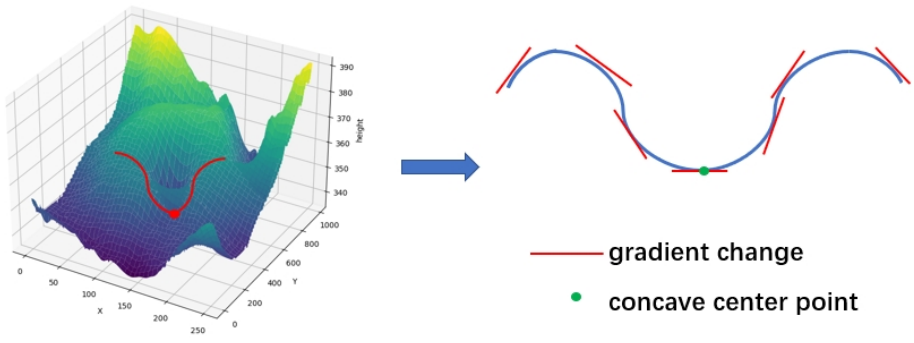


Figure 6.3: Gradient change in depression center

### 6.3.1 Gradient change row and column cross positioning

Analysis based on gradient trend maps is an effective method to locate the centroid of a depression, which makes full use of the gradient information to determine the location of the nadir, i.e. the centroid of the depression.

First of all, we can point out that the main function of the gradient map is to reveal the intensity variations in different regions of the image, thus helping us to find the region with the largest gradient variation, i.e., the region that may contain the depression centroid. In all 3d ground truth models, the depression centroid must be located at the lowest point, which means that the gradient at that point is zero. Therefore, we can determine the exact coordinates of the depression centroid by finding the location where the gradient value is zero.

To achieve this, we first traverse a two-dimensional array of 3D ground truth depth values and compute the mean value of the gradient for each row or column by row and column, respectively. This process is achieved by calculating the gradients of neighbouring elements in the array. The specific steps are as follows:

- Iterating through each row or column, we compute the gradients between neighbouring elements and take the absolute values of these gradient values and then compute the mean of these absolute values. This gives us the mean value of the gradient for each row or column.
- Next, we find the rows and columns with the largest mean values of the gradients, and the coordinates of these rows and columns will indicate where the gradient changes are most pronounced, i.e., where they may contain the centroid of the depression.

With this method, we can determine the coordinates of the depression centroid in the gradient trend graph because this point has a gradient of zero and because it is located at the intersection of the rows and columns with the largest gradient change. The advantage of this method is that it is an image information-based localisation approach that automatically adapts to different fundus images and does not depend on specific image size or location constraints.

However, we found through experiments that some fundus 3D models are not applicable, which shows that the row gradient cross positioning method may have limitations on some fundus 3D models. Different fundus 3D models may have different characteristics and changing trends, resulting in a sunken center. The positions of points are diverse. In some cases, the row with the largest change in row gradient may pass through the center point of the depression, but the column with the largest change in column gradient is in the edge area, and the gradient of the point where the two intersect is exactly 0 (As shown in Figure 6.4). In this case Our row and column cross positioning cannot accurately locate the coordinates of the center point of the depression.

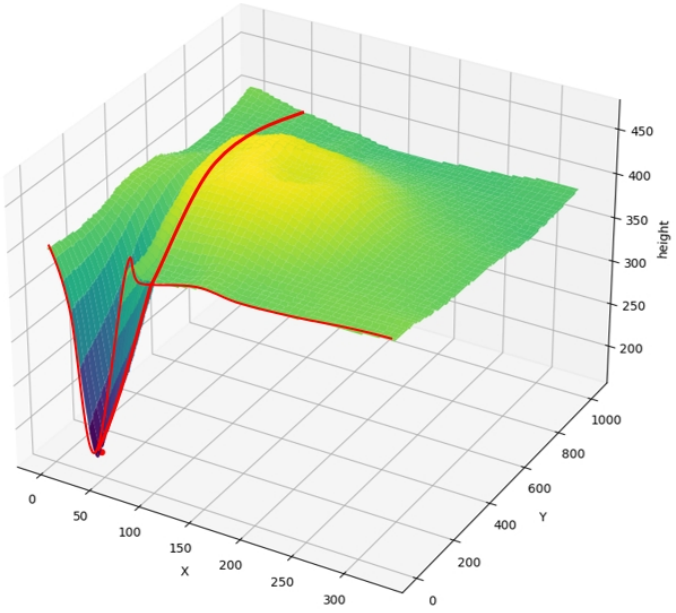


Figure 6.4: Defects in row and column cross positioning coordinates

### 6.3.2 First-order derivative improved fundus sunken center positioning method

According to the image we obtained to continue the analysis, we found that in fact the vast majority of the depression centre is near the centre of the image, and the depression centre point as the lowest point in the range, its first-order derivative must be zero, which means that the pixel value in the vicinity of the point does not change much, that is to say, this point may be a smooth region or inflection point in the image. By combining the conditions that the first order derivative is zero and the gradient changes the most, we restrict the selection range on the basis of the original, as follows:

- First, still traverse each row or column and calculator its gradient value change, but here we need to add an array to calculate and save the coordinates of all the points in the row or column where the first-order derivative is zero.
- Then, the coordinates of the centre of the image are calculated and the distance of the points contained in the array from the centre of the image is calculated, here we set the threshold to 50 and discard all points with zero first order derivatives above 50.
- Finally, we cross-locate the remaining points by row and column gradients, and we can obtain the centroid of the depression that is closest to the centre of the image and has a first-order derivative of 0 and the largest gradient change.

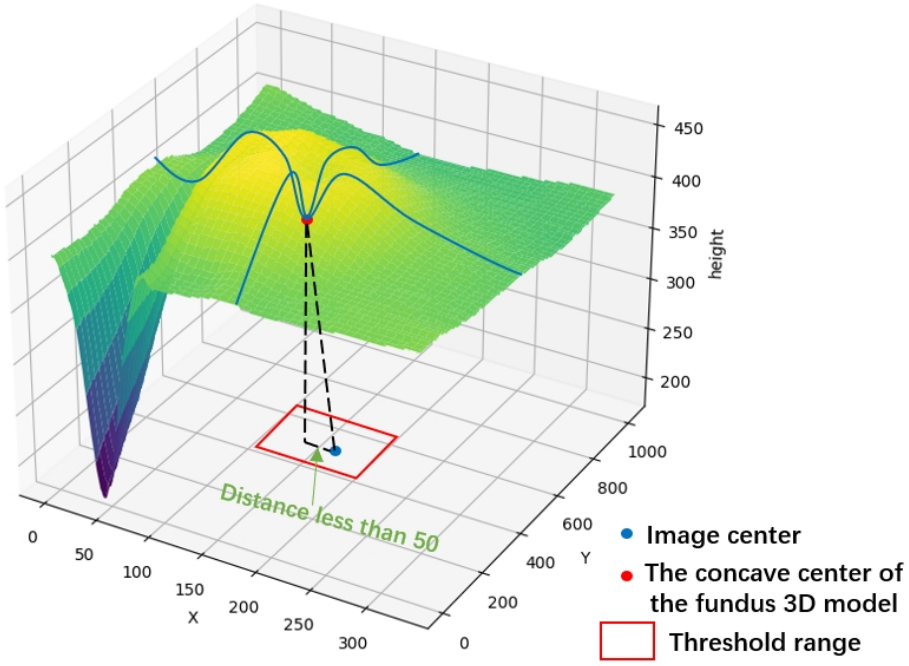


Figure 6.5: Defects in row and column cross positioning coordinates

After a series of experiments and analyses, we conclude that the cross-location method using first-order derivatives as well as a threshold range-constrained rank gradient variation performs well in fundus image processing (As shown in Figure 6.5). This method not only accurately locates the position of the centre point of the fundus depression, but also shows strong adaptability in coping with the situation of different 3D models of the fundus. Most importantly, it is able to robustly cope with underlying fundus structural lesions without being affected by them.

The successful application of this method lays a solid foundation for us to construct a universal fundus 3D model structure. By accurately locating the centre point of the depression, we can more accurately capture key features of the fundus structure without relying too much on individualised data or being limited by the diversity of fundus structures.

## 6.4 Extraction of fundus elliptical circular coordinates

After determining the locations of the fundus depressions, we tried to combine the experience of eliminating the black circular localisation lines in the fundus OCT images at the very beginning, constructing standard elliptical circles and traversing the pixel points around the centroid, hoping to use this to find the corresponding elliptical circle coordinates of the location of each fundus 3d model.

However, we found that this method is not applicable to the structural localisation

of fundus 3D models because the elliptic rings in fundus 3D models have complex 3D information, and their constituent pixel points are not only located in different planes, but also difficult to be captured due to their different sizes and shapes, e.g., some elliptic rings have very similar long and short axes, and thus are closer to a circle, which makes it difficult to adapt the standard template to the various fundus 3D models. This makes it difficult to adapt standard templates to various fundus 3d model structures.

For these reasons, there is an urgent need for a more applicable method to accurately locate elliptical rings in fundus 3D models. This approach needs to take into account the diversity of locations and shapes of elliptical rings in 3D space (As shown in Figure 6.6) in order to more accurately capture the features of the fundus structure and to be able to adapt to a variety of different situations.

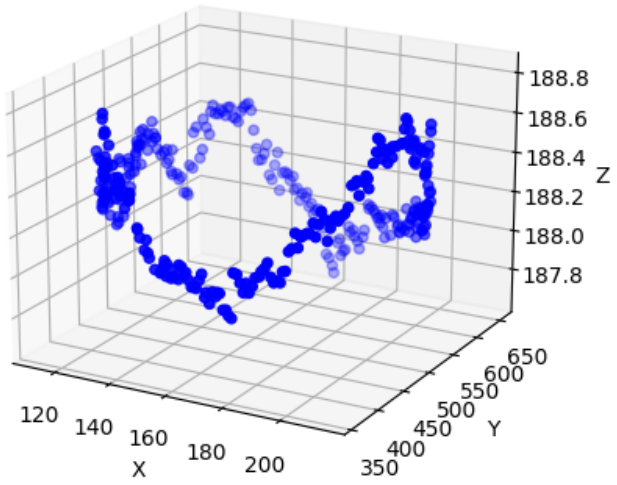


Figure 6.6: Expected renderings of fundus elliptical ring positioning

### 6.4.1 Method for finding the coordinates of elliptical ring

Before embarking on defining a method for finding the coordinates of an elliptical torus, we need to take a deeper look at the characteristics of different 3D models of the fundus. The essential task of finding the elliptical rings remains to determine the coordinates of the individual pixel points that make up the ellipse. Considering the trend of fundus 3D models, it usually shows a gradual increase and then decrease from the periphery to the centre. Therefore, the first order derivatives of the pixel points on the circle of the ellipse we are looking for should also be zero.

Although we realise that the method used to find the black ring from a 2D image cannot be directly applied to the search for 3D elliptical rings, we can learn from the method previously used to find the centre point of a depression. We can modify this method appropriately to make it applicable to the task of finding the coordinates of elliptical rings. The specific implementation is as follows:

- First, the traversal is altered to start from the previously retrieved depression centroid coordinates and scan along the 360-degree direction in 1-degree steps to find possible elliptical rings, a process that still uses gradient change trend analysis to find the point in the image with the largest change in gradient as the centroid of a possible elliptical ring.
- Then, the scanning range is set, here we set the radius of the circle to 100, to check whether these points are within a certain distance and satisfy the condition that the first-order derivative is 0, so as to avoid selecting irrelevant points.
- Finally, the scanning formally starts along 360 directions, gradually increasing the radius along the given direction vectors, and recording the point with the largest pixel value and its coordinates among the pixel points that meet all the conditions during the scanning process.

The experimental results prove that the improved method works very well and can be applied to different fundus 3D models including disease cases. In different situations, it can find the corresponding local highest points in 360 directions and record the ellipse (As shown in Figure 6.7). The pixel coordinates of the circle. At this point, we have successfully obtained the parameter information for the digital parametric 3D fundus model. Now, we will further investigate how to build a standard fundus 3D model based on this information.

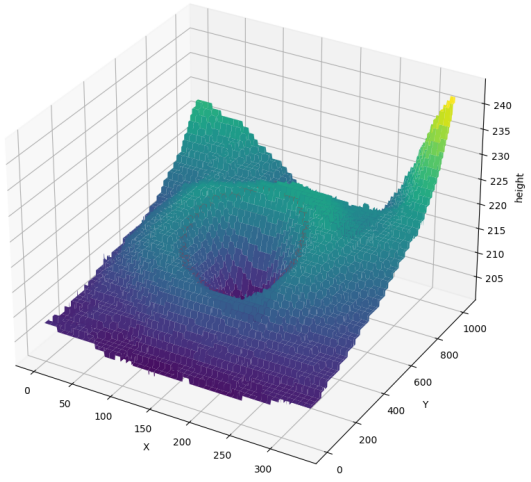


Figure 6.7: Fundus 3D model elliptical ring coordinates

## 6.5 Fit the fundus ellipse according to the circular coordinates

Now we already have the elliptical ring coordinates corresponding to each fundus 3D model, but observing the saved elliptical ring pixel coordinate points, we can find that the distribution of these pixel points is irregular, and they are not perfectly distributed according to the elliptical shape (As shown in Figure 6.8). in the 3d model. This also means that based on these pixels that are not on the same plane, we can construct many elliptical rings containing different numbers of pixels in different planes, but what we need is the ellipse that best represents the 3D gradient transformation trend of the fundus surface Ring, so we need to fit an ellipse that best fits the fundus 3D model through these ellipse pixel coordinates, which we call the fundus ellipse.

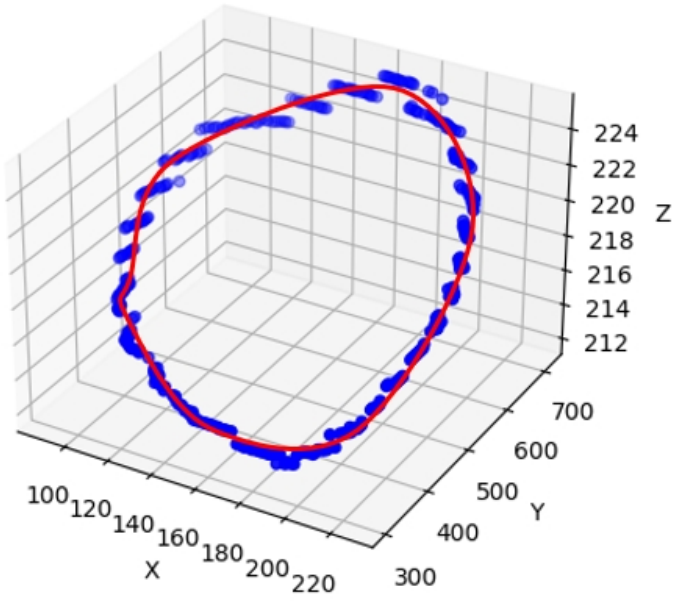


Figure 6.8: Distribution of elliptical ring coordinates in 3D space

Based on the 3D image in elliptical coordinates, it is observed that different pixel points do not all lie in the same plane. This can complicate dealing with the 3D ellipse fitting problem in subsequent studies. To address this challenge, we chose to downscale the problem from 3D to 2D, using a top-down view to deal with these 360 pixel points. Our approach is to fit an ellipse that passes through most of the pixel points by using an ellipse fitting function and obtaining the corresponding ellipse parameters. The ellipse circle thus obtained will contain the largest number of pixel points and will best represent the positional coordinates of the 3D gradient transform of the fundus surface. The key steps in this method are as follows:

- Downscaling:

Since the pixel points are located in different 3D planes, we have chosen to adopt a top-down view for these pixel points. This means that we ignore the depth information and only consider the projection of the ellipse on the horizontal plane, obtaining the ellipse that passes through the most pixels through the horizontal projection, and then upscaling it to 3D space.

- Fitting the ellipse:

  We fit these pixel points using an ellipse fitting function to find the ellipse circle that best fits these points. This fitting process will give us the parameters of the ellipse, such as the long axis, short axis, rotation angle and centre coordinates. Result shown in Figure 6.9.

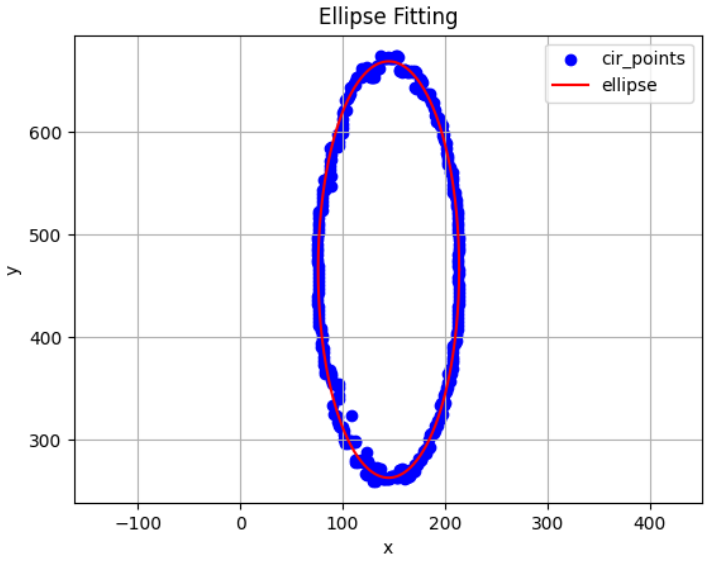


Figure 6.9: Fitting of elliptical ring coordinates in 2D space

With this approach, we are able to process 3D images in elliptical coordinates in a simpler and controlled manner, resulting in the most representative elliptical circles. This will help us to understand the fundus structure more accurately and construct a digital parametric 3D model of the fundus.

## 6.6 Construct template corresponding to the fundus 3D model

The production of generic templates is important for the study and analysis of 3D models of the fundus. A generic template is a standardised 3D model of the fundus with a set

of shared features that can represent common features across different samples. Secondly, generic templates can be used as a basis for further analysis. Similar to the MANO hand parameter model, once we have the same standardised 3D generic template of the fundus, we can input the gradient transformation trend, 3D depth information, etc. that the 3D model of the fundus has during network training to further refine the network model training.

In our study, the individual fundus 3D models have different poses, sizes, and rotation angles from one another, and attempting to analyse the shape features they share requires us to artificially apply control variables to the models. Therefore, in our study, we rotated all 3D models to a plane parallel to the x-axis to eliminate these differences, making feature differences between models easier to compare and analyse. This standardisation also allowed us to identify and understand important features of the fundus structure more easily.

However, when dealing with the rotation of 3D models of the fundus, we must take into account a variety of complex factors. Unlike simple angular rotation of 2D images, rotation of 3D models [D.-Y. Chen et al., 2003] involves a number of key concepts that need to be handled carefully to ensure accuracy and controllability.

- Axis of rotation:

  We need to determine the axis of 3d rotation. Typically, we can choose to rotate around the X, Y, or Z axes, but in the case of the fundus 3D model, we rely on the coordinates of the ellipsoid ring of the fundus 3D model that we previously acquired to determine the axis of rotation. The plane where this elliptical ring is located can be regarded as the plane that the whole fundus 3d model is facing, and the normal vector of that will be the rotation axis for our rotation.

- Sequence of rotation:

  The order of rotation involves the order in which rotations are performed around multiple axes. For example, XYZ order means rotating first around the X axis, then around the Y axis, and finally around the Z axis. A different order of rotation may lead to completely different results, in our experiment, since we want to rotate all the fundus 3d models to be parallel to the x-axis plane, the order of rotation is from the fundus 3d model rotation axis rotating it to be parallel to the z-axis and perpendicular to the x-axis, ensuring that the plane of rotation is parallel to the plane where the x-axis is located.

- Centre of rotation:

  The centre of rotation is the centre of the rotation operation, and we need to explicitly define a centre of rotation to ensure that the model is rotated around the correct point. In our study, we have tried to use the centre point of the depression, the centre of the ellipse circle and the centre of the image as the centre of rotation, respectively, and obtained different conjunctions.

By taking these key factors into account, we were able to ensure that the rotation of the fundus 3D model was accurate and controllable, which provided theoretical support for our subsequent construction of a generic template for fundus 3d models.

### 6.6.1 Rotate all fundus 3d models to the same plane

After obtaining the fitting plane corresponding to each fundus 3D model, we also obtain the parameters of the plane as well as the normal vectors, and combining this known information we can rotate these planes one by one to a common plane and make them share the same centre point:

This was achieved by firstly we constructed a unit matrix of the normal vectors of the corresponding planes of each fundus 3d model and calculated the angle of rotation between the normal vectors of each model and the normal vector of the reference unit (which is parallel to the Z-axis) and the corresponding axis of rotation. This step aimed to obtain the rotation parameters needed to make the plane of each model parallel to the plane where the X-axis is located.

Next, we used these rotation matrices to rotate all pixel points on each fundus 3D model to ensure that they were parallel to the reference plane.

After completing the rotation, we also needed to translate the rotated points according to the distance of each model from the reference centre of rotation (the image centroid in the plane where the X-axis is located) to ensure that all rotated fundus 3D models shared the same centroid. The example shown in Figure 6.10.

Through these steps, we successfully rotated and translated each fundus 3D model to a plane aligned with the X-axis, providing a consistent baseline for subsequent analyses and comparisons. This process helps eliminate rotational and translational differences between models, making them easier to analyse quantitatively and qualitatively.

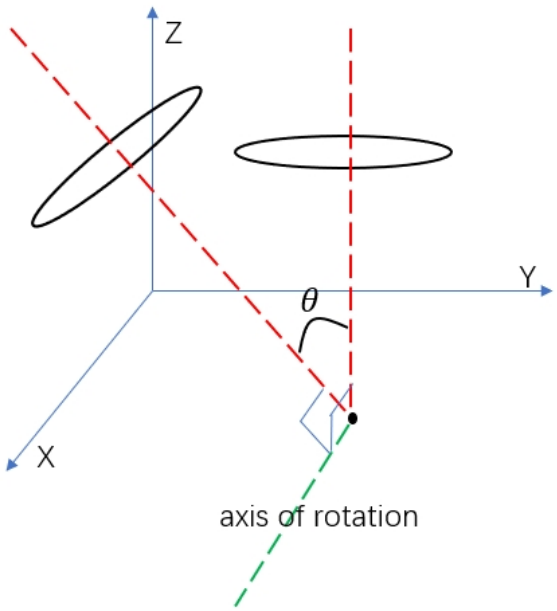


Figure 6.10: Plane rotation example (where Θ is the rotation angle)

### 6.6.2 Construct average template for fundus

After completing the rotation and centre alignment of all the fundus 3D models, we attempted to construct a universal fundus 3D model template. However, since the individual images differed considerably in all aspects except for the distinguishing features such as the fundus fitting ellipse and the centroid, it became exceptionally difficult to construct a universal template that included all features. Based on these considerations, we decided to take a mean-computation approach to create a template for the 3D model of the fundus.

We generated this universal fundus 3D model template by calculating the mean value of each pixel point for each of the 32 fundus 3D model structures by traversing them row by row. This method captures the common features of the fundus structures uniformly across all images without interference from other differences, and is simple and effective to implement.

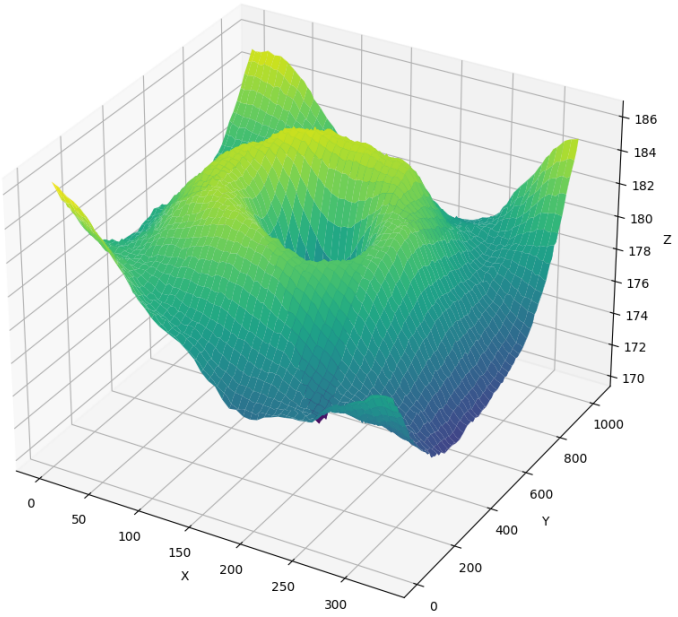


Figure 6.11: Average template of fundus 3d model

It should be noted here that due to the rotation of the fundus 3D models, the depth value of the universal template changes significantly relative to the initial fundus 3D models. To ensure the rigor and feasibility of the experiment, we need to carry out some human intervention. The range of depth values of the unrotated fundus 3D model is first calculated and then averaged. Then add the average template depth value to the average depth value range to ensure that it is between (0, 496), so as to obtain the final average template of the fundus 3D model, laying the foundation for the subsequent implementation of the template-based U-Net regression network.

In this way, we can obtain a more representative 3D model of the fundus that reflects the average features of the fundus structure and provides a strong basis for further research and

analysis. In subsequent experiments, we can make full use of the structural features of this template to provide initialisation information for the neural network to help the network better understand the fundus structure. In addition, we can achieve higher structural similarity by allowing the neural network to learn the structural features of this model so that it can mimic the structure of the target 3D model of the fundus.

## 6.7 Template-based regression u-net network results and analysis

After constructing a generic template for the fundus 3D model, we need to choose how to input this template into our U-Net network to provide structural features, and different input strategies will have different impacts on the performance of the network. In our study, we take the approach of inputting both the generic template and the fundus 3D image into the U-Net network simultaneously, allowing the network to learn two aspects of information at the same time, i.e., the detailed features of the fundus image and the structural features of the generic template. In this way, we can fuse different levels of information at the output in the network to better capture the nuances and overall structure of the fundus structure.

At the output of the network, we merge these two parts of information to improve the structural similarity of the predicted values for the actual 3D structure. The benefit of this process is that it does not require large-scale changes to the U-Net network, only adjustments to the computation of the loss function and the input channels. This greatly simplifies the complexity and maintenance cost of the code, while improving the performance and predictive power of the network.

## 6.8 Result analysis

After several rounds of experiments, the overall results of the network training are quite satisfactory. For most of the 3D models of the fundus without lesions, the predictions are very close to the true values. However, for those cases that contain specific lesion areas, the performance of the network still needs to be improved.

We reviewed and examined the performance of the network using several rubrics including mean square error, structural similarity index, Pearson's correlation coefficient, and cosine similarity, which are widely used to assess the quality and accuracy of image prediction models.

In MSE data analyses [Table 6.1], the highlighted areas usually indicate areas where the network predictions are better. By taking a closer look at the tabular data, it can be seen that the vast majority of the samples show a significant boost after the application of the generic template, suggesting that the use of the generic template was very effective for these samples. This enhancement may be attributed to the fact that the generic template is able to provide the network with valuable structural features that help the network to better predict the depth values of the 3D model of the fundus.

However, it is worth noting that there is a portion of the samples where the effectiveness decreases relative to before. After in-depth analysis, we found that these samples usually contain special lesion regions. The depth values of these regions varied considerably, and there were significant differences between them and the 3D models of the generic templates. Despite some structural similarities, the generic templates may not even be as good at predicting these samples as if no template had been used due to the differences in depth values. This implies that the use of templates may have a negative impact on the network training results when there are significant differences between the samples and the generic templates.

SSIM (Structural Similarity) is an image quality evaluation metric used to compare the similarity between two images. Through it we can compare the similarity of structural information between different images, not just the similarity of pixel values. SSIM is computed based on brightness similarity, incorporating a weighted combination of contrast similarity and structural similarity. The resulting value ranges from -1 to 1, where a value closer to 1 indicates greater structural similarity between the two images. A value of 0 implies no structural similarity, while -1 denotes complete dissimilarity between the two images.

In the data analysis [Table 6.2], we can observe that almost all samples show significant improvements after applying the universal template. This shows that the use of universal templates is very effective in improving the structural similarity between the fundus prediction model and the ground truth. This once again confirms that universal templates can provide valuable structural features for the network, further enhancing the performance and reliability of our model in fundus image analysis.

## 6.9 Conclusion and discussion

In the second phase of research, we successfully constructed a universal template of the fundus 3D model and applied it to the depth value prediction task. Actual results show that the universal 3D model template we constructed is very effective. It enables the network to fully combine structural features during the training process, better learn and predict 3D models, and improve the structural similarity between model predictions and real data.

To summarize, our research has initially achieved 3D model prediction based on monocular fundus images, and provided a template model containing shared structural features of fundus 3D models. This achievement provides a solid foundation for future fundus image processing and medical diagnosis research, and also provides strong support for further development in the medical field. In future research, we will continue to explore how to further optimize the template model to better fit fundus images including lesion samples, and expect to apply this technology to actual medical practice in the future.

Table 6.1: Compares the MSE estimates of fundus 3D models (considering the use of a universal template). MSE is used to evaluate the pixel-level difference accuracy between model prediction result and label. Its calculation is based on the regression U-Net model, and the data type used is fundus OCT images and depth labels. The first column represents the fundus 3D subject used for prediction, the second column corresponds to the scenario where the universal template is applied, and the third column represents the scenario where the template is not used. Solutions with excellent MSE evaluation results are highlighted in red.

| Subject | Template_MSE | MSE |
|---|---|---|
| 1 | 12.3614836 | 80.91336823 |
| 2 | 15.14782969 | 56.91899109 |
| 3 | 16.88923188 | 96699160 |
| 4 | 35.61440597 | 9416958 |
| 5 | 42.68319129 | 237.9021759 |
| 6 | 45.7116713 | 65.7665863 |
| 7 | 48.84662025 | 67719.26563 |
| 8 | 122.6052081 | 280197152 |
| 9 | 127.8628406 | 24773854 |
| 10 | 148.4345572 | 14.86326218 |
| 11 | 217.0648647 | 36.88617325 |
| 12 | 249.6140609 | 362520960 |
| 13 | 260.6320753 | 1528.672119 |
| 14 | 396.7403977 | 1977.305908 |
| 15 | 414.0925701 | 21.51996803 |
| 16 | 467.4578377 | 325.0846863 |
| 17 | 474.3960847 | 34879372 |
| 18 | 504.2673745 | 532.2302246 |
| 19 | 525.0994288 | 40.8406868 |
| 20 | 650.7512414 | 17909376 |
| 21 | 718.3414949 | 312.6500854 |
| 22 | 722.2468451 | 30624404 |
| 23 | 787.7619538 | 104.9335327 |
| 24 | 933.2428543 | 115.9980469 |
| 25 | 1045.694062 | 10630474 |
| 26 | 1114.920979 | 25548984 |
| 27 | 1439.015502 | 3348531.75 |
| 28 | 1806.800601 | 346.7709351 |
| 29 | 2078.00298 | 1292.505493 |
| 30 | 2116.339417 | 368.3284912 |
| 31 | 2175.876901 | 6817828 |
| 32 | 2916.398308 | 4527.938477 |

Table 6.2: Compares the SSIM estimates of fundus 3D models (considering the use of a universal template). SSIM is used to evaluate the structural similarity between model prediction result and label. Its calculation is based on the regression U-Net model, and the data type used is fundus OCT images and depth labels. The first column represents the fundus 3D subject used for prediction, the second column corresponds to the scenario where the universal template is applied, and the third column represents the scenario where the template is not used. Solutions with excellent SSIM evaluation results are highlighted in blue.

| Subject | Template_SSIM | SSIM |
|---|---|---|
| 1 | 0.309957861 | 0.160092813 |
| 2 | 0.368556838 | 0.211438074 |
| 3 | 0.314710477 | 0.109986005 |
| 4 | 0.421658318 | 0.223939794 |
| 5 | 0.354966576 | 0.192360925 |
| 6 | 0.231940488 | 0.162960583 |
| 7 | 0.274446169 | 0.139932696 |
| 8 | 0.246549575 | 0.152055737 |
| 9 | 0.245414813 | 0.085069172 |
| 10 | 0.340885333 | 0.152512824 |
| 11 | 0.308847197 | 0.16505761 |
| 12 | 0.175178239 | 0.214976578 |
| 13 | 0.243875195 | 0.162884957 |
| 14 | 0.314483949 | 0.167974559 |
| 15 | 0.303206265 | 0.157944098 |
| 16 | 0.295468195 | 0.125537772 |
| 17 | 0.1434751 | 0.103765445 |
| 18 | 0.298501999 | 0.157907577 |
| 19 | 0.321736441 | 0.159980542 |
| 20 | 0.124661653 | 0.071921662 |
| 21 | 0.294543924 | 0.134211002 |
| 22 | 0.224788544 | 0.059701347 |
| 23 | 0.314312627 | 0.144573086 |
| 24 | 0.337558703 | 0.18767955 |
| 25 | 0.115480629 | 0.151643251 |
| 26 | 0.293167554 | 0.063000767 |
| 27 | 0.203230481 | 0.116723158 |
| 28 | 0.161465101 | 0.135863357 |
| 29 | 0.108840358 | 0.132600015 |
| 30 | 0.20757415 | 0.118227432 |
| 31 | 0.26041153 | 0.104746451 |
| 32 | 0.323379329 | 0.152754062 |

# Chapter 7

# Discussion

The core goal of our research was to achieve 3D model prediction based on monocular fundus images, which is a pioneering advancement in the field of fundus medical imaging, where we have successfully achieved 3D reconstruction of a single view of the fundus in the absence of depth information. Our work has made significant progress in the following aspects:

Firstly, we have successfully achieved single-view 3D reconstruction of fundus images, which is the first of its kind in the field. Fundus images typically provide only 2D information and lack depth information, whereas our method allows 3D models to be reduced from these single images, providing ophthalmologists with more comprehensive visual information.

Secondly, we overcome the problem of shot noise such as localisation lines and scanning areas in fundus OCT images by proposing an effective algorithm to remove these interfering factors. At the same time, we used a Gaussian interpolation method to fill in the blank areas left after scanning to maximise the preservation of feature details in the images. This step is crucial for the quality and usability of fundus OCT images and provides a fundus OCT dataset that can be used for training in our subsequent studies.

Thirdly, we constructed 3D truth labels for the fundus OCT images. By establishing world coordinates and integrating the fundus OCT top view and its corresponding slice images, we mapped the depth value information contained in the slice OCT images to their corresponding locations in 3D space, and obtained the corresponding truth labels for the fundus 3d model, which provided a reliable benchmark for subsequent studies.

In addition, we designed an effective U-Net model that can be used for the regression task and proposed the corresponding composite loss function, which combines the MSE loss function and the SML1 loss function, and obtained the appropriate weight ratio through multiple tests, so that it retains the ability of MSE to differentiate the difference between the predicted and the true values while improving its generalisation ability corresponding to the complex environment.

Finally, we obtained the trained U-Net model after several rounds of training, and after testing, we proved that it is indeed capable of extracting detailed features from monocular fundus images, and the effect is more significant for features with obvious changes in the central gradient, and weaker for edge features.

However, we also recognise some challenges and limitations. it is difficult for the U-Net

model to extract structural features from 2D images for the fundus 3d model. To solve this problem, we constructed an average template of the fundus 3D model by learning the MANO hand parametric model, and input it into the U-Net model along with the fundus OCT images to participate in the training, as well as providing 2D feature details as well as initialised structural features for the network model. Ultimately, we succeeded in improving the learning effect of the model for edge features and SSIM similarity, which signifies that the predicted 3d model is more similar to the 3d truth in terms of structural features, in line with our prediction of the template effect.

In summary, our study provides a new approach for 3D reconstruction of fundus medical images, overcoming a series of challenges and laying a solid foundation for future ophthalmic research and clinical practice.

# Chapter 8

# Conclusions

In this study, we aimed to achieve 3D model prediction based on monocular fundus images, which is a pioneering work in the field of fundus medical imaging. We successfully overcame the challenge of single-view 3D reconstruction and provided a new depth-informed solution for fundus images.

Firstly, we initially achieved single-view 3D reconstruction of fundus images without providing depth information and relying only on the U-Net model for feature extraction, which reduces the cost of constructing fundus 3d models and provides more comprehensive 3d visual information for doctors. In addition to this we also provide a fundus OCT dataset containing the corresponding 3d truth values, we not only remove the shot noise in the fundus OCT images, but also fill the blank area after scanning by Gaussian interpolation method to maximally preserve the feature details of the images, which will provide an effective dataset support for future research and clinical applications. The algorithms we provide to clean up the black localisation lines and green scanning area noise left behind by fundus OCT image capture also ensure that we can subsequently obtain more fundus OCT images that can be used for training.

Second, we proposed a composite loss function suitable for extracting 3d feature information from 2d fundus images, and tested the best weights of MSE and SML1 in its constituent functions, which further improved the model performance and generalisation ability.

Finally, we abstracted the fundus 3d model as a mathematical parametric model, and used the elliptical rings with obvious gradient changes and the concave centre point as the key points of the template to construct a fundus 3D model average template, which can provide structural features of the fundus for the training of the network model, and make up for the poor effect of the convolutional network in the extraction of 3d structural features.

We have made significant progress in this study by successfully achieving 3D reconstruction of fundus medical images and overcoming several technical challenges, which provides a basis for further innovation and improvement in ophthalmic clinical care. However, we also recognise that there is still much potential room for improvement. The highest ssim value we obtained was 0.3, which is far from 1. This indicates that our structural similarity is still not high enough, so we need to continue to strengthen the model's ability to extract 3D

structural features in subsequent studies. Meanwhile, the size of our dataset is too small, which inevitably produces overfitting prematurely in multiple rounds of training, but we cannot lose too many detailed features using dropout, so we need to further expand the size of the fundus OCT dataset to better train the network model in future studies.

Another possible direction of improvement is the introduction of the Transformer model.Transformer, as a powerful deep learning architecture with a self-attention mechanism, allows the model to establish weight relationships between positions in the input sequence, which means that the model can take into account all elements in the fundus OCT image at the same time, not just the local context, which promises to make the model to focus not just on the central region where gradient changes are evident, but to have an enhanced ability for edge feature extraction. This is expected to provide better feature capture and sequence modelling capabilities in our task, helping to further improve the performance of the model and improve the results of 3D reconstruction of fundus medical images.

We believe that future efforts will further improve our results and contribute to further improvements in ophthalmic medicine and patient well-being.

# References


Aggarwal, Preeti et al. (2011). "Role of segmentation in medical imaging: A comparative study". In: *International Journal of Computer Applications* 29.1, pp. 54–61.

Aghasi, Alireza et al. (2017). "Net-trim: Convex pruning of deep neural networks with performance guarantee". In: *Advances in neural information processing systems* 30.

Ahuja, Sanjay Kumar and Manoj Kumar Shukla (2018). "A survey of computer vision based corrosion detection approaches". In: *Information and Communication Technology for Intelligent Systems (ICTIS 2017)-Volume 2 2*, pp. 55–63.

Allen, David M (1971). "Mean square error of prediction as a criterion for selecting variables". In: *Technometrics* 13.3, pp. 469–475.

Amin, S Hassan and Duncan Gillies (2007). "Analysis of 3d face reconstruction". In: *14th International Conference on Image Analysis and Processing (ICIAP 2007)*. IEEE, pp. 413–418.

Borse, Megha, S Patil, and B Patil (2013). "Literature survey for 3D reconstruction of brain MRI images". In: *Int. J. Res. Eng. Technol* 2.11, pp. 743–748.

Chen, Ding-Yun et al. (2003). "On visual similarity based 3D model retrieval". In: *Computer graphics forum*. Vol. 22. 3. Wiley Online Library, pp. 223–232.

Chen, Yujin et al. (2021). "Model-based 3d hand reconstruction via self-supervised learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10451–10460.

Deng, Jia et al. (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Facil, Jose M et al. (2019). "CAM-Convs: Camera-aware multi-scale convolutions for single-view depth". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11826–11835.

Gnonnou, Christo and Nadia Smaoui (2014). "Segmentation and 3D reconstruction of MRI images for breast cancer detection". In: *International image processing, applications and systems conference*. IEEE, pp. 1–6.

Goodman, Nathaniel R (1963). "Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction)". In: *The Annals of mathematical statistics* 34.1, pp. 152–177.

Guan, Steven et al. (2021). "Dense dilated UNet: deep learning for 3D photoacoustic tomography image reconstruction". In: *arXiv preprint arXiv:2104.03130*.

Guedri, Hichem, Jihen Malek, and Hafedh Belmabrouk (2015). "Three-dimensional reconstruction of blood vessels of the human retina by fractal interpolation". In: *Journal of Nanotechnology in Engineering and Medicine* 6.3, p. 031003.

Gunduzalp, Doga et al. (2021). "3D U-NetR: Low Dose Computed Tomography Reconstruction via Deep Learning and 3 Dimensional Convolutions". In: *arXiv preprint arXiv:2105.14130*.

Ham, Hanry, Julian Wesley, and Hendra Hendra (Aug. 2019). "Computer Vision Based 3D Reconstruction : A Review". In: *International Journal of Electrical and Computer Engineering (IJECE)* 9, p. 2394. DOI: 10.11591/ijece.v9i4.pp2394-2402.

Handa, Ankur et al. (2016). "Scenenet: An annotated model generator for indoor scene understanding". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5737–5743.

He, Kelei et al. (2021). "Synergistic learning of lung lobe segmentation and hierarchical multi-instance classification for automated severity assessment of COVID-19 in CT images". In: *Pattern recognition* 113, p. 107828.

Heo, Hoon and Ok-Sam Chae (2004). "Segmentation of tooth in CT images for the 3D reconstruction of teeth". In: *Image Processing: Algorithms and Systems III*. Vol. 5298. SPIE, pp. 455–466.

Hosny, Ahmed et al. (2018). "Artificial intelligence in radiology". In: *Nature Reviews Cancer* 18.8, pp. 500–510.

Hu, Tao et al. (2021). "Self-supervised 3D mesh reconstruction from single images". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6002–6011.

Huang, Qixing, Hai Wang, and Vladlen Koltun (2015). "Single-view reconstruction via joint analysis of image and shape collections." In: *ACM Trans. Graph.* 34.4, pp. 87–1.

Jebara, Tony, Ali Azarbayejani, and Alex Pentland (1999). "3D structure from 2D motion". In: *IEEE Signal processing magazine* 16.3, pp. 66–84.

Kar, Abhishek et al. (2015). "Category-specific object reconstruction from a single image". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1966–1974.

Lavest, J-M, Gerard Rives, and Michel Dhome (1993). "Three-dimensional reconstruction by zooming". In: *IEEE Transactions on Robotics and Automation* 9.2, pp. 196–207.

Lee, Youn Joo et al. (2012). "Single view-based 3D face reconstruction robust to self-occlusion". In: *EURASIP Journal on Advances in Signal Processing* 2012, pp. 1–20.

Lequellec, J-M and Frédéric Lerasle (2000). "Car cockpit 3D reconstruction by a structured light sensor". In: *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511)*. IEEE, pp. 87–92.

Li, Mikhail et al. (2019). "Deep neural network based shape reconstruction for application in robotics". In: *2019 International Conference on Robotics and Automation in Industry (ICRAI)*. IEEE, pp. 1–6.

Liu, Shangqing et al. (2022). "Graph-enhanced U-Net for semi-supervised segmentation of pancreas from abdomen CT scan". In: *Physics in Medicine & Biology* 67.15, p. 155017.

Loper, Matthew et al. (Oct. 2015). "SMPL: A Skinned Multi-Person Linear Model". In: *ACM Trans. Graph.* 34.6. ISSN: 0730-0301. DOI: 10.1145/2816795.2818013. URL: https://doi.org/10.1145/2816795.2818013.

MacCormick, Ian JC et al. (2019). "Accurate, fast, data efficient and interpretable glaucoma diagnosis with automated spatial analysis of the whole cup to disc profile". In: *PloS one* 14.1, e0209409.

Mariakakis, Alex et al. (2017). "PupilScreen: using smartphones to assess traumatic brain injury". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3, pp. 1–27.

McCann, Michael T, Michael Unser, et al. (2019). "Biomedical image reconstruction: From the foundations to deep neural networks". In: *Foundations and Trends® in Signal Processing* 13.3, pp. 283–359.

Oktay, Ozan et al. (2018). "Attention u-net: Learning where to look for the pancreas". In: *arXiv preprint arXiv:1804.03999*.

Pichat, Jonas et al. (2018). "A survey of methods for 3D histology reconstruction". In: *Medical image analysis* 46, pp. 73–105.

Qian, Rui et al. (2020). "End-to-end pseudo-lidar for image-based 3d object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5881–5890.

Yu-Qian, Zhao et al. (2006). "Medical images edge detection based on mathematical morphology". In: *2005 IEEE engineering in medicine and biology 27th annual conference*. IEEE, pp. 6492–6495.

Ram, Ranjith et al. (2022). "3D Reconstruction from Images: A Review". In.

Romero, Javier, Dimitrios Tzionas, and Michael J. Black (Nov. 2017). "Embodied Hands: Modeling and Capturing Hands and Bodies Together". In: *ACM Trans. Graph.* 36.6. ISSN: 0730-0301. DOI: 10.1145/3130800.3130883. URL: https://doi.org/10.1145/3130800.3130883.

Schwab, Katie et al. (2017). "Evolution of stereoscopic imaging in surgery and recent advances". In: *World journal of gastrointestinal endoscopy* 9.8, p. 368.

Seok, Jungirl et al. (2021). "A personalized 3D-printed model for obtaining informed consent process for thyroid surgery: A randomized clinical study using a deep learning approach with mesh-type 3D modeling". In: *Journal of Personalized Medicine* 11.6, p. 574.

Shin, Daeyun, Charless C Fowlkes, and Derek Hoiem (2018). "Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3061–3069.

Singh, Satya P et al. (2020). "3D deep learning on medical images: a review". In: *Sensors* 20.18, p. 5097.

Song, Tzu-Hsi et al. (2017). "Dual-channel active contour model for megakaryocytic cell segmentation in bone marrow trephine histology images". In: *IEEE transactions on biomedical engineering* 64.12, pp. 2913–2923.

Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Sumijan, Sumijan et al. (2017). "Hybrids Otsu method, feature region and mathematical morphology for calculating volume hemorrhage brain on CT-scan image and 3D reconstruction". In: *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 15.1, pp. 283–291.

Sun, Yongbin et al. (2018). "Im2avatar: Colorful 3d reconstruction from a single image". In: *arXiv preprint arXiv:1804.06375*.

Tian, Yating et al. (2023). "Recovering 3D Human Mesh From Monocular Images: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20. DOI: 10.1109/TPAMI.2023.3298850.

Toppe, Eno, Claudia Nieuwenhuis, and Daniel Cremers (2013). "Relative volume constraints for single view 3D reconstruction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 177–184.

Wang, Jia-Ji et al. (2021). "Medical 3D reconstruction based on deep learning for healthcare". In: *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing Companion*, pp. 1–5.

Wang, Yifan, Zichun Zhong, and Jing Hua (2019). "DeepOrganNet: on-the-fly reconstruction and visualization of 3D/4D lung models from single-view projections by deep deformation network". In: *IEEE transactions on visualization and computer graphics* 26.1, pp. 960–970.

Widya, Aji Resindra et al. (2019). "3D reconstruction of whole stomach from endoscope video using structure-from-motion". In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, pp. 3900–3904.

Yao, Jianhua et al. (2003). "Assessing accuracy factors in deformable 2D/3D medical image registration using a statistical pelvis model". In: *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, pp. 1329–1334.

Zhao, Wei and Lina Wang (2019). "Research on 3D reconstruction algorithm of medical CT image based on parallel contour". In: *IEEE Sensors Journal* 20.20, pp. 11828–11835.