



Convolutional Neural Network Based Portrait Segmentation

by

Chenghui He

Supervisor: **Gordon Dickers**

Project submitted as part of the requirements for
the award of MSc Master of Science

September 2022

DECLARATION

I, Chenghui He, declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed: Chenghui He

Date: 02/09/2022

FORM OF CONSENT

I, Chenghui He, hereby consent that my Project, submitted in candidature for the MSc Master of science degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed: Chenghui He

Date: 02/09/2022

Abstract - Computer vision technology has developed rapidly in the last decade, resulting in significant advances in related technologies such as human image segmentation. Because of the epidemic's influence in recent years, internet communication is becoming increasingly necessary, which makes portrait segmentation techniques even more important.

Portrait segmentation, where the portrait is preserved as the foreground and the rest of the image is blurred and replaced as the background, is widely used in video conferencing, online lectures and other applications. By blurring the background of the video or replacing it with a favourite background, privacy is protected and the quality of the video conference is improved.

With the effect of the pandemic and the quick development of computer vision, it has become important to study portrait segmentation techniques. This project focuses on the following aspects, aiming to achieve real-time portrait segmentation while improving model segmentation accuracy.

Firstly, by analysing different image segmentation models, a suitable image segmentation model is selected. The network structure of this project is determined by using different backbone networks to analyse the models' accuracy and real-time performance.

Secondly, By accumulating the majority of data, creating a home-made dataset and enhancing the data on the original image (e.g. scaling, panning, colour perturbation, etc.) to increase the richness of the data and improve the generalisation capability of the model.

Thirdly, the impact of different image sizes on the model is analysed. By varying the image size of the input model (e.g. 112*112, 224*224, 512*512), the accuracy of the model and its real-time performance are analysed to select the appropriate image size.

Fourth, the effect of various loss functions on model correctness(e.g. BCE loss, Dice loss, Focal loss) is analysed to determine the loss function.

Finally, the TTA strategy is used for model inference and error case analysis is performed on samples with low segmentation metrics, and finally the model is applied on video streams for testing.

Keywords –*Computer vision, Portrait segmentation, Data enhancement, error case analysis*

Table of Contents

CHAPTER 1 - INTRODUCTION.....	7
1.1 Introduction	7
1.2 Aims.....	10
1.3 Objectives	10
1.4 Structure of The Project	11
CHAPTER 2 - LITERATURE REVIEW AND THEORIES	12
2.1 Literature Review.....	12
2.1.1 Semantic segmentation based on traditional methods	12
2.1.2 Semantic segmentation based on convolutional neural networks	13
2.2 Basic Theory of Neural Networks	16
2.2.1 Convolutional Neural Network	16
2.2.2 Residual Networks.....	22
2.2.3 Principles And Analysis Of Network Optimisation Algorithms	25
2.2.4 Attentional Mechanisms	26
2.2.5 Evaluation Indicators	29
2.3 Summary of the chapter.....	30
CHAPTER 3 - A BISECTION-BASED SEMANTIC SEGMENTATION ALGORITHM FOR PORTRAITS	31
3.1 Overview.....	31
3.2 Two-branch Convolutional Neural Networks.....	31
3.2.1 Spatial Branching Network	31
3.2.2 Contextual Branching Network.....	32
3.2.3 Feature Fusion Module	33
3.3 Results Analysis	33
CHAPTER 4 - DATA ENHANCEMENTS.....	35
4.1 Introduction To The Dataset	35
4.1.1 The Portrait Dataset 2000.....	35

4.1.2 The Portrait Dataset 34427	36
4.1.3 MS COCO Dataset.....	37
4.2 Data Enhancement Method.....	37
4.2.1 Data Set Stitching.....	37
4.2.2 Home-made Datasets	38
4.2.3 Online Data Enhancements.....	40
4.2.4 Transfer Learning.....	41
4.3 Results Of Data Enhancement Experiments	42
CHAPTER 5 - ANALYSIS AND DETERMINATION OF LOSS FUNCTIONS.....	43
5.1 Softmax Function	43
5.2 Binary Cross Entropy Loss.....	43
5.3 Focal Loss	44
5.4 Dice Loss.....	44
5.5 Results of Different Loss Functions	45
CHAPTER 6 - MODEL TRAINING, REASONING AND ERROR CASE ANALYSIS	46
6.1 Pytorch Network Framework.....	46
6.2 Initial Optimal Learning Rate Finding Strategy.....	47
6.3 Model Training	48
6.3.1 Project Documentation	48
6.3.2 Project Environment	48
6.3.3 Model Training Steps and Parameter Selection	49
6.4 Model Inference and Case Study of Errors.....	50
6.4.1 Model Inference	50
6.4.2 Case Study of Errors.....	52
CHAPTER 7 – CONCLUSION AND OUTLOOK.....	53
7.1 Conclusion.....	53
7.2Future Work.....	53
References.....	54

Table of Figures

Figure 1 Portrait Segmentation Application	9
Figure 2 Convolution calculation diagram	17
Figure 3 Maximum pooling and average pooling process.....	18
Figure 4 Sigmoid Activation function	19
Figure 5 Tanh Activation function.....	19
Figure 6 ReLU activation function	20
Figure 7 Bilinear interpolation.....	21
Figure 8 Fully connected layer	22
Figure 9 Residual Unit	23
Figure 10 SE Module	27
Figure 11 Spatial Attention Map.....	28
Figure 12 CBMA network diagram	29
Figure 13 spatial branching network.....	32
Figure 14 Contextual Branching Network	32
Figure 15 The Overall Network Structure	33
Figure 16 MIOU values and inference times.....	34
Figure 17 Data sample in The Portrait Dataset 2000	35
Figure 18 Data sample in The Portrait Dataset 34427	36
Figure 19 Data sample in MS COCO	37
Figure 20 Model data loading process	38
Figure 21 Image fusion process	39
Figure 22 Abnormal data in the home-made data set	40
Figure 23 Selected data enhancement samples.....	41
Figure 24 Determine the initial learning rate	47
Figure 25 The overall code file	48
Figure 26 Final training results	50
Figure 27 Test Time Augmentation.....	51
Figure 28 Inference result	51
Figure 29 Real-time portrait segmentation	52
Figure 30 Sample error cases.....	52
 Table 1 Residual network structures.....	 24
Table 2 Training results at different resolutions and backbone	34
Table 3 Results of data enhancement experiments	42
Table 4 Training results with different loss functions	45
Table 5 Mainstream learning framework statistics.....	46
Table 6 Project Environment	49
Table 7 The results of the TTA strategy	51

CHAPTER 1 - INTRODUCTION

1.1 Introduction

With the growth of smart hardware and the internet, smartphones, tablets and wearable devices are everywhere. People tend to use various mobile devices to document their lives. Images are widely accepted because of their intuitive and easy-to-accept characteristics. Scientific research shows that people obtain information from the outside world through their five senses, 87% of which is obtained through vision, while only 13% is obtained through touch, smell and hearing. This shows the importance of visual information in human society. However, as the demand for visual information grows, a single natural picture is no longer sufficient to meet people's needs. In order to meet this demand for higher level visual information, the technology of digital image matting and compositing has been developed.

Digital image matting and compositing is an image processing technique that separates the foreground object from the background in a pair of images or a video frame, and then combines the separated foreground object with the rest of the background. In the process of development, keying technology has gradually moved from optical keying to digital keying. The main principle of optical keying is to extract the foreground object from the whole image by applying a filtering technique to the image with a specified background colour, the background colour is usually chosen as blue, hence optical keying is also known as blue screen keying. The prerequisite for blue screen keying is a single colour background, which requires the target object to be photographed against a specific colour background, greatly limiting its application. Therefore, how to accurately and effectively extract the image information required by the user in a complex natural image background, shorten the human-computer interaction time and reduce the difficulty of human-computer interaction has become a hot topic of keying technology research in recent years. With the continued growth and maturity of digital keying methods gradually replacing optical keying methods.

Image semantic segmentation is a basic problem in computer vision, yet one of the obstacles is overcoming this fundamental task. It is the foundation for perceiving and comprehending the content of an image, and it is consistent with the goal of image segmentation, which is to accurately predict the category of each pixel of the input image in such a way that pixels that belong to the same category have the same category label. In contrast to this, instance segmentation is a further differentiation of the segmented objects based on semantic segmentation. In this type of segmentation, the result of the segmentation contains not only the category information of each pixel, but also the differentiation of different instances of the same category. This type of segmentation is known as "instance segmentation." This is a simple task for human visual recognition in real image scenes; however, it is a very difficult task for machines, as the computer needs to predict the correct category for the detected pixel points in the target region in order to build the image's structural information and then further parse the target objects that are contained in the scene. In real image scenes, this task is simple

for human visual recognition. For the purpose of image semantic segmentation, it is thus essential to build an appropriate image segmentation algorithm in order to make the proper class prediction for each pixel point of the input picture. This may be accomplished by designing a good image segmentation method.

The fast advancement of technology in the field of multimedia has resulted in an increase in the stringency of requirements placed on the process of removing necessary pictures from complicated backgrounds. For example, the extraction of the target object of interest needs to be better and more accurate, the speed of extraction needs to be faster, and even the need for real-time keying for extraction, these requirements have rapidly driven the development of automatic digital image keying technology to achieve no special requirements for the background of the image, and can be implemented in any image background keying processing, which greatly increases the scope of application of keying technology. Digital keying technology can achieve a very realistic image background of natural replacement, the human eye is difficult to distinguish whether post-processing, so that the post-image synthesis to achieve the effect of falseness. The usage of digital keying and compositing technology is growing more and more popular in a variety of sectors as a result of the progression of both time and the developments that have occurred in multimedia technology. In the film industry, digital keying and compositing technology has been used to produce a series of Hollywood blockbusters such as "Avatar" and "Top Gun". Digital keying and compositing techniques have not only brought first-class visual feasts to film-goers and huge profits to the film industry, but they have also been widely used with great success in the video, advertising and magazine industries. In personal portrait and wedding photography scenarios, for example, people want to apply image keying and compositing techniques to naturally blend different portrait poses into different landscapes or surreal scenes. While digital keying techniques are suitable for most natural scenes in specific industry applications, they are often less effective for special or delicate scenes requiring keying. Examples include animal hair and situations where the foreground and background pixels are similar in grey scale. This makes accurate keying often difficult to achieve.

Deep learning has gone through many ups and downs since the concept of artificial neural networks was introduced in 1943 to 2016 when DeepMind designed AlphaGo to defeat the famous Go champion Lee Sedol. With the advent of big data and the exponential growth of computer computing power, computer vision has grown significantly. Computer vision is mainly involved in three areas, including target detection, image recognition and image segmentation. Of these three areas, image segmentation can play an important role in assisting and improving other vision tasks because it understands image content at the pixel level. At the same time digital image keying and image segmentation are very similar in the form of their respective concrete implementations. Therefore, inspired by the research on image segmentation based on deep learning methods, deep learning methods are applied to digital image matting.

Portrait refinement is a sub-task of image segmentation and has the same goal as other tasks in image segmentation, to accurately classify each pixel point. However, portrait segmentation has unique features compared to other semantic segmentation tasks, such as blurred boundaries and complex scenes, and a variety of positions of the person target

in the whole image. Masked images from portrait refinement segmentation can be used to edit the foreground and background of portraits, such as portrait image background removal, background replacement, portrait colour retention and portrait stylisation. Traditional image segmentation methods segment images based on low-level visual information, such as the colour, shape, brightness and texture of image pixel points [1]. The use of this low-level visual information, however, leads to too coarse segmentation results in overly complex scenes due to the lack of mid- and high-level semantic information, and does not achieve the desired segmentation accuracy.

With the popularity of smartphones and the rise of applications such as webcasting, short videos, online meetings and Meitu cameras. More and more users are using mobile phone cameras to capture videos and photos to record their lives. At the same time, users' requirements for processing the captured images are also increasing, and there is an influx of demand for portrait segmentation, such as background editing (background replacement, depth-of-field rendering, etc.) and foreground editing (portrait styling) using the results of portrait segmentation, as shown in Figure 1.

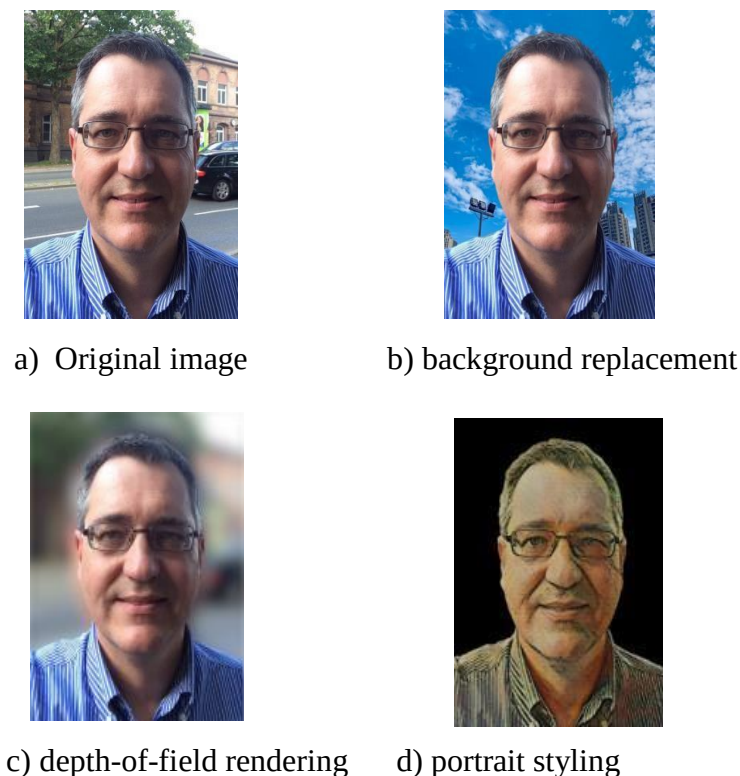


Figure 1 Portrait Segmentation Application

1.2 Aims

This project mainly uses convolutional neural network to implement portrait segmentation. By training the segmented portrait dataset, this project can achieve the purpose of portrait segmentation and ID photo production.

1.3 Objectives

This project is to achieve automatic portrait snapping and video background segmentation by using neural networks. By designing neural network models, optimizing neural network model parameters, and enhancing data from multiple angles to improve model accuracy while meeting real-time requirements, we achieve portrait segmentation. In order to complete this project, mainly from the following sub-objectives:

- **Backbone network selection and effect of image size**

Different backbone networks (e.g. Resnet18 and Resnet101) were selected and the values of Miou and inference times were analysed to determine the appropriate backbone network. Put images of different resolutions into the model for training, analyse the relationship between accuracy and inference speed, and determine the appropriate image size.

- **Data enhancement**

1. by using data enhancement methods: colour enhancement, image affine transformation, etc.
2. by increasing the amount of data: by adding the Portrait Dataset 34427 dataset.
3. by customizing the dataset: Analyse the background categories in the COCO dataset that match the actual scene, select the categories that match the realistic scene as the portrait background, and increase the richness of the dataset to achieve data enhancement.

- **Selection of loss function**

Choose different loss functions, such as BCE loss, Dice loss, Focal loss, etc. Analyse the impact of different loss functions on the model and determine the appropriate loss function.

- **Error case analysis**

The images with low Miou values are analysed and the reasons for poor segmentation accuracy are considered.

1.4 Structure of The Project

Chapter 1: This section describes the background, purpose, and significance of this project.

Chapter 2: This section provides a literature review of the mainstream image segmentation algorithms and introduces the relevant theoretical knowledge involved in this project.

Chapter 3: This section introduces the two-branch image segmentation network model and compares and analyses the effect of different backbones (resnet18, resnet101) on the inference speed and accuracy of the model, and determines that resnet101 is the backbone of this project.

Chapter 4: This section introduces the dataset involved in this project and performs 3 aspects of data enhancement on the data and analyses the effect of each data enhancement.

Chapter 5: This section compares the impact of different loss functions and the weighted use of loss functions on the accuracy of the model, and determines that BCE + Dice loss as a loss function is the best model.

Chapter 6: This section introduces the network framework, development language and model training details used in this project, and performs model training, model inference and error case analysis.

Chapter 7: This section provides a summary of the project and plans for future work.

CHAPTER 2 - LITERATURE REVIEW AND THEORIES

2.1 Literature Review

Image segmentation is a subdivision of the image segmentation field, and image segmentation techniques may be broadly classified into two categories: the more conventional deep learning-based image segmentation techniques and the more recent deep learning-based image segmentation techniques, and many researchers have made great contributions and achieved fruitful results in the traditional image segmentation field, building a complete theoretical system for traditional image segmentation research. In recent years, the significant improvement in the accuracy of image semantic segmentation based on deep learning has attracted a large number of researchers to devote themselves to this research work. This method makes full use of the middle and high-level image semantic features compared to the traditional method that only utilises the underlying image features, which enables the method to achieve high accuracy segmentation results in image segmentation tasks of complex scenes. This section next briefly describes the current state of research in portrait segmentation based on traditional and deep learning-based methods.

2.1.1 Semantic segmentation based on traditional methods

Since the 1970s, the problem of image segmentation has been of great interest to researchers and many contributions have been made to it, with many image segmentation methods being proposed for different practical problems. Traditional image segmentation methods classify images by extracting low-level semantic features such as colour, texture or shape in the image, and are broadly classified into segmentation methods based on thresholding, edge detection, region and graph theory. Threshold-based image segmentation methods select one or more suitable thresholds and classify the image into two or more classes by comparing the thresholds with the pixel grey scale values. Commonly used threshold-based image segmentation methods are: balanced histogram-based methods [2], maximum entropy-based methods [3], and adaptive thresholding methods [4], [5]. Edge detection-based image segmentation methods use the abrupt changes in the grey value of the edge pixels in different regions of the image to segment the image, and the specific implementation method is to design and use different filters to calculate the derivative of the first order or the second order to detect the abrupt changes in the edge pixels to achieve image segmentation. Commonly used edge based detection methods are graph cut based methods and Hough transformations. Region-based image segmentation is a direct region finding based segmentation method. It has two basic forms: one is the gradual merging of similar neighbouring pixels into one region by an iterative method until all pixels are fully divided, representing the method region growth method [6], [7]; the other is the gradual cutting of large regions or whole images to the desired segmentation region, representing the method region splitting method [8]. Graph theory-based image segmentation methods create a weighted undirected graph for the image. The points in

the undirected graph represent each pixel point in the image, and the weights of the edges in the undirected graph represent the similarity of points and dots, which are cut into subsets based on the weights of the edges (similarity), with points in the same subset representing points with high similarity and points in different subsets representing points with low similarity.

Traditional image segmentation has obvious drawbacks: high requirements for images, only for specific requirements of the picture, such as the image segmentation method based on threshold for the background and the target difference is not obvious image segmentation effect is poor, based on edge detection image segmentation method is more sensitive to noise, vulnerable to noise, so the traditional image segmentation method for complex scenes of image processing effect is poor.

2.1.2 Semantic segmentation based on convolutional neural networks

AlexNet, which was developed by Krizhevsky and colleagues [9] and used a convolutional neural network, triumphed in the LSVRC2012 competition with a score that was 10.9% higher than the runner- up's. Convolutional neural networks are once again a popular study subject in the field of machine learning techniques as a result of their outstanding performance on the ImageNet dataset [10]. Image classification, target identification, and semantic segmentation are just some of the activities that have been revolutionised as a consequence of the use of convolutional neural networks. In 2014, VGGNet [11] deepened the depth of convolutional neural networks by replacing the large convolutional kernels in all convolutional layers with small 3×3 kernels and using 1×1 convolutional kernels for channel linear transformations and a maximum pooling layer with a step size of 2 and a 2×2 scale of convolutional kernels. The Inception model proposed by GoogleLeNet [12] uses three different sizes of convolutional kernels to convolve in parallel, and the Inception model uses different scales of convolutional kernels to pool more feature information while increasing the network width. In 2015, the residual structure proposed in ResNet [13] enables convolutional neural networks to effectively suppress the gradient disappearance or gradient explosion problem, greatly deepening the depth of the convolutional network to 152 layers for convolutional neural networks using the residual module.

The Fully Convolutional Network (FCN), which was built on the VGG network, was suggested in 2014 by Jonathan Long and colleagues [14]. The fully connected layer of the VGG network is replaced by a convolutional layer in the FCN, which enables the FCN to analyse input pictures of any resolution. Next, deconvolution is used to recover the low-resolution version of the image that was processed by the FCN. The FCN is the first way to perform semantic segmentation of pictures by using convolutional neural networks, which is of major relevance in the research of methods for image semantic segmentation. However, FCN still has many shortcomings. On the one hand, the local feature information extracted by small convolutional kernels in complex scenes may cause bias in global understanding, for which FCN has no strategy to exploit global contextual information; on the other hand, the spatial information is seriously lost with continuous convolution and pooling. Although FCN proposes long jump connections

to recover spatial feature information in the upsampling phase, the span of FCN in upsampling is too large and the insufficient use of high-resolution features in the downsampling path makes the spatial information recovered by FCN in the upsampling phase insufficient.

To address the problems of FCN in these two aspects, the subsequent method is to improve the segmentation effect from these two aspects: one is how to better use the contextual information to extract richer semantic information; the other is how to enhance the segmentation effect by collaborating between shallow feature information and high-level semantic knowledge.

The size of the perceptual field can give a rough indication of the extent to which contextual information is used [15]. DeepLabV1[16] proposes an expanded convolutional or null convolution structure based on the above problem, which expands the field of convolutional kernels by adding zeros to the kernels, so that a larger range of contextual information can be extracted without increasing the number of parameters and computational effort. This results in a significant improvement in the computational speed and accuracy of DeepLabV1. In literature [17], multiple images with different resolutions are formed by scaling the original image, which are used as input images for convolution operation respectively, and finally the feature maps with different resolutions are fused. However, this method requires too much computation. DeepLabV2 [18] addresses the same problem by proposing a null-space DeepLabV2 replaces the Atrous Spatial Pyramid Pooling (ASPP) model with ResNet, which has a deeper convolution depth, for better feature extraction. VGG and retraining ResNet in ImageNet to adapt it to image segmentation.

Zhao et al [15] found that in complex scenes it is difficult to distinguish small objects such as street lights, and large objects may be out of the field of perception resulting in discontinuous predictions.

It is important to pay special attention to the global information of the various receptive fields in order to enhance the segmentation of both big and tiny objects, so PSPNet was proposed, which combines four different scales of feature maps in a pyramidal structure to obtain more semantic global information and improve the segmentation effect. In the same year, RefineNet [19] designed a cascaded deconvolutional network to obtain multi-scale contextual information, and a variant of RefineNet [20,21] also demonstrated the effectiveness of multi-scale. DeeplabV3++ [22] made a new improvement on ASPP proposed by DeeplabV2, and applied DeeplabV3++ [23,24] to ASPP, Xception and encoder structure. DeeplabV3++ can reduce computation while maintaining similar performance.

In order to obtain more valuable contextual information, a method called attentional mechanisms is used in computer vision. SENet [25] proposes a squeeze incentive module that computes channel weights from the dependencies between channels and enhances the expressive power of the network by giving different weights to different channels to strengthen the feature information on important channels. This attention module, which calculates channel weights, is called channel attention. Convolutional operations can extract information features across both channel and spatial dimensions.

CBAM [26] designed attention to passage and attention to space in both channel and spatial dimensions, and sequentially concatenated the two types of attention into separate attention modules for inferring the content and location information of pixel points. Since the convolutional layer can only establish relationships between pixels in the neighbourhood and cannot obtain relationships between distant pixels, the only way to obtain connections with distant pixels is through repeated convolutions or convolutional kernels with larger sensory fields, which not only increases computation and reduces computational efficiency, but also weakens the relationships established with distant

This not only increases computational efficiency, but also weakens the relationships established with distant pixels. Dependencies between pixels at a distance can enhance the understanding of the scene as much as global features.

NLNet [27] is a self-attentive model that captures long-range dependencies, which allows for more global information by computing dependencies between a specific query location and all other locations in the feature map without limiting the relationship to adjacent locations, resulting in varying degrees of improvement in computer vision tasks such as image classification, target recognition and semantic segmentation. GCNet [28] found that the global contextual information obtained in NLNet is almost the same for different query positions in an image, and a simplified network based on query-independent formulations can still achieve similar results to NLNet while reducing a large amount of computation.

DeconvNet, which was proposed by Hyeonwoo et al [29], SegNet, which was proposed by Badrinarayanan et al [30], and U-Net, which was proposed by Ronneberger et al [31] for medical image segmentation were all based on an encoder-decoder structure in 2015. This structure allows for the resolution of the feature map to be recovered. Convolution and deconvolution make up the two halves of the DeconvNet algorithm. Convolution and deconvolution are the two components that make up DeconvNet. The convolution component makes use of the VGG16 network, while the deconvolution component makes use of inverse pooling for upsampling, which is the opposite of pooling. SegNet is similar to DeconvNet in that the decoder side uses the pooling index recorded in the maximum pooling process on the corresponding encoder side to recover the feature values to their original positions. Since neither DeconvNet nor SegNet uses long jump connections to combine high resolution spatial information at the encoder side, the segmentation efficiency of both network structures is not high. The excellent performance of UNet on medical images has led to the widespread use of UNet networks in semantic segmentation. Some scholars have improved UNet in various ways, and some scholars have stacked multiple UNets into a new structure Stacked UNets by UNet stacking [32]. The literature [33] investigated and analysed the long jump connections used in FCN and the short connections in the residual module proposed by ResNet, and demonstrated that long and short jump connections facilitate the fusion of deeper networks and short connections speed up convergence. Medical segmentation requires high accuracy in image edge prediction, for which the presence of burrs on the edges is a condition that can affect the doctor's judgement, and jump connections can effectively recover fine-grained details of the target object, so the literature [34]

proposes the UNet++ network, which uses nested, dense jump connections instead of a single jump connection in UNet, enabling the relatively rich semantic features on the decoder side to fuse fine-grained information features from multiple high-resolution features on the encoder side. By using a multi-scale jump-join to fully utilise the fine-grained information at various resolutions on the encoder side and the high-level, coarse-grained semantic information on the decoder side, MCAU-Net [35] streamlines the nested, dense jump-join in UNet++ while also enhancing the segmentation effect. In order to direct the low-level semantic features, PANet [36] suggests a Global Attention Upsampling (GAU) module at the decoder side. This module extracts the channel weights of the global context of high-level semantic features.

All of the above approaches are designed to enable better segmentation of the network, but they also make the network structure more complex and require more resources for computation and storage, but these model structures are not suitable for deployment into embedded devices on the mobile side. In order to solve the problem of inference speed, memory consumption and comparable segmentation effect, lightweight network structures that can achieve real-time semantic segmentation have gradually become a new research hotspot, and the MobileNet [37,38,39] series, ShuffleNet [40,41] series and other lightweight convolutional neural networks have emerged one after another.

ENet [42] was the first to realize real-time semantic segmentation. It built a lightweight network structure to realize rapid segmentation, reduce the number of down-sampling operations, and use expansion convolution and decomposition convolution to increase sensing field and reduce memory consumption. BiSeNet [43] utilises the dual-path technique, in which the context route extracts semantic data via the usage of Xception lightweight convolutional neural network, while the spatial path extracts spatial location characteristics through the use of three convolutional layers. After the fusion of semantic characteristics and spatial information, the next step is to construct the segmentation map by upsampling the results of the previous steps. ICNet [44] proposed a cascading network structure to help restore image edge and other details. Three images of different resolutions were used as input to perform down-sampling respectively, and small-resolution feature maps were used to help restore image details from big-resolution images.

2.2 Basic Theory of Neural Networks

2.2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) [45], has a wide range of applications in areas such as image analysis, image processing and natural language processing. Convolutional neural networks usually consist of several parts such as convolutional layer, pooling layer, activation function, batch normalization layer, upsampling layer and fully connected layer. Two important features of a convolutional neural network are that the convolutional kernel extracts only local features of the input image, i.e. each point maps only a local region of the upper input layer, and that the convolutional kernel weights are shared, which means that the weights of the same convolutional kernel

remain the same when convolving the whole image. Therefore, the local connection-based and convolutional kernel weight sharing approach significantly reduces the number of parameters.

- Convolutional Layers

Each convolutional layer in a convolutional neural network consists of multiple convolutional kernels, and each parameter in the convolutional kernel is determined optimally by a back propagation algorithm [46]. Convolutional neural networks usually pre-process the input data in the input layer. Common pre-processing methods for image data include image normalisation, image downsampling and image de-averaging. Then the convolutional operation is performed on the processed data, because the local perceptual features of the convolutional operation, for shallow convolutional extraction are mostly some low-level features, for example, the local features extracted in portrait images are human eyes, eyebrows, corners of the mouth and nose, etc. Finally, more abstract information is obtained through high-level feature extraction. For example, the convolution results are shown in Figure 2 when the feature map size is 5×5, the convolution kernel size is 3×3, the step size is 1 and the feature padding is zero.

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

 $\otimes$

0	1	2
2	2	0
0	1	2

 $=$

12	12	17
10	17	19
9	6	14

Figure 2 Convolution calculation diagram

- Pooling layer

The pooling layer is often positioned below the convolutional layer in an entire neural network. This is because the pooling layer is one of the most typical components of a convolutional neural network. It is computed in a manner that is similar to that of the convolutional operation, in which a pooling window of a defined size is first established and then slid over the feature map in order to determine the value at each place. On the other hand, in contrast to the convolutional operation, the parameters included inside the convolutional kernel are optimised through a method called backpropagation. The pooling procedure is carried out by a fixed and does not make use of any training parameters. The pooling procedure is often carried out by a fixed function, which may be broken down into two categories: maximum pooling [47] and average pooling. For instance, Figure 3 illustrates what the pooling process produces when the feature map size is 4 by 4, the pooling cell window size is 2 by 2, and the step size is 2.

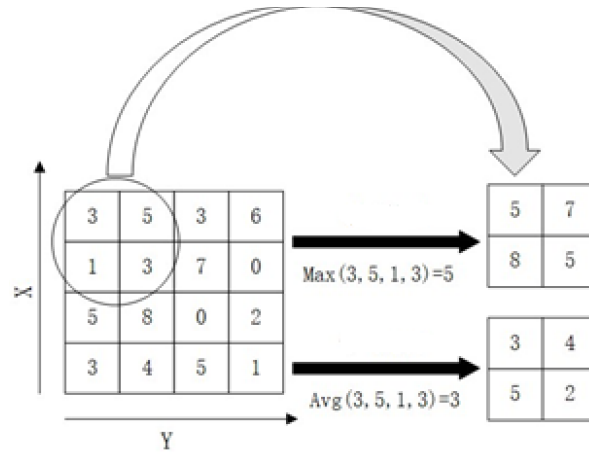


Figure 3 Both the maximum and the average pooling procedure are carried out.

Due to the similarity of adjacent pixels in an image, the main feature information of the image is retained using pooling, which reduces the feature dimensionality and further reduces the computational effort of the model. Additionally, the lower dimensional feature map has more global information when the convolutional kernel size is the same, which makes the perceptual field of the convolutional kernel larger. These are the primary effects of the pooling operation. The convolutional kernel is able to extract information that is more abstract and meaningful.

- Activation function

The convolution operation in the convolution layer can be understood as a multiplication and addition operation with fixed parameters, it implements only linear transformations, so it has limited ability to map complex input data. Activation Functions allow deep neural networks to fit more complex non-linear functions. This in turn allows the neural network to learn and simulate more complex types of data. Commonly used activation functions include the Sigmoid activation function [48], Tanh activation function and Rectified Linear Unit (ReLU) activation function.

The expression for the Sigmoid activation function is shown in (2.1): it has a value range of $[0,1]$, while Sigmoid is centrosymmetric about the coordinates $(0,0.5)$ and is often used as a threshold function, being 1 when the value is greater than 0.5 and 0 when it is less than 0.5, hence the Sigmoid function is often called a logistic function. The Sigmoid function itself features a mapping of the input values to $[0,1]$ and can therefore be used to represent probability values. Its advantage is that it is easy to derive, but its disadvantage is that when the network is back-propagated, it can lead to gradient explosion and vanishing problems as the number of layers in the network increases. Its function image is shown in Figure 4.

$$S(x) = \frac{1}{1 + e^{-x}} \quad (2.1)$$

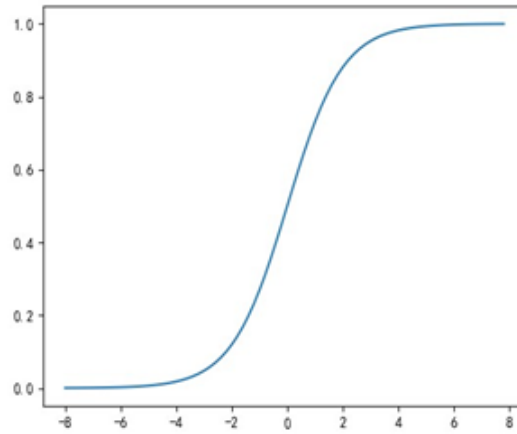


Figure 4 Sigmoid Activation function

The expression for the Tanh activation function is shown in (2.2). Its value range is $[-1, 1]$. The Tanh function converges faster than the Sigmoid function, has a larger value range for the feature mapping and is centrosymmetric about the origin, but has a similar disadvantage to Sigmoid in that it is prone to gradient disappearance and gradient explosion during training. The image of the function is shown in Figure 5.

$$\text{Tanh}(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.2)$$

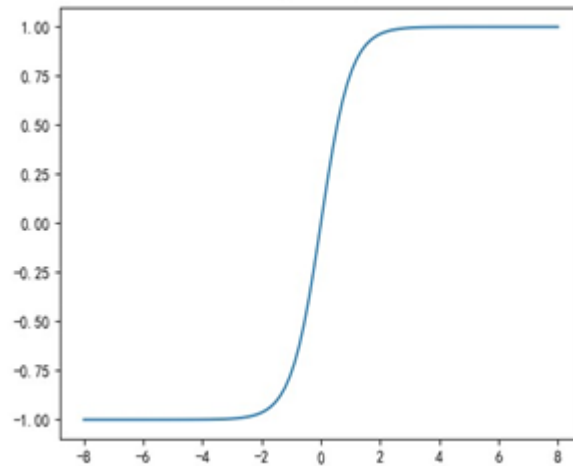


Figure 5 Tanh Activation function

The expression of the ReLU activation function is shown in (2.3). The ReLU function is also called the linear rectifier function and its function image is shown in Figure 6. When the input value is less than zero, the ReLU function's output is zero, and when the input value is larger than zero, the ReLU function's output is the same as the input value. This is because the ReLU function is computed in such a way. The ReLU function has a number of benefits, the most notable of which are its low computing cost and its ability to solve the issue of vanishing and bursting gradients. The benefits include a low computing cost, a solution to the issues of gradient disappearance and gradient explosion, and a speeding up of the network training process. The drawback is that if the input value is less than zero, the value of the function is always zero. This causes a back-propagation gradient that is equal to zero, which makes it impossible to train this portion of the neuron, which ultimately results in this portion of the neuron "death." The Leaky ReLU activation function [49] and the PReLU (Parametric Rectified Linear Unit) activation function [50] have been proposed to address the problem of neuron "necrosis" due to negative ReLU inputs. The PReLU activation function introduces a small number of additional parameters, which increases the risk of overfitting the model.

$$Relu(x) = \max(0, x) \quad (2.3)$$

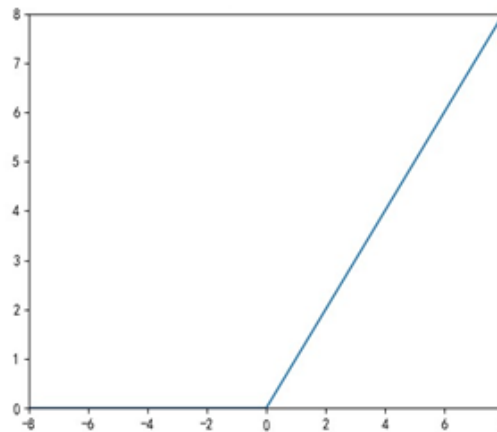


Figure 6 ReLU activation function

- Batch Normalization layer

The distribution of data in a network model is crucial to the training results. The initial input data can be solved by normalisation, while the distribution of data in each layer within the network model will change with the input data of the previous layer, and become more complex as the network deepens. The Batch Normalization (BN) [18] aims to normalize the data distribution of the intermediate network so that the data distribution of each layer is in a reasonable range, thus simplifying the initialization of network parameters and setting a larger learning rate parameter, and speeding up the training of the model. For example, if the number of input data in a batch is N , the whole input is represented as $x = [x_1, x_2, x_3, \dots, x_N]$, and the batch normalisation operation is computed as shown in (2.4).

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \quad (2.4)$$

In Equation (2.4), $\hat{x}_i$ is the output data after batch normalisation, σ_B^2 denotes the variance of the whole batch feature, and μ_B denotes the mean of the whole batch feature, an additive term to prevent the variance from being zero.

After batch normalisation, the input data is distributed within a range with a mean of 0 and a variance of 1. The result of processing in this fully restricted manner leads to a weakening of the feature extraction capability of the network, as it is impossible to estimate a reasonable distribution range for the features. This forced normalisation process ensures the distribution of the data but restricts the features that should be outside the distribution range, affecting the feature extraction capability of the network. To address this situation, the features are guaranteed to be self-adjusting by introducing two variable parameters on the feature values after the batch normalisation operation. The equation for this is shown in (2.5).

$$y_i = \gamma \hat{x}_i + \beta \quad (2.5)$$

In equation (2.5), γ is the scaling parameter and β is the translation parameter. These two parameters are determined by model training to obtain a reasonable distribution range for the features.

- Upsampling layer

The purpose of upsampling [49] is to reduce the downsampled feature map to the original image size. Common upsampling methods include UnPooling, transposed convolution and bilinear interpolation. In this paper the upsampling method uses bilinear interpolation to restore the feature map at the lower layers. The bilinear interpolation process is illustrated in Figure 7.

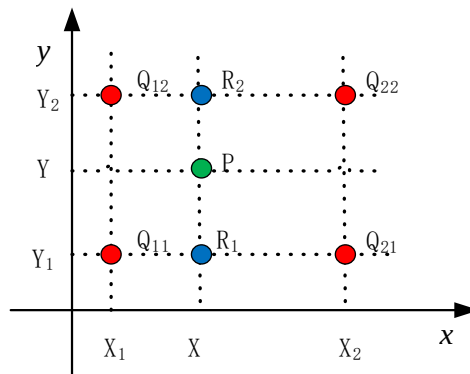


Figure 7 Bilinear interpolation

As shown in Figure 7, the four coordinates $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$, $Q_{22} = (x_2, y_2)$ are known to find the coordinates of the point P . The first linear interpolation in the direction of x yields the equation as in (2.6).

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \text{ where } R_1 = (x, y_1) \quad (2.6)$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \text{ where } R_2 = (x, y_2)$$

Then interpolation in the y direction yields the equation as in (2.7).

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \quad (2.7)$$

- Fully Connected Layers

Fully Connected Layers (FCs) are typically what are referred to as "classifiers" in the overall network model. They are located at the very end of the overall network model, after the stack of convolutional layers, pooling layers, batch normalisation layers, and non-linear activation functions, but before the Softmax function. [50,51] The convolution operation, the pooling operation, and the non-linear activation operation map the input data to the feature space of the hidden layer. The fully connected layer then takes the local features learned in the hidden layer and remaps them to the sample label space. If the entire network model is considered to be a feature extraction and then a prediction task based on the extracted features, then the fully connected layer takes the local features learned in the hidden layer and remaps them to the sample label space. The completely connected layer implements a fully connected strategy, which means that every single output from a neuron in an upper layer will be coupled to every single neuron in a lower layer. Figure 8 shows the wiring diagram in its schematic form.

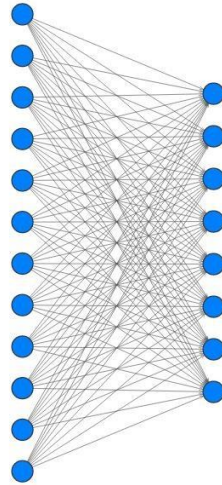


Figure 8 Fully connected layer

2.2.2 Residual Networks

The team led by Kai-Ming He at Microsoft Research came up with the idea for the Residual Neural Network (Resnet) in 2015, and it went on to win the ImageNet large-

scale image recognition competition that same year. In the meanwhile, the number of parameters that make up its network model are much lower than the number of parameters that made up the models that had the highest performance in the years before. It is commonly used as a base network for various target tasks, and it is then micro-tuned (Fine Tune) in conjunction with its specific tasks. This is because of its powerful performance and extremely scalable features, which has led to its widespread application in a variety of fields, including image recognition and image segmentation. In addition, it has been widely used in these areas.

The basic module of a residual network is the residual unit, which consists of a convolutional layer, a Batch Normalization (BN) layer and a Rectified Linear Unit (ReLU) function. The equation for a single residual unit is shown in equation (2.8).

$$x_{n+1} = x_n + F(x_{n-1}) \quad (2.8)$$

$F(x_{n-1})$ in Eq. (2.8) denotes the residual function, and the residual block is shown in Figure 9. The residual function performs 1×1 convolution operation, 3×3 convolution operation and 1×1 convolution operation on the input feature map in turn. Each of these convolution operations is followed by a batch normalization process. A 1×1 convolution operation is utilised in the first convolution to lower the number of input feature channels, and a third 1×1 convolution operation is used to raise the number of channels to match the number of input feature channels. This is done in an effort to cut down on the second convolutional layer's parameter count, which in turn cuts down on the network's overall parameter count and computation.

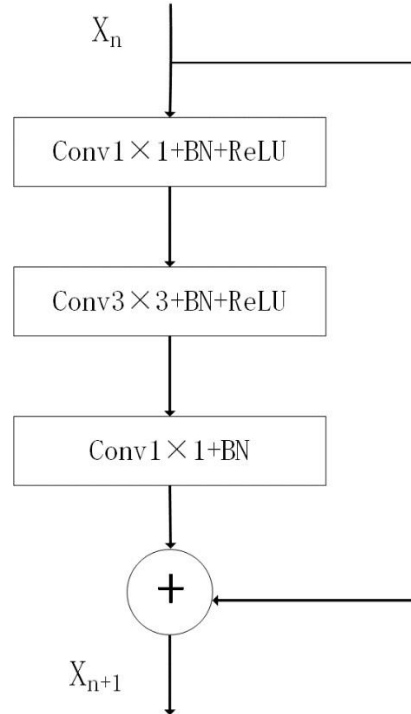


Figure 9 Residual Unit

The residual network consists of 4 blocks, each of which consists of a stack of residual units. The residual network can be divided into, depending on the number of network layers, Resnet18, Resnet34, Resnet50, Resnet101 and Resnet105. The specific network structure and the amount of Floating Point Operations (FLOPs) for a given size of feature map are shown in Table 1.

Table 1 Residual network structures

NAME	FEATURE MAP	RESNET18	RESNET101
Conv1	112*112	Conv7*7, 64, stride2	
		Max pool 3*3, stride2	
Conv2_x	56*56	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
Conv3_x	28*28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
Conv4_x	14*14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$
Conv5_x	7*7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1*1	Average pool, 1000-d, fc, softmax	
FLOPs		1.8×10^9	7.6×10^9

The proposed residual network alleviates the problem of gradient disappearance caused by deeper network models due to the limited representational power of shallow models, making it possible to train deeper network structures. The number of layers in a deep model is also critical to the model performance, and a deeper network structure can extract richer semantic information. Therefore, in most current computer vision tasks

such as target recognition, target detection, target tracking and semantic segmentation, residual networks are generally used as the feature extraction networks, also called backbone networks or front-end modules, for these computer vision tasks.

2.2.3 Principles And Analysis Of Network Optimisation Algorithms

Given a loss function, finding the appropriate value to maximise or minimise that function in the domain of definition is the purpose of the optimisation algorithm. The algorithm for solving neural network models is unconstrained optimisation: $\min_{x \in R^n} (J(y, \hat{y}))$. The most common optimization algorithms currently used in deep learning are the following.

- Mini-Batch Gradient Descent (SGD)

A tiny sample of the batch is used by the very efficient deep learning technique stochastic gradient descent, i.e. m ($m \ll n$, n is the number of whole training samples) samples for computation, so that it can reduce the variance in parameter updates and converge stably. Its formula is as follows:

$$\omega = \omega - \alpha \nabla_{\omega} J(\omega, x^{(ii+m)}, y^{(ii+m)}) \quad (2.9)$$

where: α is the step size (learning rate), which determines the length of each descent in the negative direction of the gradient during the gradient descent iteration. $x^{(ii+m)}$ is the random small batch input to the network model, $y^{(ii+m)}$ is the output of the network, ∇_{ω} denotes the partial derivative of ω with respect to the loss function $J(\omega, x^{(ii+m)}, y^{(ii+m)})$.

There are several possible problems with using stochastic gradient descent: (a) because of the need to pre-set the step size, when the step size is too small it will result in too slow convergence and too large will cause the loss function to oscillate or even deviate constantly at very small values. In a highly non-convex function like a neural network, optimising the loss function will result in local minima, or getting trapped at the saddle point, because the loss values around the saddle point are the same and the gradient in all dimensions is close to zero, causing stochastic gradient descent to be easily trapped here. (b) Stochastic gradient descent uses the same step size for all parameter updates of the network model, and when faced with a training dataset that is sparse, it is desirable to have a larger update step for features that occur less frequently in the data.

To solve the problem in (a) above, Momentum is proposed, which can accelerate the convergence speed of the stochastic gradient descent algorithm and avoid the emergence of local minima. Its formula is as follows.

$$\begin{cases} v = \gamma v + \alpha \frac{\partial}{\partial \omega} J(\omega) \\ \omega = \omega - v \end{cases} \quad (2.10)$$

- Adam algorithm

Adam is a first-order optimization technique that, by repeatedly updating the network connection weights depending on the training dataset, may take the role of stochastic gradient descent. The Adam algorithm resolves the issue raised in (b) of the SGD algorithm in the preceding section by creating mutually uncorrelated adaptive learning rates for various parameters by computing first and second order moment estimates of the gradients. This differs from stochastic gradient descent, which maintains a single step to update all network parameters.

Adam algorithm can be used to train networks with relatively sparse datasets, and in addition it has proven to be computationally efficient in many different neural networks, takes up little computer memory, is easy to implement, is scaling invariant on the diagonal of the gradient, has no smoothness requirement on the objective function, and can handle noise in the dataset better.

2.2.4 Attentional Mechanisms

In the study of human vision, because the human eye has a limited capacity to process visual information, it selectively focuses on the most valuable information and ignores secondary information in order to better access the important information in vision. This mechanism is known as the attention mechanism.

The basic principle of the attention mechanism in human vision is the same in computer vision: an allocation scheme of more of the best and less of the worst, with the objective of focusing on the crucial and valuable features among the numerous features and suppressing the non-essential and secondary features to remove the interference of secondary information in classification. The attention mechanism in semantic segmentation is divided into three types of attention: channel attention, spatial attention and self-attention, based on the relationship between channels, spatial locations and the relationship between the query location and all other locations in the feature map. There is also a hybrid attention that uses several types of attention jointly.

- Channel Attention

SENet[25] proposes the Squeeze-and-Excitation (SE) module by investigating the dependencies between feature map channels. The SE module allows the model to learn and use global information to selectively reinforce important feature information and suppress redundant, useless feature information. The SE module is shown in Figure 10.

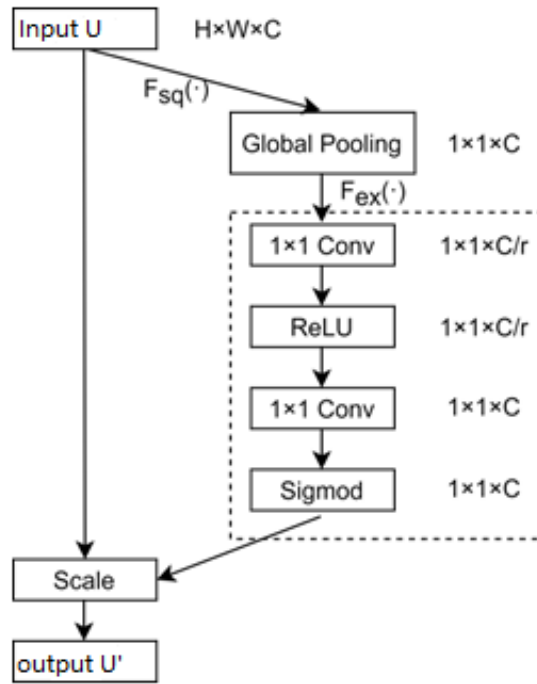


Figure 10 SE Module

The SE module has an input feature $U \in R^{C \times H \times W}$, where C, H and W are the number of channels, width and height of the input feature, respectively, and is divided into three parts: compression, excitation and weighting.

The compression operation is $F_{sq}(\cdot)$, which calculates the distribution of the C feature channels in the feature map U by globally averaging the pooling of the input features U. The formula is shown in (2.11).

$$Z_C = F_{sq}(U_C) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W U_C(i, j) \quad (2.11)$$

The excitation operation $F_{ex}(\cdot)$ is used to fully capture the dependencies between channels. Its equation is shown in (2.12).

$$S = F_{ex}(Z, W) = \sigma(g(Z, W)) = \sigma(W_2 \delta(W_1 Z)) \quad (2.12)$$

σ denotes the ReLu activation function, $W_1 \in R^{\frac{C}{r} \times C}$, $W_2 \in R^{C \times \frac{C}{r}}$ are the parameters of the two fully connected layers, and r is the parameter of the channel number scaling, which is used to reduce the number of channel dimensions and thus the computational effort.

The weighting operation $F_{scale}(\cdot)$ multiplies the channel weights S_C obtained from the excitation operation and the input feature map U_C in the channel direction to obtain the output features U_C' . The equation is shown in (2.13).

$$U_C' = F_{scale}(U_C, S_C) \quad (2.13)$$

- Spatial Attention

It is possible to collect information about "content" and "location" in both the channel and spatial dimensions via the use of the convolution operation [52]. As a result, after the presentation of SENet, CBAM[26] inferred attention along the channel as well as the spatial dimensions of the feature map. The connection between channels is taken into account when generating a channel attention map, while the spatial relationship between features is taken into account when generating a spatial attention map. In contrast to channel attention graphs, spatial attention graphs place a greater emphasis on location information, namely which feature that is situated in space should be the primary focus of attention. Figure 11 illustrates the organisation of attention as it relates to space.

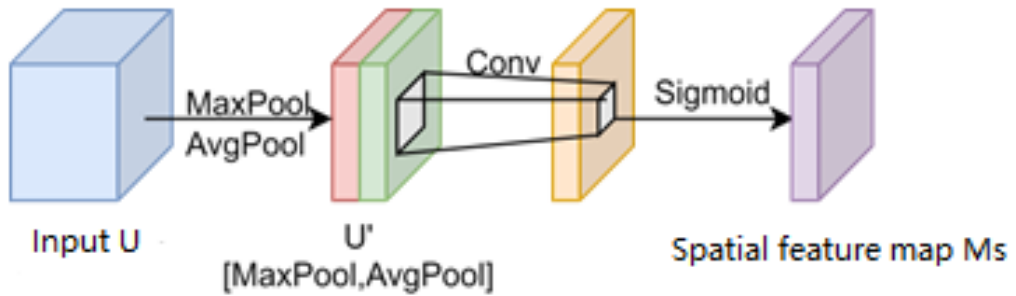


Figure 11 Spatial Attention Map

- Mixed attention

Convolutional neural networks are able to learn information about content and location in the channel and spatial dimensions respectively. CBAM attempts to extract meaningful features by concatenating channel attention and spatial attention, and its structure is illustrated in Figure 12. CBAM enables each branch using the CBAM module to learn content information and location information in the channel and spatial dimensions, respectively, by concatenating both channel and spatial attention modules.

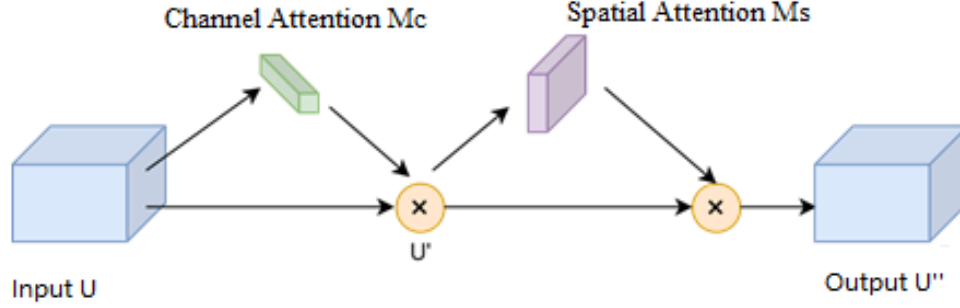


Figure 12 CBMA network diagram

The CBAM module effectively aids the flow of information within the network by learning enhanced or inhibited information. The specific steps for blending attention are as follows.

- (1) The input feature $U \in R^{C \times H \times W}$ is passed through the channel attention module to generate the channel attention map $M_c \in R^{C \times 1 \times 1}$. The channel attention map M_c is multiplied with the input feature map U to obtain the new feature map $U' \in R^{C \times H \times W}$ with the following equation.

$$U' \in M_c(U) + U \quad (2.14)$$

- (2) U' generates the spatial attention map $M_s \in R^{1 \times H \times W}$ through the spatial attention module. The spatial attention map M_s is multiplied with U' to obtain the output feature U'' , and the equation is shown below.

$$U'' = M_s(U') + U' \quad (2.15)$$

2.2.5 Evaluation Indicators

There are four widely used assessment metrics in the area of picture semantic segmentation: pixel accuracy (PA), mean pixel accuracy (MPA), intersection over union (IoU), and mean intersection over union (MIoU) [53].

The simplest assessment technique, PA, is the ratio of the number of properly identified pixels to the total number of pixels in the picture (2.16).

Pixel accuracy is computed using (2.17), where the ratio of correctly predicted pixels to the total number of pixels in a semantic category is represented by MPA, and IoU is computed using (2.18), where the ratio of the intersection and concatenation of the sets of annotated and predicted categories in the truth map is represented by IoU.

The average of each category's IoU is represented by MIoU, a division of IoU, as illustrated in (2.19). The semantic segmentation problem uses MIoU, a common metric, to assess how well the segmentation worked. The evaluation approach in this research is MIoU.

$$PA = \frac{\sum_{i=0}^k N_{ii}}{\sum_{i=0}^k \sum_{j=0}^k N_{ij}} \quad (2.16)$$

$$MPA = \frac{1}{k+1} \sum_{i=0}^k \frac{N_{ii}}{\sum_{j=0}^k N_{ij}} \quad (2.17)$$

$$IoU = \frac{\sum_{i=0}^k N_{ii}}{\sum_{i=0}^k \sum_{j=0}^k N_{ij} + \sum_{i=0}^k \sum_{j=0}^k N_{ji} - \sum_{i=0}^k N_{ii}} \quad (2.18)$$

$$MIoU = \frac{1}{k+1} \sum_{i=0}^k \frac{N_{ii}}{\sum_{j=0}^k N_{ij} + \sum_{j=0}^k N_{ji} - N_{ji}} \quad (2.19)$$

2.3 Summary of the chapter

This chapter begins with an introduction to deep learning-based image segmentation methods, starting with the fully convolutional neural network FCN, followed by the U-Net network and SegNet network, which are derived from code-and-decode based structures, and then the more classical DeepLab family of network structures in lightweight network research are presented separately, with details of how each version can be improved. Finally, a large number of network structures designed using different acceleration strategies, such as Enet networks, ICNet networks and BiSeNet networks, emerge from the research on lightweight segmentation methods. A brief introduction to the basic theory of neural networks is then given, focusing on several common components of neural networks, followed by an introduction to the commonly used and efficient residual networks and a detailed description of the residual unit. The basic theory and related techniques introduced in this chapter will serve as a solid theoretical basis for network design and improvement in subsequent chapters.

CHAPTER 3 - A BISECTION-BASED SEMANTIC SEGMENTATION ALGORITHM FOR PORTRAITS

3.1 Overview

Jonathan Long's FCN accomplishes end-to-end semantic segmentation by exploiting jump joints to make up for the spatial position information lost through several pooling processes of the high-level feature maps in VGG. Semantic segmentation is a pixel-level classification job. FCN-16s, on the other hand, can only employ one jump link to directly upsample the feature map by a factor of 16, which leads in poor segmentation outcomes. In order to better identify the type of semantic information held in each spatial location and enhance the semantic segmentation effect, UNet uses the high-resolution feature map on the encoder side to gradually recover the lost spatial location information in the low-resolution feature map on the decoder side. This is a technique to increase semantic segmentation's potency.

While the encoder-decoder structure may help semantic segmentation work better by recovering a certain amount of spatial position information, it can also result in a lot of computation and sluggish inference. In order to assure accuracy and speed of inference to meet the demands of real time, it is necessary to preserve more spatial information and make the feature map rich in perceptual field while minimising the amount of computation and boosting the speed of inference. Bisenet's dual-path strategy strikes a nice compromise between speed and precision. In this study, the feature extraction backbone networks resnet18 and resnet102 are used for contextual branching, and their effects on the model's inference speed and accuracy are examined.

3.2 Two-branch Convolutional Neural Networks

3.2.1 Spatial Branching Network

The spatial branching is performed using only three convolution operations, with each convolution followed by the addition of a batch normalisation layer and Relu activation, with a step size of 2, a padding of 1 and a convolution kernel of size 3x3. The feature map is only 1/8 of the original map after the convolution operation. This approach allows the image resolution to be reduced less and more spatial information to be fully retained. The spatial branching operation is shown in Figure 13.

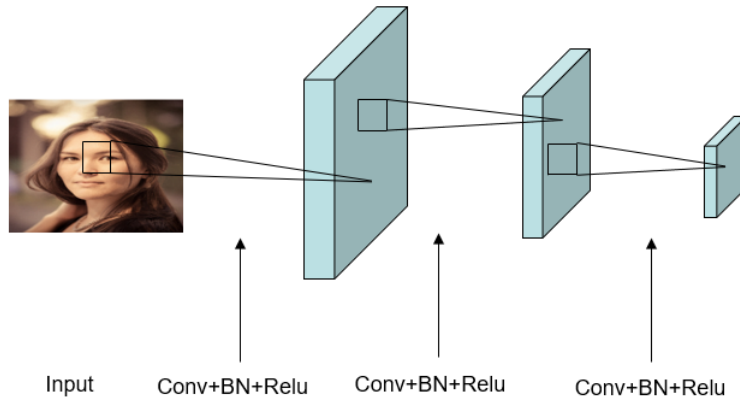


Figure 13 spatial branching network

3.2.2 Contextual Branching Network

The contextual branching network uses a fast downsampling strategy to obtain rich perceptual field information. In this project, the resnet18 and resnet101 mentioned above are used as the downsampling backbone network, which decreases the image resolution by 32 times through 4 downsamplings to obtain a rich perceptual field, and the attention mechanism module mentioned above is added to the 1/16 and 1/32 feature maps to make the network focus more on useful information. The contextual branching network is shown in Figure 14.

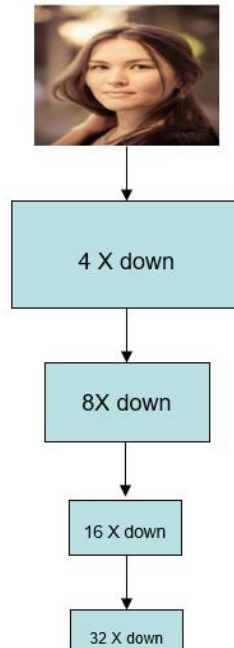


Figure 14 Contextual Branching Network

3.2.3 Feature Fusion Module

The 16x and 32x downsampled feature maps are first raised to 8x, then the feature fusion module stitches the feature maps obtained on the two branches in dimension, and performs a convolution operation, then an attention module operation, and finally fuses the results and performs 8x upsampling, and outputs the results. The overall network structure is shown in Figure 15.

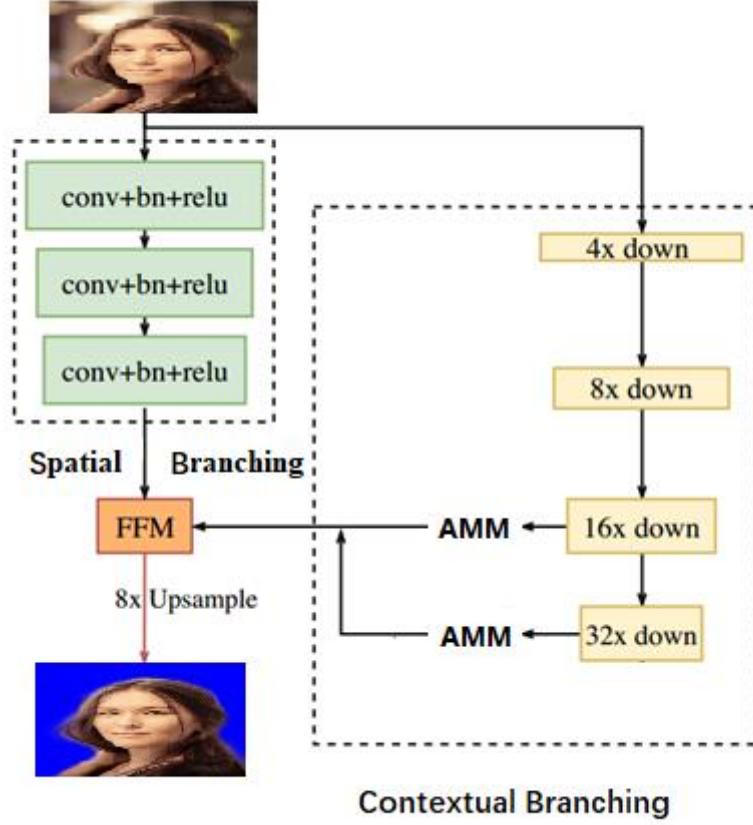


Figure 15 The Overall Network Structure

3.3 Results Analysis

In this experiment, only the image resolution size and the backbone network were changed, while other parameters were kept constant. Three different image resolutions were trained, such as 112x112, 224x224 and 512x512, as well as different backbone network models were trained as context branches for fast downsampling to extract features. The experimental results are shown in Table 2.

Table 2 Training results at different resolutions and backbone

Resolutions	backbone	MIoU	backbone	MIoU
112*112	resnet18	93.10%	resnet101	94.76% (+1.66%)
224*224	resnet18	95.23% (+2.12%)	resnet101	96.13% (+3.03%)
512*512	resnet18	95.62% (+2.52%)	resnet101	96.25% (+3.15%)

From the training results, it can be seen that the network learns less and has a lower MIoU value when the image resolution is small and the depth of the feature extraction network is small. As the image resolution increases, the model MIoU value increases. However, when increasing to 512x512, the effect of increasing MIoU values is already small. It can also be seen that the resnet101 network is more capable of learning, with a final increase of 3.15% relative to the 112x112 resolution when the backbone is resnet18.

The increase in MIoU inevitably increases the number of parameters in the model, leading to an increase in model training time as well as inference time. Figure 16 shows the time consumed for inference at MIoU values of 93.10% and 96.25% respectively.

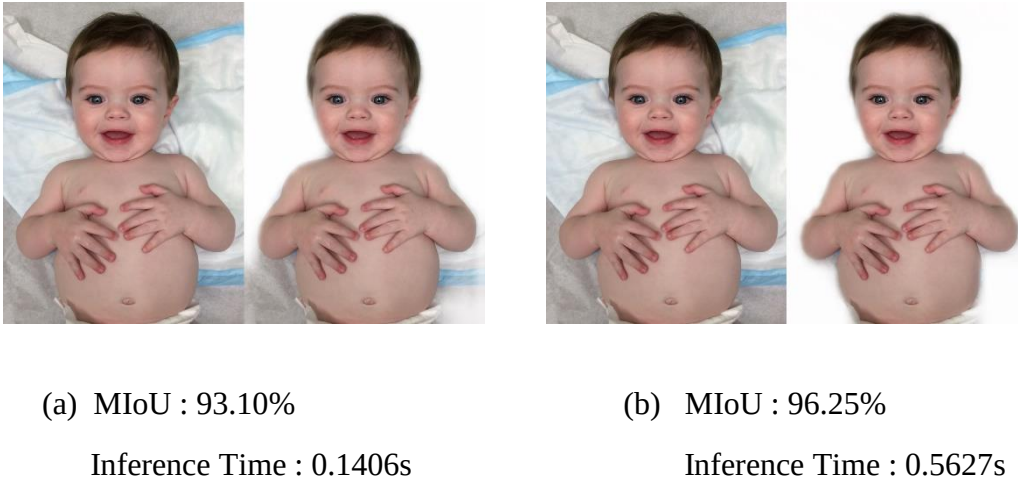


Figure 16 MIoU values and inference times

The use of a large resolution MIoU has improved, too small a resolution leads to missing information and affects accuracy, but training speed is improved. The MIoU can be improved by using a large model, which has a better fit.

CHAPTER 4 - DATA ENHANCEMENTS

Three datasets were used in this project, The Portrait Dataset 2000 as the base dataset, augmented with some of the data from The Portrait Dataset 34427 for data volume, and some of the sub-data from the COCO dataset for home-grown dataset manipulation, combined with deformation enhancement and texture enhancement methods to improve the robustness of the network model.

4.1 Introduction To The Dataset

4.1.1 The Portrait Dataset 2000

The Portrait Dataset 2000[54] is an image segmentation dataset containing 2000 images and the corresponding Matting annotation results, all of which are selfie images. The dataset was released by the Chinese University of Hong Kong in 2016, with a training set of 1700 images and a test set of 300 images, all sourced from Flickr portraits, with the original image resolution size of 600×800 , where Matting was generated using closed-form matting and KNN matting methods.

Due to the high commercial value of portrait segmentation datasets, there are very few publicly available large-scale datasets. This dataset is one of the earlier released and more widely used datasets, and it has several important features:

- The images are of uniform resolution, clearly captured and of high quality.
- All images are portraits of the upper body, with the portrait area occupying at least $\frac{2}{3}$ of the image in both length and width.
- There is very little variation in the pose of the figures, all are small frontal views with simple backgrounds.

Figure 17 lists some of the images and tags in The Portrait Dataset.

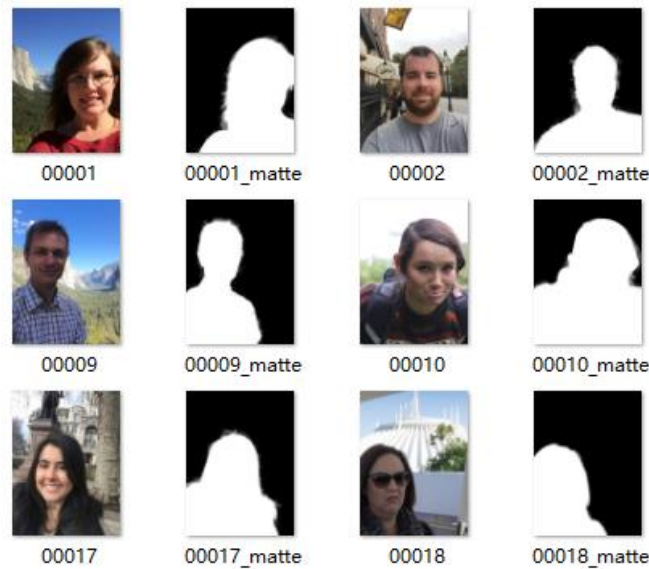


Figure 17 Data sample in The Portrait Dataset 2000

4.1.2 The Portrait Dataset 34427

The Portrait Dataset 34427 [55] contains 34,427 Portrait images collected from Flickr, Taobao and Baidu, and annotated with high quality by Beijing Star Aggregation Technology Co., LTD. The final images in this dataset were generated by running a face detector on each image to generate bounding boxes, which were then automatically scaled to the size of 800×600 pixels and cropped. The 34,427 images were divided into two groups, 30,000 training datasets and 4,427 validation datasets. In the process of using Opencv to extract labels from images, these portrait datasets need to have great differences in age, clothing, brightness, accessories, hair style and head position, etc., to ensure that the trained model has better robustness. In addition, each picture undergoes a pixel-by-pixel hand annotation process. A selection of data sets is shown here as an example.



Figure 18 Data sample in The Portrait Dataset 34427

4.1.3 MS COCO Dataset

Microsoft Common Object in Context (MS COCO) [56] is a large-scale dataset of 328,000 images, with the first release in 2014 and the final release in 2017 adding 123,000 unannotated new datasets. This dataset is suitable for various detection tasks, such as object detection, image segmentation, keypoint detection and so on. This project will use 118,287 images of 80 categories of objects for object detection, select a specific image as the background, and make the dataset. The data sample is shown in Figure 19.

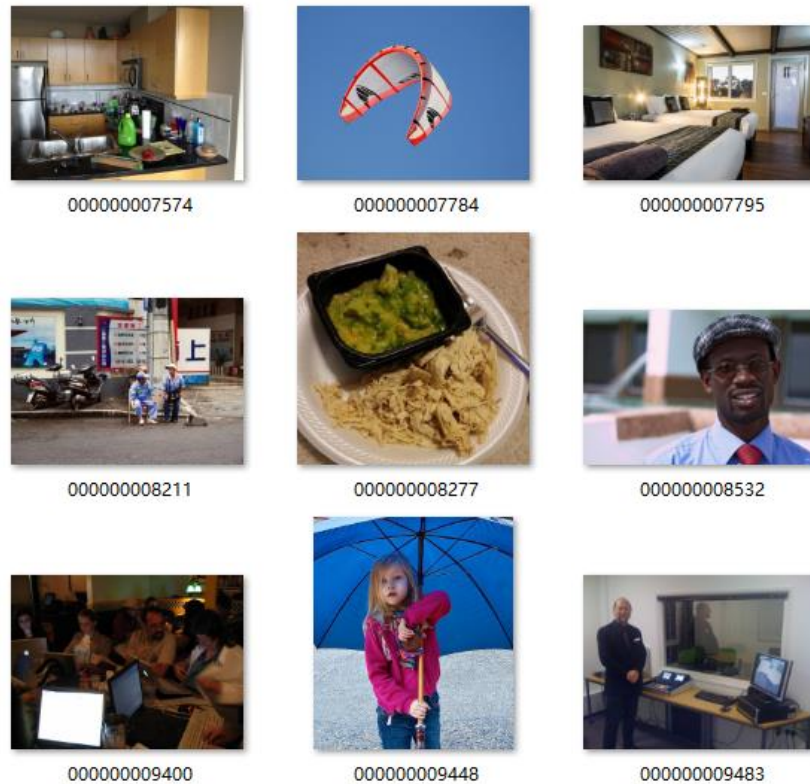


Figure 19 Data sample in MS COCO

4.2 Data Enhancement Method

4.2.1 Data Set Stitching

The most basic way of data enhancement is to increase the amount of data for training. As the dataset released by the Chinese University of Hong Kong has only 2000 images and the training dataset has only 1700 images, the model cannot achieve the best results. In order to improve the model's fitting ability as well as its generalisation ability, The Portrait Dataset 34427 was stitched on top of this dataset to increase the amount of data that the model could learn from.

The stitching operation on the dataset is implemented using the `torch.utils.data.ConcatDataset` provided in the Pytorch framework. The implementation principle is shown in Figure 20. The stitching operation on the dataset is implemented using the `torch.utils.data.ConcatDataset` provided in the Pytorch framework. The `Dataloader` class provided by Pytorch is used to load the data from the hard disk. `Sampler` is responsible for generating the index of the data and `getitem` in the `Dataset` gets the images and labels from the hard disk based on the index value. The Batch Data is retrieved by `collate_fn`. The two datasets are stitched together and fed into the model using `ConcatDataset`.

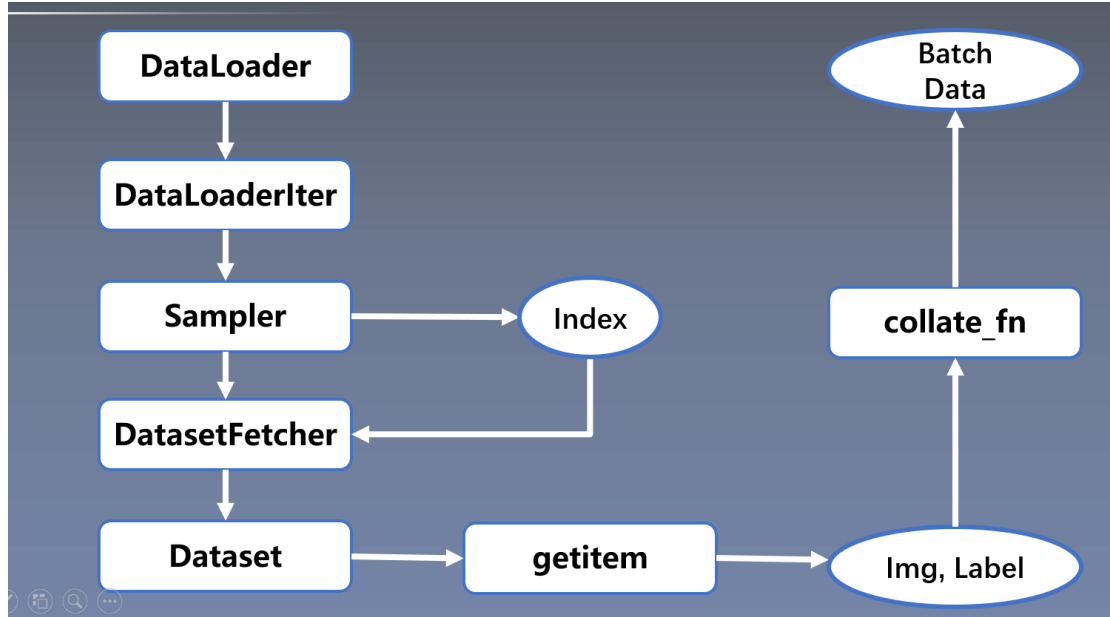


Figure 20 Model data loading process

4.2.2 Home-made Datasets

To increase the richness of the data, the dataset was created using sub-data from the MS COCO dataset as the portrait background and portraits from The Portrait Dataset 2000 as the data foreground. The principle of fusion of the data can be explained by equation (3.1).

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i, \quad \alpha_i \in [0, 1] \quad (3.1)$$

Where: I denotes the input image, i is the pixel point index of the image, F and B are the foreground and background of image I respectively, and α_i is defined as the foreground opacity of image I_i at pixel point i .

The flow chart for image fusion is shown in Figure 21.



(a) The input image



(b) The foreground image



(c) The input image



(d) The background image



(d) The fusion image

Figure 21 Image fusion process

The home-made dataset needs to match the reality as much as possible when choosing the background image, if it differs from the reality a lot, it will affect the training effect of the model instead. There are many categories in the MS COCO dataset that do not match the portrait background, such as the food category. If the food category is selected as the portrait background, as shown in Figure 22, this will result in the appearance of the character's head on the food, which is very different from reality, thus leading to a decrease in the accuracy of the model. By analysing the MS COCO dataset categories, indoor and outdoor scenes in the data were finally selected and the presence of people in the scenes was excluded.

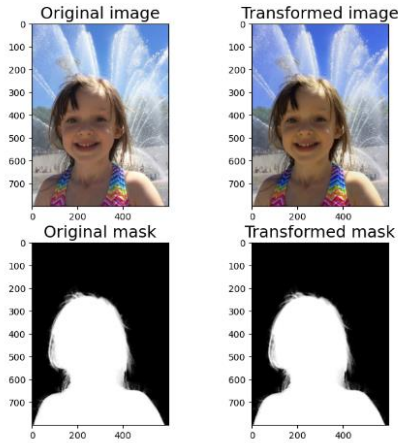


Figure 22 Abnormal data in the home-made data set

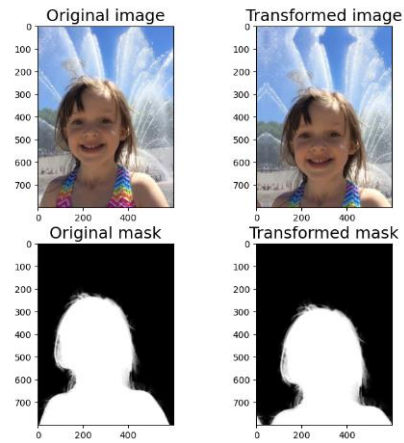
4.2.3 Online Data Enhancements

The online data enhancement method is used in this project, where data enhancement is performed before the training data enters the network, and the sum of the probabilities of the deformation enhancement and texture enhancement operations is 0.5. The input image was also normalised using the image mean ([104.94,112.78,122.68], BGR order). The normalisation formula is $(\text{image} - \text{mean}) / 255$.

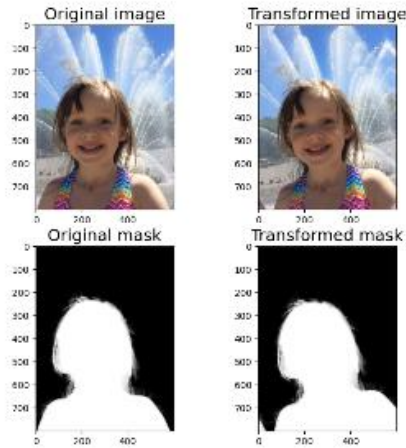
In the process of data enhancement, the original image needs to be rotated and imitated, and the corresponding labels need to be subjected to the same operations, so the Albumentations library is used to enhance both the original image and the labels. An example of partial data enhancement is shown in Figure 23.



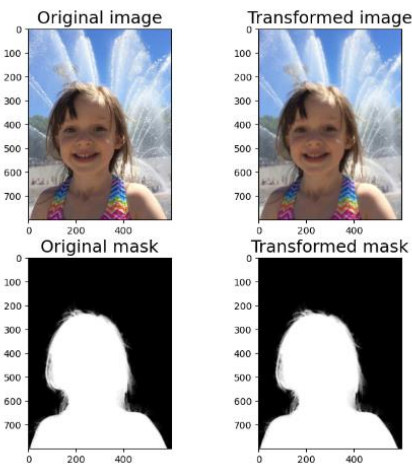
(a) Colour Scrambling



(b) Panning and Scaling



(c) Random Rotation



(d) Random Blurring

Figure 23 Selected data enhancement samples

4.2.4 Transfer Learning

Transfer learning is a training process that uses data, targets and similarities within a model to apply a model that has been trained in the original domain to a new task domain. Its core component is the discovery of similarities in the original domain and the target domain.

It is generally recognised that a big training dataset is necessary for effective deep learning, but building a large, qualified dataset is challenging owing to the high cost of data collection and labelling, which results in a lack of data in the training set and low accuracy of the trained model. The efficacy of learning on diverse task goals may be enhanced through transfer learning. The convergence and accuracy of the model may both be improved by using pre-trained neural network model parameters. Before

training, initialization parameters from the pre-trained Resnet18 and Resnet101 model parameters were employed in this research.

4.3 Results Of Data Enhancement Experiments

The model was trained for several data improvements independently, and the Backbone network was resnet18, in order to confirm the impact of the data enhancement approach on the model's accuracy. Table 3 presents the experimental outcomes. In this experiment, the accuracy of the model is examined under several conditions, including no data augmentation, colour perturbation, level switching, data fusion, and the inclusion of extra data. Only 1700 fused data and extra data for analysis were added in order to speed up training.

Table 3 Results of data enhancement experiments

Data Enhancements	Miou(Train)	Miou(Valid)	Lifting and Lowering
None	97.43%	93.74%	-
Color Scrambling	97.31%	94.82%	1.08%
Horizontal Flip	97.08%	94.14%	0.5%
Fusion (1700)	97.52%	94.44%	0.7%
Extra (1700)	97.73%	95.22%	1.48%
All	97.89%	95.28%	1.54%

As can be seen from the training results, adding additional data sets is the most effective way to enhance the data. As portrait images are susceptible to lighting, the use of colour perturbation can also substantially increase the accuracy of the model, and all data enhancement methods performed simultaneously can result in the highest model accuracy.

CHAPTER 5 - ANALYSIS AND DETERMINATION OF LOSS FUNCTIONS

The evaluation of the disparity between the actual value and the value that was anticipated requires the use of a loss function, which is also often referred to as a cost function. In the field of deep learning, the loss function is often used as a learning criteria in conjunction with an optimization problem. This is because the estimate of network parameters is accomplished by striving to minimise the loss function. This section provides an explanation of the loss functions that were implemented for this project. When training the model using a variety of loss functions, the accuracy of the model is evaluated, and from those evaluations, the loss function that is most appropriate for this model is chosen.

5.1 Softmax Function

When designing a loss function in a semantic segmentation network, it is usually necessary to first standardise the samples to be predicted using the Softmax [57] function. A typical example of a logical function is the normalised exponential function, often known as the Softmax function. The normalised exponential function transfers a K-dimensional vector of any size in the real domain to another K-dimensional vector with a value range between 0 and 1, and it makes the total of all the vectors equal to 1. In other words, the function normalises the vectors such that their sum equals 1. The logarithm of the gradient of a finite term discrete probability distribution is normalised as a result of this, which is the key reason for its relevance.

In this project, each pixel in the image needs to be classified and judged, the category probability value for each pixel is obtained, the numerical magnitude of the probability value is judged and the largest probability is selected as the result for that pixel. The use of Softmax allows the probability range to be compressed to between (0,1). The formula is shown in (5.1).

$$p_j = \frac{e^{z_j}}{\sum_{i=1}^C e^{z_i}} \quad (5.1)$$

where j is the individual category and C is the total category.

5.2 Binary Cross Entropy Loss

First presented in information theory, cross-entropy measures how much longer it takes to code the truth probability distribution $p(y_i)$ than the observed probability distribution y_i when both are based on the same event measure. When performing image semantic segmentation, a unique category must be assigned to each pixel, and

both the truth map and the prediction result map can be thought of as discrete distributions, the former assigning a probability of 1 to the correct category corresponding to each pixel and a probability of 0 to the missed category. Next, the loss function is derived from the discrete distribution of the cross-entropy function expression to create the binary cross-entropy loss function, and the network parameters are trained under supervision using the final prediction result map. In equation we see the binary cross-entropy function (5.2).

$$BCELoss = -\frac{1}{n} \sum_{i=1}^n [y_i \cdot \log p(y_i) + (1 - y_i) \cdot \log(1 - p(y_i))] \quad (5.2)$$

where y_i is the binary label value 0 or 1, and $p(y_i)$ is the probability of belonging to the label value of y_i .

5.3 Focal Loss

Focal loss [58] is used in the field of target detection to solve the class sample imbalance problem by modifying the loss function to improve the performance of one-stage target detection, and the idea of Focal loss can also be used for the possible data imbalance problem in the field of image segmentation. In order to solve the problem of difficult samples, Focal loss uses the idea of dynamic weighting, and the loss function is down-weighted for samples with high confidence. For samples with low confidence, the loss function is weighted, so that when the network is back propagated, the samples with low confidence can provide a greater proportion of the gradient, and more information is obtained from the unlearned samples. The formula is shown in 5.3.

$$FL = \begin{cases} -\alpha(1-p)^\gamma \log(p) & \text{if } y = 1 \\ -p^\gamma \log(1-p) & \text{if } y = 0 \end{cases} \quad (5.3)$$

Unaffected when p probability is low, 1-p approaches 0 when probability is high, reducing the contribution of easy-to-score samples to the total loss. α controls sample weights, γ solves sample imbalance problem, controls difficult sample problem.

5.4 Dice Loss

Dice loss [59] is a measure function used to evaluate the similarity between two events, taking values in the range (0, 1), when the result is closer to 1, it means that the two events are more similar, and vice versa, the greater the difference between the two events. the Dice loss formula is shown in (5.4).

$$\text{Dice loss} = 1 - \frac{2|X \cap Y|}{|X| + |Y|} \quad (5.4)$$

where $|X \cap Y|$ is the intersection between X and Y . $|X|$ and $|Y|$ denote the number of elements of X and Y respectively.

5.5 Results of Different Loss Functions

In order to analyse the effect of each loss function on the model, this paper uses resnet18 as the backbone network, the image size is 224x224 and other parameters are the same, BEC loss, Dice loss, Focal loss and BCE loss + Dice loss are conducted respectively.

Table 4 Training results with different loss functions

Loss Functions	mIoU	Lifting and Lowering
BCE Loss	95.51%	NA
Dice Loss	95.38%	-0.13%
BCE + Dice Loss	95.61%	+0.1%
Focal Loss alpha=0.5, gamma=0.5	62.64%	-32.87%

The BCE + Dice Loss model performs best, probably because the BCE Loss and Dice Loss models are examined from different perspectives, BCE Loss from a microscopic pixel-by-pixel perspective and Dice Loss from a global perspective. When there is an imbalance in segmentation content, Dice Loss tends to learn large targets and ignore small samples, but BCE Loss still learns small samples.

Focal Loss performs the worst because the results are very sensitive to the choice of the two hyperparameters α and γ , which are also more difficult to adjust.

CHAPTER 6 - MODEL TRAINING, REASONING AND ERROR CASE ANALYSIS

6.1 Pytorch Network Framework

In the field of deep learning, Pytorch and TensorFlow have become the two most widely used strength network frameworks in academia and industry respectively, while the closely following frameworks such as Keras and MXNet are loved by some developers due to their own uniqueness. The main stream deep learning frameworks are active on github as shown in Table 5.

Table 5 Mainstream learning framework statistics

Indicators	Tensorflow	Pytorch	Keras	MXnet
Star	167000	57700	55800	20000
Folk	87000	16100	19200	6900
Contributors	3154	2377	1043	873
Maintenance Team	Google	Facebook	Google	Amazon

Table 5 shows that Tensorflow dominates the data until August 2022, but takes a lot of time to learn due to its separate programming language, messy packaging and other drawbacks. Over time, Pytorch has gradually shown its advantages of simplicity, so this paper chooses to use the academically common Pytorch deep learning framework. Pytorch is an open source framework for building neural network models with very little code, published by the Facebook team, and is used in deep learning and other mathematically. Its main advantages include the following.

- Simplicity. Pytorch tries to avoid repetitive wheel building and aims for minimal encapsulation, so that the code is easy to read and understand.
- Speed. According to a number of evaluations, the adaptability of Pytorch does not come at the expense of its performance, and it beats other frameworks like TensorFlow and Keras. Pytorch has a higher chance than any other framework of having the same algorithm implemented in it, despite the fact that the speed of a framework is greatly dependant on the degree of coding that the programmer has attained.
- Ease of use. Pytorch is the most elegant of all deep learning frameworks; its object-oriented interface inherits the essence of Torch, and its API design and module interfaces are highly consistent with Torch's. Pytorch is designed with the user in mind, allowing for quick implementation of the user's algorithms.

6.2 Initial Optimal Learning Rate Finding Strategy

In neural network models, the learning rate controls the convergence of the model as well as the speed of training. When the learning rate is low, the network model will be more stable, but the training time will be long; when the learning rate is large, the training speed of the network increases, but the model may not converge or even scatter, so the learning rate is one of the most important parameters to adjust the neural network model.

The choice of the starting learning rate and the learning rate adjustment approach is crucial for ensuring rapid and stable model convergence at the outset of training. For this endeavour, we use a mixture of the index decay learning rate adjustment technique and the initial learning rate discovery strategy presented in the literature [60].

The initial learning rate algorithm is determined as follows, and the results are shown in Figure 24.

- In an epoch, the learning rate is gradually increased for each iteration and the change in loss is observed.
- Choose a learning rate corresponding to the point where the loss decreases the fastest.
- The final initial learning rate is determined to be 4.33×10^{-1} .

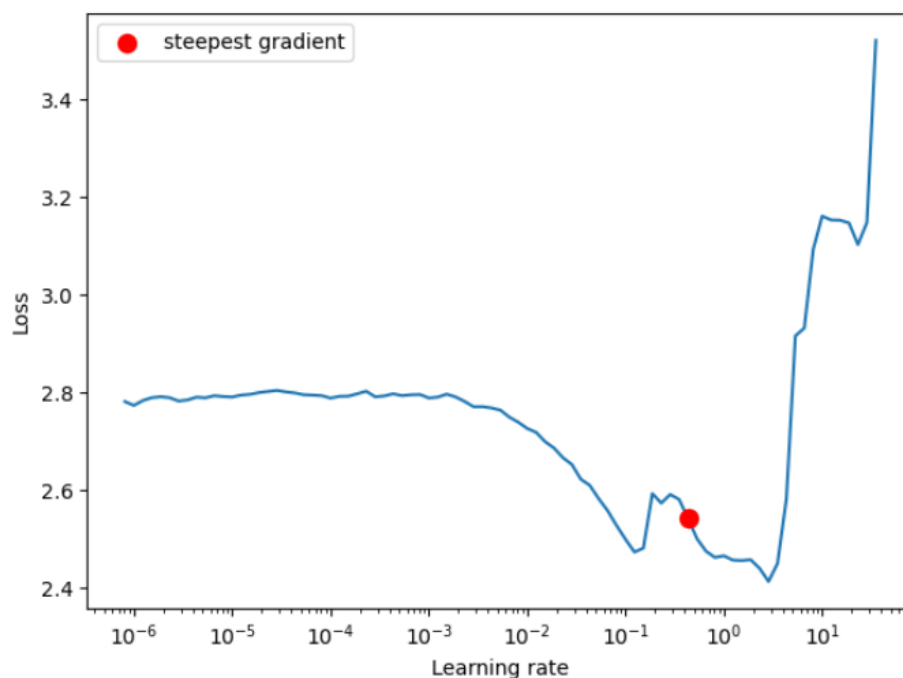
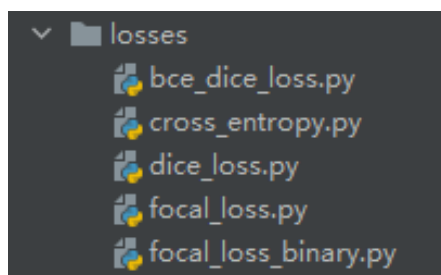


Figure 24 Determine the initial learning rate

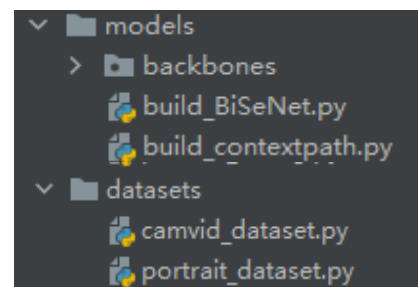
6.3 Model Training

6.3.1 Project Documentation

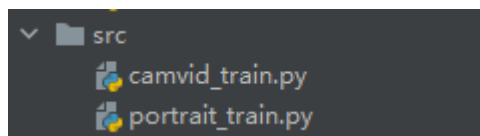
Based on the analysis of the model modules in the previous chapters, the model for this project was built in the Pytorch framework using the python language, building `portrait_dataset.py` for data enhancement and data reading, `build_BiSeNet.py` for model building, and `bce_dice_loss.py` for build loss function, `portrait_train.py` for instantiating the model and calling the training set for training and `bisenet_inference.py` for inference of the model and image segmentation. The overall code file is shown in Figure 25.



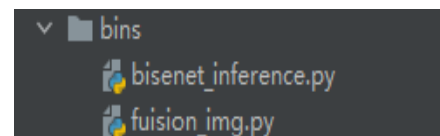
(a) Loss Function



(b) Models and Datasets



(c) Model Training



(d) Model inference

Figure 25 The overall code file

6.3.2 Project Environment

The network training and testing for this project was done under the Window 10 operating system, using the Anaconda 4.5.4 open source package manager to install the development language, image processing libraries and GPU version of the deep learning framework, and finally configuring Anaconda into the PyCharm IDE to build the network, as shown in Table 6.

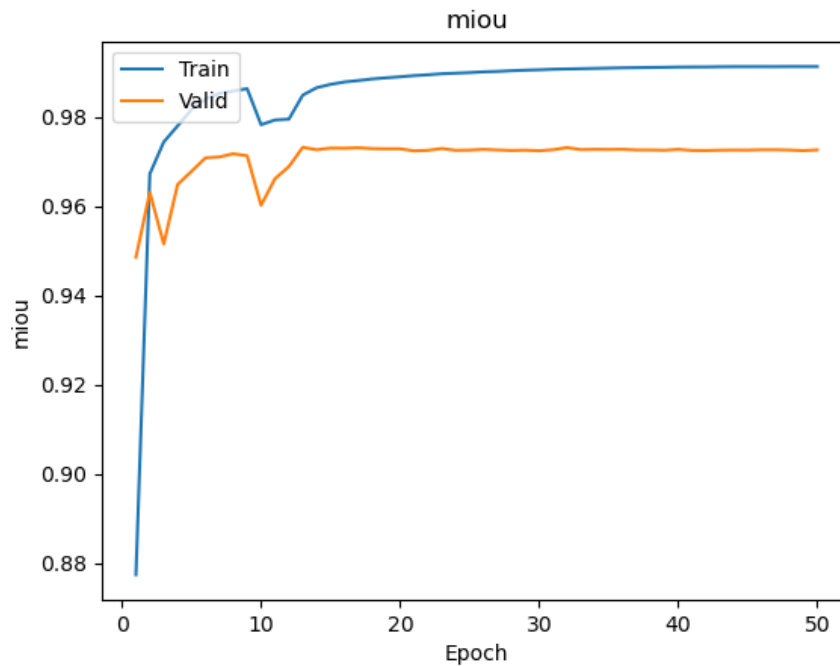
Table 6 Project Environment

Number	Environmental Categories	Type
1	Operating system	Window 10 64-bit
2	CPU	Intel(R) Core (TM) i5-10400F
3	GPU	NVIDIA GeForce RTX 3060
4	Development Language	Python 3.6
5	Deep Learning Framework	Pytorch 1.4.0
6	IDE	PyCharm2022

6.3.3 Model Training Steps and Parameter Selection

- Dataset loading. Normalisation of the training and validation set images and the data enhancement introduced in Chapter 4.
- Loss function. The loss function selection allows the results to be obtained with the optimal loss function, and the BCE + Dice loss introduced in Chapter 5.
- Definition of epoch. epoch is used to describe how many times the full dataset was used. epoch was chosen to be 50 for this project after several experiments.
- The optimizer selects the SGD algorithm, the exponential decay learning rate adjustment strategy, and the batchsize needs to be reasonably set according to the actual GPU memory size, and the batchsize size in this project is set to 4.

The final training results are shown in Figure 26. The final Miou value was 97.32%. The recall was 98.64% and the precision was 98.53%.



```
class:bg      , total num:572488420.0, correct num:569036697.0  Recall: 99.40% Precision: 99.44%
class:portrait , total num:764445980.0, correct num:761231199.0  Recall: 99.58% Precision: 99.55%
class:bg      , total num:31389687.0, correct num:30695625.0   Recall: 97.79% Precision: 97.96%
class:portrait , total num:47253513.0, correct num:46613111.0   Recall: 98.64% Precision: 98.53%
Best mIoU in 49. bg_iou:0.9583  portrait_iou:0.9722
07-18_22-44 done, best_miou: 0.9732 in :12
```

Figure 26 Final training results

6.4 Model Inference and Case Study of Errors

6.4.1 Model Inference

The test was conducted using the TTA (Test Time Augmentation) strategy (Figure 27), where the original data is augmented with different forms of data, and then the average of the results is taken as the final result. The steps are as follows.

- Uniform image scaling to 256x256.
- Top left, top right, bottom left, bottom right and centre cropping out 5 areas of 224x224.
- Flip horizontally to get 10 224x224 images.
- Inference on the 10 images, get the 10 classification probabilities and take the average output.

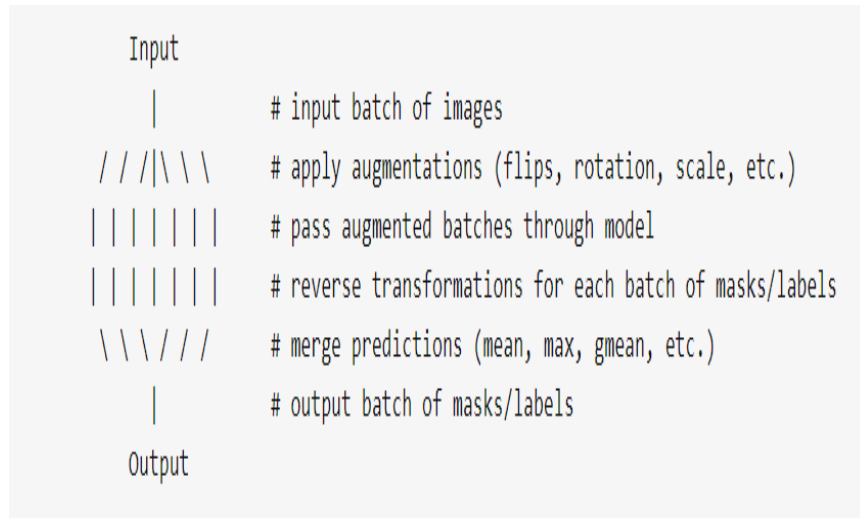


Figure 27 Test Time Augmentation

The results of the TTA strategy are shown in Table 7. Due to the use of TTA, multiple transformations can be made to an image to create multiple versions, including different region cropping and changing the scaling level, and then the data of multiple versions are calculated and finally the average output is obtained as the final result, which improves the stability and accuracy of the results.

Table 7 The results of the TTA strategy

TTA	Miou
False	97.32%
Mean	97.43% (+0.11%)

The inference result is shown in Figure 28.



Figure 28 Inference result

Real-time portrait segmentation results are shown in Figure 29, with segmentation reaching 40 fps, meeting real-time requirements.

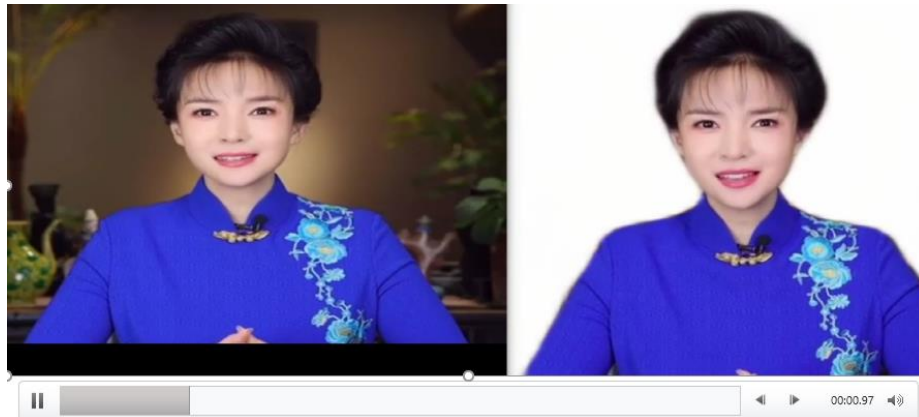


Figure 29 Real-time portrait segmentation

6.4.2 Case Study of Errors

The result with the lower Miou value of the inference result is selected for analysis of the cause of the error. Figure 29 shows some typical samples with low Miou values.

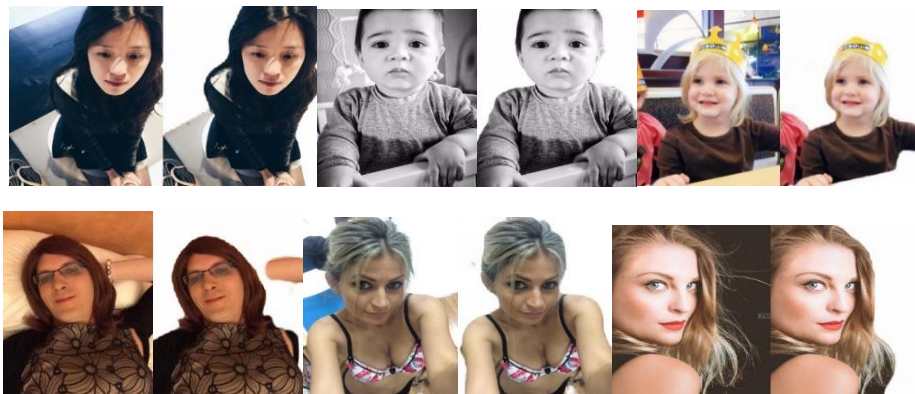


Figure 30 Sample error cases

Data that are poorly segmented can be divided into roughly 3 categories.

- Data distribution anomalies: angular anomalies, the presence of situations that do not exist in the application scenario (no human body on the bottom edge), and the presence of 2 people at the same time.
- The presence of holes: the arm forms a hole with the body; there are fewer samples in this category
- Excessive detail: poor hair segmentation

CHAPTER 7 – CONCLUSION AND OUTLOOK

7.1 Conclusion

The rapid development of deep learning has led to great breakthroughs in the field of semantic segmentation of images. With the impact of the epidemic in recent years, online activities have become extremely important and therefore the need for background blurring and portrait segmentation for conferences has increased greatly. How to ensure real-time performance while still having a good segmentation effect is the focus of this project's research.

This project adopts the idea of Bisenet, using a two-branch network to improve portrait segmentation while ensuring the real-time performance of the network, and employs three data enhancement tools to improve the robustness of the model. It is demonstrated through experiments that data augmentation methods that match realistic scenarios can better improve the model accuracy. The impact of different loss functions on the accuracy of the model is then analysed, and it is concluded that BCE + Dice loss can optimise the model from both global and detailed perspectives, and therefore the optimal results can be obtained. Finally, the model is inferred and the error cases are analysed to analyse the characteristics of samples with low Miou values.

7.2Future Work

There are three general difficulties with portrait segmentation algorithms: a) complex backgrounds and poor contrast between foreground and background; b) complex foreground areas (hair), and c) the presence of voids in portraits. The first two points are the difficulty of image keying, which directly leads to poor keying results, and the third point may be the small number of this sample. In the next step of the research, the problem of low Miou value samples can be addressed to improve the segmentation results of low Miou value samples.

References

- [1] Ding H, Jiang X, Shuai B, et al. Context contrasted feature and gated multi-scale aggregation for scene segmentation[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 2393-2402.
- [2] Lim Y W, Lee S U. On the Colour Image Segmentation Algorithm based on the Thresholding and the Fuzzy C-Means Techniques[J]. Pattern recognition, 1990, 23(9): 935-952.
- [3] Wong A K C, Sahoo P K. A Gray-Level Threshold Selection Method based on Maximum Entropy Principle[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1989, 19(4): 866-871.
- [4] Pappas T N, Jayant N S. An Adaptive Clustering Algorithm for Image Segmentation[C]. International Conference on Acoustics, Speech, and Signal Processing. IEEE, 1989: 1667-1670.
- [5] Lie W N. Automatic Target Segmentation by Locally Adaptive Image Thresholding[J]. IEEE Transactions on Image Processing, 1995, 4(7): 1036-1041.
- [6] Adams R, Bischof L. Seeded Region Growing[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, 16(6): 641-647.
- [7] Dhawan A P, Sim A. Segmentation of Images of Skin Lesions Using Colour and Texture information of surface pigmentation[J]. Computerized Medical Imaging and Graphics, 1992, 16(3): 163-177.
- [8] Xu C, Pham D L, Prince J L. Image Segmentation Using Deformable Models[J]. Handbook of Medical Imaging, 2000, 2(20): 0.
- [9] Krizhevsky A, Sutskever I, Hinton G E. Imagenet Classification with Deep Convolutional Neural Networks[J]. Advances in Neural Information Processing Systems, 2012, 25: 1097-1105.
- [10] Deng J, Dong W, Socher R, et al. Imagenet: A Large-Scale Hierarchical Image Database[C]. 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009: 248-255.

- [11] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[C]. 3rd International Conference on Learning Representations. ICLR, 2015.
- [12] Szegedy C, Liu W, Jia Y, et al. Going Deeper with Convolutions[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2015: 1-9.
- [13] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2016: 770-778.
- [14] Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE, 2017, 39(4):640-651.
- [15] Zhao H, Shi J, Qi X, et al. Pyramid Scene Parsing Network[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 2881-2890.
- [16] Chen L C, Papandreou G, Kokkinos I, et al. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs[J]. arXiv preprint arXiv:1412.7062, 2014.
- [17] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904-1916.
- [18] Chen L C, Papandreou G, Kokkinos I, et al. Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 40(4): 834-848.
- [19] Lin G, Milan A, Shen C, et al. Refinenet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 1925-1934.
- [20] Pohlen T, Hermans A, Mathias M, et al. Full-resolution Residual Networks for Semantic Segmentation in Street Scenes[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 4151-4160.

- [21] Amirul Islam M, Rochan M, Bruce N D B, et al. Gated Feedback Refinement Network for Dense Image Labeling[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 3751-3759.
- [22] Chen L C, Zhu Y, Papandreou G, et al. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation[C]. Proceedings of the European Conference on Computer Vision. Springer Verlag, 2018: 801-818.
- [23] Sifre L, Mallat S. Rigid-Motion Scattering for Texture Classification[J]. arXiv preprint arXiv:1403.1687, 2014.
- [24] Vanhoucke V. Learning Visual Representations at Scale[J]. ICLR Invited Talk, 2014, 1: 2.
- [25] Hu J, Shen L, Sun G. Squeeze-and-Excitation Networks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 7132-7141.
- [26] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]. Proceedings of the European Conference on Computer Vision. Springer Verlag, 2018: 3-19.
- [27] Wang X, Girshick R, Gupta A, et al. Non-local neural networks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 7794-7803.
- [28] Cao Y, Xu J, Lin S, et al. Gcnet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. IEEE, 2019: 0-0.
- [29] Noh H, Hong S, Han B. Learning Deconvolution Network for Semantic Segmentation[C]. Proceedings of the IEEE International Conference on Computer Vision. IEEE, 2015: 1520-1528.
- [30] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481-2495.
- [31] Ronneberger O, Fischer P, Brox T. U-net: Convolutional Networks for Biomedical Image Segmentation[C]. International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer Verlag, 2015: 234-241.

- [32] Shah S, Ghosh P, Davis L S, et al. Stacked U-Nets: a No-Frills Approach to Natural Image Segmentation[J]. arXiv preprint arXiv:1804.10343, 2018.
- [33] Drozdal M, Vorontsov E, Chartrand G, et al. The Importance of Skip Connections in Biomedical Image Segmentation[C]. 2nd International Workshop on Deep Learning in Medical Image Analysis (DLMIA)/1st International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis (LABELS). Springer Verlag, 2016: 179-187.
- [34] Zhou Z, Siddiquee M M R, Tajbakhsh N, et al. Unet++: A Nested U-Net Architecture for Medical Image Segmentation[C]. Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer Verlag, 2018: 3-11.
- [35] Li R, Duan C, Zheng S. MACU-Net Semantic Segmentation from High-Resolution Remote Sensing Images[J]. arXiv preprint arXiv:2007.13083, 2020.
- [36] Li H, Xiong P, An J, et al. Pyramid Attention Network for Semantic Segmentation[C]. 29th British Machine Vision Conference. BMVA Press, 2018.
- [37] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. arXiv preprint arXiv:1704.04861, 2017.
- [38] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted Residuals and Linear Bottlenecks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 4510-4520.
- [39] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. IEEE, 2019: 1314-1324.
- [40] Zhang X, Zhou X, Lin M, et al. Shufflenet: An Extremely Efficient Convolutional Neural Network for Mobile Devices[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 6848-6856.
- [41] Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical Guidelines for Efficient CNN Architecture Design[C]. Proceedings of the European Conference on Computer Vision (ECCV). Springer Verlag, 2018: 116-131.

- [42] Paszke A, Chaurasia A, Kim S, et al. Enet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation[J]. arXiv preprint arXiv:1606.02147, 2016.
- [43] Yu C, Wang J, Peng C, et al. Bisenet: Bilateral Segmentation Network for Real-Time Semantic Segmentation[C]. Proceedings of the European Conference on Computer Vision (ECCV). Springer Verlag, 2018: 325-341.
- [44] Zhao H, Qi X, Shen X, et al. Icnet for Real-Time Semantic Segmentation on High-Resolution Images[C]. Proceedings of the European conference on computer vision (ECCV). Springer Verlag, 2018: 405-420.
- [45] Lawrence S, Giles C L, Tsoi A C, et al. Face recognition: a convolutional neural-network approach [J]. IEEE Transactions on Neural Networks, 1997, 8(1): 98-113.
- [46] Hopfield J. Neural computation of decisions in optimization problems[J]. Biological Cybernetics, 1985, 52(3): 141-52.
- [47] Nagi J, Ducatelle F, Di Caro G A, et al. Max-pooling convolutional neural networks for vision-based hand gesture recognition[C]// 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), 2011: 342-347.
- [48] Rafferty J, Shellito P, Hyman N H, et al. Practice parameters for sigmoid diverticulitis[J]. Diseases of the Colon & Rectum, 2006, 49(7): 939-944.
- [49] Aly H A, Dubois E. Image up-sampling using total-variation regularization with a new observation model[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2005, 14(10): 1647-59.
- [50] Jang E, Gu S, Poole B. Categorical reparameterization with gumbel-softmax[J]. arXiv preprint arXiv:1611.01144, 2016.
- [51] Hinton G E, Salakhutdinov R R. Replicated softmax: an undirected topic model[J]. Advances in Neural Information Processing Systems, 2009, 22: 1607-1614.
- [52] Amirul Islam M, Rochan M, Bruce N D B, et al. Gated Feedback Refinement Network for Dense Image Labeling[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 3751-3759

- [53] Hao S, Zhou Y, Guo Y. A Brief Survey on Semantic Segmentation with Deep Learning[J]. Neurocomputing, 2020, 406: 302-321
- [54] Shen X, Tao X, Gao H, et al. Deep Automatic Portrait Matting[M]// Computer Vision – ECCV 2016. Springer International Publishing, 2016:92-107.
- [55] https://github.com/aisegmentscn/matting_human_datasets
- [56] TSUNG-YI LIN, MICHAEL MAIRE, SERGE BELONGIE, et al. Microsoft COCO: Common Objects in Context[C]. //Computer vision - ECCV 2014, part 5: 13th European conference on computer vision, 6-12 September, 2014, Zurich, Switzerland.:Springer, 2014:740-755.
- [57] JOSHUA GOODMAN. Classes for fast maximum entropy training[C]. //2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2001), vol.I: Speech Processing 1. 2001:561-564.
- [58] LIN, TSUNG-YI, GOYAL, PRIYA, GIRSHICK, ROSS, et al. Focal Loss for Dense Object Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2):318-327.
- [59] Milletari F , Navab N , Ahmadi S A . V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation[C]// 2016 Fourth International Conference on 3D Vision (3DV). IEEE, 2016.
- [60] Smith L . Cyclical Learning Rates for Training Neural Networks[C]// 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2017.