

Research on segmentation method of infected regions in COVID-19CT images based on deep learning

by Xingfei Feng

Submission date: 07-Sep-2022 09:19AM (UTC+0100)

Submission ID: 185765931

File name: 32917_Xingfei_Feng_Research_on_segmentation_method_of_infected_regions_in_COVID-19CT_images_based_on_deep_learning_430_1001602030.docx (1.8M)

Word count: 14910

Character count: 81554



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Research on segmentation method of infected regions in COVID-19CT images based on
deep learning

by

Feng Xingfei

Supervisor: Terry Walcott

Project submitted as part of the requirements for the award of MSc in Software
Engineering

September 2022

DECLARATION

I, ...Feng Xingfei..... 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 :

Date : 2022.9.5.....

FORM OF CONSENT

I _____Feng Xingfei_____ hereby consent that my Project, submitted in candidature for the M.Sc. Applied Computing, 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 __________

Date _____2022.9.5_____

Where necessary replace M.Sc. Applied Computing with your named award.

Abstract

Patients with novel coronavirus pneumonia (COVID-19) have small slices of lung in the early stages and glassy structures as the disease progresses. Due to the high noise level of lung CT images and the small size of the pre-pneumonia area, it is difficult to identify the patients, and the use of traditional image segmentation methods can easily lead to misdiagnosis and missed diagnoses. In this paper, we focus on COVID-19 lesion segmentation of lung CT images based on convolutional neural networks. In this paper, we propose a U-Net network-based lung image segmentation algorithm. In the U-Net network, the Res Block replaces the original feature extraction layer to extract more image information; the problem of network degradation and poor segmentation of COVID-19 lesions in the U-Net model for lung image segmentation is solved, and the accuracy of the proposed method reaches 79.03% by testing the model. A U-Net network-based lung image segmentation algorithm is proposed. To address the problem of poor segmentation performance for the pneumonia region when segmenting the above parties, the model introduces a dense connection module on top of the traditional U-Net to enhance the information transfer between the feature maps of all layers; the spatial void pyramid structure retains more image features; and finally, an attentional supervision mechanism is used in learning the deeper feature information so as to extract the feature information of the lung elements more effectively. After testing on CC-CCII data, the accuracy of the proposed method, reached 79.53%, and the method can effectively make accurate segmentation of lung and pneumonia region elements.

Keywords—deep learning, CT image segmentation, U-Net, ASPP, ECA module

CONTENTS

CHAPTER 1 INTRODUCTION	1
CHAPTER 2 LITERATURE REVIEW	5
CHAPTER 3 EXPERIMENTAL DESIGN OF THE DEVELOPMENT	8
CHAPTER 3.1 NEUROS	8
CHAPTER 3.2 FUNCTION	10
CHAPTER 3.2 CONVOLUTIONAL NEURAL NETWORKS	14
CHAPTER 3.2 BASIC STRUCTURE	14
CHAPTER 3.3 ADVANTAGES OF CNN.....	19
CHAPTER 3.5 INTRODUCTION TO PYTORCH	20
CHAPTER 3.6 EXPRIMENTAL ENVIRONMENT	22
CHAPTER 4 RELATED WORK.....	23
CHAPTER 4.1 U-NET	24
CHAPTER 4.2 RESNET-50	26
CHAPTER 4.3 ATTENTION MECHAMNISMS	27
CHAPTER 4.4 ASPP	33
CHAPTER 5. DESIGN/DATA COLLECTION	37
CHAPTER 5.1 NETWORK STRUCTURE	37
CHAPTER 5.2 RES BLOCK.....	37
CHAPTER 5.3 EXPERIMENTAL DATASET	38
CHAPTER 5.4 EXPERIMENTAL ENVIRONMENT	38
CHAPTER 6. RESULTS AND DISCUSSION.....	40
CHAPTER 6.1 CONTRAST EXPRIMENT.....	40
CHAPTER 6.2 ABLATION EXPRIMENTS.....	42
CHAPTER 7. CONCLUSIONS AND FUTURE WORK	44
CHAPTER 7.1 CONCLUSIONS	44
CHAPTER 7.2 FUTURE WORK.....	45

CHAPTER 1 INTRODUCTION

Novel coronavirus pneumonia (corona virus disease 2019, COVID-19) has been breaking out since late 2019 and has since rapidly become a pandemic in the world. Patients infected with COVID-19 have inflamed air sacs in their lungs, causing pleural effusion, which can lead to breathing difficulties and other symptoms of influenza such as fever and cough. COVID-19 can affect multiple tissues and organs at the same time and can establish person-to-person transmission. COVID-19 has been spreading rapidly around the world and is sweeping through many countries. More than three years after COVID-19, the world's economies are still not fully recovered, and the post-COVID-19 world economy is nowhere near what it was before the outbreak, fuelling fears of the pandemic and a gradual loss of confidence in economic recovery.

In order to observe the actual situation of the infected area in detail, medical imaging techniques are used in medicine to aid diagnosis. The discovery of X-rays by the German physicist Wilhelm Röntgen in 1895 led to the development of medical imaging, which enriched medical diagnosis and enabled doctors to analyze areas of disease through medical images of the human body. Computed tomography CT (Computed Tomography), which uses X-rays to irradiate the body to get tomographic images of human organs and tissues, converts the complete three-dimensional information of human organs and tissues into multiple tomographic images for diagnostic studies by doctors. It is currently one of the most common clinical imaging tools for lung imaging. With the increasing number of patients diagnosed with pneumonia and the increasing amount of CT data from patients' lungs, it is essential to study how to segment and identify infected areas from CT images efficiently and accurately. However, manual labeling of infected areas is a tedious and repetitive task that requires a highly skilled and experienced clinician. Therefore, designing and constructing an automated segmentation model for infected areas of New Coronary Pneumonia can improve labeling efficiency and provide a reasonable basis for further analysis and diagnosis.

The critical factors in the severity of the COVID-19 epidemic are the rate of transmission of SARS-CoV-2 in the environment and the high rate of infection in close contact groups, with a secondary factor of lethality [1]. Compared to SARS-CoV and MERS-CoV, SARS-CoV-2 has the lowest global mortality rate of approximately 2.1%. However, the virus has a relatively high basic reproduction number and is more contagious when the virus is circulating during the incubation period or when patients are mildly ill [10]. The main routes of transmission are non-contact (respiratory droplets, aerosols) and contact (contaminated environment, close contact with patients, etc.). The clinical features of COVID-19 patients are similar to those of pneumonia caused by SARS-CoV and MERS-CoV. Pneumonia is mainly characterized by lower respiratory symptoms, including fever, dry cough, and wheezing. More than 60% of patients die after the disease has progressed to a severe or critical stage. Furthermore, according to epidemiological surveys, the incubation period of SARS-CoV-2 is long, ranging from approximately 1 to 14 days, with exceptional cases reaching more than 20 days [8]. In summary, SARS-CoV-2 is highly infectious, pathogenic, and has a long incubation period, making the rapid diagnosis and precise treatment of patients with COVID-19 quite tricky.

Currently, if a suspected patient presents with respiratory symptoms, fever, cough, or pneumonia, a positive real-time fluorescent quantitative polymerase chain reaction (RT-PCR) test is required to confirm the final diagnosis of a patient with COVID-19. However, recent studies have shown that the detection rate of RT-PCR is relatively low, at around 30-60%, and repeated testing is generally required, so a negative RT-PCR result does not entirely negate the possibility of SARS-CoV-2 infection. In addition, in resource-limited clinical settings, the time required to get RT-PCR results may be longer because of the need for rapid patient confirmation. In such cases, chest CT can be used for rapid triage. Compared to RT-PCR, computed Tomography (CT) is an effective tool for rapid screening for COVID-19, with a shorter clinical referral time, the ability to get detailed imaging information related to pathology to visualize better the extent of lung involvement and severity of infection, and the ability to measure and

analyze the area of the lesion, either manually or by computer. The imaging features of patients are usually seen on CT images, such as Ground-Glass Opacities (GGO), Pulmonary Consolidation (CL), and Pulmonary Fibrosis [2, 3]. Chest CT imaging is a standard diagnostic tool for pneumonia. High-resolution thin-section CT images are often used as the primary means of early imaging. They are indispensable in screening patients for COVID-19 and assessing their progress due to their convenience, accuracy, high positivity rate, and good reproducibility.

However, manually selecting CT images of COVID-19 patients from many CT images can be cumbersome and time-consuming, as the physician must locate the lesion area on a layer-by-layer slice of the chest CT image, which can take hours to accurately describe the lesion on a single chest CT image. Screening a large number of CT images is even more demanding for the imaging physician as the patient's condition changes rapidly over a short period, and multiple CT reviews are required to confirm progression. In addition, because COVID-19 has similar features to other community-acquired pneumonia (CP), such as viral, bacterial, or fungal chest CT images, they cannot be accurately distinguished by the imaging physician without other means of detection. Therefore, a reliable CT image-based computer-aided diagnosis (CAD) method or system would help to reduce physician workload and enhance the efficiency of COVID-19 detection and could play an essential role in clinical practice. However, developing such a system is a challenging task because the appearance, size, and location of focal areas of pneumonia on CT images vary considerably. Designing suitable methods to discriminate COVID-19 from CT images of pneumonia using only classical digital image processing techniques or traditional statistical learning methods that rely on the manual identification of COVID-19 CT images is challenging. [4].

Thanks to the development of electronic information technology and the advent of the era of big data. Artificial intelligence techniques represented by deep learning have brought significant changes to global technology and industry, providing efficient solutions to problems within many fields, such as computer vision. They have received widespread attention and research from scholars. The importance people place on

healthcare is increasing each year, which has led to the development of AI technologies for integration with healthcare. In recent years, deep learning neural networks, represented by convolutional neural networks (CNNs), have made great progress in many medical fields [5-7], stimulating the development of new diagnostic radiology techniques based on deep learning [8]. Based on the background of the development of deep learning in healthcare and the practical needs of chest CT screening for COVID-19 patients. It is worth exploring the introduction of deep learning techniques into the COVID-19 computer-aided diagnosis system, which is essential for fully utilizing the potential of CT images to help treat COVID-19 while at the same time allowing for more clinical research.

CHAPTER 2 LITERATURE REVIEW

Medical image segmentation plays an integral role in diagnosing and analyzing a condition, and the segmentation results will influence the subsequent diagnosis of the disease. Early lung CT image segmentation approaches utilized basic image processing methods, including image arithmetic and morphological processing. With the development of neural network technology, researchers have applied deep learning to medical image segmentation processing. Therefore, the development and research of medical image segmentation algorithms has focused on the improvement of deep learning based image segmentation algorithms [9].

In traditional medical image segmentation algorithms, images are segmented into different regions based on spatial structure information, gradient information, texture information [10], and greyscale information to extract the target region when the same region has the same semantic information. Researchers at home and abroad have conducted many studies on traditional segmentation methods. The main segmentation methods are threshold segmentation, region growing, edge-based methods, clustering, and watershed methods.

With the continuous development of computer hardware, deep learning has gradually achieved good results in various fields. Experts in image algorithms worldwide are progressively turning their goals to research deep learning algorithms. Still, with the improvement of scientific research and hardware environment, many excellent scientific results are also produced. Research on deep learning algorithms has been extended and has more scientific achievements. However, with the improvement of scientific research level and hardware environment, many excellent scientific achievements have been made. Image segmentation is the process of classifying all the pixels in the whole image. The image segmentation technology based on deep learning has already surpassed the traditional medical image segmentation algorithm in terms of segmentation accuracy and efficiency. Deep learning is based on medical image segmentation algorithms and includes Full Convolutional Neural Network (FCN). Encoder-Decoder is based U-Net, Hole Convolution based DeepLab[11], Target

Detection based Mask R-CNN [12], Feature Fusion based RefineNet [13], Adversarial Network GAN based methods.

In 2015 Long et al [14]. proposed the Full Convolutional Neural Network (FCN), which extends the application of deep learning to image classification to the pixel level, enabling pixel-to-pixel network training. The neural network's output is a probability map of the result of segmentation, and the FCN is the pioneering work of deep learning in image segmentation. The method replaces the fully-connected layer in a traditional convolutional neural network CNN with a convolutional layer, which allows the network to get a probability score of whether all input pixels are target pixel points in a single training session. The FCN network has a high segmentation efficiency in image segmentation because of the convolutional and deconvolutional layers.

The U-Net network is further improving the FCN network. The U-Net uses a symmetric model, with the left half being the downsampling paths [15]. The authors call this a contraction path and the right half an upsampling path, called an extension path, and finally use jump connections to fuse the feature maps between the symmetric paths. U-Net allows for the combination of low-level with high-level information. The U-Net network is the most widely used network for medical image segmentation thanks to the symmetric structure and jump connections. Abdulkadir et al. [16] proposed a 3D U-Net model, which uses 3D convolution to extract feature information between CT layers, solving the drawback that 2D convolution ignores spatial information and enhancing the segmentation accuracy of the model. The nnU-Net model proposed by researchers at the University of Heidelberg in 2018 attracted attention [17]. The model can adapt to various medical data and perform training with good results.

DeepLab extends the perceptual field of the model and extracts more features by introducing null convolution. There are three versions of Deep Lab; Chen proposed Deep Lab V1 [18], the first model to introduce null convolution into deep neural networks. It is expanding the perceptual field of the model without increasing the model parameters and adding a conditional random field (CRF) to the model's last layer to increase the model's ability to capture image details and achieve accurate pixel

localization. Later, Chen et al. introduced the ASSP module and proposed DeepLab V2, which used null convolution with different sampling rates to get information at different scales to achieve multi-scale target training. 2017 Chen et al. proposed Deep Lab V3 [19] based on the V1, and V2 versions, which expanded the sampling width of the ASSP module by serial and parallel null convolution to increase further the perceptual model field and more effective Refine Net. Feature fusion-based segmentation methods enhance the ability of images to learn global contextual information by fusing features sampled at different scales and levels. Lin [20] et al. proposed a multi-path refinement network RefineNet, which can fuse feature information at different levels to get accurate segmentation results and solve the information loss problem. The network first uses a CNN network to extract four feature maps of the original image at different resolutions and scales. It then uses four refinement modules to process each feature map. The refinement module is a fusion of features based on the connection of multiple residual modules, which is a long-linked module that can efficiently transfer the gradient to the whole network. Fusing the low-level delicate semantic information with the high-level coarse semantic information, thus preserving the feature information lost during the downsampling process and generating a high-resolution segmentation map. The number of scales of input to the RefineNet segmentation network can be varied flexibly so that the network structure can be changed in many ways. The network captures feature information at different levels, effectively improving segmentation accuracy.

CHAPTER 3 EXPERIMENTAL DESIGN OF THE DEVELOPMENT

CHAPTER 3.1 NEUROS

As shown in Figure 3-1, the nerve cell serves as the basic building block of the biological nervous system and is also often referred to as a biological neuron[21]. It consists of three parts: the cell body, axons, and dendrites, as shown in the diagram. It mainly uses synapses to receive signals from other neurons. Only when the signal received is more significant than a certain threshold does the neuron become activated and then sends a signal outwards, which travels along the axon to other connected neurons, which can receive the signal and have the opportunity to be activated by it.

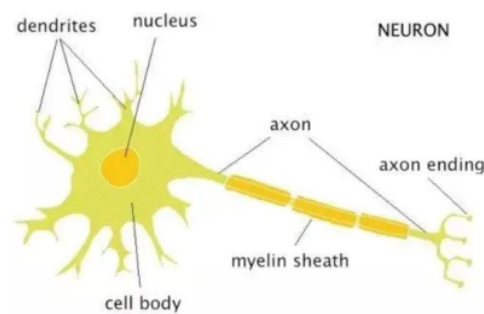


Fig.3-1 nerve cell

Artificial Neural Networks (ANNs) are usually composed of many interconnected and powerful functional elements that are used to simulate the nervous system of the animal brain and to replicate the behavior and processes of the actual animal brain [22]. ANNs can be abstractly thought of as weighted directed graphs with artificial neurons as nodes. In this diagram, the artificial neuron is an abstraction and imitation of the biological neuron; the information transfer function of each neuron is abstracted and imitated. The weights can be thought of as the strength of the influence of the interconnected neurons on each other, which is an abstraction and imitation of the number of neurotransmitters transmitted between biological neurons. The structure of an artificial neuron is shown in Figure 3-2.

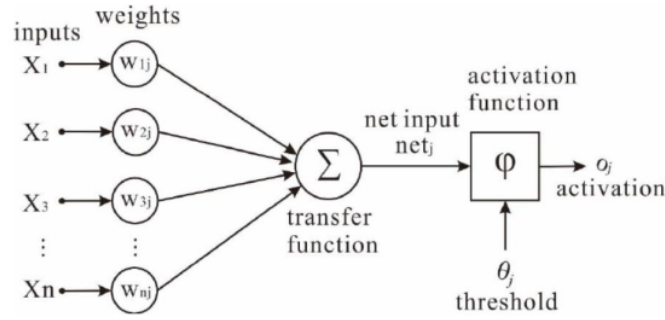


Fig.3-2. Structure image of an artificial neuron

A given neural network unit receives inputs from multiple other neural units to produce an output, which is a simulation of the behavioral process in biological neurons where nerve endings sense external stimuli to produce electrical signals. If we let the input information obtained from other neurons be $X(X = x_0, x_1, x_2, \dots, x_n)$. The output information from the neuron is Y , and the magnitude of the other neurons' influence on each other is $W(W = w_0, w_1, \dots, w_n)$, bias $B(B = b_0, b_1, \dots, b_n)$. Whether a neuron is activated can be determined using the mathematical function $g(z)$ calculation, then the input to the neuron can be expressed as $z(x)$ in equation (3-1), and the output can be seen in equation (3-2):

$$z(x) = \sum_{i=0}^n (w_i x_i + b_i) \quad (3-1)$$

$$z(x) = \sum_{i=0}^n (w_i x_i + b_i) \quad (3-1)$$

ANN can map arbitrary and complex non-linear relationships and is robust, memorable, and learning. It abstracts and mimics the animal brain at two primary levels: the first level is that the information extracted by the network is obtained by learning from external data; the second level is that the size of the interactions between neurons within the network, i.e., the interconnection weights, is used to preserve the learned knowledge. Therefore, the critical task in ANN simulation learning is to build a model to determine the connection weights. Connection weights can be positive or negative, with positive weights meaning that the input signal is reinforced and negative weights meaning that the input signal is suppressed. The numerical value of the weights

indicates the strength of the input signal on the neuron, with higher weights meaning that the input signal has a more significant impact on the neuron. Therefore, it is possible to vary the individual weights to achieve the desired output with the same input information. However, ANNs are usually composed of many neurons, and it would be highly time-consuming and almost impossible to calculate these connection weights manually. In such cases, algorithms are needed to calculate and modify the weights automatically. This series of operations automatically updating the weights is often referred to as learning or training.

The activation function is a class of functions that acts on the neurons in an ANN. Its primary function is to map the relationship between the input neurons so that the output is non-linear, making it a critical role in a neural network. The application of activation functions to ANN models can substantially enhance the model's ability to represent non-linear relations, which can also enable the network model to be infinitely close to any non-linear function. Thus it enables the neural network to deal with numerous non-linear problems in realistic scenarios. In addition, it provides good help in improving model robustness, alleviating gradient disappearance, and speeding up model convergence. Nowadays, there is a wide variety of activation functions with different characteristics; the commonly used ones are the Sigmoid, Tanh, and ReLU.

CHAPTER 3.2 FUNCTION

The Sigmoid function is one of the most common S-functions in biology[23]. Due to its own and its inverse functions having one-way incremental characteristics, it is usually used as an activation function in the information discipline. However, its shortcomings are more pronounced, generally not in the hidden layer but the last layer of the neural network model of the dichotomous classification problem; there are two main reasons: First, the Sigmoid function maps the input to between (0,1), the function about (0,0.5) symmetric, with a mean output value of 0. Once the number of layers of the neural network increases. It is not easy to calculate the last layers; secondly, when the value of x is very large or very small, the derivative of the function is close to 0, resulting in the gradient of the weights W and b will also be close to 0, making it

difficult to update the gradient and the problem of gradient disappearance. As shown in Equation (3-3), the Sigmoid is shown in the graph in Figure 3-3.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3 - 3)$$

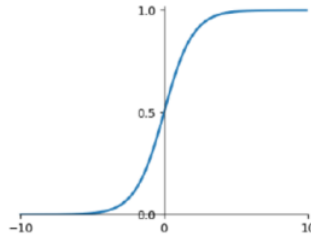


Fig.3-3 Sigmoid

Tanh is a modified version of Sigmoid, which is shown in the diagram. The former maps input values of arbitrary size onto the interval $(-1,1)$, and the output value is a centrosymmetric image about the origin of the axis and has a mean of zero. Although it compensates for the latter's output mean of 0.5, it still does not overcome vanishing gradients' shortcomings. The mathematical expression of the Tanh function is given in Equation (3-4), and the graph is shown in Figure 3-4.

The ReLU function is a segmentation function, one of the most commonly used functions in building neural network models 37. Due to its particular structure, only linear relations exist in the function. Compared to other exponential functions, it is less computationally intensive, faster, and overcomes the problem of partial gradient disappearance. When the input values are all greater than zero, the derivative of the ReLU function is one, allowing gradient-based learning, there is no gradient disappearance problem, and the ReLU function performs better. The ReLU is sensitive to negativity. When the input is negative, the function's derivative is zero, there is a gradient disappearance problem, the weights cannot be updated, the learning speed is slower, and even the neurons fail directly. The training remains silent for the rest of the iterations. The ReLU formula is shown in (3-5), and the shape diagram is shown in Figure 5.

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (3 - 5)$$

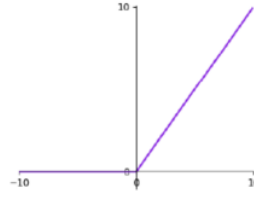


Fig.3-3. ReLU

In Machine Learning (ML) [24], all ML and DL algorithms are optimized by minimizing the objective function. This minimization function is referred to as the loss function (LF), which is crucial in optimizing all types of neural networks. It evaluates the performance of a model by quantifying the error between the predicted and actual values, and the smaller the LF, the better the model fits.

More importantly, neural network models are generally trained to determine the optimal weight parameters by minimizing the loss function, which is usually implemented by a gradient descent algorithm. In addition, the loss function provides a static representation of the running state of the model but also allows the algorithm to be judged on how well it fits the data set based on the variation in the loss value. There are many different types of loss functions, the main ones being Mean Squared Error Loss (MSE)[25], Cross Entropy Loss (CE), and Triplet Loss (TL). Generally, in most convolutional neural network models for classification and regression problems, the cross entropy loss function is the most commonly used loss function [26]. MSE as formula(3-6):

$$J_{MSE} = \frac{1}{2M} \sum_{i=1}^M (y_i - \hat{y}_i)^2 \quad (3-6)$$

MSE Loss is the mean of the sum of the squares of the errors of all training samples and is a commonly used LF in ML and DL regression tasks, sometimes referred to as L2 Loss, which takes the minimum value of 0 when all sample predictions are equal to the labeled value.

A significant step in solving real-world problems using ML and DL techniques is to feed a large amount of data to a pre-designed model and let the model learn from the data autonomously. The model's parameters are then continuously optimized so that the model can eventually match the learning task to a high degree with a suitable set of

parameters. The model will eventually match the learning task with a specific set of suitable parameters to a high degree. This "autonomous learning" step is the key to ML and DL algorithms. The standard methods for implementing this "autonomous learning" step in machine learning and deep learning are least squares, Newton's method, and gradient descent. Among these methods, gradient descent (GD) is the most classical and straightforward and is widely used in deep learning (convolutional neural networks) [27]. It is similar to the Newtonian/Newtonian method in that the optimal setting of the model parameters is solved by iteration.

In contrast, the least squares method is solved by computing a large inverse matrix. The inverse matrix is very computationally intensive. If the data size is small and an analytical solution exists, least squares are faster. They have an advantage over GD, but the advantage of iterative solving increases as the data size increases. The former uses an iterative gradient solution, while the latter uses the inverse or pseudo-inverse of a second-order Hessian matrix, with the former taking less time per iteration and being preferred over the other.

It is worth noting that there are three different forms of GD, namely Batch Gradient Descent (BGD), Stochastic Gradient Descent (SGD), and Mini-batch Gradient Descent (MBGD). BGD is the most commonly used form of GD, which uses the entire training number for each parameter update, and the training speed becomes very slow as the size of the training set increases. MBGD combines the advantages of the previous two algorithms by updating the parameters one sample at a time, with the number of samples (Batch-Size), which speeds up training and makes the training results more secure.

In addition, when using the gradient descent algorithm, attention needs to be paid to the selection of the learning rate, the setting of the initial values of the parameters, and normalization, all of which can impact the training. This speeds up the training and makes the training results more secure. In addition, when using the gradient descent algorithm, attention needs to be paid to the selection of the learning rate, the setting of the initial values of the parameters, and normalization, all of which can impact the

training.

CHAPTER 3.2 CONVOLUTIONAL NEURAL NETWORKS

In recent years, Convolutional Neural Networks (CNN) has proliferated as a convenient and competent class of algorithms for setting categories in DL and has attracted widespread interest from the community. It originated from the cat cerebral cortex vision research in the 1960s, which found that its unique network architecture could effectively reduce the complexity of Feedback Neural Network (FNN), and then proposed CNN. CNNs are widely used in AI robotics, Industry 4.0, automation, value budgeting, biochemistry, healthcare, smart cities, agricultural informatics, and other fields with great success. In particular, image recognition is widespread among scientific researchers due to its unique organization, which reduces the model's complexity and the number of parameters, and speeds up the model learning process. In dimensionality, there are 1D CNNs, 2D CNNs, and 3D CNNs. 1D CNNs are less commonly used and are generally used for serial data; 2D CNNs and 3D CNNs are more commonly used, with the former being used for data with two-dimensional characteristics, such as grey-scale image recognition and text recognition, and the latter being used for color images, medical images, and video data.

Convolutional neural networks have developed rapidly, and the commonly used shallow-level models are LeNet [28] and AlexNet [29]. Le Net was the first convolutional neural network to be successfully applied. Alex Net has a similar structure to Le Net, except that it uses cascading convolutional layers to extract features and increases the depth of the network. The GoogleNet series [30] uses the Inception module to reduce the network parameters to enhance the speed of network training. The Res Net and Dense Net series introduce spanning connections, with the former being interlayer connections and the latter connecting. Both increase the network's complexity to a certain extent but have a more excellent capability in model feature extraction.

CHAPTER 3.2 BASIC STRUCTURE

Typically, neural networks are organized in three parts: the input layer, the hidden layer and the output layer. The layers in the network are closely connected, with the

upper layer serving the lower layer, i.e., the output of the upper layer's output serves as the lower layer's input, with corresponding weight parameters between adjacent layers and the output being our classification category. Similarly, CNNs consist of an input layer, a convolutional layer, a pooling layer, a fully connected layer and an output layer. In a CNN, the convolutional and pooling layers are usually irregularly joined together to form its hidden layer, which does not necessarily need to be paired with the pooling layer. However, they can also be pooled only once after several successive convolutions.

The most critical part of a CNN is the convolutional layer, which serves to re-extract features from the output of the upper layer [31]. The convolutional layer uses several internal convolutional kernels to perform convolutional operations on the input data to extract features, and passes the extracted results to the next layer. This means that the output data from the previous convolutional layer becomes the input data for the next convolutional layer. The convolution operation is a matrix multiplication of the local data matrix of the input feature map with the filter weight parameter matrix, which is a linear operation that adds a non-linear relationship. The convolution operation is usually followed by activation using an activation function. Specifically, the filter multiplies the local data in the window from high to low, left to right. According to a given step and window size, the filter parameters at the corresponding position accumulate them so that a new feature image is created when the input feature map is scanned. The convolution operation is shown in figure 3-6.

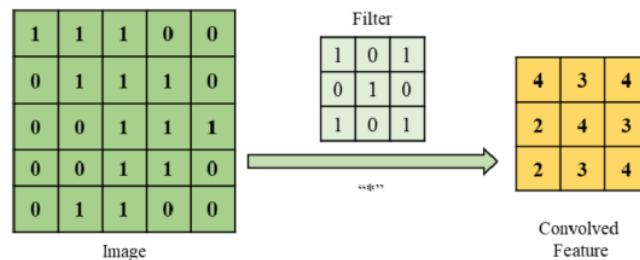


Fig.3-6. Convolution

In particular, it is worth noting that the filter depth is closely related to the depth of the input feature map in the convolution operation, and the two must be equal. The size

of the scan window must be equal to the size of the filter; the depth of the output feature image is equal to the number of filters; the width and height of the output feature image are related to the size of the filter, the step size of the scan, and the Padding. The following is an example of the RGB image, the filter; the output feature map is described, let the input image height, width, depth is $n_H^{L-1}, n_W^{L-1}, n_C^{L-1}$; scan step is s ; filter size is (f, f) ; the number of filters is n_C^L ; Padding is p , then the input graphics as in equation (3-7), the filter as in equation (3-8), the output graph shape as in equation (3-9).

$$\text{input image: } (n_H^{L-1}, n_W^{L-1}, n_C^{L-1}) \quad (3-7)$$

$$\text{filter: } (f, f, n_C^{L-1}) \quad (3-8)$$

$$\text{output image: } \left(n_H^L = \left\lfloor \frac{n_H^{L-1} + 2p - f}{s} \right\rfloor, n_W^L = \left\lfloor \frac{n_W^{L-1} + 2p - f}{s} \right\rfloor, n_C^L \right) \quad (3-9)$$

Pooling layers are also known as sampling layers and are usually added periodically after the convolution layer. The crucial role of the pooling layer is to re-filter the input data from the upper layers to reduce the size of the feature space and the computation of the model parameters. The aim is to reduce the computational effort of the model without compromising image quality. However, this approach discards some of the information and reduces the model's feature extraction capability. Generally, the pooling area is often chosen to be 2×2 in size, and the information is filtered according to established rules, resulting in the filter being used as a value for the output feature map. Maximum pooling is most commonly used, which takes the maximum value within the pooled region as the filtering result, as shown in Figure 3-7 below.

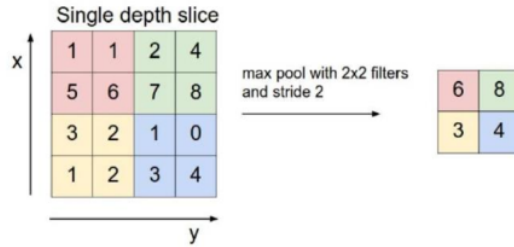
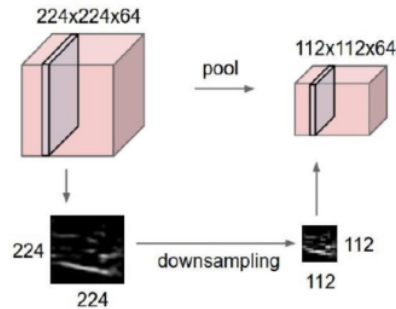


Fig.3-7. Maximum pooling

According to the example of maximum pooling in Figure 3-8 and the image size

before and after pooling, the pooling operation retains the maximum value in each small fixed region, equivalent to extracting the best matching information in this region. Compared to the input feature map, the pooling reduces the dimensionality of the output feature map by discarding some other information[32], which reduces the features' dimensionality and increases the CNN model's perceptual field. The pooling operation does not focus on exactly where in the region there is a match but only on whether there is a match in the region. The sensitivity of the convolutional neural network model to feature information in specific directions and specific locations can be weakened, allowing the model to focus more on the information itself, with some degree of freedom. In addition, this feature invariance of the pooling operation also provides more ways to expand the data. For example, fewer data can be expanded using rotation, scaling, panning, etc. The probability of less model overfitting will not impact the model feature information extraction.



The fully-connected layer in a CNN is constructed as a hidden layer of a standard feedforward neural network, and its central role is to act as a classifier. From the perspective of representation learning, it integrates the features extracted from convolution and pooling layers nonlinearly. The national learning perspective performs a non-linear fusion of features extracted from the convolutional and pooling layers. In terms of dimensionality, before entering the fully-connected layer, the data is spread into a one-dimensional feature vector, which loses the spatial structure properties of the feature map. Therefore a fully connected layer is not generally expected to have the ability to extract feature information but is usually utilized to achieve the classification

of existing extracted higher-order features to fulfill the learning objective. The fully connected layer is also not required and can be removed and replaced with other classifiers (support vector machines, random forests), which can also give good results.

During forward propagation, conventional neural networks passed the error between predicted and labeled values back to each neuron by backpropagation. The network can correct and update each neuron parameter according to the error. Dropout, on the other hand, stops specific intermediate hidden neurons randomly during forwarding propagation and later back propagates by modifying the error obtained by the network while resuming before a training round all neurons and re-inactivating some of them at random.

In neural network model training, if the neuron parameters are significant and the training sample is small, the model training will be overfitted. This is because the model learns local features. By randomly ignoring some of the hidden neurons and reducing the interaction between neurons, the Dropout layer can enhance the model's generalization ability and reduce reliance on certain local features [33]. Reducing some model parameters can also speed up the model's training. The figure below illustrates the process of randomly deactivating neurons in the Dropout layer. Figure 3-8 shows a standard neural network model; Figure 3-9 shows a schematic of a model with a Dropout layer, which only works during training and does not take effect when testing the model. The random neuron deactivation ratio is usually set to 0.3-0.5, and different models with different functions must set the appropriate ratio considering different environments and the amount of training data.

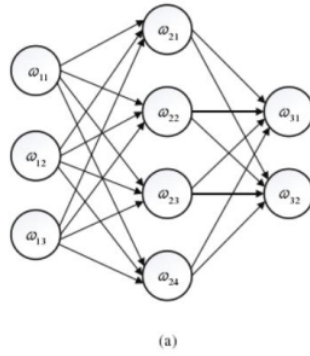


Fig.3-8. Standard neural network

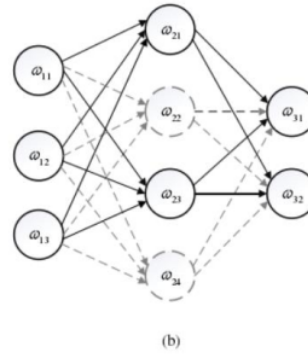


Fig.3-9. Dropout

Neural networks train networks through losses and, in essence, need to enhance network performance by calculating losses and reducing them. The role of the optimizer is to provide an optimization method for the model and backpropagate to update the network parameters. Commonly used optimizers include stochastic gradient descent, momentum, RMSProp algorithm, and Adagrad adaptive gradient method. The Adaptive Moment Estimation (Adm) algorithm is a more effective algorithm, an optimization method that adaptively adjusts the learning rate. The accumulation of gradients is performed through momentum calculations. The adaptive adjustment of the learning rate is also based on the first-order moment estimation and second-order moment estimation of the gradient to the learning rate.

CHAPTER 3.3 ADVANTAGES OF CNN

The data cannot be fed directly to the model in traditional machine learning classification problems. Some pre-processing operations on the data are usually required, such as edge detection and threshold detection. These operations aim to reduce noise in the data and highlight valuable features such as edges and textures in the data for the machine learning model to learn. In contrast, ordinary and convolutional neural networks do not require these processes and avoid these tedious operations of enhancing features. Only a simple format conversion and normalization of the data are required to feed into the network and get a relatively good classification result.

When using machine learning algorithms, the first step is to manually engineer the features, i.e., to determine which will be used as training Features, so much effort is

wasted on engineering the features before the model achieves a satisfactory result. Not only is this method less automated and more efficient, but the manually selected features may not always be valid. If the number of features is small, underfitting may occur, and the model may not accurately identify the target. However, if the number of features is large, the model will focus too much on certain features, and overfitting will affect the model's generalization. Neural network-based methods (ANN, CNN) are highly automated, efficient in feature extraction, do not require extensive feature engineering, and do not require the advanced design of feature content or the number of features. Feed the data to the model, let it train itself, self-correct and automatically extract the required features.

In the image domain, ANN models are generally unsuitable for feature extraction. We know that images are made up of pixel points and generally have three channels representing each of the three RGB colors. If an RGB image of size (1024,1024,3) is input and the ANN model has only one hidden layer with 500 nodes, the network model will have at least $1024 \times 1024 \times 3 \times 500$, or 1.57 billion parameters. Even in this modern age of computer processing power, the computational effort for a random backward derivation is enormous. Moreover, with such a considerable number of parameters, overfitting is very likely if there is not enough data to support the training. However, CNN solves this problem by switching from ANN full connectivity to focusing on local properties. It provides a powerful tool for the image field by continuously compressing the feature dimension through convolutional and pooling layers, reducing the model parameters and complexity to a large extent.

CHAPTER 3.5 INTRODUCTION TO PYTORCH

As research in DL has intensified, various DL frameworks have emerged in recent years. A deep learning framework is a tool for deep learning development, consisting of resources such as libraries and modularly packaged pre-trained models. Using Pytorch, the complex process of developing deep learning models can be greatly simplified, allowing for rapid model building, data entry, model training, model tuning, model testing, and model deployment. Pytorch is now a must-have tool for AI developers. The

mainstream DL frameworks include Tensor Flow, Keras, and Pytorch. The Pytorch framework, a Python version of Torch, was launched by Facebook's AI Institute in January 2017 and is now widely used in academia and industry. With the addition of the Caffe2 project to Pytorch, Tensor Flow's leadership in deep learning application frameworks is now under serious threat.

In the academic world, Pytorch is popular with researchers and has a significant advantage. For several reasons, most models published in deep learning-related articles are based on the Pytorch framework.

(1) Simplicity: Pytorch pursues less encapsulation, minimizes the duplication of rounds, and is designed according to the concept of three levels of abstraction from low to high (tensor $\rightarrow$ variables $\rightarrow$ neural network modules), allowing simultaneous modification and manipulation of operations. Pytorch is more intuitive and easier to understand the underlying code than Tensor Flow, which requires a lot of complex variables to be defined.

(2) Dynamic computation: This is the most distinctive feature that sets Pytorch apart from other frameworks such as Tensor Flow, allowing us to dynamically design neural networks and see the Tensor values at any time during training. Pytorch's Auto-grad architecture allows for automatic backward derivation, which changes the learning behavior of the neural network with zero latency.

(3) Simple parallel computing: Pytorch integrates a parallel data library, a model parallel library, and tensor-based GPU-accelerated parallel computing.

(4) Active community: Facebook, Nvidia, Twitter, and other internationally renowned companies are using the Pytorch framework; Pytorch is updated and maintained promptly, has a complete set of technical documentation, and the developers themselves maintain the forum, which provides a good user experience.

(5) A growing audience: In recent years, Pytorch has become the mainstream deep learning framework in the academic community, as more and more new papers have been published using the Pytorch framework as an implementation tool.

However, Pytorch has its shortcomings, mainly in model deployment, but this is being

remedied as the technology improves.

CHAPTER 3.6 EXPERIMENTAL ENVIRONMENT

During the network training process, the parameters are continuously adjusted to meet the demands of the network in order to avoid the gradient drop or gradient rage phenomenon that occurs during the network training process. The parameters of the experimental environment for network model training are shown in Table 1.

Table 1. Experimental environment

Name	Description	Parameter	Description
Processor	Intel 12700KF	Batch size	2
Graphics card	NVIDIA GTX3070ti	Initial learning	0.001
Video memory	8GB	Iteration count	150epoches
Running memory	32GB	Dropout	0.5

CHAPTER 4 RELATED WORK

Segmentation of the infected region of the new coronary in lung CT images is essential for further diagnosing and treating patients with new coronary pneumonia. When using traditional image segmentation algorithms to segment the infected region, different initial parameters set in the algorithm can affect the segmentation effect of the infected region due to the reliance on the subjective human experience of the infected region of the lung. Moreover, the traditional segmentation algorithm is based on the low-dimensional hierarchical features of the image, which will affect the accuracy of the segmented region to a certain extent. Applying deep learning to computer vision analysis has been a significant point of exploration for researchers. From the beginning of the LeNet-5 network, through AlexNet, VGG networks, and the evolution of the Dense Net network, convolutional neural networks have been used extensively in the field of image processing, and researchers have continued to enhance the network structure to achieve higher performance. In medical image segmentation, the U-Net network is a small sample data training network widely studied and used due to its strong generalization ability and good segmentation capability. The area occupied by the infected region of the new coronary pneumonia in the CT image is small. Suppose only the U-Net network is used for segmentation. In that case, problems such as incomplete capture of feature information or even gradient disappearance will occur, leading to low segmentation accuracy.

Therefore, this chapter improves the deep segmentation model U-Net and designs an end-to-end deep neural segmentation network to address its inadequate ability to capture the feature information of the infected area of the new crown. The network effectively combines different deep U-Net decoders. It uses a full-scale skip connection to aggregate high-level and low-level semantics in different scale feature maps on the decoder sub-network. In addition, an attention mechanism is introduced [34], which enables the network to extract more essential features, suppressing useless features, achieving more accurate feature extraction, and improving network segmentation performance. For the possible overfitting caused by the increase of network parameters,

a Dropout layer is added to remove some hidden neurons to train the network randomly.

In contrast, for the possible gradient disappearance problem, a batch normalization layer is introduced to make the network more stable, and this network is called the AMSU-Net model. In this chapter, the U-Net model, the MSU-Net model, the improved attention mechanism module, and the AMSU-Net network model designed in this chapter will be introduced. Finally, the segmentation process and experimental validation of the newly coronally infected lung region will be detailed.

CHAPTER 4.1 U-NET

The U-Net network is an end-to-end network based on a fully convolutional network and is commonly used in medical image segmentation tasks. The network adopts an encoder-decoder architecture, and the whole is divided into an upsampling process and a downsampling process. The downsampling process uses a CNN classification network with the fully connected layer removed, a feature extraction process. The network first receives the original image to be segmented and, after convolution, extracts the high-level features of the image to get a low-resolution feature map. The up-sampling process is a process of decoding and gradually restoring the original image information features, which uses the up-sampling method of deconvolution to recover the image's pixel positioning and classification information. The network uses the same number of convolutional layers for both down-sampling and up-sampling so that the network retains more of the original features in the decoding process, making the pixel localization and classification more accurate and the segmentation accuracy higher.

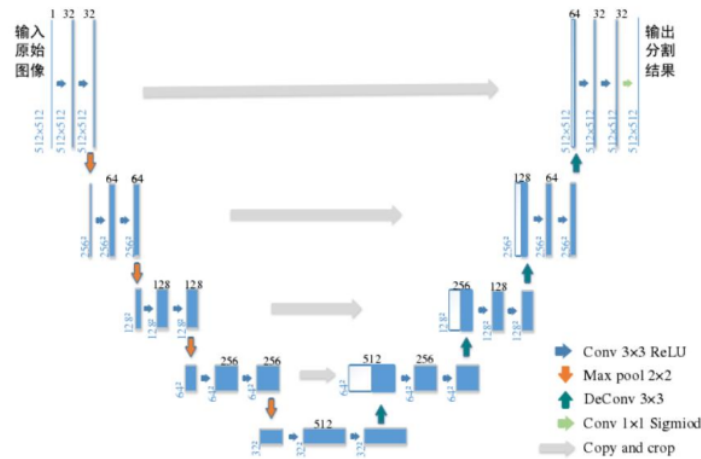


Fig.4-1. U-Net

As shown in Figure 4-1, the left half of the image shows the downsampling process of the encoding, first through two 3×3 convolutional layer sets (32 convolutional kernels, ReLU activation function) and then through a 2×2 Max Pooling layer with a step size of 2. The process is repeated four times. The number of channels is doubled with each down-sampling. The result is a 512-channel feature map with 32×32 resolution. The right half shows the up-sampling process of decoding, where the number of channels is firstly halved by up-sampling through deconvolution. Then more channels are constructed by combining the down-sampled features with the deconvolved ones through a jump join. Finally, the number of channels is halved by two convolutions, and the number of channels is repeatedly halved with each up-sampling, resulting in a 32-channel feature map with a resolution of 512×512. U-Net has a small number of parameters, a simple model, and a high generalization capability. However, after multiple downsampling, the original features of the image will be lost, which will affect the accuracy of the image pixel localization and the segmentation effect of the new crown-infected area, and will not achieve the segmentation accuracy required by professionals.

However, COVID-19 CT image segmentation differs from other tasks such as cell segmentation and blood vessel segmentation, and COVID-19 CT images are relatively much more complex. Firstly, Unet has only two convolutional layers in each module as

well as a maximum pooling layer to extract features, which cannot effectively extract all the useful information and may lead to information loss. In addition, the modules in the decoding stage of Unet are also performed with only two layers of convolution and deconvolution, which can lead to a certain degree of feature information loss, and Unet does not take into account the differences in features at different spatial locations such as global and local. To enhance the accuracy of neural segmentation of infected regions in CT images of COVID-19, Unet will be improved in this paper.

CHAPTER 4.2 RESNET-50

Many subsequent articles have proposed methods that build on ResNet50 or ResNet101. The methods proposed since then have been based on ResNet50 or ResNet101. It has been widely used in image classification, image segmentation, and target detection. It uses a connection method called shortcut connection, and the structure of the residual structure network is shown in Figure 4-2.

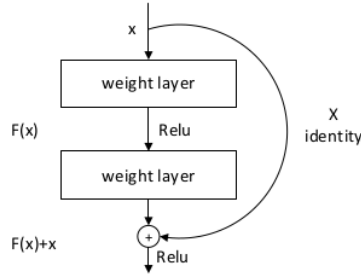


Fig.4-2. Resident structure

It takes (x) the input from the initial layer and (x) the output obtained after many layers to form a residual function. This type of residual function performs better in combining global and local features. It is easy to make the neural network layers much deeper, as in equation (4-1), where σ represents the non-linear function ReLU.

$$F(x) = W_2 \delta(W_1) \quad (4-1)$$

Then by a shortcut, and a 2nd ReLU, the output y is obtained as in equation (4-2).

$$y = F(x, \{W_i\}) + x \quad (4-2)$$

When the input and output dimensions need to be varied (changing the number of channels), a linear transformation Ws can be made to x at shortcut to get the output as in

equation (4-3).

$$y = F(x, \{W_i\}) + W_s x \quad (4 - 3)$$

CHAPTER 4.3 ATTENTION MECHANISMS

Attention mechanisms play an essential role in the human brain system, allowing us to focus more on what we find helpful when processing large amounts of information. In deep learning, the attention mechanism is considered a feature reselection mechanism, which can be understood as assigning higher weights to essential features. It has been helpful in natural language processing tasks, recommendation systems, computer vision, and speech recognition. Attention mechanisms can be divided into self-attention[35] and soft attention mechanisms. Soft attention is widely used in computer vision, where it focuses more on the region or channel information of the feature map. It is a deterministic attention mechanism that learns its internal weights through the training mechanism of deep learning. Examples include spatial domain attention, channel domain attention, and hybrid domain attention. Spatial Transformer Network (STN)[36, 37] uses spatial attention mechanisms to eliminate the effects of target affine transformations in images. The channel domain attention mechanism and the hybrid domain attention mechanism are used in this paper, and their principles and some implementations are presented.

In contrast to the spatial attention mechanism, the channel attention mechanism allows the convolutional neural network to focus on what the image is supposed to represent. In order to select the features that are more effective for the task. In layman's terms, channel attention is an optimization strategy that allows a network model to focus on the more critical information in an image. For a complex image, where there may be multiple target objects and complex background information, extracting more discriminative features from the image can help the network model to perform subsequent tasks.

The SENet (Squeeze-and-Excitation Network) proposed in the literature [38] was the winner of ImageNet 2017 for its effective use of the channel attention mechanism. In SENet, a Squeeze-Excitation (SE) module is designed, and the whole structure is shown

in 4-3.

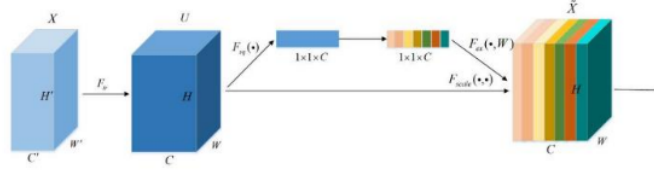


Fig.4-3. Squeeze-Excitation

The front part of the SE module, F_{tr} represents the feature map $X \in R^{x^1 \times H^1 \times C^1}$ obtained by the multi-layer convolution of the original feature map $U \in R^{W \times H \times C}$ by the pre-convolutional neural network structure. The Squeeze operation is a global average pooling operation of the input feature map U . The global average pooling operation calculates the distribution of the number of channels in each feature map layer. It is used to represent the global information of the feature map. The Squeeze operation is performed as in Equation (4-4).

$$z_c = F_{sq}(u_c) = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H u_c(i, j) \quad (4-4)$$

Where W denotes the width of the feature map, H denotes the height of the feature map, and $u_c(i, j)$ denotes the feature values in feature map channel c with coordinates (i, j) [39]. Moreover, z_c denotes the average pooling result within the feature map channel c . The channel weight vector $z \in R^{1 \times 1 \times C}$ is obtained by pooling c , and z can effectively represent the global information of the feature map. Excitation operation is used to get the strong correlation between channels using the loss function using the backpropagation process of deep learning. It consists of two fully-connected layers, the first having the number of neurons C/r . C represents the number of channels, and r represents the scaling factor, which effectively reduces the number of channels and hence the computational effort of the model. The activation function used in this layer is the ReLU function. The output feature vector of the first fully connected layer is $W_1 z \in R^{(1 \times 1 \times C/r)}$. The next step is to go through the second fully connected layer, which has C neurons and uses the Sigmoid function as its activation function. The feature vector $s = R^{(1 \times 1 \times C)}$ generated by the second fully connected layer is used to describe

the weights of each channel in the feature map. The Excitation procedure is given in Equations (4-5) as follows.

$$s = F_{ex}(z, W) = \delta(g(z, W)) = \delta(W_2 \delta(W_1 z)) \quad (4-5)$$

The principle of Excitation is to use multiple fully connected layers to further adjust the weight indices of each channel for a global average pooling process without inverse computation and training. In the end, the model is trained to get the channel enhancement and weakening patterns based on the entire data set. The SE module uses a channel-by-channel multiplication method to enhance and suppress each channel, as shown in Equations (4-5) below.

$$s = F_{ex}(z, W) = \delta(g(z, W)) = \delta(W_2 \delta(W_1 z)) \quad (4-5)$$

Where δ denotes the sigmoid activation function, W_1 denotes the trainable parameters of the first fully connected layer, and W_2 denotes the trainable parameters of the second fully connected layer. The principle of Excitation is to use multiple fully connected layers to further adjust the weight index of each channel for a global average pooling process without back calculation and training. In the end, the model is trained to get the channel enhancement and weakening patterns based on the entire data set. The SE module uses a channel-by-channel multiplication method to enhance and suppress each channel, as shown in Equations (4-6).

$$\widetilde{X}_C = F_{scale}(u_c, s_c) = s_c \cdot u_c \quad (4-6)$$

Where $\widetilde{X}_C$ denotes the feature map carrying channel attention, s_c denotes the weight of channel c , and u_c denotes the feature matrix with channel number c in the special diagnosis map U . The literature [40] was able to increase the accuracy of effective scene classification by adding the SE module to the ResNet model. The effectiveness of channel domain attention was also demonstrated.

A hybrid domain attention mechanism is a fusion of multiple attention strategies. A typical implementation is to implement both a channel-domain attention mechanism and a spatial-domain attention mechanism in the attention module and fuse them. In the field of digital image processing, engineers have designed several hybrid attention mechanisms, such as the Residual Attention Network [41], the Convolutional Block

Attention Module, CBAM [42], and the Dual Attention Network, DANet [43]. In principle, they all combine channel-domain and spatial-domain attention mechanisms, but there are differences in the combination of the different approaches. In the case of CBAM, for example, the convolutional block attention module obtains the appropriate attention in both the channel and spatial domains. When the attention module is embedded in a mainstream network, it can enhance the network model's ability to extract features without significantly increasing the overall computational effort and number of parameters of the model. When embedded in a mainstream network, CBAM can effectively enhance the feature extraction capability of the network model without significantly increasing the computational effort and number of parameters [44]. As shown in Figure 4-4, CBAM consists of a combination of two modules, the two channel attention module and the spatial attention module.

Assuming a given feature map $F = R^{(C \times H \times W)}$ as input, after processing by the Channel Attention Module, a one-dimensional vector $F_c(F) \in R^{(C \times 1 \times 1)}$ is generated to represent the attention of the channel domain. The resulting $M_c(F)$ is multiplied with the original feature map F by elements of the channel to generate the channel optimization feature map F' , which is then processed by the Spatial Attention Module to generate a vector to represent the spatial attention. Module to generate a two-dimensional matrix $M_s(F) = R^{(C \times 1 \times 1)}$ used to represent spatial attention. $M_s(F)$ will also be multiplied element by element with F' , and the final optimization result will be generated. The entire calculation process is as follows.

$$\begin{aligned} F' &= M_c(F) \otimes F \\ F'' &= M_s(F') \otimes F' \end{aligned} \quad (4-7)$$

As seen from Equation (4-7), CBAM (4-4) is designed to focus on more important features and suppress unnecessary features by sequentially processing the input features through its channel and spatial attention modules [44].

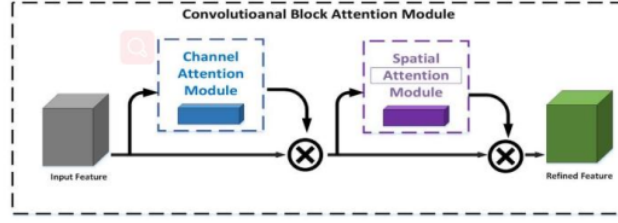


Fig.4-4. CBAM structure

In the input feature map, a one-dimensional vector representing the channel attention is used to represent the correlation between channels. Each channel in the feature map can be treated as a feature extractor, and the channel attention can propose the feature content in the image. In order to implement channel attention more effectively, CBAM designs a channel attention module, the structure of which is shown in Figures 4-5. First, CBAM performs a spatial squeeze on the input feature map. This is achieved by combining an average pooling operation with an entire operation. The average pooling operation is used to extract spatial information efficiently and helps the neural network model learn the image's target objects to some extent. The maximum pooling operation captures feature cues representing different objects, thus enabling a better channel domain attention mechanism. The combined use of pooling and max pooling operations has been shown to enhance network performance compared to average pooling and max pooling operations alone.

The CBAM completes the compression of the spatial information in the feature map through the average pooling operation and the maximum pooling operation and generates two different spatial contexts, including the feature map F_{avg}^C representing the average pooling and the feature map F_{max}^C representing the maximum pooling. that is used to generate the final one-dimensional vector $M_c(F) = R^{(C \times 1 \times 1)}$ representing the channel attention. This shared parameter network consists of a Multi-Layer Perceptron (MLP), a fully connected neural network structure including a hidden layer. In order to reduce the computational parameters, the number of neurons in the hidden layer is designed to be C/r , where r is the decay rate. After the two feature maps are used as separate inputs to the shared network, the two generated feature vectors are summed element by element to get a one-dimensional vector M_c representing the channel

attention. In summary, the process of calculating channel attention is as follows (4-8).

$$M_c(F) = \sigma \left(MLP(AvgPool(F)) + MLP(Maxpool(F)) \right)$$

$$M_c(F) = \sigma \left(W_1 \left(W_0(F_{avg}^c) \right) + W_1 \left(W_0(F_{max}^c) \right) \right) \quad (4-8)$$

Where σ denotes the Sigmoid, $W_0 \in R^{(C/r \times C)}$ and $W_1 \in R^{(C \times C/r)}$ denote the weights of the MLP. W_0 , and W_1 are shared by the two inputs, while the activation function of the W_0 layer is the ReLU function. CBAM generates a two-dimensional matrix for representing spatial attention by exploiting the spatial correlation of the input feature maps. The spatial attention module designed by CBAM is illustrated in Figure 4-5 the spatial attention module first performs a mean pooling operation and a maximum pooling operation on the channel axis for the channel optimization feature map F' . It aggregates them together to generate a valid feature descriptor, while the pooling operation on the channel axis effectively helps the model select regions with high information content in space. For the generated feature descriptors, the spatial attention module uses a convolution layer to convolve them and ultimately generates a two-dimensional matrix $M_s(F) \in R^{(H \times W)}$ or representing spatial attention.

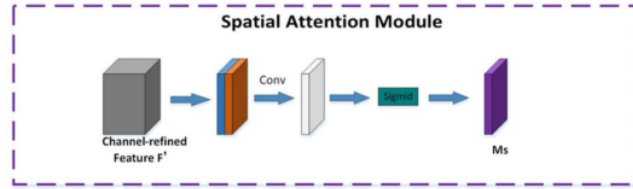


Fig.4-5. Spatial Attention Module

The ECANet[45] proposed by Qilong Wang et al. also implements the channel attention mechanism, which can be seen as an improved version of SENet. The authors of ECANet argue that SENet has side effects on predicting the channel attention mechanism. Capturing all channel dependencies is inefficient and unnecessary. The authors argue that convolution has good cross-channel information acquisition capabilities, removing the fully-connected layer from the original SE module and learning directly from a 1D convolution on the globally averaged pooled features. Since 1D convolution is used, the choice of the size of the convolution kernel for 1×1

convolution becomes very important, as the size of the kernel for 1×1 convolution affects the number of channels to be considered for the calculation of each weight of the attention mechanism. The ECANet structure is shown in Figure 4-6.

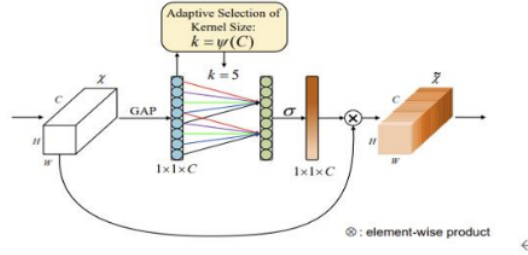


Fig.4-6 ECANet Structure

CHAPTER 4.4 ASPP

In ASPP, downsampling operations are usually indispensable. Successive downsampling of the combined layers can lead to a severe reduction in the spatial resolution of the feature map. If the compression level is high, it will make the subsequent upsampling or deconvolution to the original map size more difficult. The Deep Lab team has proposed null convolution for dense feature extraction and increased the perceptual field to solve this problem.

The perceptual field is the size of the area of the previous layer corresponding to an element in the output of a layer in a CNN, commonly understood as which elements of that layer's feature values are influenced by which elements of the previous layer [52]. It is usually calculated sequentially from the first layer after the input layer backward and satisfies the following equation.

When $n=1$:

$$RF_n = k_n \quad (4-9)$$

When $n \geq 2$:

$$S_n = S_{n-1} \times s \quad (4-10)$$

$$RF_n = RF_{n-1} + (k_n - 1) \times S_{n-1} \quad (4-11)$$

Where s denotes the step size of the current layer, S_n denotes the step size of the

first n layers of the network, RF_n is the receptive field of the previous layer, RF_{n-1} is the receptive field of the current layer, and k_n represents the size of the convolution kernel. Therefore, when a denser feature map is obtained by simply reducing the step size. If the perceptual field is to remain constant, the size of the convolution kernel needs to be increased, giving rise to the hole convolution algorithm defined in Equation (4-12). Taking a one-dimensional signal as an example, assume that the input is x , for a point i , the corresponding output y can be expressed as

$$y_i = \sum_{k=1}^K x_{i+d \cdot k} w_k \quad (4-12)$$

Where d denotes the null rate, K denotes the size of the convolution kernel, and w_k represents the number k weight of the convolution kernel [46]. When $d = 1$, the Equation reduces to an ordinary convolution. When $d > 1$, the hole convolution is equivalent to inserting $d - 1$ holes between adjacent points in the ordinary convolution kernel to upsample the kernel and then convolve the input x . From this definition, when the hole rate is d , the size of the convolution kernel for the hole convolution is:

$$k_{atrous} = (d - 1) \times (k_{size} - 1) + k_{size} \quad (4-13)$$

Where k_{atrous} denotes the size of the null convolution and k_size denotes the size of the standard convolution kernel. When only the perceptual field of the convolution kernel on the current layer is calculated, it is represented by equation (4-14).

$$RF = k_{atrous} = (d - 1) \times (k_{size} - 1) + k_{size} \quad (4-14)$$

Dilated convolution uses an increase in holes to increase the receptive field so that a 3x3 convolution kernel has a 5x5 receptive field (dilation = 2) or larger, without any change in the number of parameters or computation of the neural network, and therefore no downsampling operation is required. [47]. Atrous convolutions introduce a new parameter to the convolution layer: This parameter defines the spacing of the values as the convolution kernel processes the data. In other words, compared to the original standard convolution, the dilated convolution has an additional hyperparameter called the dilation rate, which refers to the number of intervals before each point of the kernel [48]. As shown in the diagram, this is the null convolution at rate=1. For the null

convolution, the size of the perceptual field is positively related to the null rate. However, a more significant null rate will result in a larger field of perception, which can capture more global features and thus help to control the whole. However, as the field of perception gradually increases, it sacrifices more and more local information, such as edge details.

In order to effectively balance local and global information, the Deep Lab team proposed the ASPP model in DeepLabv2. Inspired by SPPNet, ASPP concatenates a set of null convolutions with different null rates to extract and fuse multi-scale features under different perceptual fields. As shown in Figures (4-7), the model adopts a four-level pyramid structure, corresponding to four mutually independent branches. Each branch consists of a three-layer network (fc6-fc7-fc8), where fc6 is the null convolution layer, and fc7 and fc8 are the regular convolution layers. Deep Labv2 designed two different combinations of null rates for the null convolution of fc6 to extract feature maps at different scales: ASPP-S (2, 4, 6, 8) and ASPP-L (6, 12, 18, 24), both with a convolution kernel size of 3. The combination of ASPP-L with a more significant hole rate has been shown to perform better through comparative experiments. In addition, fc7 and fc8 both use regular convolution of size 1×1 , which can deepen the number of layers of the network and thus enhance the fitting ability. They perform a slightly different function from the 1×1 convolution in the PSPNet pyramid pooling module. The convolution in PSPNet is the downsampling of the output of the pooled feature blocks to prepare for the subsequent upsampling and feature fusion. In contrast, the fc6 outputs of the four branches of ASPP have the same feature map size so that they can be directly stitched together into a feature tensor without up-sampling operations.

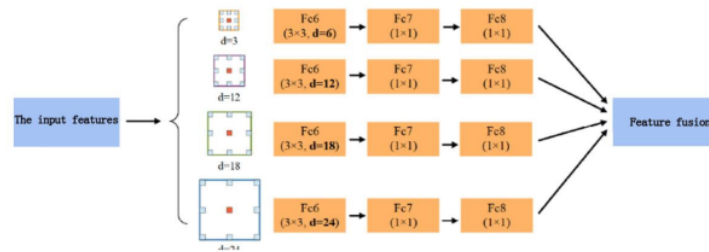


Fig 4-7. ASPP Structure

Although the experimental results show that ASPP-L performs better with a larger void ratio, a larger void ratio is not better. Because too large a cavity rate increases the perceptual field, it also increases the difficulty of local feature extraction and eventually leads to insufficient edge detail information. Therefore, it is essential to choose the correct cavity rate. Simply increasing the cavity rate to increase the perceptual field is not advisable, and other techniques and strategies must be incorporated.

CHAPTER 5. DESIGN/DATA COLLECTION

CHAPTER 5.1 NETWORK STRUCTURE

As shown in Figure 5-1, CEA-UNet is our proposed network for automatic segmentation of COVID-19 CT images, and we all model a structure based on coding and decoding. We constructed a network based on the U-Net network, which is suitable for segmenting the focal regions of COVID-19 lung CT images. The network consists of 4 encoders and four decoders. U-Net has four encoders, while ResNet50 has four large modules. First, we replace the backbone network for feature extraction of U-Net with the idea of ResNet50 and then modify the residual module of ResNet50. A feature fusion approach is used, an ECA attention mechanism is added after Res Block1, and an ASPP module is added after Res Block4. The input image size is 512×512 and the number of channels is 3. After the initial input to the Conv block1 module, the output size of the image is $256 \times 256 \times 64$; after the extracted features are input to the Res Block1 module, the output size of the image is $128 \times 128 \times 256$; after the extracted features are input to the Res Block2 module, the output size of the image After the extracted features are input to the Res Block3 module, the output size of the image is $32 \times 32 \times 1024$; after the extracted features are input to the Res Block, the output size of the image is $16 \times 16 \times 2048$.

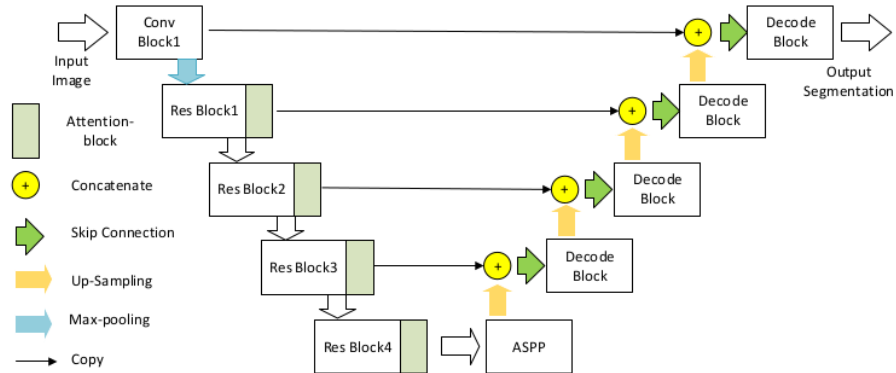


Fig.5-1 . CEA-UNet Structure

CHAPTER 5.2 RES BLOCK

The Res Block is divided into two steps, including the Conv Block and the Identity Block, and the entire Res Block is based on the results of the ResNet50 network, which

is divided into four parts, Res Block1, Res Block2, Res Block3 and Res Block4. Res Block1, Res Block2, Res Block3, Res Block4, where Res Block1 has 9 layers, Res Block1 has 12 layers, Res Block1 has 18 layers, and Res Block4 has 9 layers. In addition to the initial trial of the image input, a convolutional layer and a pooling layer are added, making a total of 50 layers. The Conv Block is composed as shown in Figure 5-2 and the Identity Block is composed as shown in Figure 5-3.

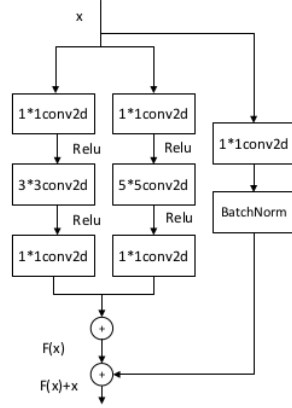


Fig.5-2. Res block

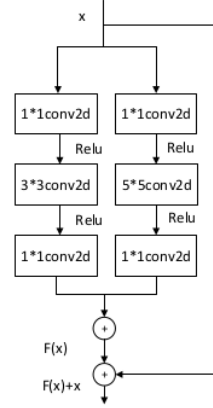


Fig.5-2. Identity block

CHAPTER 5.3 EXPERIMENTAL DATASET

The CT image segmentation dataset for this project was obtained from the Chinese National Center for Biological Information CC-CCII [49], which is a public dataset and available worldwide. The CC-CCII dataset was obtained from the Third Affiliated Hospital of Sun Yat-sen University, the First Affiliated Hospital of Anhui Medical University, the West China Hospital of Sichuan, the People's Hospital of Jiangsu Province, the Central People's Hospital of Yichang, and the People's Hospital of Wuhan University in China. Medical imaging experts manually labeled 750 slices from 150 patients as Background, Lung field, Ground-glass opacity, and Consolidation for image segmentation. The image size of this dataset is all 512×512 pixels.

CHAPTER 5.4 EXPERIMENTAL ENVIRONMENT

The mIoU evaluation index was used to evaluate the experimental results, and the evaluation was calculated as in equation (5-4) :

$$mIoU = \frac{1}{k+1} \sum_{i=0}^k \frac{p_{ii}}{\sum_{i=0}^k p_{ij} - \sum_{i=0}^k p_{ji} - p_{ii}} \quad (5-4)$$

Where i denotes the true value, j denotes the predicted value, p_{ij} means to predict i as j , p_{ii} means to predict i as i , and p_{ji} means to predict j as i . The larger the value of the evaluation metric, the more accurate the model prediction is.

This paper used the open-source deep learning framework PyTorch 1.8.0 to implement the proposed CEA-Unet and used a GPU accelerated network model for training and testing with a GeForce RTX3070ti graphics card. One hundred fifty rounds of a batch size of 2 were used for training and testing. The optimizer was Adam, and the initial learning rate was 0.0001.

CHAPTER 6. RESULTS AND DISCUSSION

CHAPTER 6.1 CONTRAST EXPERIMENT

Four sets of comparison networks were set up to evaluate the effectiveness of replacing the backbone of U-Net with the ResNet50 network proposed in this paper and verify the impact of different attention mechanisms on network performance.

Using different attention mechanism modules, including ECA-Block, CBAM-Block, and SE-Block, the Res-Unet named Combine-Res-Unet after feature fusion was firstly interspersed in Res Block3 and Res Block4, named as CoRes-Unet-ECA and CoRes-Unet-CB, respectively, and the experimental results are detailed in Table 2.

Table 2. Experimental Results

模型	mIoU
Combine-Res-Unet	74.56%
CoRes-Unet-ECA	78.52%
CoRes-Unet-CB_	76.64%
CoRes-Unet-SE	77.91%

The loss as well as mIoU training plots of the above four models are shown in Figures (6-1), (6-2), (6-3) and (6-4) respectively.

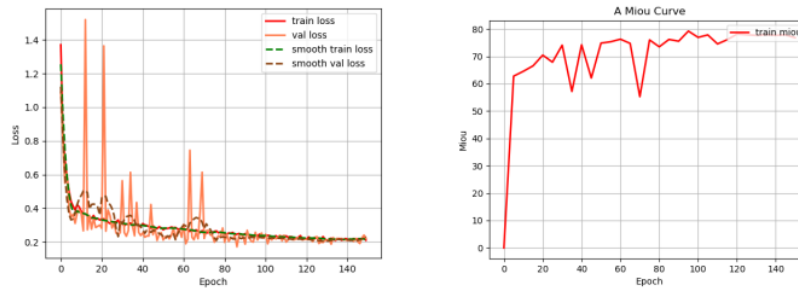


Fig 6-1. Combine-Res-Unet

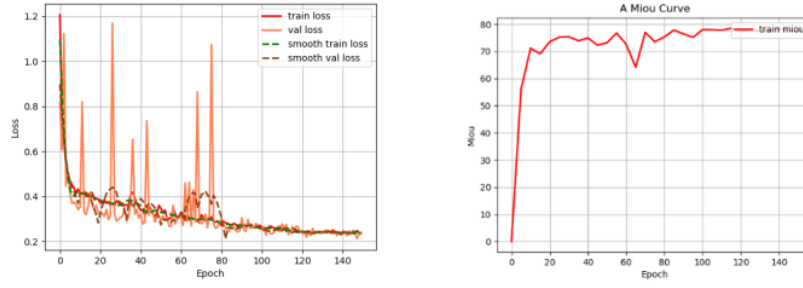


Fig.6-1. CoRes-Unet-ECA

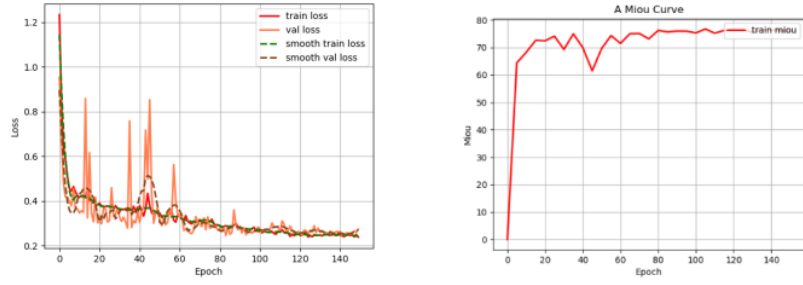


Fig.6-3. CoRes-Unet-CB

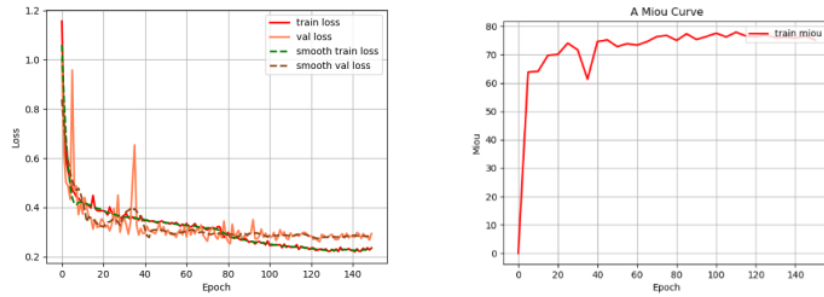


Fig.6-4. CoRes-Unet-SE

From the above experiments, we can see that adding the attention mechanism module to the neural network has a certain degree of improvement on the neural network segmentation, with the bolded representing the module with the best results. The attention mechanism is a plug-and-play module, and placing it in different locations may have different effects, so we added the attention mechanism module to different locations. The ECA module and CBAM attention module are interspersed in Res Block1, Res Block2, Res Block3, and Res Block4, named CoUnet-ECA and CoUnet-

CB, respectively; the ECA module attention module is interspersed in Res Block1 and Res Block2, and the CBAM module has interspersed. The details of the experimental results are shown in Table 3.

Table 3. Experimental Results

模型	mIoU
CoUnet-CB	79.03%
CoUnet-ECA	79.53%

As seen from the table, the model metrics increased by 0.5% with the addition of ASPP, demonstrating the improved segmentation capability of the model images with the addition of the ASPP module. Loss and mIoU plots for CEA-Unet are shown in Figure (6-5) below, with the entire network model stabilizing by the 150th round of training, as seen in the Loss plot.

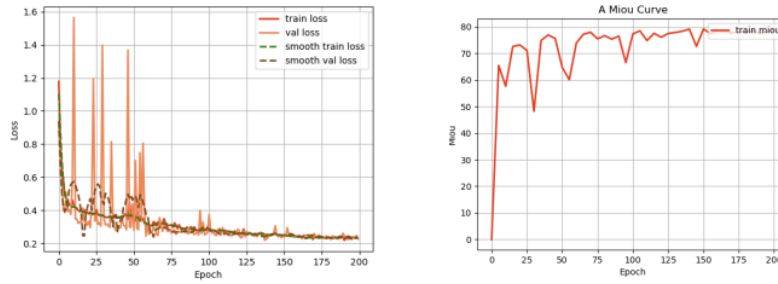


Fig.6-5. CEA-Unet

CHAPTER 6.2 ABLATION EXPERIMENTS

This article introduces the experimental results obtained from the experiments presented in the paper [50] on the CC-CCII dataset with U-Net, U-Net++ [51], PSPNet [52], and MA-Net [53], respectively. After the introduction of ASPP in CoUnet-ECA, named CEA-Unet. To validate the proposed CEA-UNet model's performance, we cite this paper's training results and compare them with DeeplabV3, and the comparison results are detailed in Table 4.

TABLE.4 Comparison results

model	mIoU
MA-Net	60.76%
DeeplabV3	65.56%

PSPNet	51.53%
U-Net	61.05%
U-Net++	61.21%
CEA-UNet	79.53%

Table test results show that the mIoU metric of the CEA-UNet model proposed in this article is better than the comparison model.

CHAPTER 7. CONCLUSIONS AND FUTURE WORK

CHAPTER 7.1 CONCLUSIONS

Medical imaging is currently being used extensively in medical image segmentation as a tool to assist clinicians in diagnosing medical conditions. In order to improve the survival rate of pneumonia patients, early detection and treatment must be used. Early detection is, therefore, essential for treating patients with pneumonia. The application of computer processing techniques to help doctors detect areas of the lung and pneumonia effectively avoids missed and misdiagnosed cases and helps to treat the patient's condition more effectively. Therefore, it is clinically essential to enhance the accuracy of image segmentation of lung and pneumonia regions. Since deep learning models perform better in processing complex information, this paper uses a convolutional neural network-based algorithm for COVID-19 lesion segmentation of lung CT images. Compared to other neural network models, the U-Net model has the advantage of having a high accuracy rate for image segmentation, even with a small training set. The U-Net model differs in adding an up-sampling operation after the fully connected layer to ensure that the output and input images are the same sizes. Up-sampling corresponds to down-sampling and is a process of increasing the image's resolution. The inverse pooling operation gradually restores the size of the image. The pooling operation causes a loss of image feature information, and the process is irreversible. However, the inverse pooling operation can be considered an approximate recovery of image feature information. The U-Net model has shown better performance in medical image processing compared to other models. Therefore, this paper designs an algorithm based on the U-Net model for processing the segmentation problem of lung CT images. The main innovations of the algorithm used in this paper compared to the classical U-Net model are: (1) the introduction of ResNet50, which has the advantage that all layers of the network can be connected while ensuring that the information in each layer is closely linked. Each layer fuses the input features from all its previous layers and passes the output feature map to all the layers after it. (2) Introduction of the ASPP module.

This increases the perceptual field of image feature information and improves the robustness of image features. (3) Introduction of CBAM. Attentional supervision in learning deep-level feature information can effectively extract feature information of lung image elements. In the algorithm implementation module, Tensor Flow is used as the back-end of the Keras open source framework. Python was used to implement COVID-19 lesion segmentation based on lung CT images, including the experimental environment, parameter configuration, and data pre-processing. In the pre-processing data stage, the original image histogram is equalized, median filtering is used to increase the contrast of the image and reduce the effect of noise, and the data is enhanced by image rotation, flipping, scaling (scaling up or down the image) and adding image brightness to form a new image. The network performance was evaluated using three metrics: accuracy, recall, and precision. The experimental results show that the algorithm outperforms other algorithms in segmentation. The algorithm can be widely applied to different medical image segmentation tasks due to its strong generalization.

CHAPTER 7.2 FUTURE WORK

The primary research of this paper is COVID-19 lesion segmentation of lung CT images based on convolutional neural networks. Deep learning technology has a good performance in lung image segmentation. However, due to the singularity of the image dataset used in this paper, further experimental studies are still needed for COVID-19 lesion segmentation of lung CT images using deep learning methods. Early detection and diagnosis of pneumonia are crucial, and further research on pneumonia will continue in the future to enhance the method proposed in this paper based on this network, to design an efficient network model for lung image segmentation, to enhance accuracy, and to solve problems and provide assistance in clinical medicine applications. (1) The attention mechanism is introduced in the U-Net network, which focuses more on the practical features of the image when learning the image feature information, improving the model's ability to extract features and the accuracy rate of segmenting the target region of the image. The attention mechanism has a particular influence on the

speed of the model operation, so the following research should focus on improving the speed of the model operation to ensure the accuracy of the model segmentation. (2) Since the COVID-19 focal areas of lung CT images are not evident in the early stage when segmenting the images, the target area may be small or non-existent. However, the background area is large, and many useless image segmentation results may occur if the image is segmented directly. It is, therefore, necessary to first detect the target region of the image before segmentation and then perform effective segmentation when the target region meets the segmentation requirements. This reduces the model's time complexity and operational complexity when segmenting the image. (3) The data used in this paper is relatively homogeneous on how to expand and normalize the COVID-19 lesion data of lung CT images used for the study and build a complete and standardized data size dataset. In addition, with a small dataset, how to make the model have higher performance when training samples is also a director of our subsequent research. (4) Deep learning algorithms can accurately segment images, gaining time for treating patients' conditions and providing a reliable basis for doctors to diagnose their conditions. We, therefore, use python to design a system that can automatically segment the COVID-19 focal areas of lung CT images for use in practical clinical medicine to reduce the workload of doctors.

CHAPTER 8. REFLECTION

Thesis Summary During the year of postgraduate life in life, we also did not travel to the UK to receive offline lectures due to COVID-19. Nor did we get to experience the humanities of the UK, for which I feel very sorry. I hope that I will still visit the UK and go around the campus when I have the chance in the future. During my short year of study, I was still impressed by many of the teachers and I am very grateful to the professors for their teaching during the year. To gain a deeper understanding and mastery of the postgraduate program and to strengthen my professional skills. I chose the final project course: COVID-19 image segmentation based on deep learning. The graduation design is practical, comprehensive, exploratory, and applied. This chosen topic aims to improve the accuracy of image segmentation. The final project I did was also related to COVID-19, and we did not study in the UK because of COVID-19. With the completion of my final project, it will mean that my year of postgraduate life is coming to an end, but I feel that I have worked hard and had fun during this time. During my busy days, I used to go through the books I had studied to solve technical problems. I experienced and felt a lot after this time. It has been an intense and fulfilling few months of work.

The graduation project has finally come to an end. Looking back on my experiences and feelings during this period, I am overwhelmed with emotions and I have numerous unforgettable memories and gains during the graduation design process. To improve the accuracy of the segmentation of the lesion area. The system solution was gradually established by reviewing data and constantly adjusting the hyperparameters and modifying the network model. But at the same time, it was sweet to look at the graduation design on the computer screen. The process of creating the graduation design was a re-learning and improvement process for me.

I have learned new knowledge and gained insight into the process. In the future, I will continue to enrich myself. I have benefited from the careful guidance and selfless care of my supervisors throughout this year of study and life, from the choice of topic to the

specific idea and content. I would like to express my deepest gratitude and respect to all my teachers. The experience of doing my graduation design will also benefit me for life. I feel that doing a graduation design is something that I have to put my heart and soul into, and it is truly a process of independent learning and research. Without learning, you cannot have the ability to research, and without your research, you will not make a breakthrough. I hope that this experience will inspire me to continue to improve in my future studies. I owe the successful completion of this design to the conscientiousness of my teachers, who enabled me to grasp and apply my professional knowledge in design. It is thanks to their help and support that I was able to complete my graduation project successfully. Finally, I would like to thank my supervisor, Terry, for his careful guidance and care, which enabled us to complete our graduation design successfully.

REFERENCE

- [1] H. N. Silverman J D, Washburne A D, “Using influenza surveillance networks to estimate state-specific prevalence of SARS-CoV-2 in the United States,” *Science translational medicine*, vol. 12, pp. 554, 2020.
- [2] B. A. Chung M, Mei X, et al. , “CT imaging features of 2019 novel coronavirus ” *Radiology*, vol. 295, no. 1, pp. 202-207, 2020.
- [3] M. D. Younus, M. J. M. Zedan, F. L. Malallah, and M. G. Saeed, “COVID-19 Improved Diagnoses Based on the Open-morphology Filter and Deep-learning,” *Recent Patents on Engineering*, vol. 16, no. 3, 2022.
- [4] X. L. Minglei Li, Yuchen Jiang, Jiushi Zhang, Hao Luo, Shen Yin, “Explainable multi-instance and multi-task learning for COVID-19 diagnosis and lesion segmentation in CT images,” *Knowledge-Based Systems*, vol. 252, pp. 109278, 2022.
- [5] E. J. Topol, “High-performance medicine: the convergence of human and artificial intelligence,” *Nat Med*, vol. 25, no. 1, pp. 44-56, Jan, 2019.
- [6] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, “A guide to deep learning in healthcare,” *Nat Med*, vol. 25, no. 1, pp. 24-29, Jan, 2019.
- [7] G. B. S. Norgeot B, Butte A J, “A call for deep-learning healthcare,” *Nature Medicine*, vol. 25, no. 1, pp. 14-15, 2019.
- [8] B. Y. Le Cun Y, Hinton G., “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.
- [9] J. F. Hong'an Li, Qiaozhi Hua, Xinpeng Li, Zheng Wen, Meng Yang, “Biomedical sensor image segmentation algorithm based on improved fully convolutional network,” *Measurement*, vol. 197, 2022.
- [10] J. a. L. Hao, Xiaoxi and Hou, Yanxia, “Magnetic Resonance Image Segmentation Based on Multi-Scale Convolutional Neural Network},” *IEEE Access*, vol. 8, pp. 65758-65768, 2020.

- [11] G. P. L. -C. Chen, I. Kokkinos, K. Murphy and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.40, no. 4, pp. 834-848, 2018.
- [12] K. H. Girshick, "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980-2988, 2017.
- [13] A. M. G. Lin, C. Shen and I. Reid, "RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5168-5177, 2017.
- [14] S. E. Long J, Darrell T. Fully Convolutional Networks for Semantic Segmentation, "Fully Convolutional Networks for Semantic Segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431-3440.
- [15] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pp. 234-241, 2015.
- [16] L. S. S. Abdulkadir A, et al. , "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," *Computerized Medical Imaging & Graphics*,, vol. 76, no. 1, pp. 3-6, 2016.
- [17] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," *Nat Methods*, vol. 18, no. 2, pp. 203-211, Feb, 2021.
- [18] P. G. Chen L C, Kokkinos I, et al., "Deep Lab: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," in *ICLR*, 2014, pp. 834-848.
- [19] P. G. Chen L C, Kokkinos I, et al., "Rethinking Atrous Convolution for Semantic Image Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 587-597.

- [20] M. A. Lin G S, Shenc H, et al., "Refinenet: Multipath Refinement Networks for High-resolution Semantic Segmentation," in Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1925-1934.
- [21] G. R, "Optical stimulation connects electrical and biological neurons," *Laser Focus World*, vol. 57, pp. 2, 2021.
- [22] M. C. SERHAT K, "RSig ELU: A nonlinear activation function for deep neural networks," *Expert Systems With Applications*., pp. 114805, 2021.
- [23] H. P. D. B, "Sigmoid transfer functions in backpropagation neural networks," *Analytical Chemistry*, vol. 65, no. 15, pp. 2167-2168, 1993.
- [24] B. Y. GOODFELLOW I, COURVILLE A. Machine learning basics "Deep learning," vol. 1, pp. 98-64, 2016.
- [25] J. J. DAS K, RAO J, "Mean squared error of empirical predictor " *The Annals of Statistics*, vol. 32, no. 2, pp. 818-840, 2004.
- [26] J. a. S. Cao, Zhe and Yu, Liyun and Chang, Dongliang and Li, Xiaoxu and Ma, Zhanyu, *Softmax Cross Entropy Loss with Unbiased Decision Boundary for Image Classification*, p.^pp. 2028-2032, 2018 Chinese Automation Congress (CAC), 2018.
- [27] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.
- [28] A. a. E.-B. El-Sawy, Hazem and Loey, Mohamed, "CNN for handwritten arabic digits recognition based on LeNet-5," *International conference on advanced intelligent systems and informatics*: Springer, 2016.
- [29] F. N. a. H. Iandola, Song and Moskewicz, Matthew W and Ashraf, Khalid and Dally, William J and Keutzer, Kurt, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," *arXiv preprint arXiv:1602.07360*, 2016.
- [30] Z. a. J. Zhong, Lianwen and Xie, Zecheng, "High performance offline handwritten chinese character recognition using googlenet and directional

feature maps," *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 846--850: IEEE, 2015.

- [31] M. a. X. Shi, Lijun and Chen, Xiang, "A Novel Facial Expression Intelligent Recognition Method Using Improved Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 57606-57614, 2020.
- [32] G. Z. Liangji Zhang, Chao Lu, Aibin Chen, Yanfeng Wang, LiuJun Li, Weiwei Cai, "A fusion data augmentation method for tomato-leaf disease identification," *Applied Soft Computing*, vol. 123, pp. 108969, 2022.
- [33] H. a. C. Ma, Jianqiu and Mi, Bo and Huang, Darong and Liu, Yang and Zhang, Zhenyuan, "Dark web traffic detection method based on deep learning," *2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 842-847, 2021.
- [34] C. Qin, Y. Wu, W. Liao, J. Zeng, S. Liang, and X. Zhang, "Improved U-Net3+ with stage residual for brain tumor segmentation," *BMC Med Imaging*, vol. 22, no. 1, pp. 14, Jan 27, 2022.
- [35] S. S. R. Kumar A, Arora A, et al. , "Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network," vol. 7, pp. 23319-23328, 2019.
- [36] L. S. Lin C H, "Inverse compositional spatial transformer networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2568-2576.
- [37] H. a. F. Qi, Xiaoyan and He, Xuejie and Liu, Honghong, "A Multiple Object Tracking Method Based on Optimized FairMOT," *2021 2nd International Conference on Computer Science and Management Technology (ICCSMT)* , pp. 358-362, 2021.
- [38] S. L. Hu J, Sun G, " Squeeze-and-excitation networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132-7141, 2018.
- [39] C. Li, T. Liu, and W. Ye, "A novel feature-based network with sequential information for textile defect detection," *Coloration Technology*, vol. 136, no. 6,

pp. 476-484, 2020.

- [40] S. L. Hu J, Sun G, “Squeeze-and-excitation networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132-7141.
- [41] J. M. Wang F, Qian C, et al., “Residual attention network for image classification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3156-3164.
- [42] P. J. Woo S, Lee J Y, et al, “Cbam: Convolutional block attention module,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3-19.
- [43] L. J. Fu J, Tian H, et al. Dual attention network for scene segmentation, “Dual attention network for scene segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, pp. 3146-3154.
- [44] J. Z. Fagui Liu, Lailei Zheng, Cheng Chen, “Combining attention-based bidirectional gated recurrent neural network and two-dimensional convolutional neural network for document-level sentiment classification,” *Neurocomputing*, vol. 371, pp. 39-50, 2020.
- [45] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, “ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11531-11539.
- [46] I. Barukčić, Q. Liang, Y. Li, K. Yang, X. Wang, and Z. Li, “Long-term recurrent convolutional network violent Behaviour recognition with attention mechanism,” *MATEC Web of Conferences*, vol. 336, 2021.
- [47] D. E. Sizhe Chen, Bo Ma, Ying Cai, Fangyao Liu, “Improvement of AnyNet-based end-to-end phased binocular stereo matching network,” *Procedia Computer Science*, vol. 199, pp. 1450-1457, 2022.
- [48] T. Ku, Q. Yang, and H. Zhang, “Multilevel feature fusion dilated convolutional

network for semantic segmentation,” *International Journal of Advanced Robotic Systems*, vol. 18, no. 2, 2021.

- [49] K. Zhang, X. Liu, J. Shen, Z. Li, Y. Sang, X. Wu, Y. Zha, W. Liang, C. Wang, K. Wang, L. Ye, M. Gao, Z. Zhou, L. Li, J. Wang, Z. Yang, H. Cai, J. Xu, L. Yang, W. Cai, W. Xu, S. Wu, W. Zhang, S. Jiang, L. Zheng, X. Zhang, L. Wang, L. Lu, J. Li, H. Yin, W. Wang, O. Li, C. Zhang, L. Liang, T. Wu, R. Deng, K. Wei, Y. Zhou, T. Chen, J. Y. Lau, M. Fok, J. He, T. Lin, W. Li, and G. Wang, “Clinically Applicable AI System for Accurate Diagnosis, Quantitative Measurements, and Prognosis of COVID-19 Pneumonia Using Computed Tomography,” *Cell*, vol. 181, no. 6, pp. 1423-1433 e11, Jun 11, 2020.
- [50] B. A. Krinski, D. V. Ruiz, and E. Todt, “Spark in the Dark: Evaluating Encoder-Decoder Pairs for COVID-19 CT’s Semantic Segmentation,” in 2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE), 2021, pp. 198-203.
- [51] N. T. Z. Zhou, and J. Liang, “UNet++:A nested U-Net architecture for medical image segmentation,” *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, vol. 11045, pp. 3-11, 2018.
- [52] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid Scene Parsing Network,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6230-6239.
- [53] T. Fan, G. Wang, Y. Li, and H. Wang, “MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation,” *IEEE Access*, vol. 8, pp. 179656-179665, 2020.

Research on segmentation method of infected regions in COVID-19CT images based on deep learning

GRADEMARK REPORT

FINAL GRADE

/100

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58
