Research and implementation of lumbar muscle segmentation based on U-net network for CT images

Lu Zhang[a], Chenkai Song[a], Shichen Huang[b], Xiaozhao Chen*[a]
[a]Shenyang Pharmaceutical University, 103 Wenhua Rd, Shenhe Dist, Shenyang, Liaoning, P.R. China, 110016; [b]University of Nottingham International College, Nottingham, United Kingdom NG15NJ
* Corresponding author: chenxiaozhao2013@163.com

ABSTRACT

The paravertebral muscle is a vital structure that maintains the stability of the lumbar spine. The decrease in muscle mass and the increase in fat content of the paravertebral muscle are closely related to the occurrence of various lumbar diseases. The degeneration of the paravertebral muscles is associated with various diseases, such as low back pain, degenerative scoliosis, osteoporosis, and so forth. At present, although a large number of studies have reported on vertebral computed tomography (CT) image segmentation, studies on paravertebral muscle segmentation are few. This study aimed to achieve the segmentation of the muscle region in vertebral CT images to provide clinically feasible index observation data for the diagnosis, treatment, and prognosis of related diseases. The traditional segmentation method is time-consuming and laborious. Also, the segmentation results vary significantly due to the different levels of experience of clinicians. Further, the repeatability is poor. This study used the automatic image segmentation model based on a deep learning algorithm, namely the U-net network model, to achieve vertebral muscle segmentation in CT images. The average Dice coefficient reached 0.9178, indicating a good segmentation effect of the segmentation model based on the U-net network. Based on the results of paravertebral muscle segmentation, automatic measurement and analysis of the cross-sectional area, paravertebral muscle density, and degree of fat infiltration can be further realized, thereby guaranteeing the prognosis of patients with spinal diseases.

Keywords: Deep learning, U-net network, muscle segmentation

1. INTRODUCTION

1.1 Research significance

Related studies found that the normal lives of 619 million people were affected by low back pain (LBP) in 2020. It is expected that the number of patients with lower back pain will reach 843 million people by 2050. Studies on LBP are now gradually focusing on the application of spinal skeletal muscle imaging to disease diagnosis and treatment. The state of the paraspinal muscles is closely related to LBP. Clinically, the quantitative observation of spinal muscles can be performed using medical images, and muscle morphology and composition can be evaluated [1].

The degeneration of paraspinal muscles is also related to many diseases besides lower back-related diseases. For example, degenerative scoliosis (DS), whose incidence increases with age, has a significant impact on the quality of life of patients and has gradually become a global public health problem. Existing studies indicate that clinical doctors should design surgical plans for patients with DS to ensure the stability of internal fixation while preserving as much muscle tissue as possible, thereby stabilizing the spine and playing a positive role in subsequent recovery [2]. In clinical analysis, the reduction in cross-sectional area (CSA) indicates paraspinal muscle atrophy, and the measurement values of paraspinal muscle CSA are affected by various factors [3].

The implementation of image segmentation technology not only improves the accuracy of medical image analysis but also has important application value in the fields of disease diagnosis, treatment plans, and postoperative monitoring. In this study, the U-net network model, a classic model, was used to realize the segmentation of the vertebral muscle area in computed tomography (CT) images, thereby obtaining a muscle segmentation effect map. By examining the related clinical information in the segmentation effect map, this study laid a foundation for further exploring the correlation between muscle morphological changes and disease progression [4].

International Conference on Future of Medicine and Biological Information Engineering
(MBIE 2024), edited by Yudong Yao, Xiaoou Li, Xia Yu, Proc. of SPIE Vol. 13270,
1327006 · © 2024 SPIE · 0277-786X · doi: 10.1117/12.3044686

1.2 Application of deep learning in medical image processing

In modern medicine, the role of medical images has become increasingly prominent; doctors use these images for diagnosis and treatment. In particular, doctors pay close attention to the region of interest in the images, with experienced doctors manually drawing regions of interest (ROIs) during image analysis to more accurately analyze the patient's current condition [5]. However, manually drawing ROIs is time-consuming and labor-intensive. Hence, an automated medical image segmentation algorithm is urgently needed in clinical practice.

Deep learning–based medical image segmentation has developed rapidly and become a key step in computer-assisted diagnosis. The application of deep learning algorithms in clinical research has increased, providing doctors with a large amount of valuable data to help them better understand the development of diseases and grasp the treatment response and prognosis of patients. The application of deep learning avoids the disadvantages of manual annotation, including excess time consumption and low repeatability [6]-[7]. After learning the features of large amounts of imaging data, deep learning models can capture subtle changes that are difficult for the human eye to detect, making it easier to find subtle disease features.

2. METHODS

The study chose the U-net network as the segmentation algorithm, which employed a fully convolutional network for semantic segmentation. It consisted of a compression path and an expansion path in a symmetrical U-shaped structure, which had a certain impact on the design of several subsequent segmentation networks [4]. The classic segmentation algorithm is briefly introduced as follows.

2.1 U-net network architecture

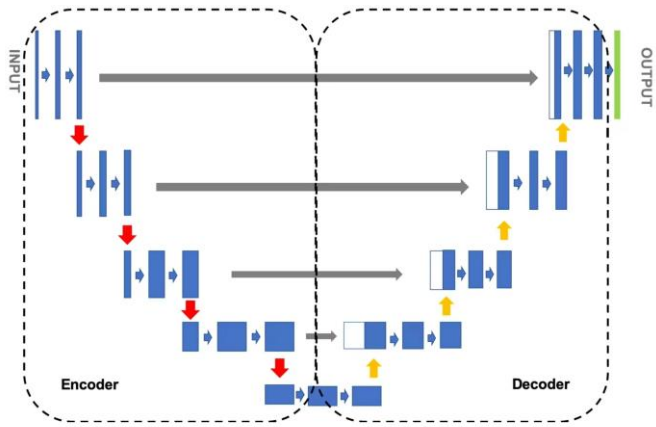

Figure 1. U-net architecture diagram: The blue arrow represents conv 3x3, ReLU, the gray arrow represents copy and crop, the red arrow represents max pool 2x2, the yellow arrow represents up-conv 2x2, and the last arrow represents conv 1x1.

The U-net network is divided into two parts, the left side is the contraction path. The downsampling and convolution modules are used to extract features of different scales. On the right side is the extended path. The upsampling and convolution modules are used to restore the scale, fuse the previous features, and then gradually restore the image [8]. The convolution module is composed of two consecutive convolution layers, which can achieve feature extraction with larger size and higher depth. Downsampling can reduce the image scale, and upsampling or deconvolution layer can increase the image scale [9]. The gray arrow line in Figure 1 represents the jump connection. It merges the features on the left contraction path with the image on the right expansion path on the same scale, to provide better constraints on the expansion path. The network can easily output the expected results of the experiment [10].

2.2 Adam optimizer

The training of the deep learning model is an iterative optimization process, in which the selection of the optimizer is crucial to the performance of the final model. The appropriate optimizer can train a network model with a more accurate segmentation effect [11]. In this study, the Adam optimizer was selected, which is one of the most commonly used optimizers in deep learning [12]. The optimizer is suitable for various types of problems, such as processing image classification, object detection, semantic segmentation, etc.

2.3 Dice coefficient

In this study, the self-defined Dice coefficient was used as the evaluation function. As a set similarity measure function, it could be used to calculate the similarity between two samples [13]. The formula used for calculation was as follows.

$$Dice(A,B) = \frac{2A \cap B}{|A| + |B|} \tag{1}$$

Where A is the artificially delineated vertebral muscle region, and B is the vertebral muscle region obtained by the algorithm segmentation. The value of the Dice coefficient was in the range of [0,1]. The closer the value was to 1, the more accurate the segmentation of the vertebral muscle part was [10].

3. EXPERIMENT

3.1 Datasets

The data used in this experiment were 150 CT images of the L1 segment of the lumbar spine obtained from the Department of Spinal Surgery of Shengjing Hospital of China Medical University. The study used the Keras framework of tf2.0 to conduct experiments, mainly using PyCharm to write codes such as network construction.

3.2 Image preprocessing

The CT image was composed of pixels of different gray levels arranged in order according to the matrix. The analysis of CT value expression revealed that the CT value was positively correlated with tissue density. Usually, the larger the CT value, the higher the image tissue density, and vice versa. The image information obtained using different CT devices was different. The image size varied from 1.0×1.0 mm2 to 0.5×0.5 mm, and the number of images was 256×256 and 512×512.

The data used in this experiment were the CT images of the L1 segment of the lumbar spine, which were in DICOM format. First, the ImageJ software was used to open the image that needed to be preprocessed. Then, the image was saved in PNG format to obtain a Python-readable image format, which was convenient for data input in the U-net network. Then, MATLAB was used to process the image pixel as 256×256 to realize the input of the whole CT image into the U-net network.

According to the experimental requirements, the mask image was obtained, which was mainly used to manually select and label the muscle area of the CT vertebral image for model training. The specific steps were as follows: First, open the image to be marked → irregularly select the area to be segmented → click Edit; then select Section → Select Create mask, and save it in PNG format.

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 256, 256, 1) 0
__________________________________________________________________________________________________
conv2d (Conv2D)                 (None, 256, 256, 16) 160         input_1[0][0]
__________________________________________________________________________________________________
batch_normalization (BatchNorma (None, 256, 256, 16) 64          conv2d[0][0]
__________________________________________________________________________________________________
activation (Activation)         (None, 256, 256, 16) 0           batch_normalization[0][0]
__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 256, 256, 16) 2320        activation[0][0]
__________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 256, 256, 16) 64          conv2d_1[0][0]
__________________________________________________________________________________________________
activation_1 (Activation)       (None, 256, 256, 16) 0           batch_normalization_1[0][0]
__________________________________________________________________________________________________
max_pooling2d (MaxPooling2D)    (None, 128, 128, 16) 0           activation_1[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 128, 128, 16) 0           max_pooling2d[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 128, 128, 32) 4640        dropout[0][0]
__________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 128, 128, 32) 128         conv2d_2[0][0]
__________________________________________________________________________________________________
activation_2 (Activation)       (None, 128, 128, 32) 0           batch_normalization_2[0][0]
__________________________________________________________________________________________________
```

Figure 2. Print U-net network structure diagram.

3.3 U-net network construction

According to the need of creating the U-net network, this study first introduced its definition uniformly and put the function in class so as to realize the flexible call of the function. First, the U-net network reuse convolution module was defined, and the function was placed inside the build_unet function. The convolution module of the network comprised the convolution layer, standardization layer, activation function, convolution layer, standardization layer, and activation function, and finally returned the result. Second, the convergence path was compiled. The main process was the sequential downsampling of the input image and the repeated module. The repeated module included the convolution module, the downsampling, and the dropout layer. The process was repeated four times to complete the downsampling. Then, the extended path compilation was performed. The main process was to input the output of the previous layer, followed by the sequential upsampling of the repetitive module. The repetitive module included upsampling, feature merging, dropout layer, and convolution module. Symmetric with the convergence path, it was divided four times to complete the upsampling. Finally, the output layer was defined according to the data format of the output image. After the aforementioned experiments, the function creation of the U-net network was completed, and the network was created by calling the function. After executing self.unet.summary ( ), the network structure was printed out, indicating the creation of the U-net network. A part of the network structure is shown in Figure 2.

3.4 Network training

First, the dataset was imported. The obtained pixel-processed data were stored in the file train for the preprocessed image of the lumbar L1 segment, and the mask image was stored in the file label. Before entering the U-net network, the image was called in the form of a file. Then, the training mode was set. The training process was monitored in real time. The validation set loss could be terminated in advance if it did not decrease 100 consecutive times. At the same time, the amount of training set data, the number of cycles, the size of each batch of cycles, and the minimum learning rate were set to retain the optimal model [9]-[10]. Finally, the training was started. In the experiment, the relevant parameters affecting the segmentation effect were continuously adjusted, and the segmentation result map was saved. While training the training set, a certain proportion of data was separated as the verification set; 0.1 was selected in this experiment. Training batches and each batch of training da ta could be set differently to find the optimal segmentation model.

3.5 Training parameter settings

The image processing performance of the network model is affected by many factors. This experiment mainly adjusted the amount of training set data, the number of cycles, the size of each batch of cycles, and the minimum learning rate to obtain a parameter combination with a better segmentation effect. The adjustment of parameters has different effects on the segmentation effect and network training speed. The learning rate represents the step size of the gradient moving to the optimal solution of the loss function in each iteration. Its size is proportional to the learning speed of the network. The setting of the minimum learning rate is of great significance to the model optimization and training speed.The training parameter setting was mainly divided into two stages. The first stage explored the parameter setting, and the second stage determined the optimal parameter combination and retained it for model training.

Table 1. Training parameter exploration table.

serial number	Loss	T/sample	Dice	set the cycle	actual cycle	each batch	gross data	training data	minimum learning rate
1	0.0052	1s83ms	0.8943	300	187	15	150	140	0.00005
2	0.0047	1s74ms	0.8809	250	218	15	150	135	0.00005
3	0.0053	1s77ms	0.8780	150	150	15	150	140	0.00005
4	0.0056	1s84ms	0.8993	260	260	20	150	140	0.00005
5	0.0061	2s119ms	0.8909	200	200	25	150	130	0.00005
6	0.0063	1s86ms	0.8851	200	200	20	150	140	0.0005
7	0.0056	1s73ms	0.8463	300	210	20	150	140	0.00005
8	0.0044	1s82ms	0.8821	200	190	20	150	140	0.00004
9	0.0045	1s65ms	0.9063	300	225	25	150	140	0.00004
10	0.0052	2s175ms	0.8880	200	200	30	150	140	0.00006

Note: The cycle is terminated when Loss is set to decline for 100 consecutive times.

The first stage of training-related data is shown in Table 1. The analysis could be concluded as follows.

(1) Available from 1 and 7, and 3 and 7, the smaller the data per batch, the larger the Dice, the larger the loss, and the faster the data training speed; (2) Available from 4 and 6, and 4 and 8, the smaller the minimum learning rate, the smaller the loss and the faster the data training speed; (3) Available from 1 and 2, and 5 and 9, the larger the amount of training

set data, the larger the Dice and the smaller the loss; (4) 1 and 3 show that the more the actual number of cycles, the larger the Dice, the smaller the loss, and the lower the data training speed.

In the second stage, the influence of each parameter on the Dice coefficient, loss value, and data training speed was comprehensively considered, especially the Dice coefficient. The minimum learning rate of 0.00004 in the first stage was selected for training. The adjacent values of 0.000035 and 0.000045 were taken for training, and 0.00004 was determined as the minimum learning rate. Finally, a Dice coefficient of 0.9178 could be obtained after continuous attempts, and the segmentation effect was good. Although the training speed under this parameter combination is slightly lower, its Dice coefficient and Loss data perform well and can meet the requirements of segmentation effect. For the sake of training speed, Table 2 (9) is also a good parameter selection. Although its Dice coefficient 0.9098 is slightly lower than the former, its loss and training speed are very noteworthy, the training speed reaches 65 milliseconds for a single image.

Table 2. Training parameter verification table.

serial number	Loss	T/sample	Dice	set the cycle	actual cycle	each batch	gross data	training data	minimum learning rate
1	0.0048	1s109ms	0.9178	300	205	15	150	140	0.00004
2	0.0051	88ms	0.9145	200	188	15	150	145	0.00004
3	0.0049	1s72ms	0.8945	300	171	10	150	140	0.00004
4	0.0048	75ms	0.9039	200	200	15	150	145	0.00003
5	0.0046	1s127ms	0.8963	300	225	15	150	140	0.00003
6	0.0040	1s76ms	0.9043	300	204	15	150	140	0.000045
7	0.0051	72ms	0.8972	300	220	20	150	145	0.00004
8	0.0044	1s71ms	0.8832	300	300	15	150	140	0.000035
9	0.0041	65ms	0.9098	200	187	15	150	145	0.00004
10	0.0071	2s118ms	0.8509	300	197	15	150	130	0.00004
11	0.0047	1s77ms	0.9099	300	189	15	150	140	0.00004
12	0.0037	1s72ms	0.9047	300	186	15	150	140	0.00004
13	0.0041	1s124ms	0.8991	300	195	15	150	140	0.00004
14	0.0045	1s85ms	0.8961	300	224	15	150	140	0.00004
15	0.0049	1s84ms	0.8874	300	300	20	150	140	0.00004

Note: The cycle is terminated when Loss is set to decline for 100 consecutive times.

4. RESULTS

4.1 Loss and metric curve

The training set loss and metric of each epoch, as well as the val _ loss and val _ metric of the validation set, were completely saved. Figure 3 shows the final curves drawn. Among these, loss corresponded to the loss function value, and metric corresponded to the Dice coefficient. As shown in Figure 3, the loss of the validation set was always larger than that of the training set, and the Dice coefficient was smaller than that in the training set. In this experiment, val _ loss did not decrease 100 consecutive times, and hence the training was terminated in advance at the 205 epoch.

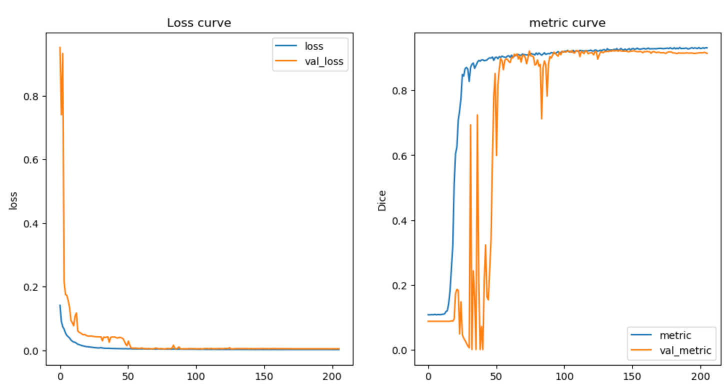

Figure 3. Loss curve and metric curve in the training process.

4.2 Split effect diagram

As shown in Figure 4, the first column is the input of the U-net network, that is, the pre-processed image; the second column is the output of the network, that is, the segmentation result graph based on the U-net network; the third column is a manually sketched mask, that is, the muscle area of the selected vertebral image is identified by the naked eye. As shown in Figure 5, (a) is the segmentation effect map obtained by the deep learning segmentation algorithm in the test set, and (b) is the segmentation effect map manually sketched.

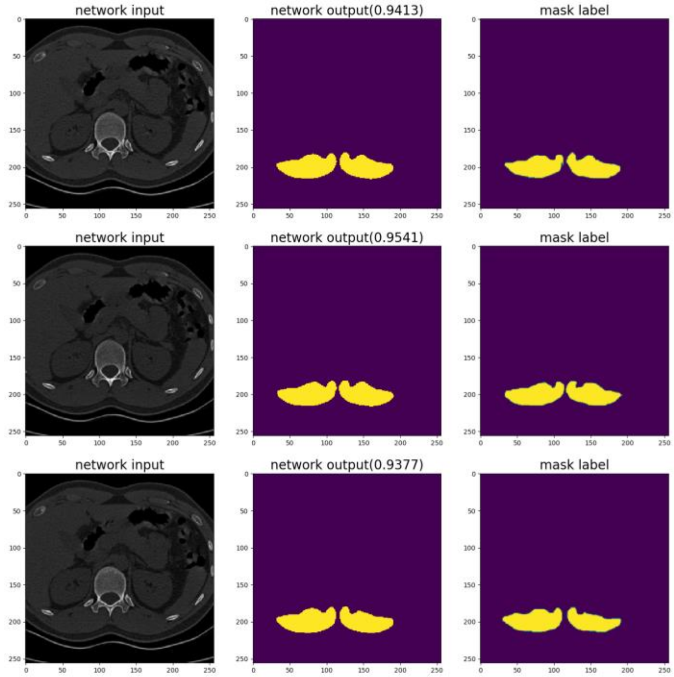

Figure 4. Network segmentation effect diagram of vertebral muscle.

(a) network predictive

(b) manual segmentation

Figure 5. Network prediction and artificial segmentation result graph.

4.3 Test evaluation

```
10/10 [==============================] - 1s 109ms/sample - loss: 0.0048 - metric_fun: 0.9178
schedule: 0/10
schedule: 1/10
schedule: 2/10
schedule: 3/10
schedule: 4/10
schedule: 5/10
schedule: 6/10
schedule: 7/10
schedule: 8/10
schedule: 9/10
```

Figure 6. Test set Dice coefficient results.

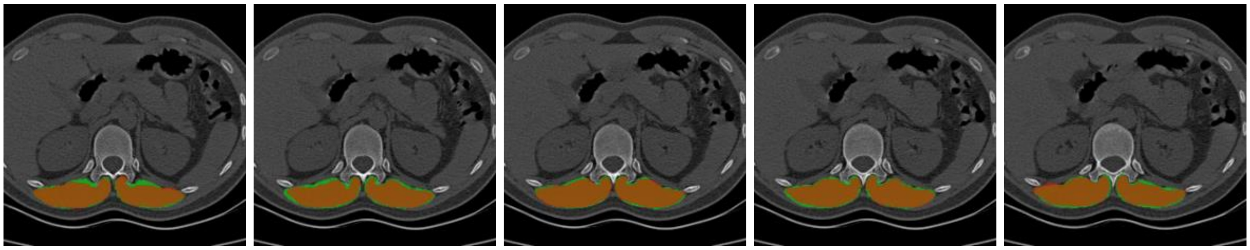

Figure 7. Comparison of network prediction and artificial segmentation: The green area is the muscle part of the lumbar L1 segment segmented by the deep learning algorithm, the red area is the artificially delineated muscle part of the lumbar L1 segment, and the orange area is the overlapping part of the two.

In this study, the self-defined Dice coefficient is used as the evaluation function, and the segmentation effect is evaluated by combining the segmentation algorithm with the manual sketch mask comparison map. As shown in Figure 6, the average Dice coefficient in this study reached 0.9178 in 10 test sets. This result indicated that the muscle region segmented by the network was extremely similar to the artificial mask. Both the data results and the segmentation effect map showed that the segmentation model based on the U-net network was quite effective. In Figure 7, according to the experimental principle, the larger the proportion of overlapping parts, the better the segmentation effect of the algorithm.

5. DISCUSSION AND OUTLOOK

5.1 Summary and discussion

This study proposed and implemented an automatic segmentation algorithm of vertebral muscles in CT images based on the U-net convolutional neural network. First, the original medical image was preprocessed to obtain an image that could adapt to the input format of the network. Then, the processed image was divided into training and test sets according to the proportion and input into the U-net network for training. A certain proportion was randomly selected from the training set as the verification set. Finally, the vertebra muscle segmentation effect map was obtained based on CT images. In this study, the average Dice coefficient of the segmentation of the region of interest reached 0.9178, indicating that the segmentation method based on the U-net neural network could effectively segment the muscle part in the vertebral image. Compared with the traditional segmentation method, the proposed method had many advantages including efficient and accurate segmentation and involvement of less time and labor. In recent years, deep learning has developed rapidly. Also, its application in medical image processing is extensive, especially for medical image segmentation. In practical work, image segmentation technology can assist clinicians in making more accurate diagnoses even for small lesions that are difficult to distinguish by the naked eye. The research and implementation of muscle segmentation in vertebral CT images is of far-reaching significance for diagnosing and treating spine-related diseases. The disease diagnosis and prognosis can be achieved to a certain extent by observing and analyzing muscle morphology and composition [14]. Although the U-net network is effective in image segmentation applications, its shortcomings cannot be ignored. The findings of the present study indicated that the following two aspects still need improvement. First, the valid convolution in the network reduces the universality of the model, and hence a more suitable convolution module is urgently needed for the model. Second, the form and feature map through trimming are not symmetrical, and perhaps the effect of bilinear interpolation is better.

5.2 Prospects

In this study, the U-net network was built using Pycharm and Keras framework in the tf2.0 environment, and the related experiment on muscle region segmentation in vertebral CT images was carried out. Although the segmentation effect of the muscle region was good, the optimization and exploration of its segmentation algorithm need further investigation. One concern was the reasonable setting of the minimum learning rate. This study selected 0.00004 between [0.000035, 0.000045] and found that its segmentation effect was better. Based on this, it was speculated that the range could be further narrowed to explore the minimum learning rate with a better segmentation effect. The second was the increase in attention mechanism. The bottom-up feature fusion method of the U-net network itself determined its neglect of low-level features [15]. In the follow-up studies, we plan to increase the attention mechanism in the U-net network to more accurately lock the region of interest, thereby improving the accuracy of vertebral muscle segmentation. The third was the control of network training speed. The experiment terminated in advance under the limitation of the loss condition setting of the verification set. Therefore, a lower epoch can be considered to shorten the whole training process.

In addition, exploring the application of the U-net network in medical image segmentation is of great significance. The automatic measurement and analysis of CSA, paraspinal muscle density, and fat infiltration degree will be further realized in future studies on the basis of muscle segmentation, so as to provide a more accurate grasp of the rehabilitation effect of spinal diseases. With the continuous development of deep learning, its application in the field of medical image segmentation has significantly improved, and is of great significance to clinical work. However, it focuses only on the changes in preoperative indicators. Therefore, further studies are required to investigate the changes and functions of postoperative paraspinal muscles [16].

ACKNOWLEDGEMENTS

This research was funded by The University Basic research project of The Educational Department of Liaoning Province (Project JYTMS20231373), Natural Science Foundation of Liaoning Province (Project 2022-KF-12-03), Rolling support of Shenyang Pharmaceutical University Young and Middle-aged Teachers' Career Development Support Plan (Project ZQN2019019), Shenyang Pharmaceutical University 2024 Innovation and Entrepreneurship Training Program for College Students (Project Chenkai Song), National Training Program of Innovation and Entrepreneurship for Undergraduates (Project Chenkai Song).

REFERENCES

[1] Liao Y., Zhu M., Zhang D., "Neuropathic low back pain mechanism and diagnosis and treatment progress," Chinese Journal of Pain Medicine, Papers 27(12), 930-933 (2021).
[2] Liu C., Chen K., Li B., "Advances in paraspinal muscle degeneration and its relationship with degenerative scoliosis," Chinese Journal of Spinal Cord, Papers 33(11), 1047-1051 (2023).
[3] Wang W., Li W., Chen Z., "Research progress on evaluation methods of paravertebral muscles and their correlation with lumbar diseases," Chinese Journal of Repair and Reconstruction Surgery, Papers 34(11), 1462-1467 (2020).
[4] Zhou L., Zhu X., "Research on lung tumor image segmentation algorithm based on U-net network, " Information and computer (theoretical version), Papers (05), 41-44 (2018).
[5] Yang Jie., "Application of deep learning method with active contour constraint in medical image segmentation," Hebei: Hebei University, (2023).
[6] Yun F., Yin Y., "Multi-scale liver CT image segmentation based on improved Attention-Unet," Journal of Inner Mongolia Normal University (Natural Science Chinese Edition), Papers 52(02), 175-180 (2023).
[7] Zheng F., "Application of Deep Learning in Image Recognition and Analysis of Medical Archives," Information System Engineering, Papers (03), 142-145 (2024).
[8] Xu J., Hu T., Liu Y., "Design and implementation of rectal cancer CT image segmentation system based on U-Net," Computer knowledge and technology, Papers 19(32), 8-10 (2023).
[9] Zhang B., Xiao J., "Optimizer experimental design for deep learning model training," Electronic production, Papers 32(02), 114-117 (2024).
[10] http://www.tipdm.org/bdrace/tzjingsai/20181226/1544.html
[11] Ma X., Du X., Cai Q., "A Review of Deep Learning Framework Testing Research," Journal of Software, Papers 05(16), 1-33 (2024).
[12] O. Ronneberger, P. Fischer and T. Brox., "U-net: Convolutional networks for biomedical image segmentation", Computer Science, Papers 42, 234-241 (2015).
[13] Liu X., Yu H., Li B., "Vertebral CT image segmentation method based on improved U-Net model," Journal of Harbin University of Science and Technology, Papers 26(03), 58-64 (2021).
[14] Jiang B., Liu J., Qiu D., "An overview of the application of deep learning in spinal image segmentation," Computer Engineering, Papers 50(03), 1-15 (2024).
[15] Liu B., "Research on medical image segmentation and application based on deep learning," Shandong: Yantai University, 2024.
[16] Sun D., Liu P., Cheng J., "Correlation between intervertebral disc degeneration, paraspinal muscle atrophy, and lumbar facet joints degeneration in patients with lumbar disc herniation," BMC Musculoskelet Disord, Papers 18(1), 167 (2017).