BMC Medical Imaging

RESEARCH

Open Access

Accelerated muscle mass estimation from CT images through transfer learning

Seunghan Yoon[1], Tae Hyung Kim[2], Young Kul Jung[3*] and Younghoon Kim[1*]

Abstract

Background The cost of labeling to collect training data sets using deep learning is especially high in medical applications compared to other fields. Furthermore, due to variances in images depending on the computed tomography (CT) devices, a deep learning based segmentation model trained with a certain device often does not work with images from a different device.

Methods In this study, we propose an efficient learning strategy for deep learning models in medical image segmentation. We aim to overcome the difficulties of segmentation in CT images by training a VNet segmentation model which enables rapid labeling of organs in CT images with the model obtained by transfer learning using a small number of manually labeled images, called SEED images. We established a process for generating SEED images and conducting transfer learning a model. We evaluate the performance of various segmentation models such as vanilla UNet, UNETR, Swin-UNETR and VNet. Furthermore, assuming a scenario that a model is repeatedly trained with CT images collected from multiple devices, in which is catastrophic forgetting often occurs, we examine if the performance of our model degrades.

Results We show that transfer learning can train a model that does a good job of segmenting muscles with a small number of images. In addition, it was confirmed that VNet shows better performance when comparing the performance of existing semi-automated segmentation tools and other deep learning networks to muscle and liver segmentation tasks. Additionally, we confirmed that VNet is the most robust model to deal with catastrophic forgetting problems.

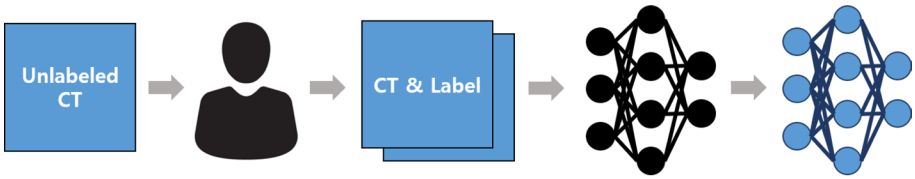
Conclusion In the 2D CT image segmentation task, we confirmed that the CNN-based network shows better performance than the existing semi-automatic segmentation tool or latest transformer-based networks.

Keywords Medical image segmentation, CT image segmentation, Deep learning, Convolutional neural network

*Correspondence:
Young Kul Jung
free93cool@gmail.com
Younghoon Kim
nongaussian@hanyang.ac.kr
[1] Department of Computer Science & Engineering (Major in Bio Artificial Intelligence), Hanyang University at Ansan, 55, Hanyangdaehak-ro, Sangnok-gu, 15588 Ansan-si, Gyeonggi-do, Republic of Korea
[2] Division of Gastroenterology and Hepatology, Hallym University Sacred Heart Hospital, 22, Gwanpyeong-ro 170beon-gil, Dongan-gu, 14068 Anyang-si, Gyeonggi-do, Republic of Korea
[3] Division of Gastroenterology and Hepatology, Department of Internal Medicine, Korea University Ansan Hospital, 123, Jeokgeum-ro, Danwon-gu, 15355 Ansan-si, Gyeonggi-do, Republic of Korea

Introduction

Automatic medical image segmentation has long been a research topic for a long time because organ labeling consumes a lot of time and effort from experts [1]. After the development of UNets [2], similar networks have been presented, and their performance has improved steadily. In addition, public datasets have been released for research, serving as benchmarks to compare performance in an equivalent environment [3, 4]. However, the data is only helpful if there is a label on the part to be segmented in the public dataset. In the actual clinical model deployment process,



(a) Existing neural network training process. Users perform labeling on unlabeled CT images and train the network

(b) Neural network training process using transfer learning. Users can create prelabels from pretrained networks and modify them to perform labeling efficiently.

Fig. 1 Comparison of model training process according to whether transfer learning is used

data may be collected from different institutions using varying imaging protocols and scanner suppliers. Since their data distributions do not match, performance degradation occurs during testing [5]. Furthermore, research is underway to enhance the accuracy of models by integrating CT images and clinical demographics in the machine learning process [6]. Even if a model trained from images taken by a specific device is distributed, a user cannot use it as it is. Briefly, deep learning-based segmentation technology is challenging to use in many institutions because it requires a large amount of learning data.

In this study, we present a learning strategy using transfer learning to alleviate practical difficulties in training a CNN-based neural network to perform segmentation on the muscle and liver in 2D CT images collected of various organs. There are three things we want to confirm through transfer learning. First, we check whether the performance can converge when we perform transfer learning using a small number of data on a model trained with data from different devices. Second, by observing the learning curve according to whether or not transfer learning was performed, it was confirmed that the performance converged faster, and higher division performance could be achieved. Third, after performing transfer learning, we checked whether catastrophic forgetting occurs on the previously learned dataset. Experiments were performed on CNN-based networks, and the latest transformer-based models and their performances were compared.

In the muscle segmentation task, we compare the performance of deep neural networks with the existing popular segmentation tool, BMI TOOL [7][1] additionally, which does not require training. Existing tools can segment muscle and abdominal fat without needing a learning process. However, segmentation can be performed on only one CT image at a time, and Hounsfield Units for focusing on muscle and fat must be entered each time the operation is performed. In addition, it takes a significant amount time to perform segmentation because a dividing line must be drawn manually to distinguish the inside and outside of the abdomen. Finally, since the output result cannot be saved, the work has to be performed again whenever the segmentation result is needed, even for an image that has already been segmented. After training, artificial neural network models are competitive with existing technologies because they can quickly perform segmentation on multiple images and store the output results.

The difference in the learning process depending on whether transfer learning is performed is shown in Fig. 1. When transfer learning is used, we could improve the performance of a neural network model with only a small amount of data taken from other devices. In addition, the annotator can save time and effort by using the model's output before transfer learning as the prelabel.

The goal of segmentation tasks in medical imaging includes studying anatomical structure, identification of regions of interest, measurement of tissue volumes, and assisting in treatment planning prior to radiation therapy [8]. However, pixel-level annotation is a laborious and costly task requiring trained clinicians [9]. The deep learning model training strategy proposed in this study

[1] https://sourceforge.net/projects/muscle-fat-area-measurement/

can help reduce the labeling costs required for model training.

The contribution of our study is summarized as follows:

- We collected data for muscle segmentation, trained the CNN-based model VNet, and compared the performance with a baseline that did not require training, and the latest transformer-based networks.
- We measured the model's performance on CT images taken by other devices and observed the performance improvement while additionally training the data.
- We performed transfer learning on several datasets for the liver segmentation task and observed the learning curve to confirm the advantages of transfer learning.
- After transfer learning, we observed how much the performance changed for the previously learned data.
- Through our experimental results, we demonstrate the potential to reduce labeling costs by utilizing a pretrained network, as shown in Fig. 1, to generate pseudo labels.

The remaining sections of this paper consist of a total of five sections. "Related work" section provides an overview of related studies in the field of medical image segmentation using deep learning. "Automatic CT image labeling for organ segmentation by transfer learning" section describes the method employed for muscle labeling in CT data. Furthermore, "Experiment settings" section elucidates the configurations for experiments, including datasets, deep learning models, and evaluation metrics used in transfer learning experiments. The experimental results are then discussed in "Experimental results" section. Finally, the findings of this study are summarized in the last section.

Related work

This section covers research related to medical image segmentation. First, we deal with deep learning-based medical image segmentation methods. Second, studies that applied transfer learning in medical image segmentation tasks are introduced.

Medical image segmentation

Manual segmentation requires outlining structures slice by slice and visual inspection, which is mentally demanding and a highly time consuming process [10]. Many studies have been conducted to perform segmentation in medical images using deep learning to overcome these disadvantages [2, 11]. Among them, UNet is a representative network. After the success of UNet on medical image segmentation tasks, many studies have attempted to utilize its architecture or applying additional techniques to improve performance [12, 13]. Zhou et al. [12] proposed UNet++ network modified from the UNet network and demonstrated higher segmentation performance than UNet in cell nuclei, colon polyp, liver, and lung nodule datasets through experiments. Huang et al. [13] proposed UNet 3+ network to reduce the number of parameters of UNet++ and improve segmentation accuracy. The experiment showed better performance than previous studies in the liver segmentation task using the LiTS 2017 dataset.

There is also a semi-automatic segmentation method that does not require machine learning [7]. This method is difficult to perform with a large amount of segmentation because a person must directly mark the ROI for each slice of the CT image. However, it has the advantage of not requiring a large amount of training data. Kim et al. [14] used it as a tool to measure the skeletal muscle index (SMI) in the L3 region of the spine.

Neural networks for muscle segmentation on medical images

Kanavati et al. [15] detected a slice near the L3 vertebra using a UNet-like network structure, and segmented the muscle using manually labeled CT images. Similarly, Edwards et al. [16] performed muscle segmentation from CT data of 33 adult patients. Some studies segmented muscles by simultaneously segmenting multiple CT slices or inputting 3D CT images instead of segmenting a single image. 2.5D CNN automatically searches for skeletal muscles near L3 in 3D CT images and performs muscle segmentation in [17]. The 2.5D CNN receives three images simultaneously as input, including the CT slice adjacent to the L3 region. In [18], 3D UNet was used to confirm the segmentation performance on several images. There is also a study that labels muscle, subcutaneous fat, and visceral fat and trains a deep-learning model [19, 20]. In addition, Castiglione et al. [21] collects and learns CT data of children to diagnose sarcopenia. There are also studies proposing a network structure that is not based on UNet to segment muscles at the L3 and T4 levels [22]. Lee et al. [23] also proposed a CNN structured network rather than UNet, but performs segmentation only at the L3 level of the spine. Fu et al. [24] used a CNN-based network to segment the abdominal cavity and the muscles through post-processing. There are also studies that segmented muscles from MRI images [25, 26]. In [25], labeling is performed on 13 types of muscles in MRI images ,and CNN-based network segmented them. Li et al. [26] segmented multifidus and erector spinae using deformed UNet. In [27], segmentation of muscle, subcutaneous fat, and abdominal fat as well as abdominal region prediction were performed using

UNet. Nishiyama et al. [28] used a generative adversarial network(GAN) based model to train it to produce realistic segmentation results.

Research is also underway to provide insights into building reliable artificial intelligence systems in healthcare systems. Albahri et al. [29] systematically categorize various studies aimed at constructing trustworthy artificial intelligence in healthcare systems and investigate the feasibility of explainable AI (XAI) application. Alzubaidi et al. [30] address strategies to tackle the problem of data scarcity.

Research aiming to enhance the performance of deep learning networks in the medical field also exists. Shamrat et al. [31] improved various networks for classifying lung diseases through transfer learning. Sutradhar et al. [32] proposed a method to increase the classification accuracy of thyroid diseases using ensemble techniques. Shamrat et al. [33] proposed a modified network called AlzheimerNet, based on the InceptionV3 network [34] to improve the classification accuracy of disease of alzheimer stages and normal control classes.

Recently, research has been conducted to enhance performance by combining transformer and CNN network architectures. Sun et al. [35] proposed a network structure combining CNN and transformer, demonstrating high performance in the task of segmenting organs from abdominal CT images. Heidari et al. [36] introduced Hiformer, which combines CNN and transformer, showing superior performance in organ segmentation from abdominal CT images and skin lesion segmentation tasks. Li et al. [37] proposed ATTTransUNet, a combination of transformer and CNN, exhibiting high segmentation performance on three types of medical image datasets, including thyroid ultrasound (ThyroidUS). Yang et al. [38] proposed CSwin-PNet, connecting CNN and Swin Transformer, and demonstrated high performance in breast lesion segmentation tasks from ultrasound images. Kawamoto et al. [39] proposed a site-specific 3D segmentation method of skeletal muscle in L3 slices from CT images. Kamiya et al. [40] discuss the necessity of musculoskeletal analysis and the necessary image processing techniques and introduce deep learning-based segmentation techniques from CT images. Ashino et al. [41] improved performance in the segmentation of the sternocleidomastoid and other skeletal muscles by using multiclass learning.

Transfer learning for medical imaging

Collecting training data is particularly challenging when training deep learning models on medical images. Many studies have conducted transfer learning to solve this problem. Chen et al. [42] pretrained a CNN-based model on multiple medical image datasets and then performed transfer learning with a small number of labeled images to segment the lungs and liver. The model achieved better performance through transfer learning. In [43], transfer learning was performed with both identical and non-identical domain data respectively for the diabetic foot ulcer (DFU) classification task. The experiments showed that transfer learning with identical domain data significantly improves performance. Raghu et al. [44] showed that the performance improvement is not significant when transfer learning is performed on medical datasets using networks pretrained on ImageNet dataset. To obtain a model trained with the same domain data while reducing the labeling effort, Alzubaidi et al. [45] trained a model with unlabeled medical image data. Then, transfer learning was performed with a few labeled images. Also, transfer learning was conducted for the red blood cell classification task in [46]. The performance of the model was improved by performing same domain transfer learning. Heker et al. [47] conducted transfer learning using the ImageNet and LiTS datasets for deep learning models that perform classification and segmentation tasks. Experiments showed that models pretrained with the same domain data (LiTS dataset) show higher performance after transfer learning. Unlike previous studies, our research performs transfer learning with various deep learning models and datasets, comparing the training process and performance across different aspects. We also compare the results of segmentation tasks from semi-automatic segmentation tools and deep learning models in terms of segmentation speed and performance.

Automatic CT image labeling for organ segmentation by transfer learning

First, we segment the muscle on CT images using a CNN-based model and the BMI TOOL, which does not require training, and compare their performances. Then, we check whether performance can be improved with fewer data through transfer learning. To do this, we label muscles in the BTCV dataset. Then training was performed while increasing the training data in the fully initialized and pretrained models, and the performance was observed. In addition, we quantitatively compared the performance with the latest transformer-based models to confirm that the CNN-based model is competitive.

In addition, we conducted experiments on scenarios in which transfer learning was performed by collecting data from various institutions. Transfer learning was conducted with three public datasets labeled for the liver, and the learning curve and final performance were confirmed. Then, by checking the performance on the untrained dataset, the possible reusability of the model is explored. Finally, when a model trained on a specific dataset was transfer-learned to another dataset,

we observed a change in performance on the previously trained data. This section explains the methods used in the experiments for each task.

Labeling tool to collect training data

To proceed with labeling manually, we created an application that allows manual labeling in the Windows operating system. This program displays CT images with overlapping labels. Since the label is shown as translucent, it is possible to work while checking which area of the CT image has been labeled. Users can mark muscle, subcutaneous fat, and visceral fat by zooming in and moving the photo. In this application, muscle tissue is expressed in green, subcutaneous fat in red, and visceral fat in blue. Figure 2 shows how to use the labeling tool we made.

The manufactured labeling tool requires a CT image and a prelabel image file. The CT image and the prelabel are overlapped, and the prelabel is translucent, so a user can work while checking the contour of the CT image behind it. Enlarging or reducing the image by scrolling with the mouse is possible. Moreover, the labeling tool can work in units of files and units of folders. In the file unit operation, a user can select one CT image and a prelabel image file and then save the corrected label image. For folder unit work, folders containing CT images, prelabels, and modified label images in one folder should be located in subfolders, respectively. Also, CT images and prelabel images should be prepared in the same order and the same number. In that state, if a user selects a folder in the labeling tool, the first CT and prelabel images are displayed on the screen. Suppose a user clicks the button to work with the next file after working with the previous one. In that case, the label file a user is working on is saved, and the next image is output so that a user can perform the work sequentially.

VNet network

We trained the currently popular VNet [48] CNN-based network for muscle segmentation on CT images which can approve baseline weakness. VNet is a neural network model composed of an encoder-decoder architecture with a structure similar to UNet. The architecture of the VNet is shown in Fig. 3. The encoder-decoder extracts the feature map by performing downsampling and upsampling four times and determines the class belonging to each pixel by applying the softmax function on the output of the last layer. Like UNet, a residual connection directly connects the encoder output to the decoder. Such an architecture reduces information loss due to multiple downsampling, enabling more accurate segmentation. VNet was originally studied to perform segmentation on 3D MRI data. However, in this study, all convolution layers were modified to be two-dimensional to perform segmentation on 2D CT data. The model's input is a size 512 * 512 grayscale image, and the model's output is the same size. In VNet, the authors proposed a dice loss function and the network structure for performing segmentation on 3D images. Dice loss is effective when training a model that segments a target that occupies a small portion of the entire image. However, there was little difference between dice loss function and cross-entropy since

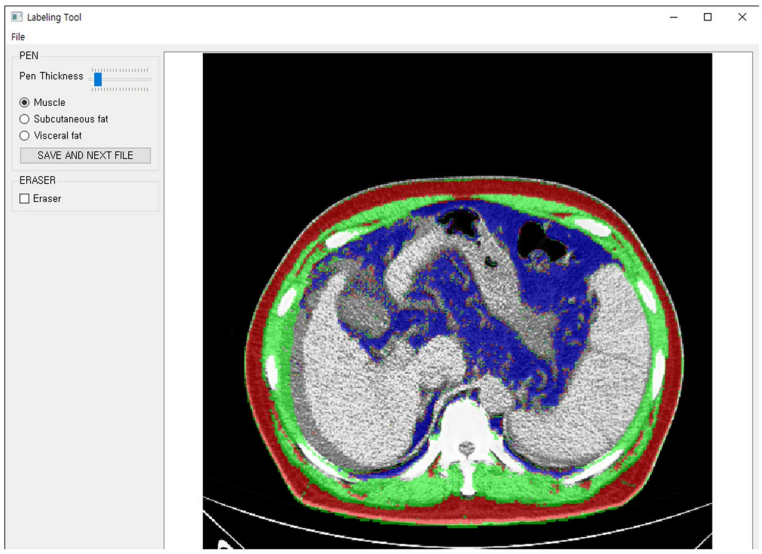


Fig. 2 Execution screen of proposed labeling application. Users can edit CT and prelabel images after uploading them. We marked muscles in green, subcutaneous fat in red, and visceral fat in blue

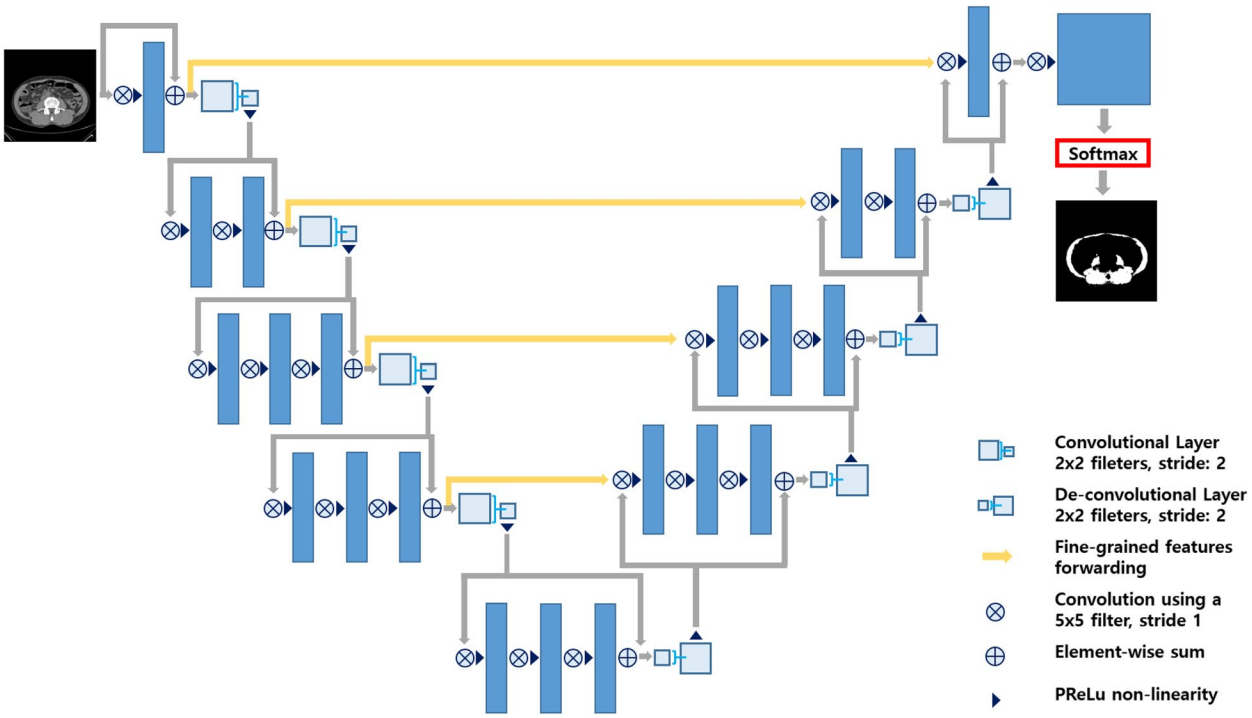


Fig. 3 The architecture of VNet network

muscles are distributed over a large area on CT images. In this study, we used cross-entropy as the loss function.

Experiment settings

This section explains the performed experiments, datasets we used, implemented models, and evaluation metrics.

Baseline and implemented models

We used BMI TOOL as the baseline to compare performance with the neural network model. BMI TOOL was developed for the semi-automatic segmentation of subcutaneous fat, visceral fat, and muscle in CT images. Users can segment muscle and fat by uploading a single Dicom file to the application. First, the preprocessing step removes the background image from the CT image. Secondly, the boundaries between muscles and internal organs are distinguished in the boundary step. BMI TOOL transforms the initial curve manually drawn by a user using the active contour method [49]. Finally, subcutaneous fat, muscle, and visceral fat are detected in the preprocessed CT images in the identification step. Then, it calculates the BMI index by multiplying the number of pixels for each type by the pixel surface area value to obtain the area. We show the execution screen of BMI TOOL in Fig. 4.

The baseline technique has the advantage of not requiring a learning process and can segment muscle and fat. Still, it is difficult to use to perform tasks on many CT images. First, the baseline can only segment one CT image at a time. It is necessary to input the Hounsfield Unit for each image and draw a line to distinguish subcutaneous and visceral fat. As a result, it takes at least 2-3 minutes to perform segmentation on one CT image. Moreover, because the baseline cannot save the output, a user has to perform segmentation again whenever it is needed. Finally, the baseline will only measure correctly if the split line is drawn correctly because the user cannot modify it.

UNETR [50] is a network developed to perform 3D medical image segmentation tasks. UNETR differs from existing UNet-based networks in that the transformer architecture is used as an encoder. However, it shares similarities in that the encoder delivers outputs to the decoder and has a "U-shaped" structure. The encoder consists of 12 transformers. An input image and the encoder outputs of the 3rd, 6th, and 9th encoder are delivered to the decoder. In this study, 2D CT images were used for muscle and liver segmentation tasks to test the segmentation performance in 2D data.

Swin-UNETR [51] is a network based on the swin transformer developed to compensate for the vision transformer's shortcomings that follow the transformer's

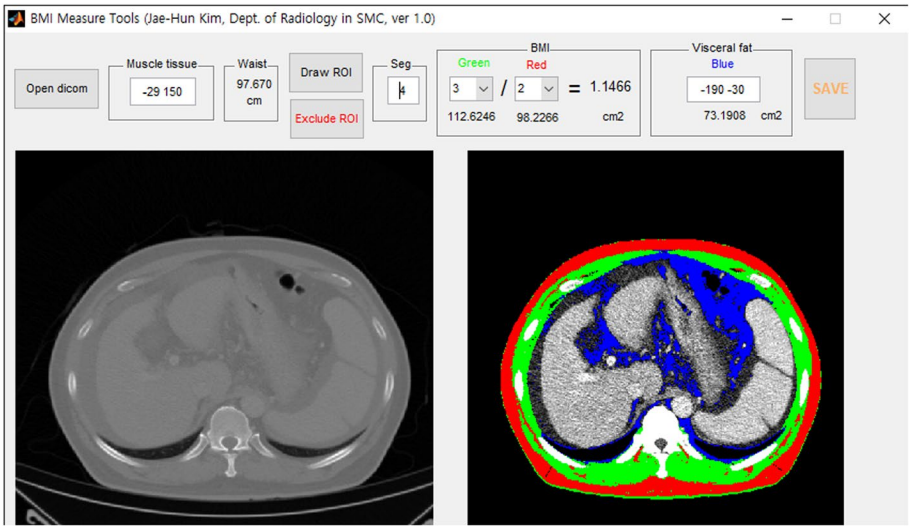


Fig. 4 Execution screen of BMI TOOL. A user must input the Houndsfield Unit values for fat and muscle, and draw an ROI to separate the inside and outside of the abdomen after uploading the CT image in Dicom format. The image on the right shows the segmentation result

structure for natural language processing. The swin transformer is a network suitable for computer vision work with fewer calculations than the existing vision transformer. After performing the convolution operation, the input image and the output of the encoder are connected to the decoder. The encoder consists of four swin transformer blocks, and the decoder consists of CNNs.

ParaTransCNN [35] achieved high performance in organ segmentation tasks on abdominal CT images by combining CNN and Transformer architectures. It consists of an encoder composed of transformers and an encoder composed of CNNs arranged in parallel. The outputs of both encoders are passed through a channel attention module to the decoder. For ParaTransCNN, the input image size was adjusted to 224*224 to utilize the pretrained ResNet34 encoder.

VNet utilizes all convolutional layers within the encoder and decoder as 2D convolutional layers. On the other hand, UNETR and Swin-UNETR were adjusted to receive two-dimensional patches from the transformer encoder, and then constructed the decoder's convolutional layers as 2D convolutional layers.

Dataset

Training data collected for muscle segmentation

The data set collected for learning consists of 6 CT data sets, each of which differs in whether one patient used the contrast agent and the shooting time. Among them, we used five sets of CT data as the training dataset and the remaining one as the test dataset. For labeling, first, the output result of the existing BMI TOOL was captured and used as a prelabel. Then, an expert manually modified the prelabel and constructed the ground truth data using the suggested labeling tool.

Additionally, the BTCV dataset[2] was used for performing transfer learning and measuring performance. Since the data set is 3D CT data and there is no label for muscles, the image was sliced to make 2D data and labeled with BMI TOOL. For two CT images, labeling was performed using BMI TOOL, transfer learning was performed using the first CT image, and performance was measured using the second CT image as test data.

Datasets for observing effects of transfer learning

We trained the models on three datasets and observed their performance. All datasets are 512 * 512 in size when converted to 2-dimensional data. Preprocessing of CT data involved clipping the images from DICOM data within the range of [-175, 250] based on HU(Houndsfield Unit) values, followed by rescaling the pixel values to the range of [0, 1]. When conducting muscle segmentation experiments, the experiments were conducted using the original DICOM data size (512 * 512). However, paratransCNN conducted experiments by resizing CT images to 224 * 224 to utilize the CNN encoder, ResNet34, within the model.

For liver segmentation experiments using LiTS, BTCV, and Chaos datasets, experiments were conducted after resizing the data size to 128 * 128 to facilitate efficient training for various learning scenarios.

[2] https://www.synapse.org/#!Synapse:syn3193805/wiki/217789

LiTS dataset The LiTS dataset [3] consists of 201 sets of CT data, of which 131 sets can be used publicly. In the CT images, labeling is performed for liver and liver tumors. The image data was collected from seven clinical sites all over the world. In this study, a model was trained using only liver labels to perform transfer learning on the liver segmentation task. We used 118 sets as training data and 13 sets as test data. The capacity of the LiTS dataset is about 50 GB after decompression.

The dataset is provided in the 3D NIFTY format. Since we had to perform segmentation on a 2D CT image, we extracted pixel data and converted it into 2D NumPy(.npy) data. The total number of converted data is 58,638. We used 52,188 data as training data and 6,450 as test data.

BTCV dataset The 30 sets of 50 CT images can be used as training data in the BTCV dataset. The remaining 20 sets are test data, and their labels have not been revealed publicly. People with expertise conducted manual labeling on 13 types of organs in the abdomen. We extracted only the liver label and trained deep-learning models in this study. Among the total dataset, we used 26 sets as training data and the others as test data. The size of the used BTCV dataset before resizing is about 1GB. This dataset was collected at Vanderbilt University Medical Center (VUMC).

Because the BTCV dataset is provided in 3D NIFTY format, we converted it into 2D data for our use. Among the total of 3,779 data, we used 3,295 for training and 484 for testing.

Chaos dataset We used CT data and labels for segmentation from the Chaos dataset [4], which consists of 20 sets of CT data. We used 17 sets as training data and 3 sets as test data. Among the Chaos datasets, the size of the dataset we used is about 1GB. The Chaos dataset was collected from the Department of Radiology, Dokuz Eylul University Hospital.

The dataset consists of 2,874 data in 2D DICOM (.dcm) format. We changed only the data format to NumPy for our use. Of the total data, we used 2,568 for training and 306 for testing.

Evaluation metrics

To compare the performance of the VNet and baseline, we used the dice score as an evaluation scale. The expression for the dice score is as follows,

$$DICE = \frac{2(P \cap G)}{|P| + |G|} \tag{1}$$

where $|P|$ and $|G|$ mean predicted pixels and ground truth pixels, respectively. The more the model's prediction matches the ground truth, the higher the dice score.

In addition, we measured accuracy and precision. Measurements should be calculated after obtaining the values of TP, TN, FP, and FN. TP and TN signify true positive and true negative, respectively. The two values represent correctly predicted pixels. False positive and false negative refer to pixels incorrectly predicted as positive and negative, respectively. The equations of accuracy and precision are as follows. Accuracy represents the number of correctly segmented pixels out of the total number of pixels that have been segmented.

$$Hausdorff\ \ Distance = \max\{\max_{x\in X}\min_{y\in Y} d(x,y), \max_{y\in Y}\min_{x\in X} d(x,y)\} \tag{2}$$

Hausdorff Distance refers to the maximum distance among the shortest distances from points in one set to the nearest points in another set. It is mathematically expressed as shown in Eq. 2. In this study, we calculate the Hausdorff Distance between the predicted segmentation region and the ground truth region to further assess segmentation accuracy. To mitigate the influence of outliers, we use the 95% Hausdorff Distance.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

Accuracy is not good as a measure of performance when the number of pixels corresponding to the target class is unbalanced in the segmentation operation. This is because even if all data is predicted as true or false, the performance will be measured as high. So, to overcome this, precision was additionally measured.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Precision is the percentage of the number of pixels predicted to be true that are actually true. In this study, the accuracy value can be high even if the muscle is not appropriately segmented because the number of muscle pixels is fewer than background pixels. Therefore, it is necessary to correctly count the number of pixels divided into muscle.

Transfer learning on multiple datasets

Transfer learning aims to extract knowledge from one or more source tasks and apply that knowledge to a target task. In this study, there are two main experiments conducted using transfer learning. First, to segment

muscles, labeling was performed on CT images through BMI TOOL, and modified by experts. After labeling, we trained the VNet and compared its performance to that of BMI TOOL. Additionally, to explain the reason for using a CNN-based neural network model in 2D CT images, performance was compared with the latest transformer-based models. Then, we tested the performance on data collected from other institutions, performed labeling on a few data, and performed transfer learning to observe the performance.

Then, to confirm the effect of transfer learning on a large-scale dataset, the neural network was trained for the number of all cases for the three datasets in which liver labeling was performed, and various aspects of the performance were observed. The first performance measures are the learning curve and the final performance. For each dataset, we observed the difference in learning curve and final performance according to transfer learning. Next, we checked the performance of a completely untrained model on a specific dataset. Finally, we observe the performance change of the models on which transfer learning has been performed. A total of 15 models were trained, and the types are shown in Table 1.

Transfer learning strategy

Transfer learning for muscle segmentation involved training the VNet network, trained on datasets collected from hospitals, on a small subset of BTCV datasets. A learning rate of 0.0005 and Adam optimizer were used for all model architectures. Regardless of the number of data used for transfer learning, training was conducted for 500 epochs. When performing transfer learning for the liver, the same learning rate and optimizer were utilized. Upon observing the learning curve, convergence was faster for the liver compared to muscles. Thus, training was conducted for a total of 200 epochs. In this paper, for transfer learning, additional training was conducted for all layers without freezing any part of the network.

Experimental results

In this section, we describe the experiments and results performed in the environment described in the previous section.

Segmentation speed comparison between baseline and VNet

In this section, we compare the segmentation performance speed of VNet and the baseline to determine whether the trained artificial neural network has an advantage in segmentation speed. Table 2 shows the average and standard deviation of the time taken when 5 people who were trained to use the baseline technique performed segmentation on 50 CT images. It took an average of 19 to 39 seconds for humans to perform the split. However, VNet consumed an average of about 0.033 seconds per sheet. Experimental results show that neural network models require a lot of time and effort to construct training data, but can quickly segment many CT images after training.

Performance comparison between BMI TOOL and deep-learning-based models

Figure 5 shows the results of comparing the muscle segmentation performance of VNet and BMI TOOL. The BMI TOOL may misclassify a part of an organ as a muscle because it needs to manually separate the inside and outside of the abdomen and segments based on the HU(Hounsfield Unit) value. It can also misclassify the spinal cord as a muscle. However, the trained model no longer segments organs or the spinal cord as the muscle. We tried to measure the performance using a confusion matrix and it illustrated in Fig. 6. In the case of precision, VNet was more dominant with a score of 0.876713 and BMI tool 0.759561. In many cases, these results are shown because the BMI tool misclassified the organ region and spinal cord. For accuracy, VNet scored 0.986411, and the BMI tool scored 0.979780, confirming that VNet better performed than the BMI tool.

Moreover, the quantitative performance comparison is shown in Table 3. Deep learning-based models demonstrate superior segmentation performance compared to traditional semi-automatic techniques. Although deep learning models require time for training, they can save time compared to traditional semi-automatic tools when performing segmentation tasks on a large volume of images (the time required for segmentation using the semi-automatic model is shown in Table 2). Additionally, CNN-based models still exhibit competitive performance compared to Transformer-based models.

Transfer learning to BTCV dataset

In the previous section, we compared the segmentation performance of baseline and VNet for muscles in terms of speed and accuracy. Through the experimental results, VNet showed faster and better segmentation performance than the baseline after training. In this section, transfer learning was performed on CT images from other devices with the already trained network. The training curve is compared with the initialized VNet model, and the performance advantage of using the pretrained network is verified through experiments.

Transfer learning aims to extract knowledge from one or more source tasks and apply that knowledge to a target task. Even for a trained model, it can be challenging to expect good performance if the model is trained on a dataset taken from another device. For this scenario, we

Table 1 Order of models trained to observe the effect of transfer learning on multiple medical image datasets

Model No.	Training dataset		
1	LiTS		
2	LiTS	BTCV	
3	LiTS	Chaos	
4	LiTS	BTCV	Chaos
5	LiTS	Chaos	BTCV
6	BTCV		
7	BTCV	LiTS	
8	BTCV	Chaos	
9	BTCV	LiTS	Chaos
10	BTCV	Chaos	LiTS
11	Chaos		
12	Chaos	LiTS	
13	Chaos	BTCV	
14	Chaos	LiTS	BTCV
15	Chaos	BTCV	LiTS

sought to determine how many images must be used in training to reach the desired performance through transfer learning.

The distribution of pixels for CT images differs depending on the filming equipment. Therefore, we cannot be convinced that a model will perform as well if another institution has trained it. We experimented to determine how much data is needed to improve performance. We assume that transfer learning will improve the model's performance by collecting additional labeled CT data taken by another device. We constructed the training dataset by labeling the BTCV dataset. Transfer learning was performed and performance was measured while increasing the training data by 5 to 30. The results are shown in Fig. 7. As a result, we confirmed that performance can be significantly improved even if transfer learning is performed with only a small number of data.

Through experimental results, it can be inferred that models trained on SEED data collected from different devices outperform initialized models in terms of segmentation performance. Utilizing the trained model for pseudo labeling during additional labeling tasks for training presents the potential to reduce labeling costs.

Observing the learning curve of deep learning models

In this section, we perform transfer learning for the segmentation task using VNet, transformer-based UNETR, and Swin-UNETR. We observe the model's learning curve and segmentation performance. Our primary focus is on two aspects: first, that the convergence speed of performance is faster with transfer learning; second, that transfer learning yields similar or better segmentation performance.

VNet The learning curve and division performance for the VNet network are shown in Fig. 8. First, for the LiTS dataset, it was found that the model trained through transfer learning converged faster, and the performance did not differ significantly between the models. Next, for the BTCV dataset, the model that conducted transfer learning using other data performed better than the model trained with only BTCV data. For the Chaos dataset, similar to the learning curve for the BTCV dataset, the models trained through transfer learning showed significantly faster convergence and better final performance.

UNETR The learning curve and final performance of the UNETR model are shown in Fig. 9. In the learning curve for the LiTS dataset, the convergence rate of the model trained only on the LiTS dataset was low. In addition, models trained on the BTCV dataset showed relatively better performance and faster performance convergence than those trained by transfer learning. Similarly, for the Chaos dataset, the transfer learning model after learning with other datasets performed better, and the convergence speed of performance was faster than the model trained only with the Chaos dataset.

Swin-UNETR Finally, the learning curve and final performance of the Swin-UNETR model are shown in Fig. 10. As a result of training the Swin-UNETR model on the LiTS dataset, there was no significant difference in convergence speed or performance. However, for the BTCV and Chaos datasets, the models trained with transfer learning converged faster and had a better final performance. Among the models trained with transfer learning, models trained with the LiTS dataset show higher performance.

Table 2 Comparison of segmentation speed of the VNet and baseline

	Person 1	Person 2	Person 3	Person 4	Person 5	VNet
Avg	19.80s	29.28s	26.04s	33.80s	39.46s	0.03s
SD	2.84s	6.37s	3.34s	6.74s	8.47s	0.001s

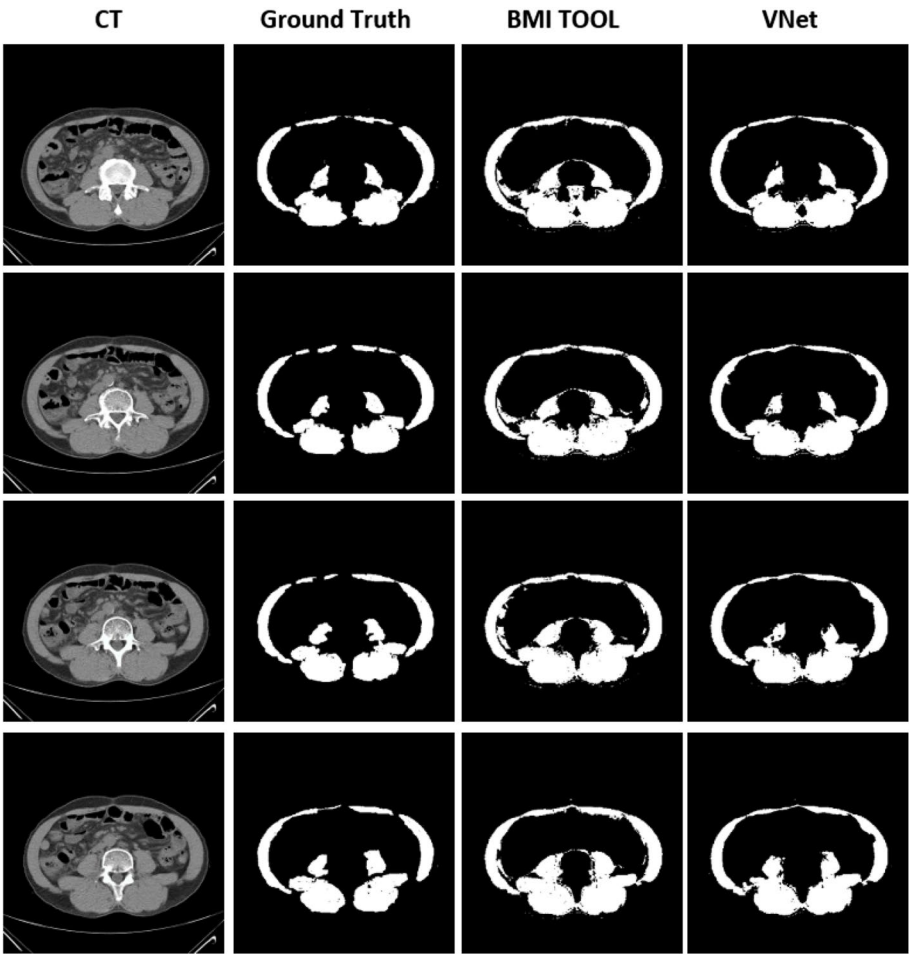
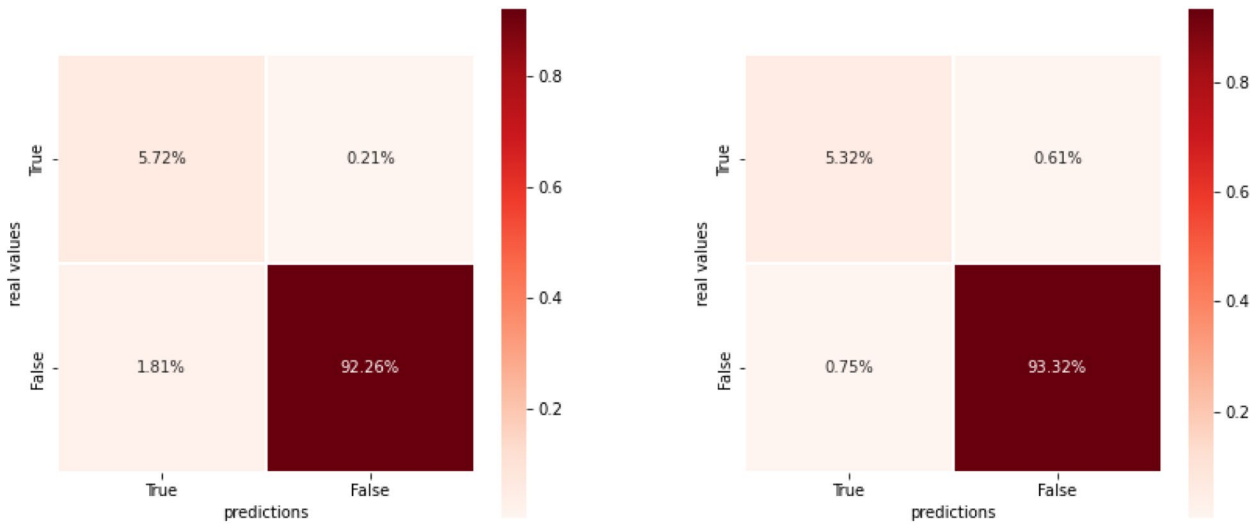


Fig. 5 The segmentation results using VNet and BMI TOOL



(a) Confusion matrix of BMI TOOL (b) Confusion matrix of VNet

Fig. 6 Confusion matrix of BMI TOOL and VNet

Table 3 Comparison of segmentation performance of BMI TOOL and neural network models

Model	Dice cofficient	95%HD	Accuracy	Precision	Training time
BMI TOOL	0.84	17.1793	0.9798	0.7595	-
VNet	0.8846	5.3277	0.9864	0.8767	1.133hr
UNETR	0.8663	5.7770	0.9840	0.8563	3.213hr
Swin-UNETR	0.8595	6.9311	0.9836	0.8630	2.292hr
UNET	0.8737	6.8585	0.9853	0.8858	1.237hr
ParaTransCNN	0.8637	4.0527	0.9837	0.8564	3.036hr

Observation of performance on untrained dataset

In the previous section, we observed the performance convergence speed and final performance of each model according to whether or not transfer learning was performed. In this section, the performance of each dataset is compared according to whether or not it is trained. Experimental results demonstrated that models trained on the LiTS dataset achieved relatively high accuracy on previously unseen datasets across all models. Unlike the CHAOS and BTCV datasets, which were collected from a single institution, the LiTS dataset was gathered from multiple institutions and contains the largest volume of data. This indicates that models trained on large amounts of data collected from various institutions achieve better generalization.

VNet First, the performance comparison for the VNet network is shown in Table 4. Models not trained on the LiTS dataset showed relatively low performance on the LiTS dataset. However, in the performance comparison on the BTCV dataset, there were cases with higher performance. The high-performing models had all been trained on the LiTS dataset before. Finally, for the Chaos dataset, the model trained on the Chaos data showed the best performance. However, unlike the results for the LiTS dataset, the performance difference was insignificant.

UNETR Secondly, the same experiment was performed on the UNETR model, and the results are shown in Table 5. Regarding the LiTS dataset, the models that were not yet trained on the LiTS dataset have relatively low performance compared to the trained models. Next, a model that performed better than the model trained on the BTCV dataset was observed. These models had all been trained on the LiTS dataset. Finally, we can see

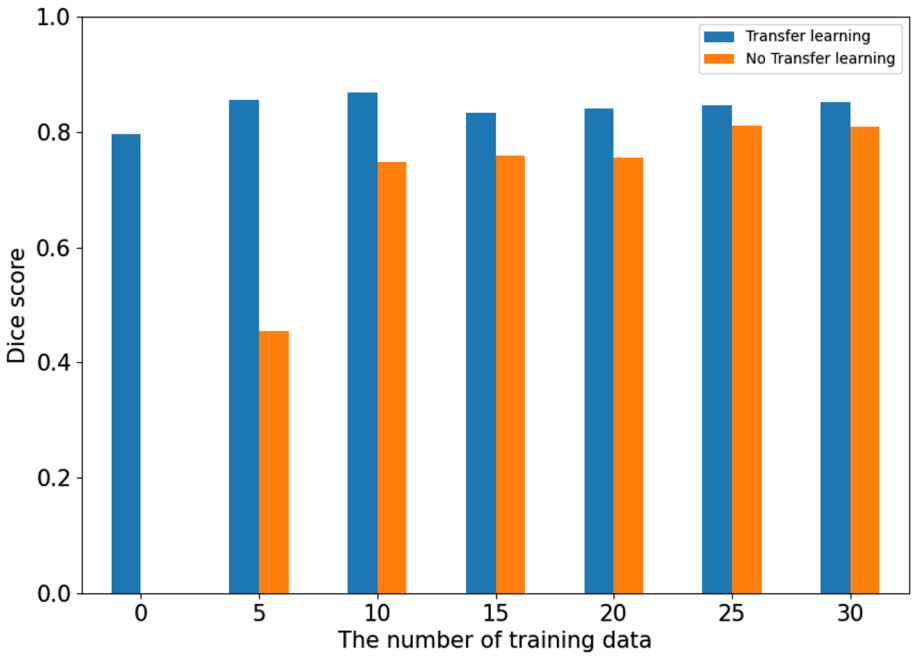
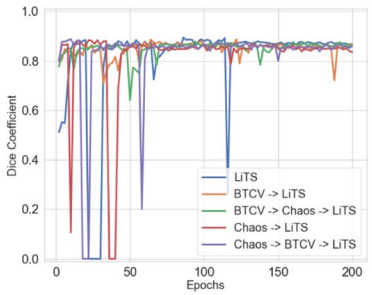
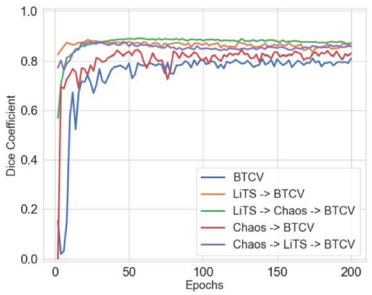


Fig. 7 Model performance according to the number of additional training data for transfer learning



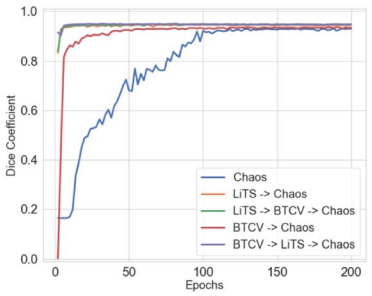
Model	Dice
L	0.8940
B→L	0.8865
C→L	0.8833
B→C→L	0.8772
C→B→L	0.8876

(a) LiTS Dataset



Model	Dice
B	0.8107
L→B	0.8843
C→B	0.8845
L→C→B	0.8903
C→L→B	0.8763

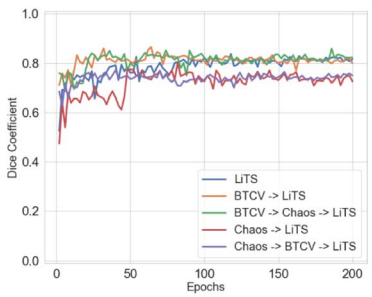
(b) BTCV Dataset



Model	Dice
C	0.9310
L→C	0.9466
B→C	0.9354
L→B→C	0.9489
B→L→C	0.9503

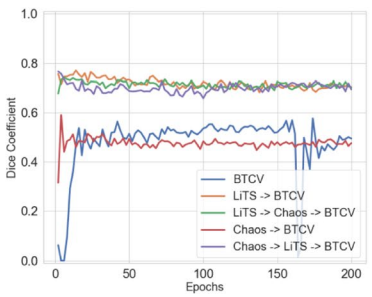
(c) Chaos Dataset

Fig. 8 The learning curves of all VNet networks trained in the order presented in Table 1 (L : LiTS, B : BTCV, C : Chaos). When performing transfer learning across all datasets, the convergence speed of performance is faster. Moreover, in cases of small training data sizes, employing transfer learning results in better final performance



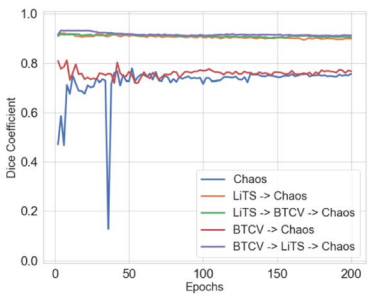
Model	Dice
L	0.8371
B→L	0.8654
C→L	0.8021
B→C→L	0.8582
C→B→L	0.7710

(a) LiTS Dataset



Model	Dice
B	0.5768
L→B	0.7704
C→B	0.5902
L→C→B	0.7423
C→L→B	0.7665

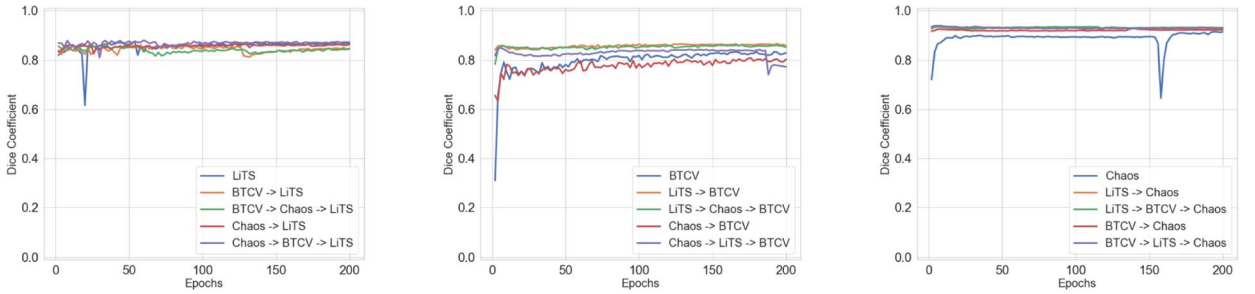
(b) BTCV Dataset



Model	Dice
C	0.7782
L→C	0.9127
B→C	0.8118
L→B→C	0.9212
B→L→C	0.9326

(c) Chaos Dataset

Fig. 9 The learning curves of UNETR models. The performance convergence speed of the models trained through transfer learning is generally faster, and the final performance is similar or better. In both the BTCV and Chaos datasets, there is a significant performance difference depending on whether transfer learning is applied. Performance is notably better when transfer learning is employed (L : LiTS, B : BTCV, C : Chaos)



Model No.	Dice
L	0.8721
B→L	0.8636
C→L	0.8662
B→C→L	0.8663
C→B→L	0.8800

(a) LiTS Dataset

Model No.	Dice
B	0.8831
L→B	0.8660
C→B	0.8077
L→C→B	0.8610
C→L→B	0.8492

(b) BTCV Dataset

Model No.	Dice
C	0.9143
L→C	0.9343
B→C	0.9272
L→B→C	0.9356
B→L→C	0.9393

(c) Chaos Dataset

Fig. 10 Graphs illustrating the learning curve of the Swin-UNETR model. As with other models, the convergence speed of models trained through transfer learning is faster, and the final performance is similar or better (L : LiTS, B : BTCV, C : Chaos)

Table 4 Performance comparison of the VNet models trained with one dataset and models that were never trained with it

Trained datasets	Dataset		
	LiTS	BTCV	Chaos
LiTS	0.8940[a]	0.8736	0.9124
BTCV→LiTS	0.8865[a]	-	0.9205
Chaos→LiTS	0.8833[a]	0.8602	-
BTCV→Chaos→LiTS	0.8772[a]	-	-
Chaos→BTCV→LiTS	0.8876[a]	-	-
BTCV	0.6849	0.8107[a]	0.8999
LiTS→BTCV	-	0.8843[a]	0.9193
Chaos→BTCV	0.7321	0.8845[a]	-
LiTS→Chaos→BTCV	-	0.8903[a]	-
Chaos→LiTS→BTCV	-	0.8763[a]	-
Chaos	0.6083	0.7631	0.9310[a]
LiTS→Chaos	-	0.8538	0.9466[a]
BTCV→Chaos	0.6420	-	0.9354[a]
LiTS→BTCV→Chaos	-	-	0.9489[a]
BTCV→LiTS→Chaos	-	-	0.9503[a]

[a] Performance on the last trained dataset

Table 5 Performance comparison of the UNETR models trained with one dataset and models that were never trained with it

Trained datasets	Dataset		
	LiTS	BTCV	Chaos
LiTS	0.8371[a]	0.6857	0.8729
BTCV→LiTS	0.8654[a]	-	0.8798
Chaos→LiTS	0.8021[a]	0.6379	-
BTCV→Chaos→LiTS	0.8582[a]	-	-
Chaos→BTCV→LiTS	0.7710[a]	-	-
BTCV	0.6408	0.5768[a]	0.7994
LiTS→BTCV	-	0.7704[a]	0.8868
Chaos→BTCV	0.6260	0.5902[a]	-
LiTS→Chaos→BTCV	-	0.7423[a]	-
Chaos→LiTS→BTCV	-	0.7665[a]	-
Chaos	0.4613	0.5668	0.7782[a]
LiTS→Chaos	-	0.8126	0.9127[a]
BTCV→Chaos	0.6019	-	0.8118[a]
LiTS→BTCV→Chaos	-	-	0.9212[a]
BTCV→LiTS→Chaos	-	-	0.9326[a]

[a] Performance on the last trained dataset

that the untrained model performs better on the Chaos dataset. Among them, the performance was better in the model trained on the LiTS dataset.

Swin-UNETR Finally, the experimental results for the Swin-UNETR model are shown in Table 6. Models not trained on the LiTS dataset performed relatively poorly

Table 6 Performance comparison of the Swin-UNETR models trained with one dataset and models that were never trained with it

Trained datasets	Dataset		
	LiTS	BTCV	Chaos
LiTS	0.8721[a]	0.8467	0.8909
BTCV→LiTS	0.8636[a]	-	0.8652
Chaos→LiTS	0.8662[a]	0.8093	-
BTCV→Chaos→LiTS	0.8663[a]	-	-
Chaos→BTCV→LiTS	0.8800[a]	-	-
BTCV	0.7013	0.8331[a]	0.8823
LiTS→BTCV	-	0.8660[a]	0.8876
Chaos→BTCV	0.7104	0.8077[a]	-
LiTS→Chaos→BTCV	-	0.8610[a]	-
Chaos→LiTS→BTCV	-	0.8492[a]	-
Chaos	0.5933	0.8494	0.9143[a]
LiTS→Chaos	-	0.7733	0.9343[a]
BTCV→Chaos	0.6804	-	0.9272[a]
LiTS→BTCV→Chaos	-	-	0.9356[a]
BTCV→LiTS→Chaos	-	-	0.9393[a]

[a] Performance on the last trained dataset

compared to the trained models. For the BTCV dataset, the models that had been trained on the LiTS dataset performed relatively well. Finally, the model trained on the Chaos dataset performed better than the non-trained model, but the difference was insignificant.

We observed that the VNet performs better segmentation on similar domain data that has not been trained.

Observing the effect of transfer learning and catastrophic forgetting

In this section, we observe the segmentation performance on previously trained datasets after transfer learning. Experimental results showed that performance on the datasets trained before transfer learning generally declined slightly. These results highlight the limitations of transfer learning and suggest directions for future research.

VNet First, the performance change of the VNet network is shown in Table 7. It shows that models trained on the LiTS dataset and then transferred learning to other datasets show lower performance on the LiTS dataset. In the BTCV dataset, we confirmed that the model's performance was well preserved without significant change. The performance of the models trained on the Chaos dataset and then subjected to transfer learning is shown on the right of the table. Although the performance was lower than the model trained only on the Chaos dataset, the performance gap was insignificant.

UNETR Next, the results of observing the performance change of the models that performed transfer learning on the UNETR network are shown in Table 8. The transfer learned models trained once on the LiTS dataset show degraded performance on the LiTS dataset. In addition, the performance of the models trained on the BTCV dataset and transferred to other datasets is shown in the center of the table. We found cases where the models with transfer learning performed better than the model trained with only the BTCV dataset. However, the increased performance was not competitive compared to the VNet network. Finally, the performance on the Chaos dataset is shown on the right of the table. We found cases where the transfer learned model's performance on the chaos dataset improved. However, the improved performance did not exceed that of the other networks.

Swin-UNETR The performance changes of the models that performed transfer learning on the Swin-UNETR network are shown in Table 9. The transfer

Table 7 Performance measurement of transfer learned VNet models on the previously trained dataset

LiTS		BTCV		Chaos	
Trained dataset	Dice score	Trained dataset	Dice score	Trained dataset	Dice score
LiTS	0.8940[a]	BTCV	0.8107[a]	Chaos	0.9310[a]
LiTS → BTCV	0.8288	BTCV → LiTS	0.8533	Chaos → LiTS	0.8937
LiTS → Chaos	0.7728	BTCV → Chaos	0.8032	Chaos → BTCV	0.9056
LiTS → BTCV → Chaos	0.7902	BTCV → LiTS → Chaos	0.8608	Chaos → LiTS → BTCV	0.9178
LiTS → Chaos → BTCV	0.7996	BTCV → Chaos → LiTS	0.8240	Chaos → BTCV → LiTS	0.9164
BTCV → LiTS	0.8865[a]	LiTS → BTCV	0.8843[a]	LiTS → Chaos	0.9466[a]
BTCV → LiTS → Chaos	0.6744	LiTS → BTCV → Chaos	0.8638	LiTS → Chaos → BTCV	0.9143
Chaos → LiTS	0.8833[a]	Chaos → BTCV	0.8845[a]	BTCV → Chaos	0.9354[a]
Choas → LiTS → BTCV	0.8146	Chaos → BTCV → LiTS	0.8621	BTCV → Chaos → LiTS	0.8946

[a] Performance on the last trained dataset

Table 8 Performance measurement of transfer learned UNETR models on the previously trained dataset

LiTS		BTCV		Chaos	
Trained dataset	**Dice score**	**Trained dataset**	**Dice score**	**Trained dataset**	**Dice score**
LiTS	0.8371[a]	BTCV	0.5768[a]	Chaos	0.7782[a]
LiTS → BTCV	0.8322	BTCV → LiTS	0.6790	Chaos → LiTS	0.8189
LiTS → Chaos	0.7441	BTCV → Chaos	0.8140	Chaos → BTCV	0.8112
LiTS → BTCV → Chaos	0.7161	BTCV → LiTS → Chaos	0.7493	Chaos → LiTS → BTCV	0.8750
LiTS → Chaos → BTCV	0.7883	BTCV → Chaos → LiTS	0.6966	Chaos → BTCV → LiTS	0.8135
BTCV → LiTS	0.8654[a]	LiTS → BTCV	0.7704[a]	LiTS → Chaos	0.9127[a]
BTCV → LiTS → Chaos	0.7307	LiTS → BTCV → Chaos	0.8249	LiTS → Chaos → BTCV	0.8957
Chaos → LiTS	0.8021[a]	Chaos → BTCV	0.5902[a]	BTCV → Chaos	0.8118[a]
Choas → LiTS → BTCV	0.7754	Chaos → BTCV → LiTS	0.4617	BTCV → Chaos → LiTS	0.8696

[a] Performance on the last trained dataset

learned models trained once on the LiTS dataset show low performance on the LiTS dataset. In the results for the BTCV dataset, the performance of transfer learned models was well preserved or slightly improved compared to the results on the LiTS dataset. In the results for the BTCV dataset, the performance of transfer learned models was well preserved or slightly improved. The experiment result of the Chaos dataset is shown on the right of the table. After transfer learning was conducted, the models showed slightly lower segmentation performance on the Chaos dataset. The overall performance was not significantly different from the VNet.

Through experimental results, it can be observed that the 2D CNN-based model demonstrates competitive segmentation performance across various scenarios. Additionally, it is noted that even after further training with additional datasets, the model maintains its performance at a competitive level.

Conclusion

In this section, we summarize the experimental results, point out the limitations, and propose directions for future research. This study assumes a scenario where training data is scarce in clinical settings. We first compared the performance of existing semi-automatic segmentation methods and deep learning-based methods. Then, we observed the performance of various deep learning models for segmentation tasks on 2D CT images. Finally, we demonstrated through experiments that learning strategies using transfer learning are effective across different deep learning models.

The experimental results indicated that deep learning-based methods surpassed semi-automatic segmentation methods in both segmentation speed and performance. Although deep learning-based models require training time, their advantage becomes more apparent when a large amount of labeling is needed, as the time required for manual segmentation exceeds the time needed for training.

Table 9 Performance measurement of transfer learned Swin-UNETR models on the previously trained dataset

LiTS		BTCV		Chaos	
Trained dataset	**Dice score**	**Trained dataset**	**Dice score**	**Trained dataset**	**Dice score**
LiTS	0.8721[a]	BTCV	0.8831[a]	Chaos	0.9143[a]
LiTS → BTCV	0.8501	BTCV → LiTS	0.8264	Chaos → LiTS	0.8970
LiTS → Chaos	0.7913	BTCV → Chaos	0.8455	Chaos → BTCV	0.8938
LiTS → BTCV → Chaos	0.8012	BTCV → LiTS → Chaos	0.8522	Chaos → LiTS → BTCV	0.9012
LiTS → Chaos → BTCV	0.8461	BTCV → Chaos → LiTS	0.8324	Chaos → BTCV → LiTS	0.9140
BTCV → LiTS	0.8636[a]	LiTS → BTCV	0.8660[a]	LiTS → Chaos	0.9343[a]
BTCV → LiTS → Chaos	0.8223	LiTS → BTCV → Chaos	0.8519	LiTS → Chaos → BTCV	0.8975
Chaos → LiTS	0.8662[a]	Chaos → BTCV	0.8077[a]	BTCV → Chaos	0.9272[a]
Choas → LiTS → BTCV	0.8409	Chaos → BTCV → LiTS	0.8539	BTCV → Chaos → LiTS	0.9072

[a] Performance on the last trained dataset

In the segmentation of 2D CT images with limited training data, CNN-based models were found to achieve competitive performance compared to the latest transformer-based models. The CNN-based VNet showed similar or better performance than transformer-based models under the proposed scenario. We proposed a learning strategy utilizing transfer learning to effectively train deep learning-based models in situations with limited training data. To demonstrate the effectiveness of this strategy across various networks, we conducted training and performance verification on multiple networks. The experimental results showed that transfer learning yielded good segmentation performance in all networks. In actual clinical environments, when new segmentation is required for data captured with different devices, models can be efficiently trained through transfer learning. If CT images captured with different devices and pre-trained models are available, they can be utilized to train segmentation models with better performance.

However, there are limitations to this study. More experiments with a wider range of comparison models are needed. Numerous model architectures have been proposed for medical image segmentation, and new architectures continue to emerge. Investigating the utility and exploring the reasons for the effectiveness of transfer learning across more model architectures could be a future research task. Additionally, research could focus on achieving more precise model generalization by integrating collected medical information for each patient into the segmentation process. Finally, addressing the performance degradation on the data trained before transfer learning remains a challenge that needs to be resolved.

Authors' contributions

Seunghan Yoon conducted data investigation, software development, experiment, and initial drafting of the paper. Tae Hyung Kim and Young Kul Jung verified the data used in experiments and provided medical opinions. Younghoon Kim led the overall process in writing, and reviewed and revised the initial draft of the paper. All authors reviewed the manuscript and Y. Kim and Y. Jung provided supervision as co-corresponding authors.

Funding

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-00155885, Artificial Intelligence Convergence Innovation Human Resources Development (Hanyang University ERICA)).

This research was financially supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D program (Project No. P0025661)

The funding bodies played no role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Availability of data and materials

The LiTS, BTCV, and Chaos datasets used in this study are public datasets. The LiTS dataset can be downloaded from https://competitions.codalab.org/competitions/17094, the BTCV dataset can be found at https://www.synapse.org/#!Synapse:syn3193805/wiki/217789, and the Chaos dataset can be downloaded from https://chaos.grand-challenge.org/Download/. Other data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate

We confirm that all methods were carried out in accordance with relevant guilelines and regulations.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 19 April 2023 Accepted: 1 October 2024
Published online: 09 October 2024

References

1. Lu H, Wang H, Zhang Q, Yoon SW, Won D. A 3D convolutional neural network for volumetric image semantic segmentation. Procedia Manuf. 2019;39:422–8.
2. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany. Cham: Springer International Publishing; 2015. proceedings, part III 18 2015. p. 234–241.
3. Bilic P, Christ P, Li HB, Vorontsov E, Ben-Cohen A, Kaissis G, et al. The liver tumor segmentation benchmark (lits). Med Image Anal. 2023;84:102680.
4. Kavur AE, Gezer NS, Barış M, Aslan S, Conze PH, Groza V, et al. CHAOS challenge-combined (CT-MR) healthy abdominal organ segmentation. Med Image Anal. 2021;69:101950.
5. Gu R, Zhang J, Huang R, Lei W, Wang G, Zhang S. Domain composition and attention for unseen-domain generalizable medical image segmentation. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24. Springer; 2021. pp. 241–250.
6. Wang J, Gao R, Huo Y, Bao S, Xiong Y, Antic SL, et al. Lung cancer detection using co-learning from chest CT images and clinical demographics. In: Medical imaging 2019: Image processing, vol. 10949. SPIE; 2019. pp. 365–371.
7. Kim SS, Kim JH, Jeong WK, Lee J, Kim YK, Choi D, Lee WJ. Semiautomatic software for measurement of abdominal muscle and adipose areas using computed tomography: a STROBE-compliant article. Medicine. 2019;98(22):e15867. https://doi.org/10.1097/MD.0000000000015867.
8. Sharma N, Aggarwal LM. Automated medical image segmentation techniques. J Med Phys Assoc Med Phys India. 2010;35(1):3.
9. Gaillochet M, Desrosiers C, Lombaert H. Active learning for medical image segmentation with stochastic batches. 2023. arXiv preprint arXiv:2301.07670.
10. Mharib AM, Ramli AR, Mashohor S, Mahmood RB. Survey on liver CT image segmentation methods. Artif Intell Rev. 2012;37:83–95.
11. Ciresan D, Giusti A, Gambardella L, Schmidhuber J. Deep neural networks segment neuronal membranes in electron microscopy images. Adv Neural Inf Process Syst. 2012;25:2843–51.
12. Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J. Unet++: a nested u-net architecture for medical image segmentation. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4. Springer; 2018. pp. 3–11.
13. Huang H, Lin L, Tong R, Hu H, Zhang Q, Iwamoto Y, et al. Unet 3+: a full-scale connected unet for medical image segmentation. In: ICASSP

2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2020. pp. 1055–1059.
14. Kim TH, Jung YK, Yim HJ, Baik JW, Yim SY, Lee YS, et al. Impacts of muscle mass dynamics on prognosis of outpatients with cirrhosis. Clin Mol Hepatol. 2022;28(4):876–89.
15. Kanavati F, Islam S, Arain Z, Aboagye EO, Rockall A. Fully-automated deep learning slice-based muscle estimation from CT images for sarcopenia assessment. 2020. arXiv preprint arXiv:2006.06432.
16. Edwards K, Chhabra A, Dormer J, Jones P, Boutin RD, Lenchik L, Fei B. Abdominal muscle segmentation from CT using a convolutional neural network. Proc SPIE Int Soc Opt Eng. 2020;11317:113170L. https://doi.org/10.1117/12.2549406.
17. Amarasinghe KC, Lopes J, Beraldo J, Kiss N, Bucknell N, Everitt S, Jackson P, Litchfield C, Denehy L, Blyth BJ, Siva S. A deep learning model to automate skeletal muscle area measurement on computed tomography images. Front Oncol. 2021;11:580806.
18. Lee YS, Hong N, Witanto JN, Choi YR, Park J, Decazes P, et al. Deep neural network for automatic volumetric segmentation of whole-body CT images for body composition assessment. Clin Nutr. 2021;40(8):5038–46.
19. Ackermans LL, Volmer L, Wee L, Brecheisen R, Sánchez-González P, Seiffert AP, et al. Deep learning automated segmentation for muscle and adipose tissue from abdominal computed tomography in polytrauma patients. Sensors. 2021;21(6):2083.
20. Park HJ, Shin Y, Park J, Kim H, Lee IS, Seo DW, et al. Development and validation of a deep learning system for segmentation of abdominal muscle and fat on computed tomography. Korean J Radiol. 2020;21(1):88–100.
21. Castiglione J, Somasundaram E, Gilligan LA, Trout AT, Brady S. Automated segmentation of abdominal skeletal muscle on pediatric CT scans using deep learning. Radiol Artif Intell. 2021;3(2):e200130. https://doi.org/10.1148/ryai.2021200130.
22. Dabiri S, Popuri K, Feliciano EMC, Caan BJ, Baracos VE, Beg MF. Muscle segmentation in axial computed tomography (CT) images at the lumbar (L3) and thoracic (T4) levels for body composition analysis. Comput Med Imaging Graph. 2019;75:47–55.
23. Lee H, Troschel FM, Tajmir S, Fuchs G, Mario J, Fintelmann FJ, et al. Pixel-level deep segmentation: artificial intelligence quantifies muscle on computed tomography for body morphometric analysis. J Digit Imaging. 2017;30(4):487–98.
24. Fu Y, Ippolito JE, Ludwig DR, Nizamuddin R, Li HH, Yang D. Automatic segmentation of CT images for ventral body composition analysis. Med Phys. 2020;47(11):5723–30.
25. Weber KA, Abbott R, Bojilov V, Smith AC, Wasielewski M, Hastie TJ, et al. Multi-muscle deep learning segmentation to automate the quantification of muscle fat infiltration in cervical spine conditions. Sci Rep. 2021;11(1):1–15.
26. Li H, Luo H, Liu Y. Paraspinal muscle segmentation based on deep neural network. Sensors. 2019;19(12):2650.
27. Zopfs D, Bousabarah K, Lennartz S, Dos Santos DP, Schlaak M, Theurich S, et al. Evaluating body composition by combining quantitative spectral detector computed tomography and deep learning-based image segmentation. Eur J Radiol. 2020;130:109153.
28. Nishiyama D, Iwasaki H, Taniguchi T, Fukui D, Yamanaka M, Harada T, et al. Deep generative models for automated muscle segmentation in computed tomography scanning. PLoS ONE. 2021;16(9):e0257371.
29. Albahri AS, Duhaim AM, Fadhel MA, Alnoor A, Baqer NS, Alzubaidi L, Albahri OS, Alamoodi AH, Bai J, Salhi A, Santamaría J. A systematic review of trustworthy and explainable artificial intelligence in healthcare: Assessment of quality, bias risk, and data fusion. Inf Fusion. 2023;96:156–91. https://doi.org/10.1016/j.inffus.2023.03.008.
30. Alzubaidi L, Bai J, Al-Sabaawi A, Santamaría J, Albahri A, Al-dabbagh BSN, et al. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. J Big Data. 2023;10(1):46.
31. Shamrat FJM, Azam S, Karim A, Ahmed K, Bui FM, De Boer F. High-precision multiclass classification of lung disease through customized MobileNetV2 from chest X-ray images. Comput Biol Med. 2023;155:106646.
32. Sutradhar A, Al Rafi M, Ghosh P, Shamrat FJ, Moniruzzaman M, Ahmed K, Azad AK, Bui FM, Chen L, Moni MA. An intelligent thyroid diagnosis system utilising multiple ensemble and explainable algorithms with medical supported attributes. IEEE Trans Artif Intell. 2023;5:2840–55. https://doi.org/10.1109/TAI.2023.3327981.
33. Shamrat FJM, Akter S, Azam S, Karim A, Ghosh P, Tasnim Z, et al. AlzheimerNet: An effective deep learning based proposition for alzheimer's disease stages classification from functional brain changes in magnetic resonance images. IEEE Access. 2023;11:16376–95.
34. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway: IEEE; 2016. p. 2818–2826.
35. Sun H, Xu J, Duan Y. ParaTransCNN: Parallelized TransCNN Encoder for Medical Image Segmentation. 2024. arXiv preprint arXiv:2401.15307.
36. Heidari M, Kazerouni A, Soltany M, Azad R, Aghdam EK, Cohen-Adad J, Merhof D. Hiformer: Hierarchical multi-scale representations using transformers for medical image segmentation. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. Piscataway: IEEE; 2023. p. 6202–6212.
37. Li X, Pang S, Zhang R, Zhu J, Fu X, Tian Y, et al. ATTransUNet: An enhanced hybrid transformer architecture for ultrasound and histopathology image segmentation. Comput Biol Med. 2023;152:106365.
38. Yang H, Yang D. CSwin-PNet: A CNN-Swin Transformer combined pyramid network for breast lesion segmentation in ultrasound images. Expert Syst Appl. 2023;213:119024.
39. Kawamoto M, Kamiya N, Zhou X, Kato H, Hara T, Fujita H. Simultaneous Learning of Erector Spinae Muscles for Automatic Segmentation of Site-Specific Skeletal Muscles in Body CT Images. IEEE Access. 2023;12:15468-76. https://doi.org/10.1109/ACCESS.2023.3335948.
40. Kamiya N. Deep Learning Technique for Musculoskeletal Analysis. In: Lee, G., Fujita, H. (eds) Deep Learning in Medical Image Analysis . Advances in Experimental Medicine and Biology, vol 1213. Cham: Springer; 2020. p. 165–176. https://doi.org/10.1007/978-3-030-33128-3_11.
41. Ashino K, Kamiya N, Zhou X, Kato H, Hara T, Fujita H. Joint segmentation of sternocleidomastoid and skeletal muscles in computed tomography images using a multiclass learning approach. Radiol Phys Technol. 2024:1-8. https://doi.org/10.1007/s12194-024-00839-1.
42. Chen S, Ma K, Zheng Y. Med3d: Transfer learning for 3d medical image analysis. 2019. arXiv preprint arXiv:1904.00625.
43. Alzubaidi L, Fadhel MA, Al-Shamma O, Zhang J, Santamaría J, Duan Y, et al. Towards a better understanding of transfer learning for medical imaging: a case study. Appl Sci. 2020;10(13):4523.
44. Raghu M, Zhang C, Kleinberg J, Bengio S. Transfusion: Understanding transfer learning for medical imaging. Adv Neural Inf Process Syst. 2019;32:3347–57.
45. Alzubaidi L, Al-Amidie M, Al-Asadi A, Humaidi AJ, Al-Shamma O, Fadhel MA, et al. Novel transfer learning approach for medical imaging with limited labeled data. Cancers. 2021;13(7):1590.
46. Alzubaidi L, Fadhel MA, Al-Shamma O, Zhang J, Duan Y. Deep learning models for classification of red blood cells in microscopy images to aid in sickle cell anemia diagnosis. Electronics. 2020;9(3):427.
47. Heker M, Greenspan H. Joint liver lesion segmentation and classification via transfer learning. 2020. arXiv preprint arXiv:2004.12352.
48. Milletari F, Navab N, Ahmadi SA, V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). IEEE; 2016. pp. 565–71.
49. Xu C, Prince JL. Snakes, shapes, and gradient vector flow. IEEE Trans Image Process. 1998;7(3):359–69.
50. Hatamizadeh A, Tang Y, Nath V, Yang D, Myronenko A, Landman B, Roth HR, Xu D. Unetr: Transformers for 3d medical image segmentation. InProceedings of the IEEE/CVF winter conference on applications of computer vision. Piscataway: IEEE; 2022. p. 574–584.
51. Hatamizadeh A, Nath V, Tang Y, Yang D, Roth HR, Xu D. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In: International MICCAI Brainlesion Workshop. Springer; 2022. pp. 272–284.

Publisher's Note