PAPER

Pool-Unet: A Novel Tongue Image Segmentation Method Based on Pool-Former and Multi-Task Mask Learning

Xiangrun LI[†*], **Qiyu SHENG**[††*], **Guangda ZHOU**[†], **Jialong WEI**[†], **Yanmin SHI**[†††], **Zhen ZHAO**[†], **Yongwei LI**[††††], **Xingfeng LI**[†††††], ***and*** **Yang LIU**[†a)], ***Nonmembers***

SUMMARY Automated tongue segmentation plays a crucial role in the realm of computer-aided tongue diagnosis. The challenge lies in developing algorithms that achieve higher segmentation accuracy and maintain less memory space and swift inference capabilities. To relieve this issue, we propose a novel Pool-unet integrating Pool-former and Multi-task mask learning for tongue image segmentation. First of all, we collected 756 tongue images taken in various shooting environments and from different angles and accurately labeled the tongue under the guidance of a medical professional. Second, we propose the Pool-unet model, combining a hierarchical Pool-former module and a U-shaped symmetric encoder-decoder with skip-connections, which utilizes a patch expanding layer for up-sampling and a patch embedding layer for down-sampling to maintain spatial resolution, to effectively capture global and local information using fewer parameters and faster inference. Finally, a Multi-task mask learning strategy is designed, which improves the generalization and anti-interference ability of the model through the Multi-task pre-training and self-supervised fine-tuning stages. Experimental results on the tongue dataset show that compared to the state-of-the-art method (OET-NET), our method has 25% fewer model parameters, achieves 22% faster inference times, and exhibits 0.91% and 0.55% improvements in Mean Intersection Over Union (MIOU), and Mean Pixel Accuracy (MPA), respectively.

key words: *tongue image segmentation, multi-task mask learning, pool-former, pool-unet*

1. Introduction

Tongue diagnosis is one of the primary methods of the four diagnostic methods of traditional Chinese medicine, which is used to diagnose lesions by observing the changes in the patient's tongue texture, tongue coating, and sublingual pulse. In recent years, important breakthroughs have been made in computer-aided tongue diagnosis, bringing objective diagnostic results and overcoming the limitations of subjectivity and individual variability in tongue diagnosis. Intelligent tongue diagnosis consists of four main steps: 1) collection of tongue images, 2) automated tongue segmentation, 3) automated tongue categorization, and 4) disease diagnosis [1]. The accuracy of tongue segmentation directly affects the final diagnosis of the tongue. For real-time computer-aided tongue diagnosis, the computation cost is critical [2]. Therefore, it is necessary to realize the reduction of the number of parameters and the improvement of the inference speed while guaranteeing the segmentation accuracy of the model, to realize the accurate and fast segmentation tongue image.

Many methods have been proposed for tongue image segmentation. For instance, Zhang et al. [3] proposed a tongue segmentation method based on grayscale histogram projection and OTSU method. Qin et al. [4] developed a hybrid tongue image segmentation algorithm using initialized SNAKE contour lines. Liu et al. [5] introduced an edge detection algorithm for unevenly illuminated images. However, these traditional methods typically rely on specific feature extraction techniques, such as edge detection and color information, which may not fully capture the diversity of the tongue, resulting in limited segmentation accuracy.

To improve the accuracy of tongue image segmentation, deep learning networks are widely used. Lin et al. [6] proposed a tongue segmentation network combining Res-50 and Deep-Mask, achieving lower loss values and higher classification accuracy. Xue et al. [7] utilized FCN-8s for tongue image segmentation and mitigated resolution degradation through up-sampling operations. Trajanovski et al. [8] employed the Unet network and integrated different color spaces for tongue image segmentation. Zhou et al. [9] improved the Atrous Spatial Pyramid Pooling (ASPP) method by utilizing four parallel convolutional layers, enabling multi-scale feature extraction and contextual information capture. Lin et al propose an end-to-end trainable tongue image segmentation method using a deep convolutional neural network based on ResNet[10]. Zhou et al. [11] propose a TongueNet tongue image segmentation network based on U-net as the backbone segmentation network and combined with morphological layers. Although the aforementioned tongue image segmentation methods exhibit high accuracy, they are hindered by large model sizes and slow inference speeds [12]. Given the rapid advancement of Internet-of-Things (IoT) applications, there is an urgent demand for training lightweight and efficient tongue image segmentation algorithms to cater to practical application requirements.

The Transformer approach [13], which has made a big splash in the field of Natural Language Processing [14], of-

Manuscript received February 2, 2024.
Manuscript revised May 8, 2024.
Manuscript publicized May 29, 2024.

[†]School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China.
[††]School of Data Science, Qingdao University of Science and Technology, Qingdao 266061, China.
[†††]Department of Cloud Network, China Unicom, Qingdao, 266071, China.
[††††]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100089, China.
[†††††]School of Computer Science and Technology, Hainan University, Haikou, 570228, China.
[*]These authors contributed equally to the manuscript.
a) E-mail: yangliu@qust.edu.cn (Corresponding author)
DOI: 10.1587/transfun.2024EAP1015

fers a new solution to the above problems. Liu et al. proposed an efficient and effective hierarchical visual Transformer, known as Swin Transformer. Based on Swin Transformer, Lin et al. [15], [16] resented a network called DS-Transunet, combining Swin Transformer with Unet. Cao et al. introduced a network named Swin-unet, which combines a pure Transformer structure with Unet [17]. They achieved Self-Attention computation from local to global in the encoder and utilized up-sampling of global features to the input resolution in the decoder for corresponding pixel-level segmentation predictions.

Inspired by the above work, this paper proposes an accurate, fast, and memory-intensive network model called Pool-unet, which borrows the U-shaped structure of Swin-unet [17] and utilizes jump connections to reduce the loss of spatial information. In addition, to solve the problem of a large number of parameters and slow inference in Swin-unet, we adopt Pool-former based on Pool-attention instead of Swin-transformer in Swin-unet. Pool-attention is a simple and effective attention mechanism that is capable of learning both locally and globally by using Pooling instead of the traditional attention structure and without learnable parameters. This structure greatly reduces the number of parameters in the model [18]. In Pool-unet, we feed labeled images into the encoder-decoder architecture to learn local and global features and reduce the number of parameters. At the same time, we propose a Multi-task mask learning strategy to improve the model's generalization and anti-interference ability, wherein the first stage, we train a basic segmentation model using tongue data so that it learns basic image segmentation capabilities. In the second stage, we generate two mask generators based on the input image and perform masking operations on pixels. Next, we input the masked image into the already trained image segmentation model to perform segmentation and pixel prediction tasks. In this way, our model can learn semantic information and boundary details effectively, thus improving the accuracy of segmentation.

Figure 1 shows the experimental results of different models. The experimental results show that the number of parameters in Pool-unet is 5.78 MB and the MIOU is 97.54% and Pool-unet inference speed of 59 ms/piece. The main contributions of this paper are listed as follows:

(1) We propose a novel Pool-unet, which is a symmetric encoder-decoder U-shape network with skip connections constructed based on Pool-former blocks and up-sampling using the patch extension layer and down-sampling using the patch embedding layer to maintain spatial resolution. This novel model enhances contextual feature extraction while substantially reducing the network's number of parameters.

(2) We introduce a Multi-task training strategy including Multi-task pre-training and self-supervised fine-tuning stages based on the image mask reconstruction mechanism, which further improves the model's generalization ability and convergence speed.

We will present our methodology in Sect. 2 and our experimental results and analyze the results in Sect. 3 and give the conclusion of this study in Sect. 5.

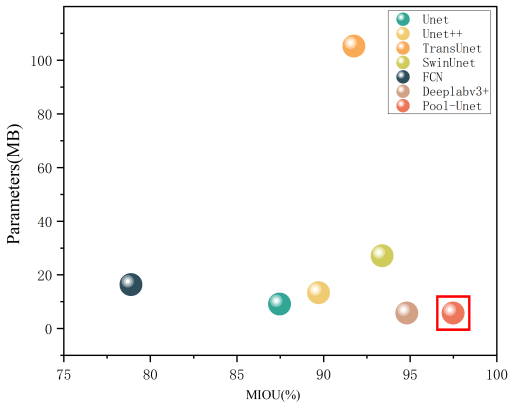


Fig. 1 MIOU-parameters scatterplot.

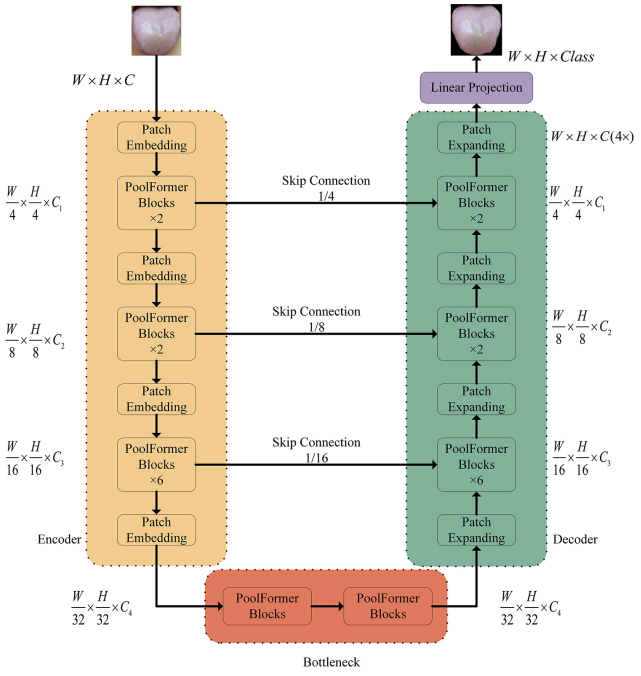


Fig. 2 Pool-unet structure diagram.

2. Methods

First, we will introduce the design structure of each part of Pool-unet in detail, as shown in Fig. 2. Then, we will introduce the Multi-task mask training method, as shown in Fig. 4.

2.1 Structural Design of Pool-Unet

The overall network architecture of Pool-unet proposed in this paper is shown in Fig. 2. Pool-unet is a network of the encoder, decoder, bottleneck, and skip connections. The basic building blocks of Pool-unet are called Pool-former blocks. The combination of the encoder and bottleneck layer forms a hierarchical structure with four stages. Suppose the

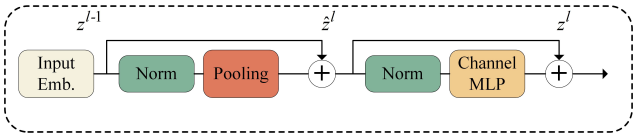


Fig. 3 Pool-former block.

shape of the input image is $H \times W \times C$. In the ith stage, the patch embedding layer will convert the input image into a feature map of $\frac{H}{\text{stride}_i} \times \frac{W}{\text{stride}_i} \times C_i, (\text{stride}_i = [4,2,2,2], C_i = [64,128,256,512])$ through convolution and normalization. We will discuss the effect of different embedding dims on the model performance in Sect. 3.4. Phase $[4,2,2,2]$ has $[2,2,6,2]$ Pool-former blocks. Subsequently, the feature map will be fed into Pool-former Blocks for feature extraction. Inspired by Swin-unet, we designed a decoder symmetric to the encoder based on Pool-former blocks. In the decoder section, we use three skip connections at 1/4 resolution, 1/8 resolution, and 1/16 resolution. This approach can fuse contextual features with multi-scale features from the encoder, thus compensating for the spatial information lost during the down-sampling process. In this way, we can better integrate feature information at different scales to achieve better image segmentation. In addition, a patch-expanding layer is designed in the decoder for up-sampling. Patch expanding expands the length and width of the input feature image by a factor of two and changes the feature dimensions to the dimensions required by the next layer of Pool-former blocks. After that, a patch expanding layer is utilized for 4-fold up-sampling to restore the resolution to that of the input image ($H \times W$). Finally, a linear projection layer is utilized for pixel-level prediction of the feature maps obtained from the up-sampling.

2.1.1 Pool-Former Blocks

Unlike traditional multi-head self-attention (MSA) modules, the Pool-former module is constructed based on a Pooling [18]. The Pool-former Block consists of two Normalization-Layers (Norm), a Pooling-Layer (Pooling), an MLP-Layer with GELU non-linearity, and two residual connections. The design of this module has several advantages. First, the normalization layer is used to alleviate the gradient vanishing problem and can simplify the tuning process, making the model more stable. Second, a spatial pooling operator without parameters is utilized as the token mixer module, which replaces the multi-attention token in the traditional Transformer and greatly reduces the number of parameters in the model. This pooling operation allows even aggregation of information around each token and does not involve learnable parameters. Finally, the MLP layer introduces a nonlinear mapping, which enhances the expressive and learning capabilities of the model. The Pool-former Block can be expressed as:

$$\hat{z}^l = z^{l-1} + \text{Pooling}\left(\text{Norm}\left(z^{l-1}\right)\right) \tag{1}$$

$$z^l = \hat{z}^l + \text{MLP}\left(\text{Norm}\left(\hat{z}^l\right)\right) \tag{2}$$

For the input sequence, the pooling operator is computed as follows, where K is the size of the pooling kernel.

$$T'_{:,i,j} = \frac{1}{K \times K} \sum_{p,q=1}^{K} T_{:,i+p-\frac{K+1}{2},i+q-\frac{K+1}{2}} \tag{3}$$

2.1.2 Encoder

The encoder is a layered structure with three layers, each consisting of a patch embedding layer and $[2,2,6]$ Pool-former blocks. The different layers of the patch embedding layer will divide the input image into different sizes, which are feature maps of one-quarter resolution (feature dimension C_1), one-eighth resolution (feature dimension C_2), and one-sixteenth resolution (feature dimension C_3). The feature maps at each scale are then subjected to a certain number of Pool-former blocks for further feature extraction. These Pool-former blocks are used to extract more advanced semantic features from the input feature maps. Finally, a separate patch embedding layer is set up to process the resolution of the image into the feature dimensions needed by the bottleneck.

2.1.3 Bottleneck

Since the Transformer is too deep to converge [19], only two consecutive Pool-former blocks are used to construct the bottleneck in this paper. We discuss the effect of the number of Pool-former blocks in the bottleneck layer on the model segmentation accuracy in Sect. 3.4. Pool-former block with an embedding of 512 is used to construct a bottleneck to alleviate the problem of the model being too deep to converge. There is no change in the feature dimension of the resolution of the image in the bottleneck.

2.1.4 Decoder

The decoder layer is similar to the encoder layer and also consists of multiple Pool-former blocks. However, unlike the encoder, the decoder replaces the encoder's patch embedding layer with a patch-expanding layer, which up-samples feature maps of neighboring dimensions by a factor of two to produce feature maps with higher resolution. At the same time, the patch expanding layer reduces the feature dimensions to half their original size. This operation helps to recover a finer representation of the features in the Decoder layer, thus improving the model's ability to capture details.

Patch expanding layer: Taking the first patch expanding layer as an example, a linear layer is applied to the input feature map $\left(\frac{H}{32} \times \frac{W}{32} \times C_3\right)$ to expand the feature dimensions to twice the original dimensions $\left(\frac{H}{32} \times \frac{W}{32} \times 2C_3\right)$ before performing the up-sampling operation. Then a Rearrange operation is applied to reduce the feature dimensions to a quarter of the input dimensions $\left(\frac{H}{32} \times \frac{W}{32} \times 2C_3 \rightarrow \frac{H}{16} \times \frac{W}{16} \times C_2\right)$.

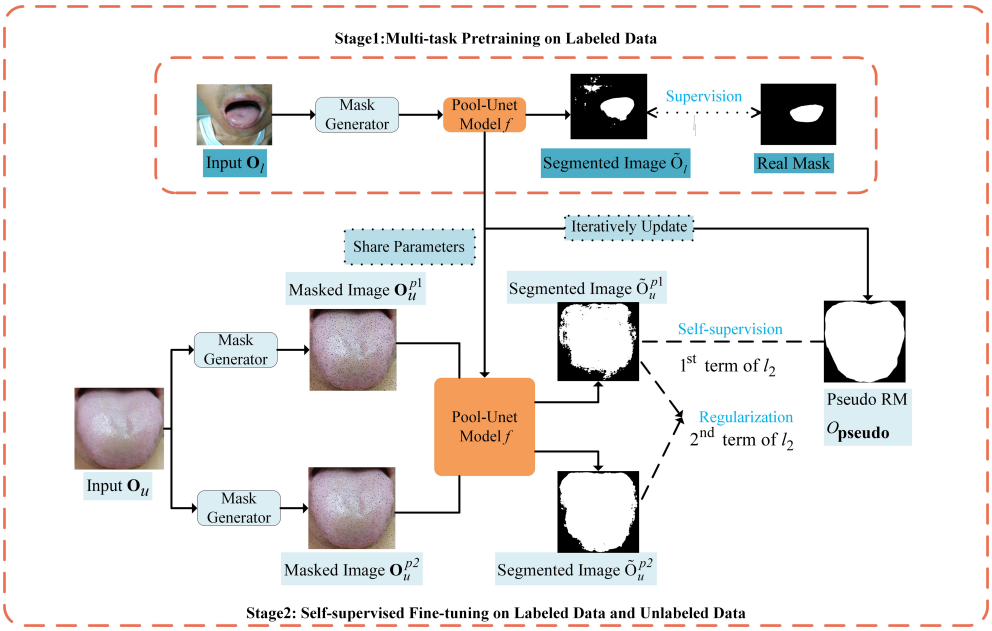


Fig. 4 The overview of the multi-task mask learning strategy. The same model architecture is shared in both pre-training and fine-tuning, except for the loss function definition. In the pre-training process, the model is trained on labeled data and learns tongue segmentation capabilities through supervision by l_1 (Eq. (8)). In the fine-tuning process, the model is initialized with pre-trained parameters and fine-tuned by the $l_{\text{fine-t}}$.

2.1.5 Skip Connection

Skip connection is used to fuse multi-scale features in the encoder with up-sampled features. Shallow and deep features are connected to minimize the loss of spatial information caused by down-sampling. We will verify the effect of skip connections at different locations on the model performance in Sect. 3.4.

2.2 Multi-Task Mask Learning

2.2.1 Overall Process

As illustrated in Fig. 4, to enhance the robustness of the model and improve the capability of tongue image segmentation in complex environments, this paper proposes a multi-task mask learning strategy to train the tongue image segmentation model. The method consists of two stages. In the first phase, we pre-train the Pool-unet model using tongue image data that has not been processed by the Mask Generator to give it basic tongue image segmentation capabilities. A Multi-task learning strategy was used for pre-training, and the loss function was divided into two parts: an image segmentation task and an image restoration task. In this way, the model can learn basic features and patterns of tongue image segmentation from labeled images. In the second phase, we introduce two different mask generators and use them to mask the pixels of the input image. Then, we input these masked images into the already pre-trained image segmentation model for tongue segmentation and pixel prediction tasks, at the same time, we can use the feature information of the labeled images to supervise the model and continuously improve its segmentation capability.

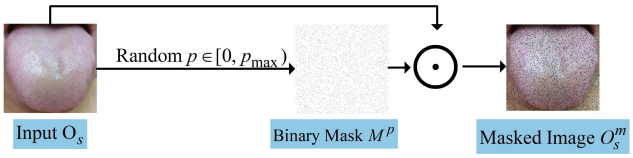


Fig. 5 Example for masked image generation.

2.2.2 Multi-Task Pretraining Stage

We first train the Pool-unet model by taking images without masked pixels as inputs to give it preliminary image segmentation capabilities.

Mask Generator: We describe how to generate a mask for the input image. First, we need to set a maximum masking percentage p_{max}, which must take a value less than one. Next, we randomly select an actual masking percentage p from the interval $[0,p_{max}]$. Then, we generate an all-1 matrix of the same size as the input image and randomly set the elements of the p percentage therein to 0. The resulting matrix is then called the masked matrix, which we name $\mathbf{M}^p$. Then we let $\mathbf{O}_l$ multiply element by element with $\mathbf{M}^p$ to obtain the masked image $\mathbf{O}_l^m$. Figure 5 shows an example of mask generation.

$$\mathbf{O}_l^m = \mathbf{O}_l \odot \mathbf{M}^p \tag{4}$$

Multi-Task Learning: For a given masked image $\mathbf{O}_l^m$,

we input it into the Pool-unet model for a tongue image segmentation operation:

$$\tilde{\mathbf{O}}_l = f\left(\mathbf{O}_l^m\right) \tag{5}$$

$\tilde{\mathbf{O}}_l$ is the segmented tongue image and $\mathbf{O}$ is the corresponding label.

$$\tilde{\mathbf{O}}_l^o + \tilde{\mathbf{O}}_l^m = \tilde{\mathbf{O}}_l, \tilde{\mathbf{O}}_l^o = \tilde{\mathbf{O}}_l \odot (\mathbf{1} - \mathbf{M}^p), \tilde{\mathbf{O}}_l^m = \tilde{\mathbf{O}}_l \odot \mathbf{M}^p \tag{6}$$

$$\mathbf{O}^o + \mathbf{O}^m = \mathbf{O}, \mathbf{O}^o = \mathbf{O} \odot (\mathbf{1} - \mathbf{M}^p), \mathbf{O}^m = \mathbf{O} \odot \mathbf{M}^p \tag{7}$$

$\tilde{\mathbf{O}}_l^o$ and $\tilde{\mathbf{O}}_l^m$ together form the image $\tilde{\mathbf{O}}_l$ after model segmentation. $\mathbf{O}^o$ and $\mathbf{O}^m$ together form the input image $\mathbf{O}$. $\mathbf{O}^o$ is the matrix of pixels that are not masked, and $\mathbf{O}^m$ is the matrix of pixels that are masked. $\mathbf{1}$ is an all-$\mathbf{1}$ matrix. The Pool-unet model is trained by minimizing the following loss.

$$l_1 = \left\|\tilde{\mathbf{O}}_l - \mathbf{O}\right\|_1 = \underbrace{\left\|\tilde{\mathbf{O}}_l^o - \mathbf{O}^o\right\|_1}_{\text{image segmentation task}} + \underbrace{\left\|\tilde{\mathbf{O}}_l^m - \mathbf{O}^m\right\|_1}_{\text{image restoration task}} \tag{8}$$

The first term of l_1 loss is a segmentation task performed on pixels that are not masked and the second term is a repair task performed on pixels that are corrupted by the mask. Therefore, the task is multitask learning.

2.2.3 Self-Supervised Fine-Tuning Stage

To enhance the model's generalization ability and robustness in the complex and diverse shooting environment, we adopted a self-supervised approach for fine-tuning, as illustrated in Fig. 4. This approach aims to improve the model's performance by leveraging a specific training objective. The training objectives is as follows:

$$l_2 = \underbrace{\left\|\tilde{\mathbf{O}}_u^{p_1} - \mathbf{O}_{\text{pseudo}}\right\|_1}_{\text{self-supervisory term}} + \underbrace{\alpha \left\|\tilde{\mathbf{O}}_u^{p_1} - \tilde{\mathbf{O}}_u^{p_2}\right\|_1}_{\text{Regularization term}}, \alpha \in [0, 1] \tag{9}$$

where $\mathbf{O}_{\text{pseudo}} = f(\mathbf{O}_u)$ denotes the segmentation result of the segmentation model f on the input image $\mathbf{O}_u$ at each training iteration of the model. The input image $\mathbf{O}_u$ is covered by $\mathbf{M}^{p_1}$ and $\mathbf{M}^{p_2}$ generated by two different mask percentages p_1, p_2, which are then fed into the segmentation model to obtain two different segmentation results $\tilde{\mathbf{O}}_u^{p_1}$ and $\tilde{\mathbf{O}}_u^{p_2}$. The loss function l_2 contains both self-supervision and regularization components. Where α is the weighting parameter used to balance these two terms. The effect of the value of α on the model performance we will discuss in Sect. 3.4.

$$\tilde{\mathbf{O}}_u^{p_1} = f\left(\mathbf{M}^{p_1} \odot \mathbf{O}_\mathbf{u}\right) \tag{10}$$

$$\tilde{\mathbf{O}}_u^{p_2} = f\left(\mathbf{M}^{p_2} \odot \mathbf{O}_\mathbf{u}\right) \tag{11}$$

Self-supervision: In the first term of l_2, the initial segmentation result $\mathbf{O}_{pseudo}$ obtained from each iteration of the segmentation model is used as pseudo labels. These pseudo labels are used as noise ground truth for self-supervised training. This self-supervised training method can learn more details from labeled data. Since the model is initialized with a loss of 0 using a pre-trained model, training will not continue if only this part is included in the loss function. To avoid this problem, we include a regularization term in the l_2 loss function.

Regularization: The second term of l_2 is used for the training of the regularization model. Its goal is to minimize the difference between two segmentation results $\tilde{\mathbf{O}}_u^{p_1}$ and $\tilde{\mathbf{O}}_u^{p_2}$ of the same image $\mathbf{O}_u$ corrupted with different masks, with α being the regularization weight.

Both self-supervision and regularization in l are essential. The training of the model cannot be adapted to unlabeled data without a regularization term; and in the absence of a self-supervision term, the model will ignore features learned from the labeled data, which in turn leads to the degradation of the model performance or early convergence.

To improve the stability of the fine-tuning process, we let the labeled data be trained together with the unlabeled data. As a result, the total loss function $l_{\text{fine-t}}$ for fine-tuning is as follows:

$$l_{\text{fine-t}} = \beta \times l_1 + (1 - \beta) \times l_2 \tag{12}$$

The role of β is to balance the effects of l_1 and l_2 on model fine-tuning. We will discuss the effect of the value of β on the model segmentation accuracy in Sect. 3.4.

Loss function: In this paper, a combination of Dice-Loss and Cross-Entropy-Loss is used as the loss function. This choice is made because Dice-Loss mitigates the negative effects caused by the imbalance between foreground and background, while Cross-Entropy-Loss calculates the loss equally for each pixel point, relating only to the difference between the current predicted value and the true labeled value. By using a combination of these two loss functions, better training results can be achieved, overcoming the problem of loss saturation that dice loss may experience when used alone. The loss function for the whole training process in this paper is as follows.

$$l_{train} = diceloss + celoss + l_{\text{fine-t}} \tag{13}$$

$$l_{valuate} = diceloss + celoss \tag{14}$$

Dice-Loss is calculated as follows:

$$diceloss = 1 - \frac{2\sum_{i=1}^{N} y_i \hat{y}_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i} \tag{15}$$

where y_i and $\hat{y}_i$ denote the labeled and predicted values of pixel i, respectively, and N is the total number of pixel points, which is equal to the number of pixels in a single image multiplied by the batch size.

The formula for calculating Cross-Entropy-Loss for the second classification is as follows:

$$celoss = -\frac{1}{M+N}\left(\sum_{i=1 \text{if} y_i=1}^{M} \log(p_i) + \sum_{i=1 \text{if} y_i=0}^{M} \log(1 - p_i)\right) \tag{16}$$

where M is the number of positive samples, N is the number of negative samples, y_i is the true value and p_i is the predicted value.

Overall Training Procedure: We summarize the entire training process in Algorithm 1.

Algorithm 1 Training of a deraining model

//Pre-training

Prepare : $\{\mathbf{O}_l^k\}_{k=1}^N$, $\{\mathbf{O}^k\}_{k=1}^N$ from labeled dataset.

while $epoch \leq epoch_{max}$ **do**:
  Randomly crop training image pairs $\{\mathbf{O}_l, \mathbf{O}\}$
  Generate a mask $\mathbf{M}^p$ using Mask Generator
  Generate masked rainy images $\{\mathbf{O}_l^m\}$ using Eq.(4)
  Obtain intermediate derained images $\tilde{\mathbf{O}}_l$
  Minimize loss in Eq.(8)
  $epoch = epoch + 1$
end while
Output the pre-trained deraining model f_{pre-t}

//Fine-tuning

Prepare : $\{\mathbf{O}_l^k\}_{k=1}^N$, $\{\mathbf{O}^k\}_{k=1}^N$ from labeled data, $\{\mathbf{O}_r^k\}_{k=1}^N$ from real-label data. Model f_{pre-t}

Initialize a model f using the parameters from f_{pre-t}
while $iter \leq iter_{max}$ **do**:
  Update $\mathbf{O}_{pseudo} = f(\mathbf{O}_u)$
  while $epoch \leq epoch_{u_{max}}$ **do**:
    Generate masked images $\mathbf{O}_{ut}^{p_1}$ and $\mathbf{O}_{ut}^{p_2}$ with different masks $\mathbf{M}^{p_1}, \mathbf{M}^{p_2}$
    Obtain deraining results $\tilde{\mathbf{O}}_u^{p_1}$ and $\tilde{\mathbf{O}}_u^{p_2}$ using f
    Obtain deraining result $\tilde{\mathbf{O}}_l$ for labeled data using Eq.(4) and Eq.(5)
    Minimize loss in Eq.(12)
    $epoch = epoch + 1$
  end while
  $iter = iter + 1$
end while
Output f as the final deraining model $f_{\text{fine-t}}$

Fig. 6 Examples of the dataset.

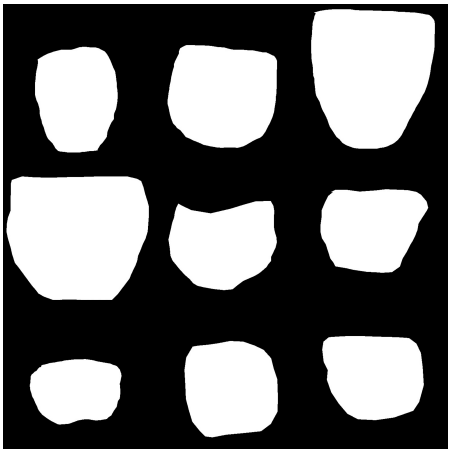


Fig. 7 Samples from the mask set corresponding to the dataset in Fig. 6.

3. Experimental Results and Analysis

3.1 Dataset Description

In this paper, a self-built tongue dataset is used for model training and testing. The dataset contains 764 tongue images from different populations, each with 24-bit depth RGB channels and a resolution of 512 × 512. This dataset is unique because of the complexity and diversity of the environments in which it was captured and the diversity of the tongue states of the different populations included. We use the LABELME tool for image annotation. The first round of labeling was performed on the images, and a professional doctor reviewed the results. Then, for the pictures that do not pass the review, based on the first round of labeling, especially for the location of the edge of the tongue and the tongue-lip-dental contact location for the second round of labeling. The second round of labeling of the images that did not pass the review was performed by the doctor for final labeling. Once the images were annotated, the LABELME tool could export the annotations in JSON format. These files include spatial information and category labels associated with each annotated object. Finally, we can utilize this JSON file to convert the original images into binary images, serving as our labels. Some examples of the data are shown in Fig. 6.

The mask image is a binary image with 255 (white) pixels in the foreground and 0 (black) pixels in the background. The mask image corresponding to the sample in Fig. 6 is shown in Fig. 7.

3.2 Experimental Details

Pool-unet is implemented based on Pytorch 2.0.0 and Python 3.8. We trained the model on an RTX 3090 with 24GB of video memory. Weight-Decay=1e-7, Momentum is 0.9, Learning Rate=1e-4, epochs=250, and K-fold cross-validation=5. The value of α in l_2 is 0.4. The value of β in l_{fine-t} is 0.4.

3.3 Evaluation Metrics

In this paper, MPA and MIOU are used as the indexes for

evaluating segmentation accuracy, and both MPA and MIOU are extremely large variables, with larger values representing higher segmentation accuracy of the model.

$$MOU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{TP}{TP+FP+FN} \tag{17}$$

$$MPA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{TP+TN}{TP+TN+FP+FN} \tag{18}$$

where TP is the actual number of positive samples predicted to be positive, TN is the actual number of negative samples predicted to be negative, FP is the actual number of samples predicted to be positive, FN is the actual number of positive samples predicted to be negative, and k is the number of categories.

Total parameters: It is the total number of parameters included in the model, which can intuitively reflect the size of the model.

Inference time: It reflects the time required for the model to infer the results, the shorter the inference time, the faster the model reasons about the results.

3.4 Selection of Hyperparameter

Impact of the number of skip connections on model performance: We add skip connections at positions of 1/4, 1/8, and 1/16 resolution of Pool-unet. In the experiments, the effect of the model on the segmentation performance of the tongue dataset was investigated by modifying the number of skip connections to 0, 1, 2, and 3, as shown in Table 1.

As can be seen from Table 1, the addition of three skip connections at 1/4, 1/8 and 1/16 resolution maximizes the possibility of incorporating low-resolution features, resulting in better model performance.

Impact of embedding dim size in encoder on model performance: The Pool-former network proposed by YU [18] has two sizes, S and M. The embedding dim in each layer of patch embedding in the M-size model is $[96, 192, 384, 768]$. Next, we discuss the effect of the number of embedding dims for M-size versus the number of embedding dims for $[64, 128, 256, 512]$ used in this paper on the model's performance.

From Table 2, it can be seen that the number of parameters of the Pool-unet structure with embedding dim number of $[64, 128, 256, 512]$ proposed in this paper is only 45.1% of that of the Pool-unet structure with embedding dim number of $[96, 196, 384, 768]$, but it achieves the same segmentation performance. Therefore We choose the network structure with an embedding dim of $[64, 128, 256, 512]$ in this paper.

The effect of the number of Pool-former blocks in the bottleneck layer: We discuss the impact of a bottleneck layer consisting of 0 (no bottleneck layer), 1, 2, and 3 Pool-former blocks on model performance. It can be seen from Fig. 8 that the model achieves the maximum segmentation accuracy when the bottleneck layer consists of two Pool-former blocks.

Table 1 Impact of the number of skip connections on model performance.

Quantities	Connected Position	MPA(%)	MIOU(%)
0	Non skip connections	78.72	62.84
1	Retain only 1/4 resolution skip connections	97.01	95.1
	Retain only 1/8 resolution skip connections	97.19	95.41
	Retain only 1/16 resolution skip connections	97.3	95.51
2	Retains two skip connections at 1/4, 1/8 resolution	96.15	94.07
	Retains two skip connections at 1/4, 1/16 resolution	96.99	96.12
	Retains two skip connections at 1/8, 1/16 resolution	97.13	96.39
3	Retain three skip connections at 1/4, 1/8, and 1/16 resolution	97.54	97.19

Table 2 Effect of embedding dim number on model performance.

Embedding Dim Number	Parameters	MPA(%)	MIOU(%)
[64,128,256,512]	5.78MB	97.19	97.54
[96,192,384,768]	12.96MB	97.17	97.15

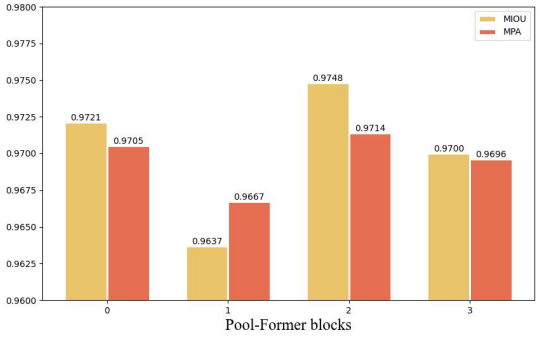


Fig. 8 Impact of the number of Pool-former blocks in the bottleneck layer on model performance.

The effect of weight coefficient α in the regularization term on the model: In l_2 loss, regularization weights α are needed to balance the degree of model regularization. We randomly selected 10 values of α for the experiment.

As can be seen from Fig. 9, both MIOU and MPA reach their maximum values when 0.4.

The effect of the value of β in l_{fine-t} on the model performance: The role of β in l_{fine-t} is to balance the proportion of labeled and unlabeled data in the fine-tuning process to ensure a more stable fine-tuning process. To determine the proportion of labeled and unlabeled data in the fine-tuning process, we did the following experiments on β.

We chose a total of nine values from 0.1 to 0.9 to discuss the effect of the value of β in l_{fine-t} on the model performance. For the cases where β takes 0 or 1, we consider them as ablation experiments for l_1 and l_2, which we will discuss in detail in Sect. 3.5. It can be seen from Fig. 10 that the model has the highest segmentation accuracy when β takes the value of 0.4.

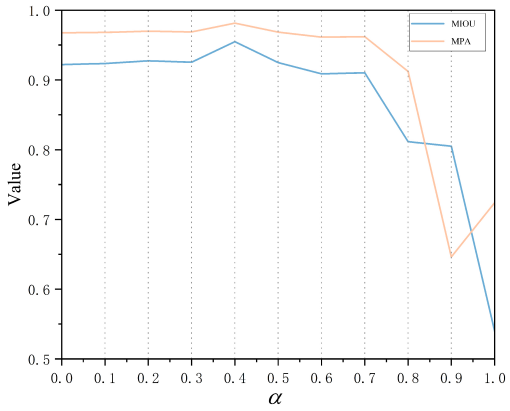


Fig. 9 Impact of α on model performance.

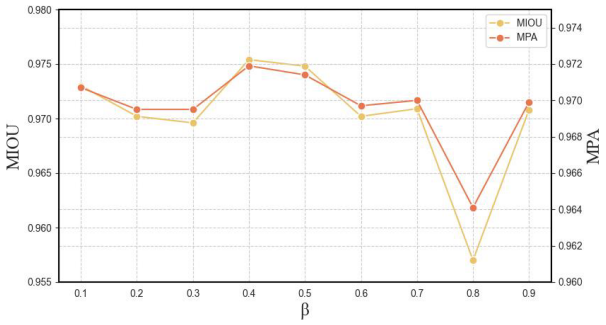


Fig. 10 Impact of β on model performance.

Impact of L1-norm and L2-norm on model performance: We have tried using L1-norm and L2-norm in Eq. (8) and Eq. (9), respectively, while ensuring that all other conditions remain the same, and comparing the performance of the models. The results indicate that using the L1-norm yielded slightly better segmentation performance compared to L2-norm. Specifically, with L1-norm, the MPA and MIOU are 97.54% and 97.19%, respectively, while with L2-norm, they are 96.94% and 96.21%.

3.5 Ablation Experiments

Impact of l_1 and l_2 on model performance: In order to explore the effect of Eq. (8) l_1 and Eq. (9) l_2 on the model segmentation accuracy, we choose to remove l_1 or remove l_2. The results are shown in Table 3. When l_1 and l_2 are not added, it is equivalent to not using a Multi-task mask learning strategy in the model, and the segmentation accuracy of the model is the lowest in this case. When only l_1 is added, the MPA and MIOU of the model are improved by 0.5% and 0.14% respectively. The MPA and MIOU of the model are improved by 0.63% and 0.21% respectively when only l_2 is added. This shows that l_1 and l_2 alone do not improve the model performance significantly. However, the MPA of the model improves by 2.08% and 2.11% when both l_2 and l_1 are added. Therefore, the Multi-task mask learning strategy can effectively improve the segmentation accuracy of the model.

Table 3 Impact of l_1 and l_2 on model performance.

Model	MPA(%)	MIOU(%)
w/o l_1 & w/o l_2	95.55	95.18
w/o l_1 & w l_2	96.03	95.31
w/o l_2 & w l_1	96.16	95.38
w/ l_1 & w/ l_2	97.54	97.19

Table 4 Comparison of parametric quantities and inference time of common segmentation models.

Methods	Parameters(MB)	Inference Time(ms/piece)
Unet [20]	9.16	58
Unet++ [21]	13.39	128
Trans-unet [22]	105.28	–
Swin-unet [17]	27.17	106
FCN [23]	16.4	–
Deeplabv3+ [24]	5.81	324
OET-NET [25]	7.78	59
Pool-unet(ours)	5.78	46

3.6 Comparison with SOTA Approaches

Computer-aided tongue diagnosis algorithms require smaller memory space lower inference time and high segmentation accuracy [2]. Therefore, the speed of inference and the total number of parameters of the model need to be considered while improving the segmentation accuracy [12]. In Sect. 3.6.1 we will use the example of 512×512×3 images to compare with the SOTA models in terms of both the number of model parameters and inference time. In Sect. 3.6.2 we use MIOU and MPA to measure the segmentation accuracy of the model.

3.6.1 Parameter Count and Inference Time Evaluations

Since OET-NET is the model with the highest segmentation accuracy achieved in this task, we use it as a benchmark to compare with our model. OET-NET has a large number of convolution and de-convolution operations, which require a large number of computations and parameters. But our model has average pooling as mixed tokens in pooling attention and only one convolution operation in Patch Embedding. So the number of parameters and inference time of our model is less than OET-NET. Table 4 also confirms this view.

As shown in Table 4, the model parameter of our proposed model is just 5.78 MB. It is 25.71% lower than the model parameter of the latest model (OET-NET). The reduction varies from 94.49% to 25.71% compared to other models. The time to infer a picture in the Pool-unet proposed in this paper is reduced by 22.03% over the latest model (OET-NET) and by 20.6% over the classical Unet network.

3.6.2 Performance Evaluations

We conducted prediction experiments using 100 test samples and used MIOU and MPA to measure the segmentation accuracy of the model. Table 5 shows that the proposed

Pool-unet outperforms other methods, especially in MIOU, which can reach 97.19%. The MIOU and MPA of the proposed method improve with other methods ranging from 0.60%–23.21% and 0.92%–15.21%.

4. Discussion

In this paper, we propose Pool-unet, a fast and accurate tongue segmentation network based on Pool-former and Multi-task learning. Various experiments were conducted to test the effectiveness of the proposed method. For example, the results of the ablation experiments show that the model has the highest segmentation accuracy when the bottleneck consists of two Pool-former blocks.

To test whether there is a significant difference in inference time between Pool-unet and the benchmark, we performed statistical analysis with inference time as the dependent variable. We used Pool-unet and OET-NET to predict ten tongue pictures recorded the inference time of each picture and performed a one-way analysis of variance (ANOVA) on these two data sets. The results show that F(1,18)=85.43, p=2.6e-8 for the inference time. This suggests that there is a significant difference in inference speed between Pool-unet and OET-NET, at the level of significance α =0.05. In addition, to assess whether there is a significant difference in the performance of the Pool-unet and OET-NET in processing the test set, we performed ten repetitions of the test recorded the values of MIOU and MPA for each model, and used ANOVA to statistically analyze the MIOU and MPA

Table 5 Comparison of segmentation performance of different methods.

Methods	MIOU(%)	MPA(%)
Unet	87.46	92.30
Unet++	89.71	92.52
Trans-unet	91.75	95.39
Swin-unet	93.38	94.24
FCN	78.88	84.66
OET-NET	96.62	96.65
Deeplabv3+	93.79	96.79
Pool-unet(ours)	97.19	97.54

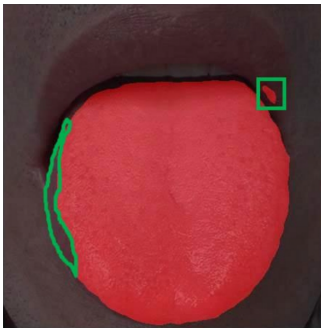


Fig. 11 Example of segmentation results.

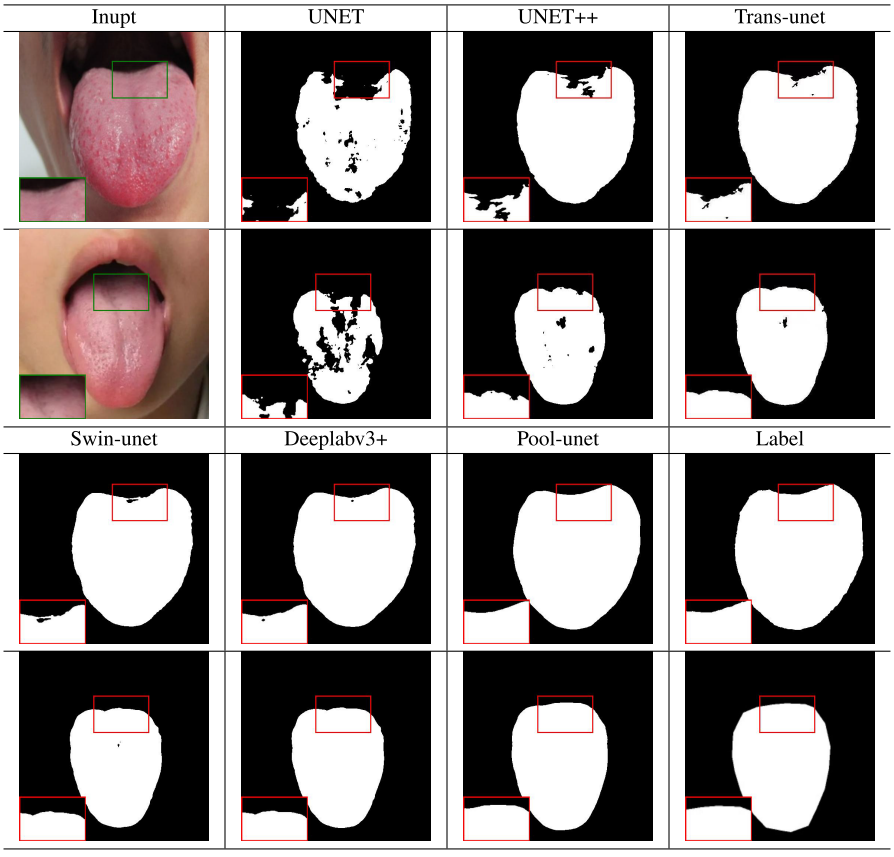


Fig. 12 Comparison of segmentation performance of different methods.

of the Pool-unet and the OET-NET. The results show that F(1,18)=11.254, p=0.0035 for MPA and F(1,18)=85.43, p= 2.96e-8 for MIOU. This suggests that there is a significant difference in segmentation accuracy between Pool-unet and OET-NET, at the level of significance α =0.05.

However, since the proximity of the tongue to the lips and teeth, it is difficult for our model to separate them perfectly and accurately. As shown in Fig. 11, it is also a major challenge to accurately segment the edges of the tongue, which will affect the computer-assisted tongue diagnosis system's ability to judge the smoothness of the tongue edges. For example, a tooth-marked tongue has a certain unevenness of the edge of the tongue body due to the compression of the tooth edge by the hypertrophy of the tongue body [26]. In the future, we will study this challenge in detail and further explore and apply the Pool-unet model for a wider range of applications in medical image segmentation.

5. Conclusions

In this study, we proposed a novel encoder-decoder network architecture called Pool-unet based on Pool-former and Multi-task mask learning. Instead of the traditional self-attentive structure, we use pool-attention, which greatly reduces the number of parameters in the model. In addition, we designed a U-shaped encoder and decoder network architecture and added skip connections at different resolution levels to better fuse low-resolution image features. In addition, we propose a model training method for Multi-task mask learning, which pre-trains the model using labeled data to give it basic image segmentation capabilities. Following this, two distinct mask generators are designed to execute masking operations on input images, and throughout the training iterations, the model undergoes fine-tuning with pseudo-label supervision. We conducted experiments on the tongue dataset, which has been labeled by ourselves, and Pool-unet achieved 97.19% and 97.54% on the MIOU and MPA metrics, respectively. The number of parameters of the model is only 5.78 MB and the inference speed reaches 46 (ms/piece).

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) (62201314, 62201571), Natural Science Foundation of Shandong Province (No. ZR2020QF007), and Key Technology Tackling and Industrialization Demonstration projects of Qingdao (23-1-2-qdjh18-gx).

References

[1] H.Z. Zhang, K.Q. Wang, D. Zhang, B. Pang, and B. Huang, "Computer aided tongue diagnosis system," 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, pp.6754–6757, 2005.

[2] M.H. Tania, K. Lwin, and M.A. Hossain, "Advances in automated tongue diagnosis techniques," Integrative Medicine Research, vol.8, no.1, pp.42–56, 2019.

[3] L. Zhang and J. Qin, "A tongue image segmentation method based on gray scale projection and automatic threshold selection," Tissue Engineering Research in China, pp.1638–1641, 2010.

[4] W. Qin, B. Li, and X. Yue, "A hybrid tongue image segmentation algorithm based on initialized snake contours," Journal of the University of Science and Technology of China, pp.807–811, 2010.

[5] J. Liu, L. Yin, J. Pan, Y. Cui, and X. Tang, "Edge detection algorithm for uneven illumination images based on parametric logarithmic image processing model," Laser & Optoelectronics Progress, vol.58, no.22, p.2210005, 2021.

[6] B. Lin, J. Xie, C. Li, and Y. Qu, "Deeptongue: Tongue segmentation via resnet," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.1035–1039, 2018.

[7] Y. Xue, X. Li, P. Wu, J. Li, L. Wang, and W. Tong, "Automated tongue segmentation in Chinese medicine based on deep learning," Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, Dec. 2018, Proceedings, Part VII 25, pp.542–553, 2018.

[8] S. Trajanovski, C. Shan, P.J.C. Weijtmans, S.G.B. de Koning, and T.J.M. Ruers, "Tongue tumor detection in hyperspectral images using deep learning semantic segmentation," IEEE Trans. Biomed. Eng., vol.68, no.4, pp.1330–1340, 2021.

[9] C. Zhou, H. Fan, and Z. Li, "Tonguenet: Accurate localization and segmentation for tongue images using deep neural networks," IEEE Access, vol.7, pp.148779–148789, 2019.

[10] B. Lin, J. Xle, C. Li, and Y. Qu, "Deeptongue: Tongue segmentation via resnet," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.1035–1039, 2018.

[11] J. Zhou, Q. Zhang, B. Zhang, and X. Chen, "Tonguenet: A precise and fast tongue segmentation system using U-Net with a morphological processing layer," Appl. Sci., vol.9, no.15, pp.3128–3147, 2019.

[12] Y. Kim, J. Oh, S.-H. Choi, A. Jung, J.-G. Lee, Y.S. Lee, and J.K. Kim, "A portable smartphone-based laryngoscope system for high-speed vocal cord imaging of patients with throat disorders: Instrument validation study," JMIR mHealth and uHealth, vol.9, no.6, p.e25816, 2021.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," Neural Information Processing Systems, Neural Information Processing Systems, 2017.

[14] Y. Jiang, X. Yu, Y. Wang, X. Xu, X. Song, and D. Maynard, "Similarity-aware multimodal prompt learning for fake news detection," SSRN Electronic Journal, 2023.

[15] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," Proc. IEEE/CVF International Conference on Computer Vision, pp.10012–10022, 2021.

[16] A. Lin, B. Chen, J. Xu, Z. Zhang, G. Lu, and D, Zhang, "DS-TransUNet: Dual swin transformer U-Net for medical image segmentation," IEEE Trans. Instrum. Meas., vol.71, pp.1–15, 2022.

[17] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-Unet: Unet-like pure transformer for medical image segmentation," European Conference on Computer Vision, pp.205–218, 2022.

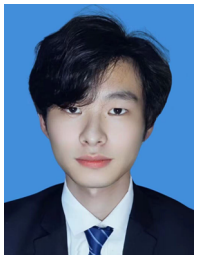
[18] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "MetaFormer is actually what you need for vision," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.10819–10829, 2022.

[19] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," Proc. IEEE/CVF International Conference on Computer Vision, pp.32–42, 2021.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, Oct. 2015, Proceedings,

Part III 18, pp.234–241, 2015.
[21] Z. Zhou, Md M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested u-net architecture for medical image segmentation," Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pp.3–11, 2018.
[22] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A.L. Yuille, and Y. Zhou, "TransUNet: Transformers make strong encoders for medical image segmentation," Cornell University - arXiv, Cornell University - arXiv, 2021.
[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[24] C. Wang, P. Du, H. Wu, J. Li, C. Zhao, and H. Zhu, "A cucumber leaf disease severity classification method based on the fusion of DeepLabV3+ and U-Net," Computers and Electronics in Agriculture, vol.189, p.106373, 2021.
[25] Z. Huang, J. Miao, H. Song, S. Yang, Y. Zhong, Q. Xu, Y. Tan, C. Wen, and J. Guo, "A novel tongue segmentation method based on improved U-Net," Neurocomputing, vol.500, pp.73–89, 2022.
[26] X. Li, Y. Zhang, Q. Cui, X. Yi, and Y. Zhang, "Tooth-marked tongue recognition using multiple instance learning and CNN features," IEEE Trans. Cybern., vol.49, no.2, pp.380–387, 2019.

Xiangrun Li is an undergraduate student at the School of Information Science and Technology at Qingdao University of Science and Technology, majoring in software engineering, and his research focuses on medical image segmentation.



Qiyu Sheng is an undergraduate student at the School of Information Science and Technology at Qingdao University of Science and Technology, majoring in artificial intelligence, and his research focuses on computer vision.

Guangda Zhou received the B.Eng. degree in computer science and technology from Qingdao University of Science and Technology in 2021. Currently, he is also a postgraduate student in the Department of Information Science and Technology at Qingdao University of Science and Technology, China.

Jialong Wei received the B.Eng. degree in Software engineering from Huaxia University of Technology in Wuhan, China in 2020. Currently, he is a postgraduate student in the Department of Electronic Information at Qingdao University of Science and Technology, China.

Yanmin Shi received the M.S. degree in Department of Information Science and Technology at Qingdao University of Science and Technology, China, in 2019. Currently, she is a senior engineer in the Department of Cloud Network of China Unicom, Qingdao, China.

Zhen Zhao received the the Ph.D. degree in systems engineering from Tongji University, China in 2011. Currently, he is an associate professor in the Department of Information Science and Technology at Qingdao University of Science and Technology, China. His research interests include speech emotion recognition, artificial intelligence and edge computing.

Yongwei Li received the M.S. and Ph.D. degrees in information science from the Japan Advanced Institute of Science and Technology, Nomi, Japan, in 2014 and 2018. Currently, he is an assistant professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Bejing, China.

Xingfeng Li received the M.S. degrees in software engineering and information science from Tianjin University, China, and Japan Advanced Institute of Science and Technology (JAIST), Japan, in 2016, respectively, and the Ph.D. degree in information science from JAIST, in 2019. Since 2022, he has been on the faculty of the School of Computer Science and Technology, Hainan University, Haikou, China, and is currentlyh an Associate Professor. His research interests are affective computing, speech processing, and speech perception, emphasizing how para/non-linguistic information (speech emotion) impacts spoken communication. He was a member of the Acoustical Society of Japan (ASJ) and the International Speech Communication Association (ISCA).

Yang Liu received the B.Eng. and M.Eng. degrees in computer science and technology from Tianjin University, China in 2010 and 2012, respectively, and the Ph.D. degree in information science from Japan Advanced Institute of Science and Technology (JAIST), Japan in 2016. Currently, he is an associate professor in the Department of Information Science and Technology at Qingdao University of Science and Technology, China.