Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 4, No. 1, 2025

Research on Embedded Tongue Image Segmentation Technology Based on Deep Learning

Heyue Jiang

School of Southwest Mizu University, Chengdu, Sichuan, China

Abstract: With the continuous development of deep learning technology in the field of medical imaging, objective and intelligent tongue diagnosis in Traditional Chinese Medicine has gradually become a research hotspot. However, due to the complex structure of tongue image features and their high subjectivity, as well as the limited computational power of embedded devices, achieving an efficient, accurate, and low-power tongue diagnosis system still faces many challenges. The goal of this paper is to explore a deep learning-based embedded tongue image segmentation method and to design an efficient low-power segmentation network. The segmentation of tongue images is fundamental to the objectification of tongue diagnosis in Traditional Chinese Medicine, as the accuracy of tongue body segmentation directly affects the identification of tongue color, coating color, and morphological analysis diagnosis in the tongue diagnosis system. Tongue image segmentation involves completely isolating the tongue body region along its edge contour from the image, so that the resulting tongue image contains all the image information of the tongue. During the segmentation process, when the color of the lips or skin around the tongue is similar to that of the tongue body, it increases the difficulty of segmentation, which can lead to low efficiency and accuracy in segmentation results, significantly impacting subsequent tongue image analysis. Therefore, this paper introduces an improved Attention UNet model designed with an attention mechanism, and based on this, it undergoes lightweight processing and optimization of training hyperparameters to achieve precise extraction of the tongue body area, while also considering segmentation performance and device compatibility. Ablation and comparative experiments are conducted on the improved components, followed by a horizontal comparison with currently popular and widely used networks, and finally, it is deployed on a Raspberry Pi hardware platform to verify the effectiveness and advancement of the improvements.

Keywords: Deep Learning, Lightweight, Segmentation, Embedded Deployment.

1. Introduction

Traditional Chinese Medicine (TCM) is an experience-based medical practice accumulated over thousands of years of human labor and life, reflecting people's wisdom. TCM diagnosis, guided by the foundational principles of TCM, is a method of exploring the condition and distinguishing diseases and syndromes for treatment65 [1]. It has the advantage of 'preventing illness before it occurs,' meaning it can detect diseases early or when they are just beginning, or adjust both the body and mind when a person is in a suboptimal health state, achieving timely prevention and early treatment. TCM is non-invasive, non-intrusive, and does not require mechanical assistance, with benefits such as simplicity, ease of operation, and convenience.

Tongue segmentation is a key step in tongue diagnosis. Currently, there are various algorithms for tongue body segmentation, mainly classified into two categories: traditional tongue body segmentation methods that include thresholding and edge detection, and deep learning-based tongue body segmentation techniques. The thresholding method [2] is the most basic image segmentation method, and threshold-based tongue body segmentation algorithms perform well on images where the target area and background area are distinctly different. The commonly used edge detection method is the Snake model, also known as the active contour model [3], which is a contour line capable of elastic deformation through its own template. When the energy function is minimized, the contour line closes to complete the image segmentation [4]. Traditional tongue body segmentation methods can achieve relatively good segmentation results to some extent, but they have a lower degree of automation and poorer accuracy. Therefore, with the improving performance of deep convolutional neural networks in recent years, deep learning-based segmentation techniques have also received increasing attention. The emergence of FCNs has led to the rise of various segmentation technologies based on fully convolutional networks. Subsequently, Ronneberger et al. [5] proposed the UNet architecture, which features an encoder-decoder structure based on FCNs. Despite achieving significant success, this network still faces issues such as a large number of parameters, long training times, and insufficient adaptability to large-scale variations. Furthermore, networks deployed on hardware platforms should have low power consumption characteristics to achieve a balance between accuracy and model efficiency. This paper proposes an improved method based on these considerations.

2. Basic Principles of Unet Network

The improved UNet based on FCN has shown good performance in deep learning with small sample sizes, especially in deep learning problems related to medical impact, where UNet plays a significant role. The entire UNet network structure resembles a U shape, hence the name UNet. The structure is illustrated in Figure 1. In terms of model structure, UNet is mainly composed of two parts; the left half performs convolution and pooling for downsampling the image, capturing contextual information within the image, while the right half employs convolution and pooling for upsampling, accurately locating and restoring the resolution of the features extracted from the left side for the segmentation task. Through Skip Connection, it combines high-level semantic information with low-level feature

information, resulting in better segmentation outcomes.

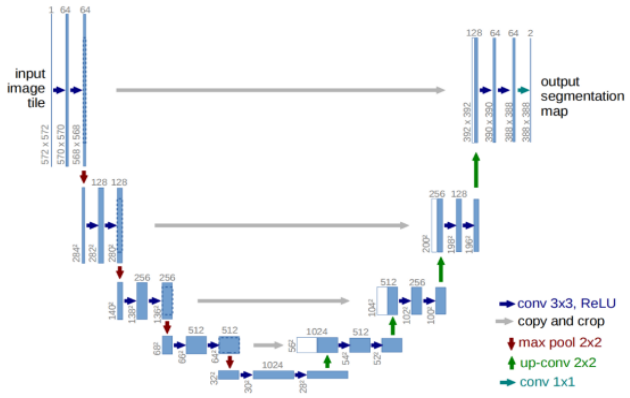


Figure 1. UNet Network architecture

3. Basic Principles of the Attention UNet Network

The structure of the Attention UNet network is shown in Figure 2. Its overall framework continues the classic design philosophy of UNet, primarily consisting of three parts: the encoder, the decoder, and skip connections. The difference between the two lies in the fact that Attention UNet introduces an attention mechanism into the basic encoder-decoder structure. During the upsampling process of feature maps in the decoding phase, the embedded Attention Gate module selectively focuses on the channels and spatial regions of the feature maps. This not only inherits the powerful contextual representation capabilities of the UNet encoder-decoder structure but also enhances the attention capabilities toward target areas. It can effectively filter out background information that is irrelevant or distracting to the current task, significantly improving the model's discriminative ability and segmentation accuracy. This network structure exhibits significant advantages in image segmentation tasks, particularly suitable for segmenting key organ regions with complex shapes, blurred boundaries, or small areas.

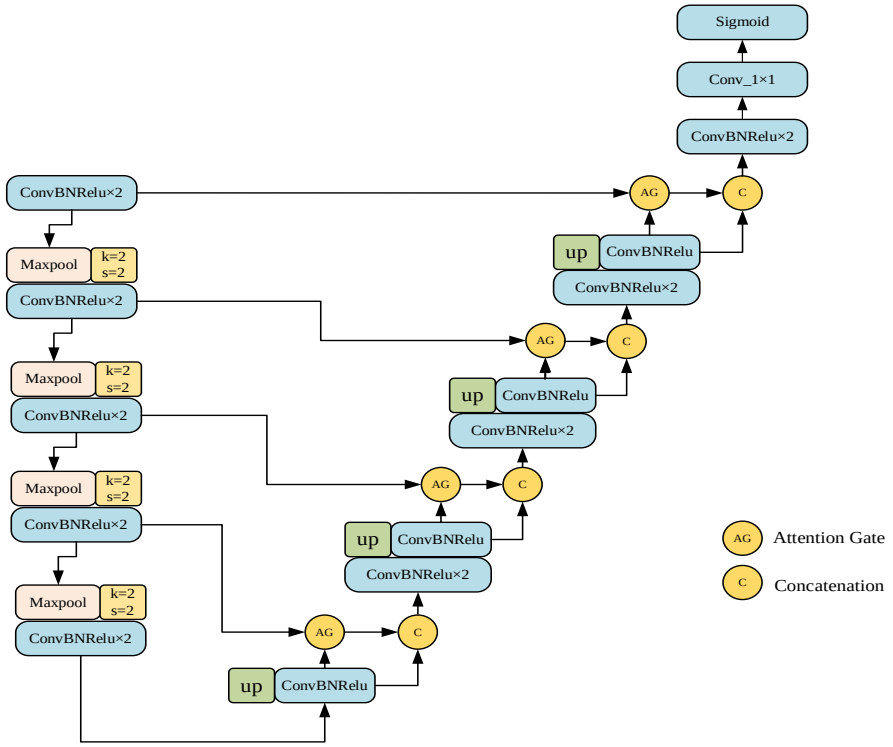


Figure 2. Attention UNet Network architecture

The core module, the Attention Gate, is capable of dynamically generating a set of attention weights based on the interactive relationship between high-level semantic features and low-level spatial features, which are used to explicitly filter the encoded feature maps. Its specific functional structure diagram is shown in Figure 3.

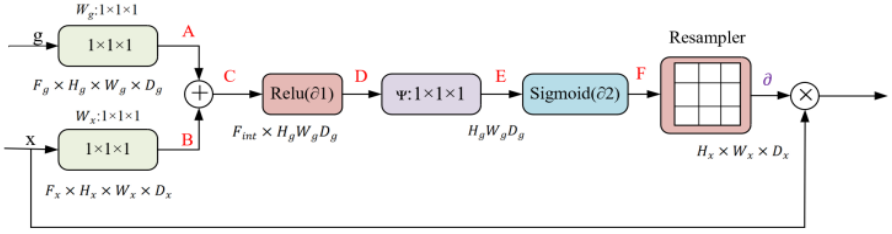


Figure 3. Attention Gate Specific Structural Diagram

He input consists of two parts: the feature map upsampled by the decoder (denoted as g); and the feature map corresponding to the encoder's layer (denoted as x). First, 1×1 convolutions are applied separately to these two feature maps g and x to compress their channels to an intermediate dimension, followed by normalization; then, the processed feature maps are added together and subjected to the ReLU activation function; next, a 1×1 convolution is used to compress the number of channels of the aforementioned feature maps to 1, followed by normalization and the Sigmoid

activation function, yielding the attention map α, which is an attention weight map of the same size as the input feature map (with values ranging from 0 to 1). This attention map measures the importance of each pixel position in the spatial dimension and serves as the output of the attention gate; finally, the attention weight map is multiplied by the encoder feature map to obtain the feature map weighted by attention, which is then concatenated with the decoder's feature map g. This effectively highlights the regions in the image relevant to the segmentation target while suppressing unrelated or distracting areas.

4. Improvement Based on Attention UNet Network

Although the Attention UNet network has demonstrated good segmentation performance in tongue image segmentation tasks, its model structure is relatively complex, with a large number of parameters and high computational costs, making it unsuitable for deployment on mobile terminals or resource-constrained devices in traditional Chinese medicine clinical auxiliary diagnosis. To meet the practical application requirements for lightweight, low-latency, and high real-time performance in tongue segmentation, this section proposes structural simplification and optimization of the Attention UNet model, thereby constructing a more efficient network structure.

4.1. Lightweight Network Construction

"Based on the Attention UNet, we introduce the GhostNet architecture, replacing the encoder part while simplifying the strategy through structural compression and attention mechanisms to achieve a lightweight model design. The main architecture is shown in Figure 4 below."

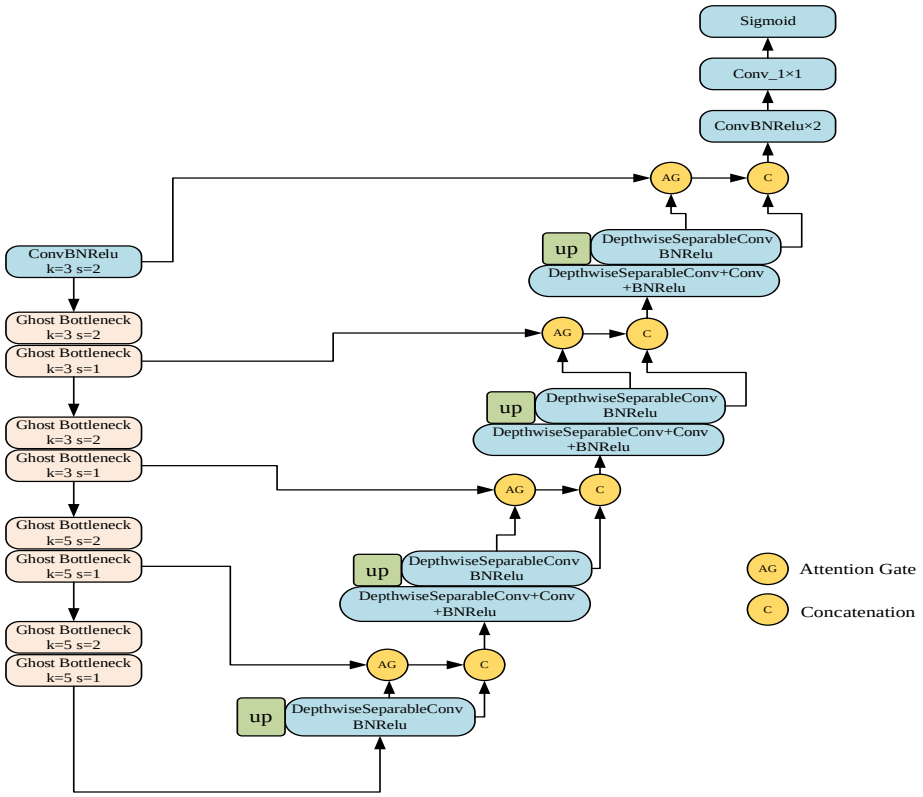


Figure 4. Ghost_Attention UNet structure

The original Attention UNet network's encoder part consists of multiple layers of standard convolutions. Although it has strong feature extraction capabilities, it comes with significant parameter redundancy and computational burden during actual inference. To improve the model's operational efficiency, this paper replaces that part with the GhostNet network. It is a lightweight neural network structure designed for mobile applications, with its core idea being the introduction of the Ghost Module. This module is composed of standard convolution, depthwise separable convolution, and shuffle operations. It uses only a small number of standard convolutions (1×1 Conv) to generate the basic feature map, and then generates the remaining redundant feature maps through inexpensive linear transformation operations (such as depthwise separable convolution). Finally, it concatenates the basic feature map with the redundant feature map obtained through linear transformation and performs a shuffle operation. The goal is to allow the information generated by the standard convolution to permeate into various parts of the information generated by the depthwise separable convolution, thereby reducing information loss and achieving faster operations. To this end, the Ghost Module achieves a certain degree of unification between the standard convolution and the depthwise separable convolution. This significantly reduces the number of parameters and floating-point calculations in the model, enhancing operational efficiency. The structural diagram of its Ghost Module is shown in the following Figure 5.

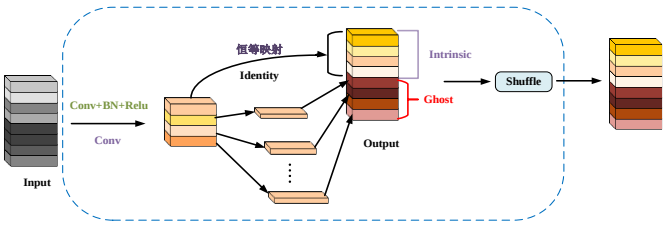
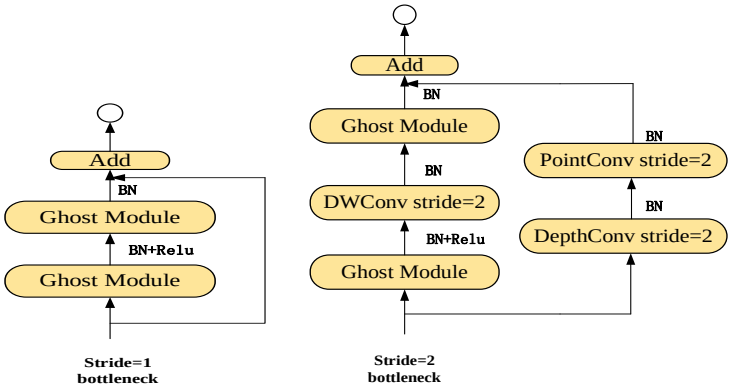


Figure 5. Ghost Module

The Ghost Module is the core component that forms the

Ghost Bottleneck, which in turn composes the GhostNet network. This network is specifically divided into two parts: stride=1 and stride=2, allowing for different depths and architectures of feature extraction on feature maps with varying strides. The architecture of this module is illustrated in Figure 6(a)(b).



(a) (b)

Figure 6. Ghost Bottleneck

For 6(a), it refers to feature maps with a stride of one. The primary function is to refine, fuse, or apply nonlinear transformations to the features without changing the spatial dimensions of the feature map. This is beneficial for extracting more complex feature patterns such as edges and textures. In contrast, 6(b) is similar to downsampling and serves the purpose of 'compressing space, expanding semantics,' providing high-level features rich in global information for the decoder. This function is similar to parts replaced in the current network, but with the addition of residual connections, which also help alleviate the vanishing gradient problem. Both parts with different strides have a structure that includes two ghost convolutions and residual connections. The first convolution mainly serves to increase the dimensionality of channels, while the second convolution compresses the channels, followed by residual addition. The difference is that for the part with stride two, after the first convolution, a depthwise convolution operation is required. This is mainly used for downsampling and reducing the size, allowing it to match the dimensions of the residual connections. The residual connections will also undergo dimensional (depthwise convolution) adjustments and channel (pointwise convolution) adjustments to ensure complete alignment with the main branch output. Compared to direct pooling, depthwise convolutions help preserve key features during size reduction by learning, thus minimizing information loss.

In this design, to match the feature maps of the decoder and achieve lightweight implementation, a moderate compression of the channel numbers was carried out, resulting in the output of five feature maps, starting from the first initial convolution, which began to compress dimensions and expand the number of channels. The subsequent four layers from layer1 to layer4 are composed of feature extraction blocks built with GhostBottleneck, and their quantity is limited to two: the first for downsampling and the second for feature refinement. These are then encapsulated into an ordered container to form a complete feature extraction layer, with the aim of achieving "compression of convolution depth" to avoid redundant computations typical of traditional deep networks.

"For the decoder part, firstly, the standard convolution in the upsampling is replaced with depthwise separable convolution to reduce the number of parameters; then, in the convolution block after upsampling, one standard convolution is replaced with depthwise separable convolution while keeping the other unchanged, forming a mixed convolution module; finally, the channel numbers of the decoder features g and encoder features x in the Attention Gate are compressed using depthwise separable convolution."

This part moderately reduces the number of channels and the depth of the convolutions, decreasing the redundant computations during the gradual restoration of feature map sizes. Additionally, depthwise separable convolutions are introduced in the fusion method of skip connections to reduce parameter overhead and enhance the fusion efficiency among features of different scales.

"After implementing the improvements mentioned above, the enhanced model maintains the structural symmetry of the Attention UNet encoding-decoding while achieving parameter compression and inference acceleration. This lays the foundation for subsequent optimization of training parameters for the network."

4.2. Improving Network Optimization

4.2.1. Introduce Regularization Optimization

Regularization in deep learning is adopted to prevent model overfitting by constraining model parameters, reducing model complexity, and improving its generalization ability. Common regularization functions include Dropout, Spatial Dropout, and DropBlock. ① Dropout regularization [7] enhances model generalization by randomly dropping a portion of neurons; ② Spatial Dropout [8] randomly drops entire channels (i.e., all output feature maps of a certain convolutional kernel) along the channel dimension, which preserves the spatial structure within single channels. This prevents the model from relying on a single channel feature, enhances robustness against channel loss, and encourages the model to learn more comprehensive inter-channel relationships. ③ DropBlock, introduced by Google in 2018 [9], differs from Dropout in that while Dropout randomly masks a portion of features, DropBlock masks a portion of contiguous regions, causing the network to lose overall information from that area, thus focusing more on learning features from other parts of the model.

The feature of tongue image segmentation lies in its spatial continuity. DropBlock improves the model's focus on the overall structure of the tongue by discarding contiguous spatial regions, thereby enhancing the learning of the tongue's overall shape.

4.2.2. Loss Function Optimization

Common loss functions used in segmentation tasks include cross-entropy loss, Dice loss, IOU loss, focal loss, and boundary loss, among others. Choosing the appropriate loss function can ensure segmentation accuracy while also achieving relatively smooth segmentation boundaries. In this experiment, we opted to use a variant of the binary cross-entropy loss function [10], known as BCE with Logits Loss, which is:

$$\mathrm{BCEWithLogitsLoSS}(x, y) = -y \cdot \log(\sigma(x)) - (1-y) \cdot \log(1-\sigma(x)) \tag{2.1}$$

Assume x is the model's output (Logits), and y is the true label (0 or 1), representing the sigmoid activation function, where the probability prediction value is obtained through the sigmoid activation function.

This loss function integrates the Sigmoid function and its numerical stability handling based on BCE, making both forward propagation and backward gradients more stable, effectively avoiding issues such as gradient vanishing or explosion commonly found in traditional BCE. In practical training, this function significantly enhances convergence speed and final segmentation accuracy, particularly excelling in edge localization and complex background removal.

4.2.3. Dynamic Adjustment of Learning Rate

"To enhance the model's convergence stability while ensuring training efficiency, this paper selects the RMSprop (Root Mean Square Propagation) optimizer [11]. This optimizer is a commonly used adaptive learning rate algorithm that dynamically adjusts the learning rate based on the mean square of the historical gradients of each parameter, allowing for gradient adjustments tailored to the parameters, thus avoiding the efficiency bottlenecks caused by a fixed learning rate at different training stages. The specific approach involves setting a relatively large learning rate during the initial training phases to accelerate the network's adjustment to the initial weights. Subsequently, as the training epochs progress, the learning rate is gradually reduced, enabling the network to achieve finer parameter optimization as it approaches the optimal solution. This phased learning rate adjustment method helps to enhance the model's final segmentation accuracy while maintaining training speed."

4.3. Summary of Overall Improvements

This model has been improved in five aspects: ① Introduction of the Attention Gate module in the UNet architecture; ② Implementation of regularization strategies (DropBlock) in the Attention Gate module; ③ Optimization of the loss function, using BCE with Logits; ④ Introduce a dynamic learning rate adjustment strategy; ⑤ Introduce the GhostNet network to replace the encoding structure; below are the results of the ablation and comparative experiments based on this improvement, which include comparisons for each improvement point and a horizontal comparison of the final model.

Table 1. Ablation Experiments for Various Improvement Measures

Model number	Improve the content	mIoU (%)	PA (%)	Acc (%)	Dice
Model A	UNet (Baseline)	81.37	96.42	91.28	0.8912
Model B	Model A + Attention Gate	84.26	97.58	93.17	0.9134
Model C	Model B + GhostNet Encoder	85.13	97.69	93.84	0.9209
Model D	Model C + DropBlock Regularization	85.34	97.73	93.92	0.9239
Model E	Model D + BCE with Logits Loss	85.56	97.81	94.01	0.9253
Model F	Model E + Dynamic learning rate	85.69	97.93	94.06	0.9267

"Compared to the basic model, the introduction of the Attention Gate module (Model B) improved the mIoU by 2.89%, which is the most significant enhancement among all single-step improvements, indicating that the attention mechanism has a remarkable effect in focusing on the tongue image region. Subsequently, GhostNet was introduced as the encoder (Model C), which resulted in a slight improvement in metrics compared to Model B, while also achieving an increase in segmentation accuracy with a reduction in parameter count, thus validating the applicability of lightweight structures for this task. In Model D, the DropBlock regularization module was introduced, effectively suppressing overfitting and leading to more stable performance on the test set; after optimizing the loss function (Model E), both mIoU and Dice improved simultaneously, enhancing the model's ability to discern boundaries. Finally, using the RMSprop optimizer for dynamic learning rate adjustment (Model F) further accelerated model convergence and improved the model's fine-grained optimization capability in the later stages of training."

Table 2. Comparison of Attention Mechanism Experiments

Model Name	Params (M)	FLOPs (G)	mIoU (%)	PA (%)	Acc (%)	Dice
UNet (Baseline)	34.5	88.2	81.37	96.42	91.28	0.8912
UNet + SE	36.1	90.4	82.11	96.78	91.85	0.8986
UNet + CBAM	37.3	91.7	83.42	97.02	92.41	0.9063
Standard Attention UNet	38.5	92.6	84.26	97.58	93.17	0.9134
Improve Attention UNet (+DropBlock)	39.2	93.1	85.69	97.93	94.06	0.9267

The experiment conducted a comprehensive comparison based on parameters (Params), computational volume (FLOPs), and five segmentation evaluation metrics (mIoU, PA, Acc, Dice) to assess the contribution of various structural improvements to the model's expressive capability and segmentation performance. Using the original UNet network as the baseline model, comparative experiments were sequentially conducted with SE, CBAM, and the Attention

Gate attention mechanism, with all metrics showing an upward trend. Finally, based on the standard Attention UNet, a DropBlock regularization strategy was further introduced to construct the final improved Attention UNet model. By randomly dropping contiguous regions in the feature map, DropBlock effectively alleviated the overfitting problem during the training process, enhancing its generalization ability. The model has 39.2M parameters and 93.1G FLOPs, which is a very limited increase compared to the standard Attention UNet, yet it achieved optimal results across all metrics, indicating significant advantages in boundary handling, target region localization, and background suppression.

Table 3. Comparison of Regularization Experiments

Model Structure	Params (M)	FLOPs (G)	mIoU (%)	Dice
No regularization (Attention UNet)	38.5	92.6	84.26	0.9134
Dropout	38.5	92.6	84.78	0.9186
Spatial Dropout	38.5	92.6	84.95	0.9202
DropBlock	39.2	93.1	85.34	0.9239

Table 4. Comparison of Main Stem Experiments

Model Name	Params (M)	FLOPs (G)	mIoU (%)	Dice
Standard Attention U-Net	38.5	92.6	84.26	0.9134
EfficientNet-B0 Encoder	18.3	68.4	84.93	0.9187
MobileNetV3 Encoder	14.2	60.1	85.02	0.9198
ShuffleNetV2 Encoder	10.6	42.5	85.13	0.9209
GhostNet Encoder	9.8	38.7	85.34	0.9224

"From the table data, it can be seen that after introducing different regularization methods based on the Attention UNet, the model's segmentation performance (mIoU, Dice) has improved. The original network model without regularization achieved an mIoU of 84.26% and a Dice coefficient of 0.9134, indicating that this structure possesses strong feature focusing ability and segmentation accuracy. However, due to the absence of regularization techniques, there is a risk of overfitting, especially when training samples are limited or the edges of the tongue are unclear, and there is room for improvement in generalization ability. Therefore, we proposed to incorporate regularization operations and conducted comparative experiments with Dropout, Spatial Dropout, and DropBlock in sequence. Among them, the DropBlock regularization showed the most significant improvement in evaluation metrics, outperforming the aforementioned traditional regularization methods. This further validates the effectiveness and advancement of the improved strategy proposed in this paper for tongue image segmentation tasks."

As can be seen from the table above, when replacing the model's encoder with a lightweight architecture, both the number of parameters and the computational load are significantly reduced, while the segmentation performance also improves to varying degrees. The total GhostNet network, which features the core "phantom convolution," shows the greatest increase in mIoU among the aforementioned solutions, with the lowest computational load and parameter count after improvements, indicating that the model achieves an optimal balance between lightweight design and performance retention. This network is chosen as the backbone for encoder replacement.

Table 5. Comparison of Loss Function Experiments

Loss function	Params (M)	FLOPs (G)	mIoU (%)	Dice
BCE Loss	38.5	92.6	84.26	0.9134
Dice Loss	38.5	92.6	84.75	0.9186
BCE + Dice	38.5	92.6	85.18	0.9224
Focal Loss	38.5	92.6	84.86	0.9193
BCE with Logits	38.5	92.6	85.69	0.9267

BCE Loss, as a fundamental loss function, performs exceptionally well in pixel-level classification. However, it is highly sensitive to class imbalance, especially in edge regions, where its performance is often limited, leading to less detailed delineation in these areas and the potential for blurred segmentation boundaries. While the accuracy of other loss functions has improved, BCE with Logits Loss stands out as the best among all loss functions, achieving an mIoU of 85.69% and a Dice score of 0.9267. This loss function integrates the Sigmoid function and its numerical stability handling on top of BCE, resulting in more stable forward propagation and backward gradients, effectively avoiding issues such as gradient vanishing or explosion that are common in traditional BCE. In practical training, this function significantly enhances convergence speed and final segmentation accuracy, particularly excelling in edge localization and complex background separation.

Table 6. Comparison of Different Learning Rate Strategies

Learning Rate Strategy	Params (M)	FLOPs (G)	number of turns	mIoU (%)	Dice
Fixed learning rate	38.5	92.6	88	84.12	0.9112
Step Decay	38.5	92.6	84	84.94	0.9194
Cosine Annealing	38.5	92.6	83	85.21	0.9215
Warm-up + Cosine	38.5	92.6	81	85.34	0.9237
Dynamic Adjustment Strategy	38.5	92.6	78	85.69	0.9267

From the table above, it can be seen that the model using a fixed learning rate strategy only completed convergence after the 88th training round, demonstrating a lack of flexibility in the optimization process and a tendency to fall into local optima. This indicates that the fixed learning rate strategy limits the model's adaptability to complex data and fails to effectively explore better solutions. Therefore, using a dynamic learning rate adjustment strategy may be more beneficial for improving model performance and accelerating convergence, thereby achieving better results in practical applications. First, Step Decay [12] sets a fixed decay step length to reduce the learning rate at specific rounds, helping

the model to converge quickly in the early stages and then gradually stabilize. This strategy improved the mean Intersection over Union (mIoU) to 84.94% and the Dice coefficient to 0.9194, while reducing the number of convergence rounds to 84, showing a certain optimization advantage. Cosine Annealing [13], on the other hand, utilizes the properties of the cosine function to smoothly decrease the learning rate from a fast pace to a slow one until it reaches a minimum value, avoiding parameter oscillation near the optimal solution, which resulted in a reduction of convergence rounds to 83 and an increase in segmentation accuracy, demonstrating the good adaptability of the cosine annealing strategy in deep segmentation tasks. "Warm-up + Cosine is a dual optimization strategy that combines 'Warm-up preheating' and 'Cosine Annealing'. During the initial training phase, the learning rate is gradually increased in the Warm-up stage, and after the preheating ends, the cosine annealing strategy is employed. This smooths the learning rate down to a minimum value, reducing fluctuations in the model during the early stages and enhancing training stability. The model achieved a mean Intersection over Union (mIoU) of 85.34% and a Dice coefficient of 0.9237, successfully converging at the 81st epoch. The dynamic adjustment strategy proposed in this paper, based on performance feedback, focuses on the dynamic adjustment of parameters, ultimately achieving the highest segmentation accuracy within a minimum of 78 training epochs. This indicates that the strategy effectively promotes the convergence of network parameters towards the optimal direction while maintaining model stability, making it the best choice for the learning rate mechanism in this study."

Table 7. Cross-Comparative Experiment

Model Name	Params (M)	FLOPs (G)	mIoU (%)	PA (%)	Acc (%)	F1-score	Dice
U-Net	34.5	88.2	81.37	96.42	91.28	0.8912	0.8912
MaskR-CNN	38.7	95.6	82.74	96.65	92.05	0.8974	0.8974
DeepLabV3+	15.2	63.4	84.68	97.01	93.14	0.9125	0.9125
Attention U-Net	38.5	92.6	84.26	97.58	93.17	0.9134	0.9134
TransUNet	61.3	130.2	85.09	97.72	94.01	0.9248	0.9248
Improvement of Lightweight Design Attention UNet	10.5	42.2	85.69	97.93	94.06	0.9267	0.9267

From the perspective of mIoU and Dice coefficients, the improved Attention UNet network proposed in this paper significantly outperforms the traditional UNet and the lightweight DeepLabV3+. Even when compared to the TransUNet, which has a much higher computational load than the model presented in this paper, there is still a slight improvement, clearly demonstrating that the method in this paper has a leading edge in segmentation accuracy for practical tasks. From the PA and Acc metrics, this model has also achieved the highest pixel-level classification accuracy, showcasing strong generalization ability and robustness, making it suitable for the precise segmentation needs of tongue image analysis in real clinical environments. The segmentation results are shown in Figure 7, corresponding to the original image, UNet, Mask R-CNN, DeepLabV3+, and the algorithm proposed in this paper.

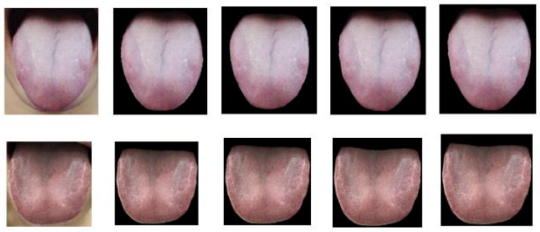


Figure 7. Segmentation Effect Diagram

5. Model Deployment

5.1. Introduction to Hardware

Using hardware with the Raspberry Pi 4B 8G to deploy a tongue segmentation model, validating the effectiveness and usability of this improvement based on the hardware platform, providing a feasible method for creating an embedded tongue segmentation model. The Raspberry Pi 4B [14] chosen for this design is a card-sized motherboard developed by the Raspberry Pi Foundation. Its dimensions are similar to that of a credit card, and it uses SD/Micro SD cards as memory storage. This motherboard serves as a high-performance development platform based on the Linux operating system, equipped with a 64-bit quad-core processor that can reach a maximum frequency of 1.5 GHz, effectively ensuring excellent processing speed and multitasking capabilities. With its 4GB RAM, it is sufficient to meet the memory demands of complex applications.

5.2. Environment Setup

Before deploying the model, it's necessary to set up the Raspberry Pi testing environment. The Debian-based Linux OS can be obtained from the official website, known for its excellent stability and security. Next, use a 64GB SD card compatible with Raspberry Pi 4B for system flashing. Before flashing, the SD card needs to be formatted. Once the system image is successfully flashed to the SD card, insert it into the Raspberry Pi's card slot and power it on. Use the SSH service on the PC and establish a connection between the PC and the board with the help of the Putty tool.

After the connection is successful, you will enter the main interface. The programming tools, connection interface, and main interface are shown in Figures 8(a), (b), and (c) below.

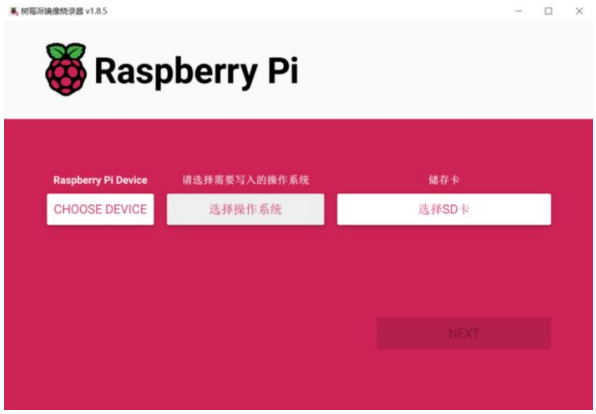


Figure 8(a). Official burning tool for Raspberry Pi

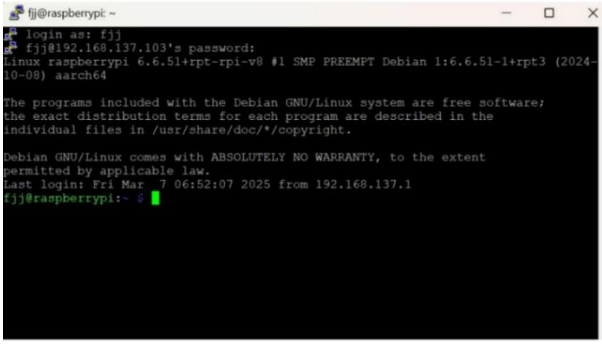


Figure 8(b). Connecting Raspberry Pi via Putty

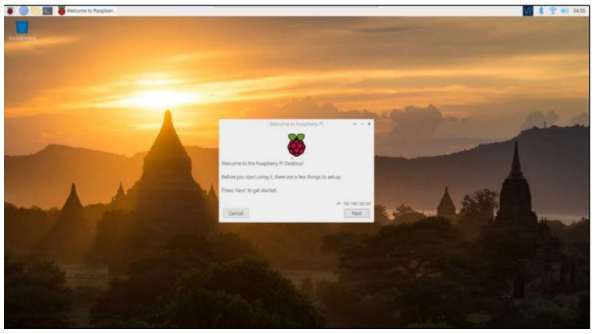


Figure 8(c). Raspberry Pi OS interface

Then install the cross-compilation tools and install the graphical processing tool OpenCV; necessary software packages and dependencies such as Python, numpy, pip, and Pytorch; followed by the required libraries and functions for the experiment.

5.3. Model Format Conversion and Acceleration

Considering the advantages of high performance and low cost, the compact and highly capable Raspberry Pi 4B was selected as the model deployment device. Accordingly, the environment setup and preliminary deployment experiments were completed. Next, to achieve precise segmentation of the tongue image, ONNXRuntime technology was employed to accelerate and optimize the segmentation model. The advantage of this framework lies in its ability to directly connect with the intermediate representation ONNX, as it natively supports reading and executing ONNX files, thus avoiding complex format conversion processes. Utilizing the torch.onnx.export function provided by PyTorch, the segmentation model was converted into the ONNX standardized format. Subsequently, acceleration was applied using ONNXRuntime technology. Below are the evaluation results of the improved network on the platform.

Table 8. Results of Tongue Appearance Segmentation Evaluation

Model	Average time taken (s)	Maximum memory usage (MB)	Accuracy (%)
Tongue splitting	0.94	210	93.1

"In terms of average inference time, the segmentation module takes 0.94 seconds, completing processing within 1 second, ensuring the overall responsiveness of the system; regarding resource consumption, the system memory usage is 210MB, and it operates stably without memory overflow or resource contention issues, compatible with the memory conditions of Raspberry Pi 4B. As for recognition accuracy, the segmentation module maintains a rate of 93.1%, which is at a high level, meeting the basic accuracy requirements for clinical applications."

6. Summary and Outlook

In the tongue segmentation phase, this paper introduces the Attention mechanism based on UNet and integrates the lightweight GhostNet encoder, designing an improved Attention UNet network that balances structural lightweightness and edge sensitivity. This model can finely depict the contour of the tongue's edge, effectively distinguishing the adhesion areas between the tongue and the background, achieving high-quality extraction of the tongue region. After validation on public datasets and a self-constructed tongue image collection, the model achieved a segmentation accuracy of 94.06%, demonstrating a good balance between performance and resource consumption. Ablation experiments further confirmed that the introduction of the Attention module and GhostNet significantly enhances the edge perception capability of tongue segmentation and the model's deployment efficiency, laying a precise imaging foundation for subsequent tongue feature analysis.

Although this study has introduced Attention mechanisms and lightweight modules to balance model performance with embedded deployment needs, there is still room for further exploration of more efficient network structures, such as the adaptability of Transformer-type architectures in image recognition, as well as the reinforcing effects of edge-aware mechanisms on the recognition of tongue body microstructures. Additionally, in response to complex situations such as blurred tongue edges and severe coverage by tongue coating, the research could incorporate multi-scale fusion strategies and uncertainty modeling methods to enhance the robustness and credibility of segmentation and classification models.

References

[1] Feng Li. Research and Application of Tongue Image Feature Recognition Algorithm Based on Deep Learning [D]. Chengdu University of Traditional Chinese Medicine, 2023. DOI: 10.26988/d.cnki.gcdzu.2023.000406.

[2] Xue Qingchen. Research on Threshold Image Segmentation Based on Improved Differential Evolution Algorithm [D]. Guangxi University for Nationalities, 2024. DOI: 10.27035/d.cnki.ggxmc.2024.000582.

[3] Guo Yuxiao. Research on Pedestrian Detection and Segmentation Methods Based on Deep Perception Algorithms

[D]. Guilin University of Electronic Technology, 2024. DOI:10.27049/d.cnki.ggldc.2024.000648.

[4] Lu Yunxi, Li Xiaoguang, Zhang Hui, et al. Research Progress on Tongue Image Segmentation Technology in Traditional Chinese Medicine: Methods, Performance, and Prospects [J/OL]. Acta Automatica Sinica: 1-12 [2019-10-07]. https://doi.org/10.16383/j.aas.c180807.

[5] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation [C]//International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015: 234-241.

[6] Padhy I, Dansena P, Sahoo S, et al. An efficient camouflaged image segmentation with modified UNet and attention techniques [J]. Scientific Reports, 2025, 15(1):21086-21086.

[7] Liou. Regularization Algorithm for Disordered Iterative Connection Layers in Neural Networks [D]. Ningxia University, 2024. DOI:10.27257/d.cnki.gnxhc.2024.002114.

[8] ZHANG J, KONG X, LI X, et al. Fault diagnosis of bearings based on deep separable convolutional neural network and spatial dropout [J]. Chinese Journal of Aeronautics, 2022, 35(10):301-312.

[9] Zhang Zhunan. Comparison and Improvement of Typical Regularization Methods in Neural Networks [D]. Shenyang Aerospace University, 2021. DOI:10.27324/d.cnki.gshkc.2021.000009.

[10] Zhang Shuang. Research on Stability Enhancement Methods for Disease Prediction Based on Loss Function Optimization [D]. Donghua University, 2025.

[11] Li Ming, Lai Guohong, Chang Yanming, et al. Performance Analysis of Different Optimizers in Deep Learning Algorithms [J]. Information Technology and Informatization, 2022, (03): 206-209.

[12] Shao Xiaoqiang, Chen Li, Zhang Shou, and others. One-step implementing three-qubit phase gate via manipulating rf SQUID qubits in the decoherence-free subspace with respect to cavity decay [J]. Chinese Physics B, 2009, 18(12):5161-5167.

[13] Liu Guoquan, Chen Shangliang, Li Yuezhong, et al. A method for target detection of high-voltage power equipment based on an improved YOLOv3 using SGD and cosine annealing algorithms [J]. Journal of Donghua University of Science and Technology (Natural Science Edition), 2024, 47(03): 294-300.

[14] Zhao Shengqing, Lu Shi. Design of an Automatic Classification Trash Can Based on Raspberry Pi 4B and YOLOv5 [J]. Industrial Control Computer, 2025, 38(06): 87-89 + 92.