ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Parameter-efficient UNet-style architecture Model with Adams-Bash forth Two-Step Method and Parallel Frequency-Spatial Attention-guided strategy

Wei Liu[1], *Jiangxin Huang*[1,*], *Shiyang Hu*[1], *Jian Luo*[1] *and Yuanchuang Hu*[1]

[1] School of Artificial Intelligence, Hezhou University, Hezhou 542899, Guangxi, China

SUMMARY: *This paper proposes the Adams Bash forth two-step method and frequency domain space parallel attention UNet-style architecture (AB2-FSA-UNet-style architecture) model for medical imaging semantic segmentation of skin melanoma. This model combines the Adams Bash forth two-step discretization neural memory ordinary differential equation (nmODE) decoder, channel level contextual anchor attention (CCAA) module, and frequency-domain spatial parallel attention-guided strategy. Introduce CCAA module in the feature representation subnetwork section, use NMODE in the feature reconstruction subnetwork section, and combine frequency domain spatial parallel attention-guided strategy to improve segmentation classification precision and reduce computational complexity. The empirical findings indicate that AB2-FSA-UNet-style architecture has achieved excellent performance on three common datasets: PH2, ISIC2016, and ISIC2017. Compared with methods such as UNet-style architecture model, EGE UNet, FEDUKD, FocalUnetR, DCSAUNet, and FatNet, AB2-FSA-UNet-style architecture has significantly improved classification precision, recall, specificity, F1 index, and IoU index. At the same time, the parameters and computation of this model are relatively low, providing an efficient solution for mobile medical devices and edge computing scenarios. The AB2-FSA-UNet-style architecture model significantly improves the classification precision and efficiency of skin melanoma segmentation by combining the Adam Bash forth two-step discretization nmODE decoder, CCAA module, and frequency-domain spatial parallel attention-guided strategy.*

KEYWORDS: *Parameter-efficient UNet-style architecture model; Anchor attention-guided strategy; Frequency domain space; Parallel attention-guided strategy; Adams Bash forth two-step method; Melanoma; image segmentation*

1 Introduction

Medical imaging semantic segmentation is a research hotspot in modern medicine. High performance medical imaging semantic segmentation can assist doctors in diagnosing and judging patients, liberate doctors' productivity, and improve work efficiency [1, 2]. However, traditional medical segmentation methods rely on manual labor, which requires doctors' rich experience and consumes a lot of time and energy. With the continuous development of artificial intelligence technology, medical imaging semantic segmentation methods based on artificial intelligence technology have become a new possibility and have achieved excellent results.

Since the UNet-style architecture model was proposed, it has achieved great success in

*201903002@hzxy.edu.cn
https://doi.org/10.65102/is2026822
Published: 30 April, 2026

the field of biomedical image analysis, especially in tasks such as cell segmentation, tissue segmentation, and lesion detection [3]. Many scholars have made significant efforts and research, proposing improved versions of U-shaped networks to enhance their performance or efficiency. For example, the R2UNet-style architecture proposed in reference [4] significantly improves the practical application of recurrent neural networks in the medical field through the gradient optimization characteristics of residual networks. The Unet++proposed in reference [5] significantly improves the sensitivity and specificity of medical imaging semantic segmentation through semantically consistent skip connections and multi-level supervision. And with the sudden rise of Transformer [6], attention-guided strategies have also entered everyone's research scope. The FatNet proposed in reference [7] includes a new dual feature representation subnetwork architecture that integrates CNN and Trans form, enhancing the model's ability to extract local features and global contextual information, and has been validated through experiments. The Perspective+Unet proposed in reference [8] achieves collaborative learning of local details and global context through dual path feature representation subnetworks, efficient non local Transformers, and cross scale integrators. Reference [9] proposed a novel UNet network based on multi-scale convolution and fusion attention-guided strategy, which improved the ability to capture image details and the overall quality of generated images. The Swin Unet proposed in reference [10] integrates the window attention-guided strategy of Swin Transformer into UNet-style architecture to enhance global context awareness.

Ordinary differential equation (ODE) systems have been widely studied and used in fields such as mathematics, physics, and engineering for a long time. Reference [11] proposes a new method called Neural Ordinary Differential Equations (NOEs), which transforms neural networks into representations of ODEs and elucidates the mathematical principles of ResNet, laying the foundation for the unification of neural networks and ODEs. The parameter efficiency of Nodes is high, and by eliminating intermediate variables in forward propagation, it greatly reduces the number of parameters and computational overhead. Subsequently, it sparked the interest of many scholars. Reference [12] proposes a new Bayesian polynomial neural network structure that combines Bayesian uncertainty and polynomial expressiveness, extending this structure to ODEs to form PNODEs for continuous time modeling. Reference [13] proposes a method to transform ODE solving from iterative optimization to forward computation through a reversible architecture and probabilistic modeling, promoting real-time applications. However, there are still some limitations to mapping data through Nodes. Traditional Nodes lack memory function and can only learn features within the same topological space as the input data. Reference [14] proposed a variant of NODs - neural memory ordinary differential equations (nmODE), which introduces memory mechanisms into traditional NODs. A dynamic memory module based on auto feature representation subnetworks is designed to store key state snapshots and dynamically retrieve them through attention-guided strategies, addressing the limitations of traditional NODs. Reference [15] combines the parameter-efficient dynamic modeling of NMODE with the feature fusion of U-like architecture, and replaces NMODE in the feature reconstruction subnetwork part without affecting the performance of the original model. Reference [16] proposed nmpls-net, which combines the dynamic memory capability and parameter-efficient characteristics of nmODE (neural memory ODE). It uses the lung lobe shape annotated by experts (such as the contour of public datasets) as the initial memory template to reduce the number of parameters to be learned, and constrains the ODE layer through Lipschitz regularization to prevent gradient explosion.

This article proposes a novel parameter-efficient UNet-style architecture - Adams Bash

forth two-step method and frequency domain space parallel attention UNet-style architecture (AB2-FSA-UNet-style architecture). This model significantly improves segmentation classification precision and significantly reduces the computational complexity of the model by introducing a Channel Level Context Anchor Attention (CCAA) module in the feature representation subnetwork section and using a Neural Memory Ordinary Differential Equation (NMODE) based on Adams Bash forth two-step discretization in the feature reconstruction subnetwork section, combined with a frequency domain spatial parallel attention-guided strategy.

2 Algorithm in this article

2.1 Overall Architecture

The overall architecture of AB2-FSA-UNet-style architecture is based on the classical UNet-style architecture structure, consisting of an feature representation subnetwork and a feature reconstruction subnetwork [17]. The feature representation subnetwork is responsible for extracting feature information from the image, while the feature reconstruction subnetwork generates segmentation results through feature reconstruction. The network structure is shown in Figure 1.

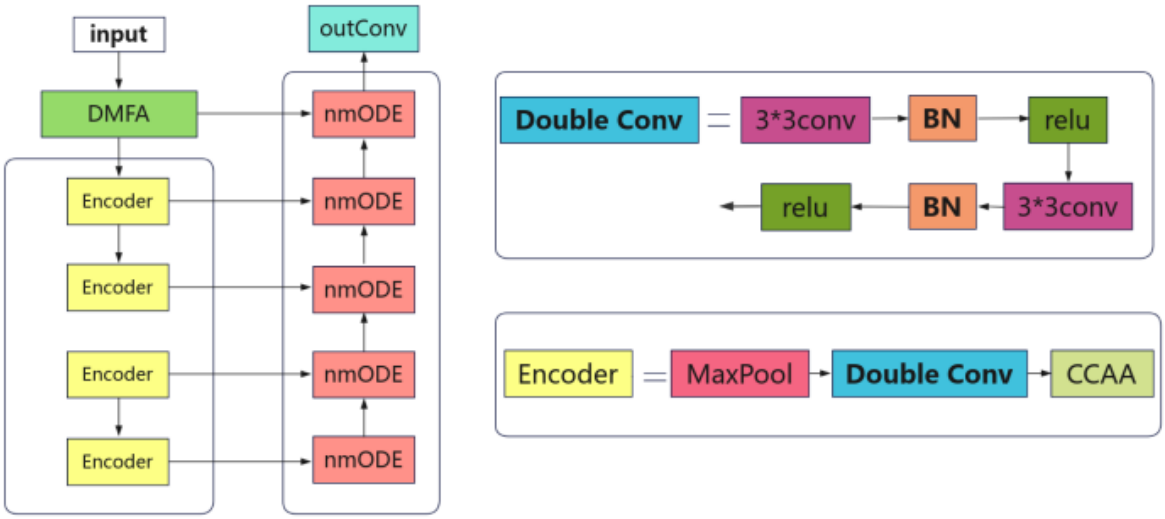

Figure 1: Network Architecture Diagram

2.2 Feature representation subnetwork Design

The feature representation subnetwork part of the paper includes a max pooling layer, a two-layer convolution module, and a channel wise Contextual Anchor Attention (CCAA) module. The maximum pooling layer has a kernel size of 2 and a stride of 2, which is used to reduce the size of the feature map while preserving the local maximum features. The double-layer convolution module consists of two 3×3 sized convolution kernels, which are sequentially passed through the BN layer and activation layer, and then concatenated together to form a feature representation chain of convolution BN Relu convolution BN Relu (as shown in Double conv in Figure 1).

2.2.1 Context Anchor Attention Module (CCAA)

Contextual Cross-dimensional Anchor Attention module (CCAA) The module achieves channel space collaborative attention through a dual path feature interaction mechanism, as shown in Figure 2. This module significantly improves feature representation capability through cascaded channel recalibration and anchor based spatial modeling.

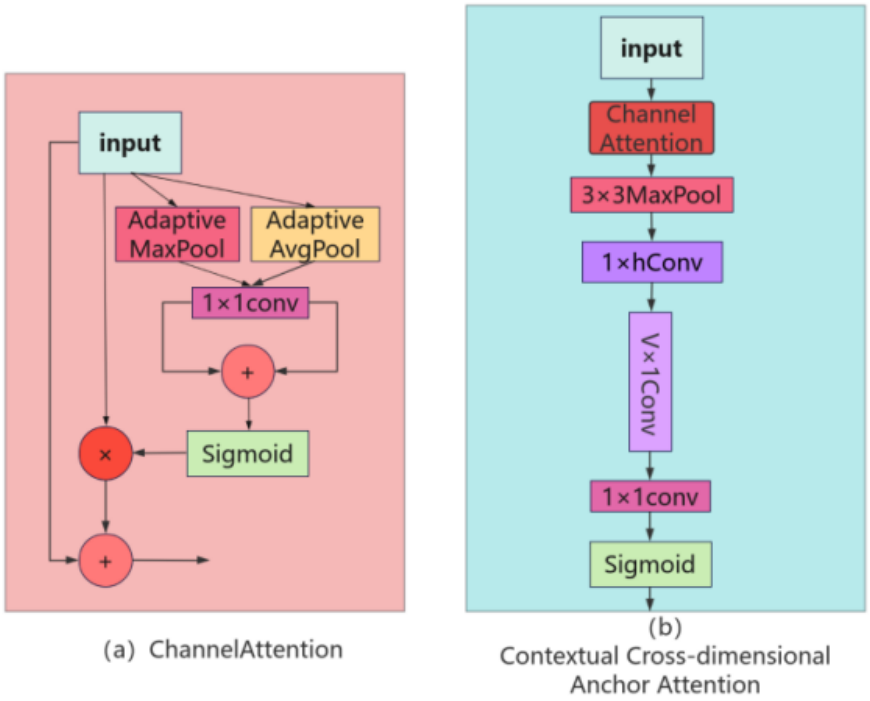

Figure 2: Contextual Cross-dimensional Anchor Attention

(1) Channel Attention Path. Firstly, a dual path global feature representation strategy is adopted to obtain global contextual features in the channel dimension through parallel execution of adaptive average pooling (AdaptiveAvgPool2d) and adaptive max pooling (AdaptiveMaxPool2d) operations. The features of the two branches are transformed nonlinearly through an 1×1 convolution operation with shared parameters (including batch normalization and SiLU activation function), and the calculation process can be expressed as:

$$\begin{cases} F' = Conv_{(1\times1)}(Pool_{avg}(F)) \\ F'' = Conv_{(1\times1)}(Pool_{\max}(F)) \\ F^{out} = \sigma(F'+F'')\otimes F + F \end{cases} \tag{1}$$

where, $Pool_{avg}$ and $Pool_{\max}$ correspond to the global average pooling operation layer and the maximum pooling operation layer, respectively. $Conv_{(1\times1)}$ represents the convolution operation constructed by (1×1) sharing parameters; σ is the activation function, and in this research context, the Sigmoid function was selected; FF refers to multiplication operations performed on a channel-by-channel basis.

First, add the extracted features of these two channels element by element, and then input the added result into the Sigmoid activation function to generate the channel attention weight matrix. Finally, by using residual connections, the features that have undergone

channel calibration are obtained.

(2) Axial spatial attention. In the spatial dimension, an anisotropic convolutional attention-guided strategy was designed. Firstly, 3 × 3 max pooling is used for local feature smoothing, followed by capturing long-range spatial dependencies through cascaded horizontal vertical separable convolutions. The specific implementation is:

(1) The horizontal convolution kernel uses $(1, k_h)$ band kernels to establish cross pixel correlations along the horizontal axis;

(2) The vertical convolution kernel uses $(k_v, 1)$ band kernel to establish cross pixel correlations along the vertical axis;

Among them, k_h and k_v are configurable odd value convolution kernel sizes (default 11). The biaxial convolution operation can be represented as:

$$\begin{cases} H = Conv_{(1\times1)}(Pool_{\max}(F^{out})) \\ W = \sigma(Conv_{(1\times1)}(Conv_{(v_k,1)}^{(g=c)}(Conv_{(1,h_k)}^{(g=c)}(H)))) \end{cases} \tag{2}$$

where, $Pool_{\max}$ corresponds to the max pooling operation layer in the 3×3 structure. $Conv_{(1\times1)}$ represents the convolution operations included in 1×1. $Conv_{(1,h_k)}^{(g=c)}$ belongs to the horizontal stripe convolution operation, which uses the convolution kernel $(1,h)$. Here, $g=c$ is used to illustrate the correlation between the number of groups and channels, and this convolution adopts a depth wise separable convolution form. $Conv_{(v_k,1)}^{(g=c)}$ is a stripe convolution in the vertical direction, with its convolution kernel set to $(v,1)$. $g=c$ is also used to indicate information about the number of groups and channels, using depth wise separable convolution. Finally, with the help of 1×1 convolution processing and combined with Sigmoid function, the generation of attention weights is completed.

2.2.2 Dynamic Multi Domain Feature Aggregation Module (DMFA)

(1) Architecture Design. The Dynamic Multi domain Feature Aggregation Module (DMFA) integrates multi domain feature representations through an adaptive weight learning mechanism, as shown in Figure 4. This module consists of three parts:

1. Large kernel feature extractor: using 11 × 11 convolution to capture long-range spatial dependencies:

$$X_b = \mathrm{Re}\,LU(BN(Conv_{11\times11}(X))) \tag{3}$$

2. Three feature enhancers: original feature path X_b, frequency domain enhancement path X_f, and spatial enhancement path X_s:

$$\begin{cases} X_f = FDA(X_b) + X_b \\ X_s = CCAA(X_b) + X_b \end{cases} \tag{4}$$

where, X_f is the global attention in the frequency domain.

3. Frequency Domain Attention (FDA) and Context Anchor Attention (CCAA) enhance frequency domain and spatial domain features, respectively;

4. Dynamic weight fusion: Multi branch feature adaptive aggregation is achieved

through learnable attention weights.

The input feature map $X \in \mathbb{R}^{C\times H\times W}$ undergoes the following process:

$$\begin{cases} X_b = \mathrm{Re}\,LU(BN(Conv_{11\times 11}(X))) \\ X_f = FDA(X_b) + X_b \\ X_s = CCAA(X_b) + X_b \\ W = \mathrm{Softmax}(Conv_{1\times 1}(GAP([X_b; X_f; X_s]))) \\ Y = X_b + \sum_{i=1}^{3} W_i \odot [X_b, X_f, X_s] \end{cases} \tag{5}$$

where, $[\bullet;\bullet]$ represents channel concatenation, and $W \in \mathbb{R}^{3\times 1\times 1}$ is a dynamic weight vector.

(2) Frequency domain attention submodule. To address the issue of insufficient representation of traditional attention-guided strategies in the frequency domain, a Frequency Domain Attention (FDA) mechanism based on Fast Fourier Transform is designed, with a calculation process consisting of four stages:

$$X_{freq} = FFT2(X), X_{freq} \in C^{C\times H\times W} \tag{6}$$

1) Spectrum analysis: Decompose and process complex frequency spectra.

$$\begin{cases} A = \mid X_{freq} \mid \\ P = \mid \angle X_{freq} \mid \end{cases} \tag{7}$$

where, $|\bullet|$ and $\angle\bullet$ are represented as amplitude spectrum and phase spectrum calculations.

2) Attention generation: Learning frequency domain weights through compressed excitation network.

$$W_f = MLP(GAP(A)), W_f \in [0,1]^{C\times 1\times 1} \tag{8}$$

where, GAP $(\cdot)$ is global average pooling, while MLP consists of two fully connected layers (dimensionality reduction ratio r=4).

3) Frequency domain inverse transform: inverse transform the weighted spectrum:

$$Y_{freq} = IFFT(W_f \odot X_{freq}) \tag{9}$$

This design effectively enhances the model's perception ability of key frequency components through a frequency domain global attention-guided strategy.

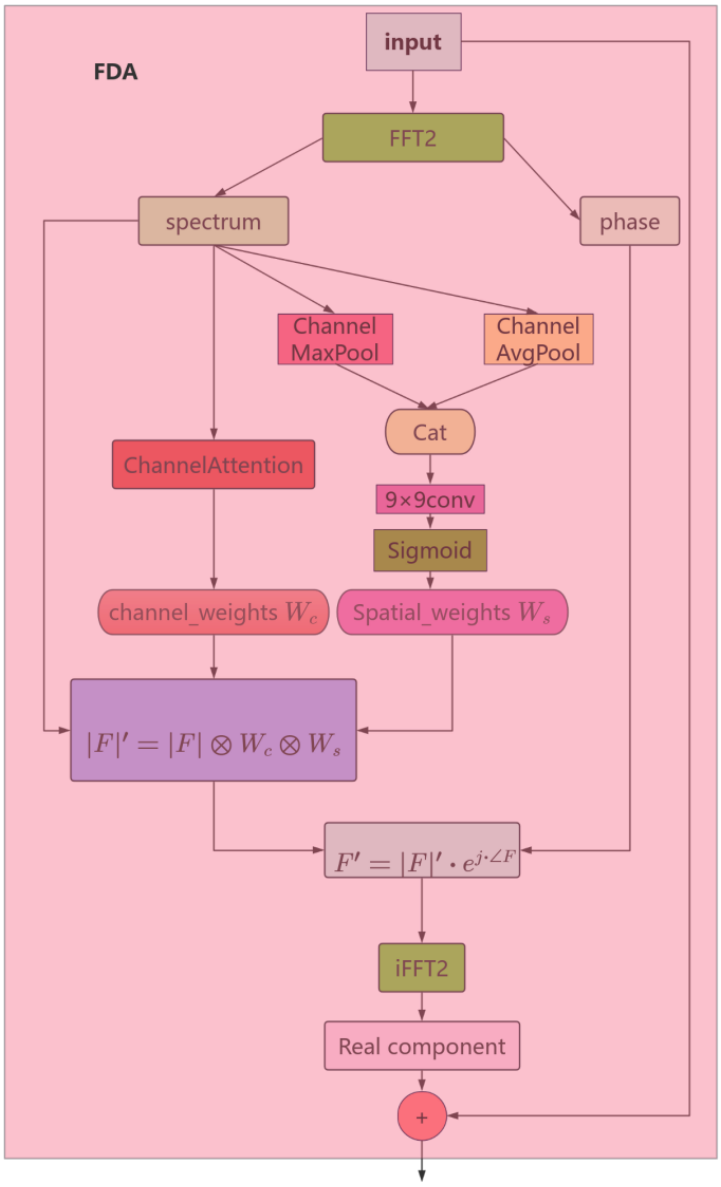

Figure 3: Frequency Domain Attention

(3) Multi domain feature fusion mechanism. To achieve effective integration of cross domain features, design a three-level feature fusion strategy:

1) Primary fusion: concatenate the large kernel convolution feature $\Phi(\bullet)$, frequency domain attention feature $\Psi(\bullet)$, and spatial attention feature $\Omega(\bullet)$ along the channel dimension:

$$X_{cat} = [\Phi(X); \Psi(\Phi(X)); \Omega(\Phi(x))] \in R^{3C \times H \times W} \tag{10}$$

2) Channel compression: dimensionality reduction and feature recombination through 1x1 convolution:

$$X_{fused} = Conv_{1\times1}(X_{cat}) \tag{11}$$

3) Residual connection: preserves the original large kernel convolution features as skip information.

$$Y = X_{fused} + \Phi(X) \tag{12}$$

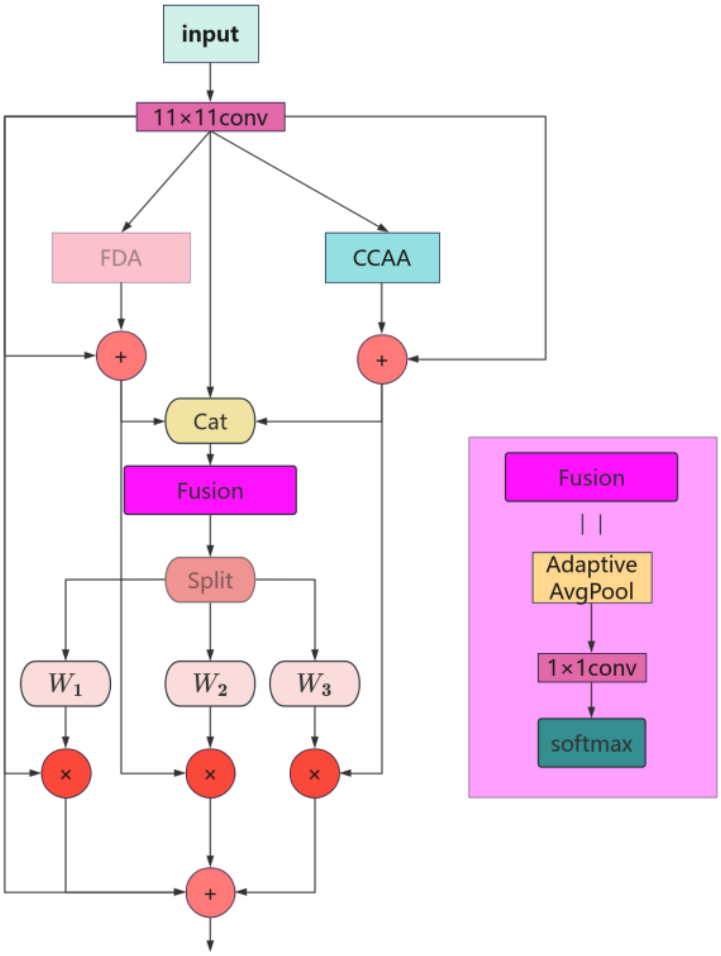

Figure 4: Dynamic Multi-domain Feature Aggregation Module

2.3 Decoder

As shown in Figure 5, the feature representation subnetwork part uses discretized neural memory ordinary differential equations (nmODE). In the nmODE architecture, the initial value $y(0)$ of the upstream path is set to a random value, and $x(t)$ is the external input data of the corresponding hierarchy. NMODE, like UNet network, accepts two inputs, with skip connections as external input data $x(t)$, and the information of the upstream path as $y(t)$, thus constructing the feature representation subnetwork part of UNet. The differential equation of NmODE is as follows:

$$y'(t) = -y(t) + f(y(t) + g(x(t), \theta_t) \tag{13}$$

where, For $t \geq 0$, where $y(t) \in R^n$ represents network status, $x(t) \in R^m$ represents external input, and θ_t represents parameters for skip connections. We use nmODE to develop the feature reconstruction subnetwork for UNet, which applies parameterized calculations in

skip connections, represented as $g(x(t),\theta_t)$. The purpose of skip connections is to convert low-level features into fixed sized high-level feature maps. Unet with nmODE feature reconstruction subnetwork performs parameter free feature aggregation. Given an L layer U-shaped network using nmODEs decoders, define the output of the network feature representation subnetwork (i.e. the skip connection input of the decoder) as x^l and represent the upstream path of the feature representation subnetwork as $y^l, l \in [1, L]$. Initialize $x(0) = x^L$ and $y(0) = y^L = 0$, and the goal of the nmODES feature reconstruction subnetwork is to obtain numerical solutions at point $(\tau, x(\tau) = x^l)$ along the trajectory of the above equation. Represent equation (13) as *F(x,y)*.

In the nmODE feature reconstruction subnetwork section, only a small number of parameters are responsible for the information integration of skip connections in the upstream path, and different discretization methods will result in feature representation subnetworks with different structures.

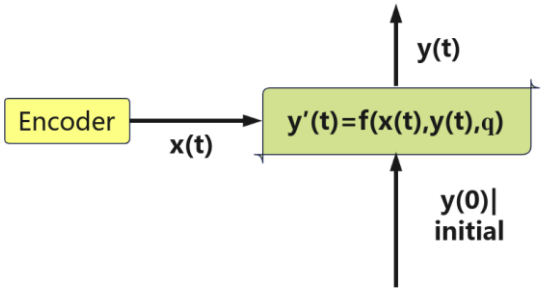

Figure 5: Schematic Diagram of Neural Memory Ordinary Differential Equation (nmODE) Module

2.3.1 Explicit Euler Method Feature reconstruction subnetwork (EED)

Given a differential equation $y'(t) = F(t, y)$, select a value δ along the time axis t as the fixed step size. Assuming the relationship between adjacent points is $t_{n+1} = t_n + \delta$, based on the current time t_n's solution y_n, the Euler method [18] can be used to derive an approximate solution for the next time:

$$y_{n+1} = y_n + \delta \cdot F(t_n, y_n) \tag{14}$$

where, y_n is an approximate solution of time t_n, i.e. $y_n \approx y(t)$. y_{n+1} is the explicit function of y_i regarding $i \le n$. From formulas (13) and (14), it can be concluded that:

$$y_{n+1} = (1-\delta) \cdot y_n + \delta \cdot f(y_n + g(x_n, \theta_n)) \tag{15}$$

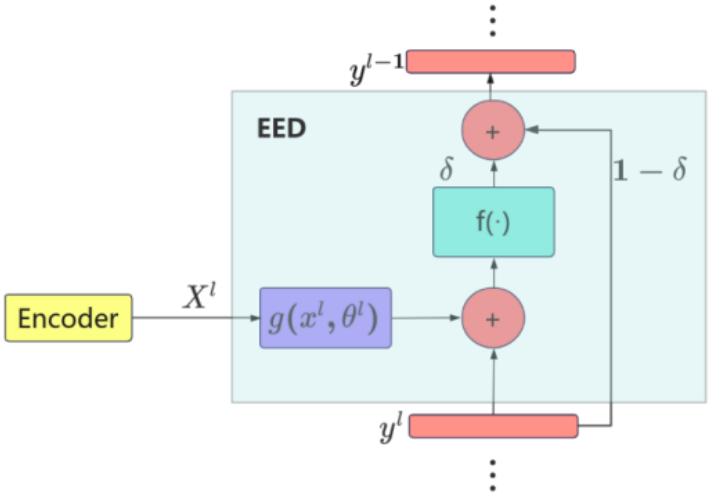

Figure 6: Explicit Euler-Discretized Neural Memory ODE Feature reconstruction subnetwork (EED)

Discretize formula (15)and in the Unet network, in the feature reconstruction subnetwork section, use the lowest layer y^l as the initial value to continuously solve it up. The process is discretized, and the number of solving steps is inversely proportional to the number of network layers:

$$y^{l-1} = (1-\delta)\cdot y^l + \delta \cdot f(y^l + g(x^l, \theta^l)) \tag{16}$$

where, If the total number of layers in Unet is L and l is the current layer, then $1 \le l \le L$. y^l is the input of the l layer in the upstream path of the decoder, and it is also the initial value for solving the y^{l-1} of the previous layer. x^l represents skip connection, δ is 0.2. θ^l represents the weight of the skip connection at the l layer. The feature reconstruction subnetwork designed based on formula (16) is shown in Figure 6.

2.3.2 Trapezoidal Formula Method Feature reconstruction subnetwork (TRD)

Given a differential equation $y'(t) = F(t, y)$, select a value δ along the time axis t as the fixed step size. Assuming the relationship between adjacent points is $t_{n+1} = t_n + \delta$, based on the solution y_n of the current time t_n, an approximate solution for the next time can be derived using the trapezoidal formula:

$$y_{n+1} = y_n + \frac{\delta}{2}[F(t_n, y_n) + F(t_{n+1}, y_{n+1})] \tag{17}$$

where, y_n is an approximate solution of time t_n, i.e. $y_n \approx y(t)$.

By substituting $F(t_n, y_n)$ and $F(t_{n+1}, y_{n+1})$ into formula (17) from formula (13), we obtain:

$$y_{n+1} = y_n + \frac{\delta}{2}[(-y_n + f(y_n + g(x_n, \theta_n))) + (-y_{n+1} + f(y_{n+1} + g(x_{n+1}, \theta_{n+1})))] \tag{18}$$

Merge and move similar items, and organize them to obtain:

$$y_{n+1}(1+\frac{\delta}{2}) = y_n(1-\frac{\delta}{2}) + \frac{\delta}{2}[f(y_n + g(x_n, \theta_n)) + f(y_{n+1} + g(x_{n+1}, \theta_{n+1})] \tag{19}$$

Dividing both sides by $(1+\delta/2)$ yields the final formula:

$$y_{n+1} = \frac{2-\delta}{2+\delta} y_n + \frac{\delta}{2+\delta}[f(y_n + g(x_n, \theta_n) + f(y_{n+1} + g(x_{n+1}, \theta_{n+1})] \tag{20}$$

Similarly, rewriting the continuous value form into a discrete network form and using the EED calculation result instead of y_{n+1} on the right side of equation (20) yields the following equation, where y_p^l is the calculation result using EED.

$$y^{l-1} = \frac{2-\delta}{2+\delta} y_p^l + \frac{\delta}{2+\delta}[f(y_p^l + g(x^l, \theta^l)) + f(y^{l-1} + g(x^{l-1}, \theta^{l-1})] \tag{21}$$

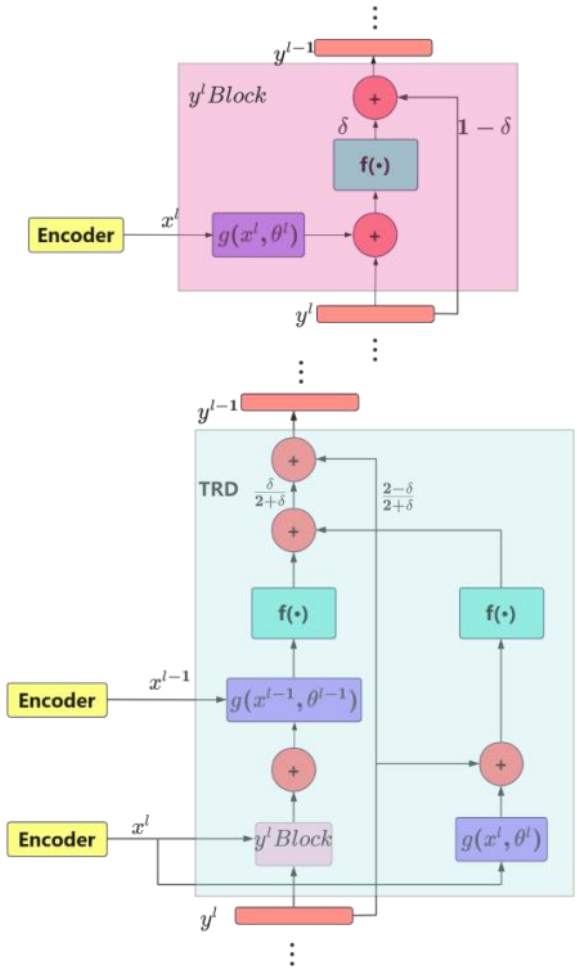

Figure 7: Trapezoidal Formula Decoder

The feature representation subnetwork structure based on this construction is shown in Figure 7. It is worth noting that the trapezoidal formula feature representation subnetwork

requires two layers of skip connection information as inputs x^l and x^{l-1}: the current layer and the previous layer. Therefore, the topmost feature reconstruction subnetwork uses EED calculation.

2.3.3 Adams Bash forth two-step feature reconstruction subnetwork (ABD)

Similarly, based on the solution y_n at the current time t_n, an approximate solution for the next time can be derived using the Adams Bash forth two-step method:

$$y_{n+1} = y_n + \frac{\delta}{2}(3F(t_n, \theta_n) - F(t_{n-1}, \theta_{n-1})) \tag{22}$$

Substitute formula (13) into $F(t_n, \theta_n)$ and $F(t_{n-1}, \theta_{n-1})$:

$$y_{n+1} = y_n + \frac{\delta}{2}(3(-y_n + f(y_n + g(x_n, \theta_n))) - (-y_{n-1} + f(y_{n-1} + g(x_{n-1}, \theta_{n-1})))) \tag{23}$$

Similarly, discretizing it yields the following equation:

$$y^{l-1} = y^l + \frac{\delta}{2}[3(-y^l + f(y^l + g(x^l, \theta^l))) - (-y^{l+1} + f(y^{l+1} + g(x^{l+1}, \theta^{l+1})))] \tag{24}$$

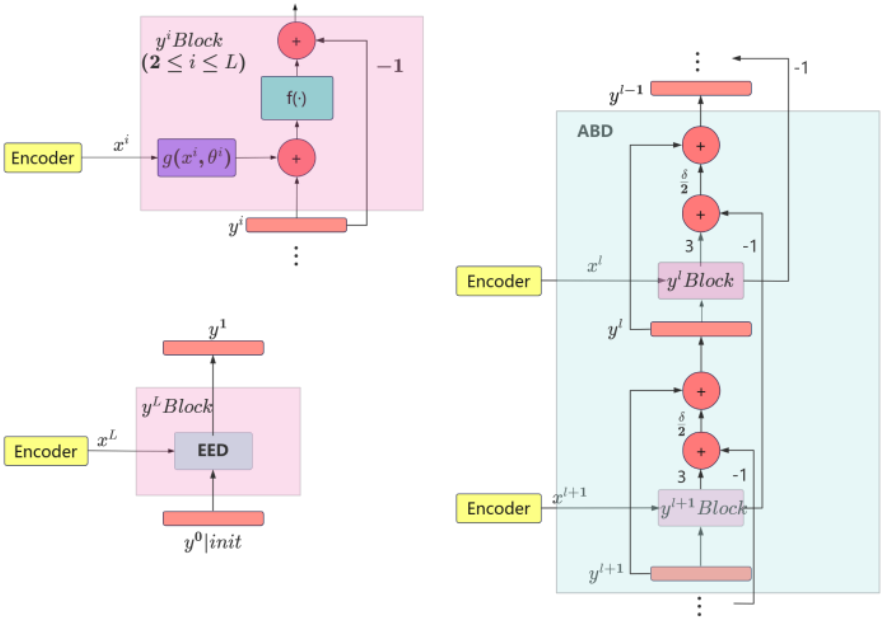

Figure 8: Adams Bash forth two-step decoder

The feature representation subnetwork structure based on this construction is shown in Figure 8. It is worth noting that the Adams Bash forth two-step feature representation subnetwork requires the use of the first two layers of skip connection information as inputs x^l and x^{l+1}: the current layer and the previous layer, and requires the use of the calculation result y^l from the previous layer. Therefore, the feature reconstruction subnetwork of the first layer uses EED calculation.

2.3.4 Third order Runge Kutta feature reconstruction subnetwork (RK3)

Similarly, based on the solution y_n at the current time t_n, an approximate solution for the next time can be derived using the 3rd order formula of Kutta:

$$\begin{cases} y_{n+1} = y_n + \dfrac{\delta}{6}(k_1 + k_2 + k_3) \\ k_1 = F(t_n, y_n) \\ k_2 = F(t_n + \dfrac{\delta}{2}, y_n + \dfrac{\delta}{2}k_1) \\ k_3 = F(t_n + \delta, y_n - \delta k_1 + 2\delta k_2) \end{cases} \tag{25}$$

By substituting formula (13), we can obtain:

$$\begin{cases} k_1 = -y_n + f(y_n, g(x_n, \theta_n)) \\ k_2 = -(y_n + \dfrac{\delta}{2}k_1) + f((y_n + \dfrac{\delta}{2}k_1) + g(x_n, \theta_n)) \\ y_t = y_n - \delta k_1 + 2\delta k_2 \\ k_3 = -y_t + f(y_t + g(x_n, \theta_n)) \end{cases} \tag{26}$$

Updating y_{n+1} yields:

$$y_{n+1} = y_n + \frac{\delta}{6}(k_1 + 4k_2 + k_3) \tag{27}$$

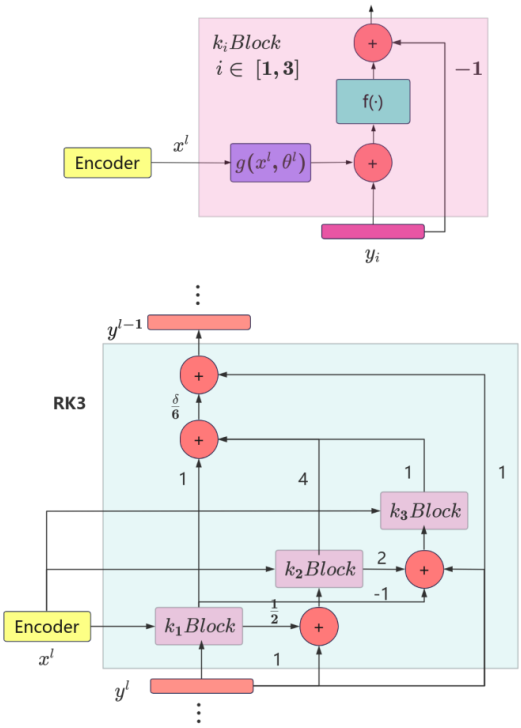

Figure 9: 3rd order Runge Kutta decoder

After merging, we can obtain:

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{6}(-y_n + f(y_n + g(x_n, \theta_n)) \\ &+4(-(y_n + \frac{k_1}{2}) + f(y_n + \frac{k_1}{2} + g(x_n, \theta_n))) \\ &+(-(y_n - k_1 + 2k_2) + f(y_n - k_1 + 2k_2 + g(x_n, \theta_n)))) \end{aligned} \tag{28}$$

Similarly, discretizing it yields the following equation:

$$\begin{aligned} y^{l-1} = y^l + \frac{h}{6}(&-y^l + f(y^l + g(x^l, \theta^l)) + \\ &4(-(y^l + \frac{k_1}{2}) + f(y^l + \frac{k_1}{2} + g(x^l, \theta^l))) + \\ &(-(y^l - k_1 + 2k_2) + f(y^l - k_1 + 2k_2 + g(x^l, \theta^l)))) \end{aligned} \tag{29}$$

The feature representation subnetwork structure based on this construction is shown in Figure 9. It is worth noting that the third-order Runge Kutta feature reconstruction subnetwork only needs to use the current skip connection information x^l and the upstream input information y^l as inputs.

2.3.5 Function Selection and Initial Value Problem

Given that the feature reconstruction subnetwork part is designed without parameters and the f function lacks adjustable parameters to change the morphological characteristics of the input y^l, it is necessary to reshape x^l in the g function to ensure its compatibility with y^l. g encompasses a range of components, such as up sampling functions for adjusting height and width, and convolution operation modules for regulating the number of channels. In the model constructed in this article, the internal implementation process of the g function includes convolution operation, up sampling operation, and activation function processing; The f function only includes batch normalization operation. For the initialization setting of y_0, initialize it as a zero matrix, and keep the width and height of the matrix consistent with the initial input.

3 Cost function

The experiment adopts a composite cost function strategy, combining the advantages of Binary Cross Entropy (BCE) and Dice loss to optimize the segmentation performance of the model. This mixed cost function can simultaneously focus on pixel level classification precision and region level segmentation consistency.

3.1 Binary Cross Entropy

The calculation inspiration for the Binary Cross Entropy (BCE) cost function comes from maximum likelihood estimation, which is a commonly used cost function in machine learning and deep learning, especially when dealing with binary classification problems. It calculates the logarithmic difference between the true label and the predicted probability, as

shown in the following formula. In the field of image segmentation, it is necessary to classify each pixel. If foreground and background segmentation is required, it can be seen as a binary classification problem, which is used to evaluate the performance of the model and improve its classification precision by optimizing parameters.

$$\mathrm{L}(o,y) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \ln(o_i) + (1-y_i)\ln(1-o_i)] \tag{30}$$

where, y_i is the label value, o is the predicted output value, and N is the total number of pixels.

3.2 Dice loss

Dice loss, in this task, is the F1 indicator. Please refer to section 3.2 for the calculation method of F1 indicator formula (35).

3.3 Final cost function

In this experiment, the mixed loss value of binary cross entropy and DICE cost function is defined as formula (31). $\lambda \in [0,1]$ is the weight coefficient, which is 0.4 in this experiment.

$$L = \lambda \times L(o,y) + (1-\lambda) \times F1 \tag{31}$$

4 Experiments

4.1 Dataset

The proposed algorithm was validated using three public datasets [19]: PH2, ISIC2016, and ISIC2017. The PH2 experimental dataset contains 200 skin mirror images corresponding to machine annotated images. ISIC2016 and ISIC2017 are two large-scale skin mirror image datasets provided by the ISIC challenge, including 1279 and 2000 images, respectively. And the experimental datasets randomly divided into training set, validation set, and testing set using a 6:2:2 ratio. Using the training set as the training subject, determine whether to perform early stopping processing through the validation set (see Section 3.3), and ultimately evaluate the model on the test set.

Simultaneously, using PH2 as the main dataset, complete experimental validation of each module and conduct final model evaluation on the ISIC2016 and ISIC2017 datasets. In order to save computational costs and improve convergence speed, the input image size of the experimental dataset was uniformly adjusted to 224×224 in this experiment.

4.2 Evaluation indicators

In order to accurately evaluate the performance of the model, the average value of five evaluation indicators, namely classification precision, sensitivity/recall, specificity, F1 index (equivalent to Dice coefficient in this task), and Intersection over union (IoU), was studied for performance evaluation.

Classification precision Acc, calculated using the formula:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{32}$$

Sensitivity/recall rate Sen:

$$Sen = \frac{TP}{TP + FN} \tag{33}$$

The specific Spe is calculated using the following formula:

$$Spe = \frac{TN}{TN + FP} \tag{34}$$

The $F1$ index, which is the $Dice$ coefficient in this task, is calculated using the following formula:

$$F1 = \frac{2 \times TP}{Fp + 2 \times TP + FN} \tag{35}$$

The intersection to union ratio IoU is calculated using the following formula:

$$IoU = \frac{TP}{TP + FP + FN} \tag{36}$$

where, TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively. All indicators are averaged on the test set to ensure statistical significance of the evaluation results.

4.3 Experimental setup

Based on the Windows 10 system environment, the model uses Python 3.10 and Python 2.2.1 frameworks to build the network. The experiment uses AMD Ryzen7 5700G processor, 32GB memory, and an NVIDIA 22G 2080Ti graphics card as training devices. The experimental environment is shown in the table below:

Table 1: Experimental Environment Configuration

Experimental environment	Experimental configuration
CPU	AMD Ryzen7 5700G
Memory	32GB
Deep Learning Framework	Pytorch2.2.1
GPU	NVIDIA 2080Ti
programming language	Python3.10
operating system	Windows10

During the training process, all models used the same dataset, set the same hyperparameters, and did not undergo transfer learning or data augmentation processing. The AA used for training is 8, and the BB training is 400 times. The Adam optimization algorithm is used for learning, with an initial learning rate of 0.01. The cosine annealing algorithm with preheating [20] (Cosine Annealing Warm Restarts) is used to periodically adjust the learning rate to avoid getting stuck in saddle points and increase the generalization ability of learning. The learning rate variation curve is shown in Figure 10.

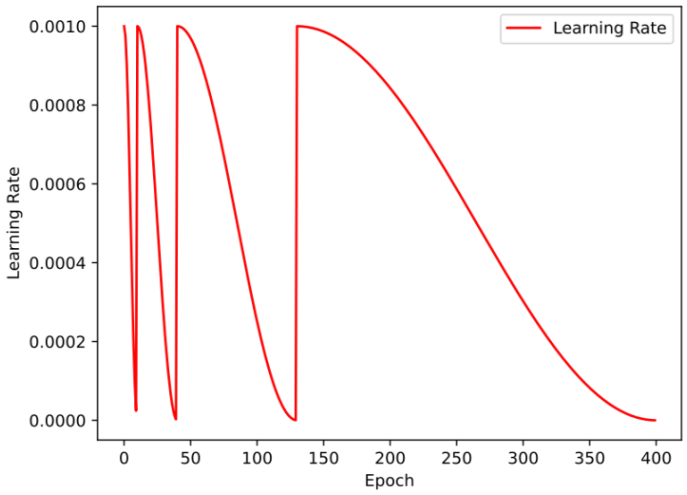

Figure 10: Changes in Learning Rate

At the same time, in order to avoid the impact of overfitting on the model, an early stopping mechanism is used. When the validation loss of the model does not improve in consecutive rounds of training, the training is terminated in advance. Assuming the optimal validation loss is $L_{\min}$, the current loss is $L_{current}$, the patience value is P, and the stopping condition is:

$$\text{Stop condition} = \begin{cases} True, if\ (L_{current} > L_{\min}) \text{ Continuously P times} \\ False, \text{otherwise} \end{cases}$$

To find the optimal P value, taking the model with feature representation subnetwork embedded SE module as an example, empirical findings were conducted with P values of 10, 20, and 30, as shown in Figure 10.

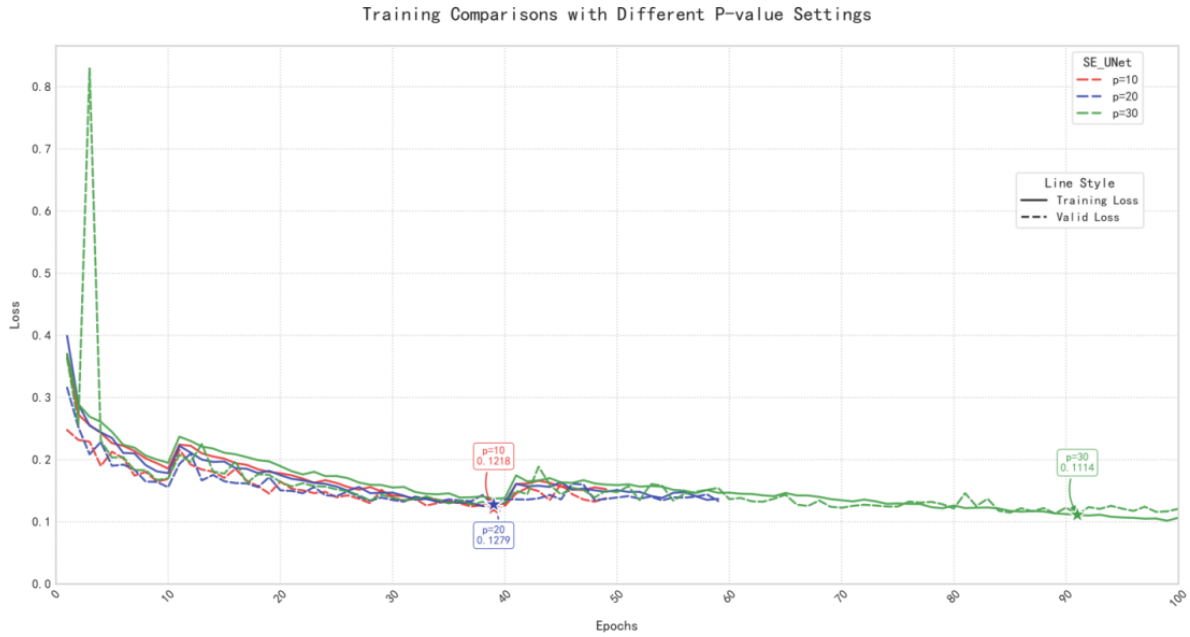

Figure 11: Training Comparisons with Different P-value Settings

From Figure 10, when the P values are 10 and 20, both stop at $epoch = 39$. But when $P = 30$ occurs, the model further converges, and the training error gradually decreases afterwards. The validation set error does not change much, and the model enters an overfitting state. The current termination point happens to be the convergence point. Therefore, choose.

4.4 Experimental comparison of various attention modules

As shown in Figure 12, to further validate the effectiveness of our proposed model, we compared the segmentation performance indicators of different attention modules integrated into the UNet-style architecture feature representation subnetwork, such as SE [21], CA [22], CBAM [23], EMA [24], AGCA [25], CAA [26], etc., using the original UNet-style architecture model as the baseline on the PH2 dataset.

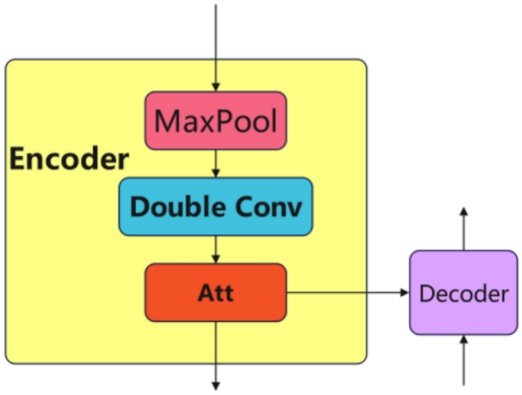

Figure 12: Feature representation subnetwork with Embedded Attention Module

The segmentation performance of each model is shown in Table 2. Meanwhile, in the feature representation subnetwork comparison experiment, we also used Grad CAM to visualize the features of the last layer of the model (as shown in Figure 13), in order to intuitively observe the actual performance of each attention module. The empirical findings showed that the classification precision (Acc: 96.70%), sensitivity (Sen: 94.87%), specificity (Spe: 97.77%), F1 score (F1: 93.44%), and intersection to union ratio (IoU: 88.03%) of the CCAA attention module proposed in this paper were significantly improved, indicating that it not only reduces false positives but also has the ability to detect lesions.

Table 2: Experimental Comparison of Attention Modules

Attention	Acc(%)	Sen(%)	Spe(%)	F1(%)	mIoU(%)
BaseLine	95.73	91.26	98.20	91.70	85.06
CA	96.61	90.58	97.88	92.73	86.79
CBAM	96.06	91.70	97.46	91.88	85.85
EMA	96.32	94.11	97.64	93.29	87.68
CAA	95.97	90.08	98.48	92.33	86.09
AGCA	96.21	92.53	96.59	91.34	85.03
SE	96.44	91.95	98.01	93.40	87.90
自己的	96.70	94.87	97.77	93.44	88.03

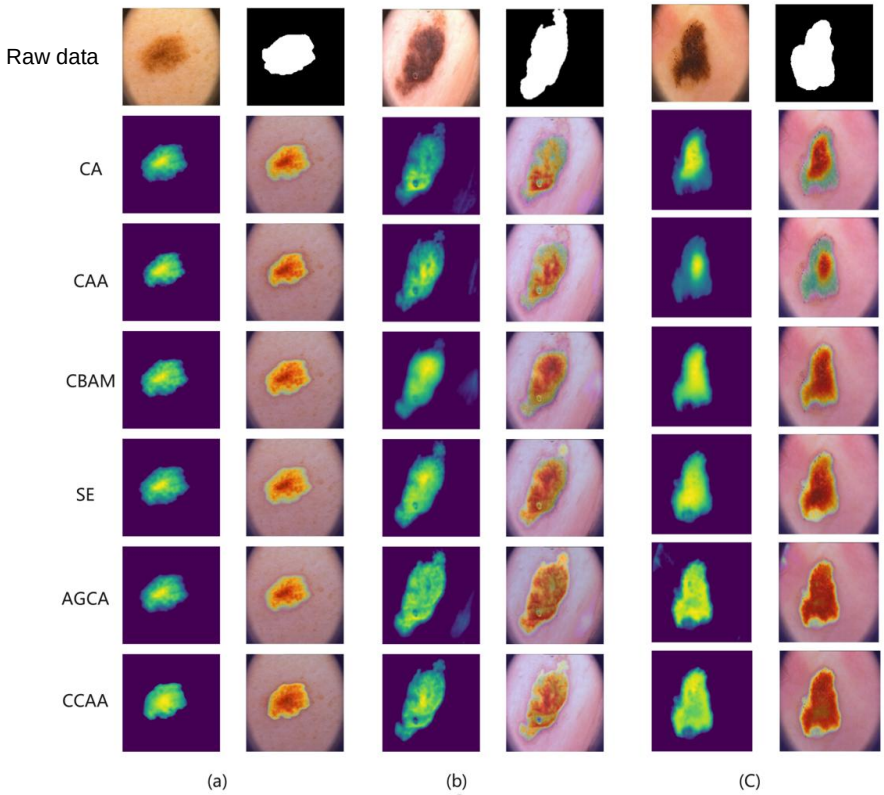

Figure 13: Comparison of Different Attention-guided strategys

4.5 DMFA module ablation experiment

On the PH2 dataset, ablation experiments were conducted using equation (3) as the Baseline. The empirical findings are shown in Table 3. Through experiments, it was found that by fusing multi domain features, the classification precision (Acc: 96.62%), sensitivity (Sen: 91.67%), specificity (Spe: 98.08%), F1 score (F1: 93.82%), and intersection to union ratio (IoU: 88.61%) of the model were greatly improved, achieving high results.

Table 3: Ablation Studies on Individual

Baseline	FDA	CCAA	Acc(%)	Sen(%)	Spe(%)	F1(%)	mIoU(%)
√			95.53	90.02	98.05	91.75	85.58
√	√		96.27	90.06	97.76	92.00	85.69
√		√	95.87	94.26	96.90	91.98	86.22
√	√	√	96.62	91.67	98.08	93.82	88.61

4.6 Feature reconstruction subnetworkExperiment Comparison

Through experiments, we implemented explicit Euler feature reconstruction subnetwork(EED), trapezoidal formula feature reconstruction subnetwork(TRD), Adams Bash forth two-step feature reconstruction subnetwork(ABD), and third-order Runge Kutta feature reconstruction subnetwork(RK3) under the same experimental conditions. The implementation results are shown in Table 4 below:

Table 4: OmODE Decoder

Decoder	Acc(%)	Sen(%)	Spe(%)	F1(%)	mIoU(%)
EED	95.76	90.98	97.78	92.16	85.98
TRD	95.89	90.21	98.43	92.03	86.02
ABD	96.42	91.85	98.19	93.41	87.87
RK3	95.58	92.11	926.98	91.61	84.87

Through comparative experiments, it was found that the performance of ABD was the most excellent, with an average classification precision of 96.42, an average sensitivity of 91.85, an average specificity of 98.19, an average F1 value of 93.41, and an average IoU of 87.87. Therefore, ABD is chosen as the feature reconstruction subnetwork for the model.

4.7 Ablation Experiment

To verify the effectiveness of each module, we conducted systematic ablation experiments based on the UNet-style architecture benchmark model, and the results are shown in Table 5. The experiment evaluated the contributions of the Context Anchor Attention Module (CCAA) in the feature representation subnetwork section, the Dynamic Multi Domain Feature Aggregation Module (DMFA), and the Efficient Encoding and Decoding Module (ABD) in the feature reconstruction subnetwork section to the model performance. The key findings are as follows:

1. Contribution of CCAA module: After introducing the CCAA module into the feature representation subnetwork, compared with the benchmark model, the average classification precision increased by 0.97%, the average sensitivity increased by 3.61%, the average specificity decreased by 0.43%, the average F1 value increased by 1.74%, and the average IoU increased by 2.97%.

2. Synergistic effect of DMFA: When CCAA and DMFA modules are jointly introduced, the average classification precision further increases by 0.13%, the average sensitivity decreases by 0.7%, the average specificity further increases by 0.38%, the average F1 value further increases by 0.2%, and the average IoU further increases by 3.65%.

3. ABD synergistic effect: When CCAA, DMFA, and ABD modules are jointly introduced, the average classification precision is further improved by 0.46%, the average sensitivity is improved by 0.22%, the average specificity is improved by 0.19%, the average F1 value is improved by 0.98%, and the average IoU is improved by 1.57%.

CCAA, DMFA, and ABD work together to promote the improvement of classification precision, recall, specificity, F1 index, IoU index, and other indicators. This validates the complementary advantages of the three in medical imaging semantic segmentation tasks.

Table 5: Ablation Studies on Individual

Baseline	CCAA	ABD	DMFA	Acc(%)	Sen(%)	Spe(%)	F1(%)	mIoU(%)
√				95.73	91.26	98.20	91.70	85.06
√	√			96.70	94.87	97.77	93.44	88.03
√		√		96.42	91.85	98.19	93.41	87.87
√			√	96.62	91.67	98.08	93.82	88.61
√	√	√		96.86	95.67	96.93	93.07	87.53
√	√		√	96.83	94.17	98.29	93.64	88.41
√		√	√	96.56	94.17	97.02	93.59	88.25
√	√	√	√	97.29	94.39	98.48	94.62	89.98

4.8 Experimental comparison of different methods on different datasets

In this experiment, the UNet-style architecture model, EGE UNet [27], FEDUKD [28], FocalUnetR[29], DCSAUNet [30], and FatNet were selected as reference models to compare the segmentation performance with the model proposed in this paper. The empirical findings are shown in Table 6 and Figure 14 below:

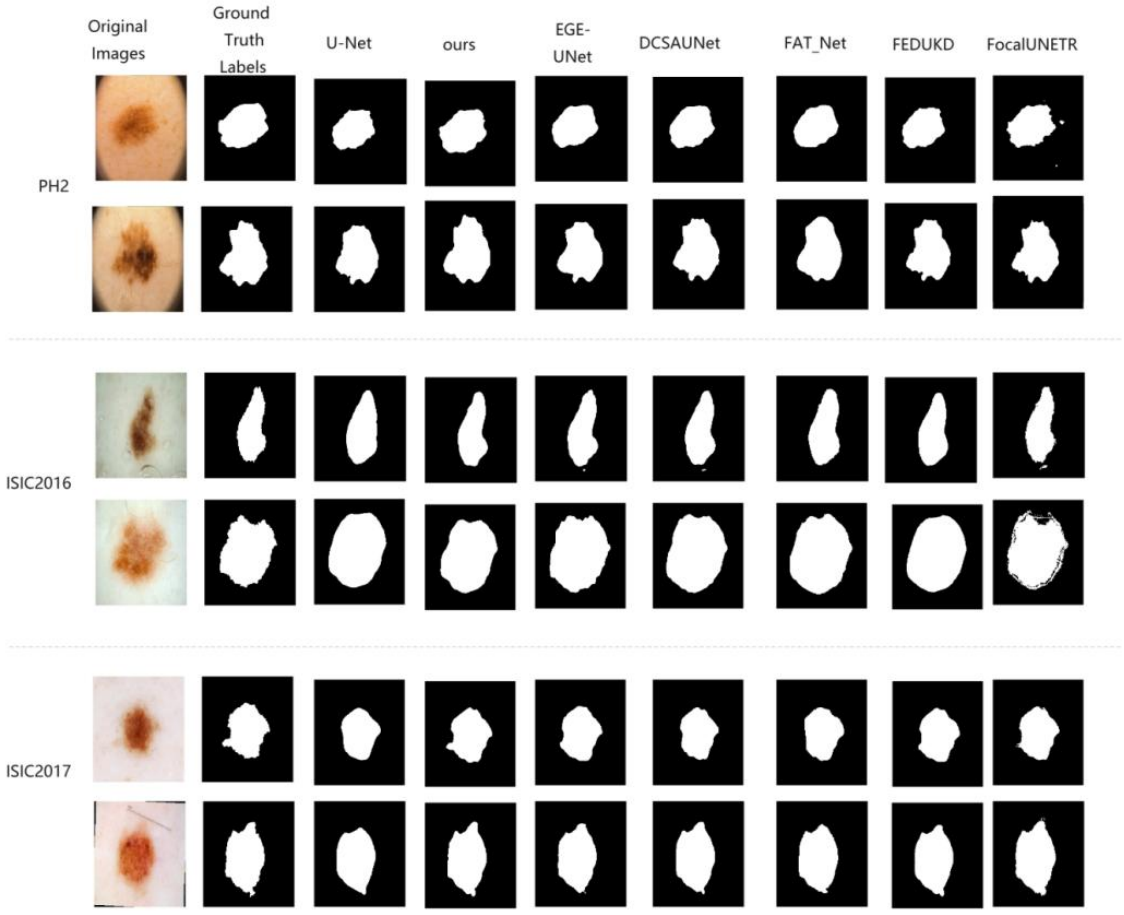

Figure 14: Empirical findings of each model

From Table 6, it can be seen that our method achieved the best performance on PH2, ISIC2016, and ISIC2017. The average values of classification precision, recall, specificity, F1 index, and IoU on the PH2 experimental dataset were 97.29%, 94.39%, 98.48%, 94.62%, and 89.98%, respectively; The average values of classification precision, recall, specificity, F1 index, and IoU in ISIC2016 were 94.72%, 92.26%, 96.37%, 89.60%, and 82.52%, respectively; The average values of classification precision, recall, specificity, F1 index, and IoU in ISIC2017 are 96.13%, 90.65%, 98.17%, 89.49%, and 82.34%. Compared with the UNet-style architecture model, EGE UNet, FEDUKD, FocalUnetR, DCSAUNet, and FatNet methods, the classification precision on the PH2 experimental dataset increased by 1.56%, 0.62%, 0.95%, 0.35%, 0.07%, and 1.01%, respectively. The recall rate increased by 3.13%, 2.35%, 2.88%, -0.93%, 2.25%, and 3.54%, the specificity increased by 0.28%, 0.9%, 0.59%, 1.95%, 0.25%, and 0.38%, the F1 index increased by 2.92%, 1.08%, 2.21%, 0.55%, 0.68%, and 1.9%, and the IoU increased by 4.92%, 1.9%, 3.78%, 0.89%, 1.09%, and 3.26%, respectively; On the ISIC2016 dataset, classification precision increased by 0.62%, 0.51%, 0.06%, 0.39%, 0.42%, and 1.32%, recall increased by 1.29%, 1.34%, 1.3%, 2.19%, 1.59%,

and 1.03%, specificity increased by 0.37%, 3.95%, -0.19%, 1.06%, 1.32%, and 1.13%, F1 index increased by 1.14%, 0.83%, 0.02%, 1.33%, 0.74%, and 2.72%, and IoU increased by 1.78%, 1.23%, 0.2%, 1.94%, 1.2%, and 3.75%, respectively; On the ISIC2017 dataset, classification precision increased by 0.69%, 0.47%, 0.24%, 0.28%, 0.34%, 0.55%, recall increased by 1.94%, -1.06%, 1.69%, 0.15%, 4.17%, 1.3%, specificity increased by 0.96%, 1.21%, 0.02%, 0.33%, -0.53%, 0.19%, F1 index increased by 1.81%, 2.27%, 0.7%, 2.12%, 2.41%, 2.18%, IoU increased by 2.5%, 3.12%, 0.82%, 3.06%, 3.54%, 3.08%, respectively.

Table 6: Performance Results Across Models

		Acc(%)	Sen(%)	Spe(%)	F1(%)	mIoU(%)
PH2	Baseline	95.73	91.26	98.20	91.70	85.06
	EGE-UNet	96.67	92.04	97.58	93.54	88.08
	FEDUKD	96.34	91.51	97.89	92.41	86.20
	FocalUnetR	96.94	95.32	96.53	94.07	89.09
	DCSAU_Net	97.22	92.14	98.23	93.94	88.89
	FatNet	96.28	90.85	98.10	92.72	86.72
	Proposed method	97.29	94.39	98.48	94.62	89.98
ISIC2016	Baseline	94.10	90.97	96.00	88.46	80.74
	EGE-UNet	94.21	90.92	92.42	88.77	81.29
	FEDUKD	94.66	90.96	96.56	89.58	82.32
	FocalUnetR	94.33	90.07	95.31	88.27	80.58
	DCSAU_Net	94.30	90.67	95.05	88.86	81.32
	FatNet	93.40	91.23	95.24	86.88	78.77
	Proposed method	94.72	92.26	96.37	89.60	82.52
ISIC2017	Baseline	95.44	88.71	97.21	87.68	79.84
	EGE-UNet	95.66	91.71	96.96	87.22	79.22
	FEDUKD	95.89	88.96	98.15	88.79	81.52
	FocalUnetR	95.85	90.50	97.84	87.37	79.28
	DCSAU_Net	95.79	86.48	98.70	87.08	78.80
	FatNet	95.58	89.35	97.98	87.31	79.26
	Proposed method	96.13	90.65	98.17	89.49	82.34

In order to compare the performance of various methods on different datasets more intuitively, the segmentation effects of different models on three datasets are presented, as shown in Figure 13. It can be seen that our method outperforms other methods in all three datasets.

At the same time, using top to calculate the computational complexity (FLOPs) and parameter count (Parameters) of each model, as shown in Table 7, the parameter count of the model in this paper is 21.74M and the computational complexity is 13.68G. The parameters are lower than EGE UNet, FEDUKD, and DCSAU-N et, and the computational complexity is lower than EGE UNet and DCSAU-N et. Compared to the benchmark model, the parameters have been reduced by 37.04% and the computational load has been reduced by 72.73%.

Table 7: Parameter Count and FLOPs Comparison Across Architectures

Model	Params (M)	FLOPs (G)	Input Size
Baseline	34.53	50.17	224×224
EGE-UNet	0.05	0.06	224×224
FEDUKD	17.27	30.77	224×224
FocalUnetR	26.91	16.28	224×224
DCSAU_Net	2.60	5.29	224×224
FatNet	29.62	42.80	224×224
Model in this article	21.74	13.68	224×224

5 Conclusion

The AB2-FSA-UNet-style architecture model proposed in this article significantly improves the classification precision and efficiency of skin melanoma segmentation by combining the Adam Bash forth two-step discretization nmODE decoder, channel level contextual anchor attention module, and frequency-domain spatial parallel attention-guided strategy. Empirical findings show that the model achieves excellent performance on multiple public data sets, providing an efficient solution for mobile medical devices and edge computing scenarios. Future work will further optimize the parameter-efficient design of models and explore more efficient parallel computing strategies.

Funding

This work was supported by the Guangxi Higher Education Institutions Early-Career Faculty Research Capacity Building Project (Grant No. No.2023KY0736), Department of Education of Guangxi Zhuang Autonomous Region; and the Project (Grant No. 2022KY0704).

About the Author

Wei Liu was born in Jingzhou, Hubei, in 1993. He obtained a Master's degree from Yangtze University in China. He is currently employed at Hezhou College. His main research direction is Data analysis, computer vision.

Jiangxin Huang was born in Yongzhou, Hunan, in 1992. She obtained a Master's degree from Central South University in China. She is currently employed at Hezhou College. Her main research direction is image recognition.

Shiyang Hu was born in Nanyang, Henan, China in 1987. He received his Bachelor's degree from Anyang Normal University and his Master's degree from Guangxi Normal University. Currently, he serves as a Senior Experimentalist at the School of Artificial Intelligence, Hezhou University. He is also a Senior Information Project Manager. His primary research focuses on big data technology and software development.

Jian Luo was born in Yulin City, Guangxi Zhuang Autonomous Region, China, in 1994. He obtained a master's degree from Guangxi Normal University in China. He currently works as a Lecturer at the School of Artificial Intelligence, Hezhou University, Guangxi. His main research areas are machine vision and 3D perception.

Yuanchuang Hu (1983-), male, from Hechi, Guangxi, is a member of the Communist Party of China, lecturer, and holds a Master's degree in Engineering. He graduated from the School of Computer Science and Engineering at Guilin University of Electronic Science and

Technology in 2010 with a major in Computer Application Technology. The current head and full-time teacher of the Department of Software Engineering at the School of Artificial Intelligence, Hezhou University. Research direction: Network information security. Hosted and participated in more than 10 projects at or above the municipal level, published over 30 relevant academic papers and educational reform papers in domestic and foreign academic journals, including 25 first authors, 1 EI indexed, 2 Chinese core indexed, and completed 18 software copyright projects.

References

[1] Siddique N, Paheding S, Elkin C P, et al. U-net and its variants for medical image segmentation: A review of theory and applications[J]. IEEE access, 2021, 9: 82031-82057.

[2] Zhang S, Zhang C. Modified U-Net for plant diseased leaf image segmentation[J]. Computers and Electronics in Agriculture, 2023, 204: 107511.

[3] Wu X, Hong D, Chanussot J. UIU-Net: U-Net in U-Net for infrared small object detection[J]. IEEE Transactions on Image Processing, 2022, 32: 364-376.

[4] Williams C, Falck F, Deligiannidis G, et al. A unified framework for U-Net design and analysis[J]. Advances in Neural Information Processing Systems, 2023, 36: 27745-27782.

[5] Jia X, Bartlett J, Zhang T, et al. U-net vs transformer: Is u-net outdated in medical image registration?[C]//International Workshop on Machine Learning in Medical Imaging. Cham: Springer Nature Switzerland, 2022: 151-160.

[6] Allah A M G, Sarhan A M, Elshennawy N M. Edge U-Net: Brain tumor segmentation using MRI based on deep U-Net model with boundary information[J]. Expert Systems with Applications, 2023, 213: 118833.

[7] Lu H, She Y, Tie J, et al. Half-UNet: A simplified U-Net architecture for medical image segmentation[J]. Frontiers in Neuroinformatics, 2022, 16: 911679.

[8] Si C, Huang Z, Jiang Y, et al. Freeu: Free lunch in diffusion u-net[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 4733-4743.

[9] Futrega M, Milesi A, Marcinkiewicz M, et al. Optimized U-Net for brain tumor segmentation[C]//International MICCAI brainlesion workshop. Cham: Springer International Publishing, 2021: 15-29.

[10] Chen J, Mei J, Li X, et al. TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers[J]. Medical Image Analysis, 2024, 97: 103280.

[11] Wang H, Cao P, Wang J, et al. Uctransnet: rethinking the skip connections in u-net from a channel-wise perspective with transformer[C]//Proceedings of the AAAI conference on artificial intelligence. 2022, 36(3): 2441-2449.

[12] Peng Y, Chen D Z, Sonka M. U-net v2: Rethinking the skip connections of u-net for medical image segmentation[C]//2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI). IEEE, 2025: 1-5.

[13] Rangaiah P K B, Augustine R. Enhanced glaucoma detection using U-Net and U-Net+ architectures using deep learning techniques[J]. Photodiagnosis and Photodynamic Therapy, 2025: 104621.

[14] Punn N S, Agarwal S. Modality specific U-Net variants for biomedical image segmentation: a survey[J]. Artificial Intelligence Review, 2022, 55(7): 5845-5889.

[15] Yin L, Wang L, Li T, et al. U-Net-LSTM: time series-enhanced lake boundary prediction model[J]. Land, 2023, 12(10): 1859.

[16] Walsh J, Othmani A, Jain M, et al. Using U-Net network for efficient brain tumor segmentation in MRI images[J]. Healthcare Analytics, 2022, 2: 100098.

[17] Yin L, Wang L, Li T, et al. U-Net-STN: a novel end-to-end lake boundary prediction model[J]. Land, 2023, 12(8): 1602.

[18] Petit O, Thome N, Rambour C, et al. U-net transformer: Self and cross attention for medical image segmentation[C]//Machine Learning in Medical Imaging: 12th International Workshop, MLMI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings 12. Springer International Publishing, 2021: 267-276.

[19] Xu Q, Ma Z, Duan W. DCSAU-Net: A deeper and more compact split-attention U-Net for medical image segmentation[J]. Computers in Biology and Medicine, 2023, 154: 106626.

[20] Zhang J, Li C, Kosov S, et al. LCU-Net: A novel low-cost U-Net for environmental microorganism image segmentation[J]. Pattern Recognition, 2021, 115: 107885.

[21] Wang X, Yang S, Fang Y, et al. Sk-unet: An improved u-net model with selective kernel for the segmentation of lge cardiac mr images[J]. IEEE Sensors Journal, 2021, 21(10): 11643-11653.

[22] Hong J S, Tzeng Y H, Yin W H, et al. Automated coronary artery calcium scoring using nested U-Net and focal loss[J]. Computational and Structural Biotechnology Journal, 2022, 20: 1681-1690.

[23] Su H, Wang X, Han T, et al. Research on a U-Net bridge crack identification and feature-calculation methods based on a CBAM attention mechanism[J]. Buildings, 2022, 12(10): 1561.

[24] Byeon H, Al-Kubaisi M, Dutta A K, et al. Brain tumor segmentation using neuro-technology enabled intelligence-cascaded U-Net model[J]. Frontiers in Computational Neuroscience, 2024, 18: 1391025.

[25] Delibasoglu I, Bagci Das D, Das O. Defect attention-based lightweight real-time adaptable surface defect analysis[J]. Ironmaking & Steelmaking, 2025: 03019233241296959.

[26] Zeng H, Fu L, Li J, et al. CAAVM-TransUNet: Integrating context anchor attention with transformer U-Net for single image dehazing[C]//2024 10th International Conference on Computer and Communications (ICCC). IEEE, 2024: 719-724.

[27] Ruan J, Xie M, Gao J, et al. Ege-unet: an efficient group enhanced unet for skin lesion segmentation[C]//International conference on medical image computing and computer-assisted intervention. Cham: Springer Nature Switzerland, 2023: 481-490.

[28] Kanagavelu R, Dua K, Garai P, et al. Fedukd: Federated unet model with knowledge distillation for land use classification from satellite and street views[J]. Electronics, 2023, 12(4): 896.

[29] Li C, Qiang Y, Sultan R I, et al. Focalunetr: A focal transformer for boundary-aware prostate segmentation using ct images[C]//International Conference on Medical Image Computing and Computer-Assisted Intervention. Cham: Springer Nature Switzerland, 2023: 592-602.

[30] Xu Q, Ma Z, Duan W. DCSAU-Net: A deeper and more compact split-attention U-Net for medical image segmentation[J]. Computers in Biology and Medicine, 2023, 154: 106626.