AFCNet: An adaptive fusion of composite attention convolutional neural network for polyp image segmentation

Bo jiao Jin
Northeastern University

Yi Zhang
zhangyi@mail.neu.edu.cn

Lin Qi
Northeastern University

Wei Qian
Northeastern University

Article

**Keywords:** Polyp segmentation, Convolutional attention, Adaptive feature fusion, Gating units

**Posted Date:** June 10th, 2024

**DOI:** https://doi.org/10.21203/rs.3.rs-4483283/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

**Additional Declarations:** No competing interests reported.
AFCNet: An adaptive fusion of composite attention convolutional neural network for polyp image segmentation

Bojiao Jin\textsuperscript{1}, Yi Zhang\textsuperscript{1,*}, Lin Qi\textsuperscript{1,+}, and Wei Qian\textsuperscript{1,+}

\textsuperscript{1}College of Medicine and Biological Information Engineering, Northeastern University, Shenyang, China
\textsuperscript{*}zhangyi@mail.neu.edu.cn
\textsuperscript{+}these authors contributed equally to this work

ABSTRACT

Accurately locating and segmenting polyps from colon images is crucial for the treatment of rectal cancer. However, the environment of rectal polyps is characterized by high noise, diverse sizes, complex boundaries, and a high demand for detailed information, making the task challenging. The acquisition and processing of polyp features are central to the research on polyp segmentation methods. This paper introduces an Adaptive Fusion Of Composite Attention Convolutional Neural Network (AFCNet) for polyp image segmentation. First, this work combines depth-wise separable convolutions and convolutional attention mechanisms with a multi-branch structure to better supplement missing details and unearth potential critical features. Secondly, we employ a multi-scale cross structure and an adaptive multi-scale feature harmonization module to balance the contribution of features at different levels, thus fully integrating information across scales to maximize the utilization of previously acquired features. Lastly, we propose an upsampling feature retrospective module to filter detailed information and use the concept of gating units to filter out interfering information. Extensive experiments on five publicly available polyp segmentation datasets demonstrate the effectiveness of our AFCNet in enhancing the accuracy of polyp segmentation. The experimental results indicate that AFCNet significantly outperforms state-of-the-art models. AFCNet is an effective polyp segmentation network, and due to its excellent generalization ability, it can also be applied to other medical image segmentation tasks.

Keywords:
Polyp segmentation, Convolutional attention, Adaptive feature fusion, Gating units

Introduction

Colorectal cancer is a common malignant tumor with an increasing incidence rate, posing a serious threat to human health. Therefore, the prevention of colorectal cancer has become an important topic. Studies have shown that polyps are often precancerous lesions in colorectal cancer. Early detection and removal of colorectal polyps is one of the most effective methods to reduce the incidence of colorectal cancer and improve the cure rate\textsuperscript{1}. Physicians rely on screening tools such as colonoscopy when diagnosing colon cancer. In clinical diagnostics, some polyps that are too small may be missed by the naked eye, potentially delaying the optimal timing for treatment\textsuperscript{2}. Automatic and precise polyp segmentation can help doctors accurately locate polyp areas within the colon\textsuperscript{1}, assisting in observation and preventing oversights. Therefore, polyp segmentation plays a crucial role in the early diagnosis of colorectal cancer.

Methods for colon polyp image segmentation must address a series of challenges, such as the complex shape of the colon and high variability. Moreover, colonoscopy images often contain noise, shadows, and other interfering factors, which further increase the complexity of accurately segmenting polyp areas. Traditional image processing methods such as threshold segmentation methods\textsuperscript{4} and region-growing methods\textsuperscript{5} are limited in effectiveness when dealing with these complex images. Therefore, more efficient and precise segmentation technologies are needed.

To solve this problem, researchers have introduced deep learning techniques into polyp image segmentation. U-networks such as Unet\textsuperscript{6} have been a great success in the field of medical segmentation. A U-shaped network structure consists of an encoder and a decoder. The encoder is responsible for extracting feature information. Many researchers have used CNN-based feature extraction encoders for images; for example, UNet++\textsuperscript{7} and Unet3+\textsuperscript{8} used nested dense skip connections to enhance the semantic connection between the encoder and decoder feature mappings. However, such networks often suffer from the lack of spatial information after pooling, and lack of long-distance dependency after multiple convolutions. Compared to CNN-based feature extraction networks, the Transformer has a stronger feature extraction capability and has become a powerful feature...
extraction network. Polyp-pvt, MSRAformer, and SSFormer introduce the Transformer as an encoder in the task of polyp segmentation, thereby optimizing the model’s segmentation performance.

During the decoding phase, the model cannot fully consider the importance of different feature information for the segmentation results, which is detrimental to the accuracy of the segmentation results. Additionally, features at different scales contain different information. Higher-level features contain more precise information about the polyp’s exact location, while lower-level features contain more spatial information highlighting the polyp’s boundary. After obtaining the feature information of the image, how to better integrate these different pieces of information poses a great challenge.

This paper proposes a new U-shaped network architecture that follows the encoding and decoding structure-AFCNet and introduces three functional modules to tackle the problems. The main contributions of this paper are as follows:

- To achieve accurate polyp segmentation, this paper proposes a polyp segmentation network based on convolutional attention and multi-scale feature adaptive fusion, addressing the characteristics of irregular polyp shapes, blurry edges, and complex backgrounds.

- A new Multi-scale Depth-wise Convolutional Attention Module (MDCA) is proposed: the MDCA module consists of a depth-separable convolutional and multi-branching network, which enhances the model’s focusing and utilization of important features. It avoids the inability of ordinary attention mechanisms to adequately capture the local features of an image.

- A new Multi-scale Adaptive Feature Fusion Module (MAFF) is proposed, which consists of a multi-scale cross-fusion network and an Adaptive Multi-Scale Feature Harmonization (AMFH) module. Through the multi-scale cross-fusion network, the feature advantages of different levels are formed to complement each other’s information, and through the adaptive multi-scale feature coordination module, the model provides a flexible way to synthesize and emphasize the feature information of different levels.

- We designed an Upsampling Feature Retrospective Module (UFR) by borrowing the idea of the gating unit. This module dynamically adjusts the fusion of information according to the content of the features themselves, effectively balancing the contribution between different features and allowing the model to better utilize contextual information.

Extensive experiments demonstrate that our method has considerable advantages over existing polyp segmentation methods in terms of segmentation performance and generalizability on five colorectal polyp datasets. In addition, we find that as the network deepens, too large a downsampling multiplier and too low a resolution of the feature map may lead to the loss of important details and feature information, which directly affects the segmentation accuracy of the model. Therefore, the network depth of one layer of encoder is reduced in the work of this paper. This is to ensure that the loss of information is minimized while capturing complex image content.

Related work

**Polyp segmentation network.**

Traditional segmentation algorithms such as Otsu’s method, Region Growing, Snake and other methods are sensitive to noise and image quality. Moreover, it is difficult to set and adjust the parameters, and the segmentation accuracy and detail capture are insufficient, resulting in low segmentation accuracy for polyps. Compared with these methods, deep learning methods can automatically learn the complex features of the image, better deal with noise, and do not need to manually adjust a large number of parameters.

Thus, deep learning methods provide more accurate and robust segmentation results in many application scenarios. With the continuous development of Convolutional Neural Networks (CNN), especially with the introduction of U-Net, many models inspired by this architecture have shown promising results in the field of medical image segmentation. U-Net gradually reduces the resolution of an image through a series of convolutional and pooling layers to capture the contextual information of the image. Then the resolution of the image is gradually restored by upsampling and convolution operations, combining high-resolution features with low-resolution features to achieve accurate pixel-level segmentation. EU-Net enhances semantic information by introducing a global context module for extracting key features. ACSNet modifies the skip connections in U-Net into a local context extraction module and adds a global information extraction module. CENet uses a ResNet pre-trained model as an encoder for feature extraction, fused with a context extraction module. It relies on Dense Cavity Convolutional Block (DAC module) and Residual Multi-Kernel Pooling (RMP module) to capture more abstract features and retain more spatial information, which ultimately results in improved performance of medical image segmentation.

Although CNN has been successful in the field of polyp segmentation, it has limitations in acquiring contextual remote information. Transformer, as a powerful image-understanding method, makes up for this deficiency well and is rapidly
developing in the field of polyp segmentation. Polyp-pvt is the first to introduce Transformer as a feature encoder in the field of polyp segmentation, and it combines high-level semantic and positional information through cascading fusion modules and similarity aggregation modules, thereby reducing noise in the features. DuAT is a dual-fusion Transformer network for polyp segmentation. It utilizes a global-to-local spatial aggregation module to combine global and local spatial features in order to localize targets of different sizes. In addition, it employs a selective boundary aggregation module to fuse the edge information at the bottom layer with the semantic information at the top layer. SSFormer combines SegFormer and PvT as an encoder and introduces a new progressive local decoder to emphasize the local features and alleviate the problem of distraction. TransNetR combines the residual network with the Transformer. The combination shows good real-time processing speed and multi-center generalization capability. However, most of the current work improves and experiments only on one kind of backbone network and ignores the generalization of the module. In contrast, the module proposed in this paper can be applied to different backbone networks, and all of them obtain stable performance improvements.

Attention Mechanism
By precisely focusing on key parts of an image, the attention mechanism enables deep learning models to identify polyps more efficiently and accurately when processing medical images, especially when processing colonoscopy images. Att-UNet integrates Attention into UNet and applies it to medical images, and for the first time, uses Soft Attention in a CNN network for medical imaging. DCRNet proposes a positional attention module to capture pixel-level contextual information. MSNet aggregates high-level features using a parallel partial decoder, mines boundary cues using a reverse attention module, and establishes relationships between regions and boundary cues. CaraNet combines axial reverse attention and channel feature pyramid (CFP) modules. This method helps to improve the segmentation performance of small medical targets. MSRF-NET uses a dual-scale dense fusion block to exchange multi-scale features with different receptive fields. It maintains the resolution and propagates high-level and low-level features for more accurate segmentation results.

ResNet, as an innovative architecture combining Residual Network (ResNet) and Split-attention, has been shown to have excellent results in semantic segmentation. ResNest, by introducing the Split-attention module, which effectively combines grouped convolution and attention mechanisms. This unique design allows the network to capture and utilize the spatial and channel features of an image in greater detail while maintaining computational efficiency. However, the application of the ResNest architecture has hardly been widely studied or adopted in the field of polyp segmentation. In this paper, it is applied as an advanced CNN backbone network in experiments to verify the advancement of the architecture in the field of polyp segmentation as well as the effectiveness and generalizability of the modules proposed in this paper.

Feature Fusion
Due to the complex shapes and different sizes of polyps, an effective method of fusing multi-scale features can substantially improve the segmentation effect of the model. DCRNet achieves feature enhancement by embedding a contextual relationship matrix and then achieves relationship fusion by region cross-batch memory. MSNet designs a phase reduction unit to generate the difference features between neighboring layers and pyramidally Equipped with different receptive fields to capture multi-scale information. CFA-Net uses a hierarchical strategy to incorporate edge features into a two-stream segmentation network while using a cross-layer feature fusion module to fuse neighboring features at different levels. Work such as PPNet and PolypSeg applies the attention mechanism to feature fusion to enhance the fusion of features in both the top and bottom layers. The gating mechanism is also an excellent method for feature fusion. Gated Fully Fusion and BANet achieve selective multi-level feature fusion by gated feature fusion. Taking the above studies together, it can be seen that how to effectively fuse and utilize the acquired features is an important challenge in polyp segmentation.

Methods
In this section, we provide a detailed introduction to the architecture of the AFCNet network and its constituent modules. Firstly, the overall structure of the network is presented in Figure 1. Subsequently, each part is described in detail, including the Multi-Scale Depth-wise Convolution Attention Module (MDCA module), the Multi-Scale Adaptive Feature Fusion module (MAFF), and the Upsampling Feature Retrospective Module (UFR).

Network architecture
The AFCNet we designed follows the classical encoder-decoder architecture. For the encoding part of the model, this chapter analyzes the traditional CNN network Res2Net50 as the backbone of our model. Unlike previous work, we find that rounding off the feature information at the highest level of the backbone network can be effective for model segmentation. This is because the loss of edges, textures, and other details during each downsampling process is detrimental to tasks that require a fine-grained understanding of the image content, and because reducing the number of downsampling layers allows for easier information fusion between high-level features and low-level features. We prove this idea in the experimental section.
Figure 1. The AFCNet network framework consists of four key parts: encoder network, Multi-scale Depth-wise Convolution Attention Module, Multi-scale Adaptive Feature Fusion Module, upsampling Feature Retrospective Module. The input image first obtains different layers of multi-scale features through Topical subheadings are allowed. Authors must ensure the encoder network, then obtains the important information of the different scale features through the Multi-scale Depth-wise Convolution Attention Module and splices them with the original features to obtain the feature $f_{k\text{Att}}$, and then sends $f_{k\text{Att}}$ to the Multi-scale Adaptive Feature Fusion Module. In the MAFF module, we use a multi-scale fusion cross-network to realize the dynamic interaction and complementarity between different scale features. Adaptive fusion of multi-scale features and key information is performed through the Adaptive Multi-scale Feature Harmonization module to obtain the aggregated information $f_{k\text{agg}}$. Finally, the fused features are sent to the Upsampling Feature Retrospective Module. The feature information is filtered and supplemented with different levels of information, and the up-sampling operation is carried out.

Therefore, we only use the first three layers of high-level features extracted by the backbone network. Suppose our input polyp segmentation image is $F \in \mathbb{R}^{H \times W}$. We utilize the feature information of each level $f_k \in \mathbb{R}^{H/2^k \times W/2^k}$ ($k \in [1, 3]$). The Multi-scale Depth-wise Convolutional Attention Module (MDCA) applies convolutional attention mechanisms to feature information at different hierarchical levels, gathering key information within the image while suppressing less significant elements. The MDCA module enhances the model’s feature representation for each pixel point in the input image by capturing multi-scale information through convolutional kernels of different sizes. Moreover, the enhanced attentional features and the original features are effectively fused in this module by a dense concatenation operation.

After subsequent feature enhancement by MDCA, the features $f_{k\text{m}}^m$ are input into the Multi-scale Adaptive Feature Fusion (MAFF) module. Within the MAFF module, a cross-network aligns features of different scales. Subsequently, the Adaptive Multi-Scale Feature Harmonization (AMFH) module performs weighted fusion on the adjusted feature maps, emphasizing differences and key information within the features to heighten the model’s sensitivity to image details. Through a 3x3 convolution, features across various scales are efficiently integrated. Finally, the multi-scale fused feature information is processed through a specially designed upsampling Feature Retrospective Module, effectively integrating features from different network layers while considering their dynamic interrelations, leading to the final segmentation prediction map. Our overall network structure is defined as follows:

$$f_k = \text{Res2Net}(F), k \in [1, 3]$$
Multi-scale Depth-wise Convolution Attention Module

In order to extract more important feature information from different layers of features, the MDCA module is designed in this paper. The module consists of a multi-branch parallel network and a multi-scale deep convolutional attention mechanism. As shown in Figure 2, we get the features $f_k$ from the encoder. First, $f_k$ is convolved by a depth-separable convolution with a convolution kernel size of $5 \times 5$ to get the spatial feature $f_k'$. Consequently, the obtained features $f_k'$ are fed into a multi-branch concurrent network structure. The multibranch parallel network consists of three different branches, and there are two depth directions of banded depth-separable convolution in each branch. Here, the size of the depth-separable convolution kernel in each branch is set to 7, 11, and 21, respectively. Capturing multi-scale contextual information in each branch through these different orientations and sizes of convolutions enables the network to capture a wider range of contextual information in the image and to understand the image features at different spatial scales. Thus, it is more sensitive to objects with diverse shapes and structures. We define depth-separable convolution as:

$$DWSConv_{m \times n}(f) = \phi(Conv_{m \times n}(f))$$

Where $\phi(\cdot)$ stands for point-by-point convolution and $Conv_{m \times n}$ stands for convolutional layers with convolutional kernel size $m \times n$. After sufficient extraction of the image information by the multi-branch network, the attention maps $f_{k7}', f_{k11}', f_{k21}'$ are obtained from different branches, respectively, and subsequently we sum the attention feature maps from different branches.
and multiply them with the input feature maps for feature optimization to obtain $f'_{k,\text{Att}}$. Finally, the module uses splicing to fuse the optimized features with the original features in the channel through an information aggregation section, followed by a final 3*3 convolution. The module integrates rich multi-scale information to enhance the model’s representation of contextual features. Mathematically, the MDCA module can be described as:

$$f'_k = \text{DWSC} \text{Conv}_{5 \times 5}(f_k)$$

(6)

$$f'_{k,7} = \text{DWSC} \text{Conv}_{7 \times 1}(\text{DWSC} \text{Conv}_{1 \times 7}(f'_k))$$

(7)

$$f'_{k,11} = \text{DWSC} \text{Conv}_{11 \times 1}(\text{DWSC} \text{Conv}_{1 \times 11}(f'_k))$$

(8)

$$f'_{k,21} = \text{DWSC} \text{Conv}_{21 \times 1}(\text{DWSC} \text{Conv}_{1 \times 21}(f'_k))$$

(9)

$$f'_{k,\text{Att}} = f_k \otimes (f'_k + f'_{k,7} + f'_{k,11} + f'_{k,21})$$

(10)

$$f_{k,\text{Att}} = \psi(\text{BN}(\text{MLP}_{3 \times 3}(\text{MLP}_{3 \times 3}(f'_k))))$$

(11)

where $f_k (k \in [1, 3])$ is a different hierarchical characterization of the input, $\text{DWSC} \text{Conv}_{m \times n}$ is depth-wise convolution, Cat represents the feature concatenation operation. $\psi(\cdot)$ means ReLU function, $\text{BN}$ denotes batch normalization, $\text{MLP}_{3 \times 3}(\cdot)$ means $\text{DWSC} \text{Conv}_{3 \times 3}$ and Concat.

**Multi-scale Adaptive Feature Fusion Module**

Due to the low contrast between polyps and surrounding tissues in some polyp endoscopic images, features extracted by traditional methods may have difficulty in distinguishing subtle differences between polyps and normal tissues. In order to fully utilize features at different scales and improve the richness of feature representation, we propose a multi-scale adaptive feature fusion module (MAFF). As illustrated in Figure 1, This module aims to enhance the model’s ability to capture detailed and global information by dynamically fusing multi-level feature maps.

MAFF consists of two parts: a multi-scale fusion cross-network and an Adaptive Multi-scale Feature Harmonization module. The multi-scale fusion cross-network part realizes dynamic interaction and complementarity between different scale features through its unique structure, thus providing a basis for the model to capture rich multilevel information. The Adaptive Multi-scale Feature Harmonization Module, as the core of this module, consists of two different feature algorithm units: a feature addition unit and a feature subtraction unit. Feature addition is a commonly used feature enhancement algorithm in the image domain, and in our module, the common information present in different levels of features is highlighted by performing addition operations on the features at different levels. The opposite feature subtraction unit is able to highlight the differences in information between features at different levels. In order to fully fuse these two complementary feature information, we introduce a trainable weighting ratio parameter, $W_i$. With the trainable parameter $W_i$, the module is able to achieve fine control of the feature fusion process, thus enhancing the model’s generalization ability and robustness to different endoscopic images. The MAFF module receives features $f'_{k,\text{Att}} (k \in [1, 3])$ from various levels that have been enriched with multi-scale information through integration by the MDCA module. After passing through the multi-scale fusion cross-network, the multi-scale cross-network unifies the scale of features from different levels via upsampling and downsampling using bilinear interpolation. It further refines the features through convolutional operations, enhancing the feature representation capability. To express this mathematically:

$$UP_k(f) = \psi(\text{BN}(\text{Conv}_{3 \times 3}(\text{ReLU}_{k}(f))))$$

(12)

$$\text{Down}_k(f) = \psi(\text{BN}(\text{Conv}_{3 \times 3}(\text{ReLU}_{1}(f))))$$

(13)
In the gated loop unit, the gating mechanism is used to control the flow of information. After obtaining the fused features, in order to dynamically adjust the amount of information fused in each scale so as to enhance the different levels of features through update gates and reset gates. We set the two inputs of the module to consist of a sequence of 3×3 convolution, BN means batch normalization, and \( \psi \) is the ReLU function. \( \mathcal{B} \) denotes the sampling method of bilinear interpolation.

We then put the aligned features into the AMFH (Adaptive Multi-scale Feature Harmonization) module. AMFH fuses two different features by feature addition and subtraction in order to efficiently capture the complementary information between different layers of features, highlight the subtle differences between them, and strengthen the module’s sensitivity to edges, textures, and other key visual details. We then enable the module to dynamically balance the effects of addition and subtraction operations on the final feature representation by introducing an adaptive weighting mechanism. This adaptivity is based on the unique properties of the input features and their contextual information, and the optimization of the weights is performed automatically. With the adaptive adjustment of the weights of addition and subtraction operations, the AMFH module takes full advantage of the complementary strengths of these two operations to produce feature representations that are both rich and fine-grained. We use \( X \) and \( Y \) as input features to the AMFH module, defining the AMFH function as:

\[
AMFH(X, Y) = \psi(BN(Conv_{3 \times 3}(|W_i \odot (X \oplus Y) + (1 - W_i) \odot (X \odot Y)|)))
\]

where \( \oplus \) is the element-by-element addition operation, \( \ominus \) is the element-by-element subtraction operation, \( \odot \) is the Hadamard product, \( W_i \) is the trainable parameter we set \( i \in [1, 6] \), \( \lvert \cdot \rvert \) computes the absolute value, where \( Conv_{3 \times 3} \) means the operation that consists of a sequence of 3×3 convolution, BN means batch normalization and \( \psi \) is ReLU function. After the AMFH module we can get three final outputs:

\[
\begin{align*}
    f_{1_{tv}} &= AMFH(AMFH(f_{1_{tv}}, f_{2_{tv}}, f_{3_{tv}}) \odot f_{4_{tv}}) \\
    f_{2_{tv}} &= AMFH(AMFH(f_{1_{tv}}, f_{2_{tv}}, f_{3_{tv}}) \odot f_{4_{tv}}) \\
    f_{3_{tv}} &= AMFH(AMFH(f_{1_{tv}}, f_{2_{tv}}, f_{3_{tv}}) \odot f_{4_{tv}})
\end{align*}
\]

**Upsampling Feature Retrospective Module**

After obtaining the fused features, in order to dynamically adjust the amount of information fused in each scale so as to realize more effective information integration, reduce spatial distortion, and enhance the semantic expression of the features in multi-scale feature fusion. We have designed the Up-sampling Feature Retrospective Module (UFR) based on the idea of the Gate Recurrent Unit. As shown in figure 3 In the gated loop unit, the gating mechanism is used to control the flow of information through the sequence model. We input different levels of features into the UFR module, respectively. The UFR module consists of an update gate module and a reset gate module, as well as a dense connection, which performs correlation enhancement of the different levels of features through update gates and reset gates. We set the two inputs of the module to be two neighboring features of different levels: \( X \) and \( Y \). Then the update gates and the reset gates at this point of time are computed as follows:

\[
Z = \sigma(\psi(BN(Conv_{3 \times 3}(X))))
\]

\[
R = \sigma(\psi(BN(Conv_{3 \times 3}(U p_{2}(Y))))))
\]

\[
\bar{R} = \mathcal{B}(R + \psi(BN(Conv_{3 \times 3}(R \odot X))))
\]

\[
H = Z \odot X + (1 - Z) \odot \bar{R}
\]
Figure 3. Upsampling Feature Retrospective (UFR) Module structure. It consists of update door unit, reset door unit and dense connections, the module uses a bilinear interpolation method to upsample features.

The obtained hidden vector \( H \) is used as one of the outputs of this layer and the inputs of the next layer. We define the above computational process as the \( \mathcal{G}(\cdot) \) function. In our module, we up-sample the bottom layer features by using linear interpolation, so as to align with the dimensions of the top layer features, where \( \sigma(\cdot) \) denotes Sigmoid function, \( \psi(\cdot) \) denotes ReLU function, \( \mathcal{T}(\cdot) \) denotes Tanh function. Our upsampling part can be expressed as:

\[
\begin{align*}
  f_{\text{up}}^{3} &= U_2(f_{\text{seg}}^{3}) \\
  f_{\text{up}}^{2} &= \mathcal{C}_{3 \times 3}(\mathcal{G}(f_{\text{seg}}^{2}, f_{\text{up}}^{3})) \\
  f_{\text{up}}^{1} &= \mathcal{C}_{3 \times 3}(\mathcal{G}(f_{\text{seg}}^{1}, f_{\text{up}}^{2})) \\
  \text{output} &= \psi(BN(\mathcal{C}_{3 \times 3}(f_{\text{up}}^{1})))
\end{align*}
\]

where \( \mathcal{C}_{3 \times 3} \) denotes Convolution with 3 \( \times \) 3 convolution kernel and Concat.

Results

To ensure the accuracy and practicality of our approach, in this section, we present the specific details of our experiments, the datasets, and the experimental results. This includes comparisons with eleven currently common and popular methods as benchmarks, as well as ablation studies and generalization experiments to validate the effectiveness of our method.

Dataset

According to the\(^{32}\), we selected five publicly available datasets commonly used in the field of polyp segmentation: Kvasir, CVC-ClinicDB, CVC-ColonDB, CVC-300, and ETIS.

Kvasir-SEG\(^{33}\): It is an open-access dataset of gastrointestinal polyp images and corresponding segmentation masks, manually annotated and verified by an experienced gastroenterologist. It contains 1000 polyp images and their corresponding...
ground truth from the Kvasir Dataset v2. The resolution of the images contained in Kvasir-SEG varies from 332x487 to 1920x1072 pixels.

CVC-ClinicDB\textsuperscript{34}: CVC-ClinicDB is a database of frames extracted from colonoscopy videos. These frames contain several examples of polyps. The CVC-ClinicDB dataset contains 612 images cut from 25 colonoscopy videos with an image size of 384 × 288 and polyps ranging from 0.34% to 45.88% in size.

CVC-ColonDB\textsuperscript{35}: The CVC-ColonDB dataset consists of 380 images cut from 15 colonoscopy videos with an image size of 574 × 500 and the polypl size of 0.30%–63.15%.

ETIS\textsuperscript{36}: ETIS contains 196 images cut from 34 colonoscopy videos with the image size of 1225 × 996. The highest resolution compared to other datasets. But the size of polyps in its images is only 0.11%–29.05%, the smallest, making this dataset also more challenging.

CVC-300\textsuperscript{37}: includes 60 colonoscopy images with a resolution of 500 × 574.

To evaluate the segmentation performance of the method, we conducted experiments on two polyp segmentation datasets, Kvasir-SEG and CVC-612. For each dataset, we randomly divided it into three subsets: 90% for the training set and the remaining 10% for the test set. To verify the generalizability of our model to data, we followed the experimental method of PraNet\textsuperscript{22}, extracting 900 and 550 images from the CVC-ClinicDB and Kvasir-SEG datasets, respectively, to form a training set of 1450 images. Meanwhile, we used the CVC-ColonDB, CVC-300, and ETIS datasets as test sets to validate the model’s generalizability on different datasets.

Training setup and experimental metrics

All of our experimental models are implemented under pytorch 2.0.0 and train for 200 epochs on an RTX4090 graphics card with 24G of memory. Throughout the training regimen, we use four basic data augmentation techniques, random rotations, horizontal flips, vertical flips, and coarse masking, to enhance the model’s robustness to variations in the input data. and use an Adam optimizer with the learning rate of 1e-4, and using the ReduceLROnPlateau learning rate scheduler.

In our experiments, four separate experiments are conducted for each model, using four fixed random seeds: 42, 8, 36, and 120. all results presented in the table are averages of these four experiments, and the variance is calculated. We combine cross-entropy loss and Dice loss as our assessment metrics for the loss function. To validate the effectiveness of our model, we have selected five metrics to evaluate the model’s performance from multiple perspectives: Dice Similarity Coefficient (Dice), Intersection over Union of polypl (IoUp), recall, Accuracy (ACC), and True Negative Ratio (TNR). Let FN, FP, TN, and TP denote false negatives, false positives, true negatives, and true positives, respectively. By definition, Dice, IoUp, recall, ACC, and TNR can be calculated as follows:

\[
\text{Dice} = \frac{2TP}{FP + FN + 2TP} \tag{25}
\]

\[
\text{IoUp} = \frac{TP}{FP + FN + TP} \tag{26}
\]

\[
\text{recall} = \frac{TP}{TP + FN} \tag{27}
\]

\[
\text{ACC} = \frac{TP + TN}{FP + TP + TN + FN} \tag{28}
\]

\[
\text{TNR} = \frac{TN}{FP + TN} \tag{29}
\]

Generally, a superior segmentation method has larger values of Dice and IoUp.

Comparisons with state-of-the-art methods

We select the following networks including Unet++\textsuperscript{7}, Unet3+\textsuperscript{8}, Attention-UNet\textsuperscript{12} (AttUNet), Context Encoder Network\textsuperscript{18} (CENet), Local Global Interaction Network\textsuperscript{38} (LGI Net), Multi-scale Subtraction Network\textsuperscript{26} (MSNet), Duplex Contextual Relation Network\textsuperscript{21} (DCRNet), Dual-Aggregation Transformer Network\textsuperscript{19} (DuAT), Polyp-pvt\textsuperscript{9}, Transformer-based Residual Network\textsuperscript{20} (TransNetR), Context axial reverse attention network(CaraNet)\textsuperscript{23}, as 11 state-of-the-art segmentation methods for comparison.
Table 1. Comparison of our designed model AFCNet with currently popular methods on the CVC-ClinicDB dataset. ([In %] and “±” for variance.)

<table>
<thead>
<tr>
<th>Models</th>
<th>backbone</th>
<th>recall</th>
<th>TNR</th>
<th>Dice</th>
<th>ACC</th>
<th>IoU p</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNet+</td>
<td>-</td>
<td>89.37 ± 0.69</td>
<td>99.33 ± 0.11</td>
<td>88.52 ± 0.10</td>
<td>98.73 ± 0.08</td>
<td>82.87 ± 0.13</td>
</tr>
<tr>
<td>U-net3+</td>
<td>-</td>
<td>87.59 ± 0.68</td>
<td>99.19 ± 0.11</td>
<td>86.84 ± 0.85</td>
<td>98.51 ± 0.08</td>
<td>80.67 ± 1.01</td>
</tr>
<tr>
<td>AttUNet12</td>
<td>-</td>
<td>89.49 ± 1.15</td>
<td>99.22 ± 0.13</td>
<td>88.38 ± 1.37</td>
<td>98.58 ± 0.01</td>
<td>82.53 ± 1.27</td>
</tr>
<tr>
<td>CENet16</td>
<td>ResNet-34</td>
<td>93.46 ± 0.60</td>
<td>99.35 ± 0.08</td>
<td>91.76 ± 0.70</td>
<td>99.07 ± 0.08</td>
<td>86.67 ± 1.05</td>
</tr>
<tr>
<td>LGINet28</td>
<td>-</td>
<td>88.65 ± 1.50</td>
<td>99.03 ± 0.21</td>
<td>87.65 ± 1.08</td>
<td>98.58 ± 0.11</td>
<td>81.52 ± 1.62</td>
</tr>
<tr>
<td>DCRNet21</td>
<td>ResNet-34</td>
<td>94.13 ± 1.66</td>
<td>99.52 ± 0.13</td>
<td>92.83 ± 0.75</td>
<td>99.24 ± 0.08</td>
<td>88.37 ± 0.60</td>
</tr>
<tr>
<td>MSNet40</td>
<td>Res2Net-50</td>
<td>94.33 ± 0.29</td>
<td>99.54 ± 0.06</td>
<td>93.66 ± 0.52</td>
<td>99.26 ± 0.05</td>
<td>88.96 ± 0.46</td>
</tr>
<tr>
<td>TransNet20</td>
<td>ResNet-50</td>
<td>93.18 ± 1.48</td>
<td>99.44 ± 0.07</td>
<td>92.13 ± 0.79</td>
<td>99.17 ± 0.04</td>
<td>87.56 ± 0.73</td>
</tr>
<tr>
<td>CaraNet23</td>
<td>ResNet-50</td>
<td>95.21 ± 0.84</td>
<td>99.47 ± 0.06</td>
<td>93.08 ± 0.65</td>
<td>99.22 ± 0.04</td>
<td>88.37 ± 0.64</td>
</tr>
<tr>
<td>Polyp − pvt21</td>
<td>PVT</td>
<td>95.48 ± 0.73</td>
<td>99.29 ± 0.14</td>
<td>92.15 ± 0.99</td>
<td>99.13 ± 0.10</td>
<td>87.03 ± 1.24</td>
</tr>
<tr>
<td>DuAF19</td>
<td>PVT</td>
<td>94.93 ± 0.81</td>
<td>99.49 ± 0.11</td>
<td>93.06 ± 0.48</td>
<td>99.26 ± 0.05</td>
<td>88.29 ± 0.71</td>
</tr>
<tr>
<td>AFCNet (ours)</td>
<td>Res2Net-50</td>
<td>94.54 ± 0.96</td>
<td><strong>99.61 ± 0.07</strong></td>
<td>94.48 ± 0.22</td>
<td>99.33 ± 0.07</td>
<td>89.88 ± 0.33</td>
</tr>
<tr>
<td>AFCNet (ours)</td>
<td>ResNet-50</td>
<td>95.33 ± 0.67</td>
<td>99.60 ± 0.06</td>
<td>94.64 ± 0.71</td>
<td>99.36 ± 0.07</td>
<td>90.46 ± 0.89</td>
</tr>
<tr>
<td>AFCNet (ours)</td>
<td>PVT</td>
<td><strong>95.79 ± 0.24</strong></td>
<td>99.59 ± 0.03</td>
<td><strong>94.78 ± 0.19</strong></td>
<td><strong>99.37 ± 0.03</strong></td>
<td><strong>90.59 ± 0.16</strong></td>
</tr>
</tbody>
</table>

Table 2. Comparison of our designed model AFCNet with currently popular methods on the Kvasir-SEG dataset. ([In %] and “±” for variance.)

<table>
<thead>
<tr>
<th>Models</th>
<th>backbone</th>
<th>recall</th>
<th>TNR</th>
<th>Dice</th>
<th>ACC</th>
<th>IoU p</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNet+</td>
<td>-</td>
<td>86.12 ± 0.79</td>
<td>98.32 ± 0.13</td>
<td>85.57 ± 1.09</td>
<td>96.00 ± 0.27</td>
<td>78.60 ± 1.40</td>
</tr>
<tr>
<td>U-net3+</td>
<td>-</td>
<td>82.94 ± 0.55</td>
<td>97.72 ± 0.42</td>
<td>81.02 ± 1.35</td>
<td>94.81 ± 0.34</td>
<td>72.77 ± 1.60</td>
</tr>
<tr>
<td>AttUNet12</td>
<td>-</td>
<td>87.47 ± 0.94</td>
<td>98.17 ± 0.26</td>
<td>84.69 ± 0.62</td>
<td>96.11 ± 0.21</td>
<td>79.62 ± 0.60</td>
</tr>
<tr>
<td>CENet16</td>
<td>ResNet-34</td>
<td>89.80 ± 0.56</td>
<td>98.19 ± 0.31</td>
<td>89.66 ± 0.37</td>
<td>96.89 ± 0.25</td>
<td>83.41 ± 0.54</td>
</tr>
<tr>
<td>LGINet28</td>
<td>-</td>
<td>88.42 ± 0.76</td>
<td>97.89 ± 0.44</td>
<td>87.19 ± 0.32</td>
<td>96.16 ± 0.34</td>
<td>80.72 ± 1.52</td>
</tr>
<tr>
<td>DCRNet21</td>
<td>ResNet-34</td>
<td>90.18 ± 1.50</td>
<td>97.77 ± 0.51</td>
<td>88.78 ± 0.96</td>
<td>96.50 ± 0.32</td>
<td>82.87 ± 1.00</td>
</tr>
<tr>
<td>MSNet40</td>
<td>Res2Net-50</td>
<td>89.91 ± 0.95</td>
<td>98.58 ± 0.39</td>
<td>89.41 ± 0.72</td>
<td>96.79 ± 0.22</td>
<td>84.01 ± 0.75</td>
</tr>
<tr>
<td>TransNet20</td>
<td>ResNet-50</td>
<td>89.25 ± 0.83</td>
<td>98.30 ± 0.33</td>
<td>88.57 ± 0.38</td>
<td>96.52 ± 0.15</td>
<td>82.35 ± 0.44</td>
</tr>
<tr>
<td>CaraNet23</td>
<td>ResNet-50</td>
<td>90.78 ± 1.01</td>
<td>98.45 ± 0.33</td>
<td>89.57 ± 0.62</td>
<td>96.85 ± 0.22</td>
<td>83.58 ± 0.69</td>
</tr>
<tr>
<td>Polyp − pvt21</td>
<td>PVT</td>
<td>92.51 ± 0.92</td>
<td><strong>99.01 ± 0.43</strong></td>
<td>91.68 ± 0.30</td>
<td>97.38 ± 0.26</td>
<td>86.51 ± 0.42</td>
</tr>
<tr>
<td>DuAF19</td>
<td>PVT</td>
<td>91.67 ± 1.19</td>
<td>98.61 ± 0.28</td>
<td>91.29 ± 0.34</td>
<td>97.32 ± 0.15</td>
<td>86.11 ± 0.41</td>
</tr>
<tr>
<td>AFCNet (ours)</td>
<td>Res2Net-50</td>
<td>90.81 ± 0.72</td>
<td>98.79 ± 0.21</td>
<td>90.48 ± 0.15</td>
<td>97.17 ± 0.07</td>
<td>85.12 ± 0.28</td>
</tr>
<tr>
<td>AFCNet (ours)</td>
<td>ResNet-50</td>
<td>92.10 ± 0.84</td>
<td>98.76 ± 0.22</td>
<td>91.44 ± 0.30</td>
<td>97.49 ± 0.07</td>
<td>86.13 ± 0.38</td>
</tr>
<tr>
<td>AFCNet (ours)</td>
<td>PVT</td>
<td><strong>92.51 ± 0.59</strong></td>
<td>98.74 ± 0.18</td>
<td><strong>92.35 ± 0.49</strong></td>
<td><strong>97.55 ± 0.10</strong></td>
<td><strong>87.53 ± 0.30</strong></td>
</tr>
</tbody>
</table>

These methods represent the current state-of-the-art in the field of medical image segmentation and cover a wide range of advanced techniques based on Convolutional Neural Networks (CNNs) and Transformer-based Residual Networks (Transformers). In order to fully validate the state-of-the-art of our proposed methods, this experiment has chosen three different backbone networks: Res2Net50, ResNet50, and PVT. Among them, Res2Net50, as a classical CNN architecture, provides a solid baseline for the comparison experiments. ResNet50 and PVT represent more advanced CNN and Transformer architectures, which are used to evaluate the impact of different network structures on model performance. Notably, our approach demonstrates significant performance improvement on both datasets while integrating these diverse backbone networks, which not only proves the effectiveness of our approach, but also highlights its sophistication and flexibility when dealing with medical image segmentation tasks. By comparing Table 1 and Table 2, it can be seen that among the models using CNN networks as backbones, AFCNet outperforms the state-of-the-art models on both datasets, whether using Res2Net50 or ResNet50. When using a Transformer as the backbone, AFCNet also achieves the best results. Specifically, on the ClinicDB dataset, our model improves over the state-of-the-art network by at least 1.72% in Dice factor, 2.3% in IoU. On the Kvasir-SEG dataset, the Dice coefficient is improved by 0.57%, IoU by 0.94%.

To demonstrate the state-of-the-art of our model, we show in Figure 4 the variation curves of two key performance metrics of the model when using different backbones as encoder. These two metrics include IoU curve, Dice curve. We have divided the graphs into two main categories, one for backbone based on the CNN technology family, and the other for backbone based on the Transformer technology. In both categories, we not only show the performance curves of our own model, but also include the curves of two state-of-the-art models using the same backbone technology as well as the curves of the baseline model for comparison. It is clear from the curves that our model is optimal regardless of which backbone is used. Based on the findings in previous experiments, our models show the best results when using PVT as the backbone network. Therefore, in the experiments on data generalizability, we will directly use the experimental results of PVT as the backbone network to compare with other models. Figure 5 and Figure 6 shows the effect of our validation on the two datasets.
**Generalisability experiments**

The generalization ability of Computer-Aided Diagnosis (CAD) systems is crucial in clinical applications. To validate the generalization ability of AFCNet, we followed the experimental methodology of PraNet\textsuperscript{22}. We selected 550 images from CVC ClinicDB and 900 images from Kvasir, forming a training set of 1450 images. To verify the network’s generalization performance, we used the entire ETIS, CVC ColonDB, and CVC-300 datasets as unseen data for testing. As shown in Table 3, Table 4, and Table 5, relative to the current popular networks, AFCNet improves Dice by 3.73\%, IoUp by 4.62\% on the ETIS dataset, and on the CVC-ColonDB dataset set, Dice improves by 0.91\%, IoUp improves by 0.71\%, and on the CVC-300 dataset, Dice improves by 0.46\%, IoUp improves by 0.94\%. It is clearly seen that our method achieves the best results on all three datasets, which shows that our method has good learning ability with more robust generalization performance.

<table>
<thead>
<tr>
<th>Models</th>
<th>recall</th>
<th>TNR</th>
<th>Dice</th>
<th>ACC</th>
<th>IoUp</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNet +++++</td>
<td>64.65 ± 1.24</td>
<td>98.29 ± 0.25</td>
<td>59.89 ± 1.77</td>
<td>94.96 ± 0.28</td>
<td>51.81 ± 1.58</td>
</tr>
<tr>
<td>Unet3+\textsuperscript{8}</td>
<td>61.75 ± 1.87</td>
<td>97.89 ± 0.59</td>
<td>55.58 ± 2.65</td>
<td>94.53 ± 0.50</td>
<td>47.16 ± 3.14</td>
</tr>
<tr>
<td>AttUNet\textsuperscript{12}</td>
<td>64.86 ± 1.04</td>
<td>98.69 ± 0.25</td>
<td>61.36 ± 0.96</td>
<td>95.35 ± 0.13</td>
<td>53.51 ± 1.14</td>
</tr>
<tr>
<td>CENet\textsuperscript{18}</td>
<td>72.85 ± 2.11</td>
<td>99.22 ± 0.18</td>
<td>71.03 ± 1.67</td>
<td>95.95 ± 0.24</td>
<td>63.45 ± 1.51</td>
</tr>
<tr>
<td>LGIUNet\textsuperscript{38}</td>
<td>70.69 ± 1.82</td>
<td>98.09 ± 0.35</td>
<td>64.92 ± 1.67</td>
<td>95.46 ± 0.31</td>
<td>56.98 ± 1.54</td>
</tr>
<tr>
<td>DCRNet\textsuperscript{44}</td>
<td>77.48 ± 4.07</td>
<td>98.60 ± 0.66</td>
<td>73.67 ± 1.99</td>
<td>96.13 ± 0.17</td>
<td>65.68 ± 1.62</td>
</tr>
<tr>
<td>MSNet\textsuperscript{20}</td>
<td>70.45 ± 2.06</td>
<td>99.52 ± 0.07</td>
<td>71.20 ± 1.65</td>
<td>96.19 ± 0.27</td>
<td>64.19 ± 1.70</td>
</tr>
<tr>
<td>TransNet\textsuperscript{R0\textsuperscript{43}}</td>
<td>64.79 ± 1.76</td>
<td><strong>99.59 ± 0.07</strong></td>
<td>66.23 ± 1.53</td>
<td>95.69 ± 0.04</td>
<td>59.29 ± 1.41</td>
</tr>
<tr>
<td>CaraNet\textsuperscript{73}</td>
<td>76.35 ± 2.74</td>
<td>99.09 ± 0.11</td>
<td>73.57 ± 3.01</td>
<td>96.22 ± 0.21</td>
<td>65.91 ± 2.73</td>
</tr>
<tr>
<td>PolyUp - pvt\textsuperscript{-1}</td>
<td>80.31 ± 0.96</td>
<td>98.95 ± 0.37</td>
<td>77.88 ± 1.00</td>
<td>96.94 ± 0.33</td>
<td>69.96 ± 0.94</td>
</tr>
<tr>
<td>DuAT\textsuperscript{79}</td>
<td>80.46 ± 0.82</td>
<td>98.59 ± 0.29</td>
<td>77.37 ± 0.76</td>
<td>96.63 ± 0.19</td>
<td>69.03 ± 0.77</td>
</tr>
<tr>
<td>AFCNet</td>
<td><strong>81.26 ± 1.16</strong></td>
<td>98.86 ± 0.10</td>
<td><strong>78.79 ± 0.35</strong></td>
<td><strong>96.99 ± 0.13</strong></td>
<td><strong>70.67 ± 0.17</strong></td>
</tr>
</tbody>
</table>

**Table 3.** Comparison of our designed model AFCNet with currently popular methods on the CVC-ColonDB dataset.([In %] and “±” for variance.)

**Ablation experiments**

In order to verify the validity of each module designed in the experiment and to study its effectiveness in polyp segmentation, we use the control variable method to validate it on two datasets, CVC-ClinicDB and Kvasir-SEG. And in order to verify that the module in the experiment is effective for different backbones, we validate it on three backbones, Res2Net50, ResNest50 and PVT2. First, we keep the complete number of network layers of the backbone network used for feature extraction and remove all the modules, keeping only the U-shaped architecture of the encoder-decoder, and then we the viewpoint that with the deepening of the network, an excessive downsampling factor and excessively low resolution of the feature maps may lead to the loss of important details and feature information. Through experiments, we find that when we reduce the depth of the backbone network by one layer, we can achieve better results. And then we add the MAFF module, MDCAModule and UFR module in turn, Table 6 and Table 7 shows the effect of each module on the accuracy of polyp segmentation. It is clear that all of our proposed modules favor the final prediction. Combining all these modules, our model achieves the highest performance.

<table>
<thead>
<tr>
<th>Models</th>
<th>recall</th>
<th>TNR</th>
<th>Dice</th>
<th>ACC</th>
<th>IoUp</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNet +++++</td>
<td>43.42 ± 3.88</td>
<td>98.72 ± 0.16</td>
<td>39.31 ± 4.23</td>
<td>96.88 ± 0.19</td>
<td>33.70 ± 3.58</td>
</tr>
<tr>
<td>Unet3+\textsuperscript{8}</td>
<td>59.05 ± 3.14</td>
<td>98.24 ± 1.13</td>
<td>56.44 ± 1.93</td>
<td>97.18 ± 1.02</td>
<td>48.99 ± 1.87</td>
</tr>
<tr>
<td>AttUNet\textsuperscript{12}</td>
<td>44.28 ± 2.25</td>
<td>98.99 ± 0.15</td>
<td>40.62 ± 1.11</td>
<td>97.09 ± 0.10</td>
<td>35.48 ± 1.09</td>
</tr>
<tr>
<td>CENet\textsuperscript{18}</td>
<td>66.98 ± 4.20</td>
<td>98.62 ± 0.59</td>
<td>62.32 ± 2.61</td>
<td>97.85 ± 0.46</td>
<td>55.29 ± 1.91</td>
</tr>
<tr>
<td>LGIUNet\textsuperscript{38}</td>
<td>47.90 ± 2.32</td>
<td>98.18 ± 0.64</td>
<td>42.76 ± 3.91</td>
<td>96.70 ± 0.72</td>
<td>37.27 ± 3.65</td>
</tr>
<tr>
<td>DCRNet\textsuperscript{44}</td>
<td>67.79 ± 4.25</td>
<td>97.96 ± 1.32</td>
<td>59.37 ± 2.14</td>
<td>97.19 ± 1.15</td>
<td>52.59 ± 2.50</td>
</tr>
<tr>
<td>MSNet\textsuperscript{20}</td>
<td>73.35 ± 3.56</td>
<td>99.10 ± 3.86</td>
<td>69.08 ± 2.06</td>
<td>98.50 ± 0.32</td>
<td>61.92 ± 1.54</td>
</tr>
<tr>
<td>TransNet\textsuperscript{R0\textsuperscript{43}}</td>
<td>58.85 ± 4.01</td>
<td><strong>99.43 ± 0.10</strong></td>
<td>57.12 ± 3.67</td>
<td>98.32 ± 0.20</td>
<td>51.15 ± 3.29</td>
</tr>
<tr>
<td>CaraNet\textsuperscript{73}</td>
<td>83.01 ± 3.86</td>
<td>97.19 ± 1.28</td>
<td>68.86 ± 2.04</td>
<td>96.84 ± 1.16</td>
<td>60.52 ± 1.93</td>
</tr>
<tr>
<td>PolyUp - pvt\textsuperscript{-1}</td>
<td>82.10 ± 1.62</td>
<td>98.47 ± 0.44</td>
<td>73.00 ± 1.97</td>
<td>98.21 ± 0.41</td>
<td>64.64 ± 2.22</td>
</tr>
<tr>
<td>DuAT\textsuperscript{79}</td>
<td>78.97 ± 1.19</td>
<td>98.69 ± 0.19</td>
<td>72.43 ± 2.04</td>
<td>98.40 ± 0.19</td>
<td>62.95 ± 2.75</td>
</tr>
<tr>
<td>AFCNet</td>
<td><strong>83.09 ± 2.56</strong></td>
<td>98.81 ± 0.26</td>
<td><strong>76.73 ± 0.91</strong></td>
<td><strong>98.51 ± 0.17</strong></td>
<td><strong>69.26 ± 0.56</strong></td>
</tr>
</tbody>
</table>

**Table 4.** Comparison of our designed model AFCNet with currently popular methods on the ETIS dataset.([In %] and “±” for variance.)
Figure 4. Change curves for the two KPIs when modeled using different backbones as encoders, as well as for the baseline model and two advanced models using the corresponding backbones on CVC-ClinicDB dataset.

Figure 5. Qualitative results comparison with state-of-the-art methods on eleven different benchmarks in CVC-ClinicDB datasets.

Figure 6. Qualitative results comparison with state-of-the-art methods on eleven different benchmarks in Kvasir-SEG datasets.
Table 5. Comparison of our designed model AFCNet with currently popular methods on the CVC-300 dataset. ([In %] and “±” for variance.)

<table>
<thead>
<tr>
<th>Models</th>
<th>recall</th>
<th>TNR</th>
<th>Dice</th>
<th>ACC</th>
<th>IoU p</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNet + ++T</td>
<td>80.64 ± 3.17</td>
<td>98.95 ± 0.32</td>
<td>73.01 ± 1.53</td>
<td>98.27 ± 0.28</td>
<td>64.20 ± 1.52</td>
</tr>
<tr>
<td>Uner3 + ++T</td>
<td>79.44 ± 3.63</td>
<td>98.11 ± 1.04</td>
<td>68.24 ± 3.32</td>
<td>97.51 ± 0.87</td>
<td>58.95 ± 3.60</td>
</tr>
<tr>
<td>AttrUNet ++T</td>
<td>79.18 ± 3.00</td>
<td>98.97 ± 0.36</td>
<td>72.30 ± 1.89</td>
<td>98.25 ± 0.30</td>
<td>64.40 ± 2.06</td>
</tr>
<tr>
<td>CENet ++T</td>
<td>90.14 ± 4.11</td>
<td>99.15 ± 0.36</td>
<td>84.60 ± 0.80</td>
<td>98.89 ± 0.31</td>
<td>77.07 ± 0.99</td>
</tr>
<tr>
<td>LGINet ++T</td>
<td>88.32 ± 2.78</td>
<td>98.74 ± 0.34</td>
<td>78.86 ± 1.64</td>
<td>98.48 ± 0.24</td>
<td>70.09 ± 1.37</td>
</tr>
<tr>
<td>DCRNet ++T</td>
<td>94.69 ± 1.72</td>
<td>99.13 ± 0.37</td>
<td>86.63 ± 1.86</td>
<td>98.96 ± 0.32</td>
<td>79.36 ± 1.82</td>
</tr>
<tr>
<td>MSNet ++T</td>
<td>93.08 ± 0.81</td>
<td>99.49 ± 0.12</td>
<td>88.78 ± 0.73</td>
<td>99.28 ± 0.10</td>
<td>81.57 ± 0.10</td>
</tr>
<tr>
<td>TransNet ++T</td>
<td>89.58 ± 1.84</td>
<td>99.54 ± 0.10</td>
<td>87.24 ± 1.06</td>
<td>99.23 ± 0.06</td>
<td>79.86 ± 0.97</td>
</tr>
<tr>
<td>CaraNet ++T</td>
<td>96.16 ± 0.54</td>
<td>99.09 ± 0.19</td>
<td>86.74 ± 0.41</td>
<td>99.00 ± 0.16</td>
<td>79.14 ± 0.47</td>
</tr>
<tr>
<td>PolyPavan ++T</td>
<td>94.37 ± 0.36</td>
<td>99.36 ± 0.11</td>
<td>87.63 ± 0.58</td>
<td>99.19 ± 0.10</td>
<td>80.27 ± 0.81</td>
</tr>
<tr>
<td>DuAT ++T</td>
<td>94.21 ± 1.16</td>
<td>99.01 ± 0.32</td>
<td>86.44 ± 0.61</td>
<td>98.85 ± 0.28</td>
<td>79.19 ± 0.43</td>
</tr>
<tr>
<td>AFCNet</td>
<td>94.54 ± 0.46</td>
<td>99.55 ± 0.08</td>
<td>89.24 ± 0.53</td>
<td>99.38 ± 0.04</td>
<td>82.51 ± 0.55</td>
</tr>
</tbody>
</table>

Table 6. Comparison of our designed model AFCNet with currently popular methods on the CVC-ClinicDB dataset. ([In %] and “±” for variance.)

<table>
<thead>
<tr>
<th>backbone</th>
<th>Models</th>
<th>recall</th>
<th>TNR</th>
<th>Dice</th>
<th>ACC</th>
<th>IoU p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Res2Net-50</td>
<td>baseline, ( \frac{n}{4} \times \frac{n}{4} )</td>
<td>90.40 ± 0.52</td>
<td>99.21 ± 0.15</td>
<td>89.44 ± 0.88</td>
<td>98.86 ± 0.11</td>
<td>83.57 ± 1.28</td>
</tr>
<tr>
<td></td>
<td>+MAFF</td>
<td>94.30 ± 0.52</td>
<td>99.57 ± 0.05</td>
<td>94.01 ± 0.52</td>
<td>99.32 ± 0.04</td>
<td>85.55 ± 0.40</td>
</tr>
<tr>
<td></td>
<td>+MAFF + MDCA</td>
<td>94.33 ± 0.68</td>
<td>99.63 ± 0.06</td>
<td>94.33 ± 0.10</td>
<td>99.33 ± 0.07</td>
<td>89.70 ± 0.29</td>
</tr>
<tr>
<td></td>
<td>+MAFF + MDCA + UFR</td>
<td>94.54 ± 0.96</td>
<td>99.61 ± 0.07</td>
<td>94.48 ± 0.22</td>
<td>99.34 ± 0.06</td>
<td>89.88 ± 0.33</td>
</tr>
<tr>
<td>ResNest-50</td>
<td>baseline, ( \frac{n}{4} \times \frac{n}{4} )</td>
<td>92.15 ± 0.64</td>
<td>99.40 ± 0.04</td>
<td>91.18 ± 0.50</td>
<td>99.04 ± 0.04</td>
<td>85.8 ± 0.56</td>
</tr>
<tr>
<td></td>
<td>+MAFF</td>
<td>95.13 ± 0.71</td>
<td>99.58 ± 0.07</td>
<td>94.34 ± 0.72</td>
<td>99.34 ± 0.05</td>
<td>90.07 ± 0.83</td>
</tr>
<tr>
<td></td>
<td>+MAFF + MDCA</td>
<td>95.16 ± 0.75</td>
<td>99.60 ± 0.06</td>
<td>94.64 ± 0.37</td>
<td>99.36 ± 0.07</td>
<td>90.36 ± 0.63</td>
</tr>
<tr>
<td></td>
<td>+MAFF + MDCA + UFR</td>
<td>95.33 ± 0.67</td>
<td>99.61 ± 0.05</td>
<td>94.74 ± 0.71</td>
<td>99.37 ± 0.06</td>
<td>90.47 ± 0.89</td>
</tr>
<tr>
<td>PVT</td>
<td>baseline, ( \frac{n}{4} \times \frac{n}{4} )</td>
<td>91.10 ± 0.64</td>
<td>99.40 ± 0.04</td>
<td>91.48 ± 0.82</td>
<td>99.04 ± 0.04</td>
<td>85.94 ± 0.82</td>
</tr>
<tr>
<td></td>
<td>+MAFF</td>
<td>92.58 ± 1.03</td>
<td>99.53 ± 0.06</td>
<td>92.25 ± 0.51</td>
<td>99.21 ± 0.02</td>
<td>87.36 ± 0.60</td>
</tr>
<tr>
<td></td>
<td>+MAFF + MDCA</td>
<td>95.85 ± 0.32</td>
<td>99.55 ± 0.05</td>
<td>94.11 ± 0.35</td>
<td>99.33 ± 0.02</td>
<td>89.95 ± 0.34</td>
</tr>
<tr>
<td></td>
<td>+MAFF + MDCA + UFR</td>
<td>95.79 ± 0.24</td>
<td>99.59 ± 0.03</td>
<td>94.78 ± 0.19</td>
<td>99.37 ± 0.03</td>
<td>90.59 ± 0.16</td>
</tr>
</tbody>
</table>

Table 7. Ablation study for the various modules with different backbone on Kvasir-SEG dataset. ([In %] and “±” for variance.)

**Effectiveness of MAFF module:** In order to verify the effectiveness of the MAFF module in our model, we directly input the multi-layer features extracted from the backbone network into the MAFF module and directly up-sampled them afterwards. From the table, we can see that all the metrics of the model with the addition of the MAFF module significantly outperform the
baseline model, no matter on different datasets or different backbone network architectures. This is due to the fact that the MAFF module is able to dynamically balance the impact of the two feature fusion methods on the final feature representation through the trainable parameters, which in turn makes the two methods complementary to each other. It is verified that the MAFF module is an effective method for multi-scale feature fusion.

**Effectiveness of the MDCA module:** After the model is added to the MDCA module, as shown in Table 6 and Table 7, the segmentation ability of the model has a more obvious improvement, which indicates that the important information in the image can be well extracted by our MDCA module, this is because the convolution with different orientations and sizes can capture a wider range of feature information, and is more sensitive to the targets with complex shapes, and can also be used with the MAFF module’s fusion mechanism, thus enhancing the model’s ability to represent image details and context.

**Effectiveness of the UFR module:** The UFR module filters the information in the up-sampling stage through the gating mechanism, and in terms of the model effect, Table 6 and Table 7 demonstrates that the UFR can filter and fuse the fused features very well, so as to optimize the segmentation capability of the model in a stable manner.

**Conclusions**

This paper proposes a novel polyp segmentation network, AFCNet, based on convolutional attention and multi-scale feature adaptive fusion. In the feature extraction and enhancement phase, the MDCA module captures wider image contextual information while increasing the number of important information in the image. A more efficient network structure is achieved by simplifying the deepest layer of features in the backbone network. In the feature fusion stage, the MAFF module fuses the features of different layers, and by dynamically balancing different feature fusion methods, it stably improves the model’s ability to capture global and detailed information and obtains better multi-scale feature fusion effects. In the up-sampling stage, the UFR module screens and guides the fused features obtained at the end. In the experimental part, we compare and generalize the module with 11 more advanced techniques in the field of polyp segmentation and combine the module with different backbone networks to conduct experiments on the generalization of the module. The results show that our method achieves the best performance and also has good generalization and adaptability. Finally, we perform rich ablation experiments on each module, thus verifying the validity and rationality of our AFCNet model.

**Data availability**

Te datasets utilized in this research study, such as the Kvasir-SEG and the CVC-ClinicDB dataset can be individually found https://datasets.simula.no/kvasir-seg/ and https://polyp.grand-challenge.org/CVCClinicDB/, respectively. Te overall training and test datasets can be found https://github.com/DengPingFan/PraNet.

**References**


**Author contributions**

Bojiao Jin developed the model architecture, all the custom innovative modules, performed the data preprocessing, data augmentation, data analysis and training/testing the model, implemented all the experiments and wrote the manuscript. Yi Zhang: reviewing, supervision and editing. Lin Qi: reviewing, supervision and editing. Wei Qian: supervision, writing—review and editing.

**Competing interests**

The authors declare no competing interests.