Image Classification and Segmentation based on Enhanced CNN and Transformer Networks

Krushi Patel

Submitted to the graduate degree program in Department of People who read Abstracts and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

,

,

,

Committee members

,

,

,

,

Date defended:

The Dissertation Committee for Krushi Patel certifies
that this is the approved version of the following dissertation :

Image Classification and Segmentation based on Enhanced CNN and Transformer Networks

,

Date approved:

Acknowledgment

I would like to take this opportunity to express my heartfelt gratitude and appreciation to all the individuals who have supported me throughout my Ph.D. journey. In particular, I extend my deepest thanks to my advisors, Dr. Guanghui Wang and Dr.Fengjun Li. Their unwavering enthusiasm for academic research has been truly inspiring, and their invaluable guidance and insights have propelled me forward in my studies.

I am also immensely grateful to my committee members, Dr. Bo Luo, Dr. Prasad Kulkarni, Dr. Cuncong Zhong, and Dr. Xinmai Yang, for their valuable suggestions and dedicated time invested in shaping my research. I extend my appreciation to all the members of ITTC for their support, which has greatly facilitated both my technical endeavors and work-life balance.

Additionally, I would like to express my sincere gratitude to my colleagues and peers for their camaraderie, intellectual exchange, and encouragement during challenging times. Their friendship and support have made this academic journey more enriching and enjoyable

Lastly, I extend my deepest appreciation to my husband and parents for their unwavering love, understanding, and encouragement throughout my academic pursuits. Their sacrifices, patience, and belief in my abilities have been a constant source of strength and motivation.

Abstract

Convolutional Neural Networks (CNNs) have significantly improved the performance on various computer vision tasks such as image recognition and segmentation based on their rich representation power. To enhance the performance of CNN, a self-attention module is embedded after each layer in the network. Recently proposed Transformer-based models achieve outstanding performance by employing a multi-head self-attention module as the main building block. However, several challenges still need to be addressed, such as (1) focusing only on class-specified limited channels in CNN; (2) limited respective field in the local transformer; and (3) addition of redundant features and lack of multi-scale features in U-Net type segmentation architecture.

In our work, we propose new strategies to address these issues. First, we propose a novel channel-based self-attention module to diversify the focus more on the discriminative and significant channels, and the module can be embedded at the end of any backbone network for image classification. Second, to limit the noise added by the shallow layers of an encoder in U-Net type architecture, we replaced the skip connections with the Adaptive Global Context Module (AGCM). In addition, we introduced the Semantic Feature Enhancement Module (SFEM) for multi-scale feature enhancement in polyp segmentation. Third, we propose a Multi-scaled Overlapped Attention (MOA) mechanism in the local transformer-based network for image classification to establish the long-range dependencies and initiate the neighborhood window communication.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deep neural networks have achieved tremendous performance improvement in computer vision-related tasks such as image classification and segmentation. In the last few years, it has evolved a lot from convolution networks and self-attention mechanisms to recently proposed vision transformer-based networks. This chapter will briefly elaborate on the CNN, self-attention mechanism, and transformer-based networks. In addition, we will discuss the limitations of these networks along with the proposed possible solutions.

1.1 Background and Related Work

Since the success of AlexNet(Krizhevsky et al., 2012), various research works have been proposed to improve the performance of various visions by increasing the depth, width, or skip connection between the layers. Though the CNN-based network achieved excellent performance, it has a few limitations, such as a limited respective field. The self-attention mechanism(Hu et al., 2018b)(Woo et al., 2018)(Park et al., 2018) is introduced to resolve this issue, which can be embedded in CNN and has significantly improved the performance on various vision tasks. Recently, Transformer based deep networks(Wang et al., 2021)(Dosovitskiy et al., 2020)(Liu et al., 2021) have been proposed, which use a self-attention mechanism as the main building block and has achieved comparable results as CNN. This section will introduce the primary mechanism of the deep convolution network, self-attention, and vision transformer.

1.1.1 Deep Convolution Neural Networks

A deep convolution neural network is the fundamental structure in most computer vision tasks, including object detection(Ren et al., 2015)(Girshick, 2015)(Redmon et al., 2016)(Lin et al., 2017b), image recognition(He et al., 2016)(Szegedy et al., 2015)(Howard et al., 2017)(Zagoruyko & Komodakis, 2016), and image segmentation (He et al., 2017)(Ronneberger et al., 2015)(Long et al., 2015). The basic architecture of the convolution neural network is shown in the figure. It consists of a series of convolution layers, pooling layers, and batch-normalization layers. As shown in the figure, after the specified number of convolution layers, pooling layers are applied to reduce the overall resolution of the image, and the number of channels is increased from the subsequent convolution layers. In the end, fully connected layers are embedded along with the SoftMax layer to produce the probability score for each class. The following section gives the basic information of each element in CNN.

1. **Convolution Layer**

   The convolution layer is the main building block of the convolutional neural network. It consists of a set of learnable kernels. Each kernel is a small slice extended to the full depth of the input volume. In the convolution layer, each kernel passes through the whole image and computes the dot product between the kernel element and corresponding elements of the image at any position. The output of the dot product produces a $2D$ activation map that gives the response of that kernel at every spatial position, as shown in the figure.

2. **Pooling Layer**

   The pooling layer is inserted between successive convolution layers to reduce the resolution of the input feature map. In the pooling layer, the specific size of the slice is passed through the image independently in all channels and resized spatially using either max or average operation, depending upon the type of pooling.

3. **Activation Layer**

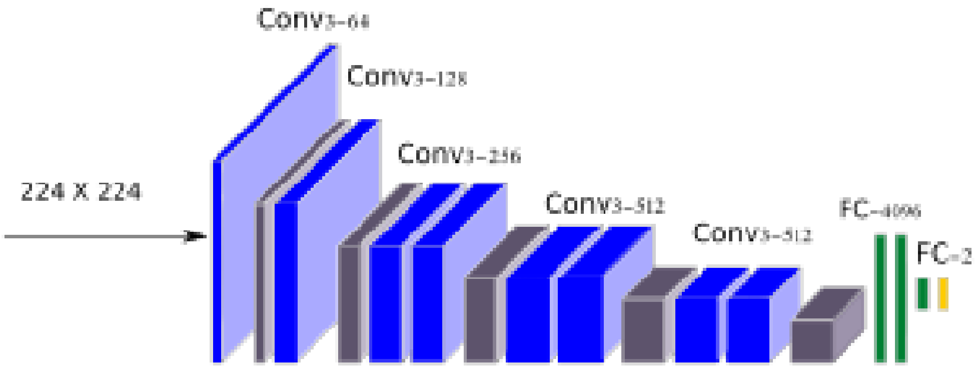


Figure 1.1: Convolution Networks: Blue block represents the convolution layer, black block represents the pooling layer and green block represents the fully connected layer

The activation layer adds non-linearity to the convolution neural network, which is essential for capturing complex features. Various types of activation functions are described below.

(a) **Sigmoid:** The mathematical formulation of sigmoid is:

$$\sigma(x) = \frac{1}{(1+e^{-x})} \tag{1.1}$$

The range of the sigmoid output is between 0 and 1. The main disadvantage of the sigmoid function is that it tends to saturate and kill gradients, and the outputs are not zero-centered.

(b) **Tanh:** The tanh activation function is the scaled version of sigmoid non-linearity. It maps the value to the zero centered, in between the range of -1 to 1 but still did not solve the gradient saturation problem. The mathematical formulation of tanh is:

$$tanh(x) = 2\sigma(2x) - 1 \tag{1.2}$$

(c) **ReLU:** The Relu non-linearity solves the problem of saturated gradients. The mathematical formulation is:

$$ReLU(x) = max(0,x) \tag{1.3}$$

The main disadvantage of the ReLU activation is dying Relu, which causes the output of the activation value to zero.

(d) **Leaky ReLU:** Leaky ReLU solves the problem of dying ReLU by including the slight slope in the negative range, as shown in the figure. This makes the function generate small negative outputs when the input is less than 0. The mathematical formulation of the Leaky ReLU is:

$$LeakyReLU(x) = (x < 0)(ax) + (x >= 0)(x) \tag{1.4}$$

4. **Batch-Noramlization Layer**

   The batch normalization layer is included after the convolution and activation layers. Basically, it preprocesses the activation at every network layer and integrates it into the network itself differently.

5. **Fully Connected Layer**

   All neurons in the fully connected layers are connected to all the neurons of the previous layer. In fully connected layers, the linear transformation is applied to the input vector using matrix multiplication followed by a bias offset.

1.1.2 Self-Attention Mechanism

The convolution network has a limited respective field with respect to the size of the kernel. To establish the long-range dependencies in the convolution network, the self-attention module(Hu et al., 2018b)(Woo et al., 2018)(Park et al., 2018)(Vaswani et al., 2017) is usually embedded after specific convolution layers. The attention module improves the representation of interest by focusing on essential features and suppressing unnecessary ones. Usually, the attention module is divided into two categories based on which dimension it is applied : (1) Spatial Attention and (2) Channel Attention. The following information provides the mathematical formulation of the attention mechanism irrespective of the dimension on which it is applied. Attention mechanisms can be grouped into two parts.

1. **Transformation layer**

   It maps input sequences $X \in \mathbb{R}^{n_x \times d_x}$ and $Y \in \mathbb{R}^{n_y \times d_y}$ to three different sequential vectors Query(Q), Key(K) and Value(V), where n and d are the length and dimensions of input sequences respectively. Each vector is generated as:

   $$Q = XW^Q, K = YW^K, V = YW^V \tag{1.5}$$

   where, $W^Q \in \mathbb{R}^{d_x \times d_k}$ and $W^K, \in \mathbb{R}^{d_y \times d_k}$ and $W^V, \in \mathbb{R}^{d_y \times d_v}$. d_k is the dimension of query and key, and d_v is the dimension of value. The query is projected through X and the key and value are projected through Y. For the self-attention mechanism, $X = Y$

2. **Attention layer**

   An attention layer explicitly aggregates the query with the corresponding key, multiplying the result with the value and producing the final output vector as described by the following equation.

   $$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1.6}$$

   As per the above equation, the attention weights are generated using a dot-product of the query and the key and scaling it by factor $\sqrt{d_k}$ and normalizing it by softmax operation. The resultant attention weights are assigned to the corresponding element of the value.

1.1.3 Vision Transformer

Inspired by the great success of transformers in the NLP field and after the demonstrated effectiveness and efficiency of the self-attention module in CNN, vision transformers(Wang et al., 2021)(Dosovitskiy et al., 2020)(Liu et al., 2021) have now become very popular and achieved

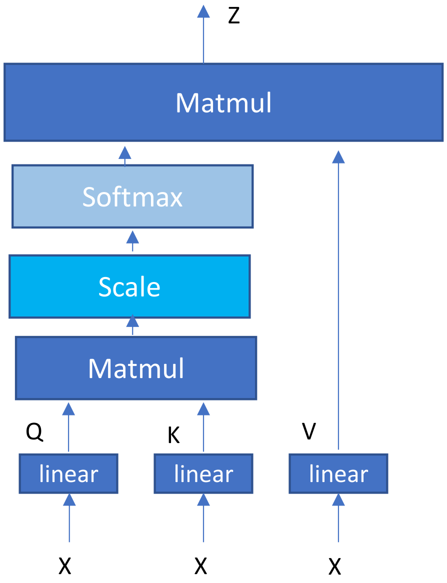


Figure 1.2: Attention block: it takes x as an input, generates the three vectors Q, K, and V on which dot product is calculated to get the attention map

comparable results. The main building block of the transformer network is the multi-head attention module(Vaswani et al., 2017) which has been explained below, followed by the fundamental architecture of the vision transformer.

1. **Multi-head Attention**:

   As the modeling capability of single head attention is coarse, the multi-head self-attention mechanism linearly projects the input into multiple feature subspaces and processes them by several independent attention heads parallelly. The resultant vectors are then concatenated and mapped to the final output. The mathematical formulation is as follows:

$$Q_j = XW^{Q_j}, K_j = YW^{K_j}, V_j = YW^{V_j} \tag{1.7}$$

$$Z_j = Attention(Q_j, K_j, V_j), j = 1, 2, ...h, \tag{1.8}$$

$$Multihead(Q, K, V) = Concat(Z_1, Z_2, ...Z_h)W^o \tag{1.9}$$

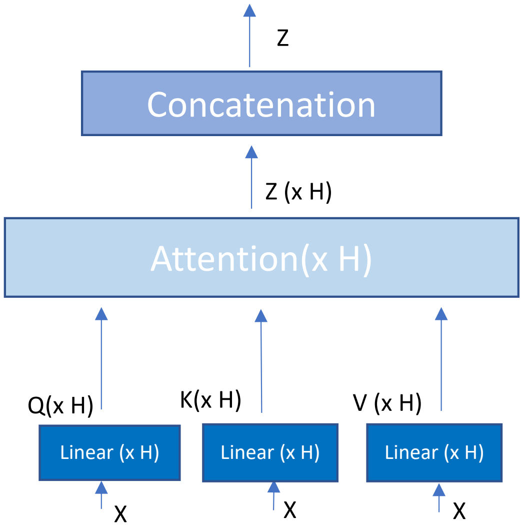


Figure 1.3: Multihead Attention Block: it performs self-attention in different subspaces, and the output is concatenated.

where h is the total number of heads, $W^o \in \mathbb{R}^{hd_v \times d_{model}}$ is output matrix, Z_i denotes the attention weights for i^th head, $W^{Q_i} \in \mathbb{R}^{d_{model} \times d_k}$, $W^K \in \mathbb{R}^{d_{model} \times d_k}$ and $W^V \in \mathbb{R}^{d_{model} \times d_v}$ are the projection matrix for query, key and value respectively.

The multi-head attention separates the input into the h independent attention heads with d_{model}/h dimensional vectors and integrates the features of each head parallelly.

As the transformer takes sequential input, the image is first divided into the sequence of patches and fed to the series of transformer layers(Dosovitskiy et al., 2020). Specifically, the input image of size $x \in \mathbb{R}^{H \times W \times C}$ is reshaped to the sequence of flattened 2D patches $x_p \in \mathbb{R}^{N \times (P^2.C)}$, where the H, W and C denotes the width, height and channel of the image, (P,P) is the size of the patch and $N = HW/P^2$ is the total number of patches, which consider as an input sequence length in the transformer.

The vision transformer consists of alternating layers of multi-head attention and MLP blocks. The MLP consists of two layers with GELU non-linearity. LayerNorm is applied before every block, along with residual connection. The overall mathematical formulation of the vision transformer is as below:

$$z_0 = [x_{cls}; x_p^1 E; x_p^2 E; .....; x_p^N E;] + E_{pos} \tag{1.10}$$

$$z_l' = MHSA(LN(z_{l-1})) + z_{l-1}, l = 1, 2, ..L \tag{1.11}$$

$$z_l = MLP(LN(z_l')) + z_l', \tag{1.12}$$

$$y = LN(z_L^0) \tag{1.13}$$

where, $E \in \mathbb{R}^{(P^2.C)\times D}$, $x_c ls$ token is the learnable embedding to the sequence of the embedded patches($z_0^0 = x_c ls$), whose state at the output(z_L^0) serves as the image representation(y) and $E_{pos} \in \mathbb{R}^{(N+1)\times D}$ is a $1D$ position embeddings are added to the patch embedding to retain the position information of the patches and served as a input to the transformer.

1.1.4 Image Classification

Image classification is the task of assigning a label to a given image from a given category. It is the core problem in computer vision and is used in many complex applications as a base network. There have been various deep network-based image classification models proposed(He et al., 2016)(Szegedy et al., 2015)(Howard et al., 2017)(Zagoruyko & Komodakis, 2016) until now. Depending upon the main component of the architecture, the classification model can be divided into the following categories: Convolutional networks, self-attention augmented convolution network, and Transformers. We briefly discuss a few architectures of the above-mentioned categories below.

1. **Convolution network**

   Various research works have been proposed to improve the performance by increasing the depth or width, adding a skip connection, or using different versions of the convolution layer. The main classification models are discussed briefly below.

   ResNet. ResNet(He et al., 2016) is the first deep network with more than 100 layers. To address the problem of vanishing gradients in the deep network, the idea of residual blocks, with the skip connection to fit the input from the previous layer to the next layer without modifying it. For each residual block, they used a stack of three convolution blocks, which include 1×1 convolution back and forth of 3×3 layer. Here 1×1 convolution is responsible for adjusting the number of channels. ResNet with 152 layers achieved state-of-the-art results on the ImageNet dataset with only 19.38% and 4.49% top-5 error.

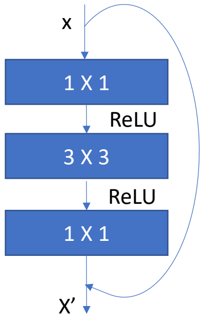


Figure 1.4: ResNet Block: it takes x as an input and produce the x' by adding the previous layer output

Wide-ResNet. Wide-ResNet(Zagoruyko & Komodakis, 2016) uses the same residual block used in the ResNet. In Wide-ResNet, it is shown that with a reduced number of layers and with increasing the number of channels in Resnet, performance still can be comparable with the original Resnet with deeper layers and make the network easy to train.

DenseNet. It is observed that convolution networks are efficient to train if they contain shorter connections between the layers close to the input and layers close to the output. DenseNet(Huang et al., 2017) is made up of several dense blocks, in which the feature map from all previous layers is used as an input, and the current feature map is used as input to all subsequent layers. DenseNet uses a concatenation operation to add the features from previous layers instead of using elementwise addition. In addition, the DenseNet has fewer filters, making the network thinner and more compact. It is easy to train because it improves the information flow and gradient throughout the network. DenseNet improves the performance on various image classification datasets compared to Resnet and Wide-ResNet.

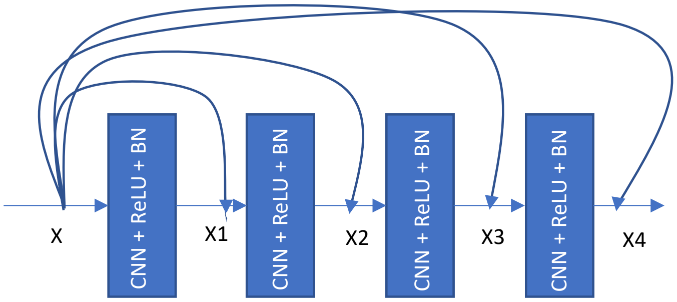


Figure 1.5: DenseNet Block: it takes all the previous layers output as well as current input is added to all the next layers

Inception. The inception(Szegedy et al., 2015) architecture was proposed to solve the problem of overfitting in the deep model and a large number of parameters. It is a sparsely connected architecture with a total of 27 inception layers. Each inception layer consists of various sizes of convolution layers(1×1,3×3,5×5) along with a pooling layer. Each layer's output is concatenated and applied as an input to the next layer. In addition, the 1×1 convolution layer is added before and above the convolution layers to reduce the number of channels. Two auxiliary losses are applied along with the main loss to prevent the vanishing gradient problem in the middle layers. The total loss is a weighted sum of the auxiliary and main losses.

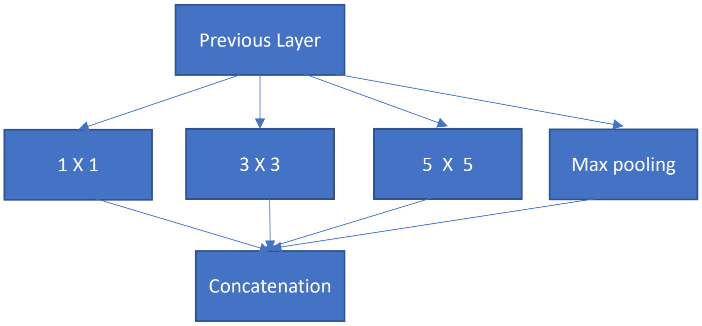


Figure 1.6: Inception Block: it consists of parallel convolution layers of different sizes and the output is concatenated

MobileNet. The MobileNet(Howard et al., 2017) is a lightweight architecture with a significantly low number of parameters. To reduce the number of parameters, depthwise separable convolution is introduced. The depthwise convolution consists of two operations: (1) Depthwise convolution and (2) Pointwise convolution. Depthwise convolution is the spatial convolution, applied independently on each channel. The pointwise convolution is applied after the depthwise convolution to reduce the number of channels. In short, MobileNet splits the standard convolution into a 3×3 depthwise convolution, which significantly reduces the computation cost and the number of parameters.

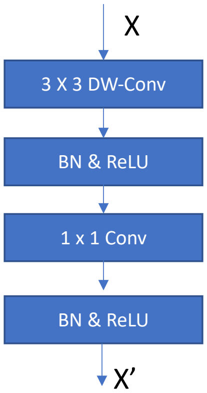


Figure 1.7: MobileNet Block consists of a depthwise separable convolution layer and a pointwise convolution layer.

2. **Self-attention augmented convolution network**

Various spatial and channel self-attention mechanisms(Hu et al., 2018b)(Woo et al., 2018)(Park et al., 2018)(Vaswani et al., 2017)(Wang et al., 2018) are introduced to give attention to essential pixels and suppress the less essential features. We discuss a few of them below.

SE-Net. (Hu et al., 2018b) proposed a channel attention mechanism called Squeez and Excitation(SE) block, which is embedded after each stage in the CNN. SE block employs global information to emphasize informative features and selectively suppress the less useful ones. It is made up of two consecutive operations: Squeeze and Excitation. The squeeze operation consists of a global average pooling to generate channel-wise statistics, an n-dimensional feature vector that passes through two fully connected layers and generates the same length vector used to weigh the original feature maps.

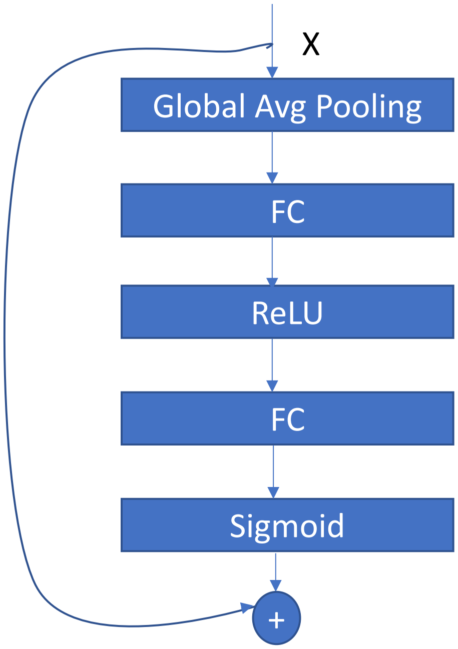


Figure 1.8: SE Block: it consists of the global average pooling layer followed by two fully connected layer

CBAM. Convolution Block Attention Module(CBAM)(Woo et al., 2018) is a channel and spatial attention block series. Channel attention uses both max pooling and average pooling for the global context information, followed by two fully connected layers. The spatial attention layer first concatenates the feature map generated by max and global average pooling in the channel attention layer and passes through a small convolution block of 7×7 kernel size, followed by sigmoid activation.

Non-Local Block. The non-local block (Wang et al., 2018) computes the relation between all pixels by matrix multiplication. It first takes the feature map as an input and passes through three different linear transformations (ϕ, θ, and) using 1×1 convolution. Then, the output generated by ϕ and θ is reshaped and multiplied, followed by the softmax operation,

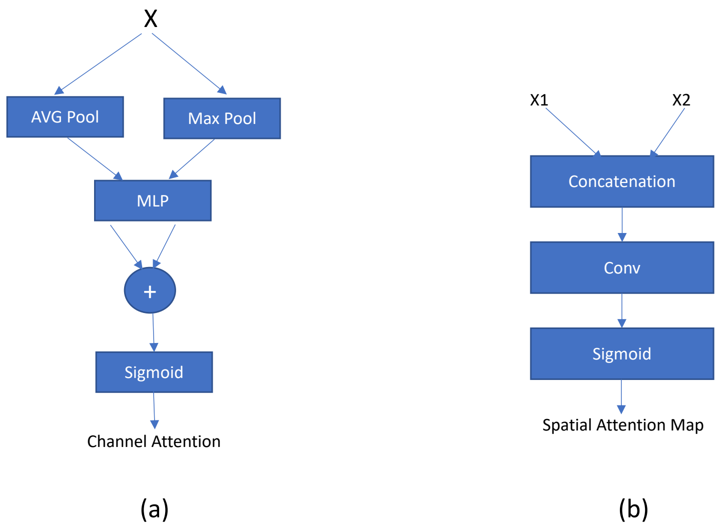


Figure 1.9: CBAM: It consists of (a) channel attention followed by (b) spatial attention.

which establishes the relationship between one pixel to all. The result of softmax is further multiplied by the output of the transformation, followed by the 1×1 convolution, which generates the attention map. The block diagram of the non-local block is shown in the figure.

3. **Transformer network**

After the success of the attention mechanism, researchers have tried to employ the self-attention mechanism as the primary building block of a deep network, recognized as a transformer network. The recently proposed three main transformer networks are explained below.

Vision-Transformer(ViT). (Dosovitskiy et al., 2020)propose the first vision transformer for the image classification. As the transformer expects the sequential input, the input image is first split into a series of non-overlapped patches, and each patch is linearly transformed using patch embedding. In addition, 1-dimensional learnable position embeddings are also added into patch embedding to retain the position information. The patch embeddings are fed into the transformer encoder, which consists of a series of transformer layers. Each trans-

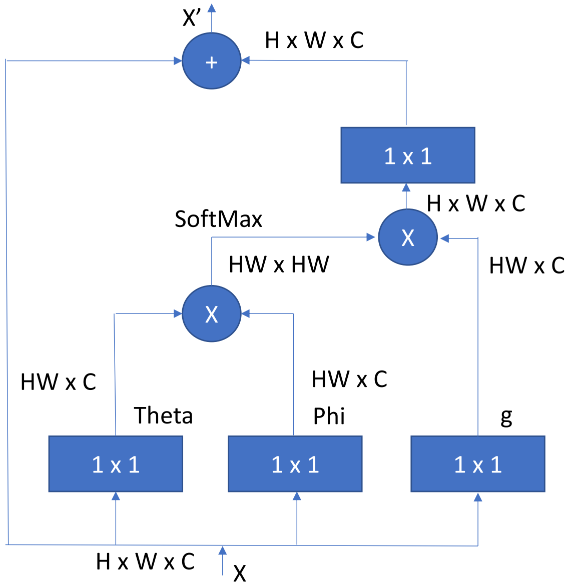


Figure 1.10: Non-local Block: it takes the input and transforms it into a different space. It then calculates the dot product between two vectors from different spaces and assigns the result by doing the dot product to the remaining vector space, which produces the attention map

former layer consists of the multi-head attention layer and MLP. In addition, class embedding is also inserted along with patch embedding as an input. The final state of the class embedding represents the classification output. ViT is first trained on a huge dataset such as JFT-300M, and then the pre-trained weight is used for medium(ImageNet) and small(CIFAR-10/100) datasets. Vit achieves comparable or improved results than the CNN networks.

Pyramid Vision Transformer(PVT). Pyramid Vision Transformer (Wang et al., 2021) divides the whole architecture into four stages that generate hierarchical feature maps of different scales. Patch embedding is calculated after each stage to increase the number of channels and decrease the resolution of the feature map. Same as ViT, it takes the series of patches as an input. After each stage, the output is reshaped to a 2D feature map. In addition, PVT uses a slightly modified version of the multi-head attention called Spatial Reduced Attention(SRA). In SRA, the spatial reduction is applied to key and value embedding, and then the regular multi-head attention is calculated. PVT outperforms the recent vision model based on the transformer.

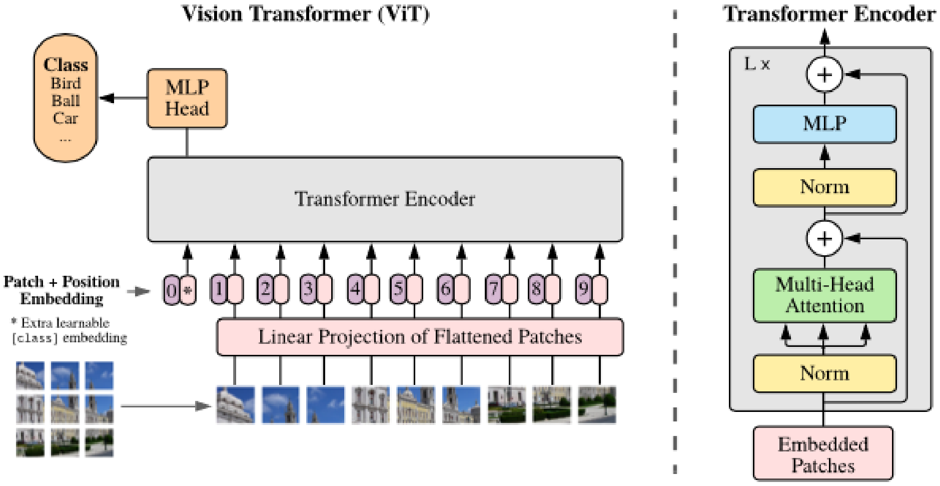


Figure 1.11: Vision transformer: it takes the patches as an input and passes through the encoder, which produces a classification result (Dosovitskiy et al., 2020)

Swin Transformer. Swin Transformer(Liu et al., 2021) is one of the local transformers proposed recently, which applies multi-head attention within the patch only. Swin transformer achieves state-of-the-art accuracy in image classification using a transformer. Same as PVT, the Swin Transformer generates the hierarchical feature maps of various scales after each stage, and unlike PVT and ViT, the Swin Transformer applies the multi-head attention within the window(patch) only. In addition, the windows(patches) are shifted in the successive multi-head attention layer, and attention is applied within those windows. This shifted window-based approach initiates neighborhood pixel communication, leading to significant performance gain

1.1.5 Polyp Segmentation

Polyp segmentation is the method to classify each pixel into two categories: polyp and background in the image. Colonoscopy is the procedure to detect the polyp, which is the primary reason for developing colorectal cancer. Specific to polyp segmentation, it is a task to categorize each pixel in a colonoscopic image in either polyp or background. Polyp segmentation is a challenging task because of the diverse shape of polyps, size, and texture of the polyp. Usually, for the segmentation encoder-decoder type, U-shape architecture(Ronneberger et al., 2015)(Zhou et al., 2018)(Zhang et al., 2020a) is used. The encoder consists of a series of convolution, pooling, and activation layers that gradually decrease the feature map resolution. The decoder consists of the convolution and upsampling operation, which gradually increases the resolution to the original size of the image. The basic U-Net type architecture and its variant are explained briefly below.

1. **U-Net.** U-Net is the U-shape (Ronneberger et al., 2015) architecture, which consists of the encoder and decoder. It takes the image as an input and produces the segmentation mask as the output. The overall architecture is as shown in the figure. The encoder consists of a series of encoding layers. In each encoding layer, the 3×3 convolution layer, ReLU activation layer, and batch-normalization layers are stacked together along with the 2×2 max-pooling operation to reduce the resolution of the feature map. After each max pooling layer, the number of channels is doubled. The decoder consists of a series of decoding layers. Each decoding layer has an upsampling layer followed by a convolution layer. In addition, the output of the corresponding encoding layer is concatenated with the decoding layer, followed by the 3×3 convolution ReLU activation layer. At the last layer, 1×1 convolution is used to map the channels to the desired number of classes.

2. **U-Net++.** In the U-Net(Zhou et al., 2018), corresponding encoder features are directly concatenated with the decoder layer, resulting in a large semantic gap. U-Net++ solved the semantic gap problem in U-Net, by introducing the nested and dense skip connection. In addition, U-Net++ implements deep supervision, which means auxiliary losses are applied at multiple stages instead of using only the main loss at the end. The architecture of U-Net++ is shown in the figure.

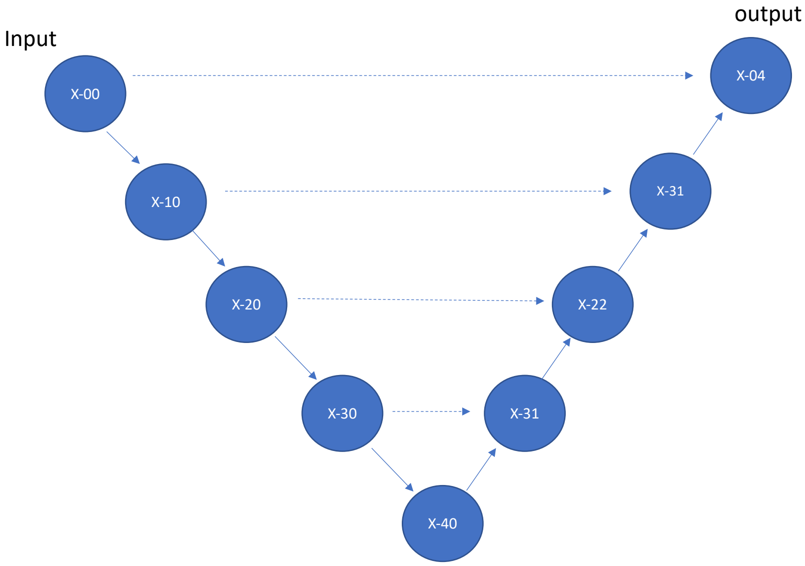


Figure 1.12: UNet: Each corresponding encoding layer is directly added to the decoding layer through skip connection

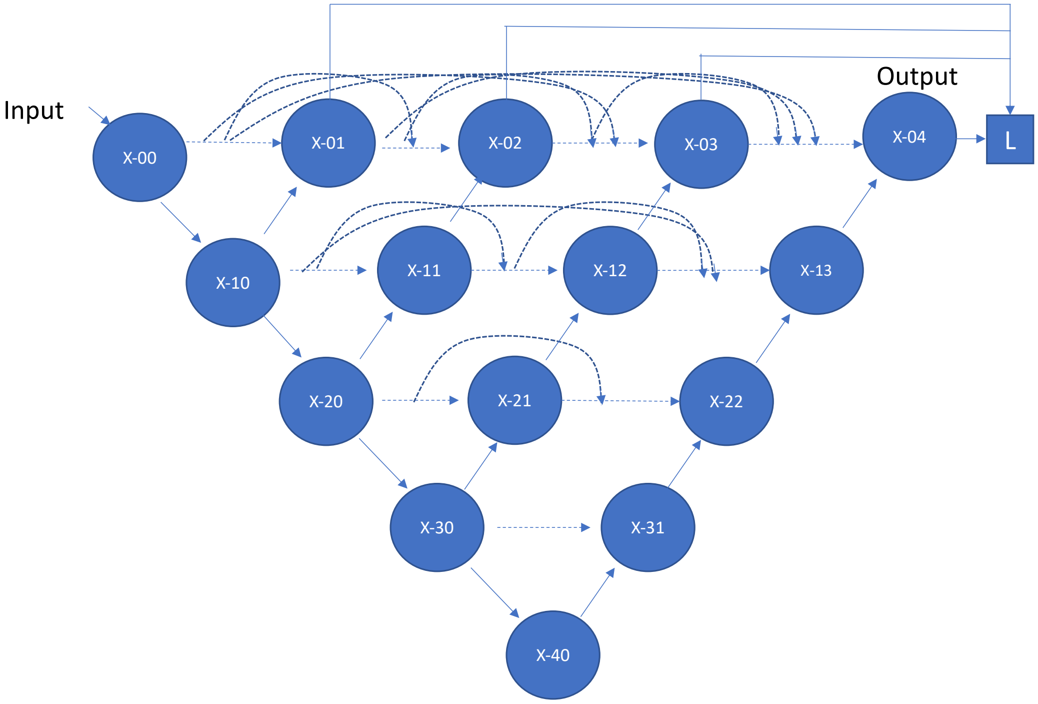


Figure 1.13: UNet++: Dense skip connection at each layer including deep supervision by applying auxiliary loss

1.2 Limitations

In this section, we will discuss the limitations associated with image classification and polyp segmentation.

1.2.1 Focusing only on class-specific dominant channels:

In general, each channel in the last layer of the CNN focuses on a specific part of the image. A natural tendency of the convolutional network is to focus only on a few class-specific dominant channels, which limits the set of clues to classify the image. For example, as shown in the figure, the basic model misclassified the lion image as a tiger. If you see the Grad CAM, it can be seen that it focuses on only the nose part whereas ignores the other parts, such as the ear or eyes, which can be significant to classifying the image correctly. It will be beneficial if it uses other parts of the image as well while taking the final prediction. We believe that if we divert the attention to other parts, the correctness of the classification can be improved.

1.2.2 Lack of global respective filed in the Local-Transformer

Local transformer-based image classification models have recently gained comparable performance with linear computation complexity. We believe there is still scope for improvement by solving the issues associated with local transformers, such as lack of long rånge dependencies and neighborhood window communication. The local transformer applies multi-head attention within the local patch. It lacks communication between all patches as well as neighborhood pixels. If we established the global attention between the local transformer layer, it could improve the performance by increasing a slight amount of computation cost.

1.2.3 Semantic gap in skip connection and lack of multi-scale features

Encoder-decoder type U-shape architecture is the standard network for polyp segmentation. Each encoding layer is directly connected to the corresponding decoding layer. However, a direct skip connection can add redundant and noisy features, especially in the shallow layers due to the semantic gap. In addition, polyp size varies according to the focus of the camera. Therefore, a multi-scale feature enhancement module needs to handle the diverse sizes of polyps in the images

Chapter 2

A Discriminative Channel Diversification Network for Image Classification

Abstract

Channel attention mechanisms in convolutional neural networks have been proven to be effective in various computer vision tasks. However, the performance improvement comes with additional model complexity and computation cost. In this paper, we propose a light-weight and effective attention module, called channel diversification block, to enhance the global context by establishing the channel relationship at the global level. Unlike other channel attention mechanisms, the proposed module focuses on the most discriminative features by giving more attention to the spatially distinguishable channels while taking account of the channel activation. Different from other attention models that plugin the module in between several intermediate layers, the proposed module is embedded at the end of the backbone networks, making it easy to implement. Extensive experiments on CIFAR-10, SVHN, and Tiny-ImageNet datasets demonstrate that the proposed module improves the performance of the baseline networks by a margin of 3% on average.

2.1 Introduction

Deep convolutional neural networks (CNN) have become a dominant approach to solve a wide range of computer vision tasks, including image classification (Cen et al., 2021), object detection (Ma et al., 2020), semantic segmentation (He et al., 2021), recognition (Sajid et al., 2021b), image translation (Xu et al., 2019). Inspired by the tremendous success of AlexNet (Krizhevsky et al., 2012) in image classification, many researchers have developed different network structures to boost the performance of deep CNN (Patel et al., 2021). In recent years, the superior performance obtained by squeeze and excitation networks (Hu et al., 2018b) has attracted many researchers to incorporate a channel attention mechanism in convolutional neural networks.

Previous channel attention networks (Wang et al., 2018; Cao et al., 2019; Gao et al., 2020; Fu et al., 2019) utilize the collection of global information by calculating pair-wise relation between channels but overlook the significance of single channel information. We believe that the effect of the single channel's overall activation on global pairwise relation could exploit global information of channels well, without adding more extra parameters, unlike ABN (Fukui et al., 2019).

In general, each channel in the feature map focuses on a specific part of the image. A natural tendency of the convolutional neural network is to focus only on a few class-specified dominant channels, which limits the set of cues to classify the image. To alleviate this problem, we propose a novel channel attention mechanism, called channel diversification module, to force the network to learn more diverse and significant features by exploiting both the given channel's overall activation and pair-wise channel relationship as illustrated in Figure 6.1. It shows that the significance of each channel is calculated by the global average pooling and to focus more on diverse features, global attention pooling is used, which are fused together using concatenation.

The proposed module can be considered as a combination of the simplified non-local block (Cao et al., 2019) and SE-block (Hu et al., 2018b). It concatenates the features generated by the global context modeling from both networks and performs a transformation without using MLP, unlike SE-block. Figure 6.2 shows the block diagrams for all three networks to illustrate the difference among them. Our module adopts the part which is highlighted using the dotted box in Figure 6.2 (a) and (b) and concatenated the features generated by those parts as shown in Figure 6.2 (c).

The channel diversification module explores the input feature map from the last convolutional layer of the backbone network as the input and produces a C-dimensional feature vector using

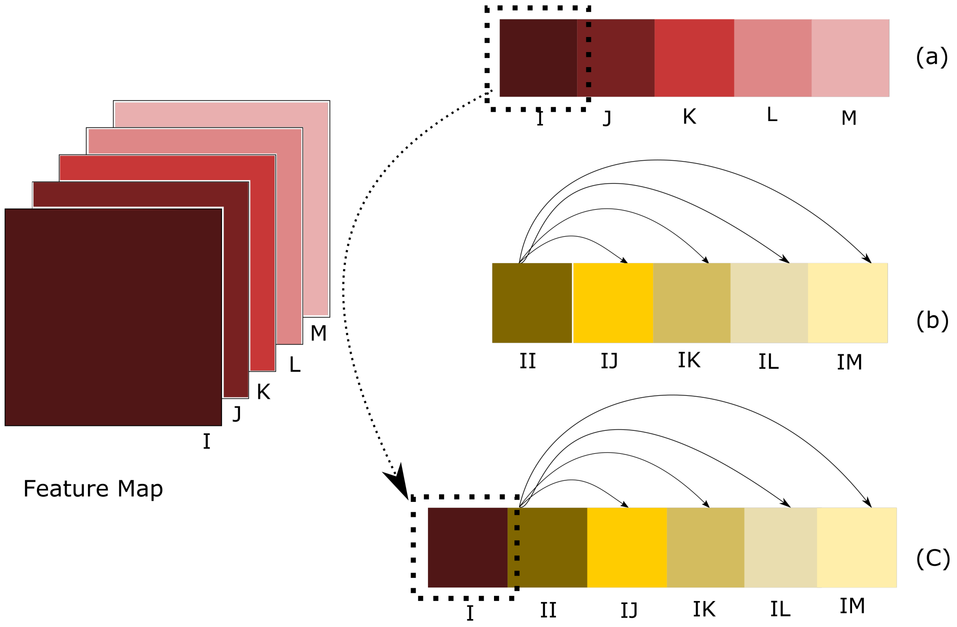
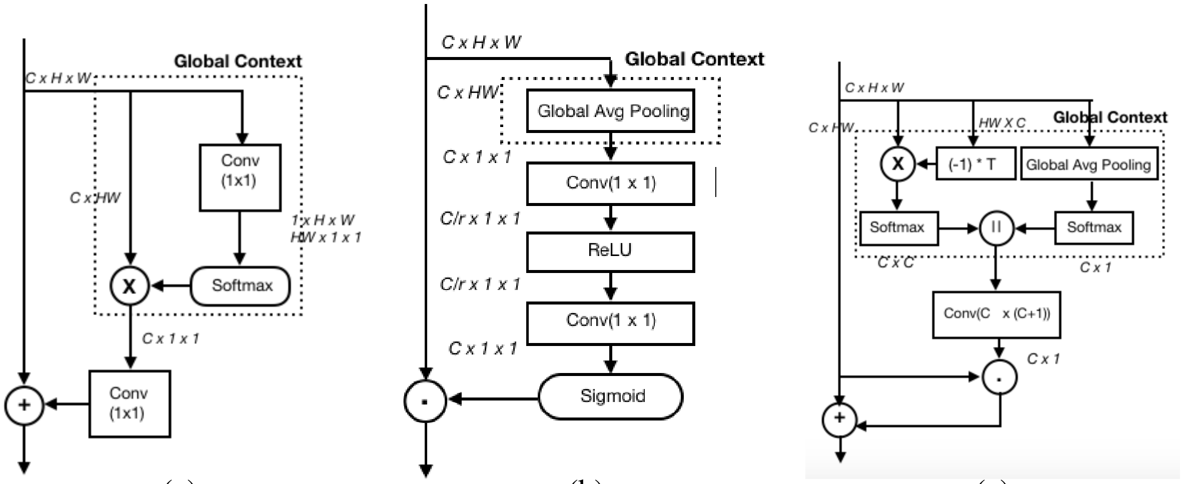


Figure 2.1: Global relationship operation: (a) Features generated by global average pooling that demonstrates the overall significance of each channel; (b) features generated using the modified global attention pooling which represents how each channel is dissimilar with the given channel. Here we only display the relationship of one channel for the illustration purpose; (c) fusing of the features generated by (a) and (b) using concatenation operation.

global average pooling, which represents the significance of each channel for a specific class. At the same time, it also generates a channel relationship matrix, which specifies how the given channel is distinct from other channels. After that, both the channel relation matrix and feature vector are stacked together, and convolution is applied to generate the weighted score for each channel.

The proposed channel diversification block focuses on spatially distinguished channels while considering the significance of each channel. It penalizes the most dominant channels and diverts the attention to different channels that are spatially distinguished from the given channel as well as have a large average activation. Therefore, our module only focuses on the channels that are diverse and significant enough at the same time. The new contributions of this work are summarized as follows:



(a) (b) (c)

Figure 2.2: Comparison of different attention based modules: (a) Simplified Non-local network; (b) SE block; and (c) channel diversification block. Where $C \times H \times W$ is the feature map dimension, "$+$" denotes the broadcast elements-wise addition, X denotes the matrix multiplication, "." represents the broadcast element wise multiplication, "$||$" represents the concatenation operation, and $(-1) * T$ represents the multiplication of matrix with its transpose and (-1).

- The paper proposes a novel channel diversification block that makes the convolutional neural networks to focus on significant and diverse channels by establishing the relationship between the local (single channel) and global information channels.
- The proposed channel diversification block can be easily plugged in before the output layer of any baseline network to improve the performance of the baseline network by adding only a few extra trainable parameters and GFLOPs.

Extensive experiments have been conducted on CIFAR-10 (Krizhevsky et al., a), CIFAR-100 (Krizhevsky et al., b), SVHN (Netzer et al., 2011), and the Tiny-ImageNet dataset with various baseline networks, including VGGNet (Simonyan & Zisserman, 2014), ResNet (He et al., 2016), Wide-ResNet (Zagoruyko & Komodakis, 2016), ResNext (Xie et al., 2017), and DenseNet (Huang et al., 2017). The results demonstrate that the proposed module outperforms the baseline models and achieves competing performance with the different attention-based models while adding less computation cost. The source code of the proposed module can be downloaded from the link https://github.com/rucv/ChannelDiversification.

There has been a rapid evolution in the field of image classification since the publication of Alexnet (Krizhevsky et al., 2012), which achieves a record-breaking image classification accuracy. After that, researchers have been focused more and more on deep learning-based approaches for image classification tasks (Li et al., 2021b). VGG-Net (Simonyan & Zisserman, 2014) and GoogleNet (Szegedy et al., 2015) introduced a block-based architecture and proved that a deeper model could significantly improve the classification accuracy. ResNet (He et al., 2016) proposed a skip connection based residual module to solve the vanishing gradient problem in deep models. ResNext (Xie et al., 2017) and Xception (Chollet, 2017) employed multi-branch architecture to increase the cardinality. Our proposed network employs the above-mentioned networks as a baseline and integrates the channel diversification block at the end.

The attention mechanism has proven to be very effective in various computer vision and natural language processing tasks. They have been widely used in sequential models. In computer vision, it started after SE-Net's large performance gain using a channel attention mechanism (Hu et al., 2018b). Inspired by this idea, the residual attention network (Wang et al., 2017) introduced bottom-up and top-down feed-forward structure in the attention mechanism, and CBAM (Woo et al., 2018) used both max and average pooling to aggregate the features as well as compute spatial attention using 2D convolution. Non-Local networks (Wang et al., 2018) compute spatial attention by taking into account features from all spatial positions. GC-Net (Cao et al., 2019) combines the non-local (Wang et al., 2018) and SE-block (Hu et al., 2018b) to produce a light-weight attention module. Our method utilizes the SE-block and non-local block in a new way.

Most previous attention mechanisms refine the intermediate features; we propose to apply the attention module at the end before the output layer like attention pooling (Girdhar & Ramanan, 2017), SOAL (Kim et al., 2020), ABN (Fukui et al., 2019), AG-CNN (Guan et al., 2020), AGNN (Zou et al., 2020) and channel interaction network (Gao et al., 2020). Attention pooling and SOAL computes spatial attention. The channel interaction network is similar to ours but it does not consider the single channel significance. ABN uses CAM (Zhou et al., 2016) generated attention weight to focus on a specific region of the image. AGNN used both self-attention and CAM-

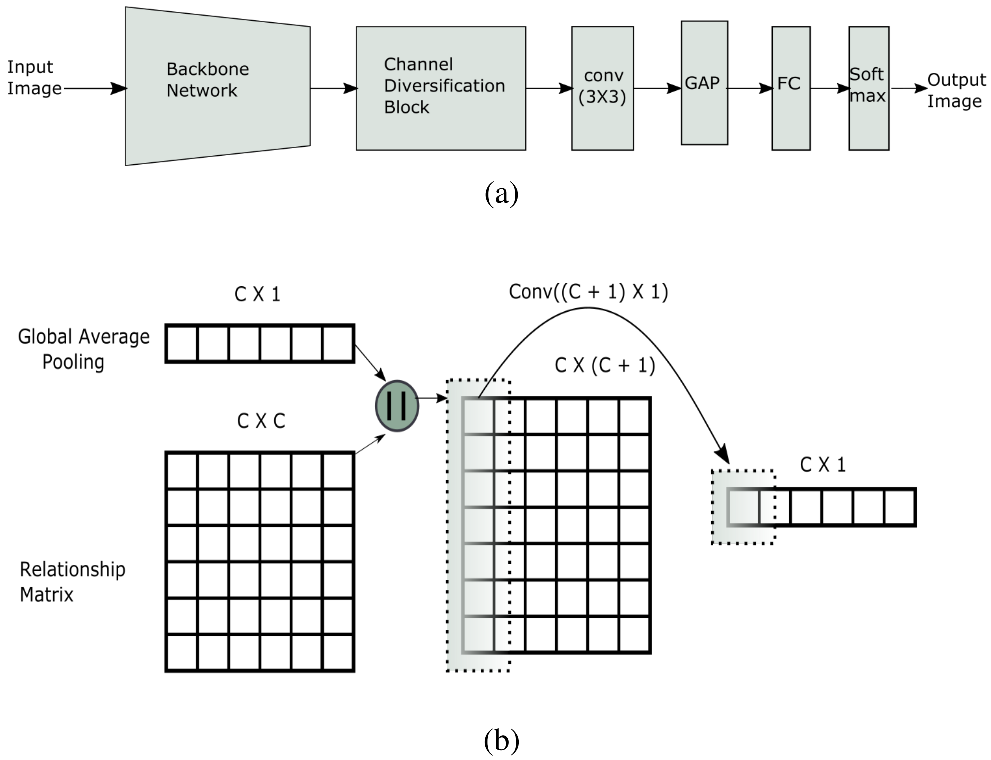


Figure 2.3: (a) The overall architecture of the classification network with the channel diversification network. (b) The architecture of the channel diversification module. The features generated by the global average pooling and the modified global attention pooling are added elementwisely and multiplied by the original feature map. The dimension of the features is displayed on the top of each feature.

based attention and combined attentive features with LSTM. AG-CNN introduced threshold-based attention to generate a binary mask, which is further used to crop a global image to extract a significant local region.

2.2 Proposed Method

In this section, we describe the proposed channel diversification block in detail. Our proposed block is a combination of global features generated by a simplified non-local block and SE-block. We first re-visit both blocks and represent which feature operation we adopted from them one by one, followed by a detailed explanation of the channel diversification network.

2.2.1 Revisit Simplified Non-Local Block:

Simplified non-local block enhances the features of a given position by aggregating feature information of other remaining positions. It can be formulated as(Cao et al., 2019):

$$z_i = x_i + W_v(\sum_{j=1}^{N_p} \frac{e^{W_k x_j}}{\sum_{m=1}^{N_p} e^{W_k x_m}} x_j) \tag{2.1}$$

Where x is the one instance of the feature map, $N_p = W \times H$, where W and H is the width and height, and W_v and W_k is the linear transformation. From the above equation, we adopted only global attention pooling, and instead of calculating spatial relationship, we calculate the channel-wise relation(Gao et al., 2020)(Fu et al., 2019), which can be formulated as:

$$\alpha_{ij} = \frac{e^{-x_i x_j}}{\sum_{m=1}^{C} e^{-x_i x_m}} \tag{2.2}$$

Where C stands for the number of channels. α_{ij} represents the relationship of the i^{th} channel to j^{th} channel. It gives more weights to dissimilar channels and smaller weights to the most correlated channels.

2.2.2 Revisit Squeeze and Excitation Block:

The Squeeze and Excitation block extracts the global features and applies transformation using MLP, which selectively emphasizes informative features and suppresses less useful ones. It consists of two parts: (1) global average pooling for global context modeling and (2) series of 1×1 convolution layer followed by non-linear transformation. SE-Net can be formulated as (Hu et al., 2018b):

$$Z = \sigma(W_2 \delta(W_1 \alpha)) \tag{2.3}$$

Where α represents the global average pooling. From the SE block, we adopt the global average pooling part, which can be formulated as below.

$$\alpha_i = \frac{1}{H \times W} \sum_{k=1,l=1}^{W,H} X_{k,l} \tag{2.4}$$

Where W and H are the width and height of the feature maps, respectively. α_i represents the average activation of the i^{th} channel, which indicates the importance of each channel.

2.2.3 Channel Diversification Network

In the channel diversification block, we fuse Eq. eq:ch and Eq. eq:avg adopted from the simplified non-local block and the squeeze and excitation block, respectively. Features generated by both equations are α_{ij}, and α_i represents the relationship between one channel to another and the overall importance of the channel, respectively, which are fused using concatenation operation as described in Figure 6.1.

Our channel diversification block takes $X \in \mathbb{R}^{C \times H \times W}$ feature map and apply global average pooling, which generates C dimensional feature vector, $A \in \mathbb{R}^{C \times 1}$

$$A_c = \frac{1}{H \times W} \sum_{k=1,l=1}^{W,H} X_{k,l} \tag{2.5}$$

Softmax normalization $A = Softmax(A)$ is applied to the output of the global average pooling of all channels of the feature map, which represents the significance of each channel. At the same time, the feature map is applied to channel-wise simplified non-local block as formulated in Eq. eq:ch and produces a channel relation matrix $J \in \mathbb{R}^{C \times C}$.

$$J = -X.X^T \tag{2.6}$$

After that, channel-wise softmax normalization is applied to the produced channel relation $J = Softmax(J)$, which indicates how dissimilar the given channel is to other remaining channels.

Both the normalized features generated by the global average pooling and channel relationship matrix are then concatenated to yield the feature $Y \in \mathbb{R}^{C \times C+1}$

$$Y = A \parallel J \tag{2.7}$$

We then apply a 2D transformation to this concatenated output using the convolution of size $1 \times (C+1)$ to enhance the global context by establishing the relationship between the single channel activation and global channel relationship and produce the weighted feature vector $Y \in \mathbb{R}^{C \times 1}$

$$Y = f(Y) \tag{2.8}$$

and the resultant attention vector is then multiplied and added to the original feature map.

$$X = X \otimes Y + X \tag{2.9}$$

After applying attention, the generated feature map is passed through one convolution layer, followed by the classification layer. The above-mentioned channel diversification block can be plugged into any classification network, as shown in Figure 5.4 (a). The input image is first passed through a backbone network to generate the feature map $X = [x_1, x_2, ..., x_c]$, where $X \in \mathbb{R}^{C \times H \times W}$. This generated feature is passed through the channel diversification block, which forces the network to focus more on diverse features using channel attention pooling and significant features using global average pooling. The detail of the channel diversification block has been illustrated in Figure 5.4 (b). From which we can see that, average pooling generates the feature vector of size $C \times 1$ and channel-wise attention pooling generates a feature of size $C \times C$, are normalized and concatenated followed by linear transformation using convolution operation with kernel size $1 \times (C+1)$, which produces $C \times 1$ dimension features.

2.3 Experiments

We evaluate the channel diversification network on CIFAR-10, CIFAR-100, SVHN, and Tiny-ImageNet datasets and compare the performance with respect to various baseline networks.

2.3.1 Datasets

We evaluate the proposed approach on the following four publicly available benchmarks.

CIFAR-10: CIFAR-10 dataset consists a total of 60,000 images of size 32×32 and 10 classes, with 6,000 images per class. There are a total of 50,000 training and 10,000 testing images.

CIFAR-100: Similar to CIFAR-10, it consists of a total of 60,000 images of size 32×32, but has a total of 100 classes, with 600 images per class. There are a total of 50,000 training and 10,000 testing images.

SVHN: SVHN dataset consists of 604,388 training images (train: 73,257 and extra: 53,131) and 26,032 testing images of size 32×32. It categorizes the images into 10 classes.

Tiny-ImageNet. The dataset contains 100,000 training images and 10,000 validation images of size 64×64. It categorizes the images into 200 classes.

2.3.2 Training

During training, we applied standard data augmentations, which include zero-padding with 4-pixels on each side, randomly cropped them in the size of 32×32 for CIFAR-10/100 and SVHN dataset, and 64×64 for the Tiny-ImageNet dataset, and randomly horizontally mirrored them. We trained the network using stochastic gradient descent (SGD) with a momentum 0.9. CIFAR-10 and CIFAR-100 datasets are trained for 200 epochs with an initial learning rate of 0.1 and batch size of 128 for all models except ResNext. For ResNext, we use the initial learning rate of 0.1 and batch size of 64. The SVHN and Tiny-ImageNet datasets are trained for 50 and 200 epochs, respectively, with an initial learning rate of 0.1 and batch-size 128 and 256, respectively, for all models except ResNext. For ResNext, we use an initial learning rate of 0.01 and batch-size 64.

	CIFAR-100	CIFAR-10
Model	Top- 1	Top-1
ResNet-110	73.12	93.57
ResNet-110-SE	76.15	94.79
ResNet-110-SAOL	77.15	95.18
ResNet-110-ABN	77.15	95.09
ResNet-110-Ours	**77.50**	**95.60**
WRN-16-8	79.57	95.73
WRN-16-8-SE	80.86	96.12
WRN-16-8-ours	**80.91**	**96.20**
WRN-28-10	80.13	95.83
WRN-28-10-SAOL	80.89	96.44
WRN-28-10-ABN	**81.88**	96.22
WRN-28-10-ours	81.86	**96.46**
ResNext	81.68	96.16
ResNext-ABN	82.30	96.20
ResNext-Ours	**83.02**	**96.43**
DenseNet	77.73	95.41
DenseNet-ABN	78.37	95.83
DenseNet-SAOL	76.84	95.31
DenseNet-Ours	**78.41**	**95.51**
VGG-16	72.18	92.64
VGG-16-ours	**74.67**	**94.29**
VGG-11	68.64	92.00
VGG-11-ours	**72.18**	**92.94**

Table 2.1: Comparison of Top-1 accuracy on CIFAR-10 and CIFAR-100 datasets with various baseline models and attention based classification models.

2.3.3 Accuracy on CIFAR-10/100 dataset

Table 2.1 shows the top-1 accuracy on CIFAR-10 and CIFAR-100 datasets for various baseline models and our channel diversification network. The accuracy of the baseline models is taken from the original papers. Accuracy with “$*$” indicates the result of re-implementation.

The results indicate that our model consistently improves the performance of all baseline networks: ResNet-110, WRN-16-8, WRN-28-10, ResNext, DenseNet, VGG-16 and VGG-11 by 4.38%, 1.34%, 1.73%, 1.34%, 0.68%, 2.55% and 3.54% respectively on the CIFAR-100 and 2.03%, 0.47%, 0.63%, 0.27%, 0.10%, 1.65% and 0.94% respectively on the CIFAR-10 dataset.

Model	GFLOPS	Parameters	Top-1
ResNet-110-SE	**0.21**	1.89M	76.15
ResNet-110-ABN	0.60	3.06M	77.15
ResNet-110-Ours	**0.21**	**1.76M**	**77.20**
WRN-16-8-SE	2.01	11.2M	80.86
WRN-16-8-ours	**1.01**	**11.0M**	**80.91**
WRN-28-10-ABN	12.4	64.48M	**81.88**
WRN-28-10-ours	**5.46**	**36.5M**	81.86
ResNext-ABN	**0.25**	120.32M	82.30
ResNext-Ours	0.69	**77.79M**	**83.02**
DenseNet-ABN	**0.32**	**1.12M**	78.37
DenseNet-Ours	0.37	1.83M	**78.41**
VGG-16	0.33	34.02M	72.18
VGG-16-ours	**0.32**	**17.4M**	**74.67**
VGG-11	0.17	28.52M	68.64
VGG-11-ours	**0.16**	**11.6M**	**72.18**

Table 2.2: Comparison of Top-1 accuracy, number of parameters, and GFLOPs on the CIFAR-100 dataset with other attention-based classification models.

The highest accuracy on the CIFAR-100 dataset is 83.02% and CIFAR-10 is 96.46% , achieved by our channel diversification block with the baseline networks, ResNext, and WRN-28-10, respectively.

In the case of VGG-Nets, we removed the last max-pooling layer and classification layer and replaced those with the channel diversification block followed by a convolution layer. This modification further reduces the number of parameters and GFLOPs compared to the baseline VGG, as shown in Table 2.

From the result, it can be seen that our model outperforms the state-of-the-art SAOL network for all baseline networks. In the case of ABN, our model outperforms all baseline networks except WRN-28-10 for CIFAR-100 and DenseNet for CIFAR-10. The accuracy difference between ABN and our model for WRN-28-10 and DenseNet is very small, around 0.02% and 0.32% on CIFAR-100 and CIFAR-10, respectively.

It is also evident from Table 2.2 that our model requires less computation and has a fewer number of parameters compared to ABN. For example, ResNet-110 with ABN requires 0.6 GFlops and 3.06M parameters, while ResNet-110 with our channel diversification block requires only 0.21

Model	Top-1
ResNet-110	97.82
ResNet-ABN	98.14
ResNet-110-Ours	**98.15**
WRN-28-10	97.58
WRN-28-10-ABN	97.76
WRN-28-10-ours	**98.20**
ResNext	97.84
ResNext-ABN	97.99
ResNext-Ours	**98.00**
DenseNet	97.93
DenseNet-ABN	97.99
DenseNet-Ours	**97.99**

Table 2.3: Comparison of Top-1 accuracy on SVHN dataset with different baseline models and ABN.

GFLOPs and 1.76M parameters. The comparison of accuracy vs GFLOPs, and the number of parameters for different baseline networks are shown in Table 2.2.

2.3.4 SVHN Accuracy

We evaluate the performance of our model on the SVHN dataset. The top-1 accuracy on SVHN has been shown in Table 2.3, in a similar manner as stated in the CIFAR-10/100 accuracy section. From the result, it is clear that our model outperforms all the baseline networks: ResNet-110, WRN-28-10, ResNext, and DenseNet on the SVHN dataset in terms of accuracy. The results also show that our model achieves competing performance with ABN. For some baseline models, it exceeds the performance of ABN. For example, the baseline networks ResNet-110 and WRN-28-10 outperform the ABN by 0.33% and 0.62% on SVHN dataset, respectively.

2.3.5 Tiny-ImageNet Accuracy

We also evaluate the performance of our model on the Tiny-ImageNet dataset. The top-1 accuracy on this dataset is shown in Table 2.4, from which we can see that the channel diversification block significantly improves the performance of all baseline networks: ResNet-110, DenseNet, Wide-

Model	Accuracy
ResNet-110	62.56*
ResNet-110(ours)	**63.27**
DenseNet	60.00*
DenseNet(ours)	**63.03**
Wide-ResNet	65.99*
Wide-ResNet(ours)	**68.00**
ResNext	68.23*
ResNext(Ours)	**70.28**

Table 2.4: Comparison of Top-1 accuracy of various baseline models with ours. Here, "$*$" indicates the re-implementation accuracy.

Model	Accuracy	Parameters
ResNet-110(Baseline)	73.12	1.70M
ResNet-110 (with only Global Avg Pooling)	76.56	1.70M
ResNet-110 (with only Attention Pooling(using negative correlation))	77.20	1.70M
ResNet-110 (SE)Hu et al. (2018b)	76.15	1.89M
ResNet-110 (GC (using positive correlation))	74.65	1.92M
ResNet-110 fusion(apply after each residual block)	76.93	1.84M
ResNet-110 fusion(using positive correlation)	76.13	1.76M
ResNet-110 fusion(Ours)	**77.50**	**1.76M**

Table 2.5: Comparison of top-1 accuracy on CIFAR-100 only using either the global average pooling, or the global attention pooling, or using both. We also show the individual accuracy of the attention model SE-Net and GC-Net.

ResNet, and ResNext by 0.71%, 3.03%, 2.01%, and 2.05%, respectively. The highest accuracy we achieved on the Tiny-ImageNet dataset is 70.28% by ResNext(ours). In this experiment, the baseline accuracy of various models was obtained by re-implementing all models following the published papers.

2.3.6 Visualizing Attention Maps

We compare the attention maps generated by the baseline network and our channel diversification network using CAM(Zhou et al., 2016). All the attention maps displayed in Figure 2.4 are from the images which could not be correctly classified by the baseline network. From the attention map,

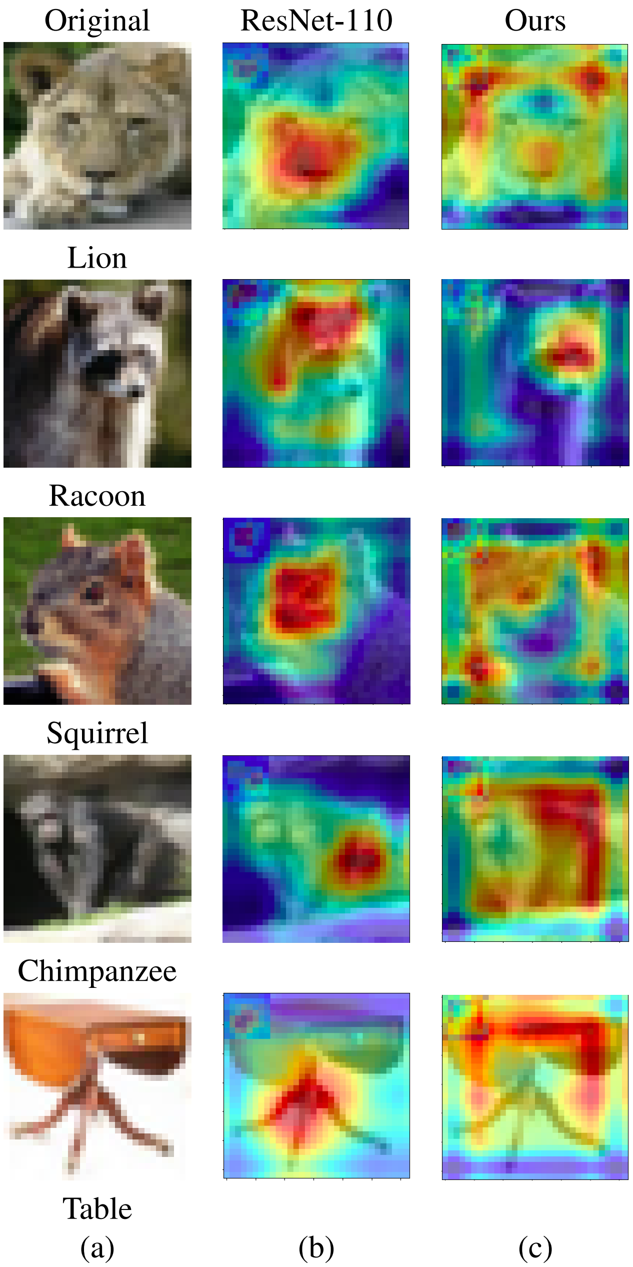


Figure 2.4: Comparison of the attention maps generated by the baseline model and the channel diversification network. (a) Original images; (b) attention maps generated by the baseline model; and (c) attention map generated by our channel diversification block.

we can see the reason behind the misclassification. It gives attention to very common features, such as in the case of Lion, Racoon, and Squirrel images, it focuses on the nose, head, and face parts, respectively, whereas our channel diversification network diverts the attention of the model to more diverse and significant features. For example, in the case of the Lion image, our model focuses on ears and nose; for the Racoon image, it gives attention to the face and body; and for the Squirrel image, the model focuses on ears and eyes, which are very important features classification. Similarly, in the case of the Chimpanzee and Table image, the baseline model focuses on unimportant and unrelated features, whereas our model focuses on the right place. We can also see from Figure 2.4 that our model did not focus on insignificant background features, though it diverts attention.

2.4 Ablation Study

In this section, we report ablation experiments to demonstrate the effectiveness of fusing the global context modeling part of the SE block and simplifying the non-local block. We train ResNet-110 on CIFAR-100 by either using only the global average pooling from the SE block or attention pooling from the simplified non-local block, followed by a transformation that includes convolution with kernel size 1×1 and $C \times 1$ respectively. Our result shows that fusing both of these context modelings and applying the transformation with one convolution layer of size $C \times (C+1)$ on fusion features clearly improves the performance. We also plugin SE-Net and Simplified Non-Local Block at the end of the backbone network. From Table 2.5, it is evident that our model outperforms the SE-Net and the simplified non-local block individually on the CIFAR-100 dataset. In addition, we conduct two comparative experiments by either using positive correlation in our module or inserting our module after each residual block, as shown in Table 2.5. We can see from the experiments that their performance is inferior to the proposed approach. We also compare the number of parameters for each of the cases for ResNet-110 and show that our model has a comparatively low number of parameters while achieving a high accuracy of 77.5%.

2.5 Conclusion

We have proposed a channel diversification block in this paper. The proposed scheme can be embedded in any CNN-based baseline networks to make them concentrate more on the diverse and significant channels at the same time. We have extensively evaluated the performance of the channel diversification block for image classification tasks on various datasets. Our model outperforms all baseline networks and the attention-based classification models: SOAL, ABN, and SE-Net in terms of accuracy, the number of extra parameters, and GFLOPs.

Chapter 3

Aggregating Global Features into Local Vision Transformer

Abstract

Local Transformer-based classification models have recently achieved promising results with relatively low computational costs. However, the effect of aggregating spatial global information of local Transformer-based architecture is not clear. This work investigates the outcome of applying a global attention-based module named multi-resolution overlapped attention (MOA) in the local window-based transformer after each stage. The proposed MOA employs slightly larger and overlapped patches in the key to enable neighborhood pixel information transmission, which leads to significant performance gain. In addition, we thoroughly investigate the effect of the dimension of essential architecture components through extensive experiments and discover an optimum architecture design. Extensive experimental results CIFAR-10, CIFAR-100, and ImageNet-1K datasets demonstrate that the proposed approach outperforms previous vision Transformers with a comparatively fewer number of parameters. The source code and models are publicly available at: `https://github.com/krushi1992/MOA-transformer`

3.1 Introduction

Transformer-based architecture has achieved tremendous success in the field of natural language processing (NLP) (Vaswani et al., 2017) (Devlin et al., 2018). Inspired by the great success of transformer in the language domain, vision transformer(Dosovitskiy et al., 2020) has been proposed and achieved superior performance on the ImageNet dataset. The vision transformer splits

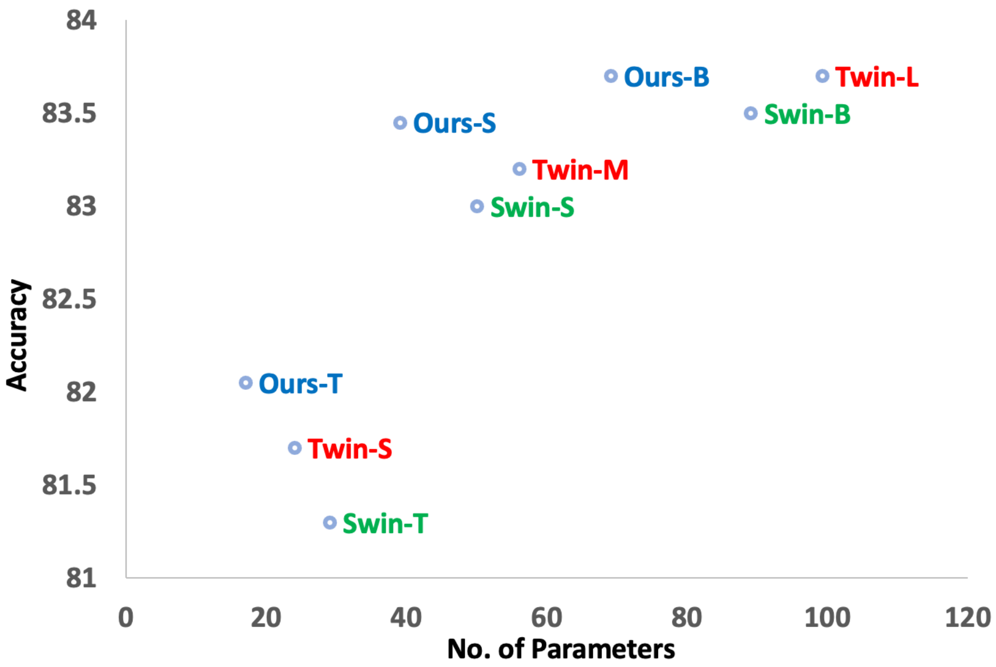


Figure 3.1: Graph of accuracy vs. number of parameters for various local transformer-based models. It shows that our all versions of the model: MOA-T, MOA-S, and MOA-B have higher accuracy and comparatively fewer parameters.

the image into patches and feeds into the transformer, the same way as words token in NLP, and passes through several multi-head self-attention layers of the transformer to establish the long-range dependencies.

Unlike the word token, a high-resolution image contains more pixels compared to words in the passage. This leads to an increase in the computation cost as self-attention in the transformer has quadratic complexity. To alleviate this problem, various local attention-based transformers (Liu et al., 2021) (Vaswani et al., 2021) (Zhang et al., 2021b) have been proposed with a linear computation complexity. However, all the proposed approaches could not establish long-range dependencies and some of them are very complicated.

To overcome these issues in the local transformers, we developed a very simple module, named multi-resolution overlapped attention(MOA), to generate global features. The proposed module only consists of multiplication and addition operations and is embedded after each stage in the transformer before the downsampling operation. As the module is added only after each stage

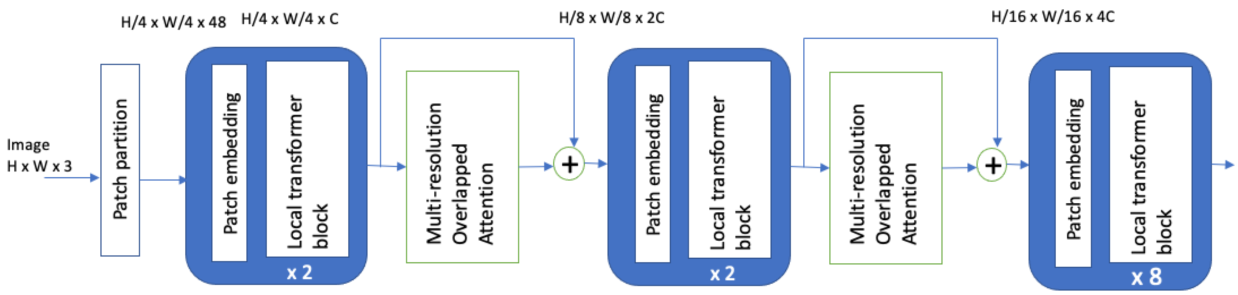


Figure 3.2: The architecture of the MOA-T is composed of three stages. Each stage consists of a patch embedding/merging layer and local transformer block along with a global multi-resolution overlapped attention module after each stage except the last stage. In the beginning patch, a partition layer is included to divide the image into a fixed number of patches.

instead of each transformer layer, it does not add much computation cost and the number of parameters. Our experiments show that aggregating the resultant features of this module to the local transformer establishes the long-range dependencies and hence significantly increases the accuracy in contrast to the total number of parameters as shown in Figure 4.1

Our proposed MOA module takes the output generated by the group of local window-based attention as an input. It first converts it to a 2D feature map, and projects it to a new low-dimension feature map. Similar to ViT (Dosovitskiy et al., 2020), the projected feature map is divided into a fixed number of patches except for a few modifications. In contrast to ViT (Dosovitskiy et al., 2020), the patch sizes of query and key-value are different. The resolution of the patches in the query is the same as the window size used in the local transformer layer. In contrast, the resolution of patches in key-value is slightly larger than the query patch and overlapped. The hidden dimension of the MOA global attention module is kept the same as the previous transformer layer. Therefore, the resultant features are directly aggregated to the output of the previous transformer layer.

Extensive experiments show that keeping the key-value patches slightly larger with overlap to each other leads to significant performance gain due to small information exchange between two neighborhood windows. In short, our method exploits the neighborhood information along with global information exchange between all non-local windows by embedding the proposed MOA mechanism in the local transformer.

The contributions of the proposed approach are summarized as below:

1. We propose a multi-resolution overlapped attention (MOA) module that can be plugged in after each stage in the local Transformer to promote information communication along with nearby windows and all non-local windows.
2. We thoroughly study the impact of global information in local Transformer using the proposed MOA module.
3. We investigate the effect of the dimension of essential architecture components through extensive experiments and discover the optimum architecture for image classification.
4. We train the proposed model from scratch on CIFAR-10/CIFAR-100 (Krizhevsky et al., b) and ImageNet-1K (Deng et al., 2009) datasets and achieve state-of-the-art accuracy using a local transformer.

3.1.1 Convolutional Neural Networks

After the revolutionary invention of AlexNet (Krizhevsky et al., 2012), convolutional neural network (CNN) has become a standard network for all computer vision tasks, such as image classification (Ma et al., 2022)(Patel & Wang, 2022), object detection (Li et al., 2021a), tracking (Zhang et al., 2020b), segmentation (He et al., 2021)(Patel et al., 2021), counting (Sajid et al., 2021a), and image generation (Xu & Wang, 2021). Various versions of CNNs have been proposed to improve the performance by making it deeper and/or broader, such as VGG network (Simonyan & Zisserman, 2014), ResNet (He et al., 2016), Wide-ResNet (Zagoruyko & Komodakis, 2016), DenseNet (Huang et al., 2017), etc. There are also several works proposed to make it more efficient by modifying the individual convolutional layer, such as dilated convolution(Yu & Koltun, 2015), depth-wise separable convolution (Chollet, 2017), group convolution (Krizhevsky et al., 2012), etc. In our work, we employ the convolutional layer along with the transformer layer to reduce the overall dimension of the feature map. Our experiments show that the combination of convolutions and multi-head attention increases the performance.

3.1.2 Self Attention in CNN

Self-attention mechanisms have become ubiquitous in the field of computer vision tasks. Various works (Gajurel et al., 2021)(Wang et al., 2018)(Cao et al., 2019)(Woo et al., 2018)(Sajid & Wang, 2021)(Fu et al., 2019)(Zhao et al., 2018)(Ma et al., 2021) have been proposed that utilize either channel-based or position based self-attention layers to augment the convolution network. Non-local network(Wang et al., 2018) and PSANet(Zhao et al., 2018) model the spatial relationship between all the pixels in the feature map and embed the attention module after each block in CNN, whereas SENet (Hu et al., 2018b) establishes a channel relationship in the convolution network by squeezing the features using global average pooling. CBAM (Woo et al., 2018), BAM (Park et al., 2018) and dual attention network (Fu et al., 2019) employ both channel and position based attention mechanisms separately, then combine the resultant features from both attention modules using either element-wise addition or concatenation and uses the resultant features into convolution output after each stage, whereas GCNet (Cao et al., 2019) combines SENet (Hu et al., 2018b) and non-local network (Wang et al., 2018) together and propose the hybrid attention mechanism that aggregates the information of both channel and spatial relationships in the same attention module.

3.1.3 Vision Transformers

Similar to AlexNet, vision Transformer (ViT) (Dosovitskiy et al., 2020) has changed the perspective of researchers towards solving computer vision problems. Since then, many vision transformer-based networks have been proposed to improve accuracy or efficiency. The ViT needs to be pre-trained on large datasets such as JFT300M (Sun et al., 2017) to achieve high performance. DeiT (Touvron et al., 2021) solves this problem by student-teacher setup, substantial augmentation, and regularization techniques. To train the transformer on the mid-sized dataset like ImageNet-1K from scratch, the token-to-token vision transformer (Yuan et al., 2021) recursively aggregates neighboring tokens (patches) into one token (patch) to reduce the number of tokens. A Cross-ViT (Chen et al., 2021) comes up with a dual branch approach with multi-scale patch size to produce robust image features and pyramid vision Transformer (PVT) (Wang et al., 2021) introduces a multi-scale-based spatial dimension design similar to FPN (Lin et al., 2017a) in CNN and demonstrated good performance. Furthermore, PVT introduced a spatial reduction in key to reduce the computation cost in multi-head attention.

Various local attention-based transformers have been introduced to alleviate the quadratic complexity issues (Vaswani et al., 2021)(Liu et al., 2021)(Zhang et al., 2021b). The HaloNet (Vaswani et al., 2021) introduces the idea of a slightly larger window of key than the query in a local attention mechanism and proves its effectiveness through various experiments. In our model, the key is also calculated using a slightly larger patch, but in the context of global attention, the idea of a larger key is different from the HaloNet. A swin Transformer (Liu et al., 2021) proposes a non-overlapping window-based local self-attention mechanism to avoid quadratic complexity and achieve comparable performance and aggregated nested Transformer (Zhang et al., 2021b) come with the multi-scale approach with block-aggregation mechanism after each stage.

Some Transformer-based works have been proposed to utilize both local and global features (Han et al., 2021) (Chu et al., 2021). A Transformer in Transformer (TNT) (Han et al., 2021) further divides the local patches (visual sentences) into smaller patches (visual words). The MHA on visual word embedding is calculated and aggregated to the sentence embedding to establish the global relationship. The twin Transformer (Chu et al., 2021) is quite the same as ours. However, global attention is applied after each local Transformer layer, increasing the computation cost significantly. In contrast, we apply it after each stage, and we have slightly larger and overlapped patches in key in multi-head attention. The proposed network efficiently utilizes global information in the local transformer and achieves higher accuracy than the above-mentioned transformer-based models.

3.2 Proposed Method

We aim to provide global information exchange across all windows in the local transformer by increasing the minimal computation cost and a number of parameters. An overview of our proposed model is shown in Figure 4.2, which shows MOA module after each stage. All stages have

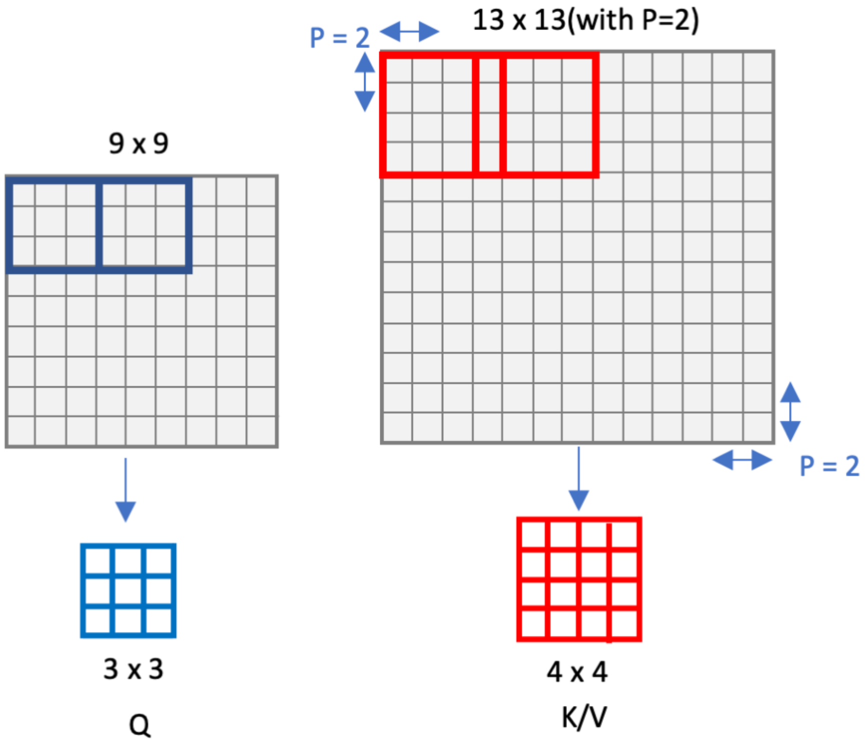


Figure 3.3: Patch creation for query embedding is shown in the blue, and key/value is shown in the red for feature map size 9×9 and window size 3×3. Blue patches have the same size as the window and are non-overlapped to each other. In contrast, red patches are larger and slightly overlapped with each other. Appropriate padding is applied while creating the key-value patches.

a similar architecture design, including patch merging layer and local transformer block except the first stage. The first stage consists of patch partition, linear embedding layer, and local transformer block. Our global MOA module is applied between each stage before the patch merging layer.

Specifically, the model takes an RGB image as an input and splits it into a fixed number of patches. Here each patch is treated as a token. In our experiment on the ImageNet dataset, we set the patch size to 4×4, which leads to $4 \times 4 \times 3 = 48$ feature dimensions for each patch. These row features are projected to a specific dimension C using the patch embedding layer in the first stage. The resultant features are then passed through consecutive stages consisting of patch merging layer, local transformer block, and MOA module in-between each stage. Unlike Swin Transformer (Liu et al., 2021), our Transformer block employs the same self-attention mechanism as ViT (Dosovitskiy et al., 2020) without any shifted window approach. Similar to Swin Transformer, the number of tokens is reduced, and the output dimension is doubled in the patch merging layer after each stage. For example, the resolution after the first, second and third stage is $\frac{H}{2} \times \frac{W}{2}$, $\frac{H}{4} \times \frac{W}{4}$, and $\frac{H}{8} \times \frac{W}{8}$, respectively. The average pooling layer is inserted at the end of the last stage, followed by a linear layer to generate a classification score. The detailed explanation of each element of architecture is as follows:

3.2.1 Patch embedding layer

It is a basic linear embedding layer applied to the row features of patches to project it to a specific dimension C.

3.2.2 Patch merging layer

Patch merging layer reduces the number of tokens by concatenating the features of 2×2 neighboring patches and doubles the number of hidden dimensions by applying a linear layer on the concatenated 4C - dimensional features.

3.2.3 Local Transformer Block

The local transformer block consists of a local window-based standard multi-head attention module, followed by a two-layer MLP with GELU non-linearity. A layer norm is used before each multi-head attention module and each MLP with residual connection after each module.

3.2.4 Multi-resolution Overlapped Attention Block

To utilize the advantage of global information in the local transformer, we apply a global attention module named multi-resolution overlapped attention (MOA) in-between each stage. The architecture of the MOA mechanism is the same as the standard multi-head attention except for a few modifications. Similar to standard MHA, it first divides the feature map into the fixed size of patches. However, unlike the standard MHA, patches for generating key and value embeddings are slightly larger and overlapped, while the patches for query embedding are non-overlapped as shown in Figure 4.3.

As shown in the Figure 4.3, the input to MOA block is of size $W \times H \times$ hidden dim, Where $W = \frac{W}{2}$, $\frac{W}{4}$ or $\frac{W}{8}$, $H = \frac{H}{2}$, $\frac{H}{4}$ or $\frac{H}{8}$, and hidden dim = 96, 192, or 384. Calculating query, key, and value embeddings directly from the input is quite expensive in computation. For example, in context to the ImageNet dataset, the feature map size of the input to MOA block after the first stage is $56 \times 56 \times 96$. Deriving query embedding directly from the input feature with a patch size 14 will lead to the resultant feature of dimension $14 \times 14 \times 96 = 18816$. Therefore, we first reduce the hidden dimension with factor R by applying 1×1 convolution, which reduces the computation cost. The resultant feature dimension after applying the convolution is $H \times W \times \frac{hiddendim}{R}$. This leads to feature size in one query patch being $14 \times 14 \times \frac{hiddendim}{R}$, which is projected to the one-dimensional vector of size: $1 \times 1 \times hiddendim$. The total number of the query is $\frac{H}{14} \times \frac{W}{14}$. Similarly, the key and value vector are projected, but the patch size is slightly larger than the query as shown in Figure 4.3. In our model, we set the key-value patch size to 16. Therefore, the number of key-value will be according to the equation: $(\frac{H-16+(2\times padding)}{stride}+1, \frac{W-16+(2\times padding)}{stride}+1)$. Multi-head attention is applied to this query, key, and value embedding, followed by two-layer MLP with GELU non-linearity in between. Similar to the Transformer block, layer norm is applied along with residual connection after each MOA module. At last, on the resultant features, 1×1 convolution is applied, followed by broadcast addition of resultant features with the output of the previous transformer block, which contains the local information.

3.2.5 Relative Position Index

We use relative position bias $B \in R^{M^2 \times N^2}$, as used by (Bao et al., 2020) (Hu et al., 2019) (Hu et al., 2018a) (Raffel et al., 2019), in the heads of both local and global attention during similarity computation:

$$Attention(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d}}+B)V \tag{3.1}$$

Model	Dataset	Input-Size	Window-Size	No. of Layers	No. of Heads	Hidden Dim
T	CIFAR	32×32	4×4	[2, 2, 6, 2]	[3, 6, 12, 24]	[96, 192, 384, 768]
B	CIFAR	32×32	4×4	[2, 2, 6, 2]	[4, 8, 16, 32]	[128, 256, 512, 1024]
T	ImageNet	224×224	14×14	[2, 2, 8]	[3, 6, 12]	[96, 192, 384]
S	ImageNet	224×224	14×14	[2, 2, 20]	[3, 6, 12]	[96, 192, 384]
B	ImageNet	224×224	14×14	[2, 2, 20]	[4, 8, 16]	[128, 256, 512]

Table 3.1: Model configuration for CIFAR/ImageNet dataset

where $Q \in R^{M^2\times d}$ is a query matrix, $K,V \in R^{N^2\times d}$ are the key and value matrices; d is the hidden dimension, M^2 is the total number of patches in the queue and N^2 is total the number of patches in the key.

3.2.6 Architecture Detail

By following the previous works(Liu et al., 2021)(Chu et al., 2021), we build three versions of the model: MOA-T, MOA-S, and MOA-B for the ImageNet dataset and only two versions of the model: MOA-T and MOA-B for the CIFAR -10/100 dataset as it is quite smaller. Table 3.1 shows the architecture configurations for the CIFAR and the ImageNet datasets. In the CIFAR based models, both MOA-T and MOA-B contain the same number of Transformer layers: 12, but have a different number of hidden dimensions. In context to the ImageNet based models, the total number of layers for MOA-T and MOA-S is 12 and 24 respectively, but the hidden dimension is kept the same, whereas MOA-S and MOA-B have the same number of Transformer layers: 24, with contrast hidden dimensions 96 and 124 respectively.

3.3 Experimental Evaluations

We verify our model through extensive experiments on CIFAR-10/CIFAR-100 and ImageNet-1K for image classification. We design three architecture versions: MOA-T, MOA-S, and MOA-B, for the classification tasks.

3.3.1 CIFAR-10/100 Results

CIFAR-10 and CIFAR-100 datasets consist of 50,000 training and 10,000 test images of resolution 32×32 with the total number of classes 10 and 100, respectively. We train the network for 300 epochs using AdamW (Kingma & Ba, 2014) optimizer with an initial learning rate of 0.009 and weight decay of 0.05. We utilize a cosine decay learning rate schedular along with 20 warm-up epochs. We implemented two models: MOA-T and MOA-B for the CIFAR dataset with a total batch-size 128 and stochastic drop-rate 0.2 (Larsson et al., 2016).

Table 3.2 shows the performance of our model on the CIFAR-10 and CIFAR-100 datasets. We presented only two models with the same number of layers but with different hidden dimensions for this dataset. As shown in the table, it can be seen that both models outperform all the previous Transformer-based models by a significant amount. It improves the performance by 0.59% and 0.98% on CIFAR-10 and 0.56% and 0.23% on CIFAR-100 for the Tiny and Base models, respectively, compared to Swin Transformer. For the Base model, our model achieves state-of-the-art accuracy on the local vision Transformer with a comparatively fewer number of parameters and GFLOPs. The accuracy of other models is reported by training the models from scratch with the same training setting reported in the papers (Liu et al., 2021) (Touvron et al., 2021) (Wang et al., 2021).

Model	CIFAR-100(%)	CIFAR-10(%)	Parameters
Deit-T	70.33	89.2	5M
PVT-T	72.80	91	13M
Swin-T	78.07	94.41	27.5M
MOA-T	**78.63**	**95**	30M
DeiT-B	71.54	93	85M
PVT-B	70.1	89.87	61M
Swin-B	78.45	94.47	86.7M
MOA-B	**78.68**	**95.05**	53M

Table 3.2: Results on CIFAR - 10/100

Model	Accuracy(%)	Parameters	GFLOPs
Deit-Small/16	79.9	22.1M	4.6
CrossViT-S	81.0	26.7M	5.6
T2T-ViT-14	81.5	22M	5.2
PVT-Small	79.8	24.5M	3.8
TNT-T	73.9	6.1M	1.4
Twins-PCPVT-S	81.2	24.1M	3.8
Swin-T	81.3	29M	4.5
Twins-SVT-S	81.7	24M	2.9
MOA-T	**82.05**	17M	4.8
T2T-ViT-19	81.9	39.2M	8.9
PVT-Medium	81.2	44.2M	6.7
TNT-S	81.5	23.8M	5.2
Twins-PCPVT-B	82.7	43.8	6.7
Swin-S	83.0	50M	8.7
Twins-SVT-B	83.2	56M	8.6
MOA-S	**83.5**	39M	9.4
ViT-Base/16	77.9	86.6M	17.6
Deit-Base/16	81.8	86.6M	17.6
T2T-ViT-24	82.3	64.1M	14.1
CrossViT-B	82.2	104.7M	21.2
PVT-Large	81.7	61.4M	9.8
TNT-B	82.9	65.6M	14.1
Swin-B	83.3	15.4M	83.7
Twins-SVT-L	**83.7**	99.2M	15.1
MOA-B	**83.7**	68M	16.2

Table 3.3: Results on ImageNet-1K

3.3.2 ImageNet Results

ImageNet-1K dataset consists of around 1.28M training images and 50K validation images with 1000 classes. We resize all the images to the resolution 224 × 224 during training. We follow the same training technique, like Swin and Twin, and train the network for 300 epochs using AdamW (Kingma & Ba, 2014) optimizer with a cosine learning rate schedular and 20 warmup epochs. We keep the batch-size 128 for MOA-T and 64 for MOA-S and MOA-B models per GPU. We employ a total of four GPUs together during training leading to a total batch-size of 512 for MOA-T and 256 for MOA-S and MOA-B models. We utilize the same augmentation technique used by (Liu et al., 2021) such as a mixture of cutmix (Yun et al., 2019) and mixup (Zhang et al., 2017) and regularization technique stochastic drop rate. We set the drop rate (Larsson et al., 2016) of 0.2, 0.3, and 0.5 respectively for MOA-T, MOA-S, and MOA-B.

Table 3.3 shows our model's result and a similar Transformer-based model on the ImageNet-1K classification task. Our proposed models: MOA-T, MOA-S, and MOA-B, achieve higher accuracy than most of the Transformer-based models with significant parameter reduction. MOA-T outperforms Twin-S and Swin-T by 0.34% with around 22% fewer parameters. Our MOA-S improves the performance by 0.5% and 0.3% compared to Swin-S and Twin-M respectively, even with the lower batch size during training. Our MOA-B achieves the state-of-the-art accuracy of 83.7% on ImageNet-1K with comparatively fewer parameters with a smaller batch size than the remaining vision transformers. Our model increases the computation cost by a negligible amount, but the performance improvement and parameter reduction are highly rewardable.

3.4 Ablation Study

In this section, we conduct ablation experiments to understand the effect of the dimension of each component, such as window size, the overlapped area between the key-value patches, and the reduction factor in global attention, in our model. We employ the Tiny model to perform all ablation experiments, and all the experiments are performed either on CIFAR-100 or ImageNet dataset. The training configurations remain the same as reported in the experiment section.

3.4.1 Window-size

The sequence length of the local-Transformer is one of the essential factors on which computation cost relies. As the sequence length increases, the computation cost in the self-attention mechanism increases as well. In a local vision Transformer, sequence length depends on the window size. There is always a trade-off between the accuracy and computation cost based on the sequence length. We perform experiments with various window sizes in our model and find that 4×4 and 14×14 window size works well on CIFAR-100 and ImageNet datasets, respectively, as shown in

Table 3.4. Furthermore, we remove the stages where the window size is greater than the feature map size to significantly reduce the number of parameters.

Window-Size	Dataset	No. of Stage	Accuracy	Parameters
2×2	CIFAR -100	4	76.04	29.7M
4×4	CIFAR-100	4	78.61	30M
8×8	CIFAR-100	3	76.02	16M
7×7	ImageNet	4	81.4	31M
14×14	ImageNet	3	82.07	17M
28×28	ImageNet	2	78.2	6M

Table 3.4: Results with different window-size on ImageNet

3.4.2 Overlapped Portion

To initiate the neighborhood information transmission, we propose to use slightly larger and overlapped keys. To investigate the effect of the portion of the overlapped area, we perform experiments with different percentages of overlapped portions in keys as shown in Table 3.5. It can be seen from the results that the performance is increased in terms of accuracy as the percentage decreases, which means only a slight information exchange between the neighborhood windows is required to improve the performance. Furthermore, fewer overlapped portions decrease the sequence length, which reduces the number of parameters and GFLOPs.

% Overlap	Accuracy	Parameters
17%	78.63	30.05M
33%	78.52	30.06M
50%	78.38	30.08M
66%	78.38	30.59M

Table 3.5: Results on CIFAR-100 with different percentages of the overlapped portion

3.4.3 Reduction

Before the MOA global attention, the hidden dimension is reduced to decrease the number of parameters and computation cost. Table 3.6 shows the performance of our model with various values of R. From the result, it is evident that R = 32 achieves the best result with a comparatively smaller number of parameters and computation cost than a smaller value of R.

Reduction	Accuracy	Parameters
8	78.38	31.67M
16	78.34	30.59M
32	78.63	30.06M
64	78.51	29.78M
num-heads	78.41	31.43M

Table 3.6: Results with different window-size on CIFAR-100

Model	Accuracy	Parameters
Without Global	75.56	27M
With Global (ViT)	78.34	30.59M
With Global (Ours)	78.63	30.06M

Table 3.7: Significance of global attention and overlapped patches

3.4.4 Effect of Overlapped Key-Value

To verify the effect of overlapped and larger key-value patches, we train the model without overlapping patches and compare the results. Furthermore, we also conduct an experiment without applying global attention in-between each stage to verify the significance of global information exchange. From the result in Table 3.7, it can be seen that including global attention and overlapped key-value patches achieves the best performance.

3.5 Conclusion

The paper has investigated the effect of aggregating global information in local Transformer after each stage and neighborhood pixel information transmission. We have also proposed a multi-resolution overlapped attention (MOA) module that can be plugged in after each stage in the local transformer to promote information communication along with nearby windows. Our results show that both types of features: global and local, are crucial for image classification. As a result, exploiting both features leads to significant performance gain on the standard classification datasets such as CIFAR10/100 and ImageNet with comparatively fewer parameters.

Chapter 4

Enhanced U-Net: A Feature Enhancement Network for Polyp Segmentation

Abstract

Colonoscopy is the procedure to detect colorectal polyps, which are the primary cause of developing colorectal cancer. However, polyp segmentation is a challenging task due to the diverse shapes, sizes, colors, and textures of polyps, the shuttle difference between polyp and its background, as well as low contrast of the colonoscopic images. To address these challenges, we propose a feature enhancement network for accurate polyp segmentation in colonoscopy images. Specifically, the proposed network enhances the semantic information using the proposed Semantic Feature Enhance Module (SFEM). Furthermore, instead of directly adding encoder features to the respective decoder layer, we introduce an Adaptive Global Context Module (AGCM), which focuses only on the encoder's significant and hard fine-grained features. The integration of these two modules improves the quality of features layer by layer, which in turn enhances the final feature representation. The proposed approach is evaluated on five colonoscopy datasets and demonstrates superior performance compared to other state-of-the-art models.

4.1 Introduction

Colorectal cancer is the third most common cancer diagnosed in the United States (Silva et al., 2014a). It is considered the second deadliest cancer in terms of mortality, causing 9.4% of total cancer deaths (Sung et al., 2021). The primary reason behind colorectal cancer is a polyp that grows in the lining of the colon or rectum. Early detection and localization of polyps can reduce the mortality rate caused by colorectal cancer. In addition, it could reduce the treatment cost by restricting cancer spread to distant organs and the need for biopsy. Colonoscopy is the standard visual examination for the screening of colorectal cancer. However, the result of colonoscopy may be misleading due to the variant nature of polyps, including their shape, size, texture, and unpredictable factors such as veins and illumination. In addition, the result of colonoscopy depends on various human factors including inexperienced gastrologists and physical and mental fatigue. Therefore, an automatic computer-aided polyp segmentation system is required to assist the physician during the procedure and significantly improve the polyp detection rate.

Various techniques have been developed for the polyp segmentation task. The available methods can be largely divided into two categories: (1) Hand-crafted feature based approaches and (2) Deep-learning based approaches. Before the invention of neural networks, the polyp segmentation task depends on hand-crafted features such as size, shape, texture, and color(Tajbakhsh et al., 2015a). However, these approaches are slow and have a high misdetection rate due to the low representation capability of hand-crafted features. Following the huge success of deep learning-based models on generic datasets, researchers started using neural networks for polyp detection and segmentation. Inspired by the early work (Tajbakhsh et al., 2015b), where FCN (Long et al., 2015) is utilized with a pre-trained model to segment the polyp, Akbari et al. Akbari et al. (2018a) proposed a modified version of FCN to improve the performance of polyp segmentation. U-Net++ (Zhou et al., 2018) and ResUNet++ (Jha et al., 2019) upgraded the architecture of U-Net (Ronneberger et al., 2015) and achieved promising results on polyp segmentation. SFANet (Fang et al., 2019) takes the area-boundary constraint into account along with extra edge supervision. It achieves good results but lacks of generalization capability. Recently introduced ACSNet (Zhang et al., 2020a) and PraNet (Fan et al., 2020) propose an attention-based mechanism to focus more on the hard region, which leads to improved performance.

U-Net and its variants U-Net++, ResUNet, ResUNet++, and ACSNet have achieved appealing results on the polyp segmentation task by using U-shape encoder-decoder architecture. However, none of them utilize decoder features to calculate the attention value of the respective encoder layer. We believe that utilizing the decoder layer feature to selectively aggregate respective encoder layer features could improve the feature quality. Moreover, all of the above-mentioned models employ the pooling-based approach to enhance the multi-scale semantic features, which may lead to loss of spatial information.

To alleviate these issues, we propose an attention-based U-Net for polyp segmentation by enhancing the quality of features. The proposed network mainly consists of two modules. First, we design a Semantic Feature Enhancement Module(SFEM), which enhances the deeper layer features by applying different sizes of patch-wise non-local attention blocks to tackle the different sizes of the polyp and fuse the output of each non-local blocks together. These fused features are then sent to each decoder layer. Second, we introduce an Adaptive Global Context Module(AGCM), which focuses on more significant features of the encoder layer by taking into account the previous decoder layer features. This mechanism suppresses the insignificant and noisy features and focuses only on essential features using spatial cross attention. It refines the decoder features layer by layer by removing unwanted features and adding significant fine-grained features only. Furthermore, to give more attention to the hard regions, we apply focal loss at each decoder layer.

In summary, the main contributions of the paper include:

- The proposed semantic feature enhancement module fully exploits the multi-scale semantic context without losing spatial information.
- The proposed adaptive global context module attends the significant and hard fine-grained features and selectively aggregates them to the respective decoder layer.
- The integration of both modules enhances the quality of features layer by layer and hence achieves state-of-the-art performance on five publicly available benchmark datasets.

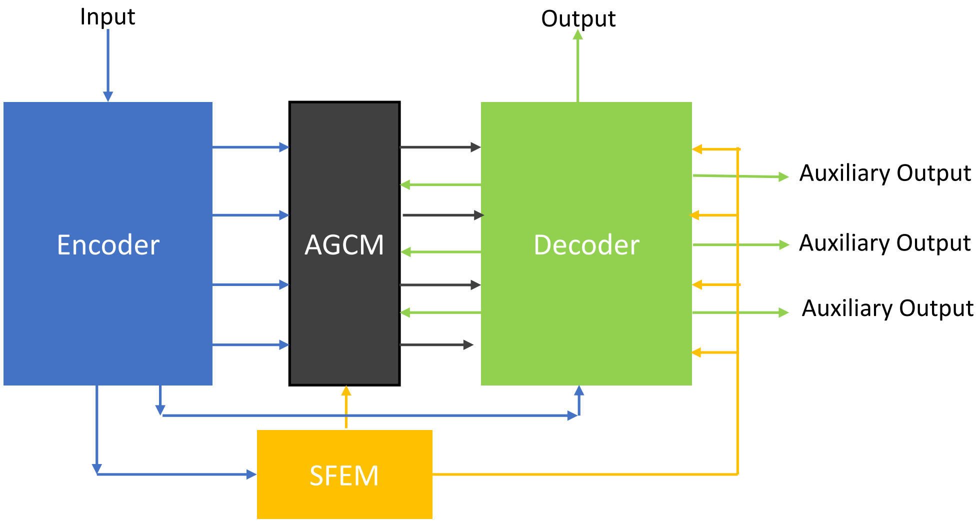


Figure 4.1: The overall architecture of the enhanced U-Net: The input image is supplied to the encoder. Each encoder layer's features are sent to the respective decoder layer through the AGCM module. The features of the last encoding layer are applied to SFEM to further enhance the multi-scale semantic features. The resultant features are sent to all decoder layers to concatenate with the features produced by AGCM and each decoder layer. Auxiliary losses are applied at the end of each decoder layer.

4.2 Method

The architecture of the proposed enhanced U-Net is shown in Figure 4.1. It mainly consists of four parts: (1) Encoder, (2) Decoder, (3) SFEM, and (4) AGCM. An encoder is made up of ResNet-34 (He et al., 2016). The encoder's output is fed to the decoder, which consists of five decoding layers. Each decoding layer consists of two convolution layers followed by batch-normalization and ReLU activation. The SFEM module is attached at the top of the last encoding layer, which consists of semantic features. We insert one convolution layer before the SFEM module to reduce the number of channels. The output of SFEM is sent to all decoding layers. The AGCM module is employed in-place of skip connection to alleviate the effect of background noise. It takes the current encoding layer and the previous decoding layer's feature maps as input and yields the resultant feature map of the same size as the current encoding layer feature map. Feature maps produced by

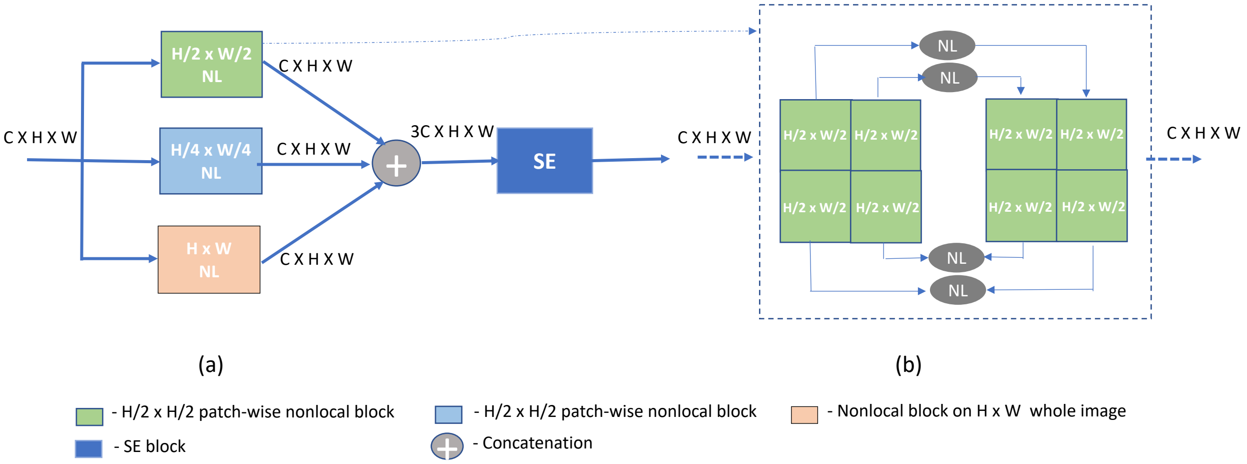


Figure 4.2: (a) The detailed architecture of SFEM. It consists of three branches: the 1st and the 2nd branch divide the image into $H/2 \times H/2$ and $H/4 \times H/4$ sizes of patches and apply non-local attention. The third branch applies basic non-local attention to the whole image. The detailed version of a patch-wise non-local block is described in (b) where it first divides the image into patches and then applies non-local attention on each patch independently and folds it back to the whole image. The result of each branch is concatenated, followed by a SE-block.

SFEM, AGCM, and the decoder layer are concatenated and applied to the next decoding layer and AGCM. Each decoding layer is attached to the auxiliary loss inspired by deep supervision. The detailed description of the two proposed modules are as follows:

4.2.1 Semantic Feature Enhancement Module

It is well known that the deeper layers in CNN networks contain the semantic features that are most significant to detecting and segmenting the objects. To fully exploit the semantic features, we introduce a semantic feature enhancement module (SFEM) inspired by the pyramid pooling (He et al., 2019) (Liu et al., 2019) (Zhao et al., 2017).

Specifically, SFEM consists of three parallel branches of patch-wise non-local blocks as shown in Figure 4.2. It takes the output of the encoder feature map as input and applies non-local attention to the patches of a specific window size instead of applying adaptive average pooling. The first branch divides the image into four patches of size $(W/2 \times H/2)$, applies non-local spatial attention individually on each patch, and folds them back together as shown in Figure 4.2(b). Similarly, the second branch produces 16 patches of the size $(W/4 \times H/4)$ and performs the same operation as the first branch on each patch. In our experiment, we set the size of the output feature map of the encoder to 8×8. Therefore, the first branch contains the 4 patches of size 4×4, and the second branch has the 16 patches of size 2×2. The last branch performs a non-local operation on the entire feature map of size 8×8. The outputs of these three branches are concatenated, followed by a squeeze and excitation block that attends to the most significant channels. The results of SE blocks are then sent to all decoder layers. To match each decoder layer's size, we upsample the output of SFEM.

Unlike pyramid pooling, the above SFEM module is capable of enhancing the semantic information without losing spatial information. In SFEM, the size of each branch's output is the same, whereas for pyramid pooling, the window size increases, the output size decreases, which requires an upsampling operation that leads to loss of spatial information.

4.2.2 Adaptive Global Context Module

Features generated using the SFEM module are at a coarse level and contain noise in it. We propose an adaptive global context module (AGCM) to improve these coarse-level features to fine-level features layer by layer using spatial cross-layer attention. The detailed architecture of the AGCM module is shown in Figure 4.3. It takes the current encoder feature map as query and concatenated features of SFEM, previous layer AGCM, and decoder layer as a key and value pair and applies cross-layer spatial attention. The resultant attention features have the same size as the encoder layer feature map, so they can be directly aggregated to the encoder feature map without resizing operation. The aggregated features are then sent to the respective decoder layer.

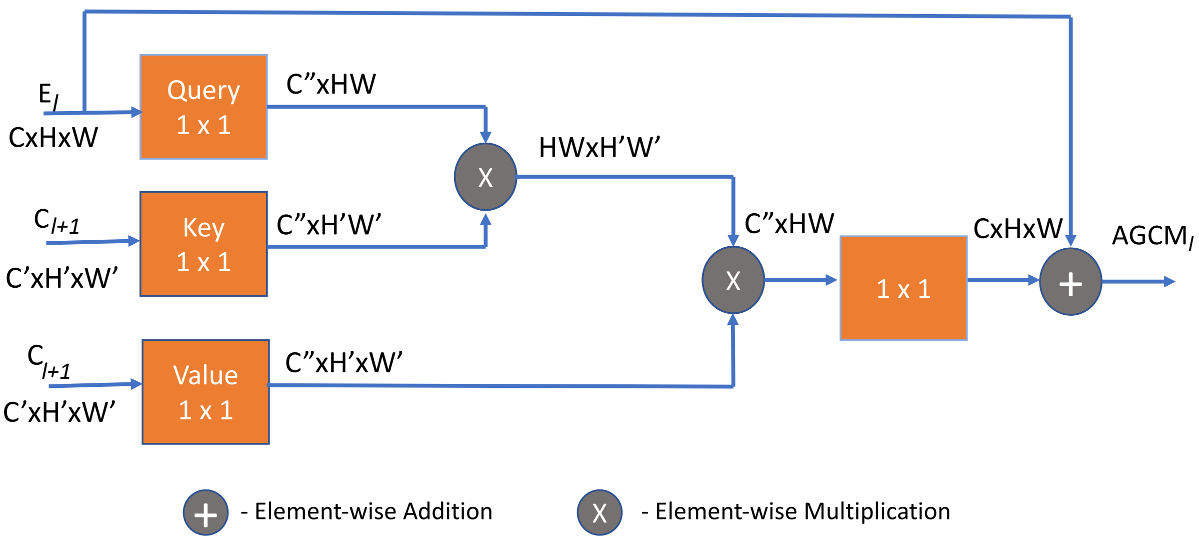


Figure 4.3: The overall architecture of AGCM. It takes current encoder features E_l as Query and concatenates features (C_{l-1}) generated from $(D_{l-1}, AGCM_{l-1}, SFEM)$ as Key and Value to perform cross attention.

4.2.3 Loss Function

Our loss function is defined as:

$$L = L_{IoU} + L_{Focal} + L_{Dice} \tag{4.1}$$

where, L_{IoU}, L_{Focal} and L_{Dice} represent the pixel-based IoU loss, focal loss and dice loss(Lin et al., 2017b)(Sudre et al., 2017)(Rezatofighi et al., 2019). Here, we include the focal loss to give more attention to hard pixels to highlight their significance. In addition, we utilize deep supervision for all decoder layer prediction maps generated by side-out. We downsample the ground truth mask to match the size of the prediction generated by appropriate decoding layer.

4.3 Experiments

4.3.1 Datasets

We evaluate the proposed model on the following five benchmark datasets for polyp segmentation: ETIS (Silva et al., 2014b), CVC-ClinicDB (Bernal et al., 2015), CVC-ColonDB (Tajbakhsh et al., 2015c), Endoscene, and Kvasir (Jha et al., 2020). We compare the enhanced U-Net with the baseline models: U-Net (Ronneberger et al., 2015), U-Net++ (Zhou et al., 2018), and ResUNet++ (Jha et al., 2019). We also compare the performance of our model with the recently proposed ACS (Zhang et al., 2020a) and PraNET (Fan et al., 2020). Specifically, we perform the experiments in two modes of the dataset: Set-1 and Set-2. For the first mode, Set-1, we divide the Kavasir-SEG and CVC-ColonDB datasets into Train, Val, and Test sets individually. In contrast, for the second mode, Set-2, we combine both datasets and used them to train the model and evaluate performance on a totally different dataset, including ETIS, CVC-300, and CVC-ColonDB.

4.3.2 Implementation Details

During the training, we resize all images of the Kavasir-SEG dataset to 384 X 288 and the remaining dataset to 320 X 320 and then randomly crop the images of size 256 X 256. We utilize several data augmentation methods to reduce overfitting, including horizontal and vertical flips, rotation, and zoom. We set the batch size to 4 and train the model for 150 epochs with an initial learning rate of 0.001. We employ the SGD optimizer with a momentum of 0.9 and weight decay of 0.0005.

To evaluate the permanence, we use recall, precision, specificity, dice score, IoU, and accuracy as evaluation metrics. To make a fair comparison, we follow the same procedure to calculate the metric as ACM and PraNet.

4.3.3 Results

We compare the performance of our "Enhanced U-Net" with FCN(Long et al., 2015), U-Net(Ronneberger et al., 2015), U-Net++(Zhou et al., 2018), SegNet(Wickstrøm et al., 2020), SFANet(Fang et al., 2019), and ACSNet(Zhang et al., 2020a) on Endoscene and the recently released Kvasir-SEG datasets. Table 4.1 and Table 4.2 show the results on EndoScene and Kvasir-SEG datasets, respectively. Our model outperforms all the above state-of-the-art models with an adequate margin on almost all metrics. Specifically, our model increases the Dice and IoU by 12.09% and 16.06% on the Endoscene dataset and 9.73% and 13.61% on the Kvasir dataset, respectively, compared to the baseline U-Net. It also outperforms the ACSNet by improving the majority of metrics by a significant amount on both datasets. This indicates the effective learning ability of our model to segment the polyp.

Models	Recall	Specificity	Precision	Dice	IoU	Accuracy
FCN8	60.21	98.60	79.59	61.23	48.38	93.77
UNet	85.54	98.75	83.56	80.31	70.68	96.25
UNet++	78.90	99.15	86.17	77.38	68.00	95.78
SegNet	86.48	99.04	86.54	82.67	74.41	96.62
SFANet	85.51	98.94	86.81	82.93	75.00	96.61
ACSNet	90.18	99.19	93.13	90.27	85.31	98.13
Ours	**92.55**	**99.47**	**93.54**	**92.40**	**86.74**	**98.97**

Table 4.1: Results on the EndoScene dataset to prove the learning capability. Train and Test sets are from the same dataset.

Models	Recall	Specificity	Precision	Dice	IoU	Accuracy
UNet	87.89	97.69	83.89	82.85	73.95	95.65
UNet++	88.67	97.49	83.17	82.80	73.74	94.49
ResUNet	81.25	98.31	87.88	81.14	72.23	94.90
SegNet	90.03	98.13	87.15	86.43	79.11	96.68
SFANet	91.99	97.05	82.95	84.68	77.06	95.71
ACSNet	93.76	**98.02**	91.94	92.23	87.20	**97.74**
Ours	**93.89**	97.92	**92.69**	**92.58**	**87.56**	97.69

Table 4.2: Results on the Kvasir-SEG dataset to prove the learning capability. Train and Test sets are from the same dataset.

To validate the generalization capability of our method, we further evaluate the performance of our model using new datasets that have never been seen before. We follow the same procedure to calculate the mean-IOU, mean Dice, and Accuracy and utilize the same train and test set as PraNet(Fan et al., 2020) for a fair comparison. We then evaluate and compared the performance of different models using the following new datasets: ColonDB, ETIS, and CVC-300. The results are shown in Table 4.3. It is evident that our model improves the mean-Dice and mean-IOU by 22.79% and 21.48% on the ColonDB dataset, 25.25% and 24.7% on the ETIS dataset, and 17.62% and 18.6% on the CVC-300 dataset compare to the baseline U-Net. It also outperforms the recently proposed PraNet by increasing the mean Dice and mean IoU by an adequate amount. In short, it outperforms state-of-the-art methods on the majority of metrics with a significant margin, which demonstrates the superior generalization capability of the method.

Dataset	Models	mean Dice	mean IoU	Accuracy
ColonDB	U-Net	51.2	44.4	93.9
	U-Net++	48.3	41.0	93.6
	SFA	46.9	34.7	90.6
	Pra-Net	70.9	64.0	**95.5**
	Ours	**73.99**	**66.28**	95.35
ETIS	U-Net	39.8	33.5	96.4
	U-Net++	40.1	34.4	96.5
	SFA	29.7	21.7	89.1
	Pra-Net	62.8	56.7	**96.9**
	Ours	**65.07**	**58.20**	96.50
CVC-300	U-Net	71.0	62.7	97.8
	U-Net++	70.7	62.4	98.2
	SFA	46.7	32.9	93.5
	Pra-Net	87.1	79.7	99.0
	Ours	**88.62**	**81.30**	**99.26**

Table 4.3: Results on ColonDB, ETIS, and CVC-300 to prove the generalization capability of the model. Train and Test sets come from different datasets.

Models	Set-1			Set-2		
	Mean Dice	Mean IoU	Accuracy	Mean Dice	Mean IoU	Accuracy
Baseline	88.77	83.81	98.33	78.52	70.27	97.37
Only SFEM	91.39	85.46	**99.04**	83.56	76.83	98.08
Only AGCM	91.31	85.34	**99.04**	83.13	76.33	98.13
Both	**92.40**	**86.74**	98.97	**88.62**	**81.30**	**99.26**

Table 4.4: Ablation study on set-1 and set-2. Performance comparison of the model using only SFEM, only AGCM, or both together.

4.3.4 Ablation Study

4.3.4.1 Effect of SFEM

Only including SFEM improves the performance of the baseline network in both test sets as shown in Table 4.4. Specifically, the mean Dice, mean IoU, and Accuracy on the set-2 dataset increase dramatically by 5.04%, 6.56%, and 0.71% respectively compared to the baseline network which indicates the generalization capability of SFEM. For the set-1 dataset, mean Dice, mean IoU, and Accuracy are increased by 2.62%, 1.65%, and 0.74% respectively compared to the baseline U-Net, indicating the improvement of our model's learning ability.

4.3.4.2 Effect of AGCM:

As shown in Table 4.4, AGCM also improves the learning and generalization capability of the model compared to the baseline. Mean Dice, mean Iou, and Accuracy of set-1 test are improved by 2.54%, 1.53%, and 0.71%. For set-2, they are improved by 4.61%, 6.06%, and 0.76%, respectively.

4.4 Conclusion

This paper has presented a novel architecture to improve the quality of features layer by layer for automatic polyp segmentation from colonoscopy images. Our extensive experiments prove that our model consistently outperforms the baseline network U-Net and its variants: U-Net++ and ResUNet, by a large margin on different datasets. It also outperforms the recently published

ACSNet and PraNet by a significant margin. The experiments demonstrate the strong learning capability and generalization ability of the proposed model. The model could also be applied to other medical image segmentation tasks.

Chapter 5

FuzzyNet: A Fuzzy Attention Module for Polyp Segmentation

Abstract

Polyp segmentation is essential for accelerating the diagnosis of colon cancer. However, it is challenging because of the diverse color, texture, and varying lighting effects of the polyps as well as the subtle difference between the polyp and its surrounding area. To further increase the performance of polyp segmentation, we propose to focus more on the problematic pixels that are harder to predict. To this end, we propose a novel attention module named Fuzzy Attention to focus more on the difficult pixels. Our attention module generates a high attention score for fuzzy pixels usually located near the boundary region. This module can be embedded in any convolution neural network-based backbone network. We embed our module with various backbone networks: Res2Net, ConvNext and Pyramid Vision Transformer and evaluate the models on five polyp segmentation datasets: Kvasir (Jha et al., 2020), CVC-300 (Vázquez et al., 2017), CVC-ColonDB (Tajbakhsh et al., 2015c), CVC-ClinicDB (Bernal et al., 2015), and ETIS (Silva et al., 2014b). Our attention module with Res2Net as the backbone network outperforms the reverse attention-based PraNet by a significant amount on all datasets. In addition, our module with PVT as the backbone network achieves state-of-the-art accuracy of 0.937, 0.811, and 0.791 on the CVC-ClinicDB, CVC-ColonDB, and ETIS, respectively, outperforming the latest SA-Net, TransFuse, and Polyp-PVT. The source code is available at: https://github.com/krushi1992/FuzzyNet.

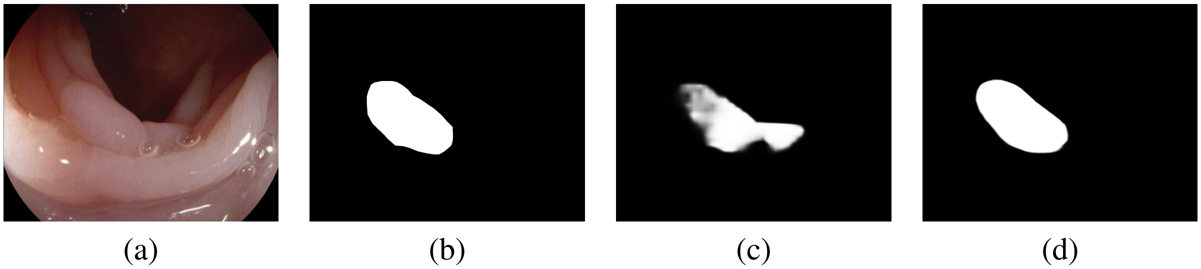


Figure 5.1: (a) An original image of the polyp; (b) the ground-truth mask; (c) the prediction mask generated by PraNet; and (d) the prediction mask generated by our Fuzzy-Net.

5.1 Introduction

Polyp segmentation is an essential task to accelerate the diagnosis of colorectal cancer(Mathur et al., 2020)(Patel et al., 2021)(Silva et al., 2014b), which is considered the most prevalent cancer worldwide. If the polyp is detected earlier, the mortality rate can further be reduced. Colonoscopy is considered the effective technique for CRC screening, which detects the polyps that may cause colon cancer. Detecting polyps is a complicated process because of their similar appearance to background pixels. Sometimes, even an experienced clinician finds it very difficult to recognize, and thus leads to missing detection of polyps because of their subtle difference (Li et al., 2021a) (Patel et al., 2020). In addition, polyps are widely varied in size, texture, and color. Therefore an accurate and automatic polyp segmentation method is required to detect the cancerous polyp in the early stage to reduce the mortality rate (Jia et al., 2019).

Convolution neural networks have achieved tremendous performance gain on various medical image segmentation tasks, including the polyp segmentation (Akbari et al., 2018b)(Brandao et al., 2017)(He et al., 2021)(Jha et al., 2019) (Ronneberger et al., 2015)(Zhou et al., 2018). Various methods have been proposed to tackle the issue of detecting difficult boundary pixels accurately, either by using separate edge supervision (Fang et al., 2019)(Murugesan et al., 2019) or attention modules. However, the use of edge supervision reduces the generalization capability of the model and requires extra boundary annotations, which are expensive. The attention-based methods used reverse attention (Fan et al., 2020), focusing on the background region to mine the boundary clues.

However, we believe the performance can be further improved if we focus more on the difficult pixels instead of the background pixels. Therefore, in this work, we propose a novel attention module named Fuzzy attention to encourage the model to focus on the hard boundary pixels.

The pixels that are not categorized straightforwardly as foreground or background pixels are considered hard pixels. The smaller the difference between the foreground and background attention score, the higher the complexity. Our attention module uses the above observation to calculate the final attention score, which results in a high score for difficult pixels, usually lying around the boundary region, and lower weights for the easy pixels. Figure 5.1 shows an image of the polyp, its ground truth, and the prediction masks generated by PraNet and our FuzzyNet, respectively. It can be seen from the original image that the region around the boundary is hard to predict, resulting in an uneven edge, as shown in Figure 5.1-(c). However, our model predicts the mask closer to the ground truth with a smooth boundary as shown in Figure 5.1-(d). Like reverse attention in PraNet (Fan et al., 2020), we apply this module in parallel on the top of the last three levels of the feature map along with deep supervision.

The encoder is considered the backbone network in the segmentation task, which extracts the row fine level to coarse level features and is further processed by various small architecture modules to enhance the feature representation. Therefore, to observe the impact of various backbone architecture types, we embed our module in three different networks: Res2Net (Gao et al., 2019), ConvNext (Liu et al., 2022), and PVT (Wang et al., 2021), and compare the performance. Our result shows that the proposed module with Res2Net as a backbone network significantly outperforms PraNet with the same backbone (Fan et al., 2020) on various polyp segmentation datasets, including Kvasir(Jha et al., 2020), CVC-ClinicDB(Bernal et al., 2015), CVC-ColonDB (Tajbakhsh et al., 2015c), CVC-300, and ETIS(Silva et al., 2014b). In addition, our attention module with the PVT as a backbone network achieves state-of-the-art accuracy on CVC-ClinicDB, CVC-ColonDB and ETIS by exceeding the performance of recently proposed SA-Net, TransFuse, and Polyp-PVT.

The main contributions of this work are summarized below:

1. We propose a novel attention module, named Fuzzy attention, to focus more on the difficult

pixels which usually lie near the boundary region. It can be embedded in any backbone network in parallel after the last three feature maps.

2. We investigate the impact of various types of backbone networks: Res2net (Gao et al., 2019), ConvNext (Liu et al., 2022), and PVT(Wang et al., 2021), along with our attention module through extensive experiments.

3. Extensive experimental results show that our Fuzzy attention module outperforms the reverse attention-based model, PraNet (Fan et al., 2020), by a significant margin with the same Res2Net backbone on the polyp segmentation datasets: Kvasir, CVC-ColonDB (Tajbakhsh et al., 2015c), CVC-ClinicDB (Bernal et al., 2015), CVC-300 (Vázquez et al., 2017). With PVT (Wang et al., 2021) as a backbone network, we achieve state-of-the-art accuracy on the CVC-ClinicDB, CVC-ColonDB, and ETIS datasets.

5.2 Related Work

Various approaches have been proposed to segment the polyp in colonoscopic images using either handcrafted features or deep features extracted by deep learning networks. All approaches can be broadly divided into two categories:

Classical computer vision approaches: Early polyp segmentation approaches use low-level handcrafted features, including texture (Mamonov et al., 2014) and geometric features. (Maghsoudi, 2017) used a simple linear iterative clustering superpixel to segment the polyp. As mentioned above, all the method has a high false detection rate because of the high similarity between polyps and the surrounding area.

Deep learning based approaches: There have been various deep learning-based approaches proposed for the polyp segmentation task. It started with the study (Akbari et al., 2018b), which first employs a convolutional network for the polyp segmentation and outperforms the traditional methods. The U-shaped encoder and decoder architectures: U-Net (Ronneberger et al., 2015), U-Net++ (Zhou et al., 2018), ResUNet++ (Jha et al., 2019), ACS-Net (Zhang et al., 2020a) and

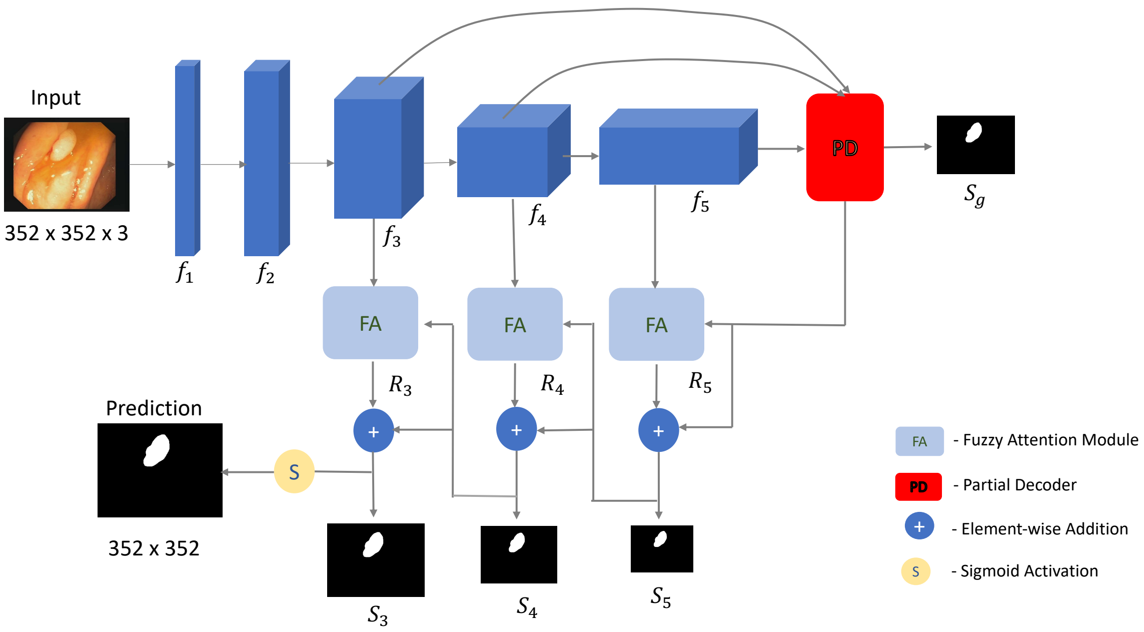


Figure 5.2: Overall architecture of FuzzyNet. It includes a partial decoder denoted as "PD" in red block and a series of fuzzy attention modules denoted as "FA" blocks. It generates the global map from the partial decoder and passes through the series of fuzzy attention modules which focuses on the difficult and fuzzy boundary pixels. The deep supervision is applied at the end of the output of each fuzzy attention module and partial decoder.

Enhanced U-Net (Patel et al., 2021) started dominating the segmentation field because of their tremendous performance gain.

To alleviate the issues of complex boundary regions, SFA (Fang et al., 2019) and PSI (Murugesan et al., 2019) include an extra edge supervision branch. However, it requires extra boundary annotation and has an overfitting problem. PraNet (Fan et al., 2020) introduces the reverse attention mechanism to mine the boundary region gradually by focusing more on background pixels. In contrast, to reverse attention, our module focuses more on the complex pixels usually lying around the boundary region. ACS-Net (Zhang et al., 2020a) also introduces the attention mechanism to focus more on the hard pixels, however, it employs a predefined pixel score to classify the pixel as hard or easy.

Other attention-based models, SA-Net (Wei et al., 2021) and Enhanced U-Net (Patel et al.,

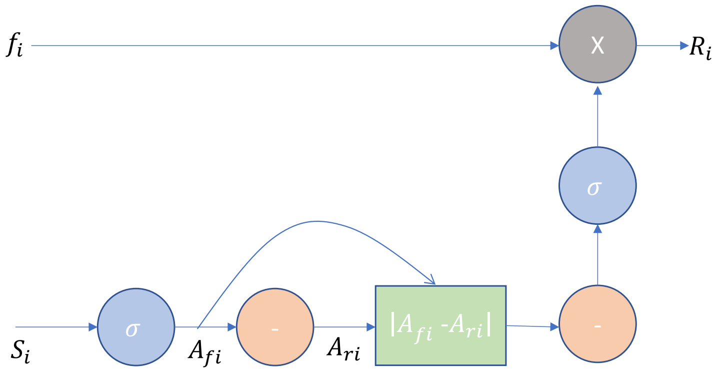


Figure 5.3: The block diagram of the Fuzzy attention module. It takes the S_i as input and passes it through the Sigmoid and the reverse block to calculate the forward and reverse attention map. The absolute difference between these attention maps is calculated followed by the reverse attention and sigmoid activation. $'-'$ represents the $1-X$ operation, where X is the input.

2021) use different attention mechanisms to give more attention to the foreground region and adaptively select the features. TranFuse (Zhang et al., 2021a) and Polyp-PVT (Dong et al., 2021) use the latest vision transformer for the polyp segmentation task and achieve an excellent result. The attention mechanism has also been successfully applied in many other applications (Gajurel et al., 2021)(Ma et al., 2021)(Sajid et al., 2021b)(Sajid & Wang, 2021). In one of our experiments, we also embedded our attention module in Pyramid Vision Transformer and established state-of-the-art accuracy on various datasets (Patel et al., 2022a).

5.3 Method

The overview of our proposed network is shown in Figure 6.1. We follow the architecture used in the PraNet (Fan et al., 2020) and replace the reverse attention module in PraNet with our proposed Fuzzy attention module. Specifically, our model takes the RGB image as an input and passes it through the backbone network, followed by the partial decoder, which employs multi-resolution feature maps to generate the initial global semantic map. This global map is then passed through a series of fuzzy attention modules, which gradually mine the boundary cues. We apply deep supervision after each attention module and the initial global map. The map generated by the last attention layer is considered the final prediction map. A detailed explanation of each element of the architecture is elaborated below.

5.3.1 Backbone Network

In the segmentation task, the encoder is considered as the backbone network, which generates the essential row multi-resolution features from fine level to abstract $\{f_i, i = 1, ...5\}$. It is considered the heart of the segmentation model because the model performance heavily relies on the features generated by it. Therefore, to observe the impact of various types of backbone networks (either convolution-based or transformer-based), we use three different networks in our experiments: Res2Net (Gao et al., 2019), ConvNext (Liu et al., 2022), and PVT (Wang et al., 2021).

5.3.2 Partial Decoder

As mentioned in the previous section, the encoder generates five levels of multi-resolution feature maps $f_i, i = 1, ...5$. These feature maps are further divided into two types: low level $\{f_i, i = 1, 2\}$ and high level $\{f_i, i = 3, 4, 5\}$. As observed in (Wu et al., 2019), low-level features increase the computation cost by a large amount and have less contribution towards increasing the performance. Therefore, we employ the parallel partial decoder proposed in (Wu et al., 2019), which aggregates only the high-level features to generate the global initial semantic map, and it is further refined in attention modules.

5.3.3 Fuzzy Attention Module

The human's natural tendency is to roughly locate the object and then gradually mine the complex region by concentrating more on that area. We apply a similar approach for medical image segmentation to clearly distinguish the background area and foreground objects (polyp or skin lesion) by focusing more on the hard pixels using the Fuzzy attention module. We apply this module parallelly on the high-level feature maps $\{f_i, i = 3, 4, 5\}$ which produce the resultant feature map

$\{R_i, i = 3,4,5\}$, where R_3 is used to generate the final prediction map. The block diagram of the fuzzy attention module is shown in Figure 6.2

Specifically, the resultant feature maps are calculated as: $\mathrm{R}_i = f_i.A_i$

The attention maps A_i have high scores for difficult pixels and low scores for easy pixels. It is mathematically formulated as: $\mathrm{A}_i = \sigma(1- \mid A_{fi} - A_{ri} \mid)$ where. A_{fi} represents the forward attention map, with a high score for the foreground object and a low score for the background area, and A_{ri} indicates the reverse attention map, which has a high score for the background pixels and low score for the foreground pixels. The attention maps can be mathematically formulated as: $\mathrm{A}_{fi} = \sigma(Up(S_{i+1}))$ $\mathrm{A}_{ri} = 1 - \sigma(Up(S_{i+1}))$ where Up indicates the upsampling operation, $\sigma(.)$ represents the sigmoid activation, and S_i is the global map from the previous layer. Pixel's difficulty can be associated with the absolute difference between the forward and reverse attention score; the lower the absolute difference, the higher the difficulty. To focus more on the complex pixels, we further subtract the absolute difference from 1 followed by sigmoid activation as shown in equation 2.

5.3.4 Loss Function

We use the combination of weighted IoU loss L^w_{IOU} and weighted cross-entropy loss L^w_{BCE} as our main loss function (Qin et al., 2019)(Wei et al., 2020). We apply deep supervision after each resultant map generated by the attention module along with the initial global map. The total loss can be formulated as: $\mathrm{L}_{total} = L(G, S^{up}_g) + \sum_{i=3}^{5} L(G, S^{up}_i)$ where S_g is the global map and S_3, S_4, and S_5 are the output maps generated by the attention module.

5.4 Experiments

5.4.1 Datasets

We conducted experiments on five publicly available polyp segmentation datasets: ETIS, CVC-ClinicDB, CVC-ColonDB, CVC-300, and KVasir. ETIS is an old dataset with 196 polyp images and its ground truth mask. CVC-ClinicDB and CVC-300 comprise 612 and 300 images from 29 and 13 colonoscopy video sequences, respectively. CVC-ColonDB is a small-scale dataset containing 380 images from 15 short colonoscopy sequences. Kvasir dataset is relatively new, with 1000 polyp images. We compare our FuzzyNet with state-of-the-art models: PraNet, Enhanced U-Net, ACSNet, MSEG(Huang et al., 2021), SA-Net, TransFuse, and Polyp-PVT, along with the previous approaches U-Net, U-Net++, and ResU-Net++.

5.4.2 Evaluation Metrics

We utilize the Dice coefficient and Intersection over Union (IOU) as our evaluation metrics which are defined below:

Dice coefficient: It is defined as:

$$DSC(A,B) = \frac{2 \times (A \cap B)}{A \cup B} \tag{5.1}$$

where A denotes the predicted set of pixels and B is the ground truth of the image.

Intersection over union (IoU): It is another standard metric to evaluate the performance of the segmentation task. It is defined as:

$$IOU(A,B) = \frac{A \cap B}{A \cup B} \tag{5.2}$$

where A denotes the predicted set of pixels and B is the corresponding ground truth of the set of pixels.

5.4.3 Implementation Details

In our experiments, we follow the same training settings used in PraNet for the Res2Net backbone and Polyp-PVT for ConvNext and PVT backbone. All the models are trained on a V100 GPU, with batch-size 16 and Adam optimizer with an initial learning rate of 0.0001. We employ multi-

model	CVC-ClinicDB		Kvasir	
	mDice	mIoU	mDice	mIoU
U-Net	0.823	0.755	0.818	0.746
U-Net++	0.794	0.729	0.821	0.743
SFA	0.700	0.607	0.723	0.611
ACSNet	0.882	826	0.898	0.838
PraNet	0.899	0.849	0.898	0.840
EU-Net	0.902	0.846	0.908	0.854
SA-Net	0.916	0.859	0.904	0.847
TransFuse	0.918	0.868	0.918	0.868
Polyp-PVT	0.937	0.889	0.917	0.864
Fuzzy-Net(Res2Net)	0.919	0.867	0.889	0.830
Fuzzy-Net(ConvNext)	0.922	0.863	0.907	0.848
Fuzzy-Net(PVT)	**0.937**	**0.889**	0.913	0.864

Table 5.1: Results on CVC-ClinicDB and Kvasir, which represent the learning capability of the model. It shows that our model outperforms the other models by a significant margin on the CVC-ClinicDB dataset and achieves a comparable result on the Kvasir dataset. The reported result is the average of three experiments.

scale training for all the backbone networks instead of data augmentation techniques by following the PraNet and PVT. We employ the backbone networks Res2Net, ConvNext, and Pyramid Vision Transformer, initialize the weights with pretrained weights trained on ImageNet-1K and train them from scratch.

5.4.4 Learning Ability

Setting: We evaluate the learning ability of our model on the dataset ClinicDB and Kvasir-Seg. Clinic-DB consists of 612 images extracted from 31 colonoscopy videos, whereas Kvasir-Seg consists of a total of 1000 polyp images. We follow the same setting as PraNet and Polyp-PVT, which include 900 and 548 images from ClinicDB and Kvasir-Seg datasets as the train set, and the remaining 64 and 100 images are used as the test set.

Results: Table 6.1 shows the result on Kvasir and CVC-ClinicDb datasets. Our FuzzyNet model with Res2Net as a backbone network achieves 2% higher mean dice than PraNet on CVC-ClinicDB and achieves comparable results on the Kvasir dataset which demonstrates the better learning ability of our model. Our model with ConvNext as a backbone network outperforms the ACSNet, PraNet, EU-Net, and SA-Net on the Clinic-DB dataset by 3.4%, 2.3%, 2%, and 0.6%, respectively, in terms of mean-dice. In addition, it also achieves 0.9%, 0.9%, and 0.03% higher mean dice than the ACSNet, PraNet, and SANet, respectively, on the Kvasir dataset. With Pyramid Vision Transformer as a backbone network, we achieve the state-of-the-art accuracy 0.937 on the CVC-ClinicDB dataset and comparable mean dice on the Kvasir dataset as Polyp-PVT.

5.4.5 Generalization Ability

Setting: To evaluate the generalization ability of the model, we use three unseen datasets: ETIS, ColonDB, and CVC-300. The ETIS, ColonDB, and CVC-300 datasets consist of a total of 190, 380, and 60 images, respectively. The images of these datasets belong to different medical centers, which means that the training and testing sets are different and the model has not seen the test images before during training.

model	CVC-ColonDB		ETIS		CVC-300	
	mDice	mIoU	mDice	mIoU	mDice	mIoU
U-Net	0.512	0.444	0.398	0.335	0.710	0.627
U-Net++	0.483	0.410	0.401	0.344	0.707	0.624
SFA	0.469	0.347	0.297	0.217	0.467	0.329
ACSNet	0.716	0.649	0.578	0.509	0.863	0.787
PraNet	0.712	0.640	0.628	0.567	0.871	0.797
EU-Net	0.756	0.681	0.687	0.609	0.837	0.765
SA-Net	0.753	0.670	0.750	0.654	0.888	0.815
TransFuse	0.773	0.696	0.733	0.659	**0.902**	**0.833**
Polyp-PVT	0.808	0.727	0.787	0.706	0.900	0.833
Fuzzy-Net(Res2Net)	0.739	0.662	0.731	0.658	0.894	0.825
Fuzzy-Net(ConvNext)	0.784	0.696	0.740	0.648	0.877	0.795
Fuzzy-Net(PVT)	**0.811**	**0.728**	**0.791**	**0.702**	0.891	0.818

Table 5.2: Results on CVC-ColonDB, ETIS, and CVC-300 demonstrate the generalization capability of the model. It shows that our model outperforms the other models by a significant margin on CVC-ColonDB and ETIS and achieves a comparable result on the CVC-300 dataset. The reported result is the average of three experiments.

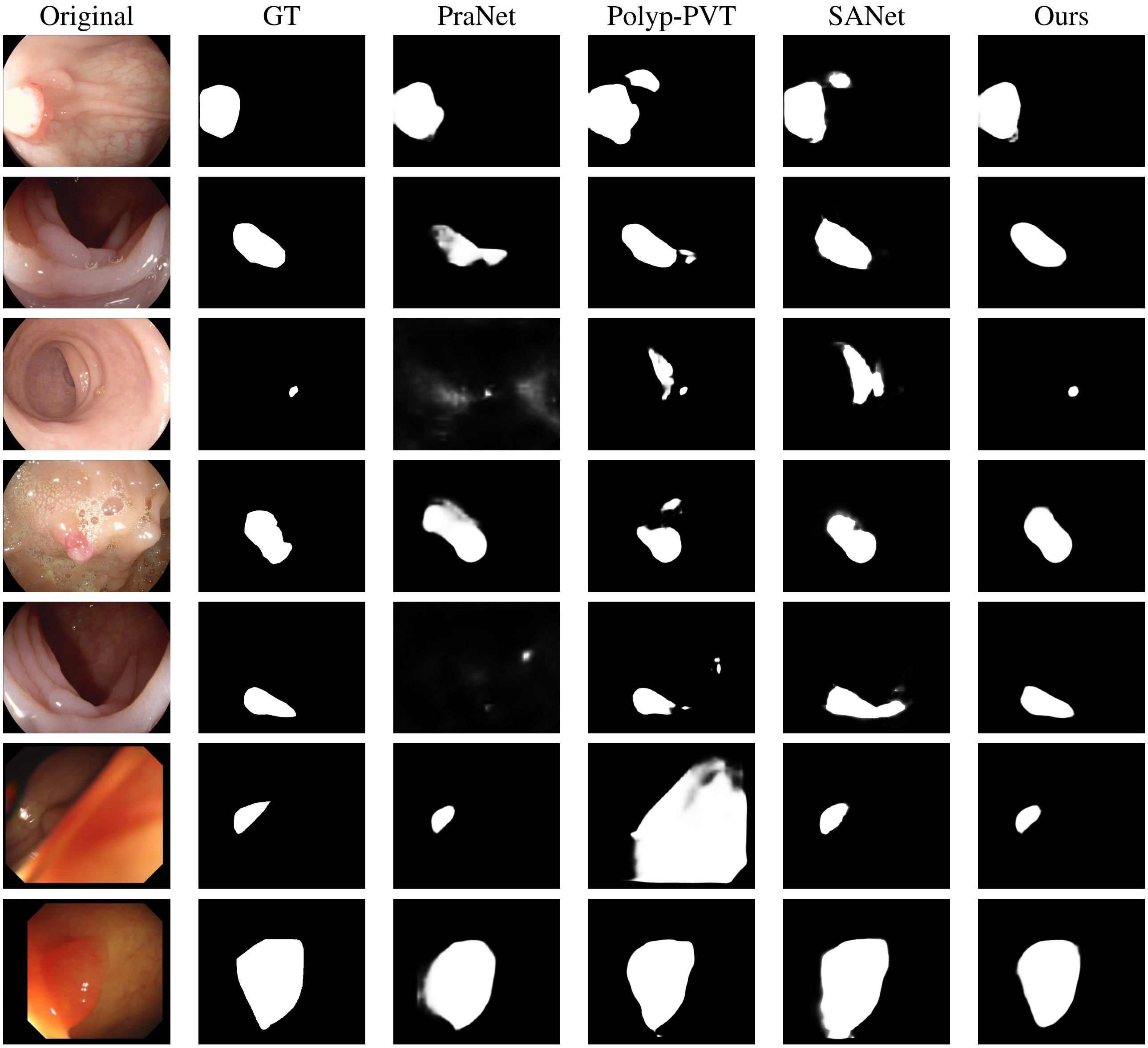


Figure 5.4: A comparison of the segmentation maps generated by our model, PraNet, Polyp-PVT and SA-Net along with the original image and ground-truth mask. Our model has more sharp boundaries than PraNet and the maps are closer to the ground-truth mask compared to other methods. The images are taken from ETIS and CVC-ColonDB datasets.

Results: The result is shown in the table 6.2. It can be seen from the result that our model has a better generalization performance compared to state-of-the-art models. On ColonDB and ETIS, we achieved the highest performance by outperforming the Polyp-PVT. On CVC-300, our model achieves comparable performance to Polyp-PVT. In addition, our model with Res2Net as a backbone outperforms PraNet by 3.1%, 2.7%, and 10% on CVC-300, CVC-ColonDB, and ETIS respectively in terms of mean dice, which demonstrates that Fuzzy Attention module has an outstanding generalization ability compared with the Reverse Attention module in PraNet. With PVT as a backbone network, our model outperforms the latest SANet and TransFuse by 5.8% and 6.7% on CVC-ColonDB, 4.2% and 5.5% on ETIS respectively. It also achieves 1.1% higher mean dice than SA-Net and comparable mean dice as TransFuse on CVC-300.

5.4.6 Effectiveness of Fuzzy Attention

The effectiveness of the Fuzzy attention module can be verified by comparing the result of our model with the Res2Net backbone with PraNet. For a fair comparison, apart from the type of attention module, we utilize a similar backbone, hyperparameters, augmentation, and regularization technique as used in PraNet.

It can be seen from the table our model outperforms PraNet on the CVC-ClinicDB dataset by 2% and achieves a comparable result on the Kvasir dataset. It improves the result on unseen datasets by a significant margin of 2.7%, 10%, and 1.9% on CVC-ColonDB, ETIS, and CVC-300 respectively. The overall result of all datasets along with GFlops and the total number of parameters are listed in Table 5.3. We can observe a significant performance gain of 0.9% and 5% on the average of seen and unseen datasets respectively. The outstanding results on the unseen dataset prove the generalization capability of the proposed fuzzy attention, which is significantly higher than reverse attention.

Furthermore, fuzzy attention does not add an extra number of parameters and the computation cost. With the same amount of parameters and GFlops: 32M and 13.11 GMac respectively, we achieve a significant performance gain compared to reverse attention-based PraNet.

In addition, we visualize the segmentation mask generated by our model and PraNet. It can be seen from the result shown in Figure 5.4 that our model correctly classifies the fuzzy pixels near the boundary compared to the PraNet model. For most of the images, PraNet seems to have misclassified background pixels as foreground pixels because of the high focus on the background

model	Attention	mDice		mIoU		GFlops	Parameters
		Seen	Unseen	Seen	Unseen		
Pra-Net	Reverse	0.8985	0.737	0.8445	0.668	**13.11**	**32.55M**
Fuzzy-Net	Fuzzy	**0.904**	**0.788**	**0.845**	**0.715**	**13.11**	**32.55M**

Table 5.3: Comparison of the GFlops and the number of parameters of reverse attention-based Pra-Net and fuzzy attention-based FuzzyNet along with average mean dice and mean IoU calculated by taking the average of mean dice and mean IoU of all datasets included in type seen and unseen, respectively.

pixels in reverse attention. In contrast to PraNet, the resultant mask generated by our model has well-defined boundaries, and the results are closer to the ground truth. In addition, our model's performance is consistent irrespective of the lighting and reflection conditions in the image. Furthermore, we also visualize the segmentation mask generated by Polyp-PVT and SA-Net. It can be seen from the figure that our resultant mask has fewer false positive pixels and a smooth boundary than all other methods.

5.5 Conclusion

This paper has presented a novel attention mechanism to encourage the network to focus more on the fuzzy region, which usually lies around the boundary. We embed our attention module with various backbone networks: Res2Net, ConvNext, and Pyramid Vision Transformer (PVT) for polyp segmentation. Our result shows that the fuzzy attention module significantly outperforms PraNet, which employs the reverse attention mechanism on all polyp segmentation datasets. With the PVT as the backbone network, our model achieves state-of-the-art accuracy on the CVC-ClinicDB, CVC-ColonDB, and ETIS datasets for polyp segmentation.

Chapter 6

Multi-Layer Dense Attention Decoder for Polyp Segmentation

Abstract

Detecting and segmenting polyps is crucial for expediting the diagnosis of colon cancer. This is a challenging task due to the large variations of polyps in color, texture, and lighting conditions, along with subtle differences between the polyp and its surrounding area. Recently, vision Transformers have shown robust abilities in modeling global context for polyp segmentation. However, they face two major limitations: the inability to learn local relations among multi-level layers and inadequate feature aggregation in the decoder. To address these issues, we propose a novel decoder architecture aimed at hierarchically aggregating locally enhanced multi-level dense features. Specifically, we introduce a novel module named Dense Attention Gate (DAG), which adaptively fuses all previous layers' features to establish local feature relations among all layers. Furthermore, we propose a novel nested decoder architecture that hierarchically aggregates decoder features, thereby enhancing semantic features. We incorporate our novel dense decoder with the PVT backbone network and conduct evaluations on five polyp segmentation datasets: Kvasir, CVC-300, CVC-ColonDB, CVC-ClinicDB, and ETIS. Our experiments and comparisons with nine competing segmentation models demonstrate that the proposed architecture achieves state-of-the-art performance and outperforms the previous models on four datasets. The source code can be available at: https://github.com/krushi1992/Dense-Decoder

6.1 Introduction

Polyp segmentation plays a critical role in expediting the diagnosis of colorectal cancer, a disease recognized as one of the most prevalent cancers globally (Mathur et al., 2020)(Patel et al., 2021)(Silva et al., 2014b). Detecting polyps at an early stage can significantly reduce the mortality rate. Colonoscopy is an effective technique for colorectal cancer (CRC) screening, identifying polyps that may lead to colon cancer. However, the challenge arises from the similarity in appearance between polyps and background pixels, making it difficult even for experienced clinicians to discern and potentially resulting in missed detections (Patel et al., 2020)(Li et al., 2021a). Moreover, polyps exhibit wide variations in size, texture, and color. Therefore, an accurate and automated polyp segmentation method is imperative for the early detection of cancerous polyps, to decrease the mortality rate (Jia et al., 2019).

Convolutional neural networks (CNNs) have been extensively employed for polyp segmentation (Akbari et al., 2018b)(Brandao et al., 2017)(He et al., 2021)(Jha et al., 2019)(Ronneberger et al., 2015)(Zhou et al., 2018). Specifically, various U-Net-shaped encoder-decoder-based segmentation networks (Jha et al., 2019) (Ronneberger et al., 2015)(Zhou et al., 2018) have demonstrated remarkable performance gains by generating high-resolution segmentation and aggregating multi-stage features through skip connections. However, they still struggle to establish long-range dependencies essential for accurate polyp segmentation. To address this limitation, some works incorporate attention-based modules into the segmentation architecture (Patel et al., 2021)(Patel et al., 2022b)(Wei et al., 2021)(Zhang et al., 2020a), leading to performance improvements. Nevertheless, these approaches still fall short of fully capturing long-range dependencies.

Recently, vision Transformer-based encoders (Dong et al., 2021)(Zhang et al., 2021a) have gained popularity and been successfully applied in polyp segmentation tasks due to their capability to capture long-range dependencies. Vision Transformers use an attention-based module to learn correlations among spatial patches, enabling them to capture the global context. However, vision Transformers are computationally expensive. To reduce the computational cost, hierarchical vision Transformers with modified attention modules have been proposed, such as PVT (Wang et al., 2021) and Swin Transformer (Liu et al., 2021). These hierarchical vision Transformers have achieved significant performance gains in various computer vision-related tasks. However, we believe that Transformer-based segmentation networks fail to establish relationships among neighboring pixels.

Various segmentation networks such as polyp-PVT (Dong et al., 2021) have attempted to address the aforementioned challenges by embedding convolution layers in the decoder network. However, these approaches primarily establish local relationships, leading to two main issues: (i) They do not fully exploit the multi-level features from previous layers during local feature modeling, which are crucial for identifying fine-grained clues. (ii) They lack hierarchical feature flows in the decoder to enhance the local feature relationships of global features progressively captured by the Transformer encoder. To resolve these issues, we propose a novel decoder network with dense connections and hierarchical feature flow. Specifically, we introduce a module called the dense attention gate, which considers all previous layers' decoder features via dense connections followed by an attention mechanism. Furthermore, we propose a novel multi-layer decoder to further refine the local features by hierarchically aggregating them, thereby improving feature flow.

The main contributions of this paper are summarized below:

1. **Dense Attention Gate:** We propose a novel module called the dense attention gate, which considers all the previous layers' encoding layer features to calculate spatial attention scores. These scores are then broadcastly multiplied by the current encoding layer features, rather than directly fusing the encoding features to the respective decoding layer.

2. **Hierarchical multi layer decoder:** We introduce a hierarchical decoding layer to enhance the flow of features, refining local features through the horizontal extension of decoding layers. This involves utilizing the output of the previous decoding layers as inputs until a singular feature map is obtained.

3. **Improved performance:** Comprehensive experiments demonstrate that the integration of the dense attention gate module, along with the hierarchical multi-layer decoder, achieves supe-

rior performance across various polyp segmentation datasets and outperforms most other approaches.

6.2 Related Work

There have been various approaches proposed to segment the polyp from the colonoscopic images. Based on the fundamental component used, they mainly categorized into three categories:

Traditional Computer Vision Approach: Before the emergence of neural networks, polyp segmentation predominantly relied on manually crafted features such as size, shape, texture, and color (Mamonov et al., 2014). In the seminal study (Maghsoudi, 2017), a simple linear iterative clustering superpixel method was proposed for polyp segmentation. However, these traditional approaches suffer from slow processing speeds and a high misdetection rate. This is mainly due to the limited representational capacity of hand-crafted features and the significant similarity between polyps and their surrounding areas.

CNN-based Deep Learning Approach: Following the success of convolutional neural networks in various computer vision tasks, researchers have turned to CNN-based networks for polyp segmentation. This trend began with the work of (Akbari et al., 2018b), where a modified version of Fully Convolutional Network (FCN) was employed to segment polyps, outperforming classical computer vision-based methods. Subsequently, U-Net (Ronneberger et al., 2015) introduced a U-shaped encoder-decoder architecture, significantly improving performance. Building upon this success, various U-shaped models such as U-Net++ (Zhou et al., 2018) and ResUNet++ (Jha et al., 2019) have been proposed, further enhancing polyp segmentation performance.

Although these networks achieved remarkable performance, they often struggle to differentiate between the polyp boundary and the surrounding area due to unfiltered encoded features. Furthermore, U-shaped encoder-decoder architectures typically employ a single-layer decoder, which we believe is insufficient for further improving decoded features. To address this limitation and enhance feature flow, we propose a horizontally extended multi-layer decoder to refine local features further. To address the boundary issue, various attention-based models such as PraNet (Fan

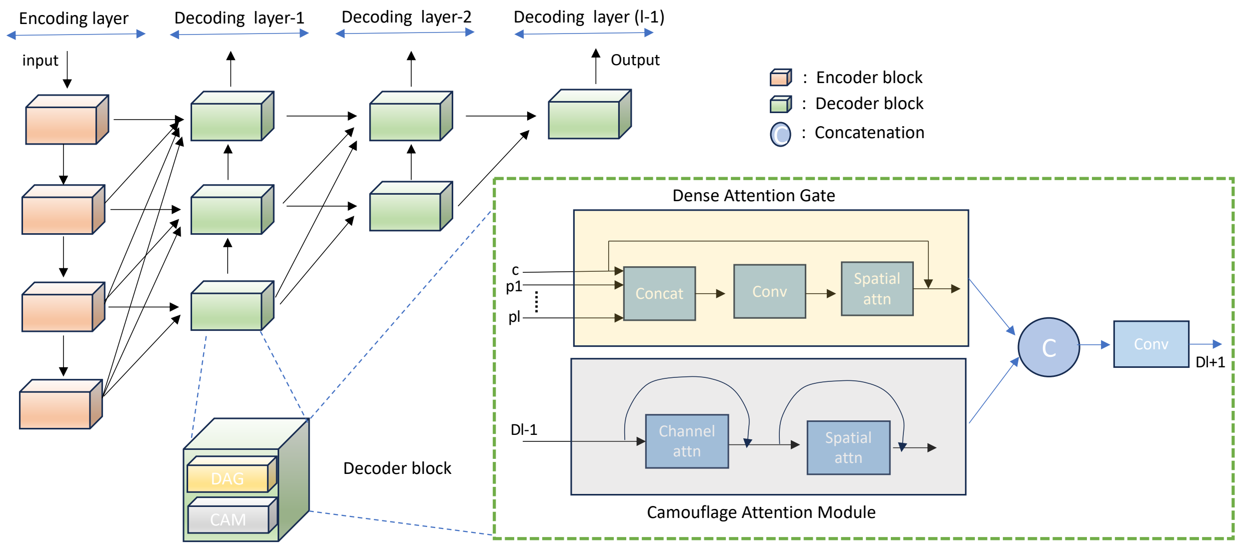


Figure 6.1: The overall architecture of the multi-layer dense decoder-based polyp segmentation network. It comprises a PVT encoder represented by a collection of orange blocks. The dense decoder consists of multiple horizontally stacked decoding layers, each composed of a series of decoding blocks interconnected, denoted as green blocks. Each decoding block encompasses two modules: DAG (Dense Attention Gate) and CAM (Camouflage Attention Module). The block diagrams of both modules are depicted in the figure outlined within the green box.

et al., 2020), ACSNet (Zhang et al., 2020a), Enhanced U-Net (Patel et al., 2021), and SANet (Wei et al., 2021) have been proposed. These models aim to suppress insignificant features and enhance important ones, thereby improving boundary prediction. However, none of them consider all the previous encoding layer features when calculating the attention score.

This paper proposes a novel module, named Dense Attention Gate, by taking into account all the previous layers' encoding features while calculating the attention score. This is essential for capturing multi-scale and fine-grained features, which can significantly improve boundary delineation.

Vision Transformer based Deep learning Approach: Vision Transformers have garnered significant attention and adoption in various computer vision tasks (Chen et al., 2023)(Wang et al., 2023)(Xiao et al., 2023). Building upon their success, researchers have begun integrating them as backbone networks in polyp segmentation tasks, as demonstrated by models such as Polyp-PVT (Dong et al., 2021), TransFuse (Zhang et al., 2021a), and FuzzyNet (Patel et al., 2022b). This integration has led to notable performance improvements attributed to the establishment of long-range dependencies. However, despite these advancements, existing approaches still struggle to establish fully comprehensive neighborhood relationships between pixels.

In our work, we propose the integration of a dense attention gate mechanism to refine local features. Additionally, our design incorporates a hierarchical multi-layer decoder to further enhance local features within the global context generated by the Transformer-based encoder. This comprehensive approach aims to address the limitations of existing methods and further improve the accuracy of polyp segmentation.

6.3 Method

Our proposed network architecture consists of two main components as shown in Fig. 6.1: the Transformer-based Encoder and the Dense Decoder. The model takes an RGB colonoscopic image as input and passes it through a series of attention layers within the PVT encoder. This process generates feature maps at each encoding layer's stage. Subsequently, these feature maps are fed into the corresponding stages of the decoding layers in our dense decoder. The dense decoder comprises a series of decoding layers stacked horizontally. Specifically, our model consists of three stacked decoding layers, each decoding layer consists of a series of decoding blocks composed of dense attention gates and camouflage attention modules arranged vertically. Below, we provide a detailed description of each component for clarity.

6.3.1 Transformer – Encoder

In segmentation tasks, the encoder serves as the backbone of the network, generating fundamental multi-resolution features at each stage of refinement, denoted as $f_i, i = 1,2,..4$. It is considered the cornerstone of the model architecture, as the model heavily relies on the encoding features. While convolutional neural network-based encoding models are widely used for polyp segmentation tasks, they often struggle to generate global features essential for segmentation. Therefore, we employ a Transformer-based encoder, whose main component is the self-attention mechanism.

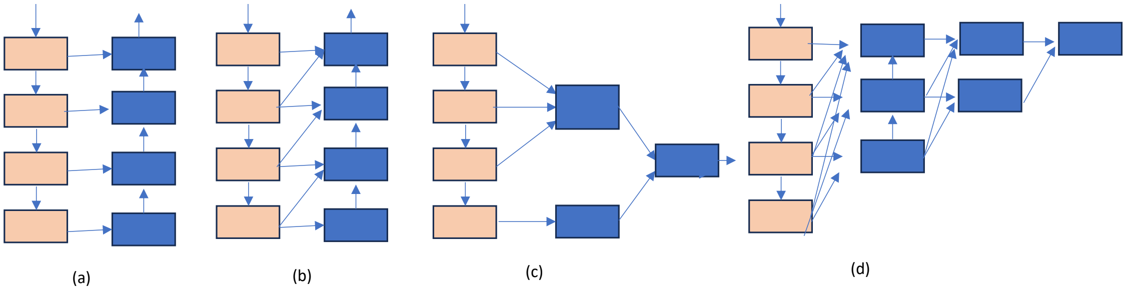


Figure 6.2: Different types of decoding structures for the polyp segmentation. (a) U-shaped decoding structure (Ronneberger et al., 2015). (b) U-shaped decoding structure with previous layer integration strategy (Patel et al., 2021). (c) Separate decoding for low-level and high-level features (Dong et al., 2021). (d) multi-layer dense decoder architecture proposed on our model.

This mechanism establishes long-range dependencies between all pixels and is capable of generating global features after each stage of processing.

6.3.2 Multi-layer Dense Decoder

To fully leverage the global features generated by the Transformer encoder, we introduce a novel decoder architecture called the Multi-layer Dense Decoder as shown in Fig. 6.1. It consists of $l-1$ decoding layers stacked together, with each decoding layer composed of a series of decoding blocks stacked vertically. Each decoding block comprises a dense attention gate followed by the camouflage attention module. Specifically, the main three components of the multi-layer dense decoder include Dense Attention Gate, Camouflage Attention Module, and Multi-layer Decoder Design. The description of each component is as follows:

Dense Attention Gate: In our approach, we replace the skip connection with the Dense Attention Gate, which considers all previous encoding block features when calculating the attention score. This differs from previously proposed attention modules, which typically only use current layer features. Specifically, we calculate the dense attention score by first concatenating the current encoding block features with all the previous encoding block features. Then, we apply a spatial attention block, which represents the significance of each pixel in the feature map. During the concatenation operation, we first upsample the previous layer features to match the size of the current layer feature map.

$$C_i = Concat(E_i, ...E_{l-i}) \tag{6.1}$$

$$S_i = \sigma(Conv(C_i)) \tag{6.2}$$

$$D_i = E_i * S_i \tag{6.3}$$

where E_i is the current encoding block features and l is the total number of blocks in the encoding layer. Here we have $l = 4$. C_i is the concatenated features of encoding block i and its previous layers. S_i is the spatial attention score calculated by applying a convolution operator followed by sigmoid activation. S_i is then multiplied by the current encoding block features which are then used as input to the Camouflage Attention Module.

Camouflage Attention Module: To further enhance the distinction between polyp features and background elements, we incorporated the Camouflage Identification Module proposed in Polyp-PVT within each decoding block following the dense attention gate. In contrast to Polyp-PVT, where the CAM is applied only once on the low-level features, we integrate the CAM into each block of our decoder.

The Camouflage Identification Module mainly consist of the channel attention mechanism $A_c(.)$ and spatial attention mechanism $A_s(.)$, which can be formulated as:

$$D_{ci} = A_s(A_c(D_i)) \tag{6.4}$$

$$A_c(D_i) = \sigma(\beta_1(P_{max}(D_i)) + \beta_2(P_{avg}(D_i))) \tag{6.5}$$

$$A_c(D_i) = A_c(D_i) * D_i \tag{6.6}$$

where D_i is the output of the Dense attention gate. σ is the softmax activation. P_{max} and P_{avg} denotes adaptive maximum pooling and adaptive average pooling respectively. β_1 and β_2 are convolution operators of size 1×1 to reduce the channel dimension 16 times, followed by a ReLU layer and another 1×1 convolutional layer to recover the original channel dimension. The spatial attention A_s can be formulated as:

$$A_s(D_i) = \sigma(\alpha(Concat(R_{max}(D_i), R_{avg}(D_i)))) \tag{6.7}$$

$$A_s(D_i) = A_s(D_i) * D_i \tag{6.8}$$

where R_{max} and R_{avg} represent the maximum and average values obtained along the channel dimension, respectively. α represent the convolution operation with 7×7 kernel and padding 3.

Multi-layer Decoder Design: To enhance the differentiation between polyps and their background while refining local features, we expanded the decoding layers horizontally. A visual comparison between our decoder design and previous approaches is illustrated in Fig. 6.2. Specifically, instead of a single decoding layer, we incorporated a total of $l-1$ layers, where l represents the total number of encoding blocks. The output of one decoding layer serves as the input to the subsequent decoding layer within its respective decoding block. The design of decoding blocks in subsequent layers remains consistent, comprising dense attention gates and camouflage identification modules. This approach progressively enhances the local features within the global context generated by the Transformer encoder, facilitating the differentiation of polyps. At the conclusion of each decoding layer, we apply deep supervision loss. The final output is computed by aggregating the endpoint outputs from all the decoding layers.

6.3.3 Loss Function

We formulate the loss function as a combination of weighted IOU loss L^w_{IOU} and weighted cross entropy loss L^w_{BCE} (Qin et al., 2019)(Wei et al., 2020). We apply deep supervision at the end of each decoding layer.

$$L_{main} = L_{IOU} + L_{BCE} \tag{6.9}$$

$$L_{total} = \sum_{i=1}^{l-1} L_{main}(D_i) \tag{6.10}$$

where l is the total number of encoding blocks. Our total number of decoding layers is dependent on the number of encoding blocks. Specifically, in our case we have $l = 4$, which leads to a total number of decoding layers $l - 1 = 3$.

6.4 Experiments

6.4.1 Datasets and Models

We conducted experiments on five publicly available polyp segmentation datasets: Kvasir (Jha et al., 2020), CVC-300 (Vázquez et al., 2017), CVC-ColonDB (Tajbakhsh et al., 2015c), CVC-ClinicDB (Bernal et al., 2015), and ETIS (Silva et al., 2014b). The ETIS dataset comprises 196 polyp images along with their corresponding ground truth masks. CVC-ClinicDB and CVC-300 consist of 612 and 300 images extracted from 29 and 13 colonoscopy video sequences, respectively. The CVC-ColonDB dataset is relatively small-scale, containing 380 images derived from 15 short colonoscopy sequences. The Kvasir dataset, on the other hand, is comparatively newer, consisting of 1000 polyp images. We compare our model with the following nine state-of-the-art models: PraNet (Fan et al., 2020), Enhanced U-Net (Patel et al., 2021), ACSNet (Zhang et al., 2020a), MSEG (Huang et al., 2021), SA- (Wei et al., 2021), TransFuse (Zhang et al., 2021a), Polyp-PVT (Dong et al., 2021), U-Net (Ronneberger et al., 2015), U-Net++ (Zhou et al., 2018), and ResU-Net++ (Jha et al., 2019).

6.4.2 Evaluation Metric

We employ the Dice similarity coefficient (DSC) and Intersection over Union (IOU) as our evaluation metrics, defined as follows:

$$DSC(A,B) = \frac{2 \times (A \cap B)}{A \cup B} \tag{6.11}$$

$$IOU(A,B) = \frac{A \cap B}{A \cup B} \tag{6.12}$$

Model	CVC-ClinicDB		Kvasir	
	mDice	mIoU	mDice	mIoU
U-Net	0.823	0.755	0.818	0.746
U-Net++	0.794	0.729	0.821	0.743
SFA	0.700	0.607	0.723	0.611
ACSNet	0.882	0.826	0.898	0.838
PraNet	0.899	0.849	0.898	0.840
EU-Net	0.902	0.846	0.908	0.854
SA-Net	0.916	0.859	0.904	0.847
TransFuse	0.918	0.868	0.918	0.868
Polyp-PVT	0.937	0.889	0.917	0.864
Ours	**0.939**	**0.890**	**0.919**	**0.869**

Table 6.1: The results obtained on the CVC-ClinicDB and Kvasir datasets serve as indicators of the model's learning capability. The results reveal that our model surpasses other models by a substantial margin on the CVC-ClinicDB dataset, while achieving comparable performance on the Kvasir dataset. The results represent the average of three experiments.

where A denotes the predicted set of pixels and B is the ground truth of the image. A denotes the predicted set of pixels and B is the ground truth of the image. **Dice coefficient:** It is defined as: DSC(A,B) = 2 $\times(A\cap B)_{\overline{A\cup B}}$

where A denotes the predicted set of pixels and B is the ground truth of the image.

Intersection Over Union: It is defined as: IOU(A,B) = A $\cap B_{\overline{A\cup B}}$

where A denotes the predicted set of pixels and B is the ground truth of the image.

6.4.3 Implementation Details

In our experimental setup, we adopt the training configurations consistent with those employed in Polyp-PVT. Our model is trained on one NVIDIA V100 GPU with a batch size of 16 using the Adam optimizer. We initialize the learning rate at 0.0001 and utilize multi-scale training, following the methodology outlined in PraNet instead of employing data augmentation techniques. The model initialization involves pre-trained weights trained on the ImageNet 1K dataset, with training extending across all layers. Network training spans 50 epochs for three iterations, and we report the average of the best results obtained.

Model	CVC-ColonDB		ETIS		CVC-300	
	mDice	mIoU	mDice	mIoU	mDice	mIoU
U-Net	0.512	0.444	0.398	0.335	0.710	0.627
U-Net++	0.483	0.410	0.401	0.344	0.707	0.624
SFA	0.469	0.347	0.297	0.217	0.467	0.329
ACSNet	0.716	0.649	0.578	0.509	0.863	0.787
PraNet	0.712	0.640	0.628	0.567	0.871	0.797
EU-Net	0.756	0.681	0.687	0.609	0.837	0.765
SA-Net	0.753	0.670	0.750	0.654	0.888	0.815
TransFuse	0.773	0.696	0.733	0.659	**0.902**	**0.833**
Polyp-PVT	0.808	0.727	0.787	0.706	0.900	0.833
Ours	**0.818**	**0.731**	**0.795**	**0.711**	0.886	0.815

Table 6.2: The results obtained on the CVC-ColonDB, ETIS, and CVC-300 datasets serve as indicators of the model's generalization capability. The results illustrate that our model outperforms other models by a substantial margin on the CVC-ColonDB and ETIS datasets, while achieving comparable performance on the CVC-300 dataset. The results represent the average of three experiments.

Dataset	One Layer	Two Layer	Three Layer
Setting-1	0.919	0.925	0.930
Setting-2	0.825	0.830	0.834

Table 6.3: The results show the performance of the model with different numbers of decoding layers.

6.4.4 Learning Ability

Setting: We evaluate the learning capability of our model using the ClinicDB and Kvasir-Seg datasets. ClinicDB consists of 612 images extracted from 31 colonoscopy videos, while Kvasir-Seg contains a total of 1000 polyp images. Following the settings established by PraNet and Polyp-PVT, we partition 900 and 548 images from the ClinicDB and Kvasir-Seg datasets, respectively, for training purposes. The remaining 64 and 100 images are reserved for evaluation, serving as the test set.

Results: Table 6.1 presents the performance metrics on the CVC-ClinicDB and Kvasir datasets, reflecting the learning efficacy of the model. In terms of the Dice score, our model demonstrates superiority over SFA, ACSNet, PraNet, EU-Net, SA-Net, TransFuse, and PolypPVT by 23.9%, 5.7%, 4%, 3.7%, 2.3%, 2.1%, and 0.2%, respectively on the CVC-ClinicDB dataset. Moreover, our model exhibits enhanced performance on the Kvasir dataset, surpassing SFA, ACSNet, PraNet, EU-Net, SA-Net, TransFuse, and Polyp-PVT by 19.6%, 2.1%, 2.1%, 1.1%, 1.5%, 0.1%, and 0.2%, respectively, in terms of the Dice score. The notable achievements on both the CVC-ClinicDB and Kvasir datasets underscore the robust learning capabilities of our model.

6.4.5 Generalization Ability

Setting: To assess the generalization capacity of the model, we utilize three previously unseen datasets: ETIS, ColonDB, and CVC-300. The ETIS dataset comprises a total of 190 images, while ColonDB consists of 380 images, and CVC-300 contains 60 images. These datasets encompass images sourced from various medical centers, implying that the training and testing sets are distinct, and the model has not encountered the test images during the training phase.

Results: The performance evaluation of our model on the CVC-ColonDB, ETIS, and CVC-300 datasets is shown in Table 6.2. The results indicate that our model achieves state-of-the-art performance on the CVC-ColonDB dataset, surpassing SFA, ACSNet, PraNet, EU-Net, SA-Net, TransFuse, and Polyp-PVT by 34.9%, 10.2%, 10.6%, 6.2%, 6.5%, 4.5%, and 1.0%, respectively. Similarly, on the ETIS dataset, our model outperforms SFA, ACSNet, PraNet, EU-Net, SA-Net, TransFuse, and Polyp-PVT by 49.8%, 21.7%, 16.7%, 10.8%, 4.5%, 6.2%, and 0.8%, respectively, in terms of the Dice score. On the CVC-300 dataset, our model surpasses SFA, ACSNet, PraNet, and EU-Net by 41.9%, 2.3%, 1.5%, and 4.9%, respectively, in terms of the Dice score, and achieves comparable results with Polyp-PVT. The consistent performance across all three datasets demonstrates the robust generalization ability of our model.

6.4.6 Ablation Study

Effect of Multi-layer Decoder: We performed an ablation study to evaluate the impact of the number of decoding layers on the performance metrics of learning ability (setting-1) and generalization ability (setting-2). The results are shown in Table 6.3. The results demonstrate that increasing the number of horizontal decoding layers leads to a corresponding improvement of 0.5% and 1.06% in accuracy on setting-1. This trend suggests that a higher number of decoders is positively correlated with increased accuracy. Furthermore, the performance on setting-2 shows a gradual enhancement as the number of decoding layers increases, with improvements of approximately 0.55% and 1.0% in terms of dice score. These observed enhancements in performance on both setting-1 and setting-2 affirm that augmenting the number of decoding layers effectively refines both local and global features.

6.5 Conclusion

This study has introduced a novel dense attention gate mechanism aimed at refining the local relationships across multi-level encoder features by incorporating all previous layer features during attention score computation. Additionally, a novel multi-layer decoder architecture has been developed to further augment semantic features. Integrating both of these new design modules into a PVT-based encoder, the proposed structure yields significant performance improvements in five public datasets for polyp segmentation. These results not only underscore the efficacy of our proposed methodology but also open new avenues for advancing the state-of-the-art in medical image analysis.

Chapter 7

Summary & Future Work

7.1 Summary

In this thesis, we have proposed new network architectures and modules to solve the previous limitations and improve the performance on various datasets for image classification and medical image segmentation. Specifically,

1. We have proposed a channel diversification block to enhance the performance of image classification networks by diversifying network attention towards more significant channels. When integrated with baseline networks, our module outperforms all baseline networks and some attention-based networks, including SOAL, ABN, and SE-Net, significantly in terms of accuracy, number of additional parameters, and GFLOPs.
2. We investigated the impact of aggregating global information into a local vision transformer network after each stage. Additionally, we introduced a multi-resolution overlapped attention module that can be incorporated after each stage in the local transformer to facilitate information exchange among neighboring windows. Our proposed network architecture achieved notable performance improvements on various datasets such as CIFAR-10/100 and ImageNet, despite having comparatively fewer parameters.
3. We proposed a novel U-Net-based architecture aimed at enhancing the quality of encoding features layer by layer for the automated segmentation of polyps from colonoscopy images. Our network substantially improves polyp segmentation performance across five polyp datasets.

4. We have developed a novel attention mechanism designed to encourage the network to focus more on challenging regions, typically located around boundaries. Our attention module is integrated with various backbone networks. Our results demonstrate that the proposed fuzzy attention module significantly outperforms other state-of-the-art polyp segmentation models on the CVC-ClinicDB, CVC-ColonDB, and ETIS datasets.

5. Finally, we introduced a novel dense attention gate mechanism intended to enhance the local relationships across multi-level encoder features by incorporating all previous layer features during attention score computation. Moreover, we developed a novel multi-layer decoder architecture to further enhance semantic features. Integrating both of these new design modules into a PVT-based encoder resulted in significant performance gains across five public datasets for polyp segmentation.

7.2 Future work: Transformer in Medical Image Segmentation

Accurate medical image segmentation is a critical task in computer-aided cancer diagnosis, reducing the mortality rate. Transformer encoder based medical image segmentation has achieved a significant performance gain on various medical image segmentation datasets. However certain limitations affect the correctness of the medical image segmentation model and in future work, we will work towards solving those limitations.

(a) **Accurate boundary detection**:

Although our study (Patel et al., 2022b) attempted to address the challenge of boundary detection, certain areas with overlapping light and shadow remained undetected by our model. Furthermore, our model erroneously identified reflective points as polyps. Eliminating these reflective points is crucial to accurately classifying boundary regions, thereby improving the overall performance of the model. One potential approach involves leveraging gray-scale images to mitigate reflections and the overlap of light and shadow, thereby aiding the model's decision-making process in conjunction with RGB images.

(b) **Lack of fine-grained features:**

Transformer-based medical image segmentation has notably enhanced performance across diverse datasets owing to its capacity to capture global context through multi-head attention mechanisms. Nevertheless, these models often struggle to incorporate neighborhood information crucial for discerning fine-grained features and facilitating refined segmentation masks. Thus, the refinement of mechanisms aimed at enhancing local features in conjunction with global context could substantially augment model accuracy.

All the aforementioned challenges can be addressed through the implementation of a specially designed decoder. This decoder serves to augment local features derived from the global context features generated by the transformer encoder. Moreover, to mitigate the impact of reflections and shadows, gray-scale image features can prove instrumental in further enhancing performance.

In recent years, foundation models have gained considerable traction, even within the medical imaging domain. These models undergo training on extensive medical datasets, thereafter offering the potential for fine-tuning on specific datasets. We posit that pre-training our network with one of these foundation models, which has been trained on a diverse array of medical images, can significantly bolster the generalization capabilities of the segmentation model.

References

Akbari, M., Mohrekesh, M., Nasr-Esfahani, E., Soroushmehr, S. R., Karimi, N., Samavi, S., & Najarian, K. (2018a). Polyp segmentation in colonoscopy images using fully convolutional network. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 69–72).: IEEE.

Akbari, M., Mohrekesh, M., Nasr-Esfahani, E., Soroushmehr, S. R., Karimi, N., Samavi, S., & Najarian, K. (2018b). Polyp segmentation in colonoscopy images using fully convolutional network. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 69–72).: IEEE.

Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., Wang, Y., Gao, J., Piao, S., Zhou, M., et al. (2020). Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning* (pp. 642–652).: PMLR.

Bernal, J., Sánchez, F. J., Fernández-Esparrach, G., Gil, D., Rodríguez, C., & Vilariño, F. (2015). Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized medical imaging and graphics*, 43, 99–111.

Brandao, P., Mazomenos, E., Ciuti, G., Caliò, R., Bianchi, F., Menciassi, A., Dario, P., Koulaouzidis, A., Arezzo, A., & Stoyanov, D. (2017). Fully convolutional neural networks for polyp segmentation in colonoscopy. In *Medical Imaging 2017: Computer-Aided Diagnosis*, volume 10134 (pp. 101–107).: SPIE.

Cao, Y., Xu, J., Lin, S., Wei, F., & Hu, H. (2019). Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 0–0).

Cen, F., Zhao, X., Li, W., & Wang, G. (2021). Deep feature augmentation for occluded image classification. *Pattern Recognition*, 111, 107737.

Chen, C.-F., Fan, Q., & Panda, R. (2021). Crossvit: Cross-attention multi-scale vision transformer for image classification. *arXiv preprint arXiv:2103.14899*.

Chen, X., Hu, Q., Li, K., Zhong, C., & Wang, G. (2023). Accumulated trivial attention matters in vision transformers on small datasets. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 3984–3992).

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251–1258).

Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., & Shen, C. (2021). Twins: Revisiting the design of spatial attention in vision transformers. *arXiv preprint arXiv:2104.13840*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255).: Ieee.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, B., Wang, W., Fan, D.-P., Li, J., Fu, H., & Shao, L. (2021). Polyp-pvt: Polyp segmentation with pyramid vision transformers. *arXiv preprint arXiv:2108.06932*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fan, D.-P., Ji, G.-P., Zhou, T., Chen, G., Fu, H., Shen, J., & Shao, L. (2020). Pranet: Parallel reverse attention network for polyp segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 263–273).: Springer.

Fang, Y., Chen, C., Yuan, Y., & Tong, K.-y. (2019). Selective feature aggregation network with area-boundary constraints for polyp segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 302–310).: Springer.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3146–3154).

Fukui, H., Hirakawa, T., Yamashita, T., & Fujiyoshi, H. (2019). Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 10705–10714).

Gajurel, K., Zhong, C., & Wang, G. (2021). A fine-grained visual attention approach for fingerspelling recognition in the wild. *arXiv preprint arXiv:2105.07625*.

Gao, S.-H., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., & Torr, P. (2019). Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 43(2), 652–662.

Gao, Y., Han, X., Wang, X., Huang, W., & Scott, M. (2020). Channel interaction networks for fine-grained image categorization. In *AAAI* (pp. 10818–10825).

Girdhar, R. & Ramanan, D. (2017). Attentional pooling for action recognition. In *Advances in Neural Information Processing Systems* (pp. 34–45).

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

Guan, Q., Huang, Y., Zhong, Z., Zheng, Z., Zheng, L., & Yang, Y. (2020). Thorax disease classification with attention guided convolutional neural network. *Pattern Recognition Letters*, 131, 38–45.

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., & Wang, Y. (2021). Transformer in transformer. *arXiv preprint arXiv:2103.00112*.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

He, L., Lu, J., Wang, G., Song, S., & Zhou, J. (2021). Sosd-net: Joint semantic object segmentation and depth estimation from monocular images. *Neurocomputing*, 440, 251–263.

He, X., Yang, S., Li, G., Li, H., Chang, H., & Yu, Y. (2019). Non-local context encoder: Robust biomedical image segmentation against adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33 (pp. 8417–8424).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, H., Gu, J., Zhang, Z., Dai, J., & Wei, Y. (2018a). Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3588–3597).

Hu, H., Zhang, Z., Xie, Z., & Lin, S. (2019). Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 3464–3473).

Hu, J., Shen, L., & Sun, G. (2018b). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).

Huang, C.-H., Wu, H.-Y., & Lin, Y.-L. (2021). Hardnet-mseg: A simple encoder-decoder polyp segmentation neural network that achieves over 0.9 mean dice and 86 fps. *arXiv preprint arXiv:2101.07172*.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).

Jha, D., Smedsrud, P. H., Riegler, M. A., Halvorsen, P., Lange, T. d., Johansen, D., & Johansen, H. D. (2020). Kvasir-seg: A segmented polyp dataset. In *International Conference on Multimedia Modeling* (pp. 451–462).: Springer.

Jha, D., Smedsrud, P. H., Riegler, M. A., Johansen, D., De Lange, T., Halvorsen, P., & Johansen, H. D. (2019). Resunet++: An advanced architecture for medical image segmentation. In *2019 IEEE International Symposium on Multimedia (ISM)* (pp. 225–2255).: IEEE.

Jia, X., Xing, X., Yuan, Y., Xing, L., & Meng, M. Q.-H. (2019). Wireless capsule endoscopy: A new tool for cancer screening in the colon with deep-learning-based polyp recognition. *Proceedings of the IEEE*, 108(1), 178–197.

Kim, I., Baek, W., & Kim, S. (2020). Spatially attentive output layer for image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9533–9542).

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Nair, V., & Hinton, G. Cifar-10 (canadian institute for advanced research).

Krizhevsky, A., Nair, V., & Hinton, G. Cifar-100 (canadian institute for advanced research).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Larsson, G., Maire, M., & Shakhnarovich, G. (2016). Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*.

Li, K., Fathan, M. I., Patel, K., Zhang, T., Zhong, C., Bansal, A., Rastogi, A., Wang, J. S., & Wang, G. (2021a). Colonoscopy polyp detection and classification: Dataset creation and comparative evaluations. *arXiv preprint arXiv:2104.10824*.

Li, K., Wang, N. Y., Yang, Y., & Wang, G. (2021b). Sgnet: A super-class guided network for image classification and object detection. *arXiv preprint arXiv:2104.12898*.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017a). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117–2125).

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017b). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988).

Liu, J.-J., Hou, Q., Cheng, M.-M., Feng, J., & Jiang, J. (2019). A simple pooling-based design for real-time salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3917–3926).

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 10012–10022).

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11976–11986).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).

Ma, W., Tu, X., Luo, B., & Wang, G. (2022). Semantic clustering based deduction learning for image recognition and classification. *Pattern Recognition*, 124, 108440.

Ma, W., Wu, Y., Cen, F., & Wang, G. (2020). Mdfn: Multi-scale deep feature learning network for object detection. *Pattern Recognition*, 100, 107149.

Ma, W., Zhang, T., & Wang, G. (2021). Miti-detr: Object detection based on transformers with mitigatory self-attention convergence. *arXiv preprint arXiv:2112.13310*.

Maghsoudi, O. H. (2017). Superpixel based segmentation and classification of polyps in wireless capsule endoscopy. In *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)* (pp. 1–4).: IEEE.

Mamonov, A. V., Figueiredo, I. N., Figueiredo, P. N., & Tsai, Y.-H. R. (2014). Automated polyp detection in colon capsule endoscopy. *IEEE transactions on medical imaging*, 33(7), 1488–1502.

Mathur, P., Sathishkumar, K., Chaturvedi, M., Das, P., Sudarshan, K. L., Santhappan, et al. (2020). Cancer statistics, 2020: report from national cancer registry programme, india. *JCO Global oncology*, 6, 1063–1075.

Murugesan, B., Sarveswaran, K., Shankaranarayana, S. M., Ram, K., Joseph, J., & Sivaprakasam, M. (2019). Psi-net: Shape and boundary aware joint multi-task deep network for medical image segmentation. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 7223–7226).: IEEE.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

Park, J., Woo, S., Lee, J.-Y., & Kweon, I. S. (2018). Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*.

Patel, K., Bur, A. M., Li, F., & Wang, G. (2022a). Aggregating global features into local vision transformer. *arXiv preprint arXiv:2201.12903*.

Patel, K., Bur, A. M., & Wang, G. (2021). Enhanced u-net: A feature enhancement network for polyp segmentation. *arXiv preprint arXiv:2105.00999*.

Patel, K., Li, K., Tao, K., Wang, Q., Bansal, A., Rastogi, A., & Wang, G. (2020). A comparative study on polyp classification using convolutional neural networks. *PloS one*, 15(7), e0236452.

Patel, K. & Wang, G. (2022). A discriminative channel diversification network for image classification. *Pattern Recognition Letters*, 153, 176–182.

Patel, K. B., Li, F., & Wang, G. (2022b). Fuzzynet: A fuzzy attention module for polyp segmentation. In *NeurIPS'22 Workshop on All Things Attention: Bridging Different Perspectives on Attention*.

Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., & Jagersand, M. (2019). Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7479–7489).

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 658–666).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234–241).: Springer.

Sajid, U., Chen, X., Sajid, H., Kim, T., & Wang, G. (2021a). Audio-visual transformer based crowd counting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2249–2259).

Sajid, U., Chow, M., Zhang, J., Kim, T., & Wang, G. (2021b). Parallel scale-wise attention network for effective scene text recognition. *arXiv preprint arXiv:2104.12076*.

Sajid, U. & Wang, G. (2021). Towards more effective prm-based crowd counting via a multi-resolution fusion and attention network. *Neurocomputing*.

Silva, J., Histace, A., Romain, O., Dray, X., & Granado, B. (2014a). Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. *International journal of computer assisted radiology and surgery*, 9(2), 283–293.

Silva, J., Histace, A., Romain, O., Dray, X., & Granado, B. (2014b). Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. *International journal of computer assisted radiology and surgery*, 9(2), 283–293.

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support* (pp. 240–248). Springer.

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision* (pp. 843–852).

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 71(3), 209–249.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Tajbakhsh, N., Gurudu, S. R., & Liang, J. (2015a). Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging*, 35(2), 630–644.

Tajbakhsh, N., Gurudu, S. R., & Liang, J. (2015b). Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging*, 35(2), 630–644.

Tajbakhsh, N., Gurudu, S. R., & Liang, J. (2015c). Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging*, 35(2), 630–644.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning* (pp. 10347–10357).: PMLR.

Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B., & Shlens, J. (2021). Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12894–12904).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vázquez, D., Bernal, J., Sánchez, F. J., Fernández-Esparrach, G., López, A. M., Romero, A., Drozdzal, M., & Courville, A. (2017). A benchmark for endoluminal scene segmentation of colonoscopy images. *Journal of healthcare engineering*, 2017.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., & Tang, X. (2017). Residual attention network for image classification. In *Proceedings of CVPR* (pp. 3156–3164).

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., & Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 568–578).

Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7794–7803).

Wang, Y., Zhu, H., & Wang, G. (2023). Pst-net: Point cloud completion network based on local geometric feature reuse and neighboring recovery with taylor approximation. In *2023 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).: IEEE.

Wei, J., Hu, Y., Zhang, R., Li, Z., Zhou, S. K., & Cui, S. (2021). Shallow attention network for polyp segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 699–708).: Springer.

Wei, J., Wang, S., & Huang, Q. (2020). F^{3}net: fusion, feedback and focus for salient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34 (pp. 12321–12328).

Wickstrøm, K., Kampffmeyer, M., & Jenssen, R. (2020). Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps. *Medical image analysis*, 60, 101619.

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3–19).

Wu, Z., Su, L., & Huang, Q. (2019). Cascaded partial decoder for fast and accurate salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3907–3916).

Xiao, X., Hu, Q. V., & Wang, G. (2023). Edge-aware multi-task network for integrating quantification segmentation and uncertainty prediction of liver tumor on multi-modality non-contrast mri. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 652–661).: Springer.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492–1500).

Xu, W., Shawn, K., & Wang, G. (2019). Toward learning a unified many-to-many mapping for diverse image translation. *Pattern Recognition*, 93, 570–580.

Xu, W. & Wang, G. (2021). A domain gap aware generative adversarial network for multi-domain image translation. *IEEE Transactions on Image Processing*, 31, 72–84.

Yu, F. & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F. E., Feng, J., & Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6023–6032).

Zagoruyko, S. & Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, R., Li, G., Li, Z., Cui, S., Qian, D., & Yu, Y. (2020a). Adaptive context selection for polyp segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 253–262).: Springer.

Zhang, T., Zhang, X., Yang, Y., Wang, Z., & Wang, G. (2020b). Efficient golf ball detection and tracking based on convolutional neural networks and kalman filter. *arXiv preprint arXiv:2012.09393*.

Zhang, Y., Liu, H., & Hu, Q. (2021a). Transfuse: Fusing transformers and cnns for medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 14–24).: Springer.

Zhang, Z., Zhang, H., Zhao, L., Chen, T., Arik, S. O., & Pfister, T. (2021b). Nested hierarchical transformer: Towards accurate, data-efficient andinterpretable visual understanding. *arXiv preprint arXiv:2105.12723*.

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2881–2890).

Zhao, H., Zhang, Y., Liu, S., Shi, J., Loy, C. C., Lin, D., & Jia, J. (2018). Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 267–283).

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2921–2929).

Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support* (pp. 3–11). Springer.

Zou, T., Yang, S., Zhang, Y., & Ye, M. (2020). Attention guided neural network models for occluded pedestrian detection. *Pattern Recognition Letters*, 131, 91–97.