Qeios

Research Article

Merging Context Clustering with Visual State Space Models for Medical Image Segmentation

Yun Zhu[1], Dong Zhang[2], Yi Lin[3], Yifei Feng[4,5], Jinhui Tang[1]

1. School of Computer Science and Engineering, Nanjing University of Science and Technology, China; 2. Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong; 3. Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong; 4. First School of Clinical Medicine, Nanjing Medical University, China; 5. Department of General Surgery, Jiangsu Province Hospital, Nanjing, China

Medical image segmentation demands the aggregation of global and local feature representations, posing a challenge for current methodologies in handling both long-range and short-range feature interactions. Recently, vision mamba (ViM) models have emerged as promising solutions for addressing model complexities by excelling in long-range feature iterations with linear complexity. However, existing ViM approaches overlook the importance of preserving short-range local dependencies by directly flattening spatial tokens and are constrained by fixed scanning patterns that limit the capture of dynamic spatial context information. To address these challenges, we introduce a simple yet effective method named context clustering ViM (CCViM), which incorporates a context clustering module within the existing ViM models to segment image tokens into distinct windows for adaptable local clustering. Our method effectively combines long-range and short-range feature interactions, thereby enhancing spatial contextual representations for medical image segmentation tasks. Extensive experimental evaluations on diverse public datasets, i.e., Kumar, CPM17, ISIC17, ISIC18, and Synapse demonstrate the superior performance of our method compared to current state-of-the-art methods. Our code can be found at https://github.com/zymissy/CCViM.

Corresponding authors: Jinhui Tang, jinhuitang@njust.edu.cn; Dong Zhang, dongz@ust.hk; Yifei Feng, fengyifei1982@163.com

1. Introduction

Medical image segmentation (MedISeg) plays a crucial role in the communities of scientific research and medical care, aiming at delineating the region of interest and recognizing pixel-level and fine-grained lesion objects[1][2][3][4], which is critical for anatomy research, disease diagnosis, and treatment planning[5][6]. Deep learning advancements have sped up the automation of MedISeg, resulting in improved efficiency, accuracy, and reliability

when compared to manual segmentation technologies[7][8]. In the past years, accurate MedISeg methods have been successfully used in daily clinical applications, e.g., nuclei segmentation in microscopy images[9][10], skin lesion segmentation in dermoscopy images[11], and multi-organ segmentation in CT images[12][13].

Various types of feature interactions in deep learning methods have a significant impact on the performance and capabilities of MedISeg models[14][15][16]. Fig. 1 illustrates different forms of feature interaction mechanisms in various MedISeg models. Convolutional neural networks (CNNs)[17][18][19] in Fig. 1(a) conceptualize an image as a grid of pixels and compute element-wise multiplication between kernel weights and pixel values in a local field. While CNNs are effective at capturing local features, their ability to model long-range feature interactions is limited, which affects their performance. To address this limitation, Vision Transformer (ViTs) models have been introduced[20][21][22]. As illustrated in Fig. 1(b), ViTs treat an image as a sequence of patches, similar to tokens in natural language processing (NLP), allowing each token to attend to every token in the image and enabling ViTs to model long-range feature interactions. Due to the long-range feature interactions of ViT models, they can capture global contexts and demonstrate superior performance in MedISeg tasks[20][23][24]. However, the quadratic complexity of ViTs' long-range feature interactions in relation to the number of tokens has posed challenges in applying downstream applications.

To address these challenges, an efficient visual state space model called VMamba is introduced with linear complexity[25][26]. VMamba draws inspiration from the Mamba, originally designed for NLP tasks[27]. The Mamba model processes input data causally, leveraging its causal properties to perform effectively and efficiently in language modeling tasks. However, the non-causal nature of images poses challenges for the Mamba model in handling image data using its causal processing approach. In contrast, VMamba addresses this issue by patching and flattening an image, as illustrated in Fig. 1(c), then scanning the flattened image to integrate information from all locations in various directions, creating a global receptive field without increasing linear computational complexity. This innovative approach has led to the development of several Mamba-based models in the MedISeg domain[28][29]. Despite the benefits of global scanning methods, they may overlook local features and spatial context information. LocalMamba introduces a local scanning method that confines the scan within a local window, enabling the capture of local dependencies while maintaining a global perspective[26]. However, existing fixed scanning approaches may struggle to adaptively capture spatial relationships.
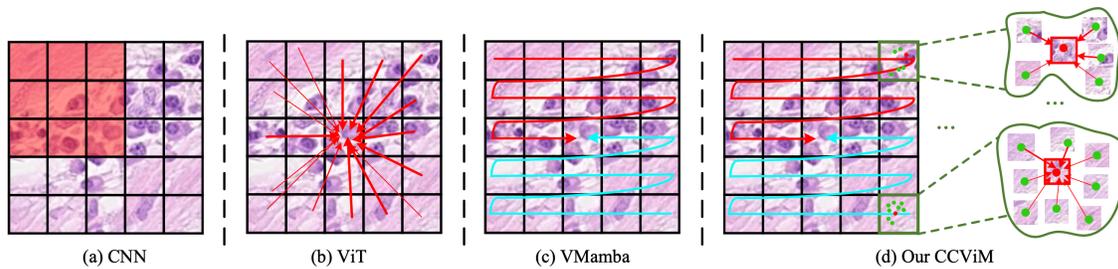
Figure 1. Illustration of the feature interaction mechanism in MedISeg. CNNs[17] in (a) conceptualize an image as structured feature grids, employing convolutional layers that slides over the local space with a certain stride. ViT[30] in (b) uses self-attention, treating an image as tokens, enabling each token to interact with other tokens. VMamba[25] in (c) uses the cross-scan module to integrate the pixels from different directions, achieving a global receptive field with linear complexity. Our CCViM in (d) combines the cross-scan module with our context clustering layer. Our method treats an image as a set of data points, dynamically grouping all points into clusters within a local window to extract local contexts.

To address the above problems, we introduce a simple yet effective Context Clustering Vision Mamba (CCViM) model, which is a U-shaped architecture designed for MedISeg that effectively captures both global and local feature interactions. As illustrated in Fig. 1(d), our CCViM can address the limitations of fixed scanning strategies by proposing the context clustering selective state space model (CCS6) layer, which combines the cross-scan layer with our context clustering (CC) layer to capture local spatial visual contexts dynamically. Compared to convolution scanning images in a grid-like manner or attention exploring mutual relationships within a sequence, our CC layer views the image as a set of data points and organizes these points into different clusters, which can dynamically capture local feature interactions. Comprehensive experiments are conducted on nuclei segmentation[31][32], skin lesion segmentation[33][34], and multi-organ segmentation[35] tasks to demonstrate the superior performance effectively and efficiently of our CCViM in MedISeg.

The main contributions of this paper are as follows: *(1)* Introduction of a novel Mamba-based model for MedISeg, CCViM, which is effective and efficient. *(2)* Proposal of the CCS6 layer, which combines global scanning directions with the CC layer, enhancing the model's feature representation capability. *(3)* Proposal of the CC layer, which can dynamically capture spatial contextual information, improving model performance compared to fixed scanning strategies. *(4)* Comprehensive experiments conducted on five public MedISeg datasets show that CCViM outperforms existing models, demonstrating superior performance.

II. Related Work

A. Medical Image Segmentation (MedISeg)

MedISeg is crucial for physicians in diagnosing specific diseases, as it delineates the regions of interest with precision[1][2][3]. Automatic MedISeg has been extensively researched to address the limitations of manual segmentation, which is often time-consuming, labor-intensive, and subjective[5][36][6]. Consequently, deep learning methods such as convolutional neural networks (CNNs)[17][18][19] and vision transformers (ViTs)[37][30][24][20] have been widely applied in MedISeg tasks[38]. For instance, UNet[39] is favored for its simple architecture and robust scalability, which derives many U-shaped models for MedISeg tasks[36]. UNet++[40] proposes nested encoder-decoder sub-networks with skip connections for MedISeg. TransUnet[20] proposes the first Transformer-based model for MedISeg tasks. Swin-Unet[24], a pure Transformer-based U-shaped model, adopts a hierarchical Swin Transformer[41] block with shifted windows as the encoder and performs effectively in MedISeg. However, CNNs' performance is limited with the local respective field, and Transformer-based models require quadratic complexity. To overcome these challenges, Mamba[27] has been proposed to solve the computational efficiency problem and modeling long-range dependencies. In this paper, our research is also based on Mamba. Our contribution is to propose a more efficient local interaction mechanism.

B. Feature Interaction for MedISeg

Feature interaction matters, and it greatly affects MedISeg's performance. CNNs have dominated the field of computer vision in recent years, benefiting from some key inductive biases e.g., locality and translation equivalence[42][14]. The CNN's locality helps identify distinct and stable points in images, e.g., corners, edges, and textures, which are significant for detecting small and irregularly shaped target features in MedISeg[43][44]. However, the local feature interaction cannot accurately address the complex and interconnected anatomical structures in MedISeg, which also requires long-range feature interactions to capture the global context and spatial relationships within the medical image[45]. In recent years, ViTs have emerged as effective long-range feature interaction methods in MedISeg tasks, e.g., Swin-Unet[24], UTNet[46], and Missformer[47]. Since ViTs abandon the local bias from CNNs, some researchers combine convolution and attention to acquire both local and global features, e.g., TransUnet[20] and Conformer[22]. Although the above methods inherit the advantages from short-range and long-range features and achieve better performance, the insights and knowledge are still restricted to CNNs and ViTs[48]. Besides, CNNs or ViTs are limited in modeling long-range dependencies or computational efficiency. Therefore, except for Mamba[27], we also attempt to employ a new paradigm for visual representation by using a clustering[48]. A clustering algorithm views an image as a set of points, allowing it to capture local topology information adaptively without introducing significant computational complexity.

C. Vision Mamba (ViM)

Mamba[27], originally designed for the NLP tasks, proposes a novel selective mechanism that focuses on relevant information or filters out irrelevant information. Mamba[27] eliminates the limitations in CNNs and Transformers, and achieves both effectiveness and efficiency. However, due to the intrinsic non-causal characteristic of images, there is a significant obstacle to Mamba's comprehension of images. Therefore, a series of Mamba-based models have been proposed to address this issue. For instance, the representative ViM[49] flattens spatial data into 1D tokens and scans these tokens in two different directions to address the non-casual problem in the vision domain. However, flattening spatial data into 1D tokens disrupts the natural 2D dependencies. To address this problem, VMamba[25] introduces a cross-scan module to bridge the gap between 1D array scanning and 2D dependencies, effectively resolving the non-causal problem in visual images without compromising the receptive field. Besides, influenced by the success of VMamba[25], VM-UNet[28] introduces a visual Mamba UNet for MedISeg. However, these Mamba-based methods overlook the local features, which are both important in the vision domain. Therefore, LocalMamba[26] introduces a local scan strategy to capture both global and local information. However, such fixed scanning approaches are limited in their ability to dynamically capture spatial relationships. To address this challenge, we propose the CCS6 module, which integrates scanning modules with our CC to adaptively extract both global and local features while capturing spatial context information.

III. Our Method

A. Preliminary and Notation

Mamba[27] is a causal visual state space model, which is both effective and efficient, identifying that a key weakness of the structured state space model (SSM) is its inability to perform content-based reasoning. To overcome this, Mamba introduces a selective state-space model (S6). The recurrent formula of SSM can be defined as follows:

$$
\begin{aligned}
h'(t) &= \overline{\mathbf{A}}h(t) + \overline{\mathbf{B}}x(t), \\
y(t) &= \mathbf{C}h(t),
\end{aligned}
\tag{1}
$$

where $h(t) \in \mathbb{R}^N$ is the intermediate latent state to map $x(t) \in \mathbb{R}$ to $y(t) \in \mathbb{R}$. $\mathbf{A} \in \mathbb{R}^{N \times N}$ and $\mathbf{B} \in \mathbb{R}^{N \times N}$ are discretized by $\overline{\mathbf{A}} = \exp(\Delta \mathbf{A})$ and $\overline{\mathbf{B}} = (\Delta \mathbf{A})^{-1}(\exp(\Delta \mathbf{A}) - \mathbf{E}) \cdot \Delta \mathbf{B}$. Δ is the timescale parameter for discretizing the parameters $\mathbf{A}$ and $\mathbf{B}$. Through discretization, continuous-time system SSM can be integrated into deep-learning models. $\mathbf{C} \in \mathbb{R}^{N \times 1}$ is the parameter matrix. $\mathbf{E}$ is the identify matrix. On the basis of SSM, S6 lets the SSM parameters be functions of the input, allowing the model to selectively focus on or filter out information. Making the parameters $\Delta, \mathbf{B}, \mathbf{C}$ depend on the input x, and the formulas are defined as follows:

$$
s_B(x) = \text{Linear}_N(x),
$$

$$
s_C(x) = \text{Linear}_N(x),
$$

$$s_\Delta(x) = \text{softplus}(\text{Parameter}+$$

$$\text{Broadcast}_D(\text{Linear}_1(x))), \tag{2}$$

where $\text{Linear}_N, \text{Linear}_1$ are fully connected (FC) layers to project the embedding dimension of x to N and 1. Broadcasrt_D matches the dimension of the output of x to the Parameter. softplus is an activation function. Integrating S6, the Mamba not only achieves linear computation, but also performs excellent in language modeling tasks[27][50][51][52]. However, due to the inherent non-causal characteristics of images, it's difficult for Mamba-based models[27] to handle image data well. Fortunately, VMamba[25] proposes a cross-scan module to scan the patched and flattened image in four directions, which can address the non-causality of the image without compromising the field of reception and increasing computational complexity.

B. Overall Architecture

The overview of CCViM is illustrated in Fig. 2(a), which is composed of a patch embedding layer, an encoder, a decoder, a final projection layer, and skip connections. CCViM is not a symmetrical structure like UNet[39], but adopts an asymmetric design. First, input the image $\mathbf{I} \in \mathbb{R}^{3 \times W \times H}$ into the patch embedding layer to divide $\mathbf{I}$ into non-overlapping patches of size 4×4, getting a feature map with dimensions of $C \times \frac{W}{4} \times \frac{H}{4}$, where C is the number of feature channels. Then, input the feature map into the encoder network. Each stage in the encoder network is composed of two CCViM blocks and one patch merging layer, while the last stage only has two CCViM blocks without patch merging layer after them. The patch merging layer is for reducing the height and width of the feature maps and increasing the number of channels, and the channels of feature maps in the four stages are $[C, 2C, 4C, 8C]$. Thirdly, the feature maps are translated into the decoder, which also has four stages. Each stage of decoder is composed of one patch expanding layer and two CCViM blocks, while the first stage only has two CCViM blocks without patch expanding layer before them. The patch expanding layer is for increasing the height and width of the feature maps and reducing the number of channels, and the channels of feature maps in the four stages are $[8C, 4C, 2C, C]$. Furthermore, we simply employ the addition skip connections to capture low-level and high-level features. Besides, we adopt the most foundational Cross-Entropy and Dice loss for our MedISeg tasks. The CCViM block is the core component of CCViM, as shown in Fig. 2(b). The input is first translated into the normalization layer and then split into two branches. In the first branch, the input passes through a linear layer followed by an activation layer. In the second branch, the input passes through a linear layer, depth-wise convolution, and an activation layer and then translates into the core module of the CCViM block: context clustering selective state space model (CCS6) layer. After normalizing the output of CCS6 layer, multiply it with the output from the first branch to merge the out of the two pathways. Finally, a linear layer is used to project the merged features onto the dimensions of the initial input features to establish residual connections.

C. Context Clustering Selective State Space Model

Context clustering selective state space model (CCS6) layer is the core component of the CCViM block, which adopts a selective state space model (S6) as its footstone. S6 processes the input data causally, resulting in only capturing vital information within the scanned part of the data, which is difficult for processing non-causal images. Based on which, numerous researches have also proposed various scan strategies to solve this problem well[49][25][28][26]. However, these global scanning methods overlook the local features and the spatial context information in medical images. Furthermore, all these fixed scanning approaches cannot effectively capture spatial relationships adaptively. To overcome these challenges, we propose a CCS6 layer to extract both global and local features while capturing spatial context information in a learnable way. We adopt VMamba's[25] cross-scan module, which proposes a selective scan mechanism across four different directions. As shown in Fig. 2(c), we patch and flatten the input image, and scan the flattened image in both horizontal and vertical directions to capture the global information. At the same time, our CC layer performs learnable local context clustering in local windows. Additionally, the CC layer employs two different numbers of clustering centers–4 and 25–to capture varying structural information. Consequently, there are two distinct CC layers and four different scanning directions available for selection. To avoid introducing too much computational complexity, in each CCS6 layer, we select four from the six choices as in[28][26]. The detailed configuration of these choices will be introduced in section IV-B.
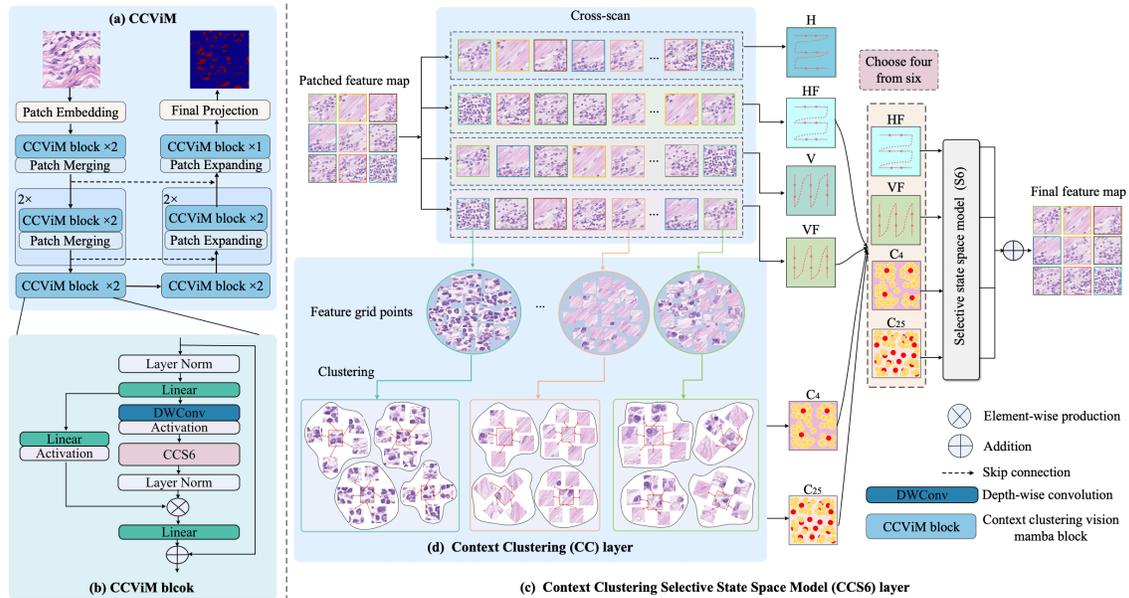
Figure 2. (a) The overall architecture of context clustering vision mamba (CCViM), a U-shaped structure for MedISeg. (b) CCViM block, the core component of CCViM. (c) CCS6 layer, the core module of CCViM block. CCS6 layer employs six different methods to process the input (patched feature maps). Four of these methods use a cross-scan module to flatten the patched feature maps and scan the flattened features in four different directions. The other two methods apply CC layers with 4 and 25 cluster centers. Then select four methods from all six methods to process the input feature map, and input the processed information into the S6 module. Finally, merge the output features to construct the final feature map. (d) CC layer, which views each patch as a set of feature grid points, and clusters these feature grid points into several centers.

D. Context Clustering

Instead of using convolution or attention to extract information, we use the context cluster (CC) algorithm[48] for local extraction and spatial context information. In our CC layer, we view the image as a set of data points and group all points into clusters. Instead of clustering data points over the entire image, which will bring a significant computational cost. We split the image into distinct windows and limit the clustering operation to a local region. In each cluster, we aggregate the points into a center adaptively. As shown in Fig. 2(d), after the patch embedding layer, the image $\mathbf{I} \in \mathbb{R}^{3 \times W \times H}$ is transformed into patched feature maps $\mathbf{F} \in \mathbb{R}^{d \times w \times h}$. We convert the patched feature maps into a set of data points $\mathbf{P} \in \mathbb{R}^{d \times n}$, where $n = w \times h$ represents the total number of data points, d is the point feature dimension. These points are unordered and disorganized. We then group the data points into several clusters based on their similarities, ensuring that each point is assigned to only one cluster.

For CC layer, We first project $\mathbf{P}$ to $\mathbf{P}_S \in \mathbb{R}^{n \times d'}$ to compute the similarity, where d' is the new point feature dimension. We evenly propose t centers in each local window and compute the center feature by averaging its

k nearest points. Then, we calculate the pair-wise cosine similarity between the resulting t center points and $\mathbf{P}_S$ to get the similarity matrix $\mathbf{S} \in \mathbb{R}^{t \times n}$. Based on the similarity, we allocate each point to the most similar center to get t clusters. Furthermore, each cluster may exhibit a varying number of data points. In exceptional cases, some clusters may contain no data points, indicating that they are redundant. During the clustering, all data points in one cluster are dynamically aggregated into the center point based on the similarities. In a cluster, there is a small set of m data points, represented by $\mathbf{P}_m \in \mathbb{R}^{m \times d'}$. Calculating the similarity between the m data points $\mathbf{P}_m \in \mathbb{R}^{m \times d'}$ and the center, which is represented by $\mathbf{s} \in \mathbb{R}^m$ and is the subset of similarity matrix $\mathbf{S}$. We map the m data points $\mathbf{P}_m \in \mathbb{R}^{m \times d'}$ to a value space to get $\mathbf{P}'_m \in \mathbb{R}^{m \times d'}$. We also generate a value center $\mathbf{v}_c$ of the $\mathbf{P}'_m$ in the value space, which is like the clustering center proposal. Then the aggregated feature $\mathbf{g} \in \mathbb{R}^{d'}$ is formulated as follow:

$$
\begin{aligned}
\mathbf{g} &= \frac{1}{T}\left(\mathbf{v}_c + \sum_{i=1}^{m} \sigma(\alpha \mathbf{s}_i + \beta) * \mathbf{p}_i\right), \\
\text{s.t.,} \quad T &= 1 + \sum_{i=1}^{m} \sigma(\alpha \mathbf{s}_i + \beta),
\end{aligned}
\tag{3}
$$

where α and β are learnable scalars used to scale and shift the similarity, and σ is a sigmoid function that re-scales the similarity to the range (0, 1). $\mathbf{p}_i$ denotes i-th point in $\mathbf{P}'_m$. For numerical stability and to emphasize locality, we incorporate the value center $\mathbf{v}_c$ in Eq. (3). To control the scale, normalizing the aggregated feature by a factor of T. Subsequently, adaptively dispatching the aggregated feature $\mathbf{g}$ to each data point in a cluster based on the similarity. This approach facilitates the communication of the points in the cluster, enabling them to share features in the cluster. For each data point $\mathbf{p}_i$ of $\mathbf{P}'_m$ in the cluster, updating it by:

$$
\mathbf{p}'_i = \mathbf{p}_i + \mathrm{FC}(\sigma(\alpha \mathbf{s}_i + \beta) * \mathbf{g}),
\tag{4}
$$

where $\mathbf{s}$ denotes similarity, using the same procedures as above. We apply a fully connected (FC) layer to project the feature dimension from the value space dimension d' to the original dimension d.

E. Post Processing

In the nuclei segmentation task (i.e., Kumar[31] and CPM17[32]), we employ the post-processing method watershed algorithm to distinguish between individual nuclei instances. Following research[9], we create horizontal and vertical distance maps by calculating the distances from nuclear pixels to their centers of mass in both the vertical and horizontal directions. Within our model we predict the vertical distance maps P_v and horizontal distance maps P_h. Additionally, we apply the Sobel operator to these distance maps to obtain the horizontal and vertical gradient maps. We then select the maximum value between these gradient maps: $M_s = \max(\mathrm{Sobel}(P_v), \mathrm{Sobel}(P_h))$. This process aids in edge detection by calculating the gradient magnitude, thereby emphasizing areas of significant intensity change. Next, we calculate the markers, M, by applying a threshold function to the probability map P and gradient map M_s, where the markers are defined as $M = \delta(\tau(P, r) - \tau(M_s, k))$. Here, $\tau(P, r)$ is a threshold function, with r being the threshold value. If the value exceeds r, it is set to 1; otherwise, it is set to 0. δ is used to set

negative values to 0. We then obtain the energy landscape $E = (1 - \tau(M_s, k)) \times \tau(P, h)$. Finally, M serves as the marker during marker-controlled watershed to determine how to split $\tau(P, h)$, guided by the energy landscape E.

F. Loss Function

For a fair comparison with other state-of-the-art methods, we employ the most fundamental loss functions in medical image segmentation: Cross-Entropy and Dice loss[28][9][53]. Cross-entropy ensures pixel-level classification accuracy, while Dice loss addresses the common issue of class imbalance by optimizing the overlap between the predicted and true segmentation[54]. As shown in Eq. (5) and Eq. (6), we combine Cross-Entropy and Dice loss to balance precise pixel-wise classification with overall segmentation performance.

$$L = L_{Ce} + L_{Dice} \tag{5}$$

$$\begin{cases} L_{\text{Ce}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}) \\ \\ L_{\text{Dice}} = 1 - \frac{2 \sum_{i=1}^{N} y_i \hat{y}_i + 1}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i + 1} \end{cases} \tag{6}$$

where, N denotes the total number of samples, and C represents the total number of categories. $y_{i,c}$ is an indicator of ground truth, equals 1 if sample i belongs to category c, and 0 if it does not. $\hat{y}_{i,c}$ is the probability that the model predicts sample i as belonging to category c. y_i and $\hat{y}_i$ represent the ground truth and prediction, respectively.

IV. Experiments

A. Datasets and Evaluation Metrics

Datasets. We evaluate CCViM on five MedISeg datasets, which contain nuclei segmentation datasets (i.e., Computational Precision Medicine (CPM17)[32] and Kumar[31]), skin lesion segmentation datasets (i.e., ISIC17[33] and ISIC18[34]), and Synapse multi-organ segmentation dataset (i.e., Synapse[35]). These datasets are detailed as follows:

- **Kumar:** The size of images is 1000×1000 pixels at $40\times$ magnification. The total number of nuclei is 21623. Following[31], we split the dataset into two different sub-datasets: (i) Kumar-Train, a training set with 16 images, and (ii) Kumar-Test, a test set with 14 images. Following previous research[9], we crop each training image into 540×540 with an overlap of 164 pixels, then resize into 250×250. Data augmentation, including flip and rotation, is applied to all methods. In the inference, the images are cropped into 250×250 with an overlap of 164 pixels.

- **CPM17:** The size of given images is 500×500 to 600×600 pixels at $40\times$ or $20\times$ magnification. The total number of nuclei is 7570. Following the challenge[32], we split 32 images in the training dataset and 32 images in the test dataset. The data processing also follows the previous research[9].

- **ISIC17 & ISIC18:** The International Skin Imaging Collaboration 2017 and 2018 challenge datasets (ISIC17[33] and ISIC18[34]) contain 2,150 and 2,694 images with segmentation masks, respectively. Following previous research[55], we allocate a 7:3 ratio for training and test sets, resize the images to 256×256, and apply data augmentation like flip and rotation.

- **Synapse:** Synapse multi-organ segmentation dataset[35] comprises 30 abdominal CT cases with 3779 axial abdominal clinical CT images, including 8 types of abdominal organs (aorta, gallbladder, left kidney, right kidney, liver, pancreas, spleen, stomach). Following previous research[20][24], we allocate 18 cases for training and 12 cases for testing. We resize the images to 224×224 and apply augmentation including flip and rotation.

Evaluation metrics. (1) In nuclei segmentation[31][32], we employ the ensemble dice (DICE), aggregated Jaccard index (AJI), panoptic quality (PQ), detection quality (DQ), and segmentation quality (SQ) as the main evaluation metrics. PQ is composed of DQ and SQ, offering precise quantification and interpretability for evaluating the performance of nuclei segmentation. (2) In skin lesion segmentation, to compare our CCViM with previous methods on ISIC17[33] and ISIC18[34] datasets, we employ mean intersection over union (mIoU), dice similarity coefficient (DSC), accuracy (Acc), sensitivity (Sen), and specificity (Spe) as the main evaluation metrics. (3) In Synapse[35] multi-organ segmentation[35], we employ dice similarity coefffcient (DSC) and the 95% Hausdorff distance (HD95) as the main evaluation metrics.

B. Configuration of Scan Directions and Local Clusters

In this paper, we do not apply the redundancy 22 scan strategies[56] in each CCS_6 layer, and there are only one, two, or three different scanning directions in each CCS_6 layer. To extract local features and capture the spatial context information adaptively, we integrate our CC layer into each CCS_6 layer. As shown in Fig. 3, each CCS_6 layer contains only four modules, which include one, two, or three different scanning directions and one or two different CC layers. This design avoids adding additional computational overhead compared to previous models. There are a total of four different scanning directions to choose from, including horizontal, horizontal flipped, vertical, and vertical flipped. Given the different sizes and number of image targets, we adopt two different CC layers. One CC layer method has 4 cluster centers in a local region, and the other one has 25 cluster centers in a local region. In nuclei segmentation tasks[31][32], there are many nuclei in one local region. In skin image segmentation tasks[33][34], the target is large and may need lower point centers in one local region. We adopt the configuration provided by LocalMamba[26], with our CC lyaer replacing the local scan. Compared with the fixed local scan strategies in LocalMamba[26], our CC layer can dynamically cluster local features and capture spatial information adaptively.
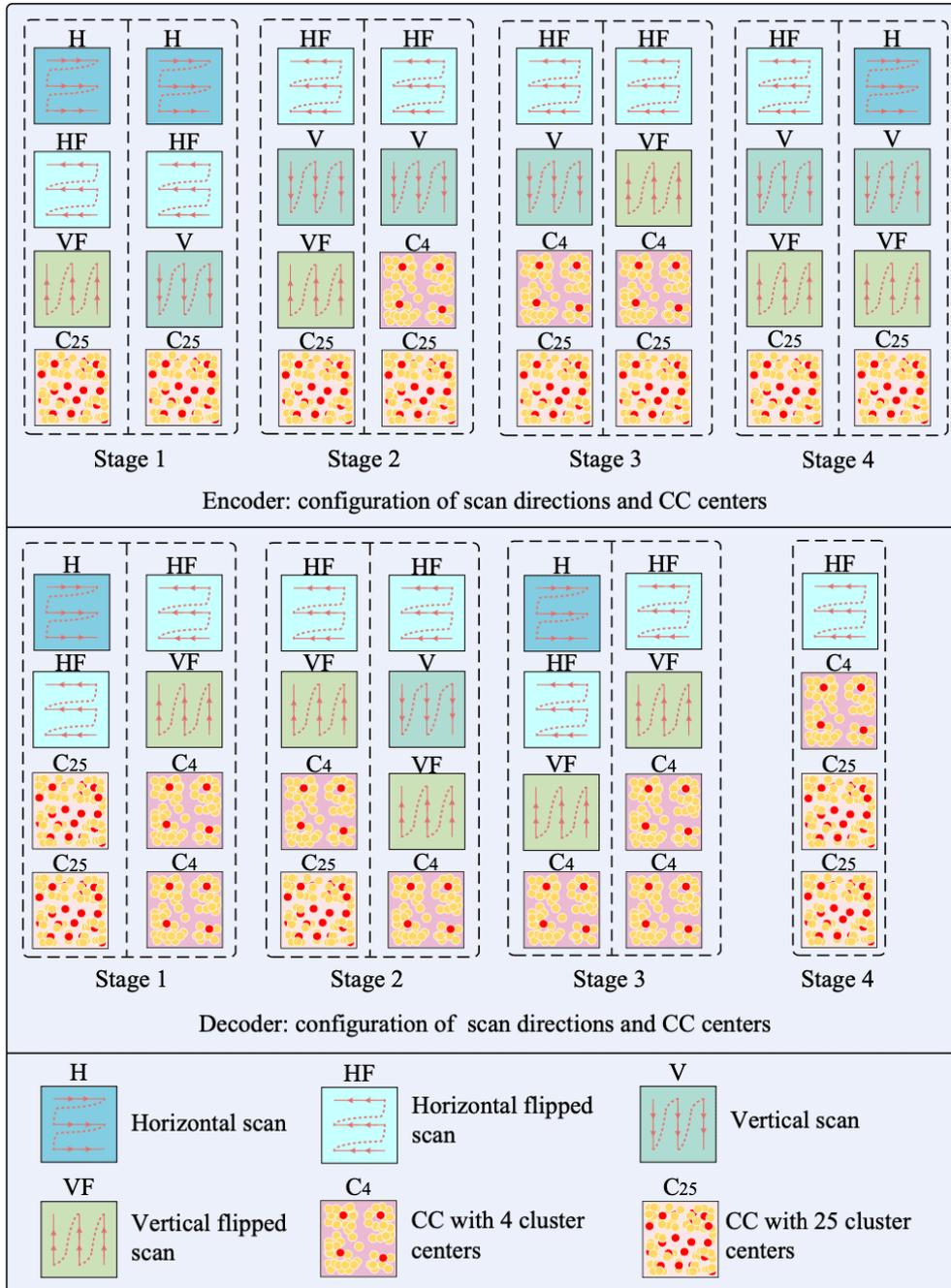
Figure 3. The configuration of various scan directions and CC layer with different cluster centers. The configuration of each stage is different. We directly use the configuration of LocalMamba[26], replacing LocalMamba's local scan with our CC layer.

C. Implementation Details

In our all experiments, we set the batch size to 32 and employ AdamW[57] optimizer with an initial learning rate of 1e-3. CosineAnnealingLR[58] is employed as the scheduler. We set the training epochs to 300. All experiments initialize the models with ImageNet[59] pretrained weights. All experiments are conducted on the PyTorch deep learning platform with a single NVIDIA GeForce RTX 3090 GPU.

D. Comparisons with State-of-the-art Methods

Results on nuclei segmentation. We compare our CCViM with CNN-based, Transformer-based, and Mamba-based models. Table 1 shows the results of different models on Kumar[31] and CPM17[32] datasets. In Table 1, the TransUnet[20] and Swin U-Net[24] have superior results than U-Net[39], especially on CPM17[32] datasets. Compared to the methods above, VM-UNet[28] demonstrates enhanced performance, particularly on the PQ and Dice metrics. This demonstrates the effectiveness of the Mamba-based model. On the Kumar dataset[31], VM-UNet[28] performs worse than Hover-Net[9] due to the small and dense nature of the nuclei on this dataset, which requires local feature interactions to capture the subtle details of these objects. In contrast, our CCViM outperforms Hover-Net[9], improving the PQ, Dice, AJI, DQ, and SQ by 0.61%, 1.16%, 1.76%, 0.88% and −0.08%, respectively. This demonstrates that our CCViM is superior at capturing local features. On the CPM17 dataset, compared to VM-UNet[28], our CCViM has improved the PQ, Dice, AJI, DQ, and SQ by 0.48%, 0.44%, 0.82%, 0.82% and −0.24%, respectively. Overall, our CCViM has the best performance in the PQ metric, indicating its ability to achieve more precise separation of individual nuclei. As shown in Fig. 4, our CCViM demonstrates superior performance on both the Kumar[31] and CPM17[32] datasets by accurately segmenting small and overlapping nuclei and delineating edges precisely. In contrast, other methods tend to merge distinct nuclei into a single entity or over-segment them. These results underscore the exceptional capability of our CC layer in local feature extraction, effectively capturing subtle differences and boundary details between nuclei, thereby significantly enhancing segmentation accuracy.
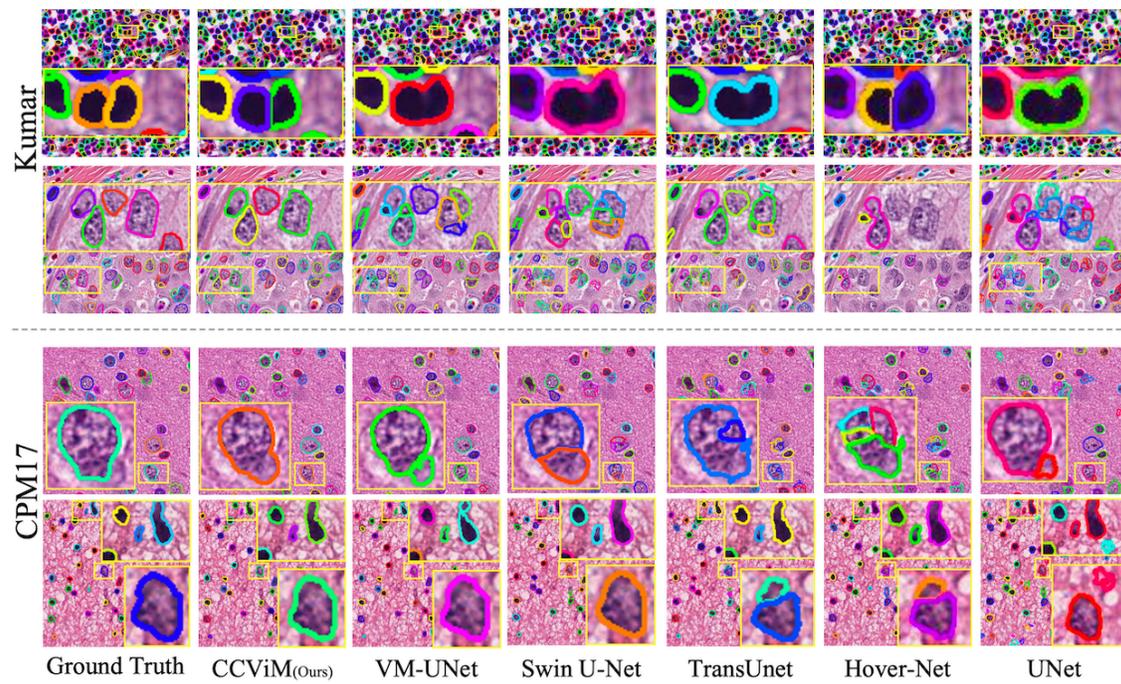
Figure 4. Visualizations on Kumar[31] and CPM17[32] datasets. Different colours of the nuclear boundaries denote separate instances.

	Methods	PQ(%)↑	Dice(%)↑	AJI(%)↑	DQ(%)↑	SQ(%)↑
Kumar[31]	SegNet[60]	40.70	81.10	37.70	54.50	74.20
	UNet[39]	47.80	75.80	55.60	69.10	69.00
	DIST[61]	44.30	78.90	55.90	60.10	73.20
	CIA-Net[61]	57.70	81.80	**62.00**	75.40	76.20
	Hover-Net[9]	58.22	81.32	59.62	75.62	76.71
	TransUNet[20]	48.14	77.76	52.90	65.74	72.90
	SwinUNet[24]	47.46	80.11	51.01	64.68	72.86
	VM-UNet[28]	56.59	81.89	60.00	74.27	75.85
	CCViM(Ours)	**58.83**	**82.48**	61.38	**76.50**	**76.63**
CPM17[32]	SegNet[60]	53.10	85.70	49.10	67.90	77.80
	UNet[39]	57.80	81.30	64.30	77.80	73.40
	DIST[61]	50.40	82.60	61.60	66.30	75.40
	CIA-Net[61]	65.70	86.20	68.30	81.10	80.40
	Hover-Net[9]	70.47	87.63	72.44	86.67	81.14
	TransUNet[20]	60.67	84.51	64.19	78.33	77.15
	SwinUNet[24]	60.12	86.08	62.88	77.11	77.52
	VM-UNet[28]	71.29	87.91	72.16	87.14	**81.67**
	CCViM(Ours)	**71.77**	**88.35**	**72.98**	**87.96**	81.43

Table 1. Result comparisons with state–of–the–art methods on the Kumar[31] and CPM17[32] datasets.

Results on skin image segmentation. We compare our CCViM with several state–of–the–art models on the ISIC17[33] and ISIC18[34] datasets, Table 2 shows the main results. Compared with CNN–based (i.e., UNet[39] and UTNetV2[62]) and Transformer–based (i.e., TransFuse[63] and MALUNet[55]) methods, the Mamba–based models (i.e., VM–UNet[28], HC–Mamba[29]) have the superior performance, which demonstrates the effectiveness of Mamba–based models in MedISeg. From Table 2, on the ISIC17 dataset, we can observe that our CCViM has improved the mIoU and DSC by 1.17% and 0.71% compared with VM–UNet[28]. On the ISIC18 dataset, our CCViM has improved the

mIoU and DSC by 0.57% and 0.35% compared with VM–UNet[28]. These superior results demonstrate that our CCViM effectively addresses the limitations of Mamba–based models, particularly in preserving local features and spatial context. Fig. 5 shows the visualizations on the ISIC17 dataset, we can observe that our CCViM exhibits accurate and sharp contours, effectively capturing object boundaries. Fig. 6 shows the visualization on the ISIC18[34] dataset, where our CCViM outputs highlight a closer alignment between the red and green contours, indicating that CCViM is more effective in accurately delineating skin lesions compared to VM–UNet[28]. The VM–UNet[28] often produces over–segmented or merged regions. While VM–UNet[28] performs well in some cases, CCViM demonstrates greater robustness, particularly when handling small lesions and irregular boundaries, offering a closer adherence to the ground truth. These visualizations further demonstrate that the local feature interactions in our CCViM can capture the subtle details of edge contours and irregularly shaped target features.
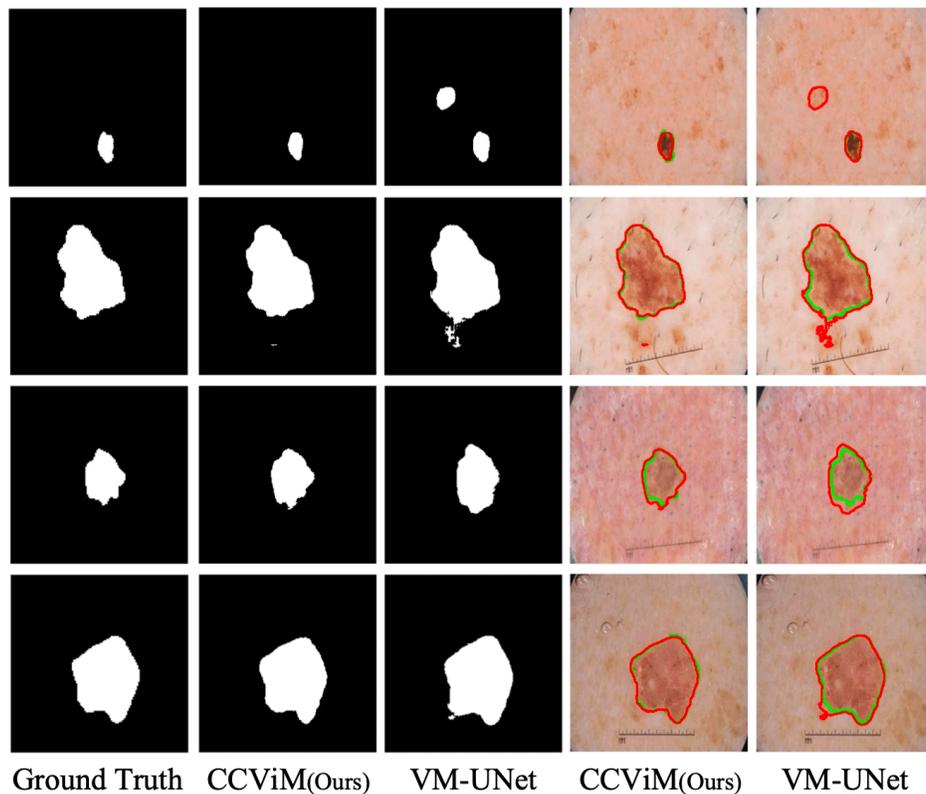
Figure 5. Visualizations on the ISIC17[33] Dataset. The left side presents the ground truth alongside the predicted masks from our model and the VM–UNet[28]. It is evident that our predicted masks are closer to the ground truth. The right side displays the original skin images annotated with lesion contours; green contours denote the ground truth, while red contours indicate the predicted segmentation results. These comparisons further demonstrate the superior effectiveness of our CCViM in accurately segmenting skin lesions.

| Ground Truth | CCViM(Ours) | VM-UNet | CCViM(Ours) | VM-UNet |
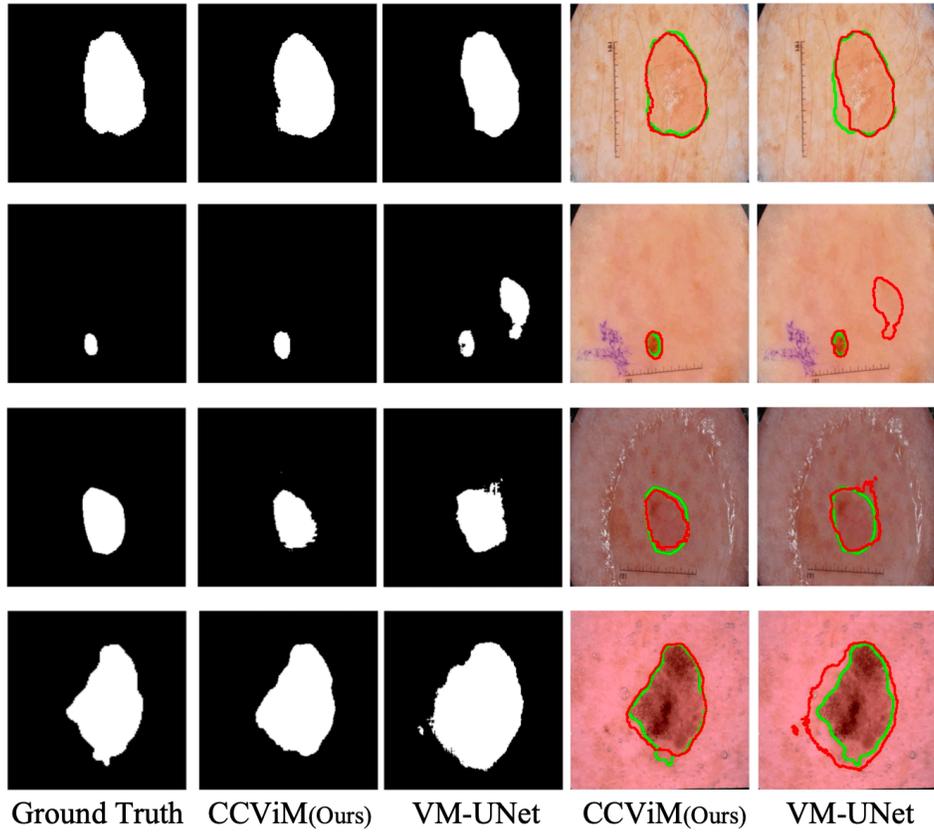
Figure 6. Visualizations on the ISIC18[34] Dataset. The same visualization format as the ISIC17 results described earlier.

	Methods	mIoU(%)↑	DSC(%)↑	Acc(%)↑	Spe(%)↑	Sen(%)↑
ISIC17[33]	UNet[39]	76.98	86.99	95.65	97.43	86.82
	UTNetV2[62]	77.35	87.23	95.84	98.05	84.85
	TransFuse[63]	79.21	88.40	96.17	97.98	87.14
	MALUNet[55]	78.78	88.13	96.18	**98.47**	84.78
	VM-UNet[28]	80.23	89.03	96.29	97.58	**89.90**
	HC-Mamba[29]	79.27	88.18	95.17	97.47	86.99
	CCViM(Ours)	**81.40**	**89.74**	**96.60**	98.19	88.70
ISIC18[34]	UNet[39]	77.86	87.55	94.05	96.69	85.86
	UNet++[40]	78.31	87.83	94.02	95.75	88.65
	Att-UNet[23]	78.43	87.91	94.13	96.23	87.60
	UTNetV2[62]	78.97	88.25	94.32	96.48	87.60
	SANet[64]	79.52	88.59	94.39	95.97	89.46
	TransFuse[63]	80.63	89.27	94.66	95.74	**91.28**
	MALUNet[55]	80.25	89.04	94.62	96.19	89.74
	VM-UNet[28]	81.35	89.71	94.91	96.13	91.12
	HC-Mamba[29]	80.60	89.25	94.84	97.08	87.90
	CCViM(Ours)	**81.92**	**90.06**	**95.23**	97.32	88.74

Table 2. Result comparisons with state-of-the-art methods on the ISIC17[33] and ISIC18[34] datasets

Results on synapse multi-organ segmentation. We also compare our CCViM with state-of-the-art models on the Synapse dataset, where similar observations and conclusions can be observed. In Table 3, Mamba-based models have superior performance compared with CNN-based and Transformer-based models. Compared with VM-UNet[28], our model has improved the DSC by 1.57% and has reduced the HD95 by 1.38%. Our model gets state-of-the-art results in segmenting 8 types of abdominal organs (aorta, gallbladder, left kidney, right kidney, liver, pancreas, spleen, stomach). Fig. 7 shows the visualizations, compared to VM-UNet[28], we can observe that our CCViM not only segments various organs more accurately but also delineates edges with greater precision. The

superior performance further demonstrates the effectiveness of both global and local feature interactions in our CCViM.

Methods	DSC(%)↑	HD95(%)↓	Aorta	Gallbladder	Kidney(L)	Kidney(R)	Liver	Pancreas	Spleen	Stomach
V–Net[53]	68.81	–	75.34%	51.87%	77.10%	80.75%	87.84%	40.05%	80.56%	56.98%
DARR[65]	69.77	–	74.74%	53.77%	72.31%	73.24%	94.08%	54.18%	89.90%	45.96%
R50 UNet[20]	74.68	36.87	87.47%	66.36%	80.60%	78.19%	93.74%	56.90%	85.87%	74.16%
UNet[39]	76.85	39.70	89.07%	**69.72%**	77.77%	68.60%	93.43%	53.98%	86.67%	75.58%
R50 Att-UNet[20]	75.57	36.97	55.92%	63.91%	79.20%	72.71%	93.56%	49.37%	87.19%	74.95%
Att-UNet[23]	77.77	36.02	89.55%	68.88%	77.98%	71.11%	93.57%	58.04%	87.30%	75.75%
R50 ViT[20]	71.29	32.87	73.73%	55.13%	75.80%	72.20%	91.51%	45.99%	81.99%	73.95%
TransUNet[20]	77.48	31.69	87.23%	63.13%	81.87%	77.02%	94.08%	55.86%	85.08%	75.62%
TransNorm[66]	78.40	30.25	86.23%	65.10%	82.18%	78.63%	94.22%	55.34%	89.50%	76.01%
Swin U–Net[24]	79.13	21.55	85.47%	66.53%	83.28%	79.61%	94.29%	56.58%	90.66%	76.60%
TransDeepLab[67]	80.16	21.25	86.04%	69.16%	84.08%	79.88%	93.53%	61.19%	89.00%	78.40%
UCTransNet[68]	78.23	26.75	–	–	–	–	–	–	–	–
MT–UNet[69]	78.59	26.59	87.92%	64.99%	81.47%	77.29%	93.06%	59.46%	87.75%	76.81%
MEW–UNet[70]	78.92	**16.44**	86.68%	65.32%	82.87%	80.02%	93.63%	58.36%	90.19%	74.26%
VM-UNet[28]	81.08	19.21	86.40%	69.41%	86.16%	82.76%	94.17%	58.80%	89.51%	81.40%
HC-Mamba[29]	81.56	26.32	**90.92%**	69.65%	85.57%	79.27%	**97.38%**	54.08%	**93.49%**	80.14%
CCViM(Ours)	**82.65**	17.83	87.63%	68.45%	**86.23%**	**83.22%**	94.67%	**67.12%**	92.05%	**81.82%**

Table 3. Result comparisons with state–of–the–art methods on the Synapse[35] dataset ("–" represents the indicator has not been measured)
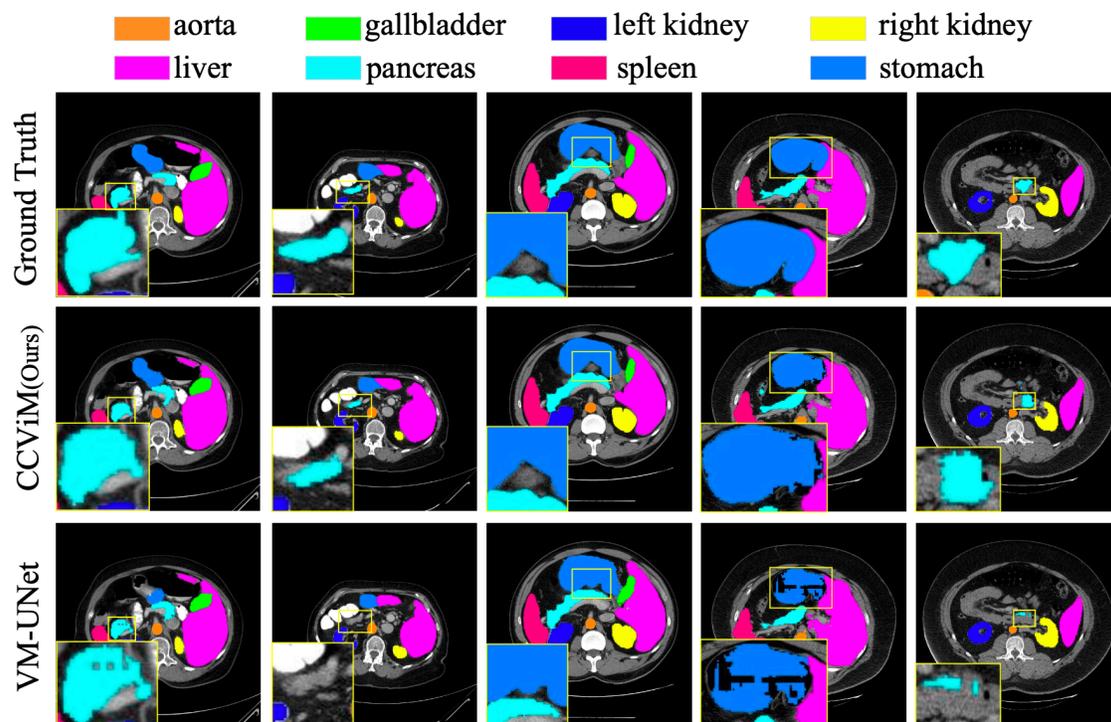
Figure 7. Visualizations on the Synapse[35] Dataset. Compare with VM-UNet[28], our model is closer to ground truth, which demonstrates our CCViM has superior performance.

E. Ablation Analysis

Superiority of the context clustering (CC) layer. To assess the effectiveness of our CC layer, we conduct the ablation experiments on the recent state-of-the-art models LocalVIM[26] and LocalVMamba[26] using Kumar and ISIC17 datasets. The local scanning strategies in LocalVIM[26] and LocalVMamba[26] can effectively capture local dependencies while maintaining a global perspective, however, the scanning strategies rely on fixed propagation trajectories, which cannot adaptively capture local features and ignore the spatial context information. Our CC layer can cluster the local features and capture spatial context information in an adaptive way. Therefore, we exchange the local scanning strategies in LocalVIM[26] and LocalVMamba[26] with our CC layer, and the results are shown in Table 4. On Kumar, our CC layer has improved the PQ, Dice, AJI, DQ, and SQ metrics by 1.57%, 0.15%, 1.68%, 1.75% and 0.39% respectively, compared to LocalVIM[26]. And our CC layer has improved the PQ, Dice, AJI, DQ, and SQ metrics by 0.49%, 0.26%, 0.34%, 1.26% and −0.09% respectively, compared to LocalVMamba[26]. On the ISIC17 dataset, our CC layer has improved the mIoU and DSC metrics by 0.29% and 0.18% respectively, compared to LocalVIM[26]. Besides, our CC layer has improved the mIoU and DSC metrics by 2.68% and 1.65% respectively, compared to LocalVMamba[26]. The improvements demonstrate the superiority of our CC layer in dynamically capturing local features and spatial information.

	Methods	PQ(%)↑	Dice(%)↑	AJI(%)↑	DQ(%)↑	SQ(%)↑
Kumar[31]	LocalVIM[26]	49.15	79.02	53.28	67.42	72.65
	LocalVIM-CC	50.72	79.17	54.96	69.17	73.04
	LocalVMamba[26]	58.02	82.23	61.08	75.32	**76.72**
	LocalVMamba-CC	**58.51**	**82.49**	**61.42**	**76.58**	76.63
	Methods	mIoU(%)↑	DSC(%)↑	Acc(%)↑	Spe(%)↑	Sen(%)↑
ISIC17[33]	LocalVIM[26]	77.94	87.60	95.90	97.78	86.54
	LocalVIM-CC	78.23	87.78	96.04	98.24	85.08
	LocalVMamba[26]	78.70	88.08	96.07	**97.97**	86.62
	LocalVMamba-CC	**81.38**	**89.73**	**96.56**	97.93	**89.75**

Table 4. The Superiority of Our CC on the Kumar[31] and ISIC17[33] datasets

Hyper-parameter analysis. Given the different sizes and numbers of the given images, we adopt two different CC layers. One CC layer method has 4 cluster centers in a local region, another one has 25 cluster centers in a local region. Besides, there are four different scan directions from which to choose. Therefore, in this section, we perform comprehensive experiments on Kumar[31] and ISIC17[33] to thoroughly analyze the performance of various scanning directions and CC centers. As shown in Table 5, the term "h-hflip" indicates that only apply the horizontal scan and horizontal flipped scan in each CCS6 layer. This configuration, which uses only scanning directions without incorporating CC layers, allows us to evaluate the effect of global scanning alone. The term "h-hflip-C_4" denotes applying the horizontal scan, horizontal flipped scan, and CC with 4 cluster centers in each CCS6 layer. In this configuration, a CC layer with 4 cluster centers is added to the global scanning directions. This enables a comparison of how the inclusion of a local CC layer enhances performance. Similarly, the term "h-hflip-C_{25}" represents applying the horizontal scan, horizontal flipped scan, and CC with 25 cluster centers in each CCS6 layer. This configuration explores the effect of increasing the number of cluster centers in the local CC layer while keeping the global scanning directions fixed. The term "h-hflip-C_4-C_{25}" represents applying the horizontal scan, horizontal flipped scan, CC with 4 cluster centers, and CC with 25 cluster centers in each CCS6 layer. By combining both CC layers (with 4 and 25 cluster centers), this configuration allows us to examine the performance of the combined effect of multiple clustering operations. The term "v-vflip" represents only applying the vertical scan and vertical flipped scan in each

CCS6 layer. The following can be analogized. This configuration provides a clearer basis for comparison, allowing us to understand the contributions of global scanning and local clustering without introducing additional variables at each stage. Although further configuration tuning may lead to performance improvements, such adjustments would be redundant and unnecessary. Our focus is on evaluating the effectiveness of different scanning directions and CC layers with varying numbers of cluster centers. Given the complexity of multiple stages, introducing additional configurations would overcomplicate the analysis. On Kumar[31], we observe that incorporating CC with 4 or 25 cluster centers improves results compared to using "h–hflip" or "v–vflip" alone. Moreover, combining CC with both 4 and 25 cluster centers yields significant performance improvements. Similarly, on the ISIC17 dataset, the performance has improved after incorporating CC with 4 or 25 cluster centers into each CCS6 layer of the "h–hflip" or "v–vflip" configurations. The "h–hflip-C_4-C_{25}" configuration yielded relatively poorer results compared to other CC combinations. However, most configurations involving CC combinations achieve satisfactory results. Overall, combining CC with both 4 and 25 cluster centers demonstrated superior performance. Notably, the "hflip–vflip–C_4-C_{25}" configuration even outperformed our CCViM on ISIC17[33]. These superior results demonstrate that the local feature interactions and dynamically capturing spatial information are significant in MedISeg, which also demonstrates our CC is effective.

	Setting	PQ(%)↑	Dice(%)↑	AJI(%)↑	DQ(%)↑	SQ(%)↑
Kumar[31]	h–hflip	56.32	81.90	60.03	74.11	75.68
	h–hflip–C_4	57.84	81.93	59.95	75.36	76.49
	h–hflip–C_{25}	57.91	82.09	60.46	75.34	76.61
	h–hflip–C_4–C_{25}	58.14	82.34	60.45	75.72	76.57
	v–vflip	57.37	81.88	59.72	74.84	76.35
	v–vflip–C_4	57.06	81.92	60.09	74.89	75.95
	v–vflip–C_{25}	57.84	81.97	60.63	75.68	76.14
	v–vflip–C_4–C_{25}	58.49	82.26	60.86	75.96	76.73
	h–hflip–v–vflip	56.59	81.89	60.00	74.27	75.85
	h–v–C_4–C_{25}	57.97	**82.57**	60.80	75.83	76.21
	h–vflip–C_4–C_{25}	58.83	82.32	61.40	76.43	76.70
	hflip–v–C_4–C_{25}	58.23	82.40	61.38	76.17	76.11
	hflip–vflip–C_4–C_{25}	58.49	82.37	60.95	76.21	76.44
	CCViM$_{(Ours)}$	**58.83**	82.48	**61.38**	**76.50**	**76.63**

	Setting	mIoU(%)↑	DSC(%)↑	Acc(%)↑	Spe(%)↑	Sen(%)↑
ISIC17[33]	h–hflip	80.21	89.01	96.27	97.5	**90.14**
	h–hflip–c4	80.91	89.45	96.49	98.03	88.83
	h–hflip–C_{25}	80.86	89.42	96.46	97.92	89.2
	h–hflip–C_4–C_{25}	80.36	89.11	96.41	98.16	87.74
	v–vflip	80.38	89.12	96.42	98.23	87.44
	v–vflip–C_4	80.9	89.44	96.44	97.72	90.03
	v–vflip–C_{25}	81.14	89.59	96.53	98.06	88.96
	v–vflip–C_4–C_{25}	80.83	89.40	96.47	98.00	88.87
	h–hflip–v–vflip	80.23	89.03	96.29	97.58	89.90
	h–v–C_4–C_{25}	80.69	89.31	96.46	98.11	88.29
	h–vflip–C_4–C_{25}	80.66	89.30	96.48	98.26	87.63

	Setting	PQ(%)↑	Dice(%)↑	AJI(%)↑	DQ(%)↑	SQ(%)↑
	hflip-v-C_4-C_{25}	81.29	89.68	96.62	98.38	87.83
	hflip-vflip-C_4-C_{25}	**81.48**	**89.80**	**96.66**	**98.46**	87.74
	CCViM$_{(Ours)}$	81.40	89.74	96.60	98.19	88.7

Table 5. The results of different cluster centers on Kumar[31] and ISIC17[33] datasets

Efficiency analysis. We conduct a comparative analysis of our model's parameters and FLOPs against state-of-the-art models, ensuring consistent comparisons by evaluating each model with an input image size of (1,3,256,256). To measure inference speed in a realistic scenario, we use the throughput metric (frames per second, fps), which reflects the average inference speed over the entire test set, calculated across all test images. Additionally, we present results from the Kumar[31] dataset, comparing the PQ and Dice metrics. As shown in Table 6, Mamba-based models, such as VM-UNet[28] and LocalVMamba[26], demonstrate superior efficiency compared to CNN-based models (i.e., U-Net[39]) and Transformer-based models (i.e., TransUnet[20] and Swin U-Net[24]), offering a balance of lower parameters and FLOPs. Our CCViM model achieves the best performance compared to other methods, as it can extract both global information and capture local spatial features dynamically. Moreover, our CCViM achieves competitive throughput, with an inference speed of 41.16 fps, compared to UNet (i.e., 34.86 fps), TransUnet (i.e., 36.84 fps), Swin U-Net (i.e., 37.51 fps), VM-UNet (i.e., 41.54 fps), and LocalVMamba (i.e., 41.01 fps). The competitive FPS of CCViM can be primarily attributed to the local CC operation. Instead of using traditional convolution or attention mechanisms to extract local features, CCViM uses a context cluster algorithm to aggregate local spatial information dynamically, operating on local windows rather than the entire image. This localized approach significantly reduces computational cost since clustering is performed within small patches rather than over the entire image, which is computationally expensive. Although the CC layer in CCViM introduces a slight increase in FLOPs compared to VM-UNet[28] and LocalVMamba[26], this does not notably impact the inference time, allowing CCViM to achieve a throughput that aligns closely with VM-UNet[28] and LocalVMamba[26],while demonstrating effective segmentation performance.

Methods	Params↓	FLOPs↓	Thru.↑	PQ↑	Dice↑
	(M)	(G)	(fps)	(%)	(%)
UNet[39]	31.04	54.75	34.86	47.80	75.80
TransUnet[20]	91.52	29.19	36.84	48.14	77.76
Swin U-Net[24]	27.15	7.73	37.51	47.46	80.11
VM-UNet[28]	**22.05**	**4.12**	**41.54**	56.59	81.89
LocalVMamba[26]	25.20	4.21	41.01	58.02	82.23
CCViM$_{(Ours)}$	23.56	4.45	41.16	**58.83**	**82.48**

Table 6. Model efffciency comparison regarding parameter number (M) and FLOPs (G)

F. Limitation Analysis

Although extensive experiments demonstrate the effectiveness of our method in MedISeg tasks, there are two notable limitations. First, while our CC layer may introduce minimal computational costs, this slight addition contributes to promising performance improvements. Second, different medical images may exhibit small lesions and irregular boundaries, which can affect the details captured by our CC layer's local extraction. Additionally, varying scanning directions and CC layers may influence the performance of MedISeg tasks. However, our current configurations for scanning directions and CC layers are fixed, underscoring the need for more adaptive strategies. As shown in Fig. 8, there are several failure examples from the ISIC18[34] dataset. We can observe that the fixed configuration of global scan directions and local CC layers may impact performance, particularly in capturing the details of irregular boundaries and complex structures. The model struggles to replicate the intricate features represented in the ground truth, leading to discrepancies in segmentation results. An adaptive configuration could enhance our model's ability to discern these features more effectively. Given that our CC layers utilize different clustering centers, varying clustering centers can capture different local details. Furthermore, different scan directions can facilitate varying global interactions, which also affect performance in MedISeg tasks. Adopting a more flexible approach could significantly improve segmentation accuracy by allowing the model to better adapt to the unique characteristics of each medical image. Therefore, developing more adaptive configurations for scan directions and CC layers could enhance performance in both global and local extraction. In the future, we will focus on implementing these adaptive strategies, exploring dynamic configuration algorithms that can adjust in real-time based on the input data characteristics.

Ground Truth | CCViM(Ours)
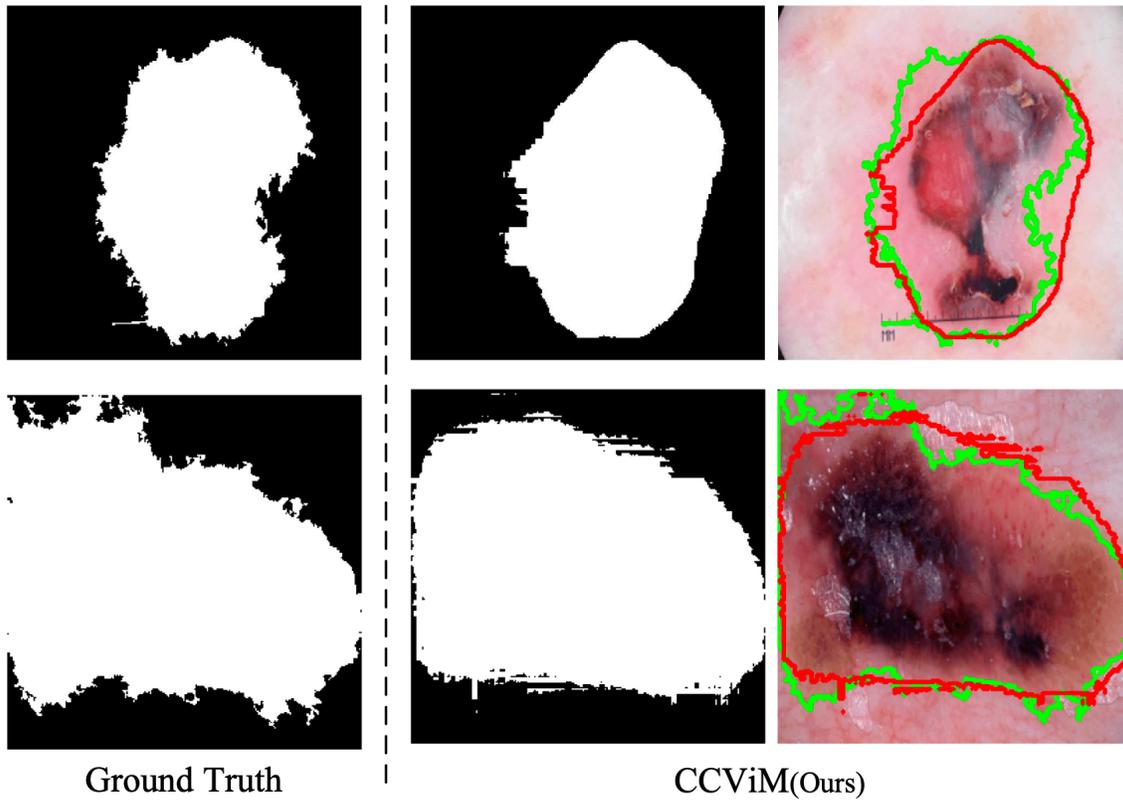
Figure 8. Visualizations of some failure examples on ISIC18[34] dataset.

V. Conclusion

In this paper, we introduce CCViM, a U-shaped architecture designed for medical image segmentation that inherits the efficiency and effectiveness of Mamba. The proposed CC layer partitions feature into distinct windows for learnable local clustering, dynamically capturing spatial contextual information. Based on the CC layer, our CCS6 layer combines the proposed CC with traditional global scanning strategies, significantly enabling our model to capture both local and global information. We compare our CCViM with state-of-the-art models on nuclei segmentation, skin lesion segmentation, and multi-organ segmentation datasets, comprehensive experiments demonstrate the promising performance of our model in medical image segmentation. In the future, we will explore effective searching methods that combine traditional scanning strategies and our CCs in a computation-free way. Furthermore, we will explore the application of Mamba-based methods in diverse medical image recognition tasks, such as registration and reconstruction.

References

1. a, b Wang S, Li C, Wang R, Liu Z, Wang M, Tan H, Wu Y, Liu X, Sun H, Yang R, et al. (2021). "Annotation-efficient deep learning for automatic medical image segmentation." *Nat. Commun.* **12** (1): 5915.

2. a, b Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, Van Der Laak JA, Van Ginneken B, S\u00e1nchez CI. "A survey on deep learning in medical image analysis." *Med Image Anal.* **42**: 60--88, 2017.

3. a, b Wang R, Chen S, Ji C, Fan J, Li Y (2022). "Boundary-aware context neural network for medical image segmentation". *Med. Image Anal..* **78**: 102395.

4. ^ Zou W, Qi X, Zhou W, Sun M, Sun Z, Shan C (2022). "Graph flow: Cross-layer graph flow distillation for dual efficient medical image segmentation". *IEEE Trans. Med. Imag..* **42** (4): 1159--1171.

5. a, b Bai W, Suzuki H, Huang J, Francis C, Wang S, Tarroni G, Guitton F, Aung N, Fung K, Petersen SE, et al. "A population-based phenome-wide association study of cardiac and aortic structure and function." *Nat. Med.* **26** (10): 1654--1662, 2020.

6. a, b Mei X, Lee HC, Diao KY, Huang M, Lin B, Liu C, Xie Z, Ma Y, Robson PM, Chung M, et al. (2020). "Artificial intelligence--enabled rapid diagnosis of patients with COVID-19." *Nat. Med..* **26** (8): 1224--1228.

7. ^ Tang H, Chen X, Liu Y, Lu Z, You J, Yang M, Yao S, Zhao G, Xu Y, Chen T, et al. "Clinically applicable deep learning framework for organs at risk delineation in CT images." *Nat. Mach. Intell.* **1** (10): 480--491, 2019.

8. ^ Khened M, Kollerathu VA, Krishnamurthi G (2019). "Fully convolutional multi-scale residual DenseNets for cardiac segmentation and automated cardiac diagnosis using ensemble of classifiers". *Med. Image Anal..* **51**: 21–45.

9. a, b, c, d, e, f, g, h, i Graham S, Vu QD, Raza SEA, Azam A, Tsang YW, Kwak JT, Rajpoot N (2019). "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images". *Med. Image Anal..* **58**: 101563.

10. ^ Sun M, Zou W, Wang Z, Wang S, Sun Z (2023). "An automated framework for histopathological nucleus segmentation with deep attention integrated networks." *IEEE-ACM Trans. Comput. Biol. Bioinform..*

11. ^ Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017). "Dermatologist-level classification of skin cancer with deep neural networks". *nature.* **542** (7639): 115--118.

12. ^ Gibson E, Giganti F, Hu Y, Bonmati E, Bandula S, Gurusamy K, Davidson B, Pereira SP, Clarkson MJ, Barratt DC (2018). "Automatic multi-organ segmentation on abdominal CT with dense V-networks". *IEEE Trans. Med. Imag..* **37** (8): 1822--1834.

13. ^ Qi X, Wu Z, Zou W, Ren M, Gao Y, Sun M, Zhang S, Shan C, Sun Z (2024). "Exploring generalizable distillation for efficient medical image segmentation". *IEEE J. Biomed. Health. Inf..*

14. a, b Zhang D, Zhang H, Tang J, Wang M, Hua X, Sun Q (2020). "Feature pyramid transformer." In: *ECCV.* pp. 323–339.

15. ^ Yan R, Xie L, Shu X, Zhang L, Tang J (2023). "Progressive instance-aware feature learning for compositional action recognition." *IEEE Trans. Pattern Anal. Mach. Intell..* **45**(8): 10317–10330.

16. ^_Yan R, Xie L, Tang J, Shu X, Tian Q (2020). "HiGCIN: Hierarchical graph-based cross inference network for group activity recognition". IEEE Trans. Pattern Anal. Mach. Intell.. **45** (6): 6955--6968._

17. a, b, c_He K, Zhang X, Ren S, Sun J. "Deep residual learning for image recognition." In: CVPR; 2016. p. 770–778._

18. a, b_Wang W, Dai J, Chen Z, Huang Z, Li Z, Zhu X, Hu X, Lu T, Lu L, Li H, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In: CVPR; 2023. p. 14408-14419._

19. a, b_Howard AG, 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._

20. a, b, c, d, e, f, g, h, i, j, k, l, m, n, o_Chen J, Lu Y, Yu Q, Luo X, Adeli E, Wang Y, Lu L, Yuille AL, Zhou Y (2021). "Transunet: Transformers make strong encoders for medical image segmentation." arXiv. 2021._

21. ^_Zhang D, Zhang L, Tang J (2023). "Augmented fcn: rethinking context modeling for semantic segmentation." Science China Information Sciences. **66** (4): 142105._

22. a, b_Peng Z, Huang W, Gu S, Xie L, Wang Y, Jiao J, Ye Q (2021). "Conformer: Local features coupling global representations for visual recognition." In: CVPR. pp. 367–376._

23. a, b, c_Oktay O, Schlemper J, Le Folgoc L, Lee M, Heinrich M, Misawa K, Mori K, McDonagh S, Hammerla NY, Kainz B, Glocker B, Rueckert D (2018). "Attention U-Net: Learning Where to Look for the Pancreas." In: MIDL._

24. a, b, c, d, e, f, g, h, i, j, k_Cao H, Wang Y, Chen J, Jiang D, Zhang X, Tian Q, Wang M (2022). "Swin-unet: Unet-like pure transformer for medical image segmentation." In: ECCV. pp. 205–218._

25. a, b, c, d, e, f, g_Liu Y, Tian Y, Zhao Y, Yu H, Xie L, Wang Y, Ye Q, Liu Y. "Vmamba: Visual state space model." arXiv. 2024._

26. a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z_Huang T, Pei X, You S, Wang F, Qian C, Xu C (2024). "Localmamba: Visual state space model with windowed selective scan". arXiv._

27. a, b, c, d, e, f, g, h_Gu A, Dao T (2023). "Mamba: Linear-time sequence modeling with selective state spaces". arXiv._

28. a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, ^_Ruan J, Xiang S (2024). "Vm-unet: Vision mamba unet for medical image segmentation." arXiv. _arXiv.__

29. a, b, c, d, e_Xu J. "HC-Mamba: Vision MAMBA with Hybrid Convolutional Techniques for Medical Image Segmentation." arXiv. 2024._

30. a, b_Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." In: ICLR._

31. a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s_Kumar N, Verma R, Sharma S, Bhargava S, Vahadane A, Sethi A (2017). "A dataset and a technique for generalized nuclear segmentation for computational pathology". IEEE Trans. Med. Imag.. **36** (7): 1550--1560._

32. a, b, c, d, e, f, g, h, i, j, k, l_Vu QD, Graham S, Kurc T, To MNN, Shaban M, Qaiser T, Koohbanani NA, Khurram SA, Kalpathy-Cramer J, Zhao T, et al. Methods for segmentation and classification of digital microscopy tissue images. Front. Bioeng. Biotechnol.. **7**: 433738, 2019._

33. a, b, c, d, e, f, g, h, i, j, k, l, m, n, o*Berseth M. "ISIC 2017-skin lesion analysis towards melanoma detection." arXiv. 2017.*

34. a, b, c, d, e, f, g, h, i, j, k, l*Codella N, Rotemberg V, Tschandl P, Celebi ME, Dusza S, Gutman D, Helba B, Kalloo A, Liopyris K, Marchetti M, et al. "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)." arXiv. 2019.*

35. a, b, c, d, e, f, g*Landman B, Xu Z, Igelsias J, Styner M, Langerak T, Klein A (2015). "Miccai multi-atlas labeling beyond the cranial vault--workshop and challenge." In: MICCAI Workshop Challenge. 5: 12.*

36. a, b*Zhang D, Zuo C, Wu Q, Fu L, Xiang X (2022). "Unabridged adjacent modulation for clothing parsing". Pattern Recognit.. **127**: 108594.*

37. ^*Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. "Attention is all you need." In: NeurIPS. 2017; **30**:5998-6008.*

38. ^*Zhang D, Lin Y, Chen H, Tian Z, Yang X, Tang J, Cheng KT (2022). "Understanding the tricks of deep learning in medical image segmentation: Challenges and future directions." arXiv. 2022.*

39. a, b, c, d, e, f, g, h, i, j, k*Ronneberger O, Fischer P, Brox T (2015). "U-net: Convolutional networks for biomedical image segmentation." In: MICCAI. pp. 234–241.*

40. a, b*Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J (2018). "Unet++: A nested u-net architecture for medical image segmentation." In: MICCAI. pp. 3–11.*

41. ^*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: ICCV. pp. 10012–10022.*

42. ^*Wang C, Xu H, Zhang X, Wang L, Zheng Z, Liu H (2022). "Convolutional embedding makes hierarchical vision transformer stronger." In: ECCV. pp. 739–756.*

43. ^*Shi Y, Jia Y, Zhang X (2024). "FocusDet: an efficient object detector for small object". Sci. Rep.. **14** (1): 10697.*

44. ^*Olson CF, Huttenlocher DP (1997). "Automatic target recognition by matching oriented edge pixels". IEEE Trans. Image Process.. **6** (1): 103–113.*

45. ^*Zhang D, Zhang H, Tang J, Hua XS, Sun Q (2020). "Causal intervention for weakly-supervised semantic segmentation." In: NeurIPS. pp. 655–666.*

46. ^*Gao Y, Zhou M, Metaxas DN. "UTNet: a hybrid transformer architecture for medical image segmentation." In: MICCAI; 2021. p. 61-71.*

47. ^*Huang X, Deng Z, Li D, Yuan X, Fu Y (2022). "Missformer: An effective transformer for 2d medical image segmentation". IEEE Trans. Image Process.. **42** (5): 1484--1494.*

48. a, b, c*Ma X, Zhou Y, Wang H, Qin C, Sun B, Liu C, Fu Y (2023). "Image as set of points." In: ICLR, 2023.*

49. a, b*Zhu L, Liao B, Zhang Q, Wang X, Liu W, Wang X. "Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model." In: ICML; 2024. p. 62429-62442.*

50. ^*Park J, Park J, Xiong Z, Lee N, Cho J, Oymak S, Lee K, Papailiopoulos D. "Can Mamba Learn How To Learn? A Comparative Study on In-Context Learning Tasks." In: ICML; 2024.*

51. ^Glorioso P, Anthony Q, Tokpanov Y, Whittington J, Pilault J, Ibrahim A, Millidge B (2024). "Zamba: A Compact 7B SSM Hybrid Model". arXiv.

52. ^Lieber O, Lenz B, Bata H, Cohen G, Osin J, Dalmedigos I, Safahi E, Meirom S, Belinkov Y, Shalev-Shwartz S, et al. "Jamba: A hybrid transformer-mamba language model." arXiv. 2024.

53. a, bMilletari F, Navab N, Ahmadi S-A (2016). "V-net: Fully convolutional neural networks for volumetric medical image segmentation." In: 3DV. pp. 565–571.

54. ^Liu B, Dolz J, Galdran A, Kobbi R, Ayed IB (2024). "Do we really need dice? The hidden region-size biases of segmentation losses." Med. Image Anal.. 91: 103015.

55. a, b, c, dRuan J, Xiang S, Xie M, Liu T, Fu Y. "MALUNet: A multi-attention and light-weight unet for skin lesion segmentation." In: BIBM; 2022. p. 1150–1156.

56. ^Zhu Q, Fang Y, Cai Y, Chen C, Fan L (2024). "Rethinking scanning strategies with vision mamba in semantic segmentation of remote sensing imagery: An experimental study." arXiv.

57. ^Loshchilov I, Hutter F (2019). "Decoupled weight decay regularization." In: ICLR.

58. ^Loshchilov I, Hutter F (2017). "SGDR: Stochastic Gradient Descent with Warm Restarts." In: ICLR.

59. ^Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009). "Imagenet: A large-scale hierarchical image database." In: CVPR. pp. 248–255.

60. a, bBadrinarayanan V, Kendall A, Cipolla R (2017). "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." IEEE Trans. Pattern Anal. Mach. Intell.. 39 (12): 2481–2495.

61. a, b, c, dNaylor P, La{\'e} M, Reyal F, Walter T (2018). "Segmentation of nuclei in histopathology images by deep regression of the distance map". IEEE Trans. Med. Imag.. 38 (2): 448--459.

62. a, b, cGao Y, Zhou M, Liu D, Metaxas DN (2022). "A Multi-scale Transformer for Medical Image Segmentation: Architectures, Model Efficiency, and Benchmarks." arXiv. 2022.

63. a, b, cZhang Y, Liu H, Hu Q (2021). "Transfuse: Fusing transformers and cnns for medical image segmentation." In: MICCAI. pp. 14–24.

64. ^Wei J, Hu Y, Zhang R, Li Z, Zhou SK, Cui S (2021). "Shallow attention network for polyp segmentation." In: MICCAI. pp. 699–708.

65. ^Alom MZ, Yakopcic C, Hasan M, Taha TM, Asari VK (2019). "Recurrent residual U-Net for medical image segmentation." J. Med. Imaging. 6(1): 014006.

66. ^Azad R, Al-Antary MT, Heidari M, Merhof D (2022). "Transnorm: Transformer provides a strong spatial normalization mechanism for a deep segmentation model". IEEE Access. 10: 108205–108215.

67. ^Azad R, Heidari M, Shariatnia M, Aghdam EK, Karimijafarbigloo S, Adeli E, Merhof D (2022). "Transdeeplab: Convolution-free transformer-based deeplab v3+ for medical image segmentation." In: PRIME Workshop. 2022. p. 91–102.

68. ^Wang H, Cao P, Wang J, Zaiane OR (2022). "Uctransnet: rethinking the skip connections in u-net from a channel-wise perspective with transformer." In: AAAI. 36 (3): 2441--2449.

69. ^Wang H, Xie S, Lin L, Iwamoto Y, Han XH, Chen YW, Tong R (2022). *"Mixed transformer u-net for medical image segmentation." In: ICASSP. pp. 2390–2394.*

70. ^Ruan J, Xie M, Xiang S, Liu T, Fu Y (2022). *"MEW-UNet: Multi-axis representation learning in frequency domain for medical image segmentation". arXiv. 2022.*

Declarations

Potential competing interests: No potential competing interests to declare.