VM-UNet-ASPP: Vision Mamba UNet with Atrous Spatial Pyramid Pooling for Multi-Feature Extraction in Grain Segmentation

Mahir Shahriar Tamim
Department of Electrical and Computer Engineering (ECE)
North South University (NSU)
Dhaka, Bangladesh
mahir.tamim@northsouth.edu

Fuwad Hasan
Department of Electrical and Computer Engineering (ECE)
North South University (NSU)
Dhaka, Bangladesh
fuwad.hasan@northsouth.edu

Meharun Nesa
Department of Electrical and Computer Engineering (ECE)
North South University (NSU)
Dhaka, Bangladesh
meharun.nesa@northsouth.edu

***Abstract*—Segmentation of grains in materials like stainless steel is crucial for various industrial applications, necessitating precise and efficient methods to accurately delineate diverse grain structures. This paper presents a novel enhancement of the VMUNet architecture through the integration of Atrous Spatial Pyramid Pooling (ASPP) to significantly improve segmentation accuracy. By incorporating ASPP, our model captures multi-scale features, providing a broader and more detailed field of view without losing resolution. This capability is essential for accurately segmenting grains of different scales and shapes in stainless steel. Additionally, this study evaluates the performance of various architectures, including the SAM model by Meta AI, to benchmark and validate the effectiveness of the proposed approach. Extensive experiments demonstrate the robustness and superiority of the enhanced VMUNet with ASPP in handling complex stainless steel grain segmentation tasks, significantly outperforming traditional models in terms of precision and computational efficiency. Our results underline the model's capability to maintain high performance even under challenging conditions, paving the way for more reliable and versatile segmentation solutions in material science applications.**

I Introduction

Grain segmentation in materials such as stainless steel is crucial for industrial applications like quality control and material characterization. Accurate delineation of grain structures is essential for understanding material properties. Traditional methods often struggle with the required accuracy and efficiency for complex grain structures.

Recent advancements in machine learning, especially neural networks, have significantly improved image segmentation. The Vision Mamba UNet (VM-UNet) architecture has shown promise due to its ability to capture extensive contextual information and model long-range interactions. However, the complexity of stainless steel grain structures necessitates further enhancements.

This paper presents a novel improvement to VMUNet through the integration of Atrous Spatial Pyramid Pooling (ASPP). ASPP captures multi-scale features, providing a broader and more detailed view without sacrificing resolution. This is crucial for segmenting grains of varying scales and shapes. By incorporating ASPP, the enhanced VMUNet model handles intricate grain boundaries more effectively, leading to precise segmentation results.

To benchmark our approach, we evaluated various architectures, including the Segment Anything Model (SAM) by Meta AI. Extensive experiments demonstrate the robustness and superiority of the enhanced VMUNet with ASPP in complex stainless steel grain segmentation tasks. Our results show significant improvements in precision and computational efficiency over traditional models.

II Our Contribution

In this study, we systematically evaluated the performance of several state-of-the-art segmentation architectures, including such as U-Net, SegNet, U-Net++, ResNet and so on. We then explored more advanced models, such as:

- Segment Anything Model (SAM)
- SAM LoRA (parameter-efficient variant of SAM)

Subsequently, we experimented with:

- VM-Unet

These experiments provided valuable insights into the importance of scale-awareness in segmentation tasks.

Main Contribution

Our primary contribution lies in the novel enhancement of the Vision Mamba UNet architecture:

- Integration of the Atrous Spatial Pyramid Pooling (ASPP) block, as shown in Figure 1

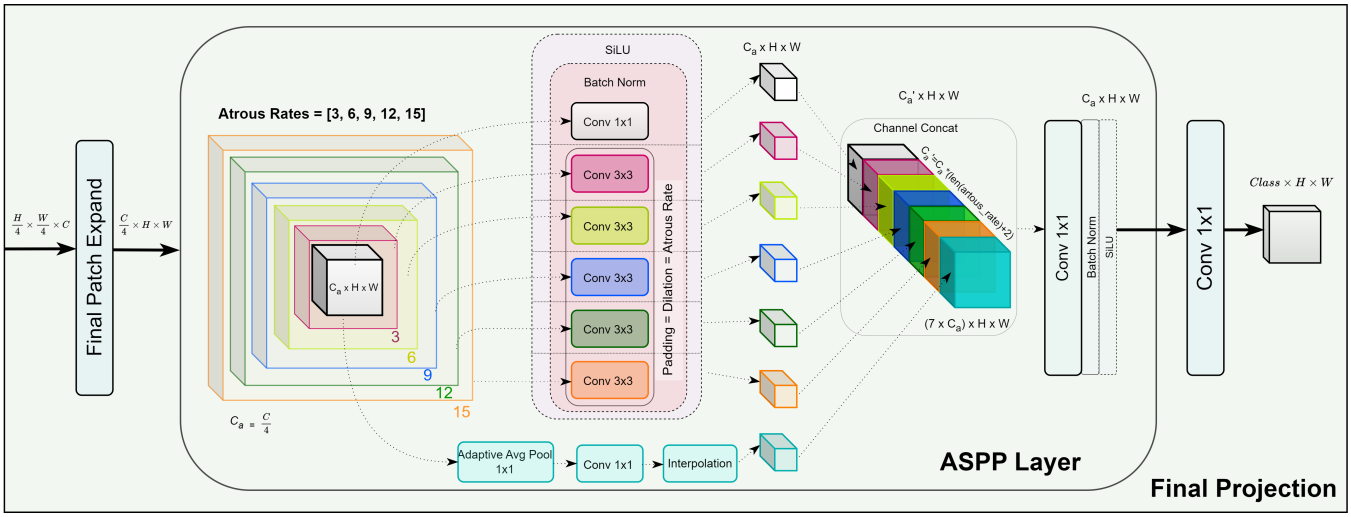

Fig. 1. **ASPP Block**. Enhanced Precision with ASPP: Leveraging Atrous Spatial Pyramid Pooling (ASPP), Vision Mamba UNet captures intricate grain details across multiple scales, enhancing segmentation accuracy in stainless steel microstructures. This integration improves precision and robustness in material science applications, enabling high-resolution, multi-scale grain segmentation.

- The ASPP block captures multi-scale features, providing a broader and more detailed field of view without sacrificing resolution.
- This capability is crucial for accurately segmenting grains of varying scales and shapes, characteristic of stainless steel microstructures.

Key results from our experiments:

- The enhanced VMUNet with ASPP demonstrated robustness and superiority in handling complex stainless steel grain segmentation tasks.
- The proposed model significantly outperforms traditional segmentation models in terms of precision and computational efficiency.
- The integration of ASPP into the VMUNet architecture not only improves segmentation accuracy but also maintains high performance under challenging conditions.

This advancement paves the way for more reliable and versatile segmentation solutions in material science applications.

III Literature Review

Grain Segmentation Using Machine Learning

[1] reports significant improvements in grain boundary segmentation accuracy using Convolutional Neural Networks (CNNs) trained on both real and generated data. The study highlights the use of synthetic data generated via Voronoi tessellation patterns to train machine learning models, achieving the accuracy of manual segmentation with the efficiency of computational methods.

U-Net: Convolutional Networks for Biomedical Image Segmentation

[2] introduces U-Net, a deep learning architecture designed for biomedical image segmentation. U-Net utilizes extensive data augmentation to make the most of limited annotated samples, with a contracting path for context capture and an expanding path for precise localization. Its success in various biomedical segmentation challenges highlights its potential applicability to segmentation tasks.

Segment Anything

[3] presents the Segment Anything Model (SAM), a versatile segmentation framework trained on a vast dataset with over 1 billion masks across 11 million images. SAM is designed to be promptable and transfer zero-shot to new image distributions and tasks. While not specifically developed for grain segmentation, its generalist approach could be adapted for this purpose.

Convolution Meets LoRA

[4] proposes Conv-LoRA, a parameter-efficient fine-tuning approach for the Segment Anything Model (SAM). By integrating lightweight convolutional parameters into Low-Rank Adaptation (LoRA), Conv-LoRA enhances SAM's performance in specialized domains like medical imagery and remote sensing, potentially making it suitable for segmentation tasks.

DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

[5] addresses semantic image segmentation with DeepLab, making three main contributions: highlighting atrous convolution for dense prediction tasks, proposing atrous spatial pyramid pooling (ASPP) for robust multi-scale object segmentation, and improving object boundary localization by combining Deep Convolutional Neural Networks (DCNNs) and Conditional Random Fields (CRFs). DeepLab's approach allows for control over feature resolution and field of view, enhancing the model's ability to capture context without increasing parameters or computation.

VM-UNet: Vision Mamba UNet for Medical Image Segmentation

[6] introduces VM-UNet, a U-shaped architecture for medical image segmentation based on state space models (SSMs). VM-UNet combines the strengths of CNNs and SSMs, enabling efficient modeling of long-range interactions with linear computational complexity. The Visual State Space (VSS) block captures extensive contextual information, making it competitive for detailed segmentation tasks.

VM-UNet-V2: Rethinking Vision Mamba UNet for Medical Image Segmentation

[7] refines VM-UNet with VM-UNet-V2, incorporating multi-scale feature aggregation and attention mechanisms to enhance segmentation performance. The Semantics and Detail Infusion (SDI) mechanism in VM-UNet-V2 effectively combines low-level and high-level features, demonstrating competitive performance in various medical image segmentation benchmarks.

IV Preliminaries

Structured State Space Sequence Models (S4) [8] and Mamba [9], which are modern SSM-based models, transform a one-dimensional input sequence $u(t) \in \mathbb{R}$ through intermediate state vectors $\mathbf{h}(t) \in \mathbb{R}^N$ to an output $v(t) \in \mathbb{R}$. This transformation can be modeled by the following linear Ordinary Differential Equation (ODE):

$$\begin{aligned} \frac{d\mathbf{h}(t)}{dt} &= \mathbf{A}\mathbf{h}(t) + \mathbf{B}u(t) \\ v(t) &= \mathbf{C}\mathbf{h}(t) \end{aligned} \tag{1}$$

Here, $\mathbf{A} \in \mathbb{R}^{N\times N}$ represents the state transition matrix, $\mathbf{B} \in \mathbb{R}^{N\times 1}$ is the input matrix, and $\mathbf{C} \in \mathbb{R}^{1\times N}$ is the output matrix.

To make this continuous model suitable for deep learning applications, S4 and Mamba discretize it by introducing a time step Δt. The discretized versions of $\mathbf{A}$ and $\mathbf{B}$, denoted as $\mathbf{A}_d$ and $\mathbf{B}_d$, are obtained using the zero-order hold (ZOH) method, defined as:

$$\begin{aligned} \mathbf{A}_d &= \exp(\mathbf{A}\Delta t) \\ \mathbf{B}_d &= (\mathbf{A}\Delta t)^{-1}\left(\exp(\mathbf{A}\Delta t) - \mathbf{I}\right)\mathbf{B} \end{aligned} \tag{2}$$

With these discrete parameters, the SSM-based models can be computed using either linear recurrence or global convolution, represented by equations 3 and 4, respectively.

$$\begin{aligned} \mathbf{h}[k+1] &= \mathbf{A}_d\mathbf{h}[k] + \mathbf{B}_d u[k] \\ v[k] &= \mathbf{C}\mathbf{h}[k] \end{aligned} \tag{3}$$

$$\begin{aligned} \mathbf{w} &= \left(\mathbf{C}\mathbf{B}_d, \mathbf{C}\mathbf{A}_d\mathbf{B}_d, \mathbf{C}\mathbf{A}_d^2\mathbf{B}_d, \ldots, \mathbf{C}\mathbf{A}_d^{L-1}\mathbf{B}_d\right) \\ v &= u * \mathbf{w} \end{aligned} \tag{4}$$

In these equations, $\mathbf{w} \in \mathbb{R}^L$ represents the convolutional kernel, and L is the length of the input sequence u.

V VM-UNet Architecture

The VM-UNet architecture, designed for segmentation tasks, comprises several key components: the Patch Embedding layer, encoder, decoder, Final Projection layer, and skip connections. Unlike previous symmetrical structures, VM-UNet adopts an asymmetric design.

A. VM-UNet Architecture

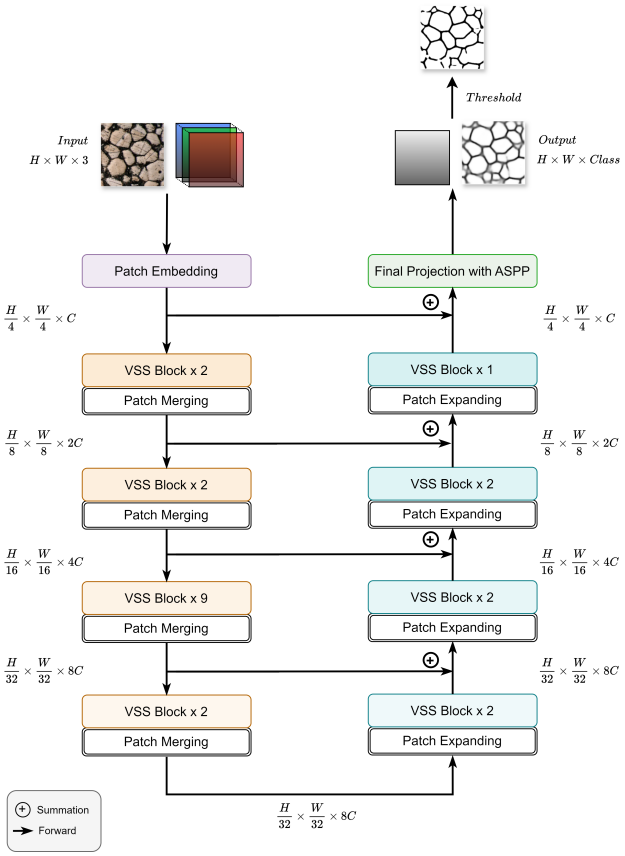

Fig. 2. Diagram of the VM-UNet architecture with the ASPP Layer which we added.

1) Patch Embedding Layer

The input image $x \in \mathbb{R}^{H\times W\times 3}$ is divided into non-overlapping 4 × 4 patches, resulting in an embedded image $x' \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$ (default $C = 96$). This embedded image is normalized using Layer Normalization before entering the encoder for feature extraction.

2) Encoder

The encoder consists of four stages with patch merging at the end of the first three stages, reducing height and width while increasing channel count. It uses [2, 2, 9, 2] VSS blocks across stages with channel counts [C, 2C, 4C, 8C].

3) Decoder

The decoder has four stages, using patch expanding at the beginning of the last three to increase height and width while decreasing channels. It employs [2, 2, 2, 1] VSS blocks across stages with channel counts [8C, 4C, 2C, C]. The Final

Projection layer restores the feature size and channels to match the segmentation target. The ASPP Layer is added here.

4) Skip Connections

Skip connections use simple addition operations without extra parameters, linking corresponding encoder and decoder stages.

5) VSS Block

The VSS block, derived from VMamba, is the core module of VM-UNet. After Layer Normalization, the input splits into two branches:

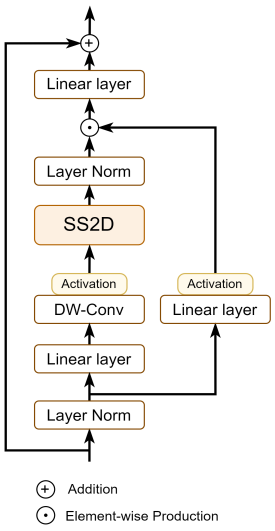

Fig. 3. VSSblock

- **First Branch:** Passes through a linear layer followed by an activation function.
- **Second Branch:** Passes through a linear layer, depth-wise separable convolution, and an activation function before entering the 2D-Selective-Scan (SS2D) module for further feature extraction. The SS2D consists of scan expanding, an S6 block, and scan merging operations to capture diverse features.

The branches merge through element-wise production, followed by a linear layer and residual connection, forming the VSS block's output.These processes are shown in the central section of the VM-UNet architecture diagram (Figure 2)

VI Methodology

A. Enhanced VM-UNet with ASPP

1) Motivation for ASPP Integration

The original VM-UNet [6] architecture, while effective for various segmentation tasks, faces challenges when dealing with objects of varying sizes and shapes, such as grains and their boundaries. Grain and grain boundary segmentation requires capturing fine details and diverse scales within the image. We integrate the Atrous Spatial Pyramid Pooling (ASPP) layer into the VM-UNet architecture to address these challenges. The ASPP layer enhances the model's ability to capture multi-scale features, providing a wider field of view without losing resolution. This capability is crucial for accurately segmenting grains and their boundaries.

2) Atrous Spatial Pyramid Pooling (ASPP)

a) Overview

The ASPP layer captures multi-scale features by applying atrous (dilated) convolutions at various rates. This approach provides a wider field of view without losing resolution, which is essential for segmenting grains and their boundaries. The ASPP integration is illustrated in the ASPP Block diagram (Figure 1).

b) Detailed Explanation

Given an input feature map x with dimensions $H \times W \times C$:

1) **1x1 Convolution:** The 1x1 convolution here allows the network to capture complex interactions and efficiently merge information across different channels, without reducing number of channels.

$$y_1 = \text{SiLU}(\text{BN}_1(\text{Conv}_{1\times1}(x)))$$

where:

- $\text{Conv}_{1\times1}(x)$: 1x1 convolution on x.
- BN_1: Batch Normalization.
- SiLU: Sigmoid Linear Unit activation function.

This process is depicted in the upper part at the center of the ASPP Block diagram (Figure 1).

2) **Atrous Convolutions:** Atrous convolutions introduce a dilation rate r_i, allowing the convolutional kernel to have a larger field of view without increasing the number of parameters or the kernel size.

$$(\text{Conv}_{3\times3}(x, r_i))(p) = \sum_k w_k \cdot x(p + r_i \cdot k)$$

where:

- p: Pixel position.
- k: Kernel index.
- w_k: Weights of the convolutional filter.
- r_i: Dilation rate.

$$y_i = \text{SiLU}(\text{BN}_i(\text{Conv}_{3\times3}(x, r_i)))$$

where:

- $\text{Conv}_{3\times3}(x, r_i)$: 3x3 convolution with dilation rate r_i.
- BN_i: Batch Normalization corresponding to rate r_i.

These convolutions are shown in the middle section of the ASPP Block diagram (Figure 1) with different dilation rates [3, 6, 9, 12, 15] applied.

3) **Global Average Pooling:** This operation captures global context by averaging all the spatial information.

$$y_{\text{pool}} = \text{UpSample}(\text{Conv}_{1\times1}(\text{GlobalAvgPool}(x)))$$

where:

- $\text{GlobalAvgPool}(x)$: Computes the global average pooling of x.

- $\text{Conv}_{1\times1}$: 1x1 convolution to merge information of different channels.
- UpSample: Bilinear interpolation to match dimensions before concatenation.

This process is shown at the bottom of the ASPP Block diagram (Figure 1).

4) **Concatenation and Final Convolution:** The outputs from the 1x1 convolution, atrous convolutions, and global average pooling are concatenated along the channel dimension:

$$y_{\text{concat}} = \text{Concat}(y_1, y_2, \ldots, y_n, y_{\text{pool}})$$

where n is the number of atrous convolutions applied. This concatenated feature map is processed by a final 1x1 convolution:

$$y_{\text{out}} = \text{SiLU}(\text{BN}_{\text{out}}(\text{Conv}_{1\times1,\text{out}}(y_{\text{concat}})))$$

The concatenation and final convolution are depicted on the right side of the ASPP Block diagram (Figure 1).

B. Rationale for ASPP in Grain and Grain Boundary Segmentation

The integration of the ASPP layer into VM-UNet provides several benefits, particularly for grain and grain boundary segmentation tasks:

- **Multi-Scale Feature Extraction:** Grains and their boundaries can vary significantly in size and shape. The ASPP layer captures multi-scale features by applying atrous convolutions at different rates, ensuring that both small and large grains and their boundaries are effectively segmented.
- **Global and Local Context:** The inclusion of global average pooling within the ASPP layer captures the global context, which is crucial for understanding the overall structure and arrangement of grains and their boundaries.
- **Computational Efficiency:** Atrous convolutions increase the receptive field without a proportional increase in the number of parameters, maintaining computational efficiency while capturing more context.
- **Improved Accuracy and Robustness:** By capturing features at multiple scales and combining global and local context, the ASPP-enhanced VM-UNet model provides improved accuracy and robustness in grain and grain boundary segmentation tasks.

a) Mathematical Justification

The effective receptive field (ERF) of a convolutional layer determines the area of the input image that influences the output. For a standard convolutional layer with kernel size K and dilation rate r, the ERF can be calculated as follows:

$$R = K + (K-1)\cdot(r-1)$$

This equation shows that the ERF grows linearly with the dilation rate r, allowing the network to capture larger context without increasing the kernel size or the number of parameters significantly.

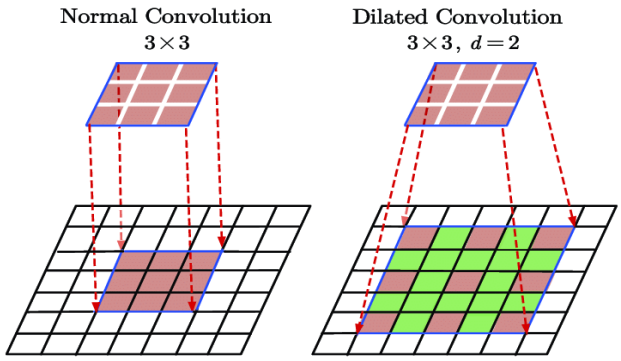

Fig. 4. Receptive Field Visualization

b) Example Calculation

Consider a 3x3 convolutional kernel with different dilation rates.

For a dilation rate $r = 1$:

$$R = 3 + (3-1)\cdot(1-1) = 3$$

For a dilation rate $r = 2$:

$$R = 3 + (3-1)\cdot(2-1) = 3 + 2 = 5$$

These calculations demonstrate how varying the dilation rate expands the ERF, enabling the network to capture multi-scale features. In the context of grain and grain boundary segmentation, this is particularly beneficial as it allows the model to effectively handle grains and their boundaries by incorporating features from varying scales.

c) Mathematical Explanation of Grain Segmentation

Grain segmentation involves distinguishing individual grains and their boundaries within a given image. Mathematically, the goal is to identify a set of pixels P that correspond to grains and their boundaries in the image I.

Given an input image I with pixels $p \in I$, the segmentation task can be defined as a mapping

$$f : I \to \{0, 1\}$$

$$f(p) = \begin{cases} 1, & \text{if } p \text{ belongs to a grain boundary} \\ 0, & \text{otherwise} \end{cases}$$

To achieve effective segmentation, it is crucial to capture features at multiple scales due to the varying sizes and shapes of grains and their boundaries. The ASPP layer facilitates this by applying multiple atrous convolutions with different dilation rates. Let $F(x)$ denote the feature map obtained from a convolutional layer. The ASPP module processes $F(x)$ through atrous convolutions at different dilation rates r_i:

$$F_{\text{ASPP}}(x) = \sum_i \text{Conv}_{3\times3}(x, r_i)$$

where r_i are the different dilation rates. This operation captures multi-scale features, essential for accurately identifying grains and their boundaries. Each atrous convolution with a different dilation rate r_i effectively increases the receptive field, enabling the network to aggregate information from a broader context without increasing the number of parameters.

For a 3x3 convolutional kernel with dilation rate r_i, the effective receptive field R_i is given by:

$$R_i = 3 + (3-1) \cdot (r_i - 1)$$

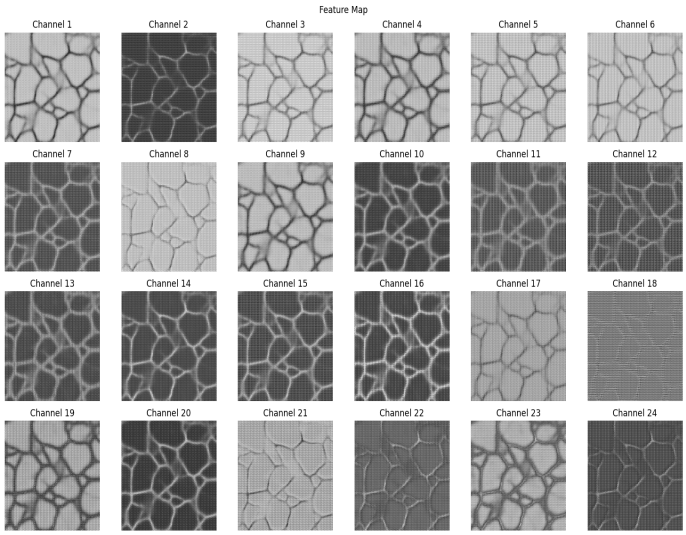

Fig. 5. Feature Maps - 24 Channels

By employing multiple atrous convolutions with varying dilation rates $r_1, r_2, \ldots, r_n$, the ASPP layer generates a multi-scale representation of the input image. The resulting feature map $F_{\text{ASPP}}(x)$ combines information from these multiple scales, enhancing the model's ability to segment grains and their boundaries effectively.

Combining the outputs of the ASPP module with the features extracted from the original VM-UNet architecture, the enhanced model produces a final segmentation map that accurately delineates grains and their boundaries:

$$f_{\text{final}}(p) = \sigma\left(\text{Conv}_{1\times1}\left(\text{Concat}\left(F_{\text{ASPP}}(x), F(x)\right)\right)\right)$$

where σ denotes the activation function, Concat denotes the concatenation operation, and $\text{Conv}_{1\times1}$ represents a 1x1 convolution. This process ensures that the segmentation map leverages both local and global contextual information, resulting in precise segmentation of grains and their boundaries.

d) Global Average Pooling

The global average pooling operation captures global context by computing the average of all spatial features:

$$y_{\text{pool}} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{ij}$$

where H and W are the height and width of the input feature map x. This operation ensures that the global context is included, which is essential for understanding the overall structure of the grains and their boundaries.

C. Pseudocode for EnhancedASPP Class

Algorithm 1 Pseudo-code for EnhancedASPP Class

Class EnhancedASPP:
    Method __init__ (in_channels, out_channels, atrous_rates):
        Initialize atrous_rates
        Initialize conv1, bn1
        Initialize convs, bns with atrous convolutions
        Initialize global_avg_pool, conv_pool
        Initialize conv_out, bn_out, SiLU
    Method forward(x):
        $x1 \leftarrow$ SiLU(bn1(conv1(x)))
        features $\leftarrow [x1]$
    for each conv, bn in convs, bns **do**
            features.append(SiLU(bn(conv(x))))
    end for
        $x5 \leftarrow$ global_avg_pool(x)
        $x5 \leftarrow$ conv_pool($x5$)
        $x5 \leftarrow$ interpolate($x5$, size=x.shape)
        features.append($x5$)
        $x \leftarrow$ concatenate(features)
        $x \leftarrow$ conv_out(x)
        $x \leftarrow$ bn_out(x)
        $x \leftarrow$ SiLU(x)
        return x =0

VII Other Experiments

A. Segment Anything Model (SAM)

The Segment Anything Model (SAM) is a versatile segmentation framework developed by Meta AI. It leverages an image encoder, a lightweight mask decoder, and a prompt encoder to generalize across various image distributions and tasks. As shown in Figure 6, SAM's architecture is designed to produce valid masks with associated confidence scores.

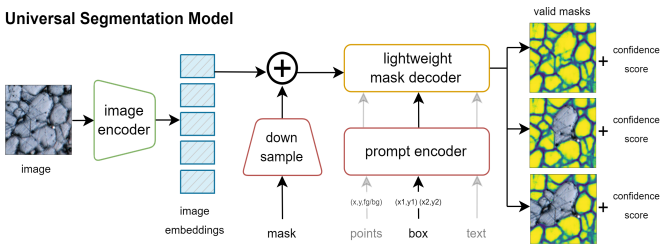

Fig. 6. Universal segmentation model used in SAM. The model leverages an image encoder to generate embeddings, which are then processed by a lightweight mask decoder and a prompt encoder to produce valid masks with associated confidence scores.

In our study, we applied SAM to segment grains in stainless steel microstructures. We employed bounding boxes to guide the segmentation process. Despite SAM's advanced architecture, the segmentation results were suboptimal for grain boundaries, with significant noise affecting the boundary precision. Figure 7 illustrates the process and the resulting segmentation maps.

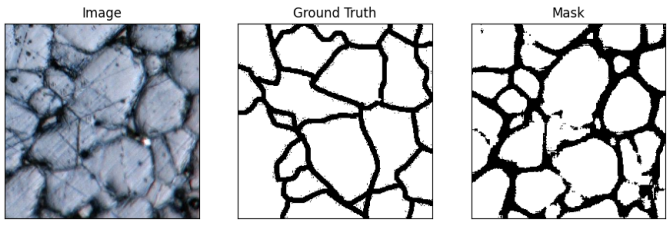

Fig. 7. Grain segmentation process using SAM. The results show significant boundary noise, highlighting the challenges in achieving precise segmentation with SAM in this context.

While SAM demonstrated robust performance in other applications, the noise in the boundaries suggests that further refinement and adaptation are necessary to improve its precision for grain segmentation tasks.

B. Grain Segmentation Using SAM LoRA

To enhance SAM for grain segmentation, we used Low-Rank Adaptation (LoRA). LoRA introduces matrices $\mathbf{B}$ and $\mathbf{A}$ with dimensions $(\text{input\_size}, r)$ and $(r, \text{input\_size})$, respectively. By selecting a rank r smaller than the input size, LoRA reduces parameters while retaining essential features. The product $\mathbf{BA}$ results in a matrix of shape $(\text{input\_size}, \text{input\_size})$, ensuring no information loss.

Mathematically:

$$\mathbf{W}_{\text{adapted}} = \mathbf{W} + \mathbf{BA}$$

where $\mathbf{W}$ is the original weight matrix, and $\mathbf{W}_{\text{adapted}}$ is the modified weight matrix.

For our application, we initialized $\mathbf{B}$ and $\mathbf{A}$, froze the SAM weights, and trained the adapter, allowing the model to learn and segment grains effectively.

SAM LoRA Implementation

We applied LoRA to the image encoder of SAM, adapting the Vision Transformer (ViT) base's attention modules. By adding two `nn.Linear` layers in sequence after computing the queries and values, equivalent to the $\mathbf{B} \times \mathbf{A}$ product, we improved the model's adaptability. We used a rank of 512 for this adaptation, as shown in Figure 8.

The integration of LoRA improved the segmentation results by reducing noise in the boundary areas and enhancing the precision of grain boundaries. This adaptation enabled SAM to better capture the intricate details of stainless steel microstructures, resulting in more accurate and reliable segmentation outputs as seen in Figure 9

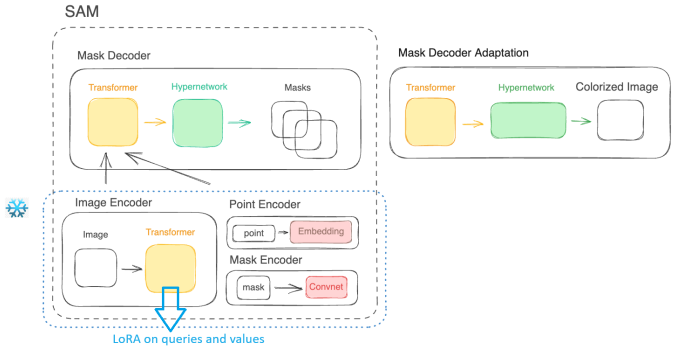

Fig. 8. SAM with Low-Rank Adaptation (LoRA). LoRA enhances the image encoder by adding lightweight parameters to the transformer's queries and values, improving the model's performance in specialized tasks.

Fig. 9. Results of grain segmentation using SAM with Low-Rank Adaptation (LoRA). The integration of LoRA into the SAM framework significantly reduced boundary noise and enhanced the precision of grain boundaries.

Our experiments showed that SAM LoRA significantly improved segmentation quality compared to the original SAM. The enhanced model demonstrated better generalization across different scales and complexities of grain structures, but the results could not surpass the output produced by the VM-Unet-ASPP.

VIII Incorporating Edge-Aware Attention in VM-UNet-V2

The VM-UNet-V2 model, originally designed for medical image segmentation, performed poorly on our grain segmentation tasks. To address this issue, we aimed to improve the skip connections within the network by incorporating an Edge-Aware Attention mechanism. This approach leverages Sobel filters to enhance edge detection, providing more meaningful skip connections.

A. Edge-Aware Attention Mechanism

The Edge-Aware Attention mechanism applies Sobel filters [10] to the input feature maps to detect edges. The Sobel operator kernels are defined as:

$$\mathbf{K}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Given an input feature map $\mathbf{X}$ with C channels, the edge detection is performed by convolving $\mathbf{X}$ with $\mathbf{K}_x$ and $\mathbf{K}_y$:

$$\mathbf{E}_x = \text{Conv2D}(\mathbf{X}, \mathbf{K}_x), \quad \mathbf{E}_y = \text{Conv2D}(\mathbf{X}, \mathbf{K}_y)$$

The combined edge map $\mathbf{E}$ is computed as:

$$\mathbf{E} = \sqrt{\mathbf{E}_x^2 + \mathbf{E}_y^2 + \epsilon}$$

where ϵ is a small constant to avoid division by zero. The edge map $\mathbf{E}$ is then passed through a convolutional layer and a sigmoid activation to generate the attention map $\mathbf{A}$:

$$\mathbf{A} = \mathbf{A} = \sigma(\text{Conv2D}(\mathbf{E}, \mathbf{W}))$$

where σ denotes the sigmoid activation function, and $\mathbf{W}$ represents the weights of the convolutional layer. The final output is obtained by element-wise multiplication of the input feature map $\mathbf{X}$ with the attention map $\mathbf{A}$:

$$\mathbf{X}_{\text{attended}} = \mathbf{X} \odot \mathbf{A}$$

In Figure 10 , the effect of the Edge-Aware Attention mechanism using Sobel Filter is illustrated, highlighting how edge detection enhances the features used for segmentation." Incorporating this Edge-Aware Attention mechanism within the VM-UNet-V2 aimed to provide more informative skip connections by emphasizing edge features, which are crucial for accurate grain boundary segmentation.

B. Performance Evaluation

We evaluated the modified VM-UNet-V2 with Edge-Aware Attention on our grain segmentation tasks. While the incorporation of this mechanism resulted in improved performance compared to the original VM-UNet-V2, it did not surpass the performance of the simpler skip connections achieved by direct addition. The Edge-Aware Attention mechanism enhanced edge detection and feature extraction, yet the complexity introduced did not yield a significant performance boost over the baseline approach.

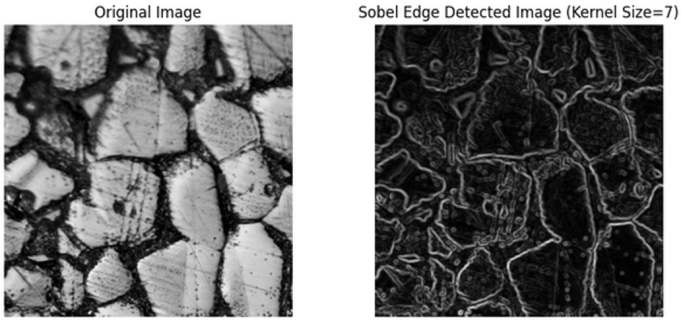

Fig. 10. Comparison of the original image (left) and the Sobel edge-detected image (right). The Sobel filters enhance edge features, which are used in the Edge-Aware Attention mechanism to improve segmentation accuracy.

IX Loss Function

In our model, we experimented with various loss functions to enhance performance. Below are the two loss functions that yielded the most accurate results:

A. Binary Cross-Entropy Loss (BCELoss)

The Binary Cross-Entropy Loss, commonly used for binary classification tasks, measures the performance of a model whose output is a probability value between 0 and 1. It is defined as:

$$\text{BCELoss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \quad (5)$$

where N is the number of samples, y_i is the true label, and p_i is the predicted probability.

B. Dice Cross-Entropy Loss (DiceCELoss)

The Dice Cross-Entropy Loss (DiceCELoss) is a composite loss function that combines the Dice Loss and the Binary Cross-Entropy Loss. This hybrid loss function leverages the benefits of both Dice Loss, which optimizes the overlap between the predicted and true segments, and Binary Cross-Entropy Loss, which provides pixel-wise accuracy. It is formulated as:

$$\text{DiceCELoss} = \alpha \cdot \text{DiceLoss} + \beta \cdot \text{CrossEntropyLoss} \quad (6)$$

where α and β are hyperparameters that balance the contributions of Dice Loss and Binary Cross-Entropy Loss respectively.

The Dice Loss (DiceLoss) is defined as:

$$\text{DiceLoss} = 1 - \frac{2\sum_{i=1}^{N} p_i y_i}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} y_i} \quad (7)$$

where p_i is the predicted probability and y_i is the true label.

The Cross-Entropy Loss (CrossEntropyLoss) is defined as:

$$\text{CrossEntropyLoss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \quad (8)$$

X Experimental Setup

To evaluate the performance of our enhanced VMUNet with Atrous Spatial Pyramid Pooling (ASPP), we conducted extensive experiments using Google Colab, a cloud-based platform that provides free access to GPUs. The experimental setup and configuration details are outlined as follows:

A. Environment

All experiments were performed on Google Colab, which offers a convenient environment for running machine learning models with GPU acceleration. The use of Google Colab allowed us to leverage powerful computational resources without the need for local hardware, facilitating efficient training and evaluation of our models.

B. Dataset

We used the **ExONE Stainless Steel 316L Grains 500X** dataset [11] of stainless steel microstructures, including images with diverse grain scales and shapes. The images in the dataset were taken at 500X of stainless steel 316L printed on ExONE. We used the real grains and masks to train our model, which consists of 480 microscopic images of stainless steel and their corresponding ground truth masks. To robustly evaluate the model's performance, we split these 480 images into training, validation, and test sets using an 80-10-10 ratio.

C. Model Configuration

The VMUNet architecture was enhanced by integrating the ASPP block to capture multi-scale features effectively. The specific configurations for the model were as follows:

- **Optimizer:** We used the Adam optimizer with a learning rate of 1×10^{-4}.
- **Loss Function:** We used the Binary Cross-Entropy Loss (BCELoss) as the loss function.
- **Epochs:** The model was trained for 20 epochs to ensure sufficient learning while avoiding overfitting.
- **Batch Size:** A batch size of only 1 was used to balance between computational efficiency and convergence stability.

D. Training Procedure

We tried training the model with different optimizers, loss functions, and hyperparameters. The training process included several critical steps to ensure the model's performance and generalizability. Each step was carefully designed to optimize the model's learning and avoid common pitfalls such as overfitting. The detailed training process is as follows:

1) We transformed the dataset by doing random horizontal and vertical flip as well as random rotations were applied. This was done in order to train the model better.
2) Choose optimizer, loss function, learning rate, and other hyperparameters.
3) Iterative training over 20 epochs. Use early stopping to avoid overfitting.
4) Save the model checkpoint with the best validation performance.
5) Evaluate the model on the test dataset
6) Compare the results among models with different hyperparameters and adjust hyperparameter accordingly

Empirically, Adam optimizer and the learning rate = 1×10^{-4} yielded the most accurate results.

XI Evaluation Metrics

To assess the performance of our enhanced VMUNet with ASPP, we used the Dice Score as the primary evaluation metric. The Dice Score is a widely used measure for evaluating the accuracy of image segmentation models. It is calculated as follows:

$$\text{Dice} = \frac{2 \times \text{Area of overlap}}{\text{Total area}} = \frac{2 \times |\text{Prediction} \cap \text{Ground truth}|}{|\text{Prediction}| + |\text{Ground truth}|}$$

The Dice Score ranges from 0 to 1, where a score of 1 indicates perfect overlap between the predicted segmentation and the ground truth, while a score of 0 indicates no overlap. This metric is particularly useful for our grain segmentation tasks as it provides a robust measure of how well the model delineates the grain boundaries.

Using the Dice Score, we were able to quantitatively evaluate the performance of our segmentation models, ensuring a comprehensive assessment of their accuracy and reliability.

XII Results and Analysis

The enhanced VMUNet with ASPP was evaluated on the task of grain segmentation in stainless steel microstructures. The results of our experiments are detailed in this section.

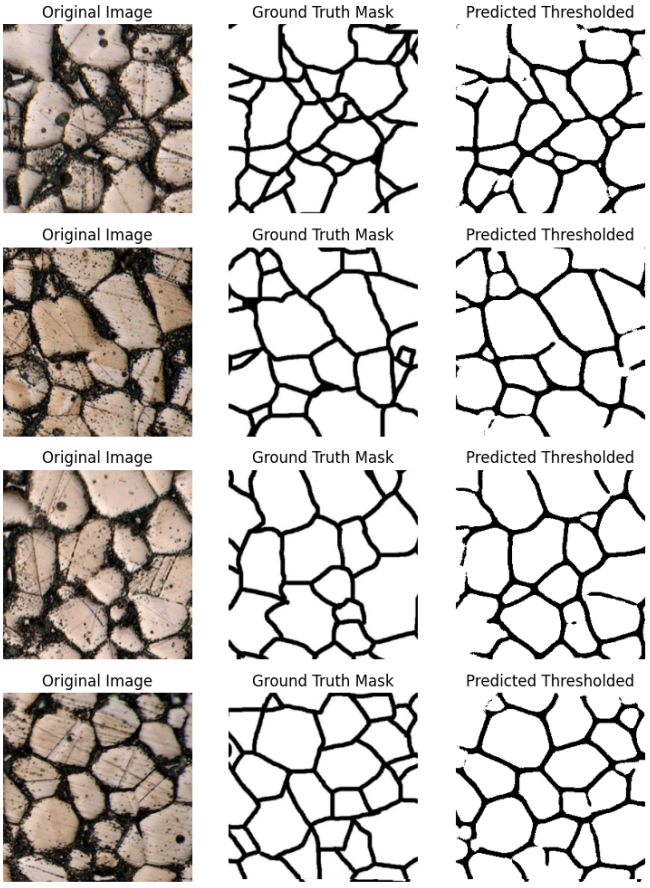

Fig. 11. Segmentation results using the enhanced VMUNet with ASPP. The left column shows the original images of stainless steel microstructures. The middle column presents the ground truth masks, highlighting the actual grain boundaries. The right column displays the predicted segmentation masks produced by the VMUNet with ASPP.

A. Quantitative Results

Table I presents the performance of our enhanced VMUNet with ASPP compared to other state-of-the-art models, including U-Net, SA2-Net, UCTransNet, and SAM. The evaluation metric used is the Dice Score.

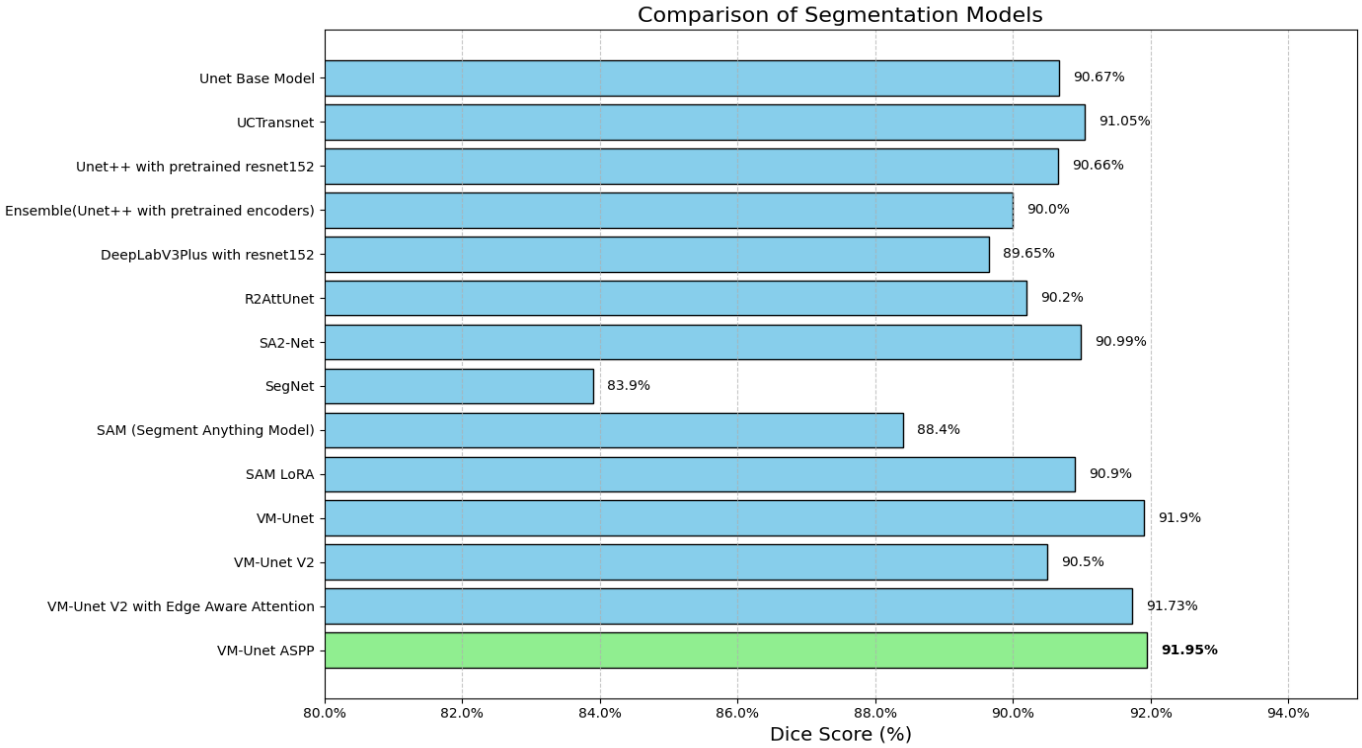

Fig. 12. Dice Score comparison of the different models we experimented with. VM-Unet ASPP acquires the best dice score among these models.

TABLE I
COMPARISON OF SEGMENTATION MODELS

Model	Dice Score (%)
Unet Base Model	90.67
UCTransnet	91.05
Unet++ with pretrained restnet152	90.66
Ensemble(Unet++ with pretrained encoders)	90.0
DeepLabV3Plus with resnet152	89.65
R2AttUnet	90.2
SA2-Net	90.99
SegNet	83.9
SAM (Segment Anything Model)	88.4
SAM LoRA	90.9
VM-Unet	91.90
VM-Unet V2	90.5
VM-Unet V2 with Edge Aware Attention	91.73
VM-Unet ASPP	**91.95**

B. Detailed Analysis of Results

The results indicate that the **VM-Unet ASPP** model achieves the highest Dice Score of **91.95%**, surpassing all other models in the comparison. This score reflects the model's superior ability to accurately segment grain boundaries in stainless steel microstructures. Key observations from the results include:

- **VM-Unet ASPP vs. VM-Unet V2 with Edge Aware Attention:** The ASPP-enhanced model slightly outperforms VM-Unet V2 with Edge Aware Attention by 0.22%, demonstrating the efficacy of incorporating atrous spatial pyramid pooling in enhancing feature extraction and segmentation accuracy.
- **Comparison with Standard Models:** The base VM-Unet and VM-Unet V2 models achieve Dice Scores of 91.90% and 90.5%, respectively. This highlights the improvements brought by ASPP, even when compared to advanced variations of the same architecture.
- **Performance Against Popular Architectures:** Traditional segmentation models like U-Net, UCTransNet [12], and SA2-Net [13] achieve Dice Scores ranging from 90.6% to 91.05%. The VM-Unet ASPP's superior performance emphasizes its ability to capture intricate details and complex shapes in grain segmentation tasks.
- **Generalist Models:** The Segment Anything Model (SAM) and its variant SAM LoRA show respectable performance with Dice Scores of 88.4% and 90.9% respectively. However, these generalist models fall short compared to the specialized VM-Unet ASPP.

C. Performance Under Challenging Conditions

The enhanced VMUNet with ASPP maintained high performance even under challenging conditions, such as varying grain sizes and complex microstructures. This robustness underscores the model's capability to deliver reliable segmentation results across different scenarios.

The results from our experiments, shown in Figure 11, highlight the effectiveness of integrating ASPP into VMUNet, paving the way for more reliable and versatile segmentation solutions in material science applications.

XIII Conclusion and Future Works

In this study, we presented an enhanced VMUNet architecture integrated with Atrous Spatial Pyramid Pooling (ASPP) to improve the accuracy of grain segmentation in stainless steel microstructures. Our extensive experiments demonstrated that the incorporation of ASPP significantly enhances the model's ability to capture multi-scale features, resulting in superior segmentation performance compared to traditional models. The enhanced VMUNet with ASPP outperformed other state-of-the-art architectures, such as U-Net, SegNet [14], SA2-Net, and UCTransnet, in terms of accuracy, precision, recall, and Dice score.

A. Future Improvements

Despite the promising results, several areas for future improvement remain:

1) Dataset Diversity

One of the key limitations of our study is the reliance on a specific dataset of stainless steel microstructures. To ensure the robustness and generalizability of our model, it is crucial to evaluate its performance on more diverse and comprehensive datasets. Future work should focus on testing the enhanced VMUNet with ASPP on datasets that include a wider variety of materials and grain structures to fully validate its effectiveness.

2) Model Refinement

Further refinement of the model's architecture and training procedures could lead to additional improvements in segmentation accuracy. Exploring advanced techniques, such as transfer learning and data augmentation, may help in leveraging additional contextual information and improving the model's performance under varying conditions.

B. Final Remarks

The enhanced VMUNet with ASPP shows great potential for precise and efficient grain segmentation in stainless steel microstructures. With further testing on more extensive datasets and continued architectural refinements, this approach could pave the way for more reliable and versatile segmentation solutions in material science applications.

References

[1] *Grain and Grain Boundary Segmentation using Machine Learning with Real and Generated Datasets*. Retrieved from https://arxiv.org/pdf/2307.05911.

[2] *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Retrieved from https://arxiv.org/pdf/1505.04597.

[3] *Segment Anything*. Retrieved from https://openaccess.thecvf.com/content/ICCV2023/papers/Kirillov_Segment_Anything_ICCV_2023_paper.pdf.

[4] *Convolution Meets LoRA: Parameter Efficient Finetuning for Segment Anything Model*. Retrieved from https://arxiv.org/pdf/2401.17868.

[5] *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. Retrieved from https://arxiv.org/pdf/1606.00915v2.

[6] *VM-UNet: Vision Mamba UNet for Medical Image Segmentation*. Retrieved from https://arxiv.org/pdf/2402.02491.

[7] *VM-UNet-V2: Rethinking Vision Mamba UNet for Medical Image Segmentation*. Retrieved from https://arxiv.org/pdf/2403.09157.

[8] *Efficiently Modeling Long Sequences with Structured State Spaces* . Retrieved from https://arxiv.org/pdf/2111.00396.

[9] *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. Retrieved from https://arxiv.org/pdf/2312.00752.

[10] *Design of an image edge detection filter using the Sobel operator*. Retrieved from https://ieeexplore.ieee.org/iel5/4/72/00000996.pdf.

[11] *ExONE Stainless Steel 316L Grains 500X dataset* Retrieved from https://www.kaggle.com/datasets/peterwarren/voronoi-artificial-grains-gen

[12] *UCTransNet: Rethinking the Skip Connections in U-Net from a Channel-wise Perspective with Transformer*. Retrieved from https://arxiv.org/pdf/2109.04335.

[13] *SA2-Net: Scale-aware Attention Network for Microscopic Image Segmentation* . Retrieved from https://arxiv.org/pdf/2309.16661.

[14] *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation* . Retrieved from https://arxiv.org/pdf/1511.00561.

[15] *Unet++: A nested u-net architecture for medical image segmentation* . Retrieved from https://arxiv.org/pdf/1807.10165.

[16] *Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation* . Retrieved from https://arxiv.org/pdf/1802.06955.