# CS-UNet: A Generalizable and Flexible Segmentation Algorithm

Khaled Alrfou[1*], Tian Zhao[1] and Amir Kordijaz[2]

[1*]Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee, 3200 North Cramer Street, Milwaukee, 53211, WI, USA.
[2]Department of Engineering, University of Southern Maine, 135 John Mitchell Center,Gorham Campus, Gorham, 04038, ME, USA.

*Corresponding author(s). E-mail(s): kalrfou@uwm.edu;
Contributing authors: tzhao@uwm.edu; Amir.kordijazi@maine.edu;

**Abstract**

This study introduces a novel U-shaped image-segmentation algorithm, CS-UNet, which contains parallel CNN and Transformer encoders. This algorithm leverages the relative strength of CNN and Transformers, and enables flexible combination of encoders pre-trained on different datasets to extract the maximum benefit of transfer-learning. CS-UNet is evaluated for its segmentation accuracy on microscopy images of materials science. The performance of CS-UNet is comparable or better than state-of-the-art algorithms based on CNN or Transformer encoders. As expected, the performance of CS-UNet is better when its encoders are pre-trained on microscopy images than when its encoders are pre-trained on natural images. However, the strength of in-domain pre-training is more significant in use cases such as out-of-distribution learning and one-shot learning. In particular, the Intersection over Union (IoU) accuracy of nickel-based super-alloy images is improved from 77.89% to 82.13% for out-of-distribution learning and IoU accuracy of environmental-barrier-coating images is improved from 65.9% to 70.45% for one-shot learning.
CS-UNet also has surprisingly good performance on medical images. For Synapse multi-organ dataset, CS-UNet pre-trained on materials microscopy images has an average accuracy of 84.2% in Dice Similarity Coefficient (DSC), and 8.89 mm in 95% Hausdorff Distance (HD). In comparison, state-of-the-art segmentation algorithms pre-trained on ImageNet have an average DSC ranging from 76.5% to 80.39% and average HD ranging from 14.7 to 39.7 mm. Even when pre-trained on ImageNet, CS-UNet still has DSC of 83.27% and HD of 15.26 mm. This suggests that Transformer and CNN complement each other and pre-training on images with similar attributes is beneficial to the downstream tasks.

**Keywords:** Microscopy, Swin Transformer, CNN and CS-UNet Segmentations.

# 1 Introduction

Deep Learning (DL) has been widely applied to complex systems because of its ability to extract important information automatically. Researchers have applied DL algorithms to image analysis to identify structures and to determine the relationship between microstructure and performance [1]. DL has been demonstrated to complement physics-based methods for materials design. However, DL requires large amount of training data while the limited number of microscopy images tends to reduce its effectiveness. Learning techniques, such as transfer-learning, multi-fidelity modeling, and active learning, were developed to make DL applicable to smaller datasets [1, 2]. Transfer-learning uses the parameters of a model pre-trained on a larger dataset to initialize a model trained on a smaller dataset for a downstream task. For example, a Convolutional Neural Network (CNN) pre-trained on natural images can be used to initialize a neural network for image segmentation such as UNet to improve its precision and reduce the training time.

In recent years, attention-based neural networks called Transformers are widely adopted in computer vision. While CNN extracts features from local regions of images using convolution filters to capture the spatial relation between the pixels, Transformer divides an image into patches and feeds them into a Transformer-based encoder to capture the long-range relation between pixels across the images [3, 4]. Thus, it is possible that a combination of CNN and Transformer may be more effective in transfer-learning than either of the models alone.

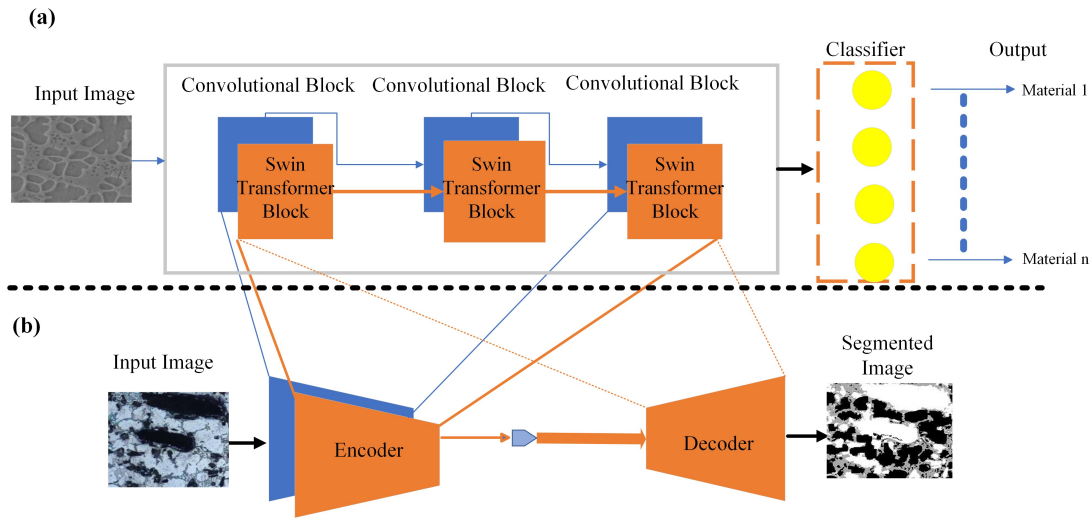


**Fig. 1** The high-level architecture of CS-UNet, where the weights of the CNN and Swin-T encoders (below the line) can be initialized from CNN and Swin-T blocks (above the line) pre-trained on natural and/or microscopy images through classification tasks.

In this paper, we present a novel segmentation algorithm called CS-UNet that includes a parallel composition of CNN and Transformer encoders. The high-level architecture of CS-UNet is illustrated in Figure 1, which shows a U-shaped encoder-decoder architecture. The parameters of the encoders are initialized from models pre-trained on natural or microscopy images. Each encoder transforms the input image into a latent representation vector to extract semantic information. Each decoder maps the extracted information back to each pixel in the input image to generate a pixel-wise classification of the image [1, 5]. The output of the

CNN and Transformer encoders are fused before connecting to the decoder. CS-UNet allows great flexibility in combining different types of CNN and Transformer encoders pre-trained on different types of data to allow optimal choices of encoders for the segmentation tasks.

Encoder-decoder architecture allows pre-training to improve segmentation accuracy. It was speculated that pre-training with microscopy images is better for microscopy image segmentation since natural images has high-level features that do not exist in microscopy images. Recent work by Stuckner *et al.* [6] confirmed the benefit of pre-training CNN encoders on a microscopy dataset called MicroNet with over 100,000 images. They evaluated the CNN encoders with the segmentation of nickel-based super alloy (Super) and environmental barrier coating (EBC) images. Their tests showed higher accuracy measured in Intersection over Union (IoU) for one-shot and few-shot learning and for out-of-distribution images that have different compositions, etching, and imaging conditions than the training images.

To evaluate the performance of CS-UNet, we pre-trained CNN and Transformer encoders on different types of datasets and performed segmentation on the same test sets used by Stuckner *et al.* [6]. We chose the tiny version of Swin-Transformer – Swin-T [7] as our Transformer encoder. While we can initialize the CNN encoders using the CNN models of Stuckner *et al.* [6], we are unable to obtain their dataset MicroNet to train our Swin-T encoder. To this end, we created a similar pre-training dataset with about 50,000 microscopy images in 74 classes, which we will refer to as MicroLite.

Our experiments showed that CS-UNet has similar or better accuracy than the state-of-the-art algorithms based on CNN or Transformer encoders including the ones included in Stuckner *et al.* [6]. We also compared the performance of CS-UNet using encoders pre-trained on microscopy images with the encoders pre-trained on natural images. The results showed improvement in segmentation accuracy for one-shot learning and out-of-distribution learning, which is largely in agreement with the result of Stuckner *et al.*. Due to visual similarity between microscopy and the computed tomography (CT) images, we speculate that CS-UNet pre-trained on microscopy image can improve segmentation accuracy of CT images. To this end, we compared the performance of CS-UNet with 6 state-of-the-art algorithms on the Synapse multi-organ segmentation dataset. Our results showed that CS-UNet pre-trained on microscopy images outperforms the previous algorithms in both Dice Similarity Coefficient (DSC) and 95% Hausdorff Distance (HD) by significant margins. This offers further evidence that the parallel combination of CNN and Transformer encoders is a good choice for U-shaped segmentation networks and pre-training on visually similar images help improving downstream tasks.

In the rest of the paper, we first survey Transformer-based algorithms on image analysis and segmentation in Section 2. We then describe the CS-UNet architecture, the pre-training dataset MicroLite, and the pre-training process of the Transformer encoders of CS-UNet in Section 3. We discuss the evaluation result of CS-UNet on test datasets of microscopy images in Section 4 and of medical images in Section 5.

# 2 Related Work

In this section, we review the recent Transformer networks for image analysis and the Transformer-based segmentation algorithms. We focus on the algorithms that are most relevant to this study.

## 2.1 Transformers for Image Analysis

CNN uses the convolution operators to provide translational equivariance but its local receptive field has limitation in capturing long-range relation between pixels [4]. In recent years, Transformer has been used in place of CNN for computer vision (CV) tasks to overcome this limitation. Transformer is a type of deep neural network introduced by Vaswani *et al.* [8], which was successfully applied in Natural Language Processing

(NLP) due to their ability to capture long-range dependencies in sequential data such as text. This approach led to significant improvements in NLP applications, such as language translation, text classification, and text generation. Compared to NLP tasks, using Transformer-based models on CV tasks is more challenging since images have size variations, noises, and redundant modalities. Self-attention process is the fundamental building block of Transformer aiming to learn self-alignment that provides the ability to capture the long-range relation between image patches [4]. This has led to much interest in the Transformer-based approach in CV domains [3] such as image recognition, image segmentation [9], object detection [10, 11], image super-resolution, and image generation [12]. Dosovitskiy *et al.* [3] proposed a Vision Transformer (ViT) based on a vanilla Transformer network for NLP [8] with as few modifications as possible to capture the global context of an input image. ViT splits each image into patches and provides the Transformer with the linear embedding of each patch in order. Image patches are handled in the same manner as tokens in an NLP application. Supervised learning is used to train the model for image classification. The model is fine-tuned using downstream recognition benchmarks such as ImageNet classification after pre-training on a JFT dataset with 300 million images [13]. ViT has better performance than traditional CNN and achieved 88.5% on ImageNet classification task. However, ViT required more computational resources to train. In addition, the complexity of computing SoftMax for each self-attention block is quadratic with respect to the length of input sequence, limiting its applicability to high-resolution images [3, 4].

To improve a Transformer model to capture local information, Liu *et al.* [7] proposed a new vision Transformer called Shifted Window Transformer (Swin Transformer). This method proposed a new general-purpose backbone for image classification and recognition tasks and achieved state-of-the-art performance. The model used a shifted-window scheme to capture large variations in the scale of visual entities and high resolutions pixels in an image with linear computation complexity to input image size. In contrast, ViT [3] model produces feature maps of a single low resolution and have quadratic computation complexity to input image size because self-attention is applied globally to all the patches. Swin Transformer achieves a good performance of 87.3% on the ImageNet classification task, 58.7% box average precision score on COCO detection task, and 53.5% mIoU on the ADE20K dataset for segmentation task. Swin Transformer V2 [14] can scale up to 3 billion parameters and train with images as high quality as $1536 \times 1536$ pixels. Swin Transformer V2 modified the Swin attention module for better window resolution and scale model capacity. This is done by replacing the pre-norm with post-norm configuration, using scaled cosine attention instead of dot product attention, and replacing the previous parameterized approach with a log-spaced continuous relative position bias approach.

The concept of unboxing the decision-making strategy in image segmentation aims to move beyond just achieving results but to understand how those results are obtained. This transparency is vital in medical applications where trust and interpretability are paramount. Deep Nearest Centroid (DNC) [15] is an approach that utilizes the Sinkhorn-Knopp algorithm [16] to expedite the clustering of features into sub-centroids corresponding to different categories. This approach replaced the traditional softmax classification layer, leading to improved optimization of network parameters and enhancements in tasks such as image classification and semantic segmentation. DNC draws inspiration from intuitive case-based reasoning. It summarizes each class into sub-centroids by clustering training data. Classifications are made based on the proximity of test data to these sub-centroids, allowing for the generation of IF-THEN rules and visualization of representative images, making the decision process clear and understandable. ClusterFormer [17] and CLUSTSEG [18] leveraged clustering in the Transformer architecture. ClusterFormer utilizes recurrent cross-attention clustering and feature dispatching to handle various tasks with varying granularity. CLUSTSEG employs task-aware initialization and recursive clustering to tackle diverse segmentation tasks like super-pixel, semantic, instance, and panoptic segmentation, all within a single framework. Both models achieve

state-of-the-art results while offering transparency through their clustering-based approaches. CS-UNet may be adapted to incorporate these clustering-based methods such as DNC [15] to introduce interpretability to the model and to improve performance.

## 2.2 Image Segmentation Algorithms

In recent years, many variations of U-shaped networks have been used in image segmentation. U-Net is a Fully Convolutional Network (FCN) proposed by Ronneberger et al. [19, 20], which is a symmetric, U-shaped, encoder-decoder neural network for semantic image segmentation. U-Net typically consists of a down-sampling encoder and an up-sampling decoder structure and a "skip connection" between them. These connections copy feature maps from the encoder and concatenate them with the feature maps in the decoder.

Transformer encoder was used in SegFormer [21], which is a semantic segmentation framework that combines Transformer encoders with lightweight MultiLayer Perceptron (MLP) decoders. SegFormer is also based on encoder-decoder architecture, where the encoder is a hierarchically structured Transformer that outputs multi-scale features without the need for positional encoding and the lightweight All-MLP decoder aggregates the information from different layers and combines local and global attention to produce the final semantic segmentation mask. SegFormer uses a patch size of $4 \times 4$ pixels to output a segmentation map. This approach helps improve dense prediction tasks and has resulted in impressive mIoU scores of 50.3% on the ADE20K dataset and 84% on the Cityscapes dataset.

CNN and Transformer were combined in TransUNet [22], which is a U-shaped architecture that employs a hybrid CNN-Transformer encoder followed by multiple up-sampling layers in the CNN decoder. This method leverages both detailed high-resolution spatial information from CNN features and the global context encoded by Transformers. The TransUNet architecture includes 12 ViT [3] layers in the encoder, which encodes tokenized image patches obtained from the CNN layers. These encoded features are then up-sampled in the decoder to generate the final segmentation map, with skip-connections incorporated. TransUNet achieved high performance compared with the CNN-based models.

Swin-Unet [23] uses only Transformer encoders in its U-shaped encoder-decoder architecture for medical image segmentation. Swin-Unet includes skip-connections for local-global semantic-feature learning by feeding the tokenized image patches into the model. Both the encoder and decoder structures of Swin-Unet were inspired by the hierarchical Swin-Transformer [7] with shifted windows.

UNEt TRansformer (UNETR) model [24] is a U-shaped encoder-decoder architecture for 3D medical image segmentation, which uses ViT [3] as encoders to capture global multi-scale information. The Transformer encoder is connected with the CNN decoder using skip connections to compute the final semantic segmentation output. UNETR has excellent accuracy and efficiency in various medical datasets for image segmentation tasks. Swin UNETR [25] is a similar model for segmenting brain tumors in multi-modal MRI images, which uses Swin Transformer [7] as encoders and connects to CNN decoders via skip connections at different resolutions. This model had one of the top performance in BraTS 2021 segmentation challenge for multi-modal 3D brain tumor segmentation.

HiFormer [26] bridges CNN and Transformer encoders for medical image segmentation, where it uses two multi-scale feature representations and a Double-Level Fusion (DLF) module to fuse global and local features. Experiments showed that HiFormer outperforms other CNN-based, Transformer-based, and hybrid methods in computational complexity and accuracy. HiFormer provides an end-to-end training strategy that integrates global contextual representations from Swin Transformer and local representative features from the CNN module in the encoder, followed by a decoder that outputs the final segmentation map.

TransDeepLab [27] is a DeepLabV3+ architecture based on pure Transformer for medical image segmentation. It uses a hierarchical Swin-Transformer with shifted windows to model the Atrous Spatial

Pyramid Pooling (ASPP) module. The encoder module splits the input image into patches and applies Swin-Transformer blocks to encode local semantic and long-range contextual representation. A pyramid of Swin-Transformer blocks with varying window sizes is designed for ASPP to capture multi-scale information. The extracted multi-scale features are then fused into the decoder module using a Cross-Contextual attention mechanism. Finally, in the decoding path, the extracted multi-scale features are up-sampled and concatenated with the low-level features from the encoder to refine the feature representation.

In summary, the hybrid architectures mentioned above either replace CNN with a Transformer in the encoder (e.g. Swin UNETR [25]) or stack a CNN with a Transformer sequentially to form a new encoder ( e.g. TransUNet [22]). Replacing CNN with a Transformer in the encoder gives the ability to model long distance dependency in the network. However, it results in a lack of detailed texture feature extraction due to the removal of CNN in the encoder. Stacking CNN with a Transformer to form a new encoder fails to account for the complementary relationship between the global modeling capability of self-attention and the local modeling capability of convolution. Instead, they treat the convolution operation and self-attention as two separate and unrelated operations [28, 29].

To overcome these drawbacks, the encoder in CS-Unet uses CNN and Transformer in parallel to obtain rich feature information from training images. CNN is used to extract low-level features and Swin-T is used to extract global contextual features, which are then fused using skip connections to the decoder at different stages/layers. Moreover, to reduce feature loss in the transmission process and to increase the contextual information extracted by the Swin-T encoder, the Multi-Layer Perceptron (MLP) in two successive Swin-T blocks was replaced by Residual Multi-Layer Perceptron (ResMLP).

# 3 Methodology

In this section, we give details on how CS-UNet is implemented, how the dataset MicroLite is created, and how the encoders of CS-UNet are pre-trained.

## 3.1 CS-UNet Architecture

CNN does not capture long-range spatial relation due to its intrinsic locality. Transformer is able to overcome this limitation but it is limited in capturing low-level features. Since both local and global information are essential for dense prediction tasks such as segmentation in challenging contexts, hybrid models with both CNN and Transformer encoders are expected to provide better performance for image segmentation. CS-UNet, as shown in Figure 2, is such a hybrid model that consists of CNN encoders, Transformer encoders, bottlenecks, Transformer decoders, and skip connections. The CNN encoders extract low-level features and the Swin-T encoders extract global contextual features. Each Swin-T encoder operates on the input image divided into non-overlapping patches, applying self-attention mechanisms to capture global dependencies. The Swin-T encoders capture long-range dependencies and contextual information from the entire image at different scales.

Inspired by the TFCN (Transformers for Fully Convolutional dense Net) [31] and Lightweight Swin-Unet [30], we replaced the Multi-Layer Perceptrons (MLP) in two successive Swin-T blocks with Residual Multi-Layer Perceptrons (ResMLP) illustrated in Figure 3. ResMLP is used to reduce feature loss in the transmission process and to increase the contextual information extracted by the encoder. ResMLP is illustrated in Figure 4, which consists of two GELU [32] nonlinear layers, three Linear layers, and two dropout layers. The CNN encoder processes the input image in a series of convolution layers, gradually reducing the spatial dimensions while extracting hierarchical features. Along the way, the encoder captures low-level features in early layers and higher-level semantic features in deeper layers.

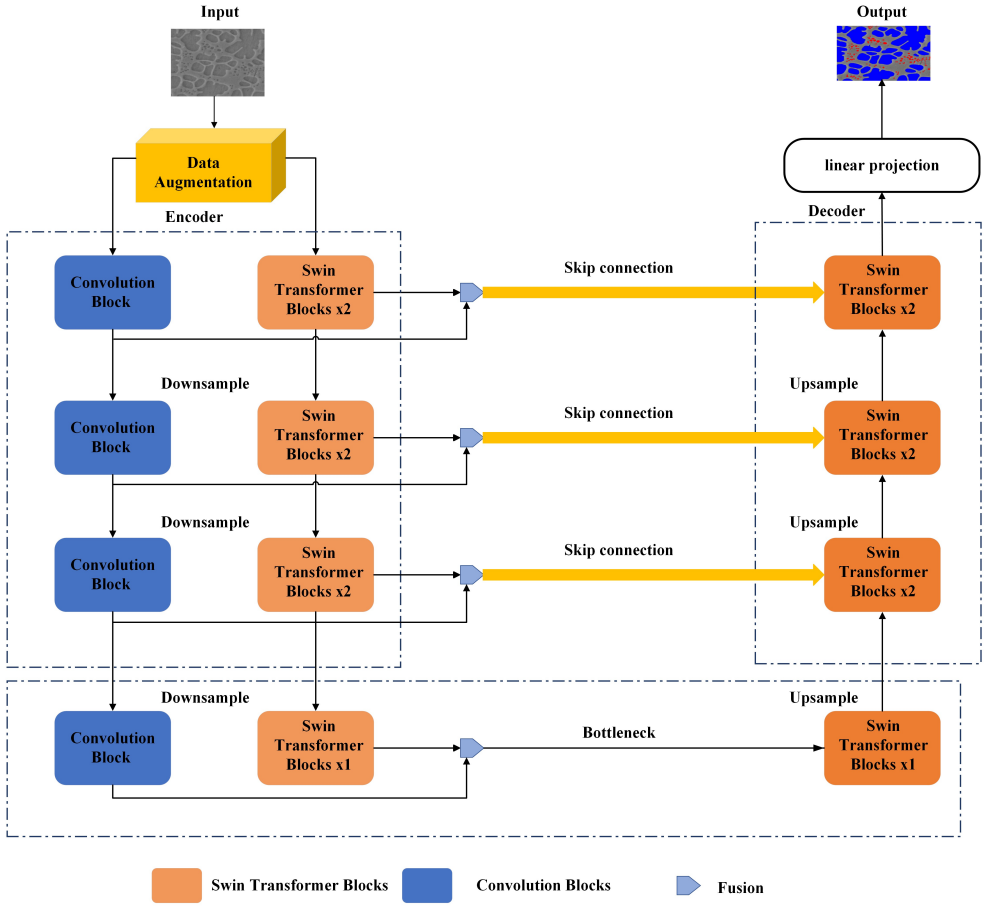


**Fig. 2** CS-UNet architecture includes CNN and Swin-T encoders, bottlenecks, skip connections, and Swin-T decoder.

To fuse the information from both encoders, the skip connections concatenate the feature maps from the CNN encoder and the Swin-T encoder with the corresponding decoder layers. To ensure compatibility between the feature dimensions of the CNN and the Swin-T encoders, we normalize the dimensions before fusing them. This is achieved by passing the features obtained from the CNN block through a linear embedding layer, which flattens and reshapes the feature map from the shape of $(B, C, H, W)$ to $(B, C, H \times W)$, where $B$, $C$, $H$, $W$ are the batch size, the number of channels, the height, and the width of the feature map, respectively. The flattened feature map is transposed to swap the last two dimensions resulting in the shape of $(B, H \times W, C)$ and is then fused with extracted features from the Swin-T encoder. By fusing the information from different encoder pathways, the skip connections enable the decoder to benefit from both the local spatial details captured by the CNN encoder and the global context captured by the Swin-T encoder.

The decoder is similar to that of Swin-Unet [23], which employs the patch-expanding layer to up-sample the extracted deep features by reshaping the feature maps of adjacent dimensions to form a higher-resolution feature map, which effectively achieves a $2\times$ up-sampling. Additionally, it reduces the feature dimension to half of the original dimension. This allows the decoder to reconstruct the output with increased spatial

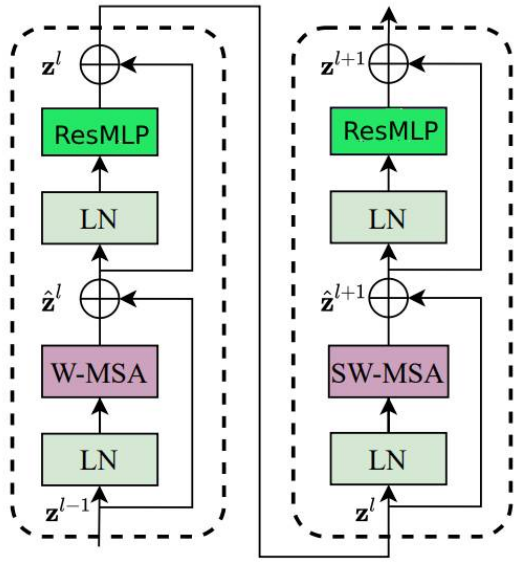


**Fig. 3** ResMLP is used to improve Swin Transformer block [30].

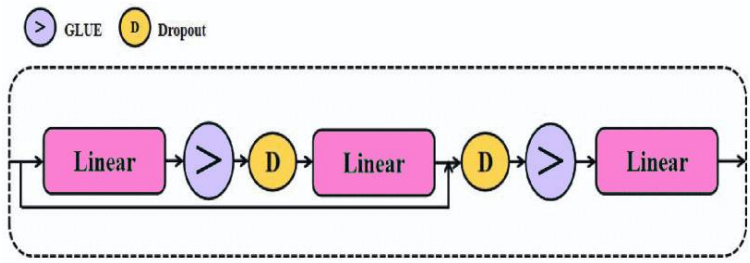


**Fig. 4** ResMLP module block [30].

resolution while reducing the feature dimension for efficient processing. The final patch-expanding layer further performs a 4× up-sampling to restore the resolution of the feature maps to match the input resolution ($W \times H$). Finally, a linear projection layer is applied to these up-sampled features to generate pixel-level segmentation predictions.

Different CNN families can be used in the encoder part such as EfficientNet, ResNet, MobileNet, DenseNet, VGG, and Inception. We initialize CNN weights using MicroNet and the transformer weights using MicroLite.

## 3.2 Pre-training Dataset

The MicroLite images were collected from multiple sources including images from different materials and compounds using several measurement techniques such as light microscopy, SEM, TEM, and X-ray. MicroLite aggregates the Aversa dataset [33], UltraHigh Carbon Steel Micrograph [34], SEM images from the Materials Data Repository, and the images from some recent publications [35–42]. The Aversa dataset includes over 25,000 SEM microscopy images in 10 classes, where each class consists of images in different scales (including

1, 2, 10, 20 um and 100, 200 nm) and contrast. To properly classify these images, we used a pre-trained VGG-16 model to extract feature maps from these images and used a K-means algorithm to cluster the feature maps so that images with similar feature maps are grouped in the same class. After the pre-processing step, we obtained 53 classes. The authors of Aversa dataset manually classified a small set of the images (1038) into a hierarchical dataset, where the 10 classes are further divided into 27 subclasses [33]. Our classification of these 1038 images is largely consistent with the manually assigned subclasses. Note that we have more classes since we processed the entire Aversa dataset.

In total, MicroLite includes about 50,000 microscopy images labelled in 74 classes, which are obtained using the following pre-processing steps.

1. Remove any artifacts such as scale bars from the images.
2. Split the images into $512 \times 512$-pixel tiles with or without overlapping depending on the size of the original images.
3. Apply data augmentation to increase the size of the dataset.
4. Aggregate the original image, the images tiles, and the augmented images to form the final dataset.

Since our current approach employs VGG-16 and K-means for feature map clustering in the pre-processing step, future work could benefit from integrating advanced clustering-based transformer methods, such as ClusterFormer [17], which can lead to more efficient and accurate results.

### 3.3 Pre-training Swin-T Encoders

We trained Swin Transformers on microscopy images to learn feature representation so that it can be transferred to tasks such as segmentation. We evaluated two types of training.

1. Fine-tune a model pre-trained on ImageNet with MicroLite (denoted by ImageNet $\rightarrow$ MicroLite).
2. Pre-train a model with MicroLite from scratch (denoted by MicroLite).

The classification tasks uses Swin-T, which is the tiny version of the Swin Transformer. Swin-T consists of two types of architectures: the original Swin-T with [2,2,6,2] transformer blocks and the intermediate network with [2,2,2,2] transformer blocks. Figure 5 shows the original architecture of Swin-T. We speculate that intermediate network may be enough for microscopy analysis tasks since the earlier layers learn corner edges and shapes, the intermediate layers learn the texture or patterns, and deeper network layers in the original models learn the high-level features such as eyespots and caudal appendages. The original and intermediate Swin-T models were pre-trained on MicroLite from scratch, where the model weights are randomly initialized. The two models were also pre-trained on ImageNet and fine-tuned on MicroLite.

The pre-training step uses an AdamW optimizer for 30 epochs with a cosine-decay learning-rate scheduler with 5 epochs of linear warm-up and batch size of 128. The initial learning rate is $10^{-3}$ and weight decay is 0.05. The fine-tuning step also uses an AdamW optimizer for 30 epochs with a batch size of 128 but the learning rate is reduced to $10^{-5}$ and the weight decay is reduced to $10^{-8}$. Models were trained until there was no improvement to the validation score using an early stopping criterion with a patience of 5 epochs. Training data had been augmented using albumentations library, which included random changes to the contrast and the brightness, vertical and horizonal flips, photometric distortions, and added noise.

Swin-T models were trained by classifying microscopy images into 74 different classes. Swin-T models were either pre-trained on ImageNet (specifically the imageNet1K dataset) and fine-tuned on MicroLite, or trained with MicroLite with randomized parameters. The training stops when the validation accuracy does not improve after 5 epochs. The model accuracy is evaluated using the top-1 and top-5 accuracy. The top-1

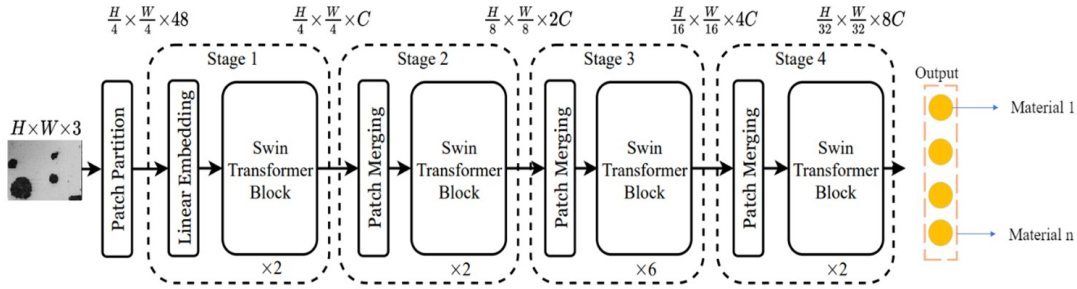


**Fig. 5** The original Swin-T architecture consists of four stages. Each stage contains two Swin transformer blocks except the stage three that contains six Swin transformer blocks [7]

**Table 1** Classification accuracy of pre-trained models, where the method "ImageNet → MicroLite" indicates that the models were pre-trained on ImageNet and then fine-tuned on MicroLite.

| Swin-T arch. | Pre-training method | top-1 accuracy | top-5 accuracy | # of epochs |
|---|---|---|---|---|
| Original | MicroLite | 84.23 | 95.91 | 23 |
| | ImageNet → MicroLite | **84.63** | 96.353 | 13 |
| Intermediate | MicroLite | 84.0 | 96.91 | 19 |
| | ImageNet → MicroLite | 84.45 | **97.83** | **12** |

accuracy measures the percentage of test samples for which the correct label is predicted while the top-5 accuracy measures the percentage of correct labeling in the top five predictions.

As shown in Table 1, the Swin-T models, when trained from scratch, take longer to converge. Specifically, the original Swin-T model takes 23 epochs, while the intermediate version takes 19 epochs. In contrast, the Swin-T models pre-trained on the ImageNet and then fine-tuned on the MicroLite converge faster. The original Swin-T model takes only 13 epochs, while the intermediate version takes 12 epochs. On average, the models initialized with ImageNet weights converged about 40.16% faster than those with the random initialization. Original Swin-T fine-tuned on MicroLite after pre-training with ImageNet achieved the top-1 accuracy of 84.63%. Overall, the Swin-T models pre-trained on ImageNet and fine-tuned on MicroLite have higher accuracy and faster convergence.

For the down-stream segmentation tasks, several models were trained for each task, which include Swin-Unet, HiFormer, and TrasDeeplapv3+. A comparative analysis was performed on these models that were pre-trained with ImageNet and microscopy images.

# 4 Evaluation of CS-UNet on Microscopy Images

In this section, we evaluate CS-UNet by comparing its performance with the results of Stucker *et al.* [6] on their 7 microscopy datasets derived from two materials: nickel-based super-alloys (Super) and Environmental Barrier Coatings (EBC). Super datasets have 3 classes: matrix, secondary, and tertiary. EBC datasets have two classes: the oxide layer and the background (non-oxide) layer. The number of images in each dataset split is shown in Table 2. Super-1 and EBC-1 contain the full dataset labeled for their respective materials. Super-2 and EBC-2 have only 4 images in the training set to evaluate the performance of few-shot learning. Super-3 and EBC-3 have only 1 image in the training set to evaluate performance of one-shot learning.

Super-4 have test images taken under different imaging and sample conditions to evaluate the performance of out-of-distribution learning. All segmentation models in this study were trained using PyTorch [43].

**Table 2** Number of Microscopy images in the training, validation, and test set for each experimental dataset

| Experiment | # of training images | # of validation images | # of test images |
|---|---|---|---|
| Super-1 | 10 | 4 | 4 |
| Super-2 | 4 | 4 | 4 |
| Super-3 | 1 | 4 | 4 |
| Super-4 | 4 | 4 | 5 |
| EBC-1 | 18 | 3 | 3 |
| EBC-2 | 4 | 3 | 3 |
| EBC-3 | 1 | 3 | 3 |

EBC and Super datasets were augmented in ways similar to [6], which includes random cropping to $512 \times 512$ pixels, random changes to contrast, brightness, and gamma, and added blurring or sharpening. EBC dataset was horizontally flipped and Super dataset was randomly flipped vertically and horizontally and rotated. The training parameters such as the optimizer and the learning rate were adopted from prior research [6]. The training step used the Adam optimizer with an initial learning rate of $2 \times 10^{-4}$ until the validation accuracy showed no improvement for 30 epochs. Afterwards, the training continued with a learning rate of $10^{-5}$ until early stopping was triggered after an additional 30 epochs without any validation improvement. Since the datasets are imbalanced, the loss function was set by the weighted sum of Balanced Cross Entropy (BCE) and dice loss with a 70% weighting towards BCE [6].

CS-UNet has a flexible architecture so that it can be trained with different encoders initialized with different pre-trained parameters. Table 3 shows the various combinations of pre-trained weights that were used to initialize the CS-UNet encoders. The second column shows the pre-trained weights that initialize the Swin-T encoder and the third column shows the pre-trained weights that initialize the CNN encoder. In the last column, we use the term *microscopy* to refer to the fact that the CNN encoders are initialized with weights pre-trained on Stuckner *et al.* [6]'s MicroNet and Swin-T encoders are initialized with weights pre-trained on our MicroLite dataset. Other combinations of pre-trained weights can also be used to train the CS-UNet model. For example, the Swin-T encoder could be initialized with the MicroLite weights and the CNN encoder could be initialized with the ImageNet→MicroNet weights. The flexibility of the CS-UNet architecture allows researchers to experiment with different combinations of pre-trained weights to find the best combination for their specific task.

**Table 3** The combinations of encoders in CS-UNet, where the second column shows the pre-training models for the Swin-T encoders and the third column shows the pre-training models for the CNN encoders. The last column shows the models for both types of encoders, where 'Microscopy' is either MicroNet or MicroLite.

| Swin-T arch. | Swin-T pre-train model | CNN pre-train model | CS-UNet pre-train model |
|---|---|---|---|
| Original | ImageNet | ImageNet | ImageNet |
| Original | MicroLite | MicroNet | microscopy |
| Original | ImageNet → MicroLite | ImageNet → MicroNet | ImageNet → Microscopy |
| Intermediate | MicroLite | MicroNet | microscopy |
| Intermediate | ImageNet → MicroLite | ImageNet → MicroNet | ImageNet → Microscopy |

**Table 4** The top performance (IoU) of UNet++/UNet pre-trained on MicroNet [6], Transformer-based algorithms pre-trained on MicroLite, and CS-UNet pre-trained on MicroNet and MicroLite.

| Dataset | UNet++/UNet + MicroNet | Transformer + MicroLite | CS-UNet + MicroNet + MicroLite |
|---|---|---|---|
| Super-1 | 96.4% | 95.72% | **96.43%** |
| Super-2 | 94.2% | 95.16% | **96.06%** |
| Super-3 | 93.0% | 92.23% | **93.5%** |
| Super-4 | 78.5% | 78.91% | **82.13%** |
| EBC-1 | 97.6% | 96.59% | **97.66%** |
| EBC-2 | **93.3%** | 91.11% | 92.82% |
| EBC-3 | 65.9% | **82.13%** | 70.46% |

## 4.1 Comparison between CS-UNet and CNN/Transformer-based Algorithms

Table 4 compares the top performance of UNet++/UNet pre-trained on MicroNet (from Figure 3–5 of *et al.* [6]), Transformer-based segmentation algorithms (including Swin-Unet, TransDeepLabV3+, and HiFormer) pre-trained on MicroLite, and CS-UNet pre-trained on MicroNet and MicroLite. The highest accuracy for each experiment is shown in bold font. CS-UNet has the best performance in most experiments except EBC-2 and EBC-3. For experiments with ample training data such as Super-1 and EBC-1, the difference between UNet++/UNet, Transformer, and CS-UNet is small. For few-shot learning experiments such as Super-2 and EBC-2, the accuracy gain of CS-UNet is modest. For one-shot learning experiments, the result is mixed, where CS-UNet has modest improvement in Super-3 while significant gain in EBC-3. For out of context learning, CS-UNet shows significant improvement over UNet or Transformer.

Note that while CS-UNet generally outperforms CNN or Transformer-based methods, CS-UNet sometimes is worse than one of them. For example, for EBC-3 (one-shot learning), Transformer-based method is significantly better in IoU accuracy (82.13% vs 70.46%). However, CS-UNet and UNet are slightly better than Transformer for Super-3, which is also one-shot learning. For one-shot and few-shot learning cases, CNN and Transformer have uneven performances. Since CS-UNet combines both types of encoders, it tends to average out the performance.

Overall, this result suggests that CS-UNet has more consistent performance than prior algorithms. CS-UNet is similar or significantly better than UNet++/UNet in all experiments and it is better than Transformers for most experiments. Note that MicroLite is about half the size of MicroNet. Despite this, Transformer + MicroLite has comparable or better performance than that of UNet++/UNet + MicroNet.

## 4.2 Performance Impact of Pre-training Data and Encoder Configurations

**Table 5** The configuration of top-performing Transformer-based algorithms when pre-trained on ImageNet and MicroLite.

| **pre-train** | **ImageNet** | | | **MicroLite** | | |
|---|---|---|---|---|---|---|
| *test data* | *Segmentation Algo.* | *Swin-T Arch.* | *IoU* | *Segmentation Algo.* | *Swin-T Arch.* | *IoU* |
| Super-1 | TransDeepLabV3+ | Orginal | 95.25% | Swin-Unet | Intermediate | **95.72%** |
| Super-2 | Swin-Unet | Orginal | 94.37% | TransDeepLabV3+ | Intermediate | **95.16%** |
| Super-3 | Swin-Unet | Orginal | 89.78% | TransDeepLabV3+ | Intermediate | **92.23%** |
| Super-4 | Swin-Unet | Orginal | 76.42% | Swin-Unet | Orginal | **78.91%** |
| EBC-1 | Swin-Unet | Orginal | 96.11% | TransDeepLabV3+ | Intermediate | **96.59%** |
| EBC-2 | HiFormer | Orginal | 84.21% | TransDeepLabV3+ | Orginal | **91.11%** |
| EBC-3 | TransDeepLabV3+ | Orginal | 65.77% | Swin-Unet | Orginal | **82.13%** |

The middle column of Table 4 gives the best performance of Transformer-based algorithms including Swin-Unet, TransDeepLabV3+, and HiFormer, where their encoders have either the original or the intermediate swin-T architecture and they are pre-trained with either ImageNet or ImageLite dataset. Table 5 lists the configurations of the Transformer-based algorithms with the best performance. The results indicate that pre-training on MicroLite consistently provides better performance than pre-training on ImageNet. Also, intermediate architecture of Swin-T may be sufficient if pre-trained on microscopy images, who leads to less training time. While the accuracy gain of pre-training on MicroLite is modest with full training data (Super-1/EBC-1), for few-shot, one-shot, and out-of-distribution learning (Super-2/3/4, EBC-2/3), the performance improvement is more significant.

**Table 6** The best performing encoder configurations of CS-UNet when pre-trained on ImageNet and microscopy images.

| **pre-train** | **ImageNet** | | | **MicroNet + MicroLite** | | |
|---|---|---|---|---|---|---|
| *test data* | *CNN Encoder* | *Swin-T Arch.* | *IoU* | *CNN Encoder* | *Swin-T Arch.* | *IoU* |
| Super-1 | ResNext50_32x4d | Original | 96.22% | InceptionV4 | Intermediate | **96.43%** |
| Super-2 | VGG16-bn | Original | 96.03% | VGG16-bn | Intermediate | **96.06%** |
| Super-3 | EfficientNet-b3 | Original | 87.01% | EfficientNet-b4 | Intermediate | **93.5%** |
| Super-4 | EfficientNet-b2 | Original | 78.89% | EfficientNet-b3 | Original | **82.13%** |
| EBC-1 | SE_ResNeXt50_32x4d | Original | **98.0%** | SE_Renet152 | Original | 97.66% |
| EBC-2 | SE_ResNet-101 | Original | 92.0% | SE_ResNet-101 | Intermediate | **92.82%** |
| EBC-3 | SE_ResNet-50 | Original | 61.71% | SE_ResNeXt-101_32x4d | Original | **70.46%** |

The right column of Table 4 gives the best performance of CS-UNet when pre-trained on ImageNet and microscopy images. In Table 6, we include the CNN and Transformer encoder combinations that have the best performance for each test when pre-trained with ImageNet and microscopy images. Similar to the Transformer-based algorithms, pre-training with microscopy images provides the best performance in most cases except EBC-1, which is the full training set. There is significant performance improvement for one-shot and out-of-distribution learning (Super-3/4 and EBC-3). This is consistent with the findings of Stuckner *et al.* [6]. However, the performance gain of few-shot learning is not as noticeable (Super-2 and EBC-2) since the performance of CS-UNet is less sensitive to pre-training data.

In Appendix D, we included some examples of segmentation results for the Super and EBC datasets to provide a visual comparison of CS-UNet when pre-trained on microscopy over natural images.

# 5 Evaluation of CS-UNet on Medical Images

While the initial application of CS-UNet is materials science images, we believe that CS-UNet can also improve the segmentation performance of medical images. Furthermore, CS-UNet pre-trained on MicroLite should have better performance for medical images such as computed tomography (CT) images since MicroLite also contains X-Ray images. To this end, we evaluated the performance of CS-UNet pre-trained on microscopy images and ImageNet on a medical image dataset called the Synapse multi-organ segmentation dataset (Synapse) [44].

Synapse dataset includes 30 patient cases with 3779 axial abdominal clinical CT images, where 18 cases are used for training and 12 cases are used for testing. The dataset contains 8 abdominal organs (aorta, gallbladder, left kidney, right kidney, liver, pancreas, spleen, and stomach). Each CT volume includes $85 \sim 198$ slices of $512 \times 512$ pixel images, with a voxel spatial resolution of $[0.54 \sim 0.54] \times [0.98 \sim 0.98] \times [2.5 \sim 5.0]$ $mm^3$.

Each Synapse image is reduced to $224 \times 224$ pixels. It should be noted that we follow [23] to choose the optimizer and learning rate for Synapse dataset. The training batch size and learning rate are 24 and 0.05, respectively. Our model is trained with the SGD optimizer with a momentum of 0.9 and weight decay of 0.0001. The average Dice-Similarity coefficient (DSC) and average 95% Hausdorff Distance (HD) are used as evaluation metrics. HD metric provides a more precise estimate of performance with respect to boundary errors. DSC values range from 0 to 1 with the larger values indicating better performance while the smaller values of HD indicate better performance.

**Table 7** Comparison of CS-UNet and State-Of-The-Art algorithms on Synapse (the columns are average DSC in %, average HD in mm, and DSC in % for each organ). Blue indicates the best result and red displays the second-best.

| Algorithm | DSC↑ | HD↓ | Aorta | Gallbladder | Kid(L) | Kid(R) | Liver | Pancreas | Spleen | Stomach |
|---|---|---|---|---|---|---|---|---|---|---|
| U-Net [19] | 76.85 | 39.70 | **89.07** | 69.72 | 77.77 | 68.60 | 93.43 | 53.98 | 86.67 | 75.58 |
| Att-UNet [45] | 77.77 | 36.02 | **89.55** | 68.88 | 77.98 | 68.60 | 93.43 | 53.98 | 86.67 | 75.58 |
| TransUNet [22] | 77.48 | 31.69 | 87.23 | 63.13 | 81.87 | 77.02 | 94.08 | 55.86 | 85.08 | 75.62 |
| Swin-Unet [23] | 79.13 | 21.55 | 85.47 | 66.53 | 83.2 | 79.61 | 94.29 | 56.58 | 90.66 | 76.60 |
| TransDeepLab [27] | 80.16 | 21.25 | 86.04 | 69.16 | 84.08 | 79.88 | 93.53 | 61.19 | 89.00 | 78.40 |
| HiFormer [26] | 80.39 | **14.70** | 86.21 | 65.69 | 85.23 | 79.77 | 94.61 | 59.52 | **90.99** | 81.08 |
| CS-UNet[1] | **84.20** | **9.89** | 88.42 | **72.86** | **86.84** | **85.59** | **95.44** | **70.51** | **90.11** | **83.81** |
| CS-UNet[2] | **83.27** | 15.26 | 88.07 | **71.32** | **88.0** | **84.38** | **94.80** | **65.64** | 89.95 | **83.49** |

[1]CS-UNet (VGG16_bn and intermediate Swin-T) is pre-trained on microscopy images.

[2]CS-UNet (VGG16_bn and original Swin-T) is pre-trained on ImageNet.

We compared the performance of CS-UNet with 6 State-Of-The-Art (SOTA) algorithms including 2 CNN-based, 2 Transformer-based, and 2 hybrid algorithms, which are U-Net, Att-UNet, TransUNet, Swin-Unet, Transdeeplab, and HiFormer. Note that all these models are initialized with the weights pre-trained on ImageNet. Table 7 shows the performance of CS-UNet and SOTA methods in terms of the average DSC and average 95% HD on 8 abdominal organs. The segmentation of 2 sample images of Synapse is shown in Figure 6, which illustrates the better accuracy of CS-UNet when pre-trained on microscopy than on natural images.

Our results highlight the remarkable performance of CS-UNet that, when pre-trained on microscopy images, has the highest segmentation accuracy at 84.20% in average DSC and the lowest average HD at 9.89 mm. Even when pre-trained on ImageNet, CS-UNet has the second best accuracy at 83.27% in average DSC though its average HD at 15.26 mm is slightly higher than the 14.70 mm of the second best algorithm HiFormer. Lower HD indicates superior edge prediction capabilities. For individual organs, with the exception of Aorta, CS-UNet has the best and/or the second best segmentation accuracy in DSC. Even for Aorta, the accuracy of CS-UNet is also very close to the best performer with about 1% difference in accuracy. Figure 6 illustrates the visual differences between the segmentation ground truth of synapse images with the segmentation masks made by CS-UNet, when it is pre-trained on microscopy images and on ImageNet. The masks made by CS-UNet pre-trained ImageNet tends to have more over-segmentation problems.

Compared to the SOTA methods, CS-UNet has larger model size than some of the models though the inference time is on the lower end. As shown in Table 8, our method has 44.96 million parameters, which is more than Hiformer, Swin-Unet, and Transdeeplab methods but fewer than TransUnet. Additionally,

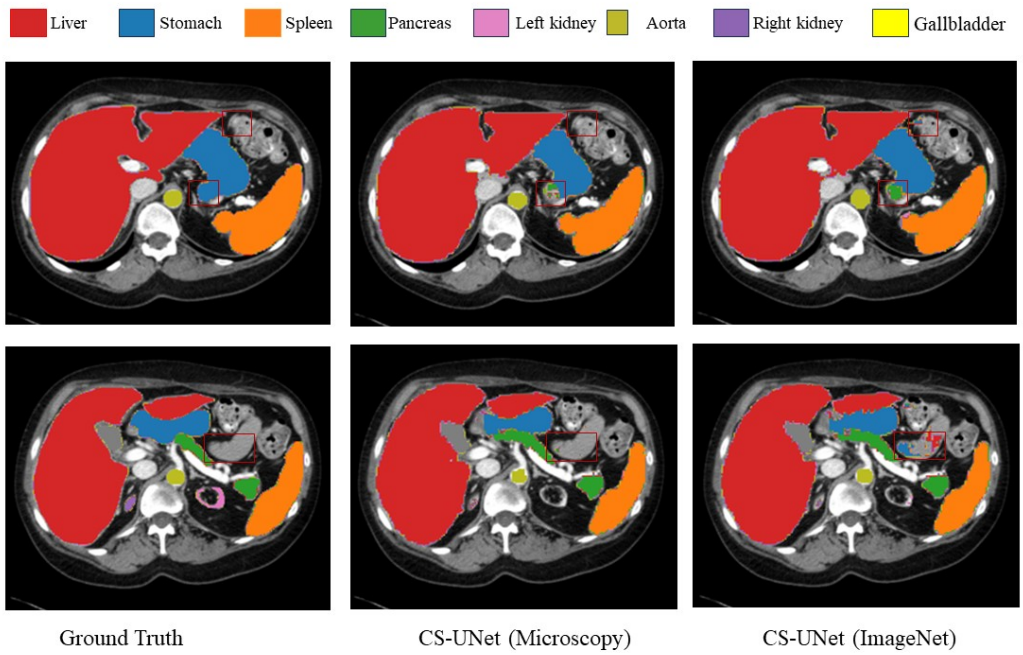


**Fig. 6** This figure compares the ground truth of 2 Synapse images (left column) with the segmentation masks of CS-UNet pre-trained on microscopy images (middle column) and CS-UNet pre-trained on ImageNet (right column). The red rectangles identify the regions where CS-UNet pre-trained on ImageNet tends to have over-segmentation problems compared to CS-UNet pre-trained on microscopy images.

CS-UNet shows a computation time score of 30:23 minutes during the testing phase. Despite having a relatively higher number of parameters, CS-UNet exhibits less computational time compared to HiFormer, Transdeeplab, and TransUnet. Importantly, it consistently outperforms them in terms of overall performance.

# 6 Ablation study

CS-UNet combines CNN and Transformer encoders with skip connections to capture local and global features. To explore the influence of different factors on the model performance, we conducted ablation studies on the Synapse dataset to evaluate how the number of skip connections and how the CNN and Transformer encoder branches influence the performance. We use the weights pre-trained on ImageNet to initialize both the encoders and the decoder of CS-UNet.

## 6.1 Effect of the encoders in CS-UNet

As shown in Table 9, we separate the CNN and Transformer encoder branches. After removing the CNN branch, the mean DSC drops by about 2.5% and HD increases by about 4.48 mm, while DCS drops by about 7.6% and HD increases by about 11.74 mm if we remove the Transformer branch. It shows that the combination of CNN and Swin Transformer in CS-UNet can fully utilize the advantages of both types of models and achieve better segmentation performance than either of them alone.

**Table 8** Comparison of CS-UNet and State-Of-The-Art algorithms based on model parameters, number of epochs to train, inference time for 1568 axial abdominal clinical CT images, and model size.

| Algorithm | # of params (M) | # of training epochs | inference time (min) | model size (MB) |
|---|---|---|---|---|
| TransUnet [22] | 105.28 | 150 | 31:09 | 414.412 |
| Swin-Unet [23] | 27.17 | 150 | 30:25 | 108.058 |
| HiFormer [26] | 25.51 | 400 | 30:56 | 101.161 |
| TransDeepLab [27] | 21.14 | 200 | 30:59 | 86.343 |
| CS-UNet (CNN: VGG16_bn)[1] | 44.96 | 150 | 30:32 | 177.613 |

[1]CS-UNet is a flexible method that can use different CNN encoders. VGG16.bn is chosen here since it provided the best performance for Synapse. Appendix C shows the number of parameters of CS-UNet with different CNN encoders.

**Table 9** Ablation study of CS-UNet encoders for Synapse dataset, separate the CNN branch and Transformer branch

| **Enoder** | **DSC↑** | **HD↓** |
|---|---|---|
| CS-UNet (Orginal) | 83.27 | 15.26 |
| CS-UNet without CNN branch | 80.68 | 19.74 |
| CS-UNet without Transformer branch | 75.61 | 27.00 |

## 6.2 Effect of the number of skip connections

The skip connections in CS-UNet connect the encoders and the decoders at 3 different stages of scale resolutions. We investigated the impact of varying the number of skip connections (0, 1, 2, and 3) on the segmentation performance of CS-UNet. Table 10 demonstrates that the higher number of skip connections enhances the segmentation performance of CS-UNet. Consequently, for better performance, we set the number of skip connections to 3 in this study.

**Table 10** Ablation study on the impact of the number of skip connections.

| Skip Connection | DSC↑ | HD↓ | Aorta | Gallbladder | Kid(L) | Kid(R) | Liver | Pancreas | Spleen | Stomach |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 75.81 | 23.72 | 79.801 | 62.24 | 81.57 | 77.66 | 93.70 | 50.83 | 85.00 | 75.64 |
| 1 | 78.84 | 18.31 | 84.07 | 62.69 | 84.93 | 80.24 | 94.02 | 58.46 | 86.91 | 79.37 |
| 2 | 82.38 | 15.72 | 88.06 | 70.17 | 86.89 | 83.86 | 95.01 | 64.52 | 88.34 | 81.64 |
| 3 | 83.27 | 15.26 | 88.07 | 71.32 | 88.0 | 84.38 | 94.800 | 65.64 | 89.95 | 83.49 |

# 7 Conclusion

This paper introduced a novel U-shaped segmentation algorithm CS-UNet, which combines CNN and Transformer encoders in parallel. The encoders of CS-UNet extract low-level and high-level features from input

images, which enables effective segmentation of images with long-range dependencies and high spatial resolution. Our tests on materials images showed that CS-UNet has better or comparable performance than prior state-of-the-art methods. Our tests also showed that the encoders pre-trained on microscopy images leads to better feature representation and thus better segmentation performance than the encoders pre-trained on natural images. We further evaluated CS-UNet on a set of medical CT images, which demonstrated clear advantages over the prior state-of-the-art methods. Pre-training on microscopy images also improved the segmentation performance on the medical images due to the inclusion of X-ray images in the pre-training dataset.

While CS-UNet can preserve spatial information and process long-range dependencies, it is computationally intensive compared to CNN and Transformer-based algorithms. As future work, we will explore the training of other Transformer architectures, such as Focal Transformer and FocalNet, on large microscopy datasets. These architectures may offer further advancements in image segmentation, expanding the range of available options and potential improvements in materials analysis tasks. Future investigations may also offer deeper insights into the decision-making processes of image segmentation. As highlighted in Section 2, the techniques such as DNC [15], ClusterFormer [17], and CLUSTSEG [18] hold promise for achieving greater transparency and interpretability in these models.

## Data Availbility

The pre-trained Swin-T models of our experiments are available at our GitHub repository:
https://github.com/Kalrfou/SwinT-pretrained-microscopy-models
This work also used the pre-trained CNN models from
https://github.com/nasa/pretrained-microscopy-models

## Author Contributions

Alrfou developed the CS-UNet method, conceived and designed the study, developed the software, evaluated results, provided datasets, and contributed to the formal analysis and writing of the original draft. Zhao and Kordijazi evaluated the results, contributed to the formal analysis and writing of the original draft, and proofread and reviewed the final manuscript.

## Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## Conflict of Interest

The authors declare that they have no conflict of interest.

## References

[1] Ge, M., Su, F., Zhao, Z., Su, D.: Deep learning analysis on microscopic imaging in materials science. Materials Today Nano **11**, 100087 (2020) https://doi.org/10.1016/j.mtnano.2020.10008

[2] Choudhary, K., DeCost, B., Chen, C., Jain, A., Tavazza, F., Cohn, R., Park, C.W., Choudhary, A., Agrawal, A., Billinge, S.J., *et al.*: Recent advances and applications of deep learning methods in materials science. npj Computational Materials **8**(1), 59 (2022) https://doi.org/10.1038/s41524-022-00734-6

[3] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., *et al.*: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) https://doi.org/10.48550/arXiv.2010.11929

[4] Alrfou, K., Kordijazi, A., Zhao, T.: Computer vision methods for the microstructural analysis of materials: The state-of-the-art and future perspectives. arXiv preprint arXiv:2208.04149 (2022) https://doi.org/10.48550/arXiv.2208.04149

[5] Jacquemet, G.: Deep learning to analyse microscopy images. The Biochemist **43**(5), 60–64 (2021) https://doi.org/10.1042/bio_2021_167

[6] Stuckner, J., Harder, B., Smith, T.M.: Microstructure segmentation with deep learning encoders pre-trained on a large microscopy dataset. npj Computational Materials **8**(1), 200 (2022) https://doi.org/10.1038/s41524-022-00878-5

[7] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021). https://doi.org/10.1109/ICCV48922.2021.00986

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) https://doi.org/10.48550/arXiv.1706.03762

[9] Ye, L., Rochan, M., Liu, Z., Wang, Y.: Cross-modal self-attention network for referring image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10502–10511 (2019). https://doi.org/10.1109/CVPR.2019.01075

[10] Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection https://doi.org/10.48550/arXiv.2010.04159

[11] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, pp. 213–229 (2020). https://doi.org/10.1007/978-3-030-58452-8_13 . Springer

[12] Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International Conference on Machine Learning, pp. 7354–7363 (2019). https://doi.org/10.48550/arXiv.1805.08318 . PMLR

[13] Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 843–852 (2017). https://doi.org/10.1109/ICCV.2017.97

[14] Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., *et al.*: Swin transformer v2: Scaling up capacity and resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12009–12019 (2022). https://doi.org/10.1109/CVPR52688.2022.01170

[15] Wang, W., Han, C., Zhou, T., Liu, D.: Visual recognition with deep nearest centroids. arXiv preprint arXiv:2209.07383 (2022) https://doi.org/10.48550/arXiv.2209.07383

[16] Knight, P.A.: The sinkhorn–knopp algorithm: convergence and applications. SIAM Journal on Matrix Analysis and Applications **30**(1), 261–275 (2008) https://doi.org/10.1137/06065962

[17] Liang, J.C., Cui, Y., Wang, Q., Geng, T., Wang, W., Liu, D.: Clusterformer: clustering as a universal visual learner. arXiv preprint arXiv:2309.13196 (2023) https://doi.org/10.48550/arXiv.2309.13196

[18] Liang, J., Zhou, T., Liu, D., Wang, W.: Clustseg: Clustering for universal segmentation. arXiv preprint arXiv:2305.02187 (2023) https://doi.org/10.48550/arXiv.2305.02187

[19] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pp. 234–241 (2015). https://doi.org/10.1007/978-3-319-24574-4_28 . Springer

[20] Alrfou, K., Kordijazi, A., Rohatgi, P., Zhao, T.: Synergy of unsupervised and supervised machine learning methods for the segmentation of the graphite particles in the microstructure of ductile iron. Materials Today Communications **30**, 103174 (2022) https://doi.org/10.1016/j.mtcomm.2022.103174

[21] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34**, 12077–12090 (2021)

[22] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021) https://doi.org/10.48550/arXiv.2102.04306

[23] Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M.: Swin-unet: Unet-like pure transformer for medical image segmentation. In: European Conference on Computer Vision, pp. 205–218 (2022). https://doi.org/10.1007/978-3-031-25066-8_9 . Springer

[24] Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H.R., Xu, D.: Unetr: Transformers for 3d medical image segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 574–584 (2022). https://doi.org/10.1109/WACV51458.2022.00181

[25] Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H.R., Xu, D.: Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 7th International Workshop, BrainLes 2021, Held in Conjunction with MICCAI 2021, Virtual Event, September 27, 2021, Revised Selected Papers, Part I, pp. 272–284

(2022). https://doi.org/10.1007/978-3-031-08999-2_22 . Springer

[26] Heidari, M., Kazerouni, A., Soltany, M., Azad, R., Aghdam, E.K., Cohen-Adad, J., Merhof, D.: Hiformer: Hierarchical multi-scale representations using transformers for medical image segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 6202–6212 (2023). https://doi.org/WACV56688.2023.00614

[27] Azad, R., Heidari, M., Shariatnia, M., Aghdam, E.K., Karimijafarbigloo, S., Adeli, E., Merhof, D.: Transdeeplab: Convolution-free transformer-based deeplab v3+ for medical image segmentation. In: Predictive Intelligence in Medicine: 5th International Workshop, PRIME 2022, Held in Conjunction with MICCAI 2022, Singapore, September 22, 2022, Proceedings, pp. 91–102 (2022). https://doi.org/10.1007/978-3-031-16919-9_9 . Springer

[28] Wang, J., Zhao, H., Liang, W., Wang, S., Zhang, Y.: Cross-convolutional transformer for automated multi-organs segmentation in a variety of medical images. Physics in Medicine & Biology **68**(3), 035008 (2023) https://doi.org/10.1088/1361-6560/acb19a

[29] Zhang, Y., Liu, H., Hu, Q.: Transfuse: Fusing transformers and cnns for medical image segmentation. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24, pp. 14–24 (2021). https://doi.org/10.1007/978-3-030-87193-2_2 . Springer

[30] Gao, Z.-J., He, Y., Li, Y.: A novel lightweight swin-unet network for semantic segmentation of covid-19 lesion in ct images. Ieee Access **11**, 950–962 (2022) https://doi.org/10.1109/ACCESS.2022.3232721

[31] Li, Z., Li, D., Xu, C., Wang, W., Hong, Q., Li, Q., Tian, J.: Tfcns: A cnn-transformer hybrid network for medical image segmentation. In: Artificial Neural Networks and Machine Learning–ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, September 6–9, 2022, Proceedings; Part IV, pp. 781–792 (2022). https://doi.org/10.1007/978-3-031-15937-4_65 . Springer

[32] Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)

[33] Aversa, R., Modarres, M.H., Cozzini, S., Ciancio, R., Chiusole, A.: The first annotated set of scanning electron microscopy images for nanoscience. Scientific data **5**(1), 1–10 (2018) https://doi.org/10.1038/sdata.2018.172

[34] DeCost, B.L., Hecht, M.D., Francis, T., Webler, B.A., Picard, Y.N., Holm, E.A.: Uhcsdb: ultrahigh carbon steel micrograph database: tools for exploring large heterogeneous microstructure datasets. Integrating Materials and Manufacturing Innovation **6**, 197–205 (2017) https://doi.org/10.1007/s40192-017-0097-0

[35] Christiansen, E., Marioara, C.D., Holmedal, B., Hopperstad, O.S., Holmestad, R.: Nano-scale characterisation of sheared $\beta$" precipitates in a deformed al-mg-si alloy. Scientific reports **9**(1), 17446 (2019) https://doi.org/10.1038/s41598-019-53772-4

[36] Mikkelsen, L.P., Fæster, S., Goutianos, S., Sørensen, B.F.: Scanning electron microscopy datasets for local fibre volume fraction determination in non-crimp glass-fibre reinforced composites. Data in Brief

**35**, 106868 (2021) https://doi.org/10.1016/j.dib.2021.106868

[37] Salling, F.B., Jeppesen, N., Sonne, M.R., Hattel, J.H., Mikkelsen, L.P.: Individual fibre inclination segmentation from x-ray computed tomography using principal component analysis. Journal of Composite Materials **56**(1), 83–98 (2022) https://doi.org/10.1177/00219983211052741

[38] Masubuchi, S., Watanabe, E., Seo, Y., Okazaki, S., Sasagawa, T., Watanabe, K., Taniguchi, T., Machida, T.: Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials. npj 2D Materials and Applications **4**(1), 3 (2020) https://doi.org/10.1038/s41699-020-0137-z

[39] Boiko, D.A., Pentsak, E.O., Cherepanova, V.A., Ananikov, V.P.: Electron microscopy dataset for the recognition of nanoscale ordering effects and location of nanoparticles. Scientific data **7**(1), 101 (2020) https://doi.org/10.1038/s41597-020-0439-1

[40] Creveling, P., Whitacre, W., Czabaj, M.: Synthetic x-ray microtomographic image data of fiber-reinforced composites (2019)

[41] Klinkmüller, M., Schreurs, G., Rosenau, M., Kemnitz, H.: Properties of granular analogue model materials: A community wide survey. Tectonophysics **684**, 23–38 (2016) https://doi.org/10.1016/j.tecto.2016.01.017

[42] Van Stone, R., Low, J., Shannon, J.: Investigation of the fracture mechanism of ti-5ai-2.5 sn at cryogenic temperatures. Metallurgical Transactions A **9**, 539–552 (1978) https://doi.org/10.1007/BF02646411

[43] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)

[44] Synapse multi-organ segmentation dataset. https://www.synapse.org/#!Synapse:syn3193805/wiki/217789 (2015)

[45] Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., *et al.*: Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999 (2018) https://doi.org/10.48550/arXiv.1804.03999

[46] Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018). https://doi.org/10.1109/TPAMI.2019.2913372

[47] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500 (2017). https://doi.org/10.1109/CVPR.2017.634

[48] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017). https://doi.org/0.1609/aaai.v31i1.11231

[49] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017). https://doi.org/10.1109/CVPR.2017.243

[50] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) https://doi.org/10.48550/arXiv.1409.1556

[51] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018). https://doi.org/10.1109/CVPR.2018.00474

[52] Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp. 6105–6114 (2019). PMLR

# Appendix A Average Performance of Segmentation Algorithms on Microscopy Images

In this section, we compare the effect of pre-training on the average performance of CNN-based, Transformer-based segmentation algorithms, and CS-UNet. After that, we compare the average performance of the three types of algorithms. Our results indicate that pre-training on microscopy images generally has positive impacts to the performance. CS-UNet outperforms UNet in all experiments while it has similar or better performance than Transformer-based algorithms.

## A.1 CNN-based Image Segmentation

**Table 8** The average performance of **UNet** when it is initialized with different pre-training weights. Each entry shows the mean and standard deviation of IoU value for a pre-training model. The best score per test is in bold.

| Test data | ImageNet | MicroNet | ImageNet→MicroNet |
|---|---|---|---|
| Super-1 | **96.09%±0.20%** | 95.99%±0.16% | 95.92%±0.19% |
| Super-2 | 95.08%±1.27% | 95.28%±0.26% | **95.38±1.03%** |
| Super-3 | 63.30%±4.55% | 74.61%±17.21% | **78.69±11.24%** |
| Super-4 | 71.46%±6.88% | 75.1%±9.0% | **77.78%±0.17** |
| EBC-1 | 95.18%±0.82% | 94.65%±1.44% | **95.69%±1.03%** |
| EBC-2 | 80.74%±10.76% | **87.06%±1.55%** | 86.0%±5.17% |
| EBC-3 | 39.36%±7.81% | 41.95%±4.89% | **46.44%±4.35%** |

We examine the performance of UNet [19] with 3 types of pre-trained encoders. We include this result since the configurations of CS-UNet only used 19 of the 35 CNN encoders in Stuckner *et al.* [6]. As shown in Appendix B, the 19 encoders have top-5 accuracy in at least one of the segmentation tasks. This selection reduces the number of experiments needed for a fair comparison,

The average performance of UNet when it is pre-trained with ImageNet or MicroNet is in Table 8, which indicates that ImageNet→MicroNet model (i.e. CNN encoders initialized with ImageNet model and fine-tuned with MicroNet) achieves the best outcome in majority of the cases. The configurations of the top-performing CNN encoders are shown in Table 9, which also indicates that pre-training on MicroNet provides better outcome in most of the cases. EfficientNet and Se_ResNet families are better than the older families such as VGG.

**Table 9** The best performing CNN encoders of UNet that are pre-trained on ImageNet and MicroNet

| | ImageNet | | MicroNet | |
|---|---|---|---|---|
| Dataset | best performing CNN encoder | IoU | best performing CNN encoder | IoU |
| Super-1 | SE ResNeXt-50 32x4 | **96.2%** | SE_ResNeXt-101_32x4d | 95.95% |
| Super-2 | EfficientNet-b5 | 95.45% | SENet-154 | **95.50%** |
| Super-3 | EfficientNet-b3 | 70.74% | EfficientNet-b3 | **92.5%** |
| Super-4 | EfficientNet-b2 | 77.34% | EfficientNet-b1 | **78.95%** |
| EBC-1 | SENet-154 | 96.23% | SENet-154 | **96.67%** |
| EBC-2 | DenseNet201 | **91.36%** | InceptionResnetV2 | 90.97% |
| EBC-3 | EfficientNet-b3 | 44.4% | EfficientNet-b3 | **52.84%** |

Unsurprisingly, our results are largely consistent with that of Stuckner *et al.* [6], since we initialized our networks with the weights pre-trained on MicroNet. Specifically, pre-training with MicroNet improves the performance of one-shot and out-of-distribution learning. Since we picked CNN encoders that have top-5 performance in at least one experiment, the IoU scores are higher than the average scores shown in Stuckner *et al.*. In fact, the performances on Super-2 (few-shot learning) are basically the same with different pre-training models.

## A.2 Transformer-based Image Segmentation

**Table 10** Average performance of **Transformer**-based segmentation algorithms when initialized with different pre-trained weights. Each entry shows the mean and standard deviation of IoU value for a pre-training model. The highest score for each dataset is in bold.

| | Original Swin-T | | | Intermediate Swin-T | |
|---|---|---|---|---|---|
| *Test set* | *ImageNet* | *MicroLite* | *ImageNet→MicroLite* | *MicroLite* | *ImageNet→MicroLite* |
| Super-1 | 94.94%±0.31% | **95.0%±0.43%** | 94.89%±0.45% | 94.72%±0.97% | 94.41%±0.51% |
| Super-2 | 93.26%±0.76% | 93.83%±0.62% | 93.55%±0.30% | **94.03%±0.83%** | 93.43%±0.97% |
| Super-3 | 76.24%±7.06% | **79.53%±13.69%** | 70.65%±10.74% | 78.74%±17.08% | 69.31%±6.44% |
| Super-4 | 73.04%±1.64% | **73.18%±1.81%** | 72.89%±2.93% | 72.10%±2.39% | 71.23%±2.32% |
| ECB-1 | 91.73%±3.40% | 94.67%±1.59% | 91.44%±2.87% | **94.95%±1.41%** | 90.56%±3.01% |
| ECB-2 | 82.47%±5.90% | 86.02%±4.98% | 83.20%±3.34% | **86.67%±2.61%** | 83.07%±4.22% |
| ECB-3 | 52.21%±6.76% | **65.90%±13.12%** | 55.15%±10.42% | 56.91%±5.24% | 49.03%±4.67% |
| mean IoU | 80.56 | **84.02** | 80.25 | 82.59 | 78.72 |

Table 10 shows the average performance of Transformer-based segmentation algorithms (Swin-Unet, HiFormer, and TransDeepLabv3+) using different configurations of pre-training and Swin-T architectures. We compared the algorithms by using the original or the intermediate Swin-T architecture and by initializing their weights with ImageNet or microscopy pre-training models. Our results indicate that the algorithms perform well with the MicroLite pre-training model and that the original Swin-T architecture is slightly better with 1-shot learning and out-of-distribution learning. Overall, pre-training with microscopy images provided better results for Transformer-based segmentation algorithms than pre-training on natural images.

## A.3 CS-UNet Image Segmentation

**Table 11** Average performance of **CS-UNet** when initialized with different pre-trained weights for each experiment. Each entry in the table shows the mean value and standard deviation of the evaluation IoU metric for a particular pre-training model. The highest accuracy score for each dataset is shown in bold.

| | Original Swin-T | | | Intermediate Swin-T | |
|---|---|---|---|---|---|
| *Test set* | *ImageNet* | *Microscopy* | *ImageNet→Microscopy* | *Microscopy* | *ImageNet→Microscopy* |
| Super-1 | 96.11%±0.11% | 96.19%±0.13% | 96.16%±0.14% | **96.22%±0.15%** | 96.02%±0.34% |
| Super-2 | 95.63%±0.23% | 95.63%±0.44% | 95.78%±0.14% | 95.67%±0.46% | **95.86%±0.11%** |
| Super-3 | 78.64%±9.3% | 76.59%±15.78% | 78.01%±13.32% | 78.18%±12.13% | **80.68%±12.60%** |
| Super-4 | 73.74%±3.74% | **77.25%±3.31%** | 72.87%±6.26% | 76.65%±2.53% | 75.07%±1.10% |
| ECB-1 | **97.09%±0.86%** | 95.48%±1.04% | 96.4%±0.79% | 94.72%±1.37% | 96.21%±0.78% |
| ECB-2 | 83.57%±7.71% | 86.12%±1.76% | **88.58%±1.46%** | 86.41%±1.65% | 88.08%±2.98% |
| ECB-3 | 45.88%±10.12% | 44.7%±8.57% | 46.08%±13.92% | **46.35%±9.91%** | 45.16%±10.58% |
| mean IoU | 81.5 | 81.71 | 81.98 | 82.03 | **82.44** |

We also compare the performance of our hybrid segmentation algorithm CS-UNet in Table 11 when it uses the original or the intermediate Swin-T architecture and when it is initialized with weights from ImageNet or microscopy models. Since CS-UNet uses both CNN and Transformer encoders, the results are mixed where pre-training with microscopy images does not provide better performance in all cases. The weaker performance of CNN encoders reduced the advantage of Transformer encoders when they are pre-trained on microscopy images. When we consider the mean IoU scores across all experiments, however, pre-training with microscopy images still has the better outcome.

## A.4 Comparison of CS-UNet, CNN-based, and Transformer-based Algorithms

**Table 12** The average performance of CNN, Transformer, and CS-UNet on all datasets. Each entry in the table shows the mean value and standard deviation of the IoU evaluation metric for a particular method.

| *Test set* | *UNet [19]* | *CS-UNet* | *Swin-Unet [23]* | *TransDeepLabV3+ [27]* | *HiFormer [26]* |
|---|---|---|---|---|---|
| Super-1 | 95.98% ± 0.20% | **96.14**% ± 0.21% | 95.30%±0.34 % | 94.91% ± 0.47% | 94.30% ± 0.58% |
| Super-2 | 95.24% ± 0.96% | **95.70**% ± 0.33% | 93.39% ± 1.1% | 94.1% ± 0.61% | 93.11% ± 0.5% |
| Super-3 | 72.20% ± 13.79% | 77.99% ± 12.90% | **83.0**% ± 6.14% | 81.99% ± 8.29% | 64.19% ± 8.14% |
| Super-4 | 74.57% ± 7.23% | 75.08% ± 4.15% | **77.16**% ± 0.97% | 70.95% ± 1.86% | 72.61% ± 1.67% |
| EBC-1 | 95.17% ± 1.21% | **95.98**% ± 1.28 | 93.49± 2.66% | 91.59% ± 4.13% | 93.96% ± 1.1% |
| EBC-2 | 84.60% ± 7.48% | **86.73%± 4.03%** | 83.13% ± 3.1% | 83.1% ± 5.68% | 86.67% ± 1.44% |
| EBC-3 | 42.58% ± 6.57% | 45.69% ± 10.78% | 58.62% ± 20.37% | **58.7**% ± 9.70% | 52.84% ± 5.41% |

In Table 12, we compare the performance of UNet, CS-UNet, and Transformer-based algorithms (Swin-Unet, HiFormer, and TransDeepLabv3+) averaged over different pre-training models and Swin-T architectures. The results show that CS-UNet is better than UNet on average across all experiments. While Transformer-based segmentation algorithms may be superior in 1-shot learning or out-of-distribution learning, their performance is not always consistently better than UNet. This result indicates that our hybrid algorithm CS-UNet is more robust regardless of the pre-training models.

# Appendix B CNN Encoders Chosen for the Evaluation of CS-UNet and UNet

**Table 13** Each encoder has at least 1 top-5 average IoU score for Super/EBC datasets (based on Figure 4 in [6]).

| CNN Encoder | Super-1 | Super-2 | Super-3 | Super-4 | EBC-1 | EBC-2 | EBC-3 |
|---|---|---|---|---|---|---|---|
| SE_ResNet-50 [46] | | | | | | | ✓ |
| SE_ResNeXt-50_32x4d [47] | ✓ | ✓ | | | ✓ | ✓ | ✓ |
| SE_ResNeXt-101_32x4d [47] | ✓ | ✓ | | | | | ✓ |
| SE_ResNet-152 [46] | ✓ | | | | ✓ | | |
| SE_ResNet-101 [46] | | | | | ✓ | ✓ | ✓ |
| SENet-154 [46] | | ✓ | ✓ | | ✓ | | |
| ResNeXt-101_32x8d [47] | ✓ | | | | | | |
| Inception-V4 [48] | | ✓ | ✓ | | ✓ | | |
| Inception-ResNet-V2 [48] | | | ✓ | | | | |
| DenseNet201 [49] | | | | | | ✓ | |
| DenseNet161 [49] | | | | | | ✓ | |
| VGG-16_bn [50] | ✓ | | ✓ | | | ✓ | |
| VGG-13_bn [50] | | | ✓ | | | | |
| MobileNet-V2 [51] | | | | | | | ✓ |
| EfficientNet-b1 [52] | | | | ✓ | | | |
| EfficientNet-b2 [52] | | | | ✓ | | | |
| EfficientNet-b3 [52] | | | | ✓ | | | |
| EfficientNet-b4 [52] | | | | ✓ | | | |
| EfficientNet-b5 [52] | | ✓ | | ✓ | | | |

# Appendix C CS-UNet Complexity

CS-UNet is a flexible method that can use different CNN families in the CNN encoder branch. As a result, the total number of parameters of CS-UNet can vary depending on the specific CNN encoder branch chosen. Table 14 summarizes the parameters for different CNN branches used in CS-UNet for this study.

# Appendix D Segmentation Examples of Super and EBC Datasets

**Table 14** The number of parameters of CS-UNet based on different CNN families in CNN encoder branch.

| **CNN encoder branch** | **# of Params (M) of CS-UNet** |
|---|---|
| DenseNet201 [49] | 50.98 |
| DenseNet161 [49] | 59.99 |
| EfficientNet-b0 [52] | 33.68 |
| EfficientNet-b1 [52] | 36.18 |
| EfficientNet-b2 [52] | 37.43 |
| EfficientNet-b3 [52] | 40.48 |
| EfficientNet-b4 [52] | 47.44 |
| EfficientNet-b5 [52] | 58.34 |
| Inception-V4 [48] | 73.11 |
| Inception-ResNet-V2 [48] | 86.29 |
| SENet-154 [46] | 145.83 |
| SE_ResNet-50 [46] | 58.83 |
| SE_ResNet-152 [46] | 97.56 |
| SE_ResNeXt-50_32x4d [47] | 58.3 |
| SE_ResNeXt-101_32x4d [47] | 79.7 |
| SE_ResNet-101 [46] | 80.07 |
| ResNeXt-101_32x8d [47] | 119.53 |
| MobileNet-V2 [51] | 33.36 |
| VGG-13_bn [50] | 39.64 |
| VGG-16_bn [50] | 44.96 |

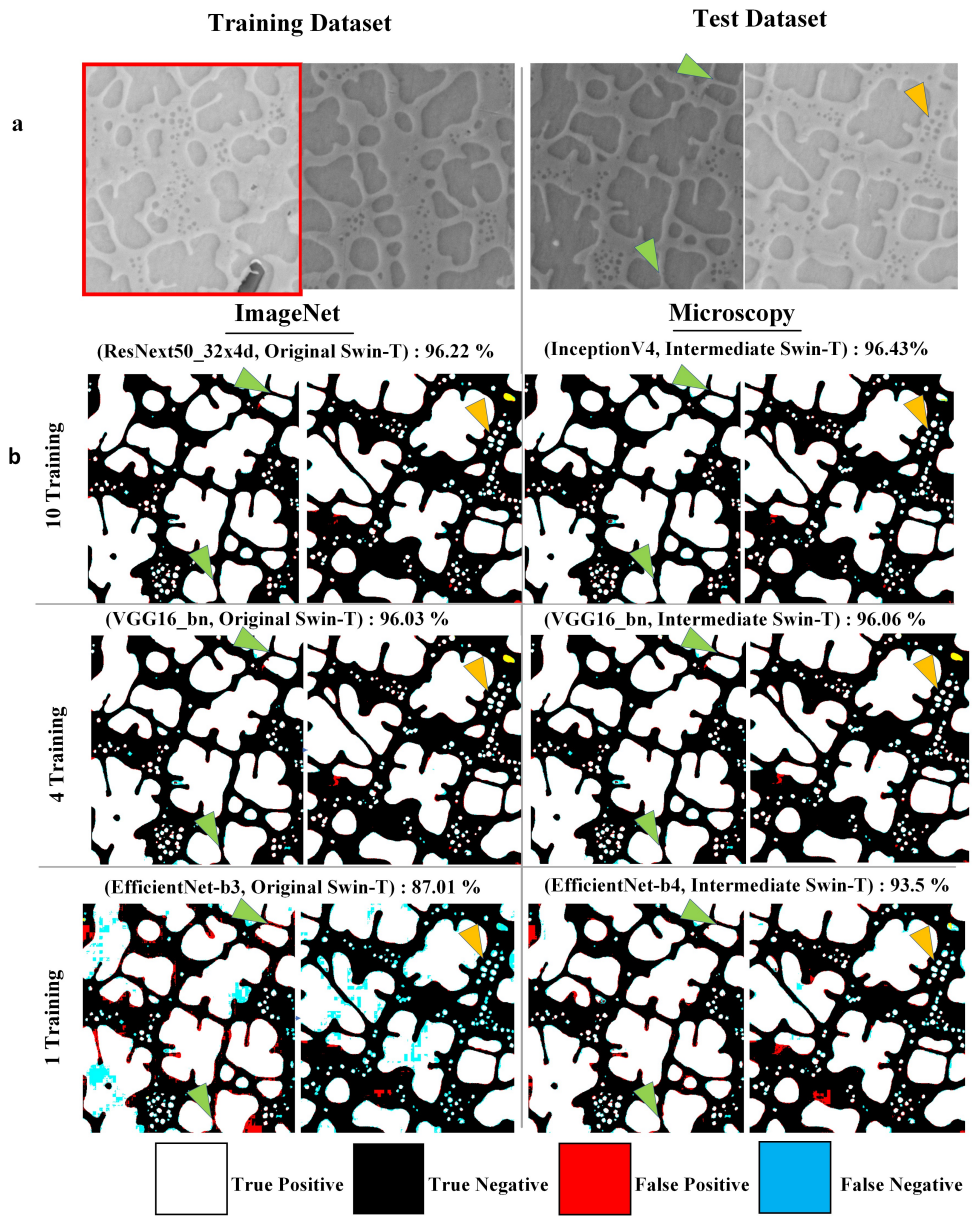


**Fig. 7** (a) examples of the training and test images of the Super datasets [6], where the training image of Super-3 is outlined in red. (b) the best segmentation masks of the test images when CS-UNet is trained with Super-1, Super-2, and Super-3 dataset, where the left is pre-trained with ImageNet and the right is pre-trained with microscopy images. The CNN/Transformer encoders and the IoU score of the best model are above the segmentation mask of each experiment. The green triangle points to the area where the ImageNet (but not the microscopy) model incorrectly segmented the secondary precipitates. The yellow triangle points to the area where the ImageNet (but not the microscopy) model incorrectly segmented the tertiary precipitates.

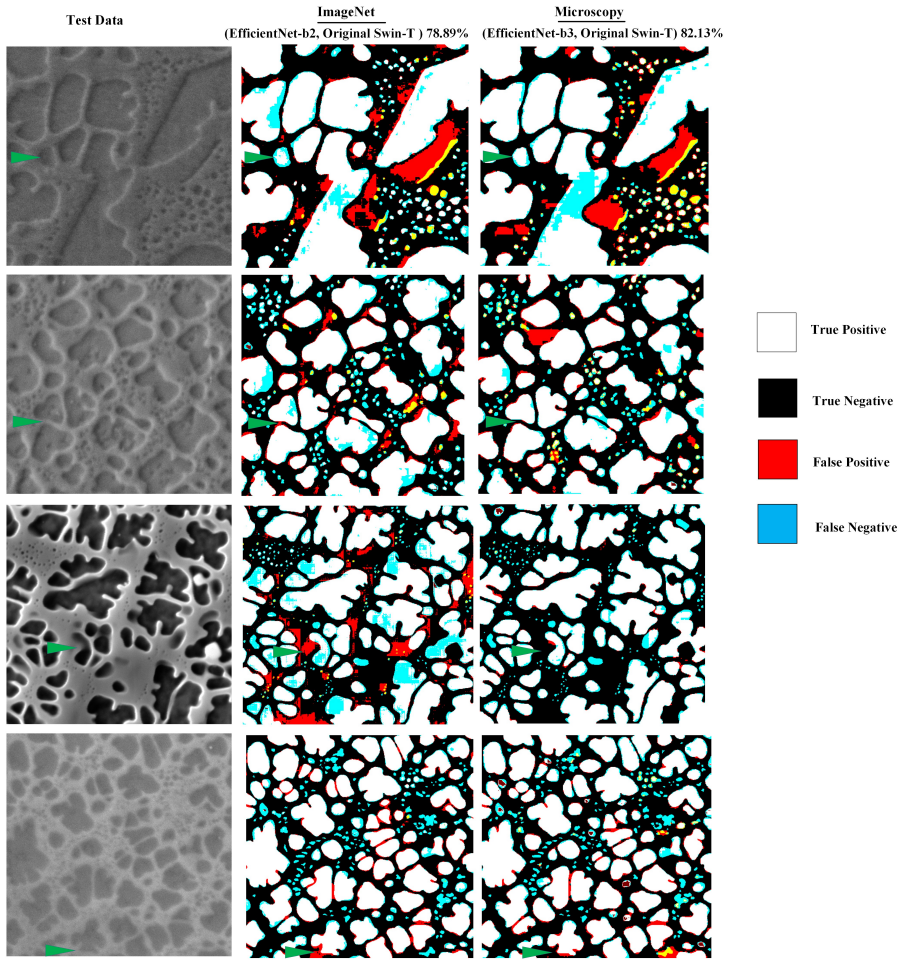


**Fig. 8** The test images of Super-4 have different imaging conditions than those of the training images. The left column shows the test images of Super-4 [6]. The accuracy mask colors are the same as those in Figure 7. The middle column shows the IoU accuracy masks for the best ImageNet model. The right column shows the same for the best microscopy model. Each row shows the test image and accuracy masks of the same image. The green arrow shows an example where the model was over-segmented and where the microscopy model accurately segmented the secondary precipitate. The yellow color indicates incorrect identification of a tertiary precipitate as a secondary precipitate. The maroon color indicates where the model improperly identified a secondary precipitate as a tertiary precipitate. The cyan color indicates where the model over-segmented the secondary precipitates.

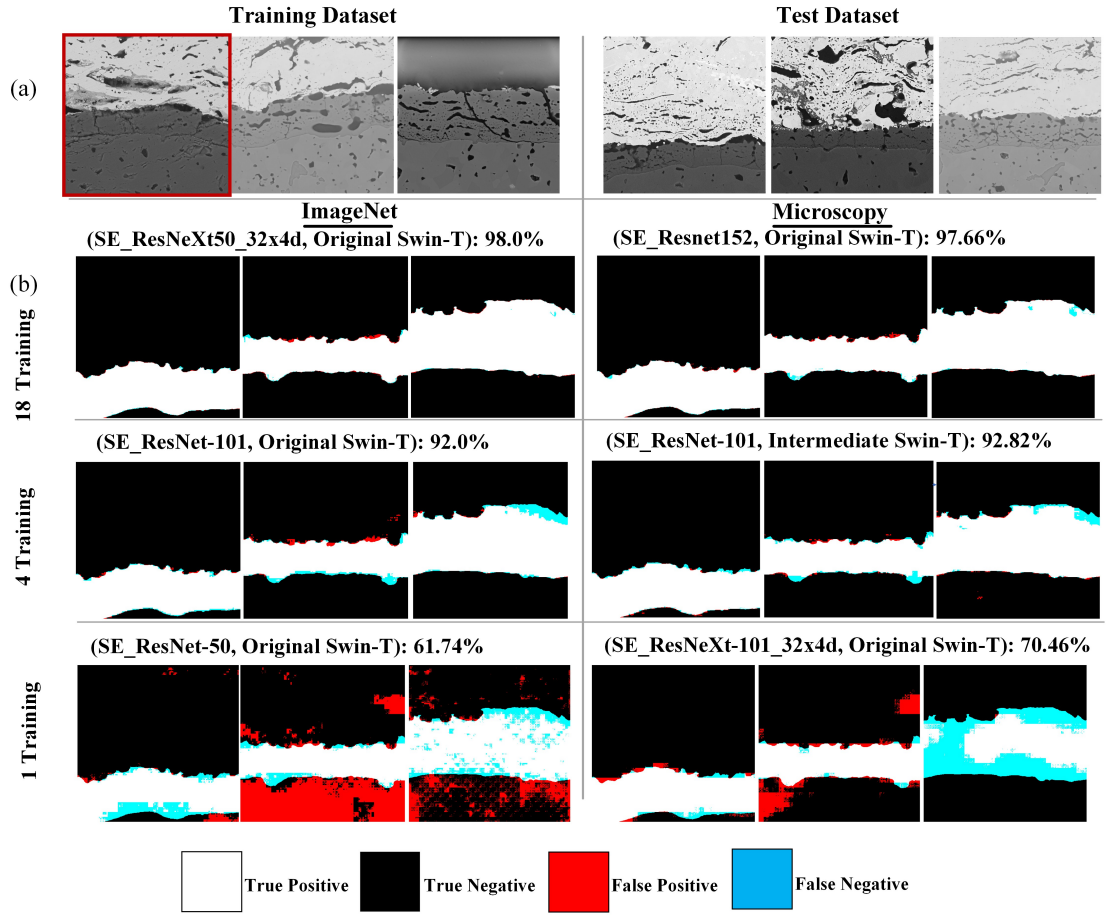


**Fig. 9** (a) shows examples of the train and test images in the EBC datasets, where the left column represents the training set and the right column represents the test set. The single training image for EBC-3 is outlined in red. (b) shows the best segmentation masks of CS-UNet for each EBC dataset and for each pre-training dataset. The CNN/Transformer encoders and the IoU score is above the segmentation mask of each test. For EBC-1/2 (the second/third rows), the segmentation masks of CS-UNet with either pre-training datasets are able to distinguish between the substrate and the thermally grown oxide layer. For EBC-3 (the last row), CS-UNet pre-trained with ImageNet is not able to distinguish between the substrate and the thermally grown oxide layer, which made it difficult to accurately measure oxide thickness after simple morphological operations.