SCTS: Instance Segmentation of Single Cells Using a Transformer-Based Semantic-Aware Model and Space-Filling Augmentation

Yating Zhou [1, 2], Wenjing Li [1, 2], Ge Yang [1, 2 *]
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
{zhouyating2020, wenjing.li}@ia.ac.cn, {yangge}@ucas.edu.cn

Abstract

Instance segmentation of single cells from microscopy images is critical to quantitative analysis of their spatial and morphological features for many important biomedical applications, such as disease diagnosis and drug screening. However, the high densities, tight contacts, and weak boundaries of the cells pose substantial technical challenges. To overcome these challenges, we have developed a new instance segmentation model, which we refer to as single-cell Transformer segmenter (SCTS). It utilizes a Swin Transformer as its backbone, combining the global modeling capabilities of a Transformer and the local modeling capabilities of a convolutional neural network (CNN) to ensure model adaptability to different cell sizes, shapes, and textures. It also embeds a three-class (background, cell interior, and cell boundary) semantic segmentation branch to classify pixels and to provide semantic features for downstream tasks. The prediction of boundary semantics improves boundary awareness, and the differentiation between foreground and background semantics improves segmentation integrity in regions with weak signals. To reduce the need for annotated training data, we have developed an augmentation strategy that randomly fills instances of single cells into open spaces of training images. Experiments show that our model outperforms several state-of-the-art models on the LIVECell dataset and an in-house dataset. The code and dataset of this work are openly accessible at https://github.com/cbmi-group/SCTS.

1. Introduction

Extracting individual cells from their microscopy images through instance segmentation is critical to analysis of biological and medical samples at the single-cell level for important applications such as disease diagnosis and drug screening [2, 26, 34]. In instance segmentation, pixels not only must be classified into different semantic classes but also must be grouped into individual instances. This is a challenging task because cells in tissues or cultures often have high densities, tight contacts, and weak boundaries.

Early models such as the U-Net [31] achieve excellent performance in semantic segmentation of cell images but cannot resolve individual cells. To address this problem, a variety of instance segmentation methods [33, 4, 36] have been developed by combining semantic segmentation with different post-processing strategies. These methods are intuitive and show good adaptability to cells of different sizes and shapes, but they are not fully end-to-end for training. Their performance is also limited by their post-processing operations. They cannot handle overlapping cells because they can only assign one instance label to each pixel.

Mask R-CNN [15] is a two-stage instance segmentation network that first generates a series of proposals using the Region Proposal Network (RPN) [29] and then performs classification and coordinate regression on the proposals. It can handle segmentation in the presence of overlap between image objects. However, as shown in Figure 1, it tends to produce incorrect boundary predictions in regions of tight cell contacts and incomplete segmentation masks in regions of weak signals. Hybrid Task Cascade (HTC) [6] is an improvement over Mask R-CNN. It introduces a semantic segmentation branch to distinguish real foreground from cluttered background, which helps to recover missing detections in regions with weak signals but brings side effects such as merging different cells in tight contacts into one instance.

Overall, convolutional neural networks (CNNs) such as the U-Net and the Mask R-CNN are commonly used in vision tasks and have achieved impressive performance. However, they have difficulty capturing long-range dependencies because they are limited by the size of their convolution kernels. This limitation in turn makes it difficult for downstream tasks to adapt to changes in cell shape, size, and texture. The Transformer architecture provides an effective solution to overcome these shortcomings [24]. In this

*Corresponding Author.

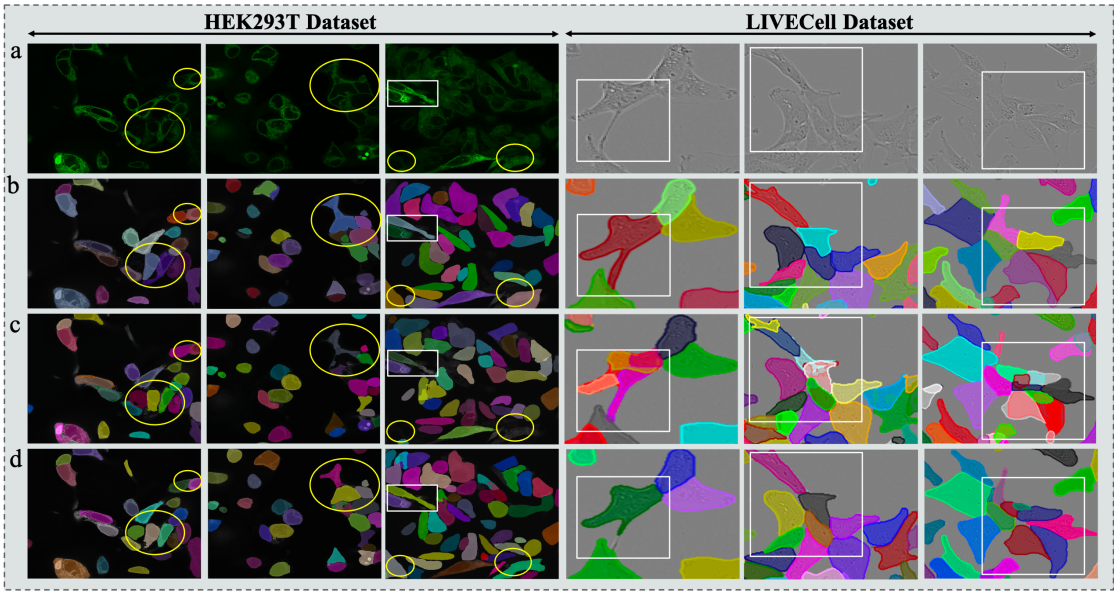


Figure 1: Comparison of performance of Mask R-CNN versus the SCTS model developed in this study. (a) Raw images. (b) Ground truth. (c) Predictions by Mask R-CNN. (d) Predictions by SCTS. Ovals highlight predictions in regions with weak signals. White rectangular boxes highlight boundary predictions under tight contacts and diffusive boundaries of cells.

study, inspired by the aforementioned models, we propose a Transformer-based semantic-aware architecture for fully end-to-end single-cell instance segmentation. It addresses several key issues in instance segmentation of single cells. First, cells show large variations in their sizes and shapes. It is difficult for CNNs to handle these variations. We introduce the Swin Transformer [24] as the backbone network, which can effectively capture global information and long-distance dependencies and can better adapt to changes in cell shape and size. Second, in cases with high cell densities, existing methods have difficulty identifying cell boundaries. We embed a three-class semantic branch to distinguish background, cell boundary, and cell interior to better differentiate cells in tight contacts. Third, manual annotation of microscopy images remains laborious and time-consuming. We propose a new strategy to augment small-scale datasets. In summary, we make the following research contributions:

- We have developed a novel model for instance segmentation of single cells. To ensure its adaptability to variations in cell size, shape, and texture, it adopts Swin Transformer as the backbone network to model global as well as local features.

- We propose to embed a three-class semantic branch to the Transformer backbone to effectively capture semantic information. The prediction of boundary semantics improves the discrimination of cell boundaries, and the prediction of foreground and background semantics improves segmentation integrity in regions with weak signals.

- We have developed a new data augmentation strategy named space-filling, which effectively and substantially increases training images by filling cell instances randomly into their object-free background regions.

2. Related Work

2.1. Instance Segmentation of Single Cells

Semantic segmentation of single-channel microscopy images classifies their pixels into two semantic classes: the foreground and the background. Fully Convolutional Networks (FCN) [25] is one of the early semantic segmentation models. It uses a fully convolutional neural network that greatly improves the accuracy of semantic segmentation. But its segmentation results often have poor details because contextual information is not considered. Based on FCN, Ronneberger and colleagues propose the U-Net, which uses a U-shaped encoder-decoder architecture that combines low-resolution and high-resolution information through skip connections. U-Net has achieved good segmentation performance on medical images and has become a widely adopted baseline model. Still, its architecture is not designed to handle instance segmentation of single cells.

Several studies have proposed to extend semantic segmentation to instance segmentation by incorporating post-processing operations [1, 4, 14, 30, 33, 36]. For example, Greenwald and colleagues propose a method named Mesmer [14] that divides instance segmentation into two pixel-

level prediction tasks. The first task predicts whether a pixel is inside a cell, at the cell boundary, or part of the image background. The second task predicts the distance of each pixel inside the cell to the cell centroid. Finally, a watershed segmentation algorithm is performed on the predicted results to separate different cell instances. These methods are intuitive and interpretative, but their segmentation accuracy is limited by their post-processing strategies. They cannot handle overlapping cells.

Benefitting from detection models such as R-CNN [13] and its variants [3, 12, 29], YOLO [28], and SSD [23], another common strategy for instance segmentation is a two-stage approach combining detection and segmentation. Examples of this strategy include FCIS [19], Mask R-CNN, MaskLab [8],Mask Scoring RCNN [17], HTC [6], and PointRend [18]. These methods first extract ROIs using convolutional layers and then perform segmentation and classification on each ROI. They can handle overlapping objects but are less effective for segmentation at tight contacts, diffusive boundaries, and regions of weak signals, which are common in cell images.

Some studies convert pixel-level classification tasks into pixel-level regression tasks to achieve instance segmentation [27, 32, 35, 38, 39, 40]. For example, the Stardist method [32] transforms the instance segmentation task into a prediction problem with a fixed number of points on each image object contour. For each pixel, the Stardist predicts its object class probability and star-convex polygons parameterized by the radial distances to capture cell instances. MultiStar [35] and SplineDist [27] are further extensions of the Stardist method. However, due to limitations of their model representation, these methods have poor performance on non-convex objects.

2.2. Vision Transformer

Despite the success of CNN-based cell instance segmentation methods, it is difficult for CNNs to capture long-distance dependencies and global contextual information due to the limitations of convolutional kernels. Recently, methods based on Transformer have achieved excellent performance in many vision tasks such as classification, detection, and segmentation. DEtection Transformer (DETR) [5] is the first Transformer-based end-to-end model for object detection. It treats the object detection task as a set prediction problem, removing many hand-crafted operations such as non-maximum suppression or anchor generation. It achieves comparable accuracy and running time as Faster R-CNN. Vision Transformer (ViT) [10] applies pure Transformer on sequences of image patches and achieves promising results in image classification tasks. Swin Transformer solves the problem of high resolution and large-scale variation of images faced in migrating from texts to images. It extends the Transformer to pixel-level dense prediction tasks by building hierarchical features. Its excellent performance on multiple vision tasks demonstrates its potential to be a general backbone network for vision tasks. In the instance segmentation model developed in this study, we employ the Swin Transformer as the backbone network to capture long-range dependencies and global contextual information, which is critical for the model to adapt to different cell sizes, shapes, and textures.

3. Method

3.1. Architecture Overview

Our instance segmentation model is an improvement over Mask R-CNN, a two-stage instance segmentation model. In the first stage, Mask R-CNN generates a series of candidate object bounding boxes by its RPN. In the second stage, Mask R-CNN extracts ROIs from each candidate box by RoIAlign and then performs classification and bounding box regression on the ROIs by its detection head and binary segmentation on ROIs by its segmentation head (Figure 2d).

Figure 2 gives an overview of our proposed model architecture. Compared to Mask R-CNN, our model differs in these key aspects: (a) It employs a Swin Transformer instead of a ResNet [16] as the backbone network to ensure model adaptability to cell size, shape, and texture (Figure 2a, Section 3.3). (b) It embeds a semantic segmentation branch to distinguish between background, cell interior, and cell boundary to improve boundary perception and foreground integrity for downstream tasks (Figure 2c, Section 3.4). (c) Before each batch, it augments training images online by randomized filling of instances of single cells into open spaces of training images using the space-filling augmentation strategy described below (Figure 2e, Section 3.2).

3.2. Space-Filling Augmentation Strategy

In practice, labeled biomedical images often are limited, and the cost of labeling is high. To fully utilize available annotated training data and improve the diversity of training images, we design a new data augmentation strategy that we refer to as Space-Filling. The workflow is depicted in Figure 3, which consists of two steps: database construction and space-filling. In the database construction step, using the bounding box annotation and segmentation annotation of the instances, we crop out images of cell instance regions, setting background pixel values to zero to avoid instances from overlapping with each other. Finally, we collect all the instance images. In the space-filling step, we first randomly select a specific number of cell instances from the cell database. We then transform the instances using strategies such as rotation and random image intensity transformation. Finally, we superimpose the foreground of individual cell instances onto the open spaces of training images where the binary mask intersection of the training image objects and

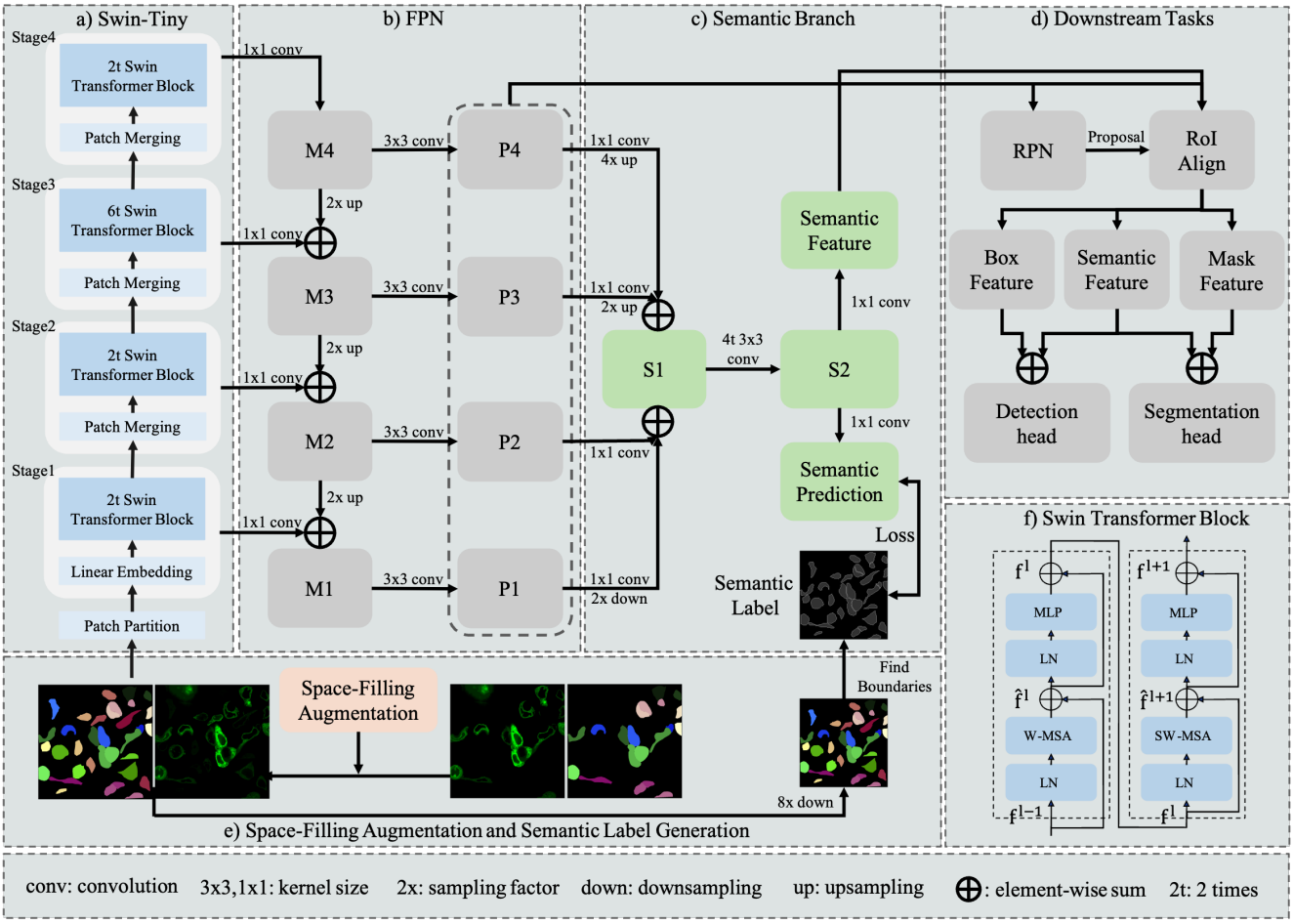


Figure 2: Overview of the proposed model. (a) Swin-Tiny backbone network. (b) Feature Pyramid Networks (FPN) [21]. (c) Three-class semantic segmentation branch. (d) Downstream detection and segmentation tasks. (e) Space-Filling augmentation strategy and semantic label generation flow. (f) Detailed structure of the Swin Transformer block.

the inserted instance image objects is empty. This augmentation is carried out online during training, so even for the same training image, the processing results are different in each batch, which effectively increases the diversity of training images and alleviates the need for manual annotation.

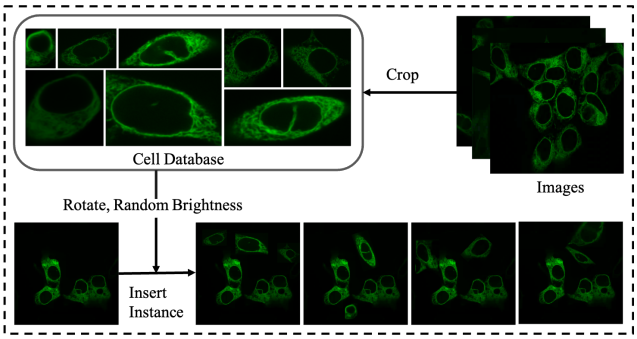


Figure 3: Workflow of data augmentation using the space-filling strategy.

3.3. Swin Transformer Backbone

Swin Transformer contains four versions: Swin-Tiny, Swin-Small, Swin-Base, and Swin-Large, where each version has twice the model size and computational complexity of the previous one. In this study we use Swin-Tiny. Its overall architecture is shown in Figure 2a. It contains four stages, the numbers of Swin Transformer blocks from stage 1 to stage 4 are 2, 2, 6, and 2, respectively. For an input image, Swin-Tiny first divides it into non-overlapping patches through a patch partition operation. In our implementation, we set the patch size to 4×4. For an RGB image, the dimension of each patch is 4×4×3. Then, Swin-tiny applies a linear embedding layer on the original features to transform them into an arbitrary dimension. Next, Swin-Tiny processes the features through several Swin Transformer blocks and employs a patch merging layer to concatenate the features of each group of 2×2 neighboring to achieve downsampling. Stages 2, 3, 4 are constructed in a similar

process and the outputs of these stages are combined into a hierarchical representation.

Swin Transformer Block. Figure 2f shows the detailed structure of the Swin Transformer Block, where MLP denotes the multilayer perceptron module, LN denotes the layernorm module, W-MSA denotes the multi-head self-attention module, SW-MSA denotes the shifted windows multi-head self-attention module, $\hat{\mathbf{f}}^l$ denotes output features of the (S)W-MSA module of block l, and $\mathbf{f}^l$ denotes the output features of the MLP module for block l. The Swin Transformer Block can be summarized by the following group of equations:

$$\begin{aligned}
\hat{\mathbf{f}}^l &= \text{W-MSA}\left(\text{LN}\left(\mathbf{f}^{l-1}\right)\right) + \mathbf{f}^{l-1}, \\
\mathbf{f}^l &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{f}}^l\right)\right) + \hat{\mathbf{f}}^l, \\
\hat{\mathbf{f}}^{l+1} &= \text{SW-MSA}\left(\text{LN}\left(\mathbf{f}^l\right)\right) + \mathbf{f}^l, \\
\mathbf{f}^{l+1} &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{f}}^{l+1}\right)\right) + \hat{\mathbf{f}}^{l+1}.
\end{aligned} \tag{1}$$

3.4. Semantic Segmentation Branch

Figure 1 highlight the problems of incomplete segmentation results in regions with weak signals and inaccurate boundary prediction in regions with tight cell contacts when using Mask R-CNN for instance segmentation. Inspired by HTC, we address these problems by designing a semantic segmentation branch with three classes of labels: background, cell interior, and cell boundary.

Semantic Label Generation. Figure 2e shows the workflow of semantic label generation. For an input image, we first perform data augmentation using the space-filling strategy. We then assign different pixel values to each instance according to segmentation annotations to generate a mask. Next, we downsample the mask to match in size with the shape of semantic prediction for loss calculation. Finally, we apply findboundaries function of the skimage library to the mask to generate three-class semantic labels of cell boundary, cell interior, and background. Specifically, the semantic class of cell boundary class is composed of a one-pixel-wide edge. The semantic class of cell interior consists of foreground pixels other than the one-pixel-wide boundary. The remaining pixels are assigned to the semantic class of background.

Semantic Segmentation Loss. For the semantic segmentation branch, we use the following weighted cross-entropy loss function:

$$Loss = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} w_c y_c^{(n)} log\left(\hat{y}_c^{(n)}\right), \tag{2}$$

where N denotes the number of samples, C denotes the number of semantic classes, and w_c denotes the loss weight of class c. $y_c^{(n)}$ is the symbolic function that equals 1 when the true class of sample n is c and 0 otherwise. $\hat{y}_c^{(n)}$ denotes the probability that sample n belongs to class c. We assign different loss weights to different semantic classes to balance between the classes. In the ablation study, we explore the effect of different weights of the cell boundary. See Section 4.6 for details.

Semantic Segmentation Architecture. The network structure of the semantic segmentation branch is shown in Figure 2c. The layer-level features of FPN are downsampled to the same size after 1×1 convolution and element-wise summation. The features are further extracted by four 3×3 convolutional layers and then divided into two branches. One branch calculates the cross-entropy loss of predictions and semantic segmentation labels. The other branch is added to the subsequent detection and segmentation branches.

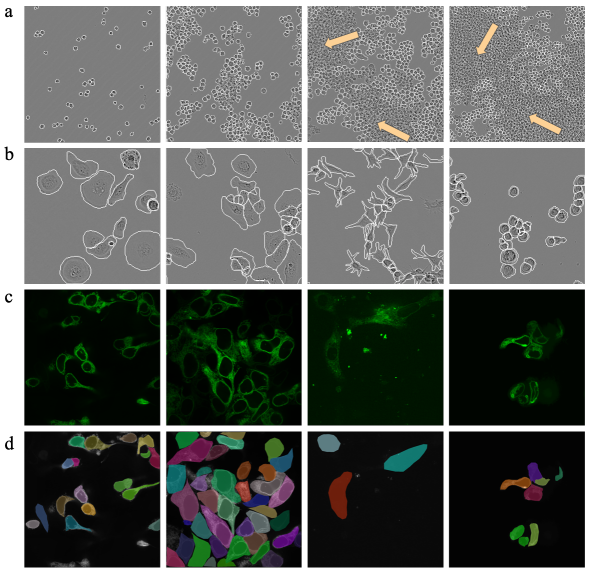


Figure 4: Representative samples from the LIVECell and HEK293T datasets. (a) Examples of images with different densities in LIVECell datasets. Regions indicated by light brown arrows are not labeled because of high cell density. (b) Examples of different cell types in the LIVECell dataset. From left to right: SKOV3, Huh7, SHSY5Y, BT474. The white outlines in (a) and (b) are drawn for visualization. (c) Examples of HEK293T dataset. (d) Annotations corresponding to the images of the HEK293T dataset. The corresponding annotations and images show that the HEK293T dataset is not uniform in image brightness. There are many regions with weak signals.

4. Experiments

4.1. Datasets

LIVECell is a large-scale public dataset that consists of 5239 phase-contrast microscopy images with a total of 1,686,352

cell instances from eight different cell types [11]. For consistency in performance comparison, we partition the dataset into 3188 training images, 539 validation images, and 1512 test images, same as in [11]. Images in LIVECell have two notable features. First, they show large variations in cell density, with images of cells growing from the initial seeding phase to a fully confluent monolayer (Figure 4a). In the case of full confluency, a LIVECell image 704×520 in size can contain more than 3000 instances, which makes it difficult even for human eyes to accurately identify cell boundaries. Second, cells in the LIVECell dataset show a wide variety in size and shape(Figure 4b), including small and round BV-2 cells, large and flat SK-OV-3 cells, and elongated SH-SY5Y cells. The high density and wide variety of cells pose substantial challenges for the design of algorithms.

HEK293T is a small-scale in-house dataset of HEK293T cells imaged by confocal microscopy. It is partitioned into 108 training images and 37 test images, with a total of 2012 training instances and 576 test instances. It is characterized by uneven cell brightness and the presence of some weak signal regions(Figure 4c-d). We first augment it into 3240 training images by rotation, flipping, blurring, and intensity transformation before training. During training, the dataset is further augmented online using the space-filling strategy.

4.2. Performance Evaluation Metrics

We use standard COCO evaluation metrics [22] AP (Average Precision), which is averaged over Intersection over Union(IoU) thresholds from 0.5 to 0.95 at an interval of 0.05 [22]. Specifically, $\mathrm{AP}^{\mathrm{bbox}}$ denotes the AP of cell detection, $\mathrm{AP}^{\mathrm{segm}}$ denotes the AP of cell instance segmentation, $\mathrm{AP}_{0.5}$ and $\mathrm{AP}_{0.75}$ denote AP at IoU thresholds of 0.5 and 0.75, respectively.

4.3. Implementation Details

We implement our network model in PyTorch, using a Swin-Tiny model pre-trained on ImageNet-22K as the backbone network. Our model is trained with the AdamW optimizer with a learning rate of 0.0001 and a weight decay of 0.05. For the semantic segmentation branch, we use the cross-entropy loss function between the prediction and the ground truth and set the weights for the background, the cell interior, and the cell boundary to be 1, 1, and 3, respectively, in the loss function.

For the LIVECell dataset, we follow the same settings as in [11]. During training, we use a batch size of 16 (distributed to two per GPU), and set the short sides of images to (440, 480, 520, 580, 620) so that the network randomly selects the short side from the sequence and changes the long side by the same ratio. Given the large amount of annotated data, space-filling augmentation is not performed on this dataset. Each model is trained for about 140 epochs.

For the HEK293T dataset, we keep the image resolution of 1200×1200 and expand the dataset online using space-filling data augmentation. We train our network for 36 epochs with a batch size of 32.

4.4. Experimental Results on HEK293T Dataset

Table 1 summarizes the performance comparisons of the proposed model against widely used instance segmentation models, including Mask R-CNN, Cascade Mask R-CNN, PointRend [18], MViTv2 [20] and Hybrid Task Cascade, these results are reproduced based on MMDetection [7]. Without cascading, our approach improves by 2.0% on $\mathrm{AP}^{\mathrm{bbox}}$ and 1.6% on $\mathrm{AP}^{\mathrm{segm}}$ over the best-performing MViTv2 model. Our method has the most significant performance improvement in $\mathrm{AP}^{\mathrm{segm}}$ at an IoU of 0.75, which improves 10.4% in $\mathrm{AP}^{\mathrm{segm}}$ compared to Mask R-CNN with ResNet-50 backbone, confirming that our model generates high-quality segmentation results. When using the cascade strategy, our approach improves by 3.1% on $\mathrm{AP}^{\mathrm{bbox}}$ and 3.6% on $\mathrm{AP}^{\mathrm{segm}}$ over the best-performing Cascade Mask R-CNN using ResNeST [41]-50 backbone.

Figure 5a shows the qualitative comparison results of our model against the competing models. Mask R-CNN, PointRend and MViTv2 tend to produce missing detection and incomplete segmentation results in regions with weak signals due to the absence of semantic information. HTC fuses the semantic branch of background and foreground so that the distinction of foreground and background semantics alleviates the problem of incomplete segmentation in regions with weak signals. However, this strategy also causes the model to merge cells in tight contact into one instance prediction. Because our model makes full use of contextual semantic information and boundary information, it performs well in regions of weak signals and tight cell contacts.

4.5. Experimental Results on LIVECell Dataset

Table 2 summarizes the performance comparisons on the LIVECell dataset of our model against the same set of models tested on the HEK293T dataset. Segmentation results using Cascade Mask R-CNN are based on the ResNeSt-200-DCN [9] backbone as in [11], while the other models are reproduced based on detectron2 [37]. Without cascading, our approach achieves a gain of 1.1% in $\mathrm{AP}^{\mathrm{segm}}$ over the best-performing MViTv2 model. When using the cascade strategy, our model achieves approximately the same $\mathrm{AP}^{\mathrm{bbox}}$ and a gain of 0.8% in $\mathrm{AP}^{\mathrm{segm}}$ over the best-performing Cascade Mask R-CNN with ResNeSt-200-DCN backbone but uses fewer parameters (29M vs. 70M) and lower FLOPs (4.5G vs. 17.5G).

Figure 5b shows qualitative comparison results of our model against the competing models. Results in rows 1 and 2 show that for elongated cells, Mask R-CNN, HTC, and PointRend have difficulty capturing long-distance dependency, while our model and Transformer-based MViTv2

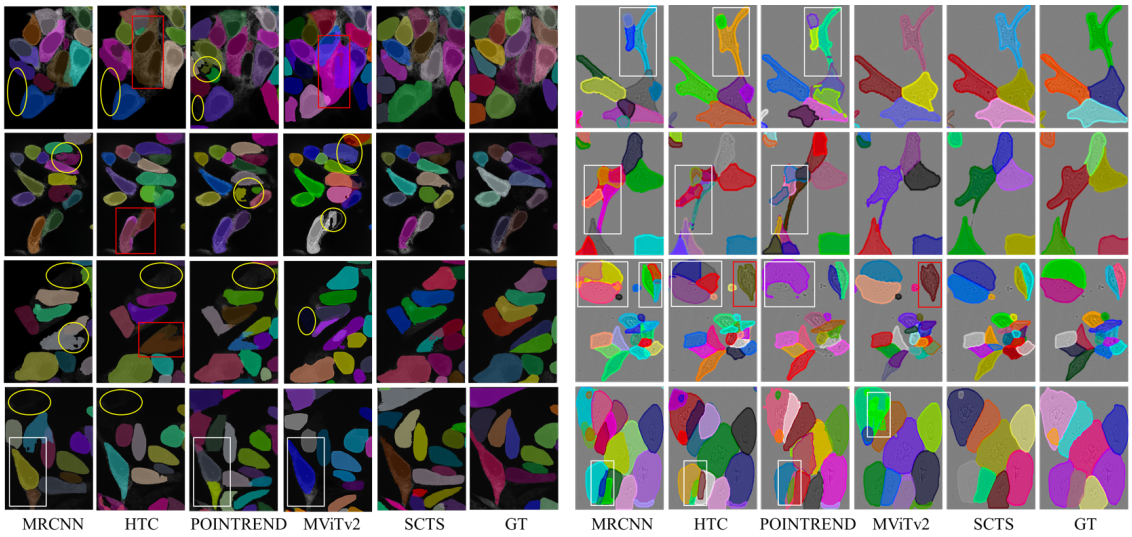


Figure 5: Qualitative results on HEK293T and LIVECell dataset. (a) Qualitative results of our method against the competing models on the HEK293T dataset. (b) Qualitative results of our method against the competing models on the LIVECell dataset. MRCNN: Mask R-CNN, GT: GroundTruth. Ovals highlight predictions in regions with weak signals. White rectangular boxes highlight boundary predictions under tight contacts and diffusive boundaries of cells. Red rectangular boxes highlight predictions in regions that merge cells in tight contact into one instance.

Method	Backbone	AP^{bbox}	$AP^{bbox}_{0.5}$	$AP^{bbox}_{0.75}$	AP^{segm}	$AP^{segm}_{0.5}$	$AP^{segm}_{0.75}$
Mask R-CNN	ResNet-50	44.7	79.5	47.1	45.3	81.0	47.2
Mask R-CNN	ResNeST [41]-50	46.2	80.5	48.0	46.5	81.2	49.4
Mask R-CNN	Swin-Tiny	47.1	81.9	51.0	48.3	84.2	51.0
PointRend [18]	ResNet-50	46.7	81.1	48.7	46.6	82.1	50.5
MViTv2 [20]	MViTv2-Tiny	49.0	84.3	51.3	50.0	85.2	54.0
SCTS	Swin-Tiny	**51.0**	**86.2**	**53.4**	**51.6**	**85.9**	**57.6**
Cascade Mask R-CNN	ResNet-50	46.9	78.5	50.2	47.0	80.5	51.4
Cascade Mask R-CNN	ResNeST-50	48.4	80.8	52.1	48.7	83.5	51.3
Hybrid Task Cascade	ResNet-50	48.3	82.0	40.9	48.5	83.1	52.7
Cascade SCTS	Swin-Tiny	**51.5**	**85.3**	**54.1**	**52.3**	**86.1**	**57.4**

Table 1: Performance comparison of different instance segmentation models on HEK293T dataset.

model make accurate predictions. This suggests that the Swin Transformer backbone helps our model to adapt to different cell shapes and sizes. It can be seen from rows 3 and 4 that our model gives better predictions on cells with tight contact cells than the competing models. This suggests that the prediction of boundary semantic labels helps the model differentiate cells in tight contact.

4.6. Ablation Study

Effects of Network Component. First, we evaluate the contributions of individual network components and the combination on the HEK293T dataset. The results are summarized in Table 3, Each component brings performance gain, and the combination of components performs the best precision. In Table 4, we also evaluate the contributions of components on the LIVECell dataset, it obtains similar improvements as HEK293T dataset. This demonstrates the effectiveness of the individual components and their combinations.

Effects of Different Semantic Segmentation Cross-entropy Loss Weights. For weighted cross-entropy loss, we examined the influence of different semantic class weights on model performance. Based on SCTS without cascade, we set weights for the background and the cell interior loss both at 1 and investigate the performance of cell boundary loss weight from 1 to 5, respectively (Figure 6a). The best performance of both AP^{bbox} and AP^{segm} is achieved when the weight of boundary loss at 3. We performed the same experiment on LIVECell validation set(Table 5), and it ob-

Method	Backbone	AP^{bbox}	AP^{segm}
Mask R-CNN	ResNet-50	42.9	44.8
Mask R-CNN	ResNeST-50	43.9	45.5
Mask R-CNN	Swin-Tiny	45.0	46.2
Pointrend	ResNet-50	44.4	44.7
MViTv2	MViTv2-Tiny	45.5	46.5
SCTS	Swin-Tiny	**45.7**	**47.6**
Cascade Mask R-CNN	ResNet-50	45.4	45.6
Cascade Mask R-CNN	ResNeST-50	47.1	46.8
Hybrid Task Cascade	ResNet-50	45.8	45.8
Cascade Mask R-CNN	ResNeST-200-DCN [9]	48.5	47.9
Cascade SCTS	Swin-Tiny	**48.6**	**48.7**

Table 2: Performance comparison of different instance segmentation models on the LIVECell test set.

Swin-T	Semantic	S-F	Cascade	AP^{bbox}	AP^{segm}
-	-	-	-	45.7	46.4
✓	-	-	-	48.3	49.6
-	✓	-	-	47.2	47.7
-	-	✓	-	48.2	48.8
-	-	-	✓	47.3	47.5
✓	✓	-	-	49.2	50.6
✓	✓	✓	-	51.0	51.6
✓	✓	✓	✓	**51.5**	**52.3**

Table 3: Ablation study of effects of network components on the HEK293T dataset. Swin-T: Swin-Tiny backbone network, Semantic: semantic branch, S-F: space-filling augmentation. Cascade: network cascade architecture.

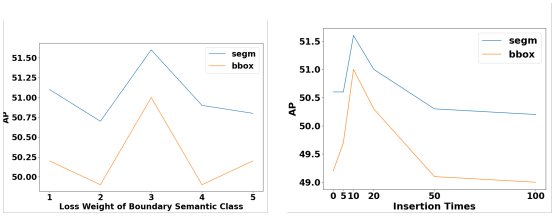
Swin-T	Semantic	Cascade	AP^{bbox}	AP^{segm}
-	-	-	43.1	45.1
✓	-	-	44.7	46.7
✓	✓	-	45.7	47.6
✓	✓	✓	**48.6**	**48.7**

Table 4: Ablation study of effects of network components on the LIVECell test set.

tains similar results as HEK293T dataset. So we set the loss weight of the boundary classes to three times the weight of the other two classes.

W	AP^{bbox}	AP^{segm}
1	46.7	48.3
2	46.7	48.4
3	**46.8**	**48.4**
4	46.7	48.2
5	46.7	48.2

Table 5: Effects of boundary semantic class loss weight on the LIVECell validation set.W:weight of boundary semantic.



(a) (b)

Figure 6: Ablation study on semantic segmentation cross-entropy loss weight and different numbers of instance insertions on the HEK293T dataset. (a) Effects of semantic segmentation loss weight. (b) Effects of different numbers of instance insertions.

Effects of Different Numbers of Instance Insertions Using the HEK293T dataset, we explore the effect of different numbers of instances inserted using the space-filling strategy. We use semantic branching without network cascades and experiment with 0, 5, 10, 20, 50, and 100 insertions (Figure 6b). Initially, the performance gradually improves as the number of insertions increases, and then the performance degrades, likely because the gap between the distribution of the training set and the test set widens under the excessive number of insertions. The best performance of detection and segmentation is achieved when the number of insertions is 10. In general, we find that the number of insertions should be set based on the distribution of the training and test sets for practical applications. Specifically for the HEK293T dataset, the training and test sets are similarly distributed, only a small number of insertions are needed.

5. Conclusion

In this study, we have developed SCTS, a novel model for instance segmentation of single cells using a Swin Transformer backbone and three-class semantic feature embedding. We have also developed a new augmentation strategy named space-filling to improve the diversity of training images. Experiments show that our model outperforms several state-of-the-art models on the LIVECell and our in-house datasets.Our model also has its limitations,and we addresses these in discussion section in the supplementary materials. Single-cell segmentation in the case of extremely high cell densities is one of the directions we will pursue in future studies.

6. Acknowledgments

The work was supported in part by the National Natural Science Foundation of China (grant 31971289 and 91954201 to G.Y.) and the Strategic Priority Research Program of the Chinese Academy of Sciences (grant XDB37040402 to G.Y.).

References

[1] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5221–5229, 2017.

[2] Feng Bao, Yue Deng, Sen Wan, Bo Wang, Qionghai Dai, Steven J Altschuler, and Lani F Wu. Characterizing tissue composition through combined analysis of single-cell morphologies and transcriptional states. *Nature Biotechnology*, 2022.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.

[4] Juan C Caicedo, Jonathan Roth, Allen Goodman, Tim Becker, Kyle W Karhohs, Matthieu Broisin, Csaba Molnar, Claire McQuin, Shantanu Singh, Fabian J Theis, et al. Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. *Cytometry Part A*, 95(9):952–965, 2019.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[6] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019.

[7] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *ArXiv Preprint ArXiv:1906.07155*, 2019.

[8] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2018.

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 764–773, 2017.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations, ICLR*, 2021.

[11] Christoffer Edlund, Timothy R Jackson, Nabeel Khalid, Nicola Bevan, Timothy Dale, Andreas Dengel, Sheraz Ahmed, Johan Trygg, and Rickard Sjögren. Livecell—a large-scale dataset for label-free live cell segmentation. *Nature Methods*, 18(9):1038–1045, 2021.

[12] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[14] Noah F Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J McIntosh, Ke Xuan Leow, Morgan Sarah Schwartz, et al. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature Biotechnology*, 40(4):555–565, 2022.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6409–6418, 2019.

[18] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9799–9808, 2020.

[19] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2359–2367, 2017.

[20] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022.

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer:

Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[26] Sophia K Longo, Margaret G Guo, Andrew L Ji, and Paul A Khavari. Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics. *Nature Reviews Genetics*, 22(10):627–644, 2021.

[27] Soham Mandal and Virginie Uhlmann. Splinedist: Automated cell segmentation with spline curves. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1082–1086. IEEE, 2021.

[28] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[30] Jos BTM Roerdink and Arnold Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41(1-2):187–228, 2000.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015.

[32] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 265–273. Springer, 2018.

[33] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, 2021.

[34] Tim Stuart and Rahul Satija. Integrative single-cell analysis. *Nature Reviews Genetics*, 20(5):257–272, 2019.

[35] Florin C Walter, Sebastian Damrich, and Fred A Hamprecht. Multistar: instance segmentation of overlapping objects with star-convex polygons. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 295–298. IEEE, 2021.

[36] Thomio Watanabe and Denis Wolf. Distance to center of mass encoding for instance segmentation. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3825–3831. IEEE, 2018.

[37] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[38] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12193–12202, 2020.

[39] Enze Xie, Wenhai Wang, Mingyu Ding, Ruimao Zhang, and Ping Luo. Polarmask++: Enhanced polar representation for single-shot instance segmentation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[40] Xuanang Xu, Thomas Sanford, Baris Turkbey, Sheng Xu, Bradford J Wood, and Pingkun Yan. Polar transform network for prostate ultrasound segmentation with uncertainty estimation. *Medical Image Analysis*, 78:102418, 2022.

[41] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2736–2746, 2022.