Supplementary Material

1. Discussion

We discuss here some of the limitations of the model. First, due to the large downsampling factor used for semantic segmentation, some small objects will disappear on the semantic feature map, which will affect the detection performance of small objects. Several examples are shown in Figure 1, ovals highlight some poor segmentation cases with small objects, and the area of these cases is mostly smaller than 18×18 pixels. Another limitation is that at extremely high cell densities, it is still difficult for our model to accurately distinguish cell boundaries by embedding three classes of semantic features, Several examples are shown in Figure 2, ovals highlight some poor segmentation cases at extremely high cell densities. An interesting future direction is to close this gap by embedding distance-based or gradient-based instance representations, which we will investigate in follow-up research.

2. Implementation Details

Our method is extended based on two open source code libraries, MMdetection and Detectron2. The experiments on the HEK293T dataset are based on MMdetection, and the experiments on the LIVECell dataset are based on Detectron2. Table 1 show the detailed parameter settings for experiments on HEK293T dataset. For HEK293T dataset, the model using SGD as optimizer required only 24 training epochs, while the model using AdamW as optimizer required 36 training epochs. Table 2 show the detailed parameter settings for experiments on LIVECell dataset.

3. More Visualization Results

Figure 3 shows some examples of space-filling augmentation processing results. Figure 4 shows qualitative results of our method on different cell types in the LIVECell dataset.

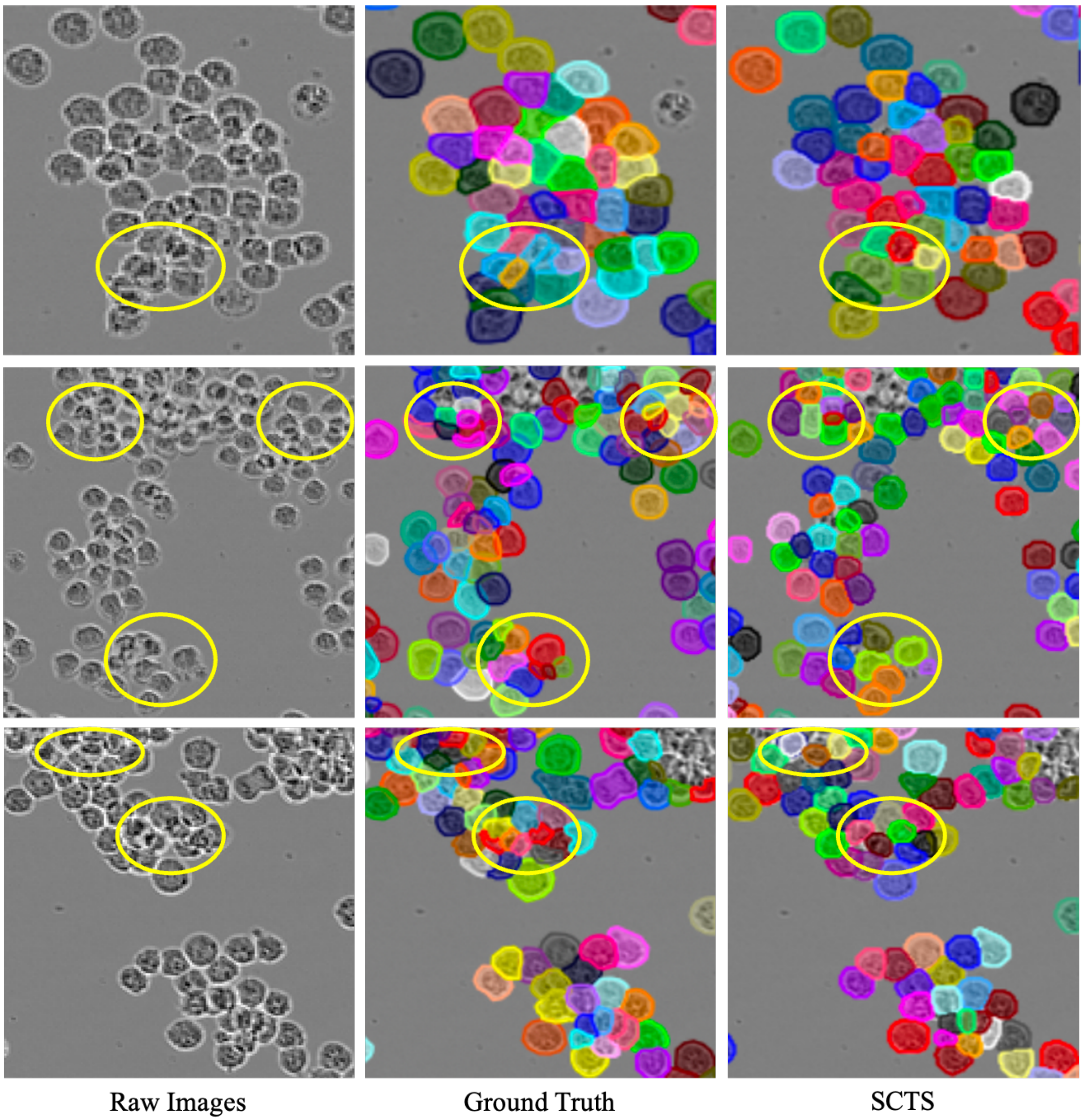

Figure 1: Examples of poor segmentation of small objects. Oval highlights specific cases.

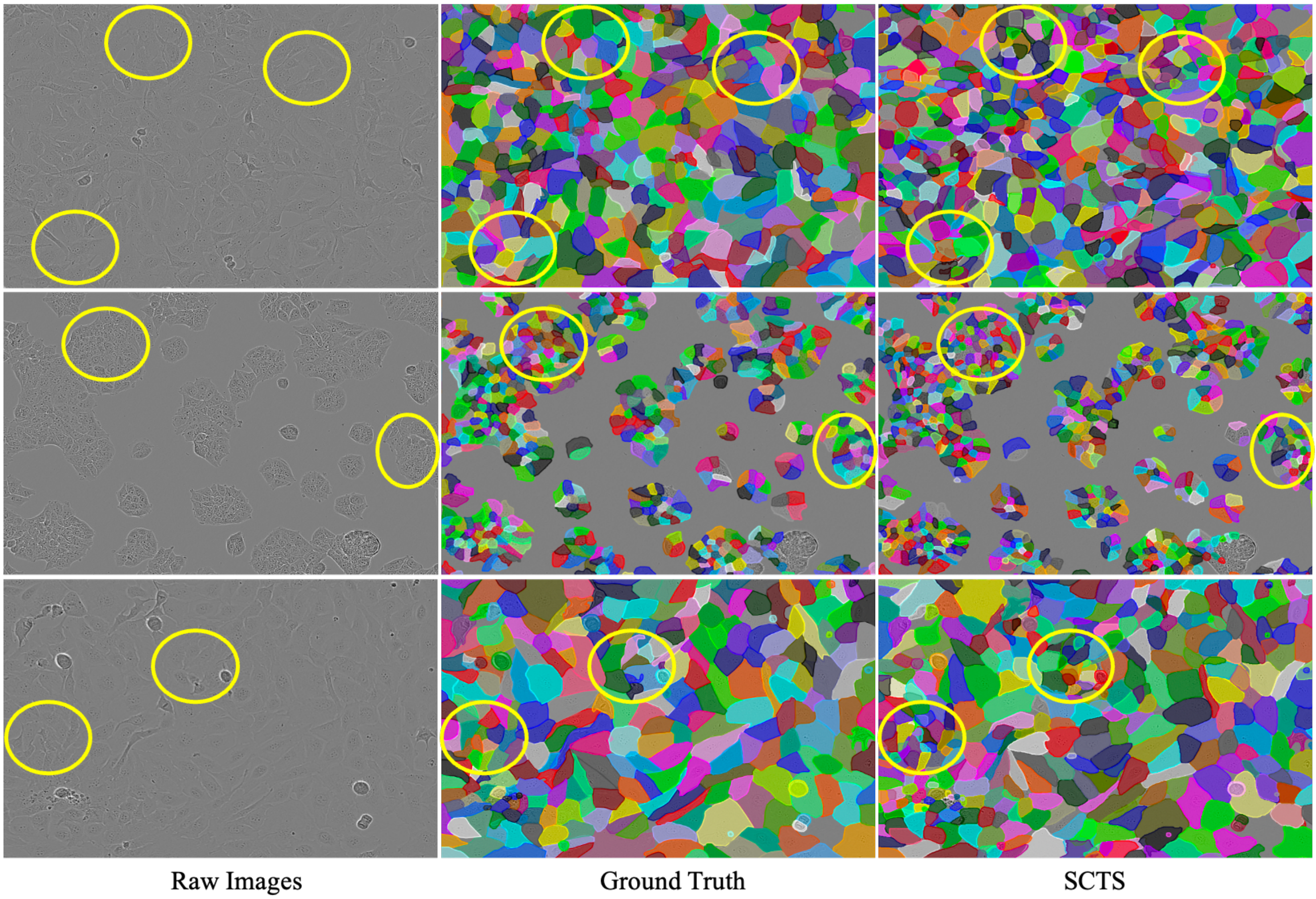

Figure 2: Examples of poor segmentation at extremely high cell densities. Oval highlights specific cases.

Method	Backbone	Pretrain	Multi Scale	Batch Size	Epochs	Optimizer	LR
Mask R-CNN	ResNet50	ImageNet	False	4 images per gpu	24	SGD	0.002
Mask R-CNN	ResNeST-50	ImageNet	False	4 images per gpu	24	SGD	0.002
Mask R-CNN	Swin-Tiny	ImageNet	False	4 images per gpu	36	AdamW	0.0001
PointRend	ResNet-50	ImageNet	False	4 images per gpu	24	SGD	0.002
MViTv2	MViTv2-Tiny	ImageNet	False	4 images per gpu	36	AdamW	0.0001
SCTS	Swin-Tiny	ImageNet	False	4 images per gpu	36	AdamW	0.0001
Cascade Mask R-CNN	ResNet-50	ImageNet	False	4 images per gpu	24	SGD	0.002
Cascade Mask R-CNN	ResNeST-50	ImageNet	False	4 images per gpu	24	SGD	0.002
Hybrid Task Cascade	ResNet-50	ImageNet	False	4 images per gpu	24	SGD	0.002
Cascade SCTS	Swin-Tiny	ImageNet	False	4 images per gpu	36	AdamW	0.0001

Table 1: Detailed parameter settings for experiments on the HEK293T dataset.

Method	Backbone	Pretrain	Multi Scale	Batch Size	Epochs	Optimizer	LR
Mask R-CNN	ResNet-50	ImageNet	True	2 images per gpu	140	SGD	0.002
Mask R-CNN	ResNeST-50	ImageNet	True	2 images per gpu	140	SGD	0.002
Mask R-CNN	Swin-Tiny	ImageNet	True	2 images per gpu	140	AdamW	0.0001
PointRend	ResNet-50	ImageNet	True	2 images per gpu	140	SGD	0.002
MViTv2	MViTv2-Tiny	ImageNet	True	2 images per gpu	140	AdamW	0.0001
SCTS	Swin-Tiny	ImageNet	True	2 images per gpu	140	AdamW	0.0001
Cascade Mask R-CNN	ResNet-50	ImageNet	True	2 images per gpu	140	SGD	0.002
Cascade Mask R-CNN	ResNeST-50	ImageNet	True	2 images per gpu	140	SGD	0.002
Cascade Mask R-CNN	ResNeST-200-DCN	ImageNet	True	2 images per gpu	140	SGD	0.002
Hybrid Task Cascade	ResNet-50	ImageNet	True	2 images per gpu	140	SGD	0.002
Cascade SCTS	Swin-Tiny	ImageNet	True	2 images per gpu	140	AdamW	0.0001

Table 2: Detailed parameter settings for experiments on the LIVECell dataset.

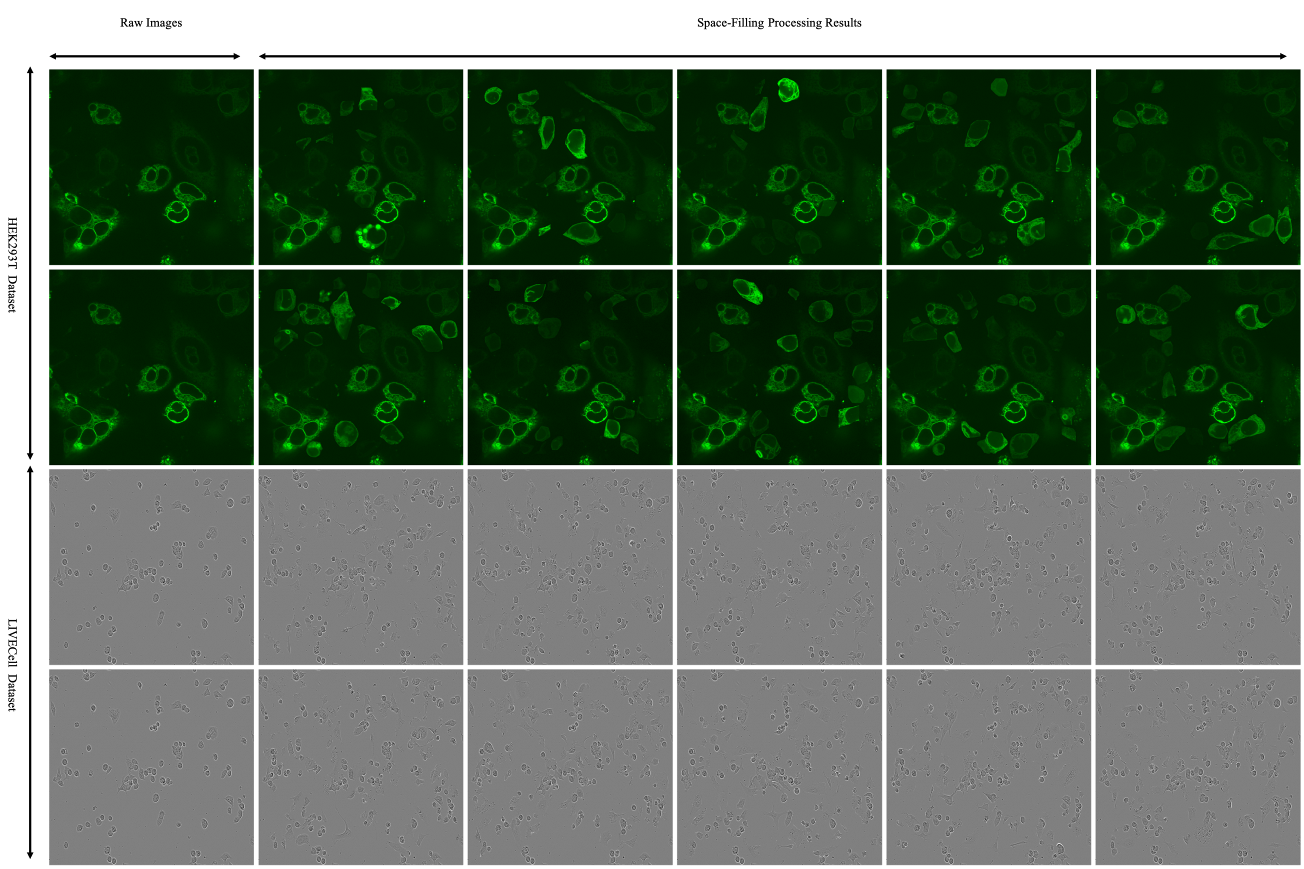

Figure 3: Example of space-filling augmentation processing results.

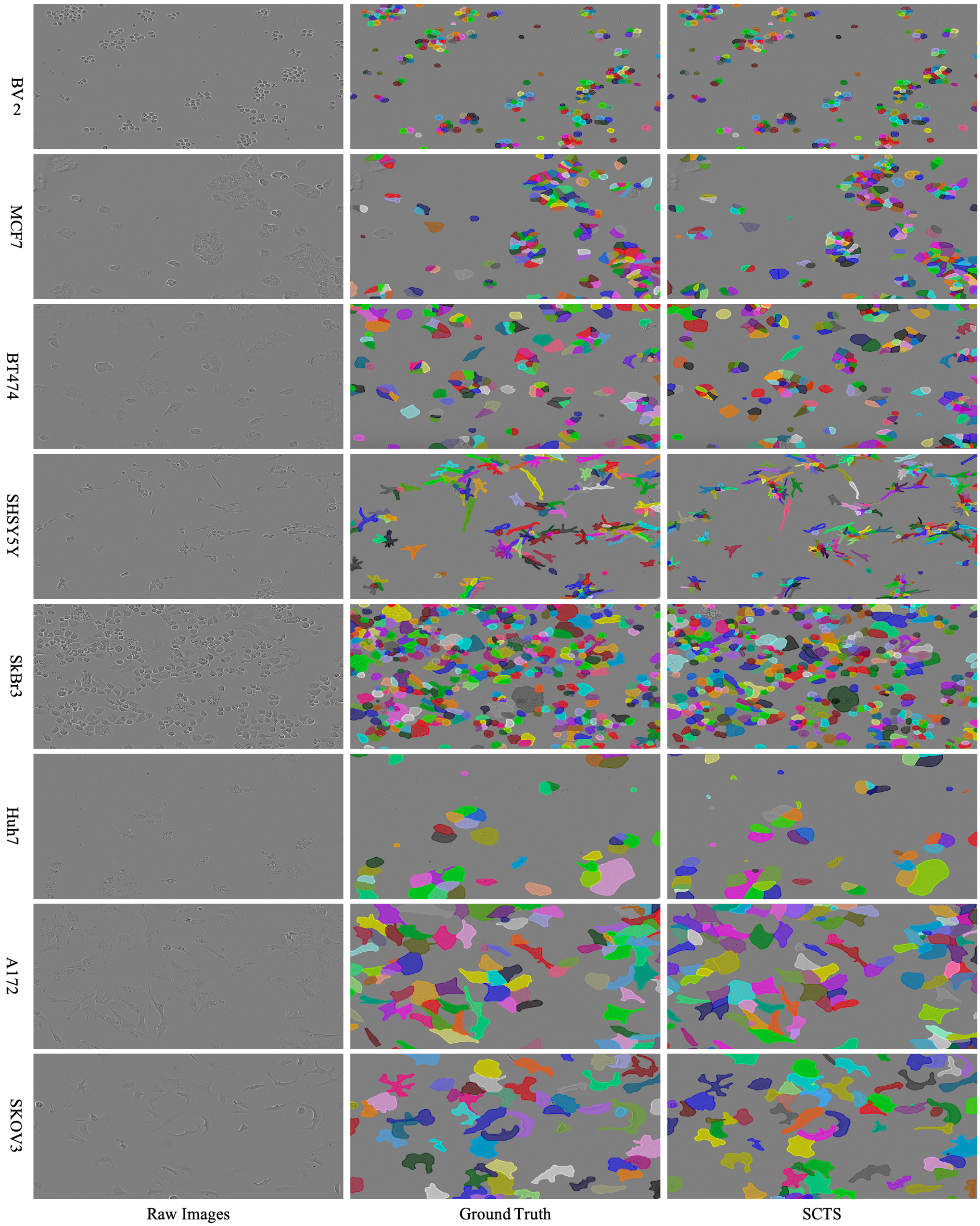

Figure 4: Qualitative results of our method on different cell types in the LIVECell dataset.