Homework 3

Ping-Yeh Chou 113550901

1 Introduction

Instance segmentation of medical images is a challenging task due to the complex morphology and subtle boundaries of different cell types. In this assignment, the goal is to accurately segment four categories of cells from colored microscopy images, providing precise masks for each instance. To address this, I adopt Mask R-CNN [He *et al.*, 2017] as the baseline framework, which is well-suited for instance segmentation tasks. To further enhance the model's ability to capture discriminative features and improve segmentation accuracy, I integrate a Feature Pyramid Network (FPN) [Lin *et al.*, 2017] backbone to better handle multi-scale information, and incorporate Convolutional Block Attention Module (CBAM) [Woo *et al.*, 2018] into the backbone. The CBAM attention mechanism adaptively refines feature representations by focusing on informative regions both channel-wise and spatially, which is particularly beneficial for distinguishing overlapping or visually similar cells. This combination aims to boost the model's sensitivity to relevant structures in medical images, ultimately leading to more accurate and robust cell segmentation results. My code is available at .

2 Method

2.1 Data Pre-processing

To ensure robust model training and improve generalization, a comprehensive data pre-processing pipeline was implemented. Each image and its corresponding instance masks were loaded and converted to RGB format. For the training set, a series of data augmentation techniques were applied to increase data diversity and simulate various real-world conditions. These augmentations included random horizontal and vertical flips, random rotations, zoom, and resized cropping, all of which were applied jointly to both images and masks to maintain alignment. Additionally, color jitter, Gaussian noise, Gaussian blur, and random gamma adjustments were used to further enhance variability in appearance. All images were normalized using the dataset mean and standard deviation. The validation and test sets underwent only tensor conversion and normalization to ensure fair evaluation. The Table 1 below summarizes the augmentations and their parameters used during training.

2.2 Model Architecture

The model (176.29 MB) for this instance segmentation task is built upon the Mask R-CNN framework, which is well-established for its effectiveness in detecting and segmenting individual objects within an image. The backbone of the model is a ResNet-50 network [He *et al.*, 2016], chosen for its strong feature extraction capabilities and proven performance in a wide range of vision tasks. To further enhance the backbone, I integrate the Convolutional Block Attention Module (CBAM) after each major stage of the ResNet. CBAM consists of both channel and spatial attention mechanisms, allowing the network to adaptively emphasize informative features and suppress less useful ones. This is particularly beneficial for medical images, where subtle differences and local details are crucial for accurate segmentation.

To address the challenge of detecting cells at different scales, a Feature Pyramid Network (FPN) is used as the neck of the architecture. FPN combines feature maps from multiple layers of the backbone, enabling the model to leverage both high-level semantic information and fine-grained details. This multi-scale representation is essential for robustly segmenting cells of varying sizes and shapes.

The head of the model follows the standard Mask R-CNN design, which includes parallel branches for bounding box regression, classification, and mask prediction. The mask head is a small fully convolutional network that predicts a binary mask for each detected instance. Both the box and mask heads are adapted to the number of cell classes in the dataset.

This architecture is selected because it combines the strengths of Mask R-CNN's flexible instance segmentation pipeline, the deep and expressive ResNet backbone, the multi-scale capabilities of FPN, and the adaptive feature refinement provided by CBAM. The main advantage of this design is its ability to capture both global context and local details, which is critical for accurate cell segmentation. The use of CBAM further improves the model's focus on relevant regions, potentially leading to better performance in challenging cases such as overlapping or low-contrast cells. The primary drawback is the increased computational complexity and memory usage due to the attention modules and multi-scale processing, which requires more resources and careful tuning during training. However, the expected gains in segmentation accuracy make this trade-off worthwhile for the task at hand.

2.3 Training

The training process follows a standard supervised learning pipeline for instance segmentation. The model is trained using the AdamW optimizer [Loshchilov and Hutter, 2017] with weight decay to help regularize the network and prevent overfitting. A cosine annealing learning rate scheduler [Loshchilov and Hutter, 2016] is employed to gradually reduce the learning rate from its initial value to a lower bound over the course of training epochs, which helps the model converge more smoothly. Mixed precision training [Micikevicius *et al.*, 2017] is enabled to accelerate computation and reduce memory usage. During each epoch, the model alternates between training on mini-batches of images and evaluating on the validation set. For each batch, images and their corresponding targets are loaded, augmented, and transferred to the GPU. The model computes the loss, which is then backpropagated to update the parameters. After each epoch, the model's performance is evaluated on the validation set using an IoU-based metric, and a checkpoint is saved. The Table 2 below summarizes the main hyperparameters used during training.

3 Result

On the unknown test set, the model achieved a mean Average Precision (mAP) of 0.3336, indicating that it performs well in accurately segmenting and distinguishing different cell types even under challenging, unseen conditions, as displayed in Figure 1. This strong performance demonstrates the model's ability to generalize beyond the training data, effectively handling the variability and complexity present in real-world samples.

The training curve clearly shows a steady decline in training loss and a corresponding improvement in validation accuracy as the epochs progress, which suggests that the model is learning effectively without significant overfitting, as shown in Figure 2. The gap between training and validation metrics remains small throughout, further supporting the model's good generalization. Overall, both the quantitative results and loss curve visualizations confirm the effectiveness and robustness of my approach, providing strong evidence that the chosen architecture and training strategy are well-suited for this instance segmentation task.

113550901	1	2025-04-23 10:17	273846	113550901	0.3336

Figure 1: Results on test datasets.

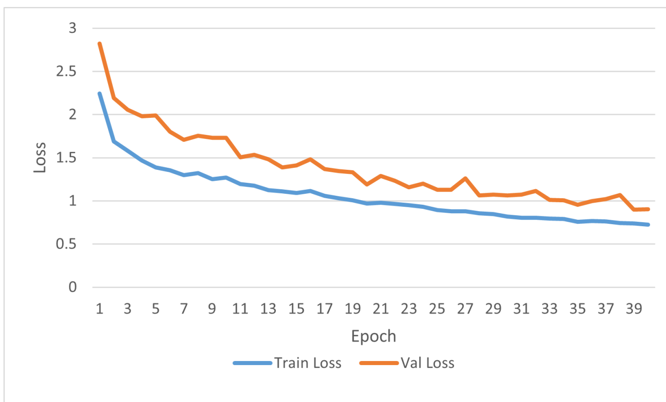

Figure 2: Convergence of the training and validation loss curve.

4 Additional Experiments

To further explore ways to improve the instance segmentation performance, I conducted ablation experiments focusing on the backbone architecture. My hypothesis was that integrating more advanced feature extraction modules, such as Feature Pyramid Networks (FPN) and Convolutional Block Attention Module (CBAM), could help the model better capture multi-scale and salient features, thus improving detection and segmentation accuracy. The rationale is that FPN enhances the model's ability to utilize features at different spatial resolutions, which is particularly beneficial for detecting objects of varying sizes, while CBAM introduces channel and spatial attention mechanisms that may help the network focus on more informative regions. To ensure fairness, all other training settings and hyperparameters were kept identical across experiments. The results are summarized in Table 3: using only ResNet-50 as the backbone yielded a test mAP of 0.2421; adding FPN increased the mAP to 0.2815; and further incorporating CBAM on top of FPN led to a significant improvement, achieving a test mAP of 0.3336. These results suggest that both FPN and CBAM contribute positively to the model's performance, with the combination providing the best results in this setting.

Table 1: Summary of data augmentations and parameters used during training.

Augmentation	Parameter	Value	Description
Random Horizontal Flip	Probability	0.5	Flip image and mask horizontally
Random Vertical Flip	Probability	0.3	Flip image and mask vertically
Random Rotation	Degrees Probability	15 0.4	Random rotation within $\pm 15°$
Random Zoom	Scale Probability	(0.85, 1.15) 0.3	Random zoom in/out
Random Resized Crop	Scale Ratio Probability	(0.8, 1.0) (0.9, 1.1) 0.3	Crop area ratio Aspect ratio range
Color Jitter	Brightness Contrast Saturation Hue Probability	0.3 0.3 0.3 0.15 0.7	Adjust brightness Adjust contrast Adjust saturation Adjust hue
Gaussian Noise	Std Probability	0.02 0.2	Additive Gaussian noise
Gaussian Blur	Kernel Size Sigma Probability	3 (0.1, 1.0) 0.2	Blur kernel size Standard deviation range
Random Gamma	Gamma Range Probability	(0.8, 1.2) 0.3	Random gamma correction
Normalize	Mean/Std	Per channel	Dataset statistics

Table 2: Training hyperparameters for the Mask R-CNN instance segmentation model.

Hyperparameter	Value	Description
Optimizer	AdamW	With weight decay
Initial Learning Rate	2×10^{-4}	Starting learning rate
Min Learning Rate	5×10^{-6}	Lower bound for cosine annealing
Weight Decay	1×10^{-4}	L2 regularization
Scheduler	CosineAnnealingLR	Learning rate scheduling
Batch Size	2	Images per training step
Epochs	40	Total training epochs
Mixed Precision	Enabled	Use of torch.amp for acceleration
Device	CUDA/CPU	Training hardware
Score Threshold	0.5	Filter predictions by confidence
NMS Threshold	0.5	Non-maximum suppression IoU threshold
Checkpoint Path	./results	Directory for saving models

Table 3: Ablation study on backbone modules for instance segmentation.

Backbone Configuration	Test mAP
ResNet-50	0.2421
ResNet-50 + FPN	0.2815
ResNet-50 + FPN + CBAM	0.3336

References

[He *et al.*, 2016] 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.

[He *et al.*, 2017] 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.

[Lin *et al.*, 2017] 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.

[Loshchilov and Hutter, 2016] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[Micikevicius *et al.*, 2017] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[Woo *et al.*, 2018] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.