Classification of Uncertainty in Nuclei Segmentation From H&E Images

Conor Messer Rohit Khurana Suhana Bedi
Department of Biomedical Data Science

Abstract

Nuclear segmentation in medicine is vital for guiding a pathologist's diagnostic interpretation of whole-slide images (WSI). Hematoxylin and eosin (H&E) staining is commonly used to highlight tissue morphology, with nuclei appearing blue or purple. Accurate detection of nuclei is crucial for procedures like cancer detection and grading. However, manual identification is labor-intensive and error-prone, and automated detection is challenging due to the diverse appearances and overlapping nature of nuclei, as well as artifacts from digital imaging. To address these challenges, we implemented, trained, and evaluated three image segmentation models: U-Net, Mask R-CNN, and YOLO. Our models were trained to output discrete confidence levels for each nucleus, simulating expert pathologist annotations. Additionally, for Mask R-CNN, we labeled segmented nuclei as certain or uncertain to capture ambiguity, incorporating label maps of these regions during training. We used metrics such as DICE score, Aggregated Jaccard Index (AJI), and Average Precision (AP50 and APm) to ensure a fair comparison. Our results show that Mask-RCNN outperformed U-Net and YOLO in both segmentation quality and instance segmentation metrics, achieving a DICE score of 0.768 on the test set. Future research will focus on utilizing ambiguous masks, confidence-based segmentation, and other advanced techniques to enhance the robustness and accuracy of automated nuclei detection in pathology images

1. Introduction

In medicine, nuclear segmentation plays a critical role in guiding a pathologist's diagnostic interpretation of a whole-slide image (WSI). Hematoxylin and eosin (H&E) staining is a widely used technique for better illuminating tissue morphology; in these images, nuclei often appear blue or purple, allowing a pathologist to easily assess their shape, size, and spatial organization. These qualitative measurements directly facilitate hospital procedures such as cancer detection and grading, underscoring the importance in ensuring accurate detection.

Given the large number of nuclei in a single tissue section as well as the number of images a pathologist must examine daily, manually identifying nuclei is time consuming and, quite importantly, prone to error. Automated identification is equally challenging, as nuclei have varying appearances and can overlap. The generation of digital pathology images can additionally introduce several batch artifacts, such as folded tissue.

Therefore, robust computational approaches are needed to fully automate nuclei detection from H&E images at scale. We propose to implement, train, and evaluate the performance of three image segmentation models: U-Net, Mask R-CNN, and YOLO. Each model - when trained - takes in an image and outputs a segmentation mask, with U-Net performing basic semantic segmentation and the other models performing instance segmentation. To further mimic expert pathologist annotators, our models will additionally be trained to output discrete confidence levels for each nuclei. In order to capture and emulate human understanding of pathology images in regards to ambiguous annotation - i.e., regions where there is no precise consensus on where the nuclei are, even for human experts - we additionally label segmented nuclei as either uncertain or certain. To achieve this, we plan to include in the training process masks corresponding to these vague areas provided in the dataset.

2. Related Work

Historically, traditional segmentation methods like thresholding, clustering, and active contouring have been employed to extract nuclei in histopathological images [4]. While these methods can be effective under controlled conditions, they often struggle with the variability and complexity of real-world tissue samples. For example, traditional algorithms may falter in the presence of uneven staining, overlapping nuclei, and heterogeneous nuclear appearances [11].

In recent years, deep learning algorithms have gained prominence over traditional methods for cell nuclei segmentation. Researchers have extensively employed various convolutional neural network (CNN)-based models, which can autonomously learn complex features from images for tasks such as classification, detection, and segmentation. For instance, Long et al. [5] utilized classification networks like AlexNet, VGGNet, and GoogleNet for semantic segmentation by transferring and fine-tuning their learned representations. These approaches marked the early use of end-to-end deep neural networks for image-semantic segmentation, paving the way for subsequent advancements in the field.

In terms of segmentation models for cell nuclei, CNNs, such as U-Net, have become popular due to their ability to learn complex features directly from data. The U-Net architecture, introduced by Ronneberger et al., has been widely used for its effectiveness in biomedical image segmentation [10]. Variants of U-Net, including Residual U-Net [12] and Dense U-Net[13], have further improved segmentation performance by incorporating advanced network designs. U-Net and its variants are especially popular for their smaller model size, which often performs better in the data-poor biomedical setting.

Relatively older models such as YOLO, have also been shown to work well. YOLO was a ground-breaking object-detection system that reframed object detection as a regression problem, simultaneously outputting bounding box coordinates and class probabilities from just image pixels [8]. In the context of computational pathology, YOLO adapted to the analysis of the tumor microenvironment (referred to as HD-YOLO by the authors) has shown exceptional promise in outperforming existing WSI analysis methods, even generating prognostic image features that correlate with survival [9]. In our implementation of YOLO, we referred to these articles for inspiration and best practices.

Mask R-CNN, introduced by He et al. (2017), extends the Faster R-CNN model by adding a branch for predicting segmentation masks on each Region of Interest (RoI) in a pixel-to-pixel manner [2]. This architecture combines the advantages of region-based object detection and pixel-level segmentation, making it particularly suitable for tasks where precise delineation of objects is required. Several studies have successfully applied Mask R-CNN for nuclei segmentation. Naylor et al. (2018) employed Mask R-CNN to segment nuclear instances in histological images, demonstrating significant improvements over traditional methods and earlier deep learning approaches [7]. Their results highlighted the model's ability to handle the diverse morphology of nuclei and its robustness across different tissue types.

Given the success of these specific models in other studies and their popularity in segmentation problems in general, we decided to compare their performance on this new segmentation dataset and explore their efficacy for defining nuclei confidence/ambiguity.

3. Dataset and Features

Recently, scientific researchers have released one of the largest datasets of labeled nuclei in H&E stained images across 31 human and mouse organs. This dataset is called NuInsSeg, and is completely open access, as discussed in its corresponding paper [6]. Additionally, the authors go beyond providing point estimates of nuclei location by providing ambiguity maps in regions where deterministic annotation, even for the expert pathologist, is difficult. This novelty motivated our decision to perform instance segmentation to classify nuclei as either confident or ambiguous.

The inputs are H&E stained image patches of size (512, 512, 3), collected across various organs from both humans and mice. Each image patch is additionally associated with two binary masks - one for nuclei and one for the patch's ambiguous areas. The outputs are annotated H&E stained image patches of the same dimension, with bounding boxes around each putative cell nuclei and a corresponding nuclei-specific label annotation of 'uncertain' versus 'certain.' To evaluate segmentation performance, we establish the original U-Net implementation as our baseline.

The data is publicly hosted on Kaggle and is provided as a nested directory structure for each tissue type, with subfolders providing different mask format files. For YOLO and Mask R-CNN, we translated original tissue image files and associated masks into the COCO format using 'pycocotools'. We then split the data into three directories: training, validation, and testing. To evaluate the models' ability to generalize to unseen tissue types, all images of human and mouse kidneys were placed in the testing directory. For the remaining images, we performed a 80/20 train-validation split. Since the dataset was limited in size, we employed several image augmentations to artificially expand and diversify the training data.

These augmentations were facilitated by libraries such as 'albumentations'. We used the following transformations:

- *random cropping*: a random section of the image is cropped to a specified size, helping the model in learning relevant features regardless of their position within the image.
- *CLAHE (contrast limited adaptive histogram equalization)*: the contrast in local image regions is adjusted, enhancing subtle differences in tissue textures that might be crucial for segmentation.
- *random brightness & contrast*: the image's brightness and contrast are randomly altered within a defined range, allowing the model to more robustly respond to variations in lighting conditions during image acquisition.
- *hue & saturation & value*: random adjustments are applied to the image's hue (color), saturation (color in-

tensity), and value (brightness), allowing the model to learn important features regardless of slight color variations in the tissue samples.

- *horizontal & vertical flip*: the image is randomly flipped horizontally or vertically, improving the model's ability to recognize features that are independent of their orientation in the image.
- *random rotate*: the image is randomly rotated by 90, 180, or 270 degrees, helping the model become more invariant to the orientation of the tissue samples during image acquisition.
- *shift, scale, rotate*: the image undergoes a shift (move), a scale (resize), and a rotation in a single transformation, allowing the model to better handle slight variations in position, size, and orientation that might occur naturally across tissues.

4. Methods

We decided to compare the performance of three image segmentation models: U-Net, Mask R-CNN and Ultralytics YOLOv8.

4.1. U-Net[1]

A fully convolutional neural network architecture specifically designed for image segmentation tasks. It excels at pixel-wise classification, allowing it to precisely delineate the boundaries of individual nuclei or other structures in an image. This makes U-Net a popular choice for nuclear segmentation due to its ability to capture the intricate details of these structures.

- Mathematical Formulation: Let $X \epsilon R^{H\times W\times C}$ represent an input image, where H and W are the image height and width, and C is the number of channels (e.g., 3 for RGB images). U-Net utilizes an encoder-decoder structure. The encoder part progressively downsamples the input image to capture high-level features, while the decoder part upsamples the feature maps and combines them with corresponding features from the encoder path to achieve precise localization. The final output layer, denoted as $y \epsilon R^{H\times W\times C}$, predicts a probability distribution over M classes for each pixel, indicating the likelihood of each pixel belonging to a specific class (e.g., nucleus, background). A commonly used loss function for image segmentation tasks is the multi-class cross-entropy loss:

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(\hat{y}_{ij})$$

where $N = HW$ is the total number of pixels, y is the model's predicted probability distribution, and y_i is the one-hot encoded ground truth label for pixel i.

4.2. Mask R-CNN (Mask Region-Convolutional Neural Network) [2]

Mask R-CNN builds upon the success of Faster R-CNN, a deep learning architecture for object detection. It introduces an additional branch to predict segmentation masks for each detected object, making it suitable for instance segmentation tasks like nuclear segmentation. Here's a breakdown of Mask R-CNN's functionalities and the underlying mathematical concepts:

- Backbone Network: Mask R-CNN utilizes a pre-trained convolutional neural network (CNN) as its backbone. This network, like ResNet-50 or ResNet-101, extracts high-level features from the input image. The mathematical operations within the backbone network involve convolutions, non-linearities (e.g., ReLU), and pooling layers. These operations can be represented as:

  Convolution:

  $$f_k(x) = \sum_{l=1}^{C} W_k^l * x_l + b_k$$

  where x_l denotes the input feature map from the previous layer (channel l), W_l^k represents the learnable kernel weights for the k-th filter in this layer, $\star$ denotes the convolution operation. b_k is the bias term for the k-th filter, $f_k(x)$ represents the output feature map of the k-th filter.

  ReLU (Rectified Linear Unit):

  $$f(x) = max(0, x)$$

  Pooling: Various pooling operations are used to reduce the dimensionality of the feature maps while capturing essential spatial information.

- Region Proposal Network (RPN): The RPN takes the feature maps from the backbone network as input and generates region proposals (bounding boxes) that are likely to contain objects. It utilizes small convolutional layers to predict bounding box coordinates and objectness scores (confidence of a box containing an object). Mathematically, the RPN outputs:
  - Bounding box coordinates: These can be represented as offsets (Δx,Δy,Δw,Δh) applied to anchor boxes predefined at various scales and aspect ratios. Predicting these offsets allows the

[1] base implementation from https://github.com/jvanvugt/pytorch-unet

[2] base implementation from https://github.com/facebookresearch/detectron2/

model to refine the anchor boxes and achieve accurate object localization.

  - Objectness Scores: The RPN predicts a binary probability for each proposed region, indicating whether it contains an object (foreground) or not (background). This can be formulated using the sigmoid function:

$$P_{foreground} = \sigma(s)$$

  where s is the score predicted by the RPN and σ is the sigmoid function.

- Segmentation Branch:This branch operates on the feature maps from the backbone network and the refined region proposals from the RPN. It utilizes a fully convolutional network to predict a segmentation mask for each object within the proposed bounding box. The mask predicts the probability of each pixel belonging to the object of interest. The loss function used for training the segmentation branch can be a binary cross-entropy loss similar to U-Net.

- Overall Loss Function: Mask R-CNN employs a multi-task learning approach, jointly optimizing for object detection (bounding boxes) and segmentation (masks). The overall loss function is a weighted sum of individual losses:

$$L = \lambda_{cls} L_{cls} + \lambda_{reg} L_{reg} + \lambda_{mask} L_{mask}$$

  where L_{cls} is the classification loss for objectness scores (typically binary cross-entropy), L_{reg} is is localization loss for bounding box coordinates, L_{mask} is the mask loss for segmentation.

4.3. YOLOv8 with Segmentation Loss (Ultralytics)[3]

You Only Look Once (YOLO) is primarily an object detection model. However, the Ultralytics implementation of YOLOv8 extends its capabilities to include segmentation through a custom loss function.

- YOLOv8 Object Detection: YOLOv8 utilizes a single-stage network architecture to predict bounding boxes and class probabilities for objects directly from the input image. It employs various convolutional layers, activation functions, and pooling layers to extract features and predict bounding boxes and class probabilities. The mathematical formulations for these operations are similar to those described for the Mask R-CNN backbone network (convolutions, ReLU, pooling).

- Segmentation Loss: Unlike traditional YOLO models, Ultralytics YOLOv8 incorporates a segmentation loss function alongside the object detection losses (bounding box and class confidence). This loss function encourages the model to predict not only the bounding box location and class of an object but also its detailed segmentation mask.The YOLOv8-seg model employs a combined loss function consisting of three individual components:

  - Bounding Box Loss: This loss calculates the discrepancy between the predicted bounding boxes and the ground truth boxes' geometry. It measures how well the model predicts the size and location of the objects of interest (e.g., nuclei) within the images. Common formulations for bounding box loss include Intersection over Union (IoU) loss or Smooth L1 loss.
  - Objectness Loss: This loss determines how confident the model is about the presence of an object within the predicted bounding box. It essentially compares the model's predicted probability of an object being present with the actual ground truth value (object present or absent). The objectness loss is formulated using the binary cross-entropy function.
  - Segmentation Loss: This loss quantifies how close the predicted segmentation mask is to the ground truth mask. It measures how effectively the model performs the semantic segmentation task, accurately delineating the boundaries of the objects within the image.

- Overall training: During training, the Ultralytics YOLOv8 model optimizes a combined loss function that incorporates both object detection losses and the segmentation loss. This allows the model to learn to perform both tasks simultaneously: identifying and localizing objects (with bounding boxes) and segmenting them (with masks).

Traditionally, YOLO is used for object detection tasks like traffic monitoring, people detection in videos, etc. However, we use the Ultralytics YOLOv8 which has a custom loss function combining the bounding box loss, objectness loss and segmentation loss. This enables the model to simultaneously perform object detection and semantic segmentation. U-Net is a popular choice for nuclear segmentation and also a state of the art method due to its ability to perform pixel-wise classification. This allows it to precisely delineate the boundaries of individual nuclei or other structures in an image. Mask R-CNN is yet another popular choice for segmentation tasks as it excels at providing pixel-wise segmentation masks in addition to the bounding box.

[3] base implementation from https://github.com/ultralytics/ultralytics

These properties might facilitate precise analysis of nuclear morphology or boundaries.

5. Experiments

Given that the models (specifically U-Net compared with the instance segmentation models) were trained using different loss functions and for slightly different objectives, we needed to come up with sufficient metrics to compare fairly. For comparing the segmentation quality, we decided to use the DICE score and Aggregated Jaccard Index (AJI), as was used in the NuInsSeg paper.

The Dice score is a measure of overlap between two samples. It is defined as:

$$\text{DSC} = \frac{2 \times |X \cap Y|}{|X| + |Y|} \tag{1}$$

where X is the set of predicted nuclei and Y is the set of ground truth nuclei. A Dice score of 1 indicates perfect overlap, while a score of 0 indicates no overlap.

The AJI is a more complex metric that evaluates the accuracy of instance segmentation by accounting for both detection and segmentation quality [3]. It is defined as:

$$\text{AJI} = \frac{\sum_i |A_i \cap B_i|}{\sum_i |A_i \cup B_i| + \sum_k |B_k|} \tag{2}$$

where A_i represents the ground truth objects, B_i represents the predicted objects, and B_k are the false positive predictions. The AJI penalizes both false positives and false negatives, providing a comprehensive measure of segmentation performance.

For comparing the performance of the instance segmentation models, we preferred to use Average Precision, reporting both AP50 (average precision at $IoU > 0.5$) and APm (mean average precision over $IoU \in (0.5, 0.95)$). These metrics are more common in the instance segmentation literature and, though they exhibit some issues such as being insensitive to small changes and producing gross changes in score near the threshold [1], they are sufficient for our setting.

For all three of our chosen models, there are well-established defaults for various hyperparameters. For YOLOv8 and Mask R-CNN especially, these are encoded as the defaults for the Ultralytics and Detectron2 frameworks respectively. This was the case for learning algorithm, for instance (we used Adam for YOLOv8 and U-Net and SGD for Mask R-CNN). We did not spend significant compute tuning the learning rate, but we did ensure suitable loss curves over a range of rates. Testing rates from 10^{-2} to 10^{-5} resulted in selecting 10^{-3} for all models. Furthermore, we used a linear scheduler for YOLOv8 and a stepwise scheduler with a 20% decay for Mask R-CNN. These slight differences in hyperparameters do make direct comparisons of the models difficult, but we attempted to balance consistency across models with choosing performant hyperparameters for each model architecture (whether the default parameter or best one empirically).

Beyond the optimizer configurations, we also experimented with other hyperparameters through ablation tests for all models. Specifically, we tested various sizes of the backbone model, batch sizes, and numbers of frozen layers (as well as use of batch normalization in the U-Net model). Given that the YOLO and Mask R-CNN models are much larger than U-Net, we chose to use pre-trained weights (trained on the COCO dataset) and finetuned them on our NuInsSeg dataset. In contrast, the U-Net model was trained from scratch on our dataset and represents the historical approach to biomedical data, assuming that this setting is too far removed from everyday images to benefit from pre-training. We then chose the best objective model resulting from these tests to perform later comparisons and exploration of the ambiguous maps. Overall, the models were quite robust to changes in architecture, as measured by the DICE and AJI scores applied to the validation sets 1. There was significant drop-off in performance when freezing a large number of block layers; however, un-freezing all the layers (as compared to freezing the first few) showed only modest gains. Similarly, using a larger model did not result in much gain in performance; for U-Net, the 7-layer model performed worse as measured by both metrics than the standard 5-layer model. This result is slightly unintuitive, however our relatively small dataset (we trained on 491 images and 22122 segmented nuclei) likely limits the ability to train larger models.

From these results, we selected our best performing model to use for comparison and further testing. For YOLO, this was the large backbone with no frozen layers (and batch size of 1 due to computational constraints). For Mask R-CNN, this was the smaller backbone with no frozen layers and batch size 4. Finally, for U-Net we used the smaller model (5 layers) with batch normalization.

In the spirit of the literature of nuclei segmentation models and building upon the NuInsSeg publication, we next compared the performance of our best-performing YOLOv8 and Mask R-CNN models to U-Net. To find the best score threshold to use for inference, we performed a metric sweep from 0.1 to 0.5 on the validation set. We report metrics on both the validation and test sets using the best threshold (0.2 for YOLO, 0.5 for Mask R-CNN and U-Net). Somewhat to our surprise, our two suggested models perform quite well as compared to U-Net, with Mask R-CNN having much better performance than U-Net on the validation set 3. Furthermore, the results on the held-out kidney test set show Mask R-CNN's consistent performance on data from a different domain. These results validate the decision to apply larger models (with pre-trained weights) to the problem of nuclei segmentation and allowed us to

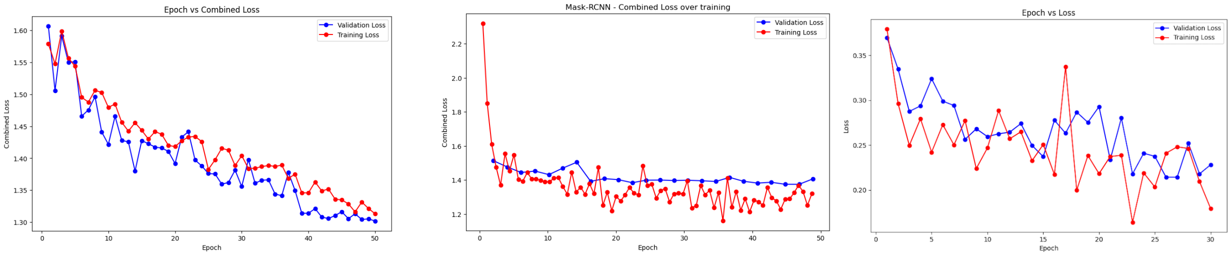

Figure 1. Loss curves for each final model, comparing the loss on the training and validation datasets.

		YOLO		Mask R-CNN	
		mAP	AP50	mAP	AP50
Backbone	Small	0.425	0.784	0.456	0.682
	Large	0.428	0.785	0.447	0.680
Frozen Layers	0	0.428	0.785	0.451	0.668
	5/2	0.402	0.771	0.439	0.679
	10/3	0.400	0.752	0.447	0.680
	20/4	0.181	0.410	0.393	0.599
Batch Size	1/4	0.428	0.785	0.447	0.680
	8/16	0.410	0.771	0.427	0.663

Table 1. Ablation tests for YOLO and Mask R-CNN, testing various architecture hyperparameters. The small and large backbone sizes correspond to 261/401 layers for YOLO and ResNet50/101 for Mask R-CNN. The frozen layers parameter corresponds to the number of layer groups frozen, groupings which differ between the two architectures. The batch size differs between the two architectures only due to computational constraints.

	5 Layers		7 Layers	
	DICE	AJI	DICE	AJI
Batch Norm	0.565	0.259	0.551	0.220
No Batch Norm	0.458	0.166	0.393	0.129

Table 2. Ablation test for the U-Net model, including model size (either the standard 5 layers or larger 7 layer model) and including or excluding a batch normalization layer.

	U-Net		YOLOv8		Mask R-CNN	
	DICE	AJI	DICE	AJI	DICE	AJI
Val	0.565	0.259	0.621	0.542	0.760	0.546
Test	0.688	0.407	0.596	0.427	0.768	0.590

Table 3. Comparison of the three models on their best runs, following the ablation study of the hyperparameter choices. The performance on both the validation and test set are compared over the DICE and AJI metrics.

further experiment with confidence in our model. See 2 for an example from the test set of the predictions from each model.

Since Mask R-CNN exhibited the best performance in the segmentation task, we chose to use it primarily to test the utilization of ambiguous region annotations. First, we analyzed the change in model performance when removing the ambiguous nuclei from the training set altogether. These results were inconclusive, with similar but improved performance on the validation set as compared to using all nuclei for training. Qualitatively, the training loss was much lower with the ambiguous nuclei removed. This is expected when removing the most difficult to segment nuclei (with associated noisy ground truth labels). However, this performance gain in the combined loss is not maintained when applying the trained model to the validation set. Nonetheless, the validation metrics do show slight improvements over the course of training, with very small but noticable gains in AJI and APm. Overall, these experiments display the benefits of having clean annotations and warrant further experimentation with these ambiguous masks; perhaps instead of removing the ambiguous nuclei, they could be down-weighted in the loss calculations.

Finally, we took advantage of the multi-class instance segmentation capabilities of Mask R-CNN by training the model to segment nuclei and classify them into two classes: ambiguous or non-ambiguous. Our interest in this task was two-fold. First, we wondered if the model could learn features that correlated with a human annotation of ambiguous, thereby enabling binary classification of nuclei with a notion of confidence. This differs from the instance confidence scores output from the model in that it is learned from the human ambiguity annotations and therefore maps more cleanly to established slide annotation processes. Second, we wondered if this classification task along with the inherent class imbalance would serve to downweight the loss from these ambiguous regions. We knew that the training loss would take a significant hit due to the addition of a new class, not to mention a class that is difficult to distinguish from the original nuclei class; however, we hoped that the resulting model would be robust to difficult regions and perform better on a wide variety of samples.

The evaluation metrics did identify many false negatives and positives, nuclei that were likely misclassified as ambiguous (the segmentation APm was 24.52 vs. 44.13). However, this strict metric does not fully represent the models performance. Taken together, the two classes do predict the nuclei mask very well (the DICE score collapsed over the classes is comparable and the AJI is even higher). Furthermore, the nuclei labelled as ambiguous do seem to occur

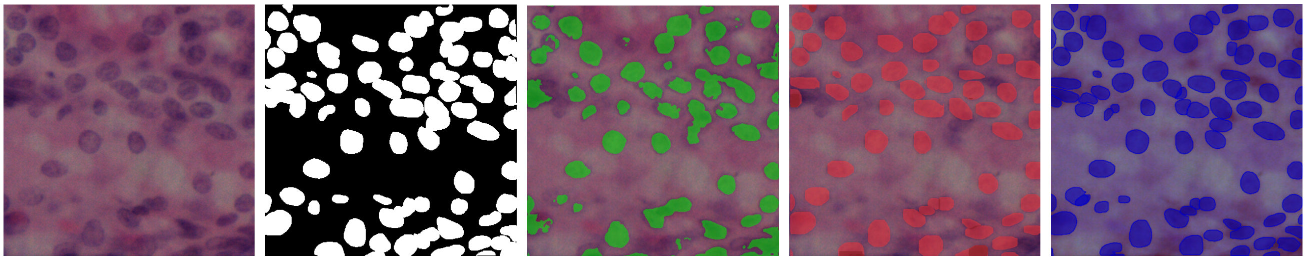

Figure 2. A human kidney image with its ground truth segmentation mask, followed by the predictions from the three models with the optimal hyperparameters: U-Net (green), YOLOv8 (red), and Mask R-CNN (blue).

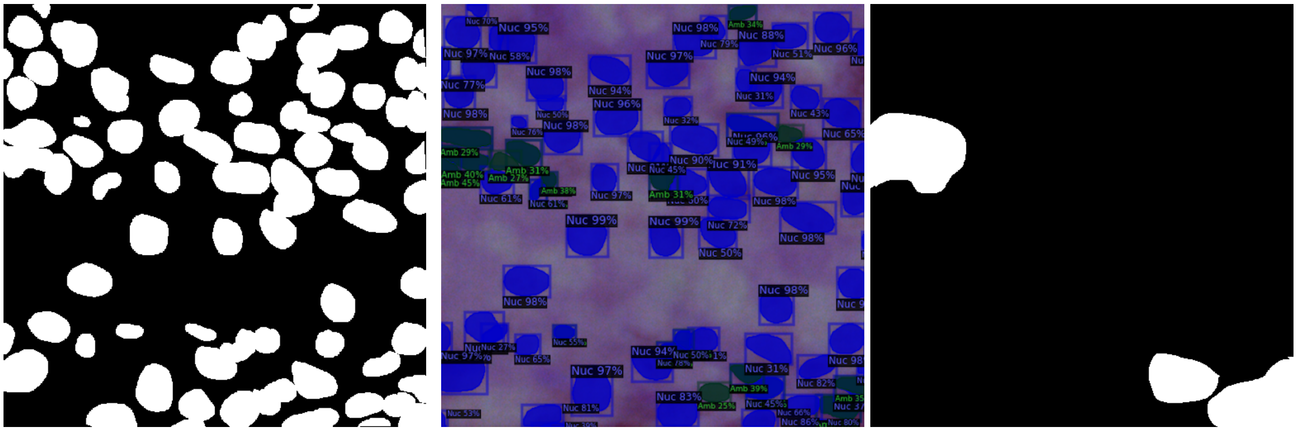

Figure 3. The same human kidney image mask with the Mask R-CNN prediction (threshold=0.2) from the ambiguous classification run. The dark green instances mark the nuclei classified as ambiguous by the model. All nuclei are also labeled by their confidence. The final mask shows the ground truth ambiguous regions.

often in the difficult regions 3. In future studies, we may explore further uses of the ambiguous masks, specifically interrogating the model's confidence levels in different regions and potentially modifying the loss function further to accommodate this added information. Overall, we want to work towards models that systematically take in and output uncertainty values for more robust interpretation and usage.

6. Conclusion

In this study, we compared different models for nucleic segmentation, specifically evaluating the performance of U-net, YOLO and Mask R-CNN. Our experiments aimed to provide a fair comparison by using appropriate metrics and hyperparameter settings for each model. We evaluated segmentation quality using the DICE score and Aggregated Jaccard Index and assessed instance segmentation performance with Average Precision (AP50 and APm).

Our findings indicate that Mask R-CNN outperformed U-Net and YOLO in both segmentation quality and instance segmentation metrics. Notably, Mask R-CNN demonstrated robust performance across different dataset partitions, including a held-out kidney test set, highlighting its ability to generalize well to new tissue types that it hadn't seen during training and validation. YOLO also showed competitive results but was slightly behind Mask R-CNN in terms of overall performance.

Our final set of experiments included using the ambiguous nuclei annotation during model training. Since Mask R-CNN exhibited the best performance in the segmentation task, we chose to use it for this task. Removing ambiguous nuclei from the training set yielded inconclusive results. While it slightly improved validation performance and reduced training loss, the validation metrics showed only minor improvements in AJI and APm, suggesting potential benefits of clean annotations.

Future studies may explore further uses of ambiguous masks, specifically interrogating the models' confidence levels in different regions and potentially modifying the loss function to accommodate this added information. Another interesting future direction to pursue would be to use advanced data augmentation techniques, such as generative adversarial networks (GANs), to create synthetic training data, enhancing model robustness to variability in tissue images.

7. Contributions & Acknowledgements

C.M. implemented Mask R-CNN and conducted corresponding experiments for this model. R.K. implemented

U-Net and conducted corresponding experiments for this model. S.B. implemented YOLO and conducted corresponding experiments for this model. C.M., R.K., and S.B. wrote the paper. Publicly available model code used throughout experimentation are credited in the footnotes.

References

[1] L. Chen, Y. Wu, J. Stegmaier, and D. Merhof. Sortedap: Rethinking evaluation metrics for instance segmentation. *arXiv preprint arXiv:2309.04887*, 2023.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[3] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE Transactions on Medical Imaging*, 36(7):1550–1560, 2017.

[4] Z. Liu, C. Ma, W. She, and M. Xie. Biomedical image segmentation using denoising diffusion probabilistic models: A comprehensive review and analysis. *Applied Sciences*, 14(2), 2024.

[5] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[6] A. Mahbod, C. Polak, K. Feldmann, R. Khan, K. Gelles, G. Dorffner, R. Woitek, S. Hatamikia, and I. Ellinger. Nuinsseg: A fully annotated dataset for nuclei instance segmentation in he-stained histological images. *Scientific Data*, 11(1), Mar 2024.

[7] P. Naylor, M. Laé, F. Reyal, and T. Walter. Nuclei segmentation in histopathology images using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 0–0, 2018.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[9] R. Rong, H. Sheng, K. W. Jin, F. Wu, D. Luo, Z. Wen, C. Tang, D. M. Yang, L. Jia, M. Amgad, and et al. A deep learning approach for histology-based nucleus segmentation and tumor microenvironment characterization. *Modern Pathology*, 36(8):100196, Aug 2023.

[10] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.

[11] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. Snead, I. A. Cree, and N. M. Rajpoot. Densely convolutional spatial attention network for nuclei segmentation of histological images for computational pathology. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6216–6225. IEEE, 2019.

[12] X. Xiao, S. Lian, Z. Luo, and S. Li. Weighted res-unet for high-quality retina vessel segmentation. In *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, pages 327–331. IEEE, 2018.

[13] H. Zhu, F. Shi, L. Wang, S. C. Hung, M. H. Chen, S. Wang, et al. Dilated dense u-net for infant hippocampus subfield segmentation. *Frontiers in Neuroinformatics*, 13, 2019.