

Supplementary Material

for “Evaluating Iterative Deep Learning as a Labeling-Efficient Strategy for Tubular Segmentation in Digital Nephropathology”

Borghild Tednes Larsen, Yvan Samir Belakebi-Joly, Maya Maya Barbosa Silva,
Sabine Leh, Hrafn Weishaupt

Department of Pathology, Haukeland University Hospital, Bergen, Norway

Sabine Leh also affiliated with Department of Clinical Medicine, University of Bergen, Norway

Borghild Tednes Larsen also affiliated with Department of Physics and Technology, University of Bergen, Norway

Corresponding author: Borghild Tednes Larsen, Email: borghild.tednes.larsen@gmail.com

List of Figures

S1	Illustration of the QA workflow used in this project	9
S2	Training procedure for initial and final models	10
S3	Illustration of class balancing procedure	10
S4	Regression model	11
S5	Comparison of observed and predicted annotation speeds across the different models	11
S6	Example of post-processing effect on image tile with normal tubules	12
S7	Image 1 (Class 0) predictions at three thresholds	13
S8	Image 2 (Class 0) predictions at three thresholds	14
S9	Image 3 (Class 0) predictions at three thresholds	15
S10	Image 4 (Class 1) predictions at three thresholds	16
S11	Image 5 (Class 1) predictions at three thresholds	17
S12	Image 6 (Class 1) predictions at three thresholds	18
S13	Image 7 (Class 2) predictions at three thresholds	19
S14	Image 8 (Class 2) predictions at three thresholds	20
S15	Image 9 (Class 2) predictions at three thresholds	21
S16	Image 10 (Class 3) predictions at three thresholds	22
S17	Image 11 (Class 3) predictions at three thresholds	23
S18	Image 12 (Class 3) predictions at three thresholds	24

List of Tables

S1	Tubular lesion classes	6
S2	Morphological characteristics of tubular lesion classes.	6
S3	Object counts before and after post-processing	7
S4	Per-class metrics at different thresholds for the different models	7
S5	Per-class Mean DSC and IoU at different thresholds for the different models	7
S6	Shapiro–Wilk normality test p-values for QA and baseline models.	8
S7	Pixel-level statistical comparison of QA and baseline models.	8
S8	Instance-level statistical comparison (IoU > 0.5) of the different models.	8
S9	Instance-level statistical comparison (IoU > 0.75) of the different models.	8
S10	Instance-level statistical comparison (IoU > 0.9) of the different models.	8

Supporting Information

Setup

Annotations were conducted using Quick Annotator (QA) [1] and QuPath [2]. QA (version 1.0) was run locally and used for interactive annotation on extracted image tiles. Specifically, 2048×2048 px PNG tiles were used, within which manual regions of interest (ROIs) of size 512×512 px were selected for annotation. QuPath (version 0.5.1) was employed for whole slide image (WSI) visualization, manual, and AI-assisted segmentation using the Segment Anything Model (SAM) extension [3].

The computational environment was managed using Anaconda, with Python 3.8 for QA and Python 3.10 for analysis. Analysis was conducted in VSCode and Jupyter Notebooks. Key packages included `openslide`, `shapely`, `geopandas`, `opencv-python`, `pytorch`, and `scikit-learn`.

Annotation and model training were performed across three computing environments: a Windows-based ASUS Vivobook S 14 Flip equipped with an Intel Core i5-12500H CPU, 16 GB RAM, and an external NVIDIA Quadro P5000 GPU via a Razer Core X enclosure; an HP EliteBook 840 G8 with an Intel Core i5-1145G7 CPU and 16 GB RAM running Windows 11 Enterprise; and a high-performance Linux server (Red Hat Enterprise Linux 9.5) with four Intel Xeon cores, 125 GB RAM, and CUDA 12.4 support. All systems supported manual annotation using a Wacom Cintiq 16 drawing tablet paired with a Wacom Pro Pen 2.

Data

Ground Truth Development

The test dataset consisted of 12 WSIs distributed evenly between the four predefined classes. Representative 2048×2048 px tiles, one for each slide in the test dataset were selected by an experienced nephropathologist. All tubules within each tile were manually annotated in QuPath using the brush tool. These annotations served as the ground truth (GT). Each annotated tubule was classified into one of four categories: normal, acutely injured, chronically injured, or atrophic. The annotation protocol followed a standardized set of rules: the basement membrane was carefully traced, even when highly wrinkled; adjacent but distinct tubules were annotated separately; overlapping annotations were avoided; and for diseased tubules, the boundary was placed along the outer edge of the basement membrane.

Data Preparation for Performance Evaluation

QA requires input images in `.png` format. To ensure compatibility, each selected test tile was exported from QuPath as a `.png` file using the ImageJ extension. Similarly, for the ComPrePS/CMIL Slide Archive framework, which accepts `.svs` or `.tif` formats, each tile was also exported as a `.tif` file, enabling evaluation of pretrained models without the overhead of full WSI processing.

To mitigate prediction artifacts at tile borders, each 2048×2048 px GT tile was centered within a larger 4096×4096 px tile containing only raw image data and no annotations. These larger tiles were used as input for both QA (`.png`) and ComPrePS (`.tif`) during prediction. After inference, predictions were post-processed and cropped back to the central 2048×2048 px region for evaluation against the GT mask.

GT annotations and predictions on the test data were converted to binary masks saved as PNG-images for performance evaluation.

QA outputs segmentation predictions as binary masks in `.png` format. To generate corresponding GT masks for evaluation, the following steps were performed:

1. All tubule annotations within a tile were merged in QuPath, excluding annotations outside the tile's ROI.
2. The merged annotations were intersected with the ROI to isolate structures fully contained within tile boundaries.
3. The intersection was converted to a binary mask and exported as a `.png` file using ImageJ.

For the Lucarelli models, segmentation predictions were provided as `.json` files containing polygon coordinates. These were converted into binary masks using the `json` and `OpenCV` libraries in Python.

This standardized data preparation pipeline ensured consistency across input formats and enabled reproducible evaluation of segmentation performance across all models.

Annotation Protocol

Annotations were created by tracing along the tubular basement membrane. In cases of thickened basement membranes, the boundary was placed along the outer edge. Adjacent tubules were annotated as separate objects, even when their basement membranes were in contact.

For tubules truncated at the biopsy edge, where the lumen was continuous with the background, the annotation excluded the lumen, following the contour of the tubular cells.

For the initial training on normal tubules, the tiles were fully annotated (excluding background), and the ROIs were placed adjacent to one another. The final model was trained on two fully annotated WSIs of morphologically normal tubules. In contrast, when incorporating data from tubular lesions, a new WSI was introduced at each iteration, and ROIs and tiles were selected based on data complexity, with the aim of improving the model’s generalization performance. This difference in annotation workflows is illustrated in S2.

QA Configuration

Models were trained iteratively using the QA tool installed locally on a Linux server and a Windows laptop. Separate projects were created for the different models. WSIs were patched into 2048×2048 px tiles in .png format using the script `extract_tiles_from_wsi_openslide.py` provided on QAs GitHub page (<https://github.com/choosehappy/QuickAnnotator>), with the option of omitting patches containing only background in use. The intensity thresholds for what is considered background were adjusted based on experience to avoid omitting lightly stained tissue (mean grayscale value > 240 or < 180).

Annotations were produced in ROIs of 512×512 px using a polygon tool and uploaded to either training or validation sets. The validation set (in QA called test set) is used for monitoring training and early stopping.

Training configuration and hyperparameters The models are based on a U-Net architecture with a depth of five layers and approximately 113k parameters, pretrained in an autoencoding fashion and subsequently fine-tuned for segmentation. Training is performed using a weighted pixel-wise cross-entropy loss with class weighting and an additional edge-weighting scheme that increases the contribution of boundary pixels to the loss.

Model training was based on the default configuration provided in the QA repository, with minor modifications. The patch size was increased from 256 to 512 pixels to provide a larger spatial context for tubular structures. The edge-weight parameter used during training was increased from 2 to 8 to place greater emphasis on boundary delineation.

All other training-related hyperparameters remained unchanged and consistent with the default QA configuration. Superpixel-related parameters were adjusted during preliminary experimentation (e.g., approximate cell size and compactness), but these were not used in the final workflow and did not affect the reported results.

Performance Metrics

The instance-wise agreement between the predicted outputs and the ground truth was assessed using precision, recall, and F1 scores (Equations 1, 2 and 3). These metrics were calculated based on the counts of true positives (TP), false positives (FP), and false negatives (FN) at IoU (intersection over union) thresholds of 0.5, 0.75, and 0.9, i.e. a prediction was considered correct if its overlap with the ground-truth annotation met or exceeded the given IoU threshold.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

The pixel-wise agreement between the predicted output and the ground truth was assessed using IoU and DSC (Dice score coefficient) (equations 4 and 5).

$$IoU = \frac{|Y \cap \hat{Y}|}{|Y \cup \hat{Y}|} \quad (4)$$

$$DSC = \frac{2|Y \cap \hat{Y}|}{|Y| + |\hat{Y}|} \quad (5)$$

Average precision (AP) over IoU thresholds 0.5 to 0.95 with 0.05 increments was also calculated and the resulting metric was denoted as AP(0.5:0.95:0.05).

Statistical significance analysis on performance

To statistically compare the QA, Lucarelli 2023, and Lucarelli 2023 (diabetic) models, it is first evaluated whether the performance of the models differs significantly using a Friedman’s test. This test is employed because a three-way comparison is conducted on paired observations (12 whole slide images), and normality of the distribution of the different metrics cannot be assumed. Normality had previously been assessed using a Shapiro–Wilk test and was rejected in multiple cases. If the result of the Friedman’s test is significant at a threshold $\alpha = 0.05$, a post hoc pairwise Nemenyi test is performed to identify which pairs of models differ. No additional correction is applied for the following reasons: the Nemenyi test already controls error rates, and, as the analysis is exploratory, the use of unadjusted p-values avoids overly conservative corrections. This analysis is conducted across the different metrics used, at both pixel and instance level.

Post-Processing Procedures

Filtering and hole-filling were applied as post-processing steps. A data-driven strategy was employed to determine an appropriate area threshold for filtering. The distribution of manually annotated tubule areas (regression dataset, $n = 352$, including both normal and diseased tissue) was analyzed using logarithmic binning with five bins per decade to accommodate the skewed nature of the data. The initial tail of the distribution, defined as the first consecutive set of bins prior to a noticeable increase in frequency, was isolated, and the median area within this subset was computed. This yielded a threshold of 1150 px^2 , or $70 \text{ }\mu\text{m}^2$, which was subsequently used to filter predicted objects during evaluation. The threshold was chosen to exclude implausibly small predictions, likely arising from noise or segmentation artifacts, while preserving anatomically plausible tubule structures. It was applied during evaluation to exclude predicted objects falling below the defined size, thereby reducing false positives.

Additionally, holes within predicted QA regions were filled to improve mask completeness and enhance overlap-based segmentation metrics. Hole filling was performed using the `OpenCV` library by identifying the external contours of predicted regions and reconstructing filled masks based on these contours. This approach ensured that interior voids, often arising due to incomplete lumen segmentation, did not negatively affect pixel-wise evaluation metrics such as DSC or IoU.

External Model Details

- Lucarelli 2023 (General) [4]: A panoptic segmentation model built on the Detectron2 framework with a ResNet-50 backbone. It was trained on a diverse renal pathology dataset, excluding tubules smaller than $200 \text{ }\mu\text{m}^2$. The evaluation cohort included both histologically normal biopsies and those with pathological changes.
- Lucarelli 2023 (Diabetic) [5]: A Panoptic-DeepLab model with a ResNet-50 backbone, trained on diabetic nephropathy (DN), lupus nephritis, and transplant kidney biopsies. Annotations were created using the H-AI-L (Human-AI-Loop) pipeline [6]. Evaluation was performed exclusively on DN cases.

Regression Model for Estimating Manual Annotation Time

To enable a context-aware speed-up analysis (Method 4), a regression model was trained to estimate manual annotation speed based on patch-level complexity. The primary predictor was the mean number of annotated structures per ROI ($512 \times 512 \text{ px}$), selected via correlation analysis over alternatives such as mean tubule area, solidity, and circularity.

The model was trained on 104 manually annotated ROIs from QA, including 50 from normal tissue and 54 from diseased kidney tissue. Multiple regression algorithms and hyperparameter combinations were evaluated using a grid search. A second-degree polynomial Lasso regression model was selected based on its low root-mean-square error (RMSE) and agreement with expected annotation patterns (e.g., slower speeds in diseased tissue).

The final model used an α value of 0.05 and was validated using 5-fold cross-validation. Predicted annotation speeds were then used as baseline estimates for each QA iteration, allowing speed-up to be calculated relative to local patch complexity rather than assuming a fixed manual speed.

Method for statistical comparison of different QA-models’ annotation intervals

All analyses were performed in Python using pandas, NumPy, and statsmodels. Annotation performance was evaluated by comparing observed annotation speed ($speed_{actual}$) with the model-predicted speed ($speed_{pred}$).

To quantify relative efficiency, we defined a Normalized Efficiency Score (NES) as the ratio of observed to predicted speed as defined in method 4:

$$NES = \frac{speed_{actual}}{speed_{pred}} \quad (6)$$

The NES was log-transformed prior to modeling to better satisfy linear model assumptions. The models should be compared on similar total manual annotation times. The study is thus separated in two phases: the first 100 minutes (before onset), and the next 100 minutes (after onset). The 5-minute model was stopped after 100 minutes of manual annotation, therefore the second phase only compares 10-minute and 20-minute annotation intervals. The effect of model condition on annotation efficiency is evaluated using a linear mixed-effects model with log-transformed NES as the dependent variable. The model includes model condition (categorical) and total annotation time (continuous) as fixed effects, and a random intercept for patch identity to account for repeated measurements across segmentation patches:

$$\log(NES) \sim C(Model) + total\ annotation\ time \quad (7)$$

The 10-minute condition served as the reference level for the categorical model variable. Model coefficients were estimated using maximum likelihood. To facilitate interpretation of the results of the second phase, the coefficient for the 20-minute condition was exponentiated and expressed as a percentage change in NES relative to the 10-minute reference:

$$Percent\ change = \exp(\beta) - 1 \quad (8)$$

This value represents the multiplicative improvement in annotation efficiency associated with the longer annotation interval.

Figure Generation

All plots were generated using Python, and workflow illustrations were created with draw.io. Some icons used in the workflow illustrations were obtained from Freepik.com and iamvector.com.

Overview of Supplementary Figures

The supplementary figures provide additional methodological details, visual analyses, and qualitative results supporting the main claims of this study. Figures S1–S3 describe the overall QA workflow and model training strategy, including the initial and final training procedures and the approach used to balance data across varying degrees of tubular damage. Figures S4–S5 support the analysis of annotation efficiency by presenting the regression model used to estimate manual annotation speed and its relation to observed annotation speed. Figure S6 illustrates the post-processing steps applied to prediction outputs. Finally, Figures S7–S18 provide qualitative evaluation results, showing TP, FP, and FN predictions across models and decision thresholds.

Supplementary Tables

Table S1: Tubular lesion classes with associated atrophy grade and example diagnoses.

Class	Tubular atrophy	Example diagnoses
Normal tubules (0)	Not present	Minimal change nephropathy, IgA nephropathy, Membranous nephropathy, Benign nephrosclerosis
Acute tubular injury (1)	Not present	Acute tubular injury, Tubulointerstitial nephritis, ANCA-associated glomerulonephritis, Anti-GBM glomerulonephritis
Chronic tubular injury (2)	Slight or moderate	Tubulointerstitial nephritis, ANCA-associated glomerulonephritis, Myeloma kidney
Tubular atrophy (3)	Moderate or severe	IgA nephropathy, Benign nephrosclerosis, Diabetic nephropathy, Focal segmental glomerulosclerosis

Table S2: Morphological characteristics of tubular lesion classes.

Class	Histological features
Normal tubules (0)	<p><i>Proximal tubules</i>: wide diameter; lined by cuboidal epithelial cells and prominent brush border (apical microvilli). Lumen often appears star-shaped in cross-section. Fewer nuclei per tubular profile.</p> <p><i>Distal tubules</i>: narrower diameter and no brush border. Display sharp apical borders and a relatively higher number of nuclei per profile.</p> <p><i>Collecting ducts</i>: similar in diameter to proximal tubules but lined by low cuboidal to columnar cells with distinct cell borders and abundant visible nuclei.</p>
Acute tubular injury (1)	<p>Tubules appear dilated and lined by flattened epithelial cells with irregular or reactive nuclei.</p> <p>Loss of apical brush border and cell polarity.</p> <p>Lumen may contain sloughed epithelial cells or amorphous debris.</p> <p>Minimal or no interstitial fibrosis.</p>
Chronic tubular injury (2)	<p>Tubules reduced in diameter; epithelium appears flattened or attenuated.</p> <p>Basement membranes may be thickened or multilamellated.</p> <p>Often associated with varying degrees of interstitial fibrosis.</p>
Tubular atrophy (3)	<p><i>Shrunk tubules</i>: small diameter, collapsed or obliterated lumen, thickened and wrinkled basement membrane.</p> <p><i>Dilated tubules with hyaline casts</i>: tubules appear enlarged and filled with protein-rich material. They are lined by flat, atrophic epithelium, while the basement membrane typically retains normal thickness.</p> <p><i>Endocrine-like tubules</i>: small, round profiles lined by cuboidal cells with round nuclei and minimal or absent lumen; basement membrane may be thin or invisible.</p> <p>Typically surrounded by dense interstitial fibrosis.</p>

Table S3: Object counts before and after post-processing for all models compared to ground truth. Image ID 1-3; class 0, Image ID 4-6; class 1, Image ID 7-9; class 2 and Image ID 10-12; class 3.

Image ID	GT	QA (Raw)	QA (Post-Processed)	Lucarelli 2023	Lucarelli 2023 (Diabetic)
1	49	56	48	45	45
2	85	96	88	86	83
3	76	88	74	77	75
4	33	103	38	30	32
5	89	111	88	86	83
6	65	77	72	66	64
7	61	115	69	57	51
8	48	57	45	47	48
9	85	263	90	77	75
10	72	108	87	77	73
11	88	128	91	81	79
12	75	107	89	78	77

Table S4: Per-class metrics at different thresholds for QA, Lucarelli (2023), and Lucarelli (2023 Diabetic) models split by the different classes in the test data

Class	QA			Lucarelli 2023			Lucarelli 2023 Diabetic		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Threshold 0.50									
0	0.929	0.929	0.929	0.957	0.948	0.952	0.931	0.900	0.915
1	0.843	0.893	0.868	0.918	0.893	0.905	0.899	0.861	0.880
2	0.766	0.884	0.821	0.918	0.928	0.923	0.859	0.807	0.832
3	0.615	0.669	0.641	0.784	0.746	0.764	0.740	0.690	0.714
Threshold 0.75									
0	0.805	0.805	0.805	0.885	0.876	0.880	0.808	0.781	0.794
1	0.444	0.471	0.457	0.758	0.738	0.748	0.615	0.588	0.601
2	0.474	0.547	0.508	0.721	0.729	0.725	0.653	0.613	0.632
3	0.144	0.157	0.151	0.326	0.310	0.318	0.299	0.278	0.288
Threshold 0.90									
0	0.124	0.124	0.124	0.394	0.390	0.392	0.197	0.190	0.194
1	0.010	0.011	0.010	0.159	0.155	0.157	0.067	0.064	0.066
2	0.038	0.044	0.041	0.213	0.215	0.214	0.088	0.083	0.085
3	0.000	0.000	0.000	0.008	0.008	0.008	0.004	0.004	0.004

Table S5: Per-class Mean DSC and IoU for QA, Lucarelli (2023), and Lucarelli (2023 Diabetic) models

Class	QA		Lucarelli 2023		Lucarelli 2023 Diabetic	
	Mean DSC	Mean IoU	Mean DSC	Mean IoU	Mean DSC	Mean IoU
0	0.927	0.863	0.944	0.894	0.921	0.854
1	0.868	0.767	0.894	0.811	0.860	0.755
2	0.890	0.803	0.921	0.854	0.890	0.802
3	0.761	0.620	0.836	0.719	0.805	0.676

Table S6: Shapiro–Wilk normality test p-values for QA, Lucarelli (2023), and Lucarelli (2023 Diabetic) models.

Metric	QA	Lucarelli 2023	Lucarelli 2023 (Diabetic)
IoU	0.037	0.17	0.27
DSC	0.011	0.12	0.14
Precision IoU>0.5	0.39	0.14	0.24
Recall IoU>0.5	0.021	0.046	0.17
F1 IoU>0.5	0.19	0.014	0.18
Precision IoU>0.75	0.82	0.079	0.095
Recall IoU>0.75	0.74	0.037	0.22
F1 IoU>0.75	0.84	0.049	0.16
Precision IoU>0.9	0.0014	0.13	0.093
Recall IoU>0.9	0.0018	0.16	0.083
F1 IoU>0.9	0.0016	0.13	0.089

Table S7: Friedman test and post-hoc pairwise Nemenyi test p-values for pixel-level statistical comparison. Abbreviations Lucarelli 2023: L23 and Lucarelli 2023 (diabetic): L23 (Dia).

Metric	Friedman	QA vs L23	QA vs L23 (Dia)	L23 vs L23 (Dia)
IoU	5.09×10^{-4}	1.51×10^{-3}	0.98	3.13×10^{-3}
DSC	3.66×10^{-4}	1.51×10^{-3}	0.98	3.13×10^{-3}

Table S8: Friedman test and post-hoc pairwise Nemenyi test p-values for instance-level statistical comparison (IoU > 0.5). Abbreviations Lucarelli 2023: L23 and Lucarelli 2023 (diabetic): L23 (Dia).

Metric	Friedman	QA vs L23	QA vs L23 (Dia)	L23 vs L23 (Dia)
Precision	5.09×10^{-4}	3.10×10^{-4}	0.23	6.38×10^{-2}
Recall	9.56×10^{-4}	0.10	0.33	1.50×10^{-3}
F1	3.76×10^{-3}	3.13×10^{-3}	0.56	6.38×10^{-2}

Table S9: Friedman test and post-hoc pairwise Nemenyi test p-values for instance-level statistical comparison (IoU > 0.75). Abbreviations Lucarelli 2023: L23 and Lucarelli 2023 (diabetic): L23 (Dia).

Metric	Friedman	QA vs L23	QA vs L23 (Dia)	L23 vs L23 (Dia)
Precision	9.61×10^{-5}	5.4×10^{-5}	3.80×10^{-2}	0.16
Recall	1.13×10^{-4}	8.5×10^{-5}	0.23	2.89×10^{-2}
F1	9.61×10^{-5}	5.4×10^{-5}	0.16	3.80×10^{-2}

Table S10: Friedman test and post-hoc pairwise Nemenyi test p-values for instance-level statistical comparison (IoU > 0.9). Abbreviations Lucarelli 2023: L23 and Lucarelli 2023 (diabetic): L23 (Dia).

Metric	Friedman	QA vs L23	QA vs L23 (Dia)	L23 vs L23 (Dia)
Precision	5.60×10^{-5}	8.5×10^{-5}	6.38×10^{-2}	0.13
Recall	3.47×10^{-5}	8.5×10^{-5}	0.13	6.38×10^{-2}
F1	5.60×10^{-5}	8.5×10^{-5}	6.38×10^{-2}	0.13

Supplementary Figures

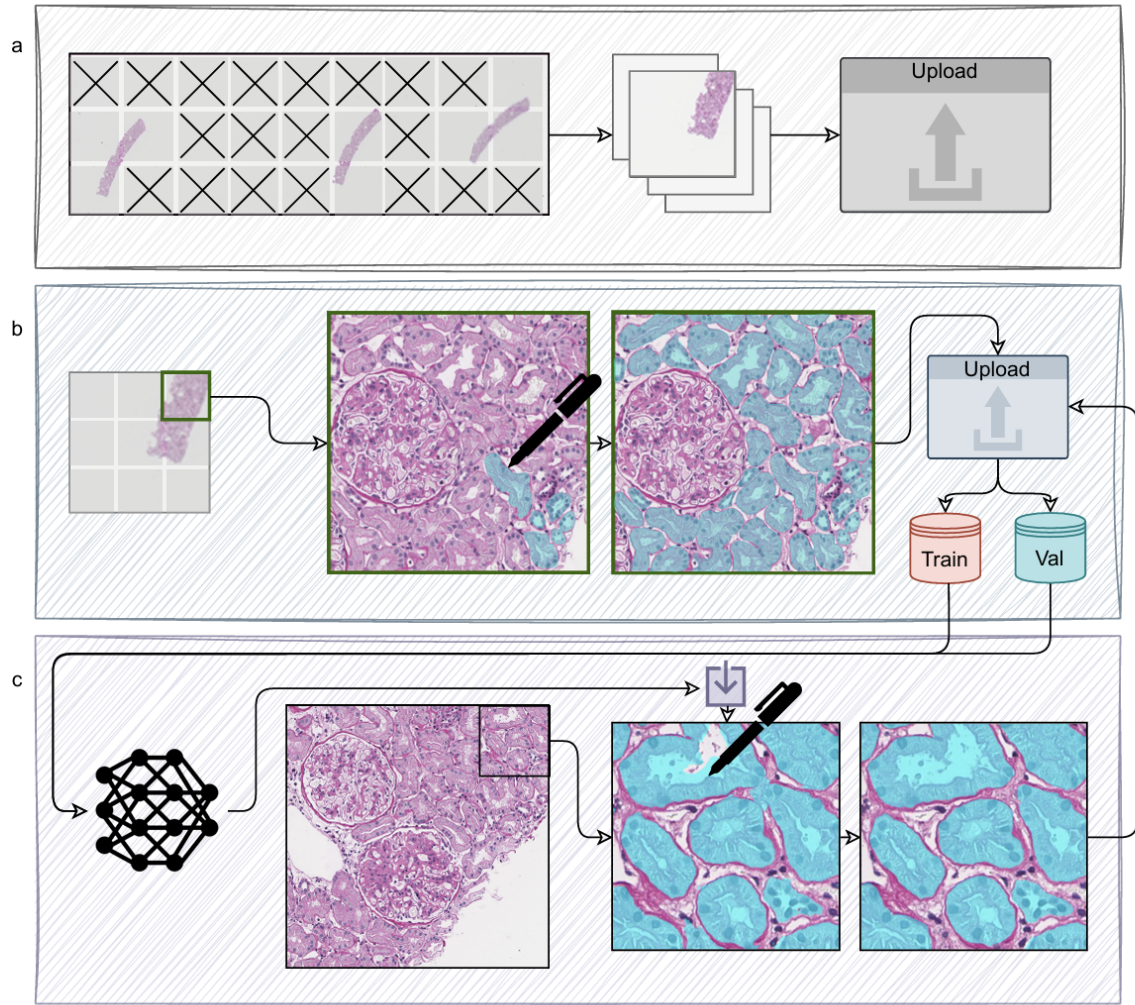


Figure S1: Illustration of the QA workflow used in this project. a) WSIs are tiled and uploaded to the QA interface, with background tiles automatically excluded. b) ROIs are annotated and the resulting data are uploaded to the training or validation dataset. c) Once a sufficient number of ROIs have been annotated (at time-dependent intervals in this study), a model can be trained. Model predictions can then be imported for each individual ROI, reviewed, and corrected if necessary. The fully annotated ROIs are subsequently added to the training or validation datasets. Steps (b) and (c) can be repeated for multiple iterations to refine the model, while step (a) can be repeated whenever a new WSI should be added.

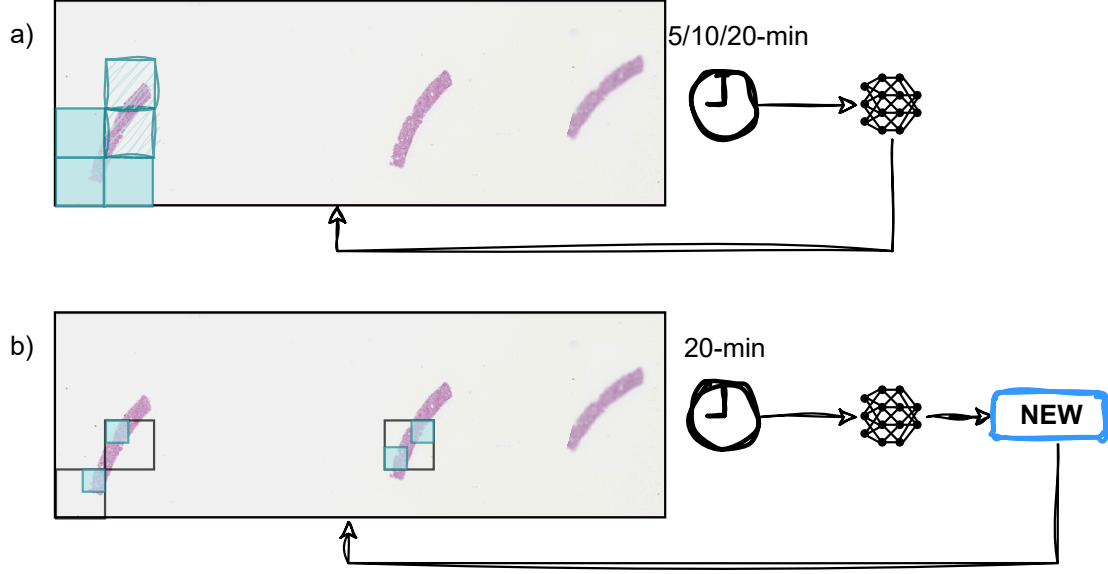


Figure S2: a) Training procedure for the initial models with fixed annotation intervals, using morphologically normal tubules. All tissue-containing tiles and all tissue-containing ROIs within each tile were fully annotated; once one WSI was completed, the next was added. b) Training procedure for incorporating data from diseased tubules into the final model, which had already been trained on two fully annotated WSIs of normal tubules. During incorporation of diseased tubules, each WSI was annotated for only one 20-minute interval, focusing on selected tiles and ROIs representing challenging and diverse regions.

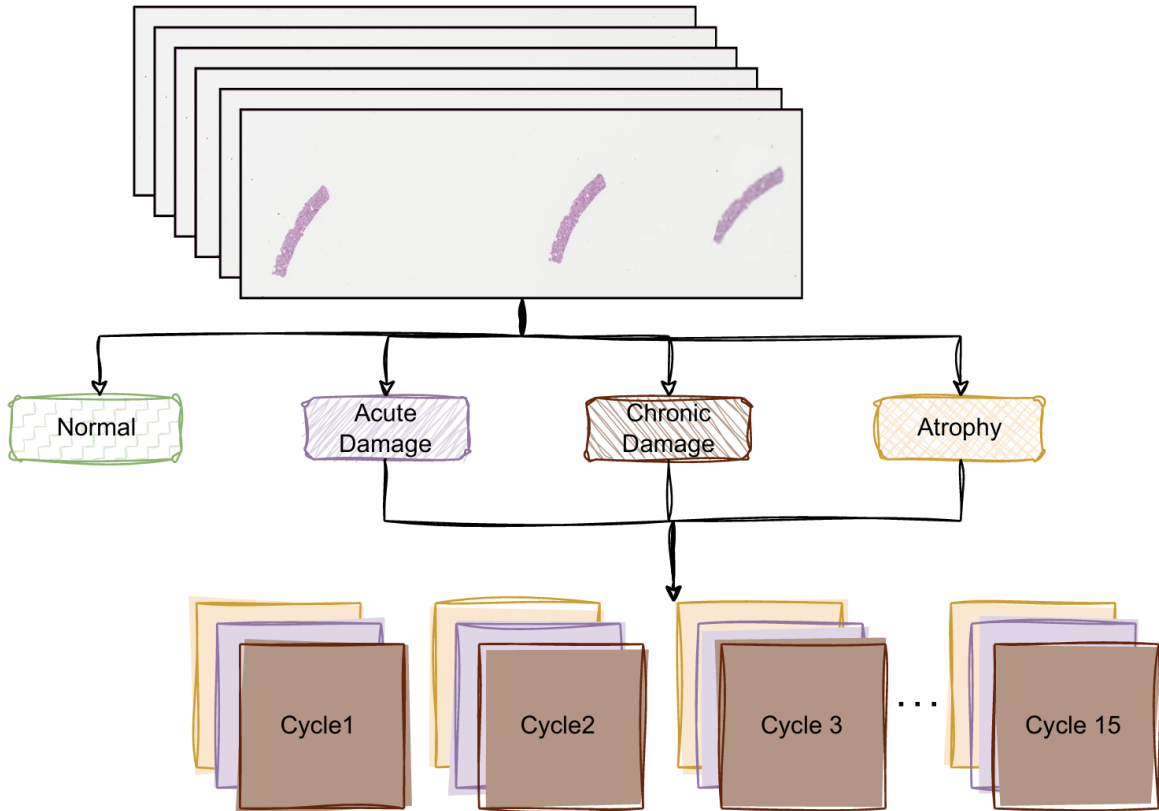


Figure S3: Procedure ensuring balanced inclusion of tubular lesions in the additional training data. WSIs were divided into four classes based on predominant tubule type. As the model had already been trained on two WSIs of normal tubules, additional normal WSIs were excluded. The remaining WSIs were organized into 15 annotation cycles, each including one WSI from each class, to maintain balanced representation across tubular lesion types as model refinement progressed.

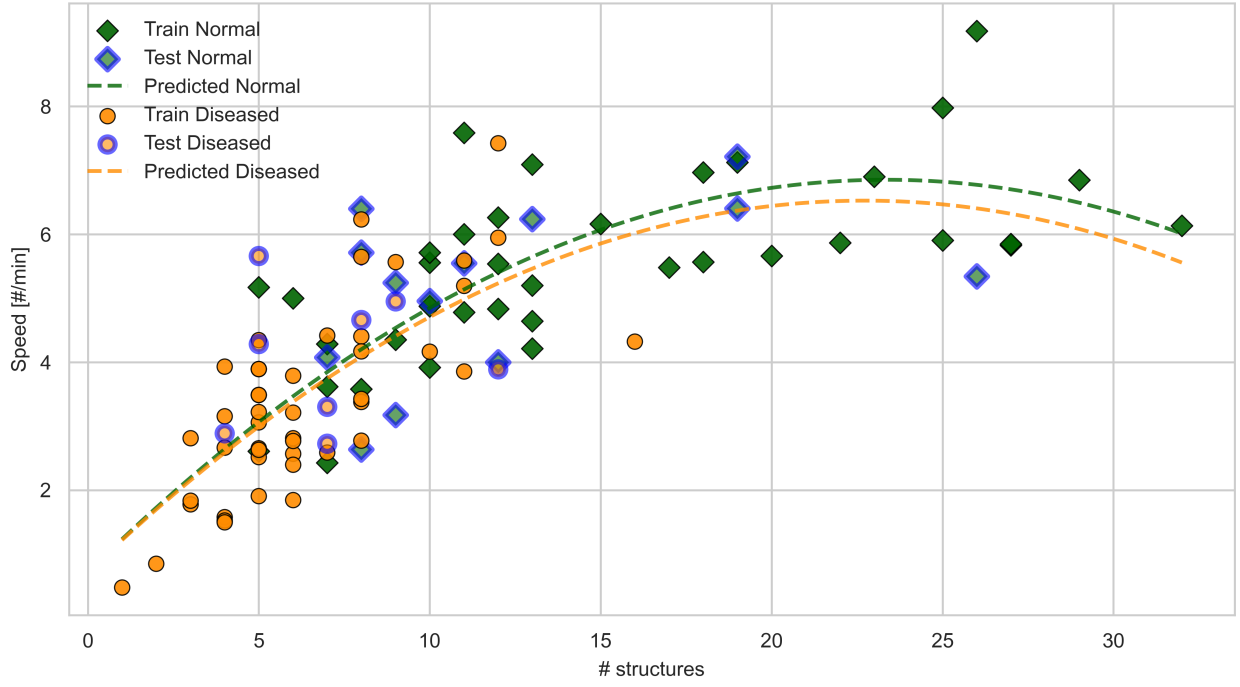


Figure S4: Prediction of manual annotation speed based on the mean number of structures per ROI and tubule state (normal vs. diseased) using a Lasso model (degree 2, $\alpha = 0.05$). Cross-validated performance: RMSE = 1.0111, MAE = 0.8368, $R^2 = 0.6386$. Test performance: RMSE = 1.1838, MAE = 0.9768, $R^2 = 0.1677$.

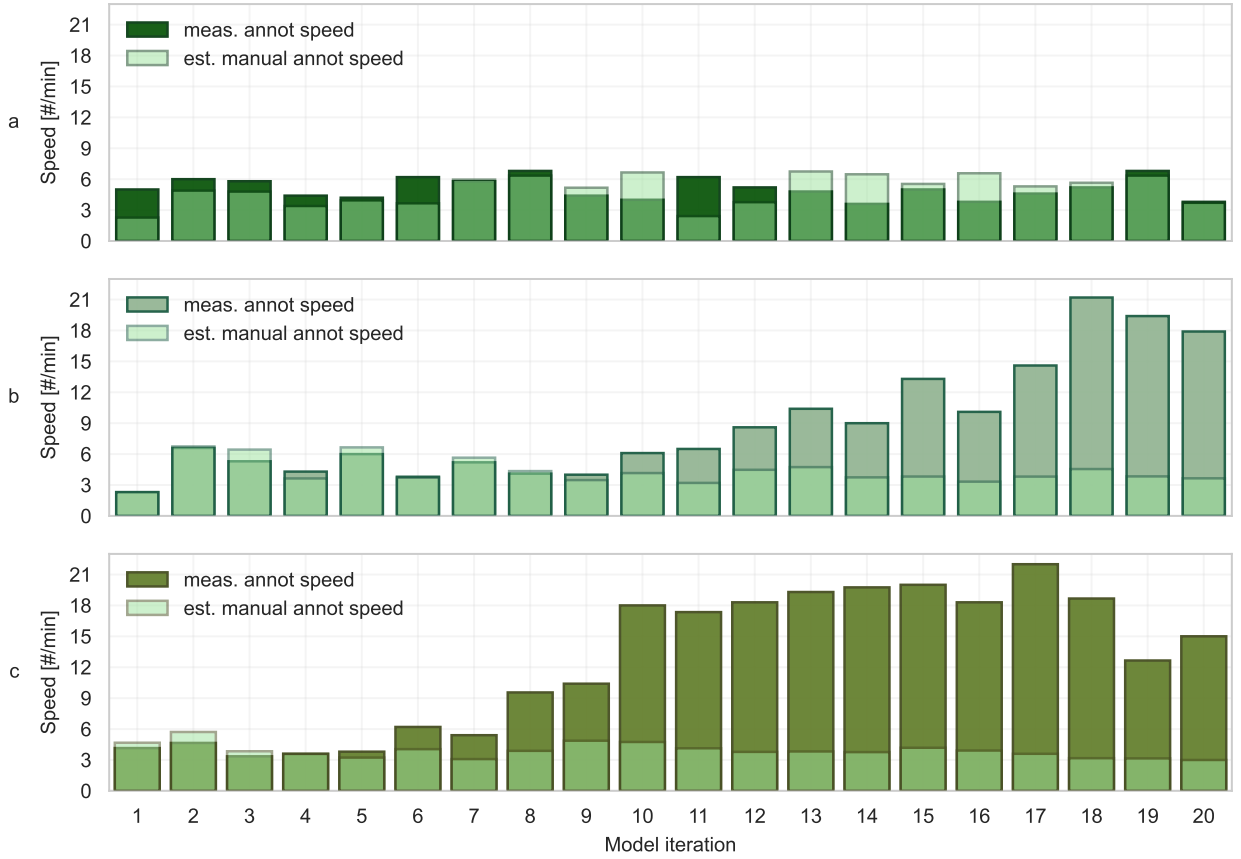


Figure S5: Comparison of observed and predicted annotation speeds across models of varying annotation duration. A. 5-minute model, B. 10-minute model, C. 20-minute model. A second-degree Lasso regression model, trained on manually annotated data, was used to estimate expected manual annotation speed per ROI. The figure illustrates the relationship between observed QA annotation speeds and the model-based predictions.

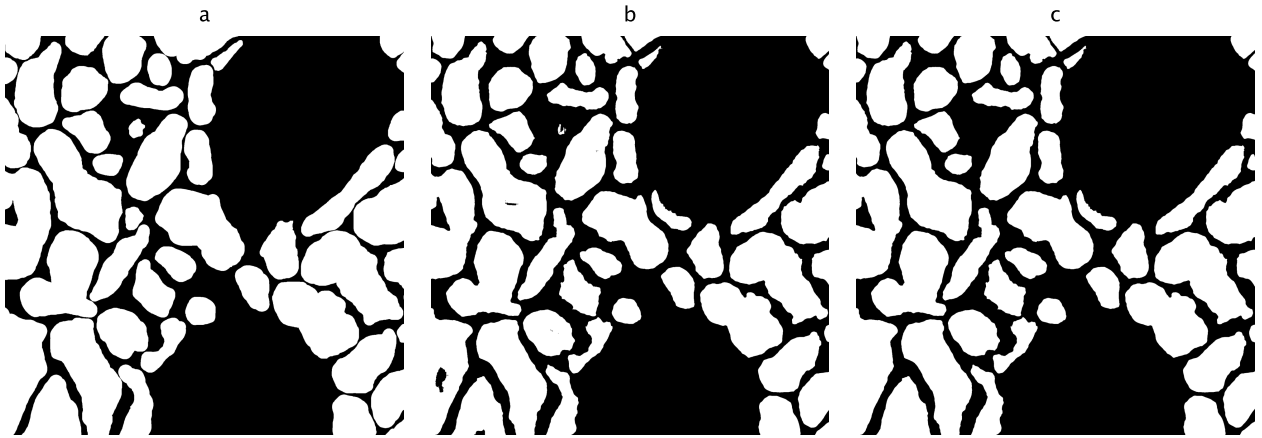


Figure S6: Example of post-processing effect on image tile with normal tubules, a) GT with 49 objects, b) QA raw prediction with 56 objects, c) QA post-processed prediction with 48 objects, where filtering and hole-filling is applied.

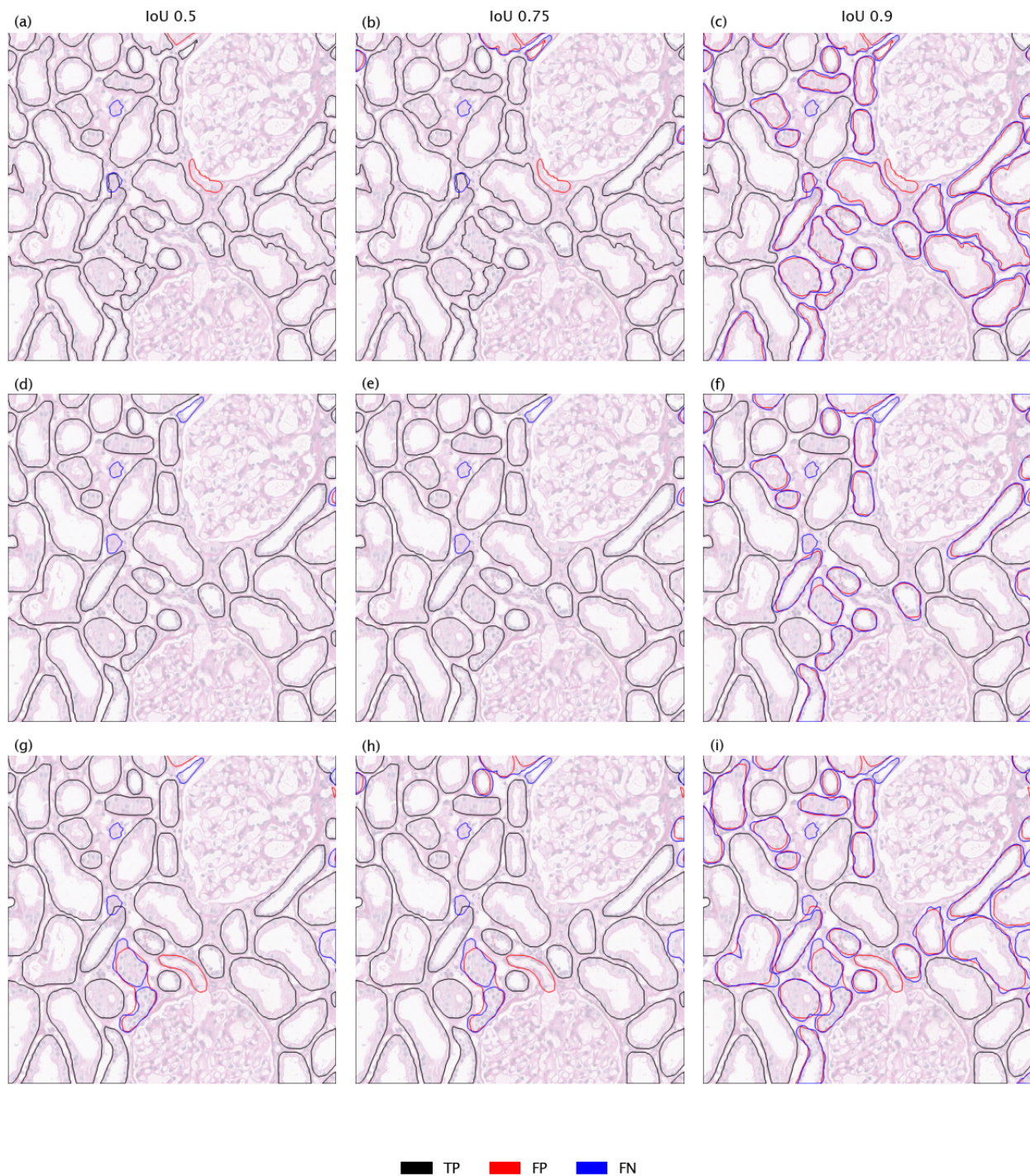


Figure S7: Image 1 (Class 0) predictions at three thresholds: (a–c) QA, (d–f) Lucarelli 2023, (g–i) Lucarelli 2023 diabetic.

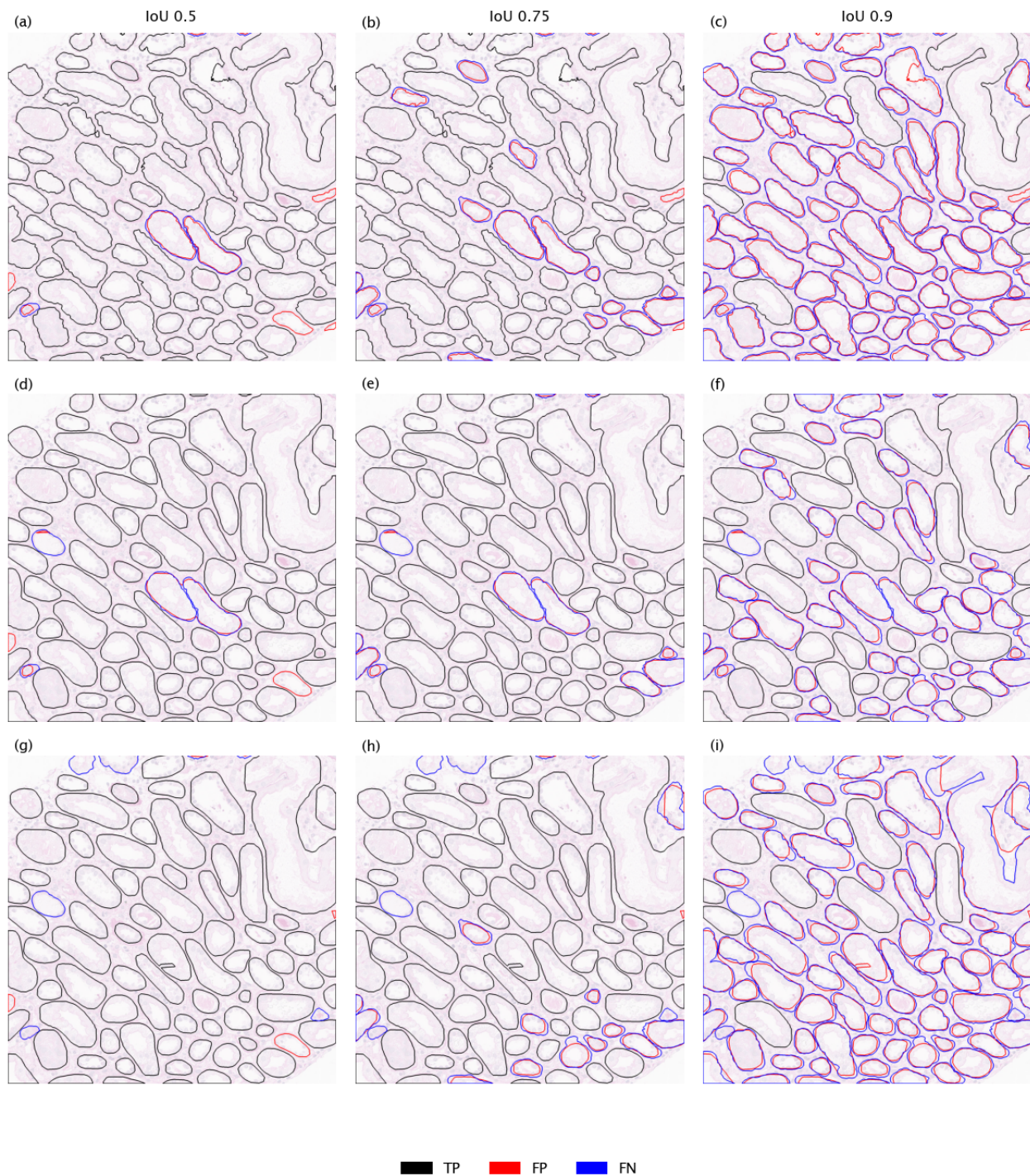


Figure S8: Image 2 (Class 0) predictions at three thresholds: (a-c) QA, (d-f) Lucarelli 2023, (g-i) Lucarelli 2023 diabetic.

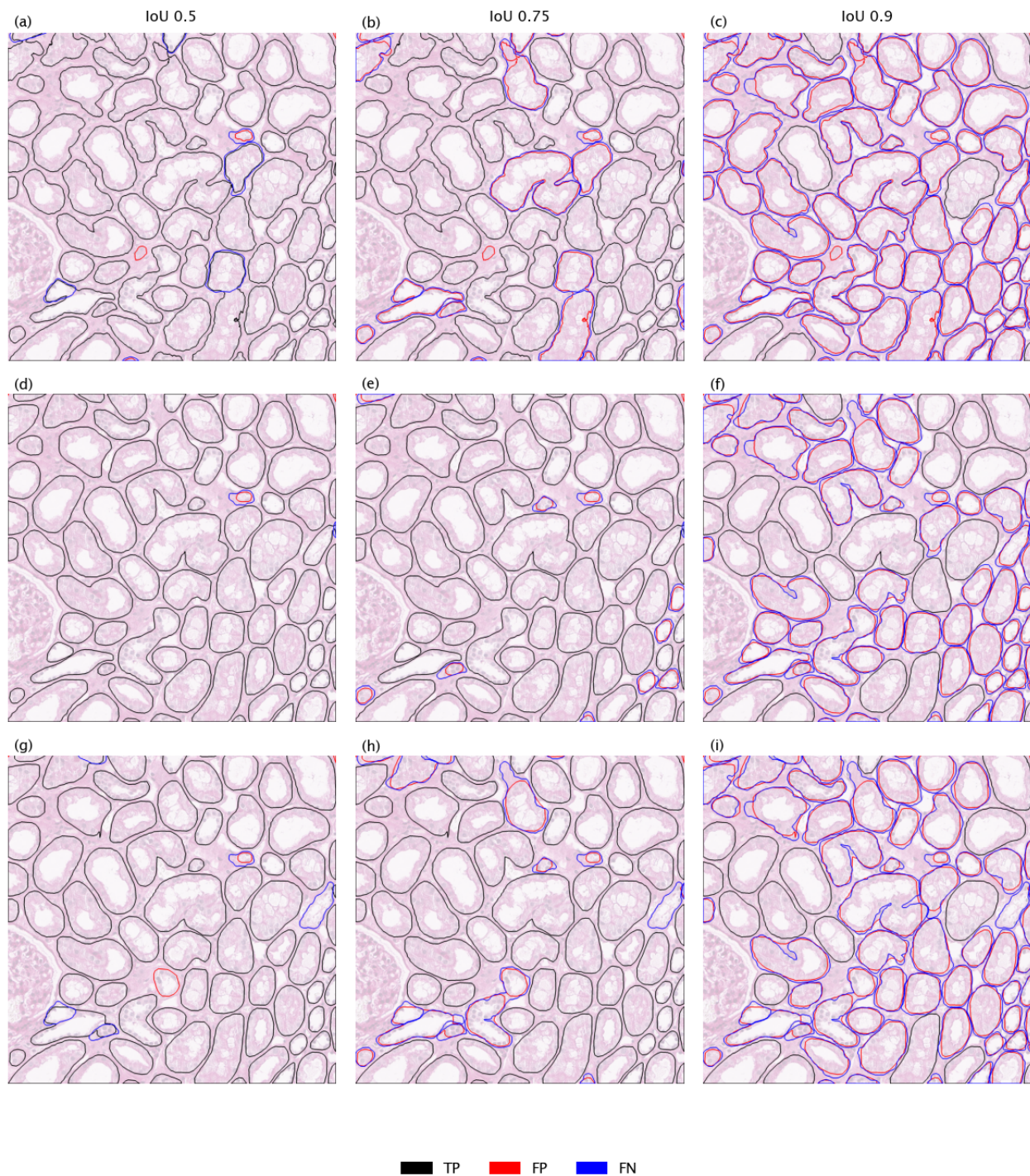


Figure S9: Image 3 (Class 0) predictions at three thresholds: (a–c) QA, (d–f) Lucarelli 2023, (g–i) Lucarelli 2023 diabetic.

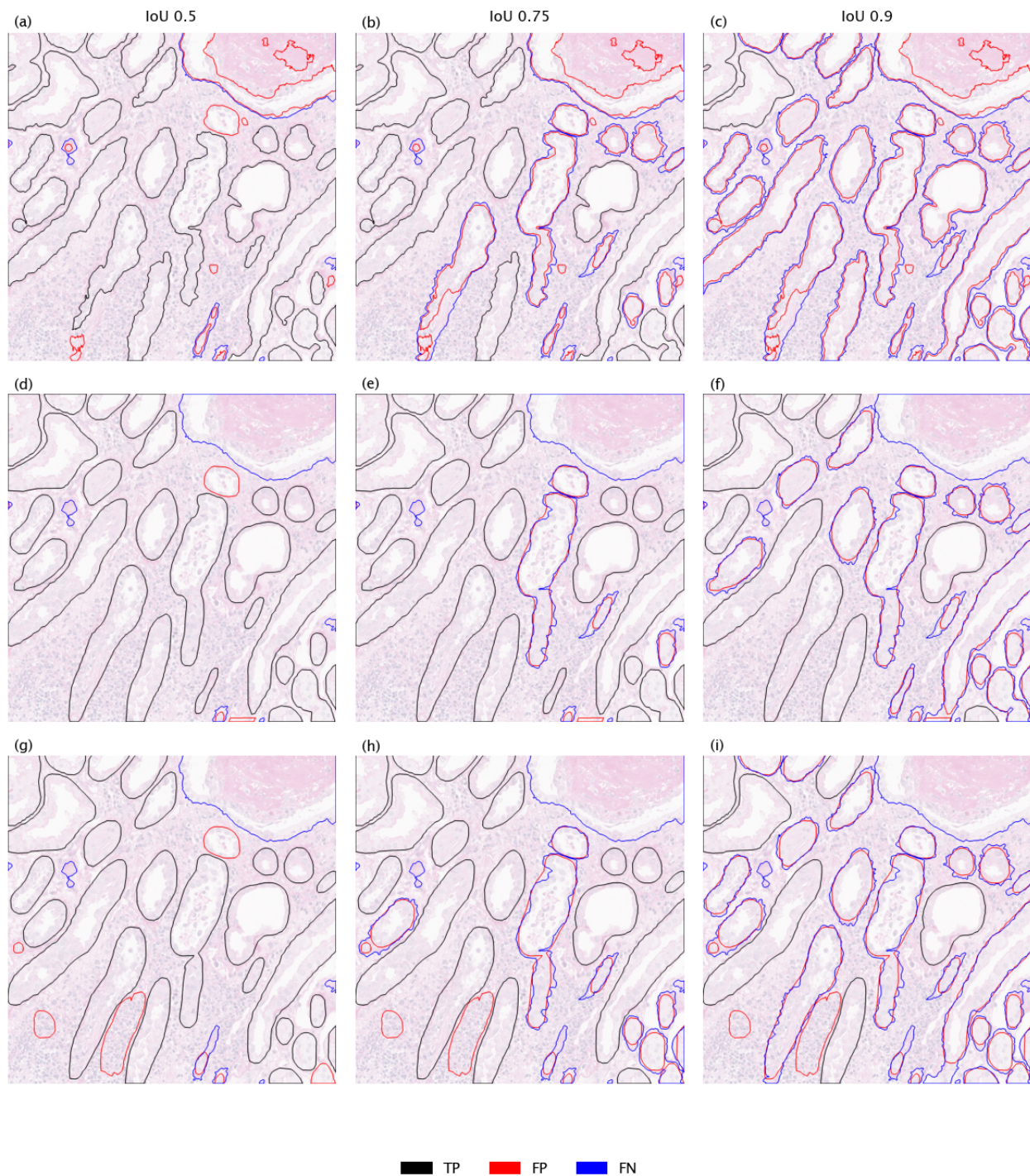


Figure S10: Image 4 (Class 1) predictions at three thresholds: (a-c) QA, (d-f) Lucarelli 2023, (g-i) Lucarelli 2023 diabetic.

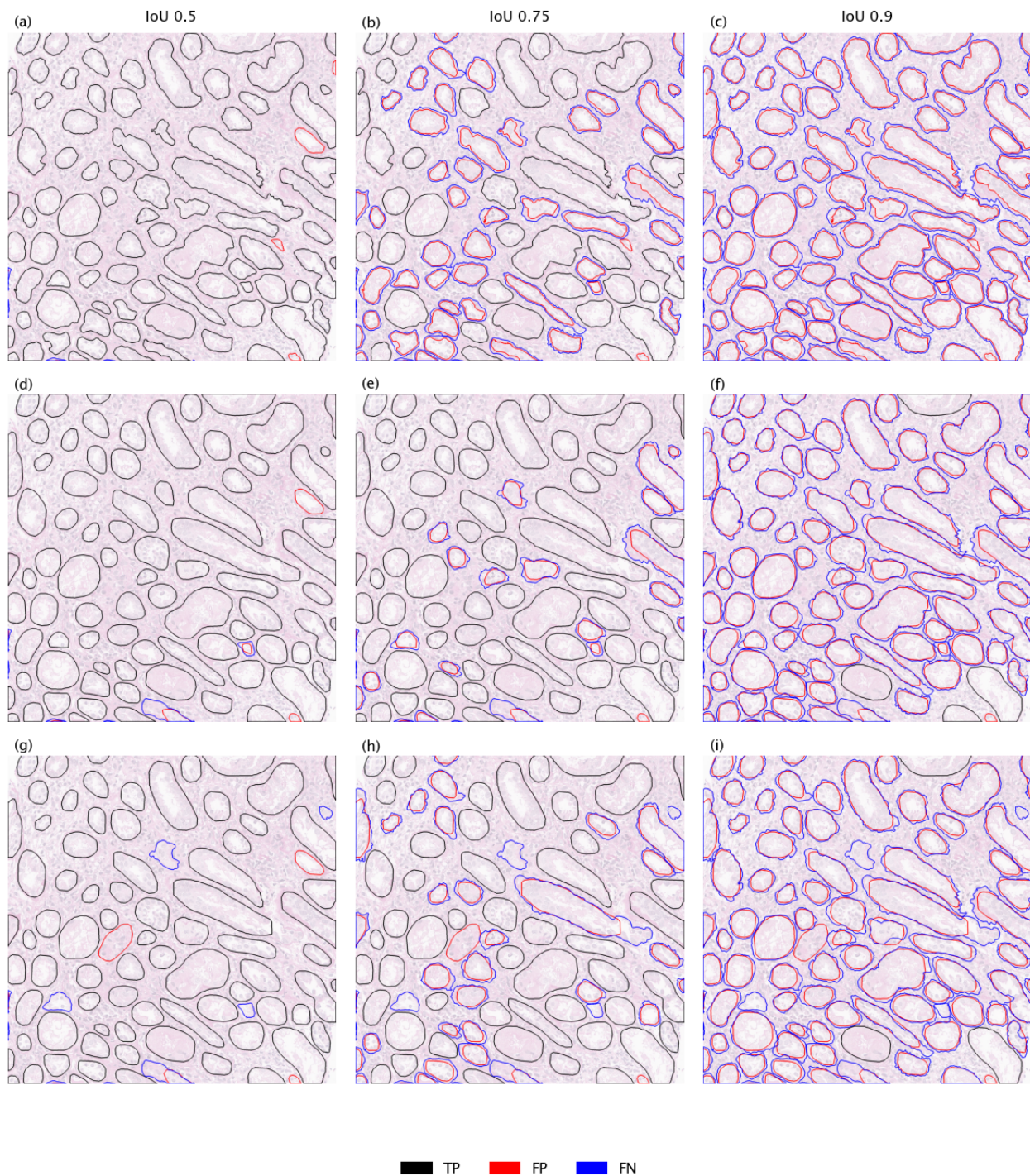


Figure S11: Image 5 (Class 1) predictions at three thresholds: (a–c) QA, (d–f) Lucarelli 2023, (g–i) Lucarelli 2023 diabetic.

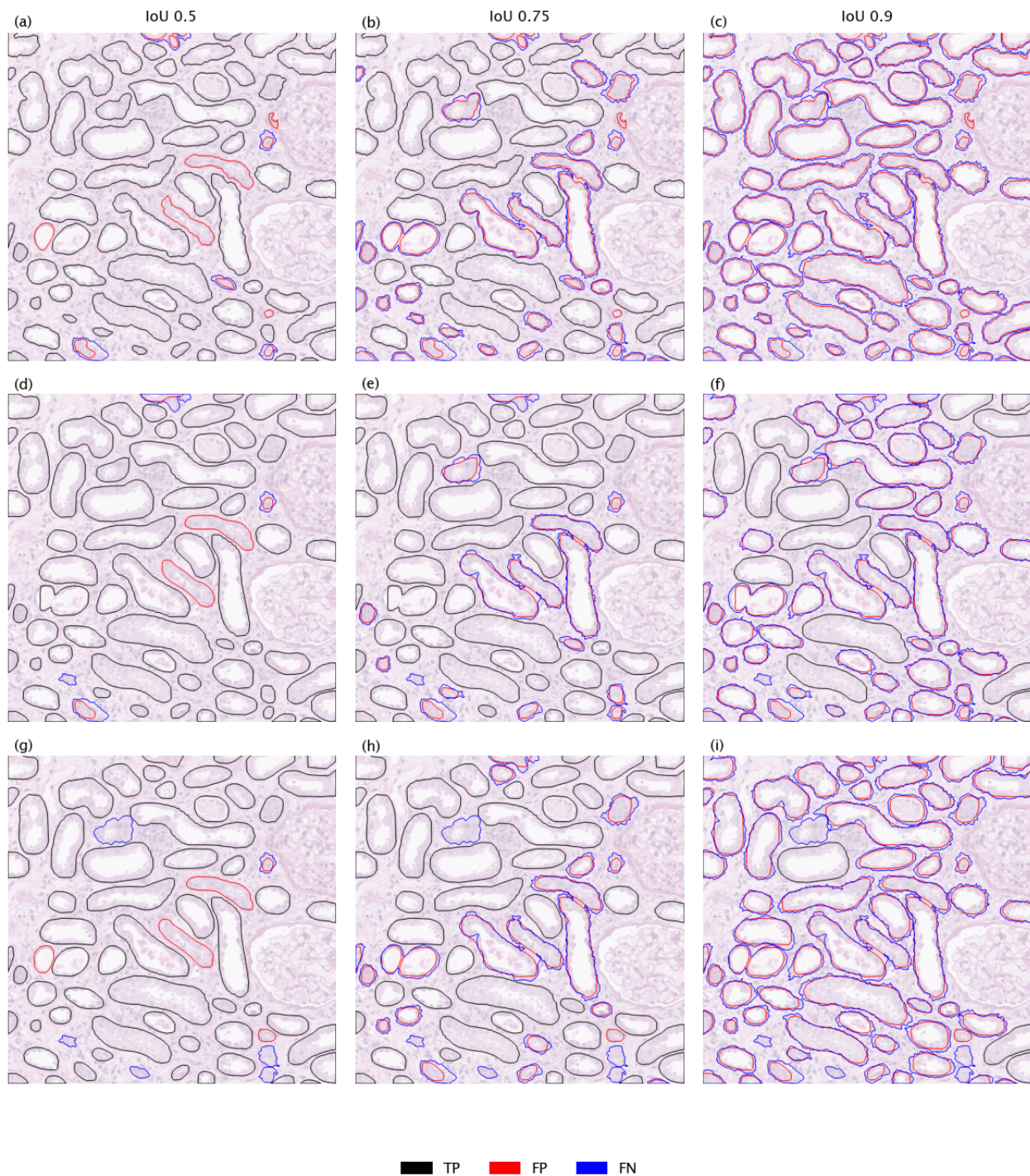


Figure S12: Image 6 (Class 1) predictions at three thresholds: (a-c) QA, (d-f) Lucarelli 2023, (g-i) Lucarelli 2023 diabetic.

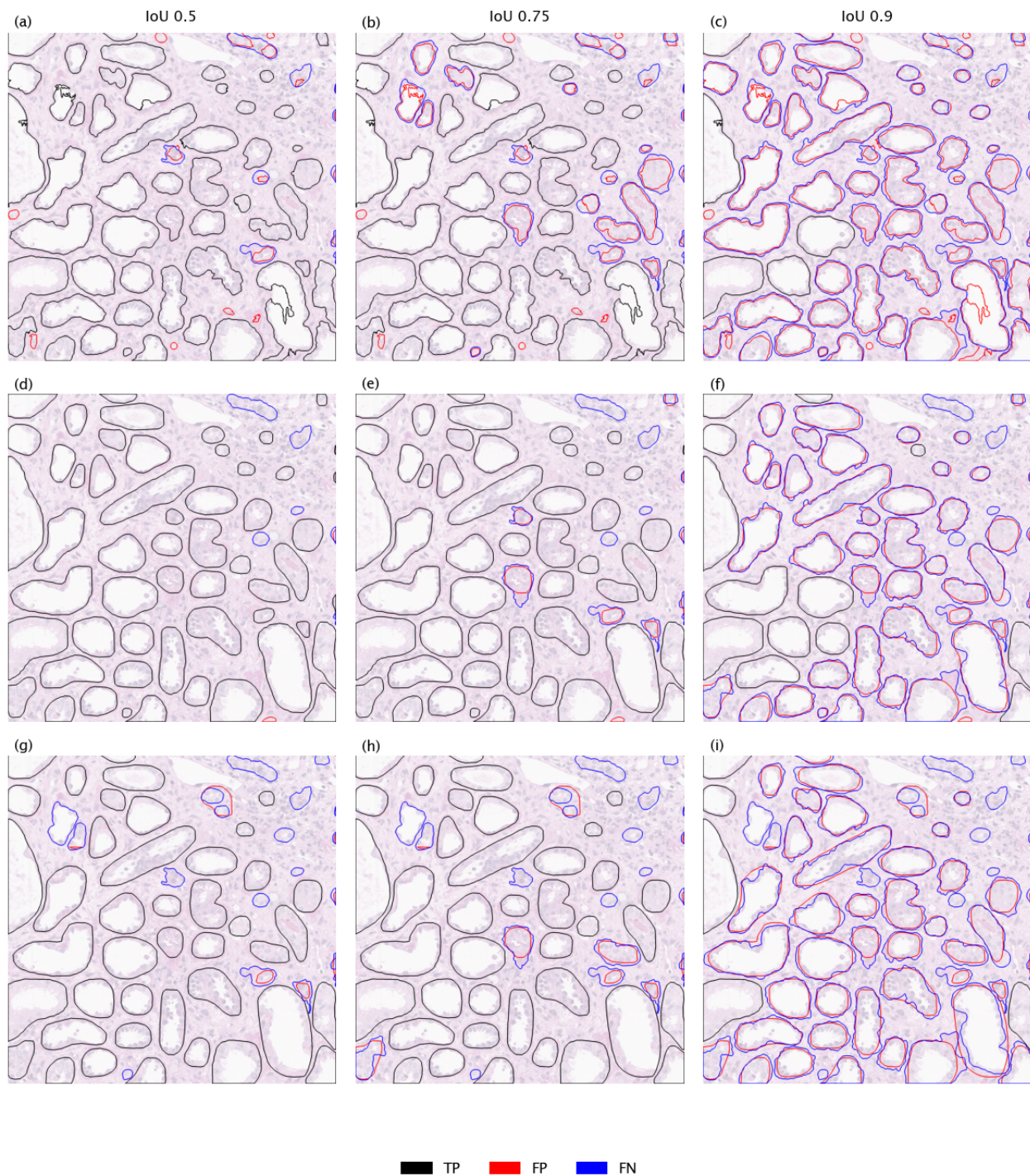


Figure S13: Image 7 (Class 2) predictions at three thresholds: (a-c) QA, (d-f) Lucarelli 2023, (g-i) Lucarelli 2023 diabetic.

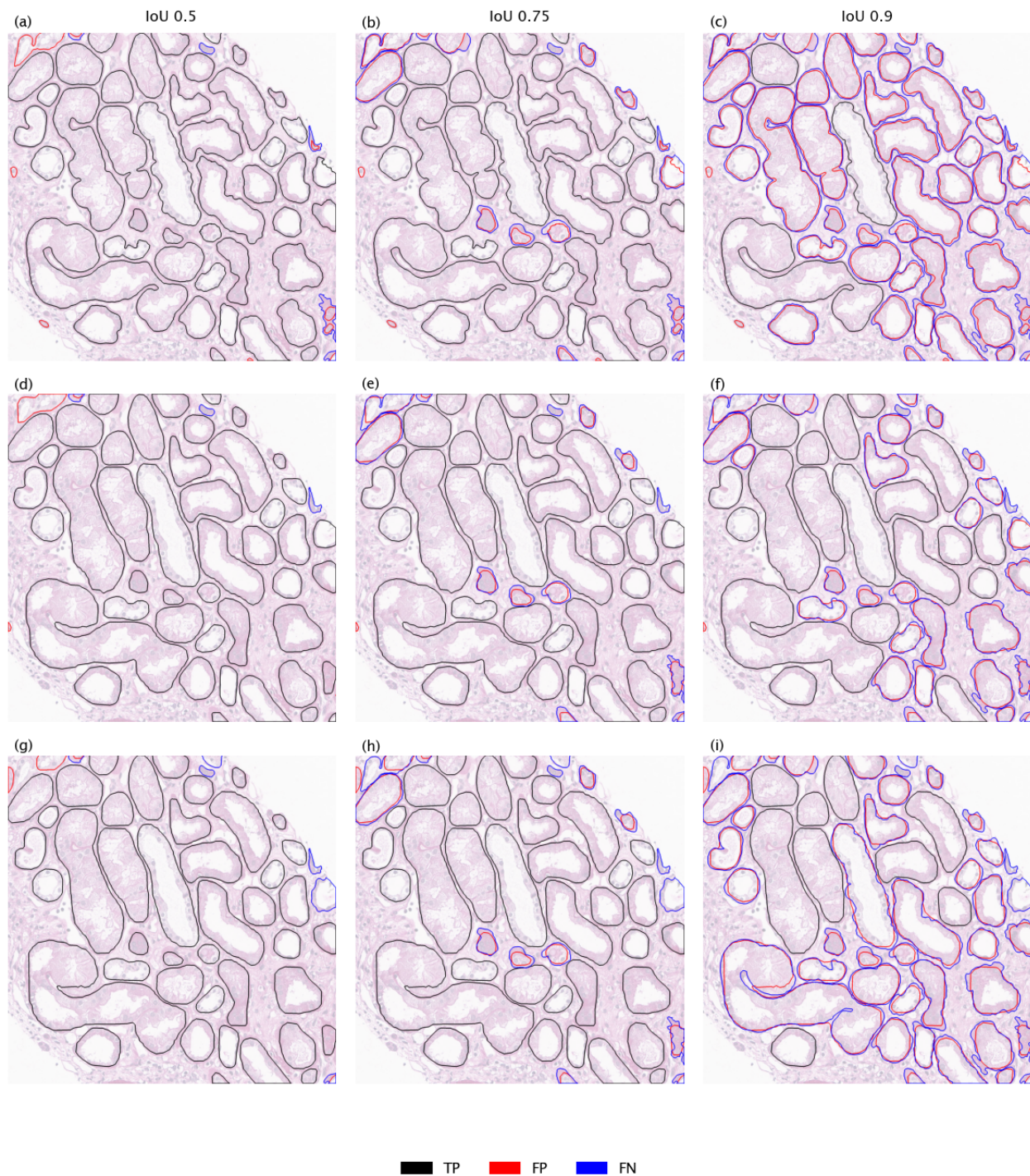


Figure S14: Image 8 (Class 2) predictions at three thresholds: (a-c) QA, (d-f) Lucarelli 2023, (g-i) Lucarelli 2023 diabetic.

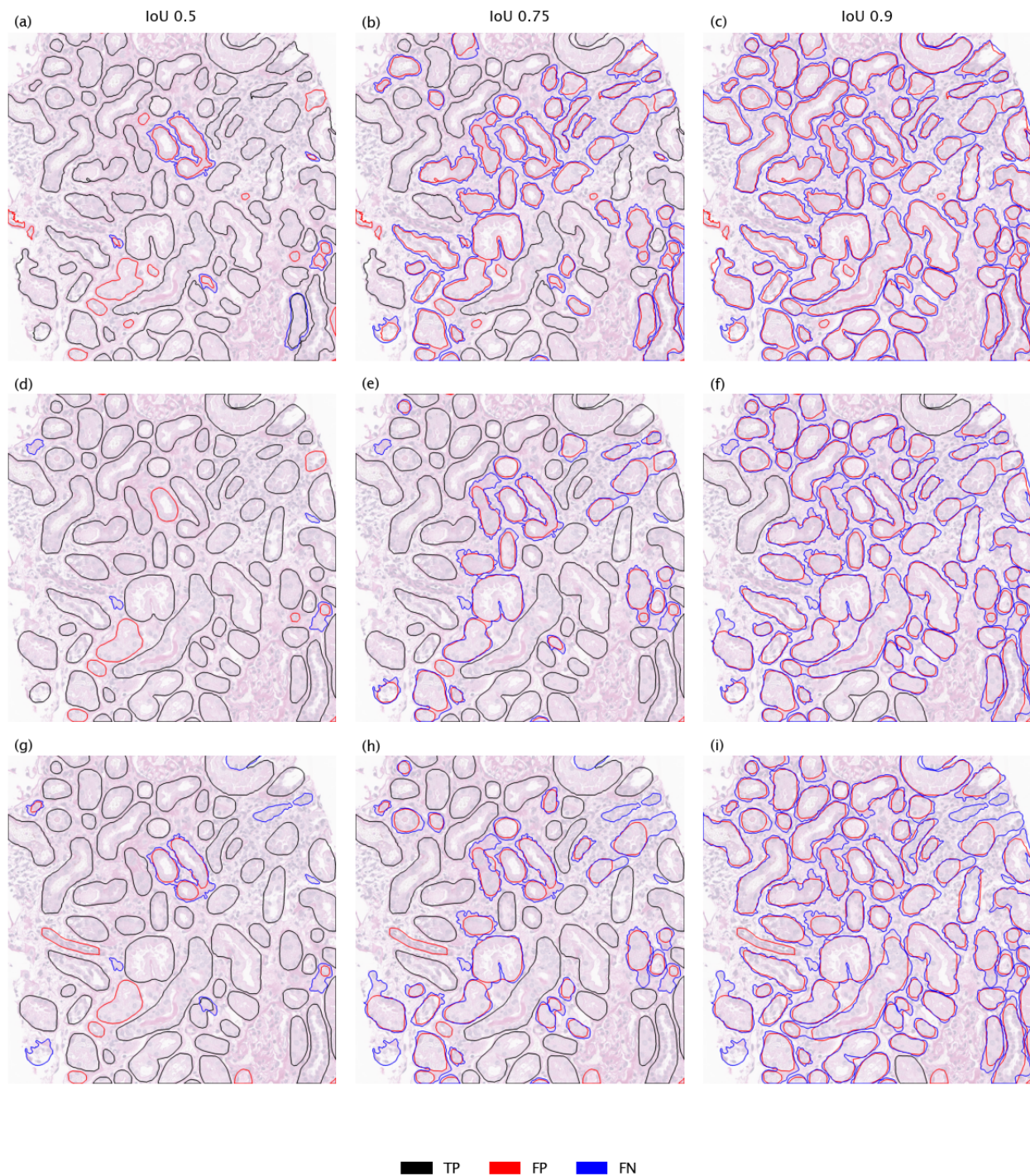


Figure S15: Image 9 (Class 2) predictions at three thresholds: (a-c) QA, (d-f) Lucarelli 2023, (g-i) Lucarelli 2023 diabetic.

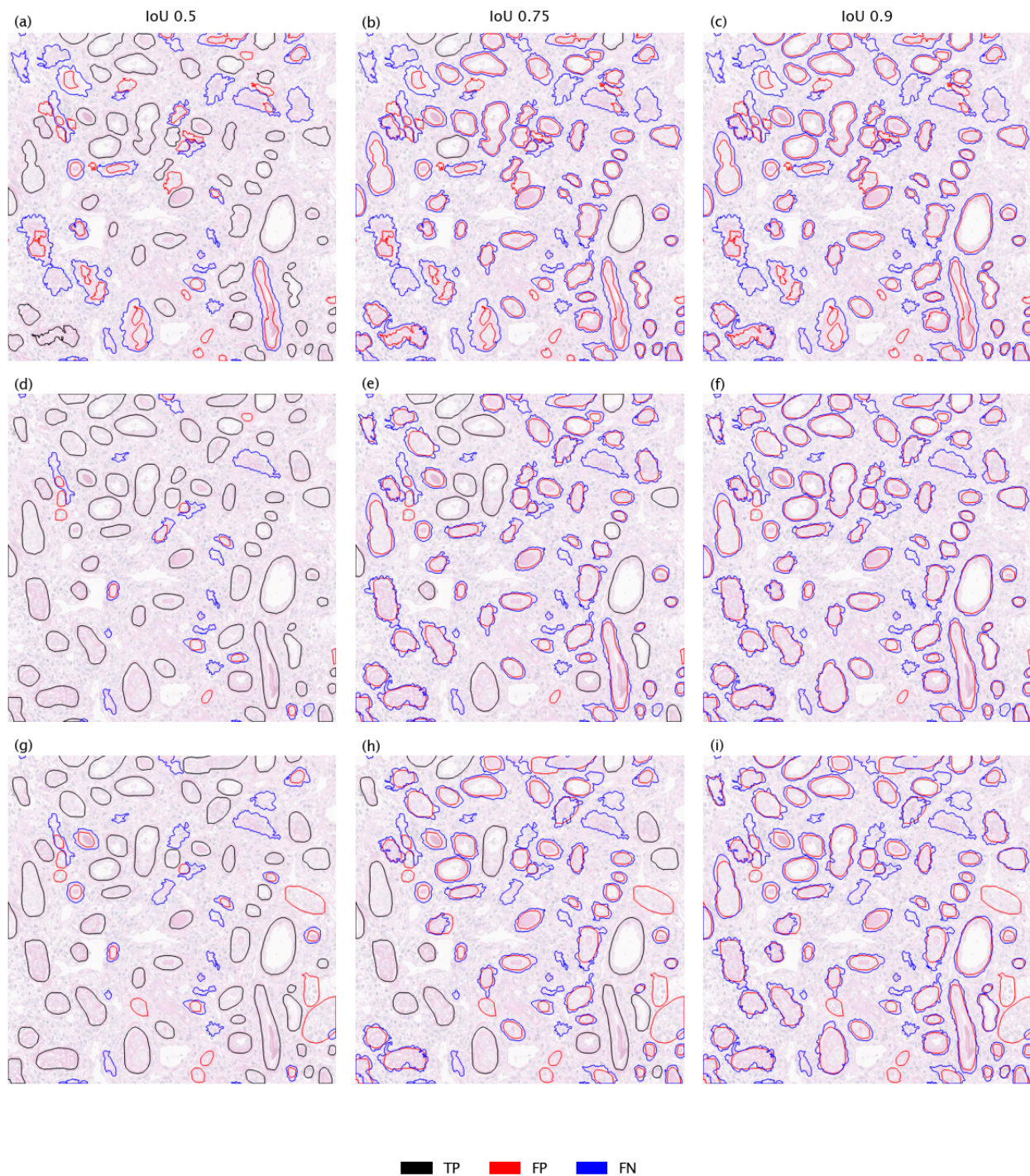


Figure S16: Image 10 (Class 3) predictions at three thresholds: (a–c) QA, (d–f) Lucarelli 2023, (g–i) Lucarelli 2023 diabetic.

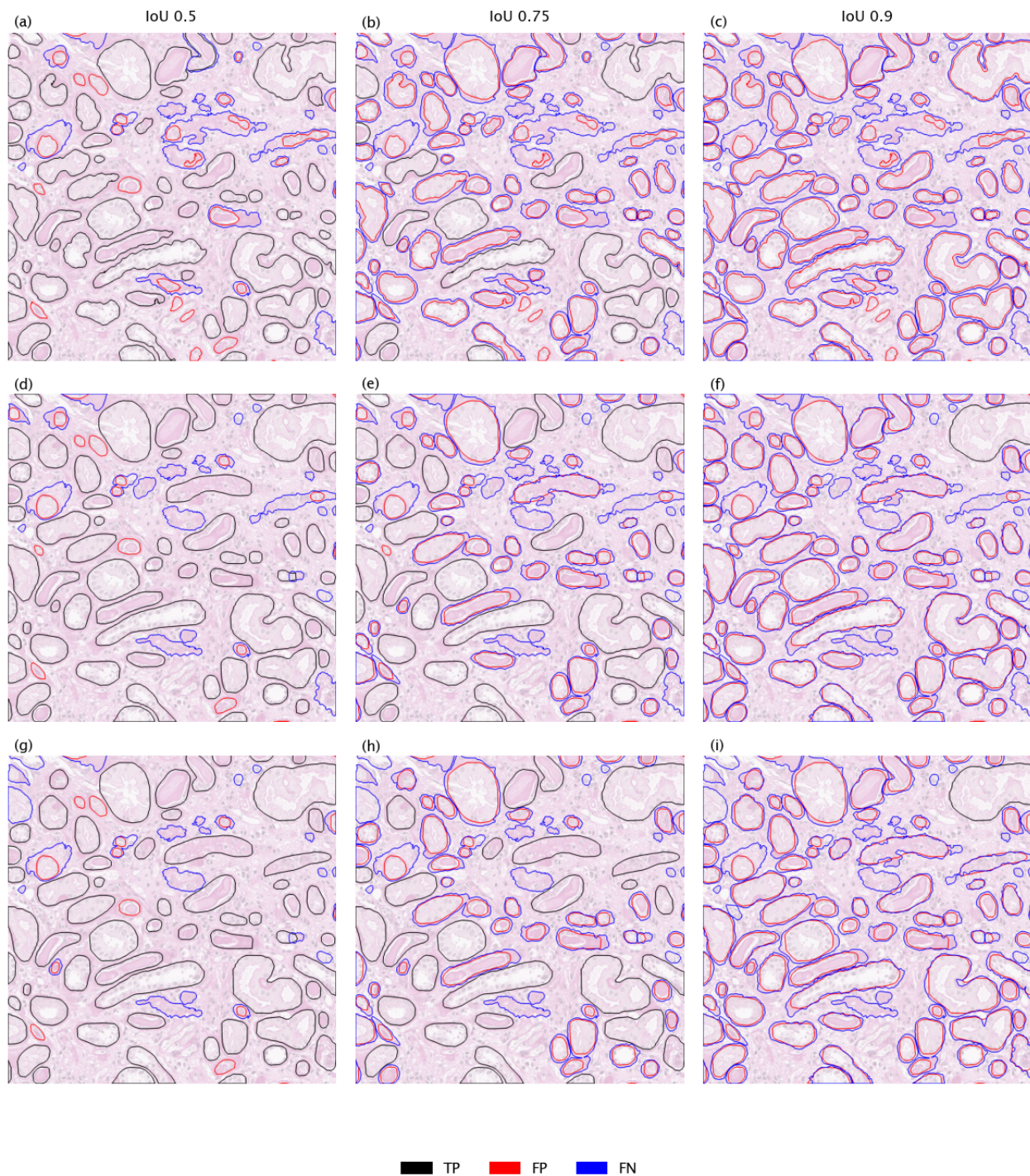


Figure S17: Image 11 (Class 3) predictions at three thresholds: (a-c) QA, (d-f) Lucarelli 2023, (g-i) Lucarelli 2023 diabetic.

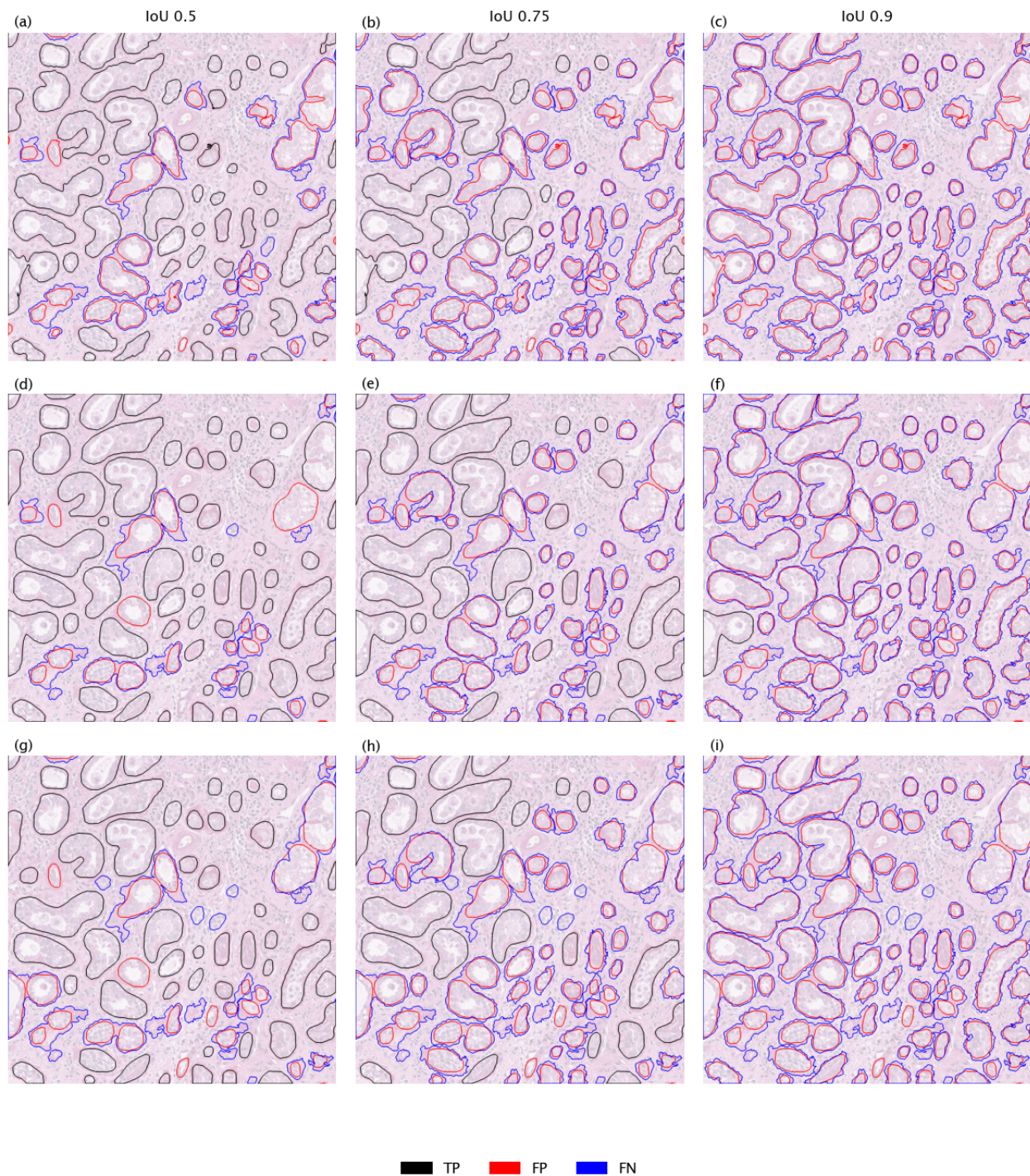


Figure S18: Image 12 (Class 3) predictions at three thresholds: (a–c) QA, (d–f) Lucarelli 2023, (g–i) Lucarelli 2023 diabetic.

References

- [1] Miao R, Toth R, Zhou Y, Madabhushi A, Janowczyk A: Quick Annotator: an open-source digital pathology based rapid image annotation tool. *The Journal of Pathology: Clinical Research*, DOI: 10.1002/cjp2.229, 2021
- [2] Bankhead P, Loughrey MB, Fernández JA, Dombrowski Y, McArt DG, Dunne PD, et al.: QuPath: Open source software for digital pathology image analysis. *Scientific Reports*, DOI: 10.1038/s41598-017-17204-5, 2017
- [3] Sugawara K: Training deep learning models for cell image segmentation with sparse annotations. *bioRxiv*, DOI: 10.1101/2023.06.13.544786, 2023
- [4] Lucarelli N, Ginley B, Zee J, Mimar S, Paul AS, Jain S, et al.: Correlating Deep Learning-Based Automated Reference Kidney Histomorphometry with Patient Demographics and Creatinine. *Kidney360*, DOI: 10.34067/kid.0000000000000299, 2023
- [5] Lucarelli N, Yun D, Han D, Ginley B, Moon KC, Rosenberg AZ, et al.: Discovery of Novel Digital Biomarkers for Type 2 Diabetic Nephropathy Classification via Integration of Urinary Proteomics and Pathology. *medRxiv*, DOI: 10.1101/2023.04.28.23289272, 2023
- [6] Lutnick B, Ginley B, Govind D, McGarry SD, LaViolette PS, Yacoub R, et al.: An integrated iterative annotation technique for easing neural network training in medical image analysis. *Nat Mach Intell*, DOI: 10.1038/s42256-019-0018-3, 2019