

Deep-Learning based segmentation and quantification in experimental kidney histopathology

Running Title: DL in experimental nephropathology

Nassim Bouteldja^{2,*}, Barbara M. Klinkhammer^{1,3,*}, Roman D. Bülow^{1,*}, Patrick Droste¹, Simon W. Otten¹, Saskia von Stillfried¹, Julia Moellmann⁴, Susan M. Sheehan⁵, Ron Korstanje⁵, Sylvia Menzel³, Peter Bankhead^{6,7}, Matthias Mietsch⁸, Charis Drummer⁹, Michael Lehrke⁴, Rafael Kramann^{3,10}, Jürgen Floege³, Peter Boor^{1,3,*,#}, Dorit Merhof^{2,11,*}

1 Institute of Pathology, RWTH Aachen University Hospital, Aachen, Germany

2 Institute of Imaging and Computer Vision, RWTH Aachen University, Aachen, Germany

3 Department of Nephrology and Immunology, RWTH Aachen University Hospital, Aachen, Germany

4 Department of Cardiology and Vascular Medicine, RWTH Aachen University Hospital, Aachen, Germany

5 The Jackson Laboratory, Bar Harbor, Maine

6 Edinburgh Pathology, University of Edinburgh, Edinburgh, UK

7 Institute of Genetics and Molecular Medicine, University of Edinburgh, Edinburgh, UK

8 Laboratory Animal Science Unit, German Primate Center, Goettingen, Germany

9 Platform Degenerative Diseases, German Primate Center, Goettingen, Germany

10 Department of Internal Medicine, Nephrology and Transplantation, Erasmus Medical Center, Rotterdam, The Netherlands

11 Fraunhofer Institute for Digital Medicine MEVIS, Bremen, Germany

Supplementary material

Content

Supp. Table 1. Glossary of technical terms.

Supp. Table 2. Criteria for definition of classes.

Supp. Table 3. Quantitative information on ground truth data.

Supp. Table 4. Architecture of our full CNN.

Supp. Table 5. Performance comparison of our model, its unmodified variant vanilla u-net, and state-of-the-art context-encoder.

Supp. Fig. 1. Annotation procedure

Supp. Fig. 2. Challenging morphology for manual and automated annotations.

Supp. Fig. 3. Segmentation on whole slide images of UUO, Alport and NTN kidneys.

Supp. Fig. 4. Quantitative segmentation performance in murine NTN and adenine kidneys.

Supp. Fig. 5. Automated segmentation in the medulla of murine kidney sections.

Supp. Fig. 6. Examples of missclassifications.

Supp. Fig. 7. Relation between amount of training data and detection performance.

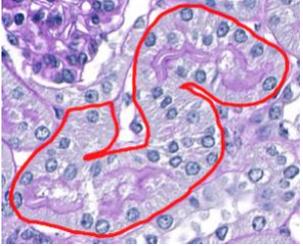
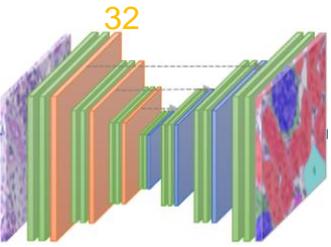
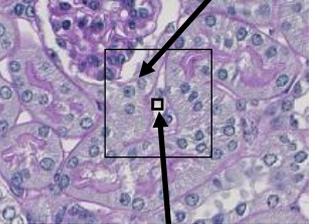
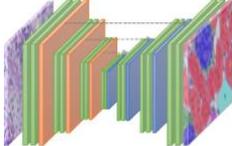
Supp. Fig. 8. Comparison between our full CNN and its variants independently trained on single models.

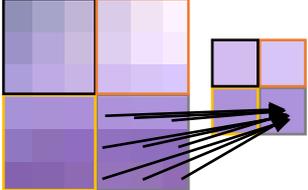
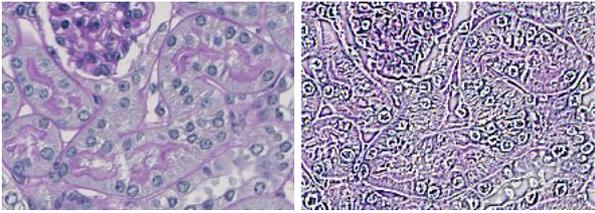
Supp. Fig. 9. Segmentation of non-trained and external murine kidney slides.

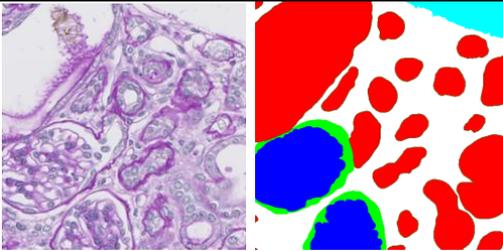
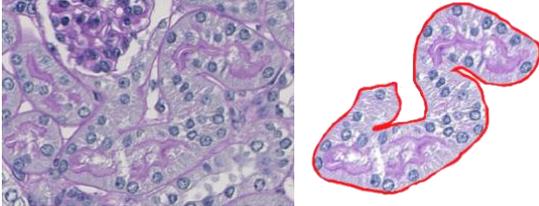
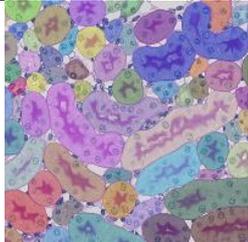
Supp. Fig. 10. Automated segmentation of renal medulla in different species.

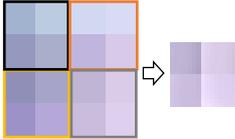
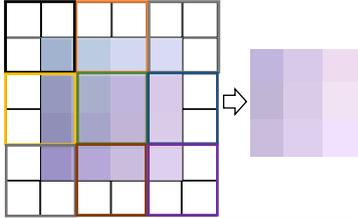
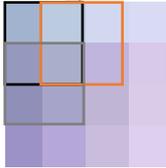
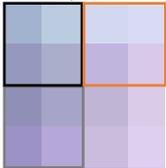
Supp. Fig. 11. Automated segmentation of human biopsies presenting with acute tubular damage.

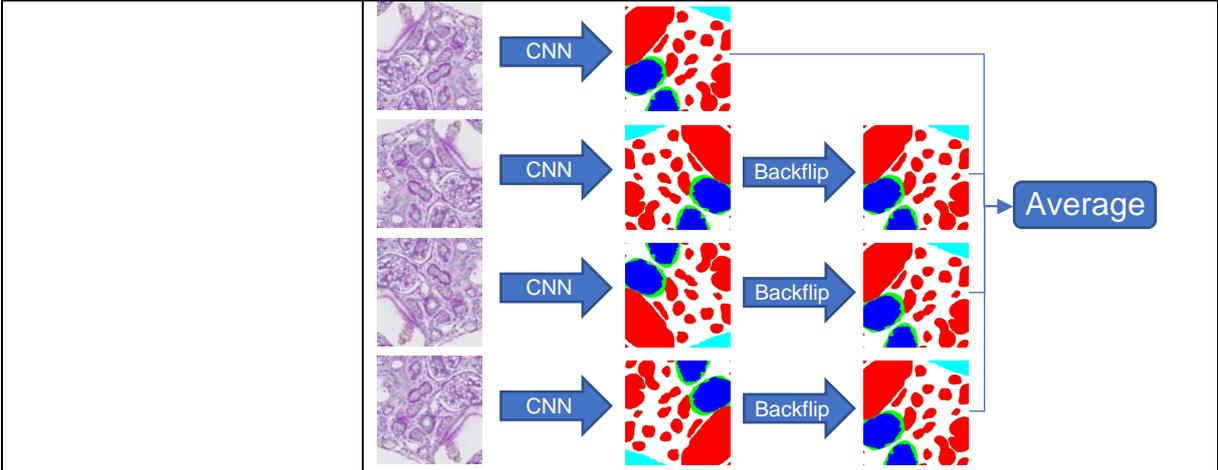
Supplementary Table 1. Glossary of technical terms.

Term	Description
Ablation study	<p>Experiment with consecutively reduced input data.</p> <p><u>In more detail:</u> A procedure where certain configurations of neural network architecture or training including modifications to data sets are changed to gain a better understanding of their importance and impact (mainly on overall performance).</p>
Border class	<p>->Class comprising borders of structures.</p> <p>Example: The tubule's border marked in red is assigned to the border class.</p> <p><u>In more detail:</u> Artificial class representing the border of specific structures. In our application, we make use of a border class, that especially represents the tubular basement membrane, to separate tubular (as well as glomerular or arterial) instances from each other, allowing for instance-level analysis.</p> 
Capacity	<p>Amount of ->parameters in a neural network.</p> <p><u>In more detail:</u> A neural network consists of many trainable parameters. Its number represents the network's capacity. It is also associated with its complexity, i.e. the degree of complexity of patterns the model is able to learn. Note that a neural network represents a mathematical function including input variables and parameters. Thus, the parameters are here defined in a mathematical way.</p>
Channel numbers	<p>Number of ->feature maps.</p> <p>Example: The channel number of the first, orange ->convolutional layer is 32.</p> <p><u>In more detail:</u> In convolutional neural networks, input data is subsequently propagated through ->convolutional layers each producing multiple output ->feature maps. Their number represents the channel number of the layer.</p> 
Class	<p>A group of structures.</p> <p>Example: All tubular structures belong to the "tubule"-class.</p>
Context-awareness	<p>Ability of a method to incorporate sufficient Context/neighborhood spatial neighborhood information for the assessment / prediction of a pixel.</p> <p><u>In more detail:</u> The more spatial context is considered for pixel prediction, the more context-aware is a technique. In our case, our network provides sufficient spatial context even for pixel prediction at patch border.</p> 
Convolutional layer	<p>Network layer performing convolutions to its input.</p> <p>Example: All green blocks represent such layers.</p> <p><u>In more detail:</u> Such layers represent substantial components in CNNs. Convolutions are performed on input data resulting in multiple ->feature maps. Convolutions are mainly specified based on the following ->parameters:</p> 

	<p>->kernel size, ->stride and ->padding. As exemplary shown on the right, a convolution (with 3x3 kernel size) slides over the image and outputs a single value for each 3x3 region.</p> 
Cross-entropy loss	<p>Information-theoretical measure of the dissimilarity between network output and ->ground truth. <u>In more detail:</u> A commonly used ->loss function when training segmentation or classification networks. The Cross-entropy loss (CE) is based on information theory and measures the difference between a target probability distribution (represented by ground truth annotations) and an estimated one (represented by model predictions). Its values range between 0 and 1. The smaller the loss, the higher the similarity. Thus, a perfect overlap results in a value of zero.</p>
Dice loss / Dice score	<p>The Dice score measures the similarity between network prediction and ->ground truth based on their spatial overlap. <u>In more detail:</u> The Dice score is a metric to quantify the similarity between two binary segmentations X and Y as follows: $DSC = \frac{2 X \cap Y }{ X + Y }$. In other words, it roughly quantifies the amount of spatial overlap between both segmentations. For multi-label evaluation, binary representations of ground truth and prediction are compared for each class. Besides, the Dice loss is represented by the Dice score in the following way: $DSC_{loss} = 1 - DSC$, since neural networks require ->loss functions instead of score functions.</p>
Ensembling	<p>->Regularization technique to improve performance. <u>In more detail:</u> Instead of one single learning algorithm, multiple neural networks are differently trained, and thus form different predictors to reduce prediction variance. Final results are performed by merging the predictions of all networks.</p>
Epoch	<p>An epoch ends when all training samples have been fed through the network once.</p>
Feature	<p>An individual, measurable property, e.g. glomerular size is a feature of the glomerulus.</p>
Feature map	<p>Spatially arranged features that are generated by applying filters to the convolutional layer input, i.e. the input image or feature map outputs from the prior layer. Example: A convolutional filter has been applied to the left image resulting in a two-dimensional feature map highlighting its edges.</p> 
Ground Truth	<p>Target data we expect the network to predict. We annotate and classify structures according to <i>our</i> renal ->class definitions in Supp. Table 2 and consider these annotations and classifications to correspond to reality, thus representing the ground truth. Example: Ground truth image of the left image is shown right.</p>

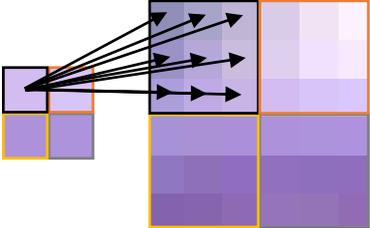
	
Hyperparameter	<p>Special ->parameters to control e.g. the learning process or architecture of the deep learning model. They are determined by the experimenter before as well as dynamically during training. Examples are the amount of ->epochs or the ->kernel size.</p>
Image segmentation	<p>Decomposition of an image into structures of interest. Example: Segmentation of a tubule.</p> 
Instance	<p>A single structure of a class. Example: All tubular instances are differently colored (Image from Supp. Fig. 5, third column).</p> 
Instance normalization	<p>->Regularization technique applied in neural networks. <u>In more detail:</u> In contrast to the widely used batch normalization, instance normalization normalizes each ->feature map independently providing zero mean and unit variance.</p>
Kernel size	<p>Specifies the size of a convolutional filter that is slid over the image.</p>
Loss function	<p>A mathematical function measuring the dissimilarity between network prediction and ->ground truth. <u>In more detail:</u> To train a neural network, a (differentiable) mathematical loss function representing a metric to measure the dissimilarity between prediction and ground-truth is required. During training, the network is consecutively optimized (with respect to the loss function) to lower the loss and thus to improve the similarity between prediction and ground-truth.</p>
Negative slope	<p>->Hyperparameter in the mathematical LeakyReLU function. <u>In more detail:</u> The LeakyReLU function is defined as follows: $\text{LeakyReLU}(x) = \begin{cases} x, & x \geq 0 \\ \text{negative_slope} * x, & \text{otherwise} \end{cases}$ Thus, the <i>negative_slope</i>-hyperparameter specifies the slope of the LeakyReLU function for negative inputs, i.e. $x < 0$. Most commonly, <i>negative_slope</i> = 0.01 is chosen by the experimenter.</p>
Padding	<p>An operation within convolutional layers to artificially enlarge the input data. <u>In more detail:</u> Specifies how much the input data is spatially padded around it. Padding an image with zeros exemplary means that zero values are added around it. Padding is used to counteract shrinkage of the input data caused by convolution.</p>

	<p>Example:</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p><u>without padding</u></p>  </div> <div style="text-align: center;"> <p><u>with padding</u></p>  </div> </div>
Parameter	<p>Components of a (deep learning) system that fully define and characterize the system.</p> <p><u>In more detail:</u> During network training, its trainable parameters are optimized. After training, all network parameters (trainable and non-trainable) are held constant, and the model is then used for prediction computation.</p>
Receptive field	<p>The prediction of a single output pixel only depends on a certain region of the input image. This region represents its receptive field. The size depends on the architecture of the network.</p>
Reduce-On-Plateau	<p>Technique to schedule the learning rate.</p> <p><u>In more detail:</u> The learning rate represents an important ->hyperparameter in neural networks that controls the speed of learning. This learning rate scheduler reduces the learning rate by a specific factor each time when the validation error has not decreased for a certain number of epochs.</p>
Regularization	<p>Regularization techniques are employed to improve network's generalization, i.e. reducing the error on test data. At the expense of increased training error, such techniques impose particularly designed constraints to the neural network preventing them to solely memorize the training data without having learned the underlying patterns.</p>
ReLU	<p>Stands for <i>rectified linear unit</i> and represents a mathematical function defined as follows: $ReLU(x) = \begin{cases} x, & x \geq 0 \\ 0, & otherwise \end{cases}$</p>
Robustness	<p>Describes the extent of input variability (e.g. in tissue morphology, staining, slide thickness, laboratory) an algorithm can cope with. Generally, it is measured by performance evaluation on those variabilities (usually held-out as in the current study).</p>
Stride	<p>An operation within convolutional layers to specify how many pixels the convolutional filter (or: ->kernel) is moved when slid over the image.</p> <p>Example:</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p><u>stride of "1"</u> (shift of 1 pixel)</p>  </div> <div style="text-align: center;"> <p><u>stride of "2"</u> (shift of 2 pixels).</p>  </div> </div>
Test-time augmentation	<p>->Regularization technique to improve performance.</p> <p><u>In more detail:</u> Regularization technique that forwards flipped versions of the input through the network and averages their respectively back-flipped predictions to yield the final prediction. In contrast to ->ensembling, just a single network/predictor is used to perform multiple estimations.</p>



Transposed convolutions

The conventional convolution provides a many-to-one relationship between input and output, since many input pixels are connected to a single value in the output. In contrast, transposed convolutions make use of a reversed pixel connectivity (in backward direction) providing a one-to-many relationship. Thus, it is designed for image -> **upsampling**.



Upsampling

Expansion or increase of the spatial resolution of an image.
In more detail: Upsampling can be exemplarily performed by pixel interpolation meaning that new pixel values can be estimated between pixels by using their neighborhood, e.g. by averaging neighboring pixels values (ultimately yielding a denser image grid). The picture in -> **transposed convolutions** exemplarily shows an upsampling of an artificial image.

Supplementary Table 2. Criteria for definition of classes.

Class	Criteria
Full glomerulus	<ul style="list-style-type: none">- annotation along Bowman's capsule- if cross section showed urinary (or vascular) pole, glomerulus was encircled in round/oval shape
Glomerular tuft	<ul style="list-style-type: none">- subclass of the full glomerulus class- annotation of glomerular tuft only (including podocytes)- for glomerular lesions: extracapillary proliferates (= crescents), parietal epithelial cells which migrated onto the tuft or tip lesions were not included
Tubule	<ul style="list-style-type: none">- annotation along, but excluding, the basement membrane
Artery	<ul style="list-style-type: none">- annotation of all arteries, including all arterial branches to arterioles- at least one visible vascular smooth muscle cell layer required
Arterial lumen	<ul style="list-style-type: none">- subclass of the artery class- annotation of lumen only, excluding also the endothelium
Vein	<ul style="list-style-type: none">- annotation of large "white" areas- only the lumen, i.e. the "white" area was annotated- for veins the definition of larger vessels next to arteries with a minimal diameter of 30µm- class includes non-tissue background and renal pelvis

Supplementary Table 3. Quantitative information on ground truth data.

Model / Species	Number of annotated patches / WSI	Train / val / test split of annotated patches	Train / val / test split of partially annotated WSI	Total number of instance annotations						Σ
				full glom.	glom. tuft	tubule	artery	arterial lumen	vein	
Healthy mouse	820 / 41	600 / 60 / 160	30 / 3 / 8	835	804	18536	1107	1416	609	23307
UUO	300 / 15	220 / 20 / 60	11 / 1 / 3	225	221	6795	301	314	177	8033
IRI	300 / 15	220 / 20 / 60	11 / 1 / 3	242	242	7555	354	397	102	8892
Adenine	300 / 15	220 / 20 / 60	11 / 1 / 3	257	256	5995	342	384	111	7345
Alport	300 / 15	220 / 20 / 60	11 / 1 / 3	413	368	7137	361	383	83	8745
NTN	300 / 15	220 / 20 / 60	11 / 1 / 3	247	237	5500	275	295	139	6693
db/db	30 / 3	0 / 0 / 30	0 / 0 / 3	27	27	652	27	22	10	765
Ext. UUO	30 / 3	0 / 0 / 30	0 / 0 / 3	46	43	879	42	27	8	1045
Human	230 / 12	200 / 0 / 30	10 / 0 / 2	123	148	1958	125	145	40	2539
Rat	80 / 8	50 / 0 / 30	5 / 0 / 3	56	59	1372	66	74	27	1654
Pig	80 / 6	50 / 0 / 30	5 / 0 / 1	50	49	900	57	67	23	1146
Marmoset	80 / 8	50 / 0 / 30	5 / 0 / 3	39	39	774	62	70	28	1012
Black bear	80 / 8	50 / 0 / 30	5 / 0 / 3	51	51	1240	85	91	28	1546
Σ	2930 / 164	2100 / 160 / 670	115 / 8 / 41	2611	2544	59293	3204	3685	1385	72722

IRI = ischemia reperfusion injury, NTN = nephrotoxic nephropathy, UUO = unilateral ureteral

obstruction, val = validation

Supplementary Table 4. Architecture of our CNN.

Network Architecture	Output size
Input image layer	640 x 640 x 3
Conv2d(i: 3, o: 32, k: 3, s: 1, p: 1) + IN(o: 32) + LeakyReLU(sl: 0.01)	640 x 640 x 32
Conv2d(i: 32, o: 32, k: 3, s: 1, p: 1) + IN(o: 32) + LeakyReLU(sl: 0.01)	640 x 640 x 32
MaxPool2d(k: 2, s: 2, p: 0)	320 x 320 x 32
Conv2d(i: 32, o: 64, k: 3, s: 1, p: 1) + IN(o: 64) + LeakyReLU(sl: 0.01)	320 x 320 x 64
Conv2d(i: 64, o: 64, k: 3, s: 1, p: 1) + IN(o: 64) + LeakyReLU(sl: 0.01)	320 x 320 x 64
MaxPool2d(k: 2, s: 2, p: 0)	160 x 160 x 64
Conv2d(i: 64, o: 128, k: 3, s: 1, p: 1) + IN(o: 128) + LeakyReLU(sl: 0.01)	160 x 160 x 128
Conv2d(i: 128, o: 128, k: 3, s: 1, p: 1) + IN(o: 128) + LeakyReLU(sl: 0.01)	160 x 160 x 128
MaxPool2d(k: 2, s: 2, p: 0)	80 x 80 x 128
Conv2d(i: 128, o: 256, k: 3, s: 1, p: 1) + IN(o: 256) + LeakyReLU(sl: 0.01)	80 x 80 x 256
Conv2d(i: 256, o: 256, k: 3, s: 1, p: 1) + IN(o: 256) + LeakyReLU(sl: 0.01)	80 x 80 x 256
MaxPool2d(k: 2, s: 2, p: 0)	40 x 40 x 256
Conv2d(i: 256, o: 512, k: 3, s: 1, p: 1) + IN(o: 512) + LeakyReLU(sl: 0.01)	40 x 40 x 512
Conv2d(i: 512, o: 512, k: 3, s: 1, p: 1) + IN(o: 512) + LeakyReLU(sl: 0.01)	40 x 40 x 512
MaxPool2d(k: 2, s: 2, p: 0)	20 x 20 x 512
Conv2d(i: 512, o: 1024, k: 3, s: 1, p: 1) + IN(o: 1024) + LeakyReLU(sl: 0.01)	20 x 20 x 1024
Conv2d(i: 1024, o: 1024, k: 3, s: 1, p: 1) + IN(o: 1024) + LeakyReLU(sl: 0.01)	20 x 20 x 1024
ConvTranspose2d(i: 1024, o: 1024, k: 2, s: 2)	40 x 40 x 1024
Conv2d(i: 1536, o: 512, k: 3, s: 1, p: 0) + IN(o: 512) + LeakyReLU(sl: 0.01)	38 x 38 x 512
Conv2d(i: 512, o: 512, k: 3, s: 1, p: 0) + IN(o: 512) + LeakyReLU(sl: 0.01)	36 x 36 x 512
ConvTranspose2d(i: 512, o: 512, k: 2, s: 2)	72 x 72 x 512
Conv2d(i: 768, o: 256, k: 3, s: 1, p: 0) + IN(o: 256) + LeakyReLU(sl: 0.01)	70 x 70 x 256
Conv2d(i: 256, o: 256, k: 3, s: 1, p: 0) + IN(o: 256) + LeakyReLU(sl: 0.01)	68 x 68 x 256
ConvTranspose2d(i: 256, o: 256, k: 2, s: 2)	136 x 136 x 256
Conv2d(i: 384, o: 128, k: 3, s: 1, p: 0) + IN(o: 128) + LeakyReLU(sl: 0.01)	134 x 134 x 128
Conv2d(i: 128, o: 128, k: 3, s: 1, p: 0) + IN(o: 128) + LeakyReLU(sl: 0.01)	132 x 132 x 128
ConvTranspose2d(i: 128, o: 128, k: 2, s: 2)	264 x 264 x 128
Conv2d(i: 192, o: 64, k: 3, s: 1, p: 0) + IN(o: 64) + LeakyReLU(sl: 0.01)	262 x 262 x 64
Conv2d(i: 64, o: 64, k: 3, s: 1, p: 0) + IN(o: 64) + LeakyReLU(sl: 0.01)	260 x 260 x 64
ConvTranspose2d(i: 64, o: 64, k: 2, s: 2)	520 x 520 x 64
Conv2d(i: 96, o: 32, k: 3, s: 1, p: 0) + IN(o: 32) + LeakyReLU(sl: 0.01)	518 x 518 x 32
Conv2d(i: 32, o: 32, k: 3, s: 1, p: 0) + IN(o: 32) + LeakyReLU(sl: 0.01)	516 x 516 x 32
Conv2d(i: 32, o: 8, k: 1, s: 1, p: 0)	516 x 516 x 8

Conv2d = two-dimensional convolutional layer, IN = instance normalization, i = #input layers, o =

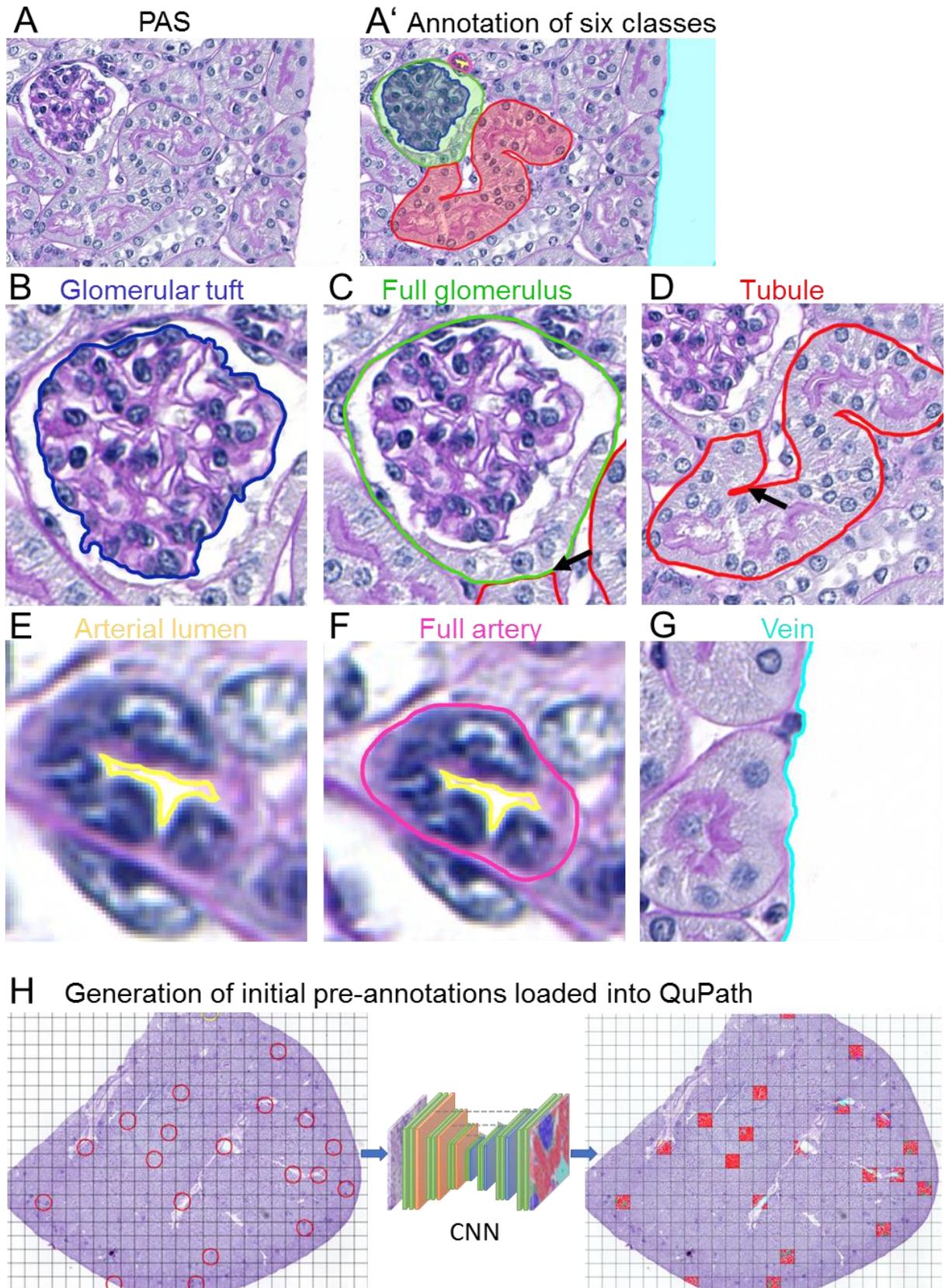
#output layers, k = kernel size, s = stride, p = padding, sl = negative slope

Supplementary Table 5. Performance comparison of our model, its unmodified variant vanilla u-net, and state-of-the-art context-encoder.

Shown are mean object-level dice scores for our model / the unmodified variant vanilla u-net / state-of-the-art context-encoder. The highest Score is marked in bold. * p < 0.05 vs. vanilla u-net and ° p < 0.05 vs. context-encoder.

Mouse	Segmentation performance of our model / vanilla u-net / context-encoder					
Model	full glomerulus	glomerular tuft	tubule	artery	arterial lumen	vein
Healthy	96.5 / 95.6 / 96.2	93.8 / 93.8 / 93.5	93.3 / 92.9 / 93.0	88.1 / 87.4 / 87.8	80.3 / 80.0 / 80.6	94.3 / 88.9 / 92.0
UUO	97.5 / 95.2 / 95.3	95.6 / 93.9 / 94.5	90.8 / 90.8 / 91.3	82.3 / 81.2 / 82.6	75.0 / 72.9 / 73.7	97.6 / 95.4 / 94.6
IRI	96.0 / 97.7 / 95.7	95.4 / 94.7 / 94.4	90.2 / 89.1 / 89.9	79.1 / 74.7 / 74.2	73.5 / 62.3 / 61.7	97.7 / 86.7 / 87.0
Adenine	98.8 / 94.1 / 98.5	97.2 / 94.1 / 97.1	93.0 / 92.0 / 92.8	87.9 / 83.3 / 83.2	80.9 / 72.7 / 76.9	93.6 / 87.6 / 96.7
Alport	94.7 / 95.5 / 96.3	91.3 / 86.4 / 87.6	90.6 / 89.7 / 89.3	80.3 / 74.2 / 72.0	81.1 / 69.9 / 65.5	89.2 / 83.2 / 81.7
NTN	95.5 / 91.5 / 96.3	94.8 / 93.9 / 93.9	93.2 / 92.5 / 92.9	86.8 / 82.7 / 83.9	78.2 / 73.9 / 79.1	92.8 / 91.8 / 95.4
∅	96.4* / 94.0 / 96.3	94.2* / 92.6 / 93.0	92.0* / 91.4 / 91.7	85.3* ° / 82.8 / 82.9	79.1* ° / 75.9 / 76.1	94.3* / 90.4 / 92.7

IRI = ischemia reperfusion injury, NTN = nephrotoxic nephropathy, UUO = unilateral ureteral obstruction

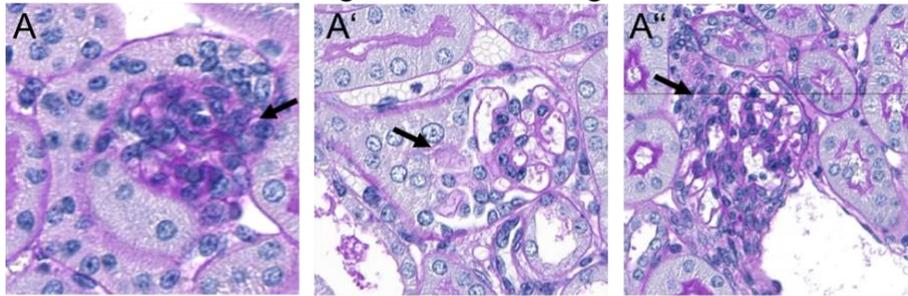


Supp. Fig. 1. Annotation procedure.

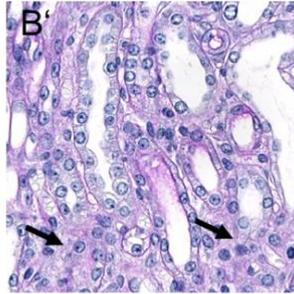
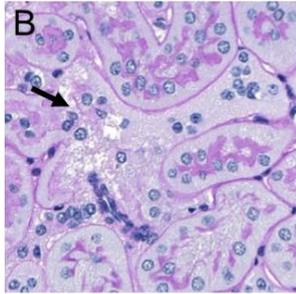
A representative picture of a PAS stained mouse kidney section (A) and an overlay with manual annotations for six classes (A'). The annotation of the "glomerular tuft" (blue (B)) included the capillary tuft, the mesangium and podocytes. A "full glomerulus" (green (C)) was annotated along bowman's capsule and included the tuft, bowman's

space and parietal epithelial cells. The glomerular tuft was always a subclass of the full glomerulus. A full glomerulus always had a round or oval shape, this determined the separation from the proximal tubule (arrow). Tubules (red (D)) were annotated along (but excluding) the tubular basement membrane, tangentially cut tubules without cytoplasm were excluded. The “arterial lumen” (yellow (D)) was always a subclass of the “artery” class (magenta (F)). Veins, background and renal pelvis were big “white” areas without tissue (cyan (G)). From the first manual annotations, we predicted initial pre-annotations for 20 patches per WSI and loaded them into Qupath for manual corrections facilitating annotation effort (H).

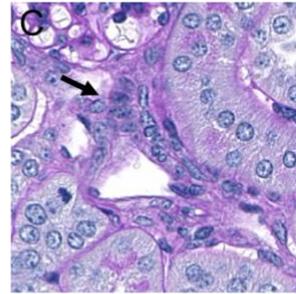
Blurred border of glomerulus and glomerular tuft



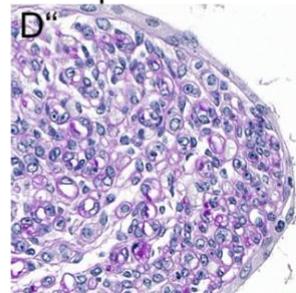
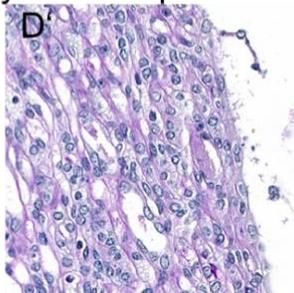
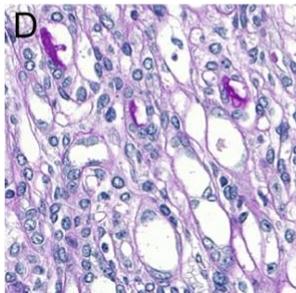
Blurred border of tubules



Arterial bifurcation

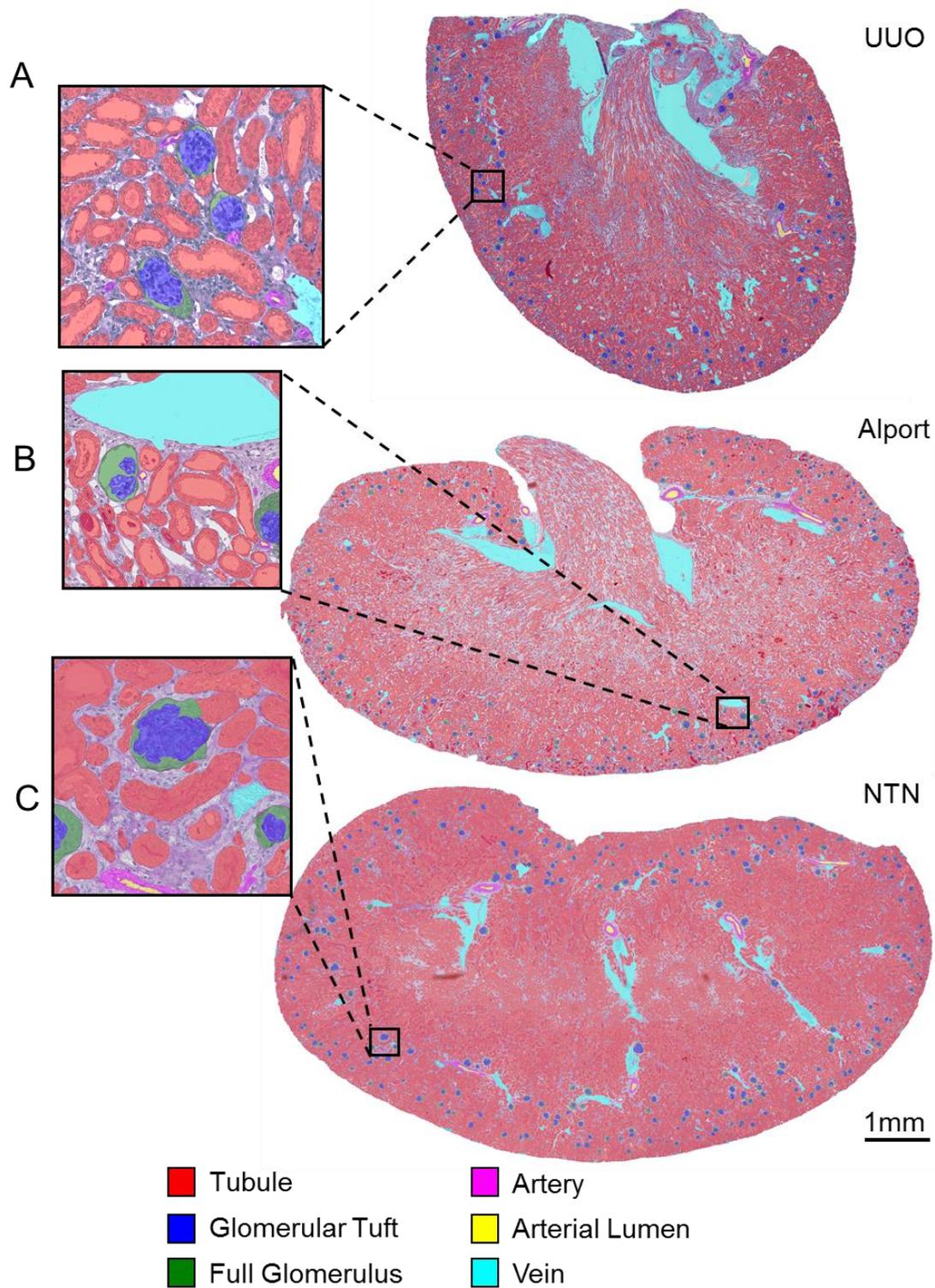


Blurred medullary net of capillaries and loop of Henle



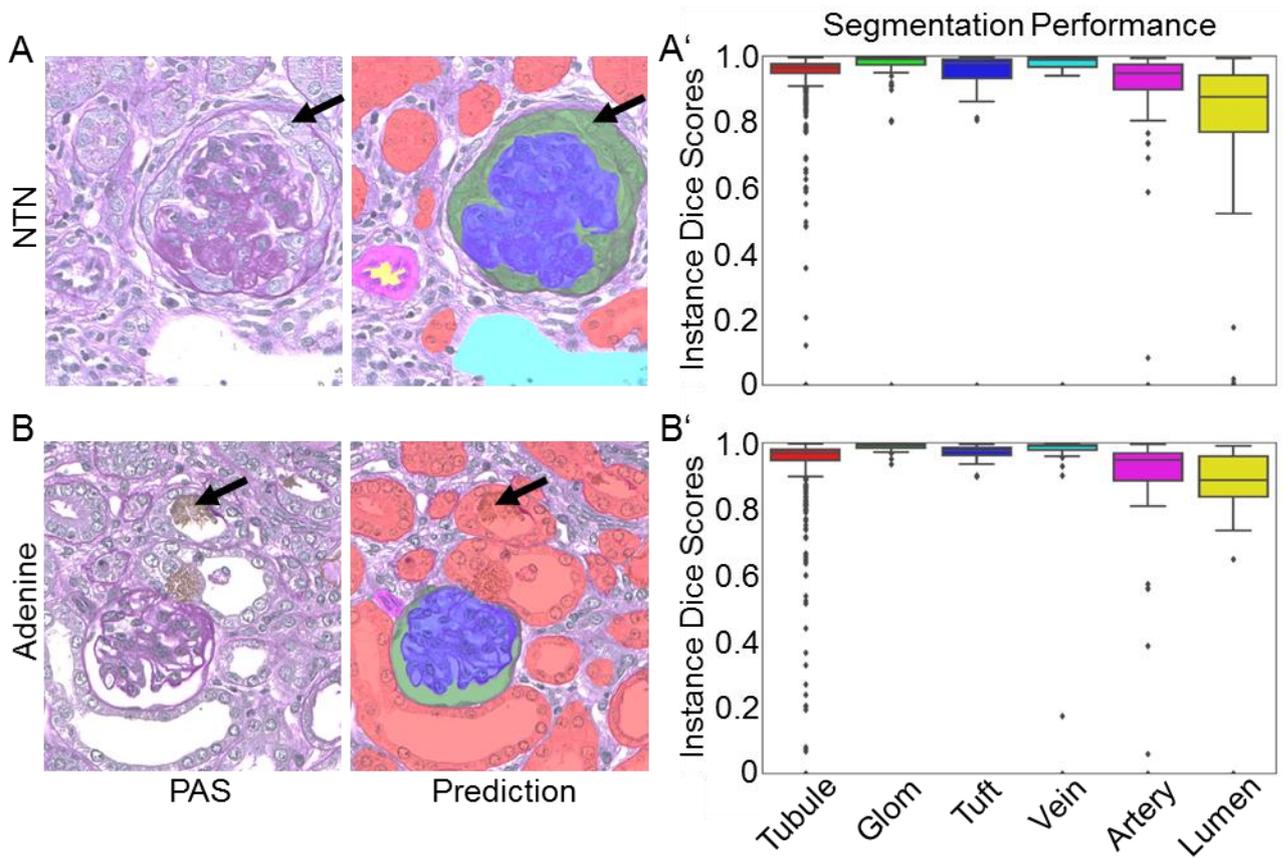
Supp. Fig. 2. Challenging morphology for manual and automated annotations.

(A-A'') show examples of glomeruli in PAS stained murine kidney sections. On a sectional plane close to the vascular or urinary pole it was difficult to discriminate between glomerular tuft and arterioles (arrow, A), or the glomerular tuft and parietal epithelial cells or tubular epithelial cells (arrows, A',A''). Sometimes the tubular basement membrane appeared discontinuous (arrows in B, B'). The distinction of medial layers of arteries was harder when vessels run side by side (arrow, C). (D-D'') show medulla of murine kidneys with the network of capillaries and the tubular system, which in some cases was not easy to discriminate.



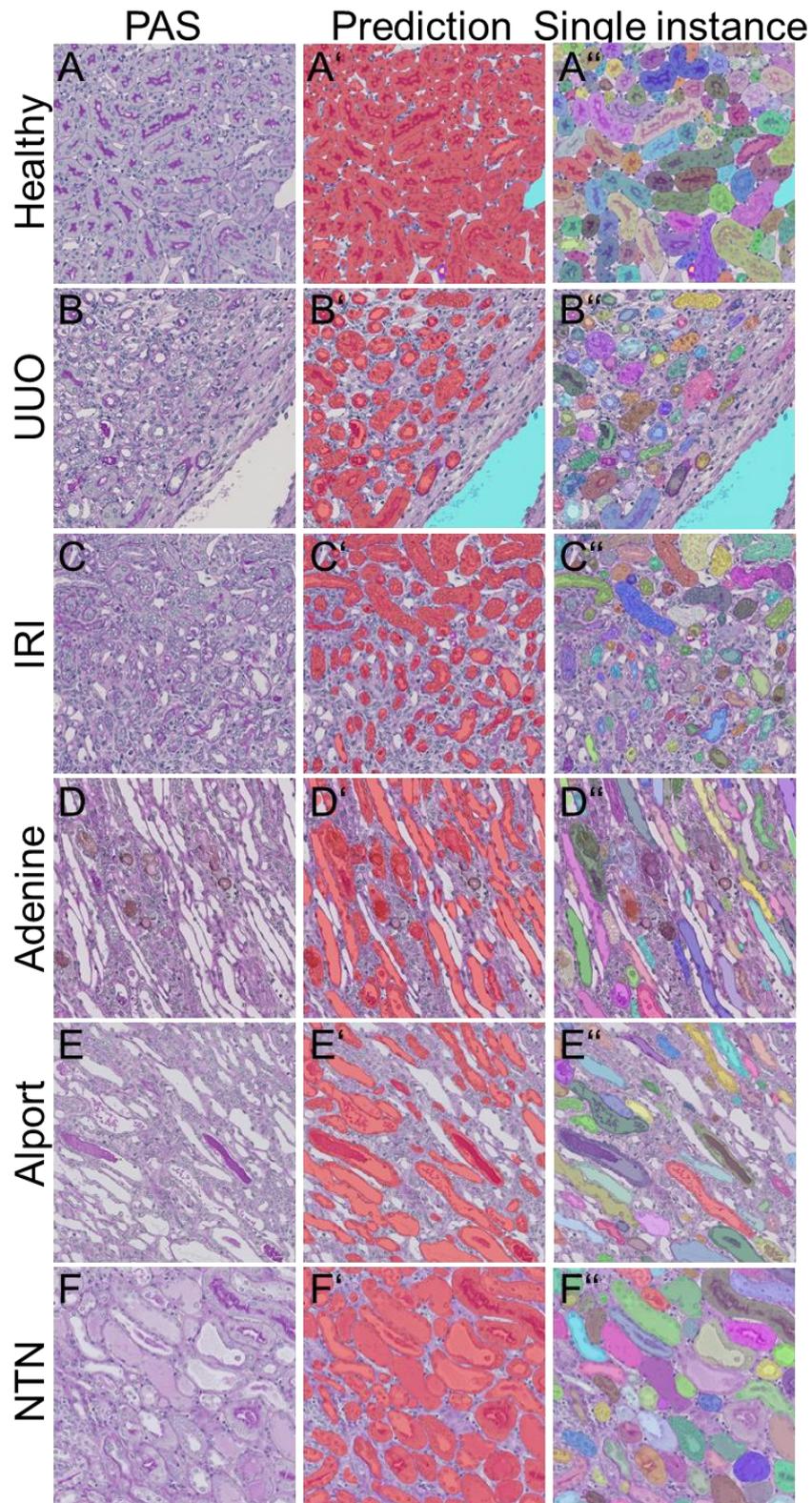
Supp. Fig. 3. Segmentation of WSI of UUU, Alport and NTN kidneys.

CNN generated segmentation predictions on a whole slide image (WSI) of an UUU (A), Alport (B) and NTN (C) mouse kidney. All six classes, were precisely segmented. NTN = nephrotoxic nephropathy, UUU = unilateral ureteral obstruction.



Supp. Fig. 4. Quantitative segmentation performance in murine NTN and adenine kidneys.

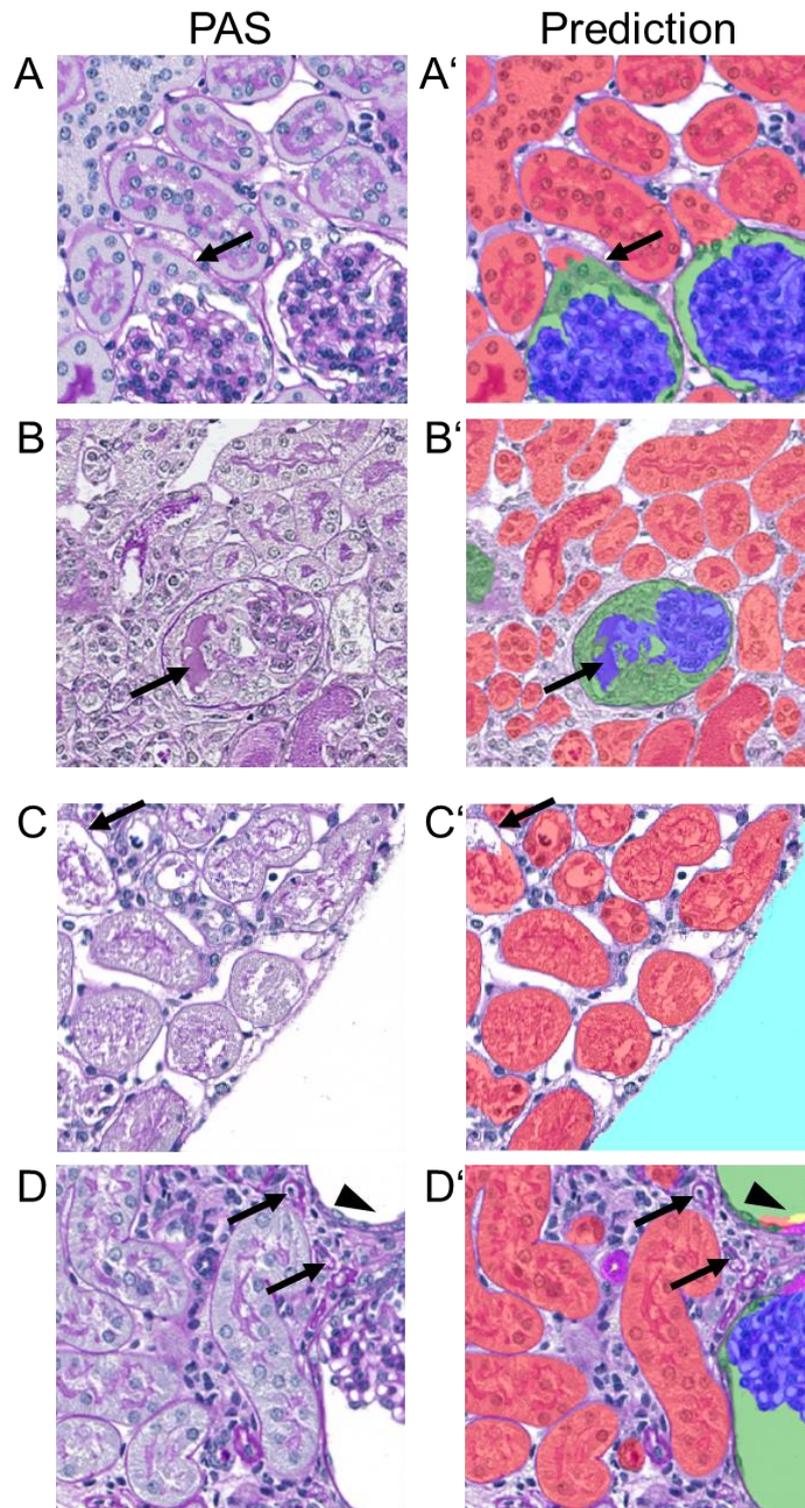
Representative PAS pictures and the corresponding segmentation prediction generated by our CNN for a murine NTN (A) and adenine kidney (B). Instance segmentation accuracy is shown by dice scores for each class in both models (A'-B'). Data are presented in Box plots with median, quartiles and whiskers. NTN = nephrotoxic nephropathy.



Supp. Fig. 5. Automated segmentation in the medulla of murine kidney sections.

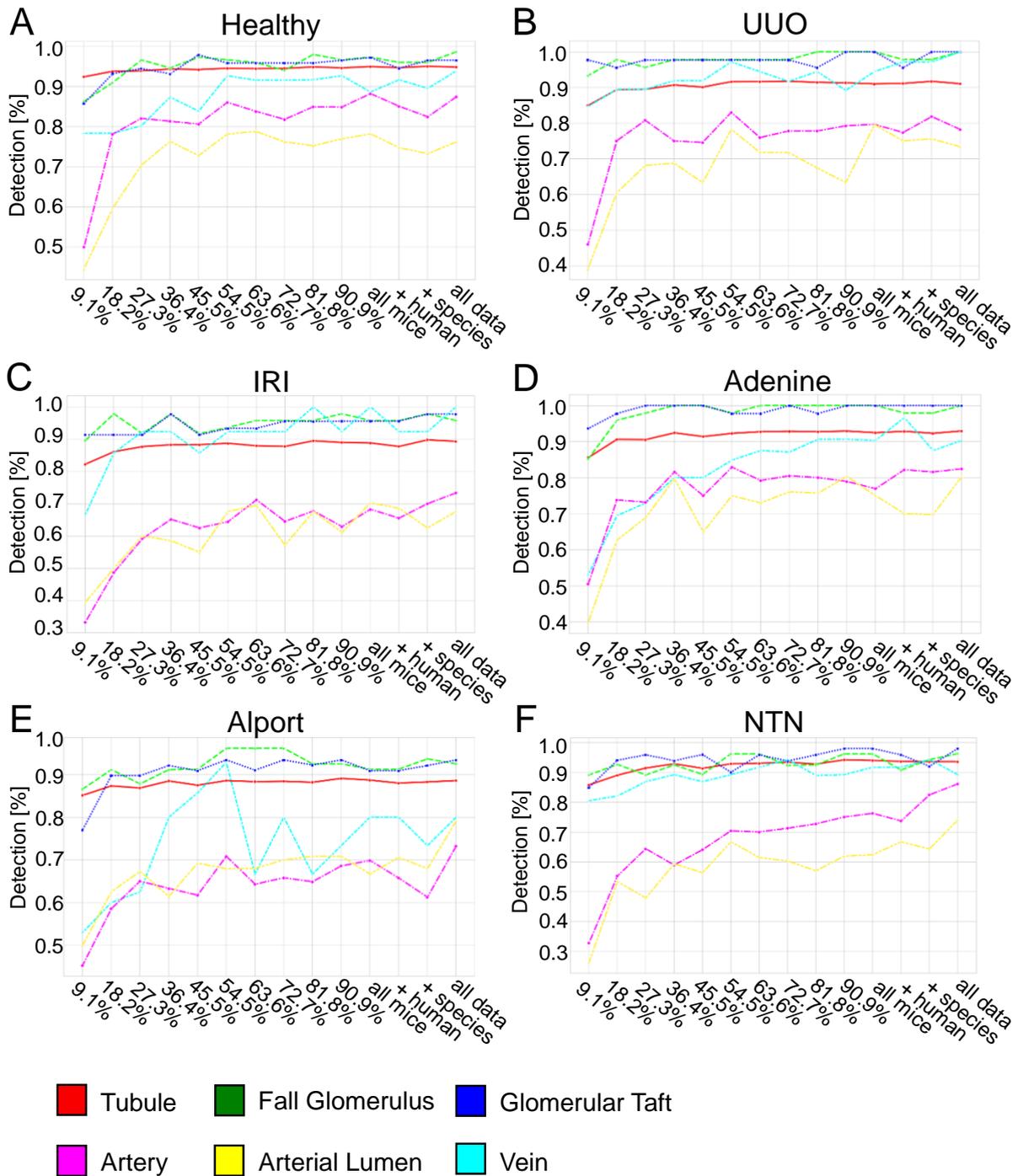
Representative PAS pictures and corresponding overlays with segmentation predictions showing either the different classes or every single instances for the medulla of murine healthy (A-A''), UUO (B-B''), IRI (C-C''), adenine (D-D''), Alport (E-E'') and NTN (F-F'') kidneys.

IRI = ischemia-reperfusion injury, NTN = nephrotoxic nephropathy, UUO = unilateral ureteral obstruction.



Supp. Fig. 6. Examples of missclassifications.

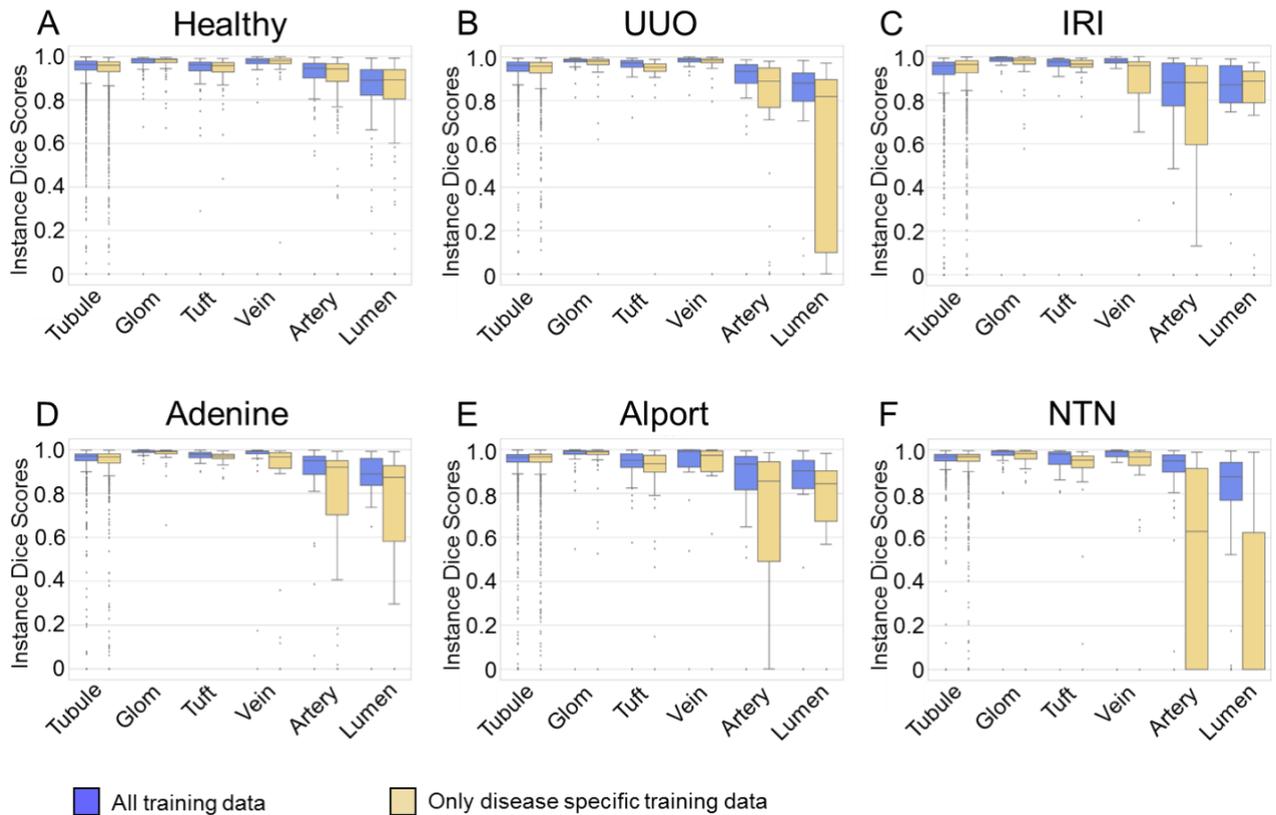
PAS photographs and prediction overlays show an incorrect separation of a “full glomerulus” and the connected proximal “tubule” (arrow in A, A’), a glomerular tuft that was inaccurately segmented with projections into the crescent (arrow in B, B’) and an incompletely segmented tubule due to extensive necrosis (arrow in C,C’). Another example shows a strongly dilated tubule which is was incorrectly classified as full glomerulus and arterial lumen (arrowheads in D,D’) and missing segmentations of atrophic tubules (arrows in D,D’).



Supp. Fig. 7. Relation between amount of training data and detection performance.

The detection performance for all six classes in healthy (A), UUO (B), IRI (C), adenine (D), Alport (E) and NTN (F) was plotted against the amount of total data used for CNN training.

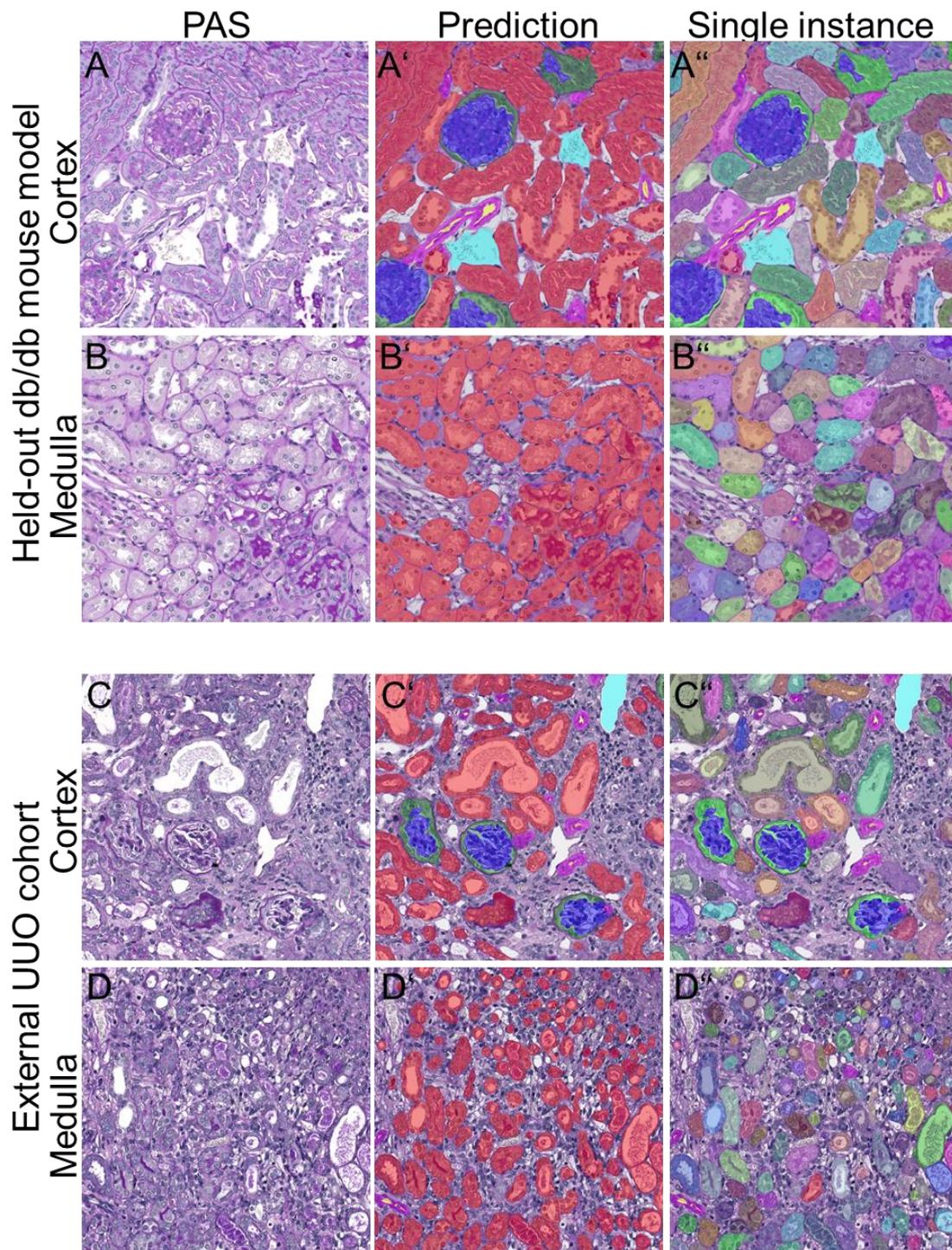
IRI = ischemia-reperfusion injury, NTN = nephrotoxic nephropathy, UUO = unilateral ureteral obstruction.



Supp. Fig. 8. Comparison between our full CNN and its variants independently trained on single models.

(A) Segmentation performance shown as instance dice scores for all six classes was compared on our healthy kidney test data between our full CNN trained on all training data (blue) and its variant that has been solely trained with data from healthy kidneys (yellow). (B) The same comparison is shown for the UUO, in which the network variant was exclusively trained with annotations from UUO kidneys. Analogously, analyses are performed for IRI (C), adenine (D), Alport (E) and NTN (F).

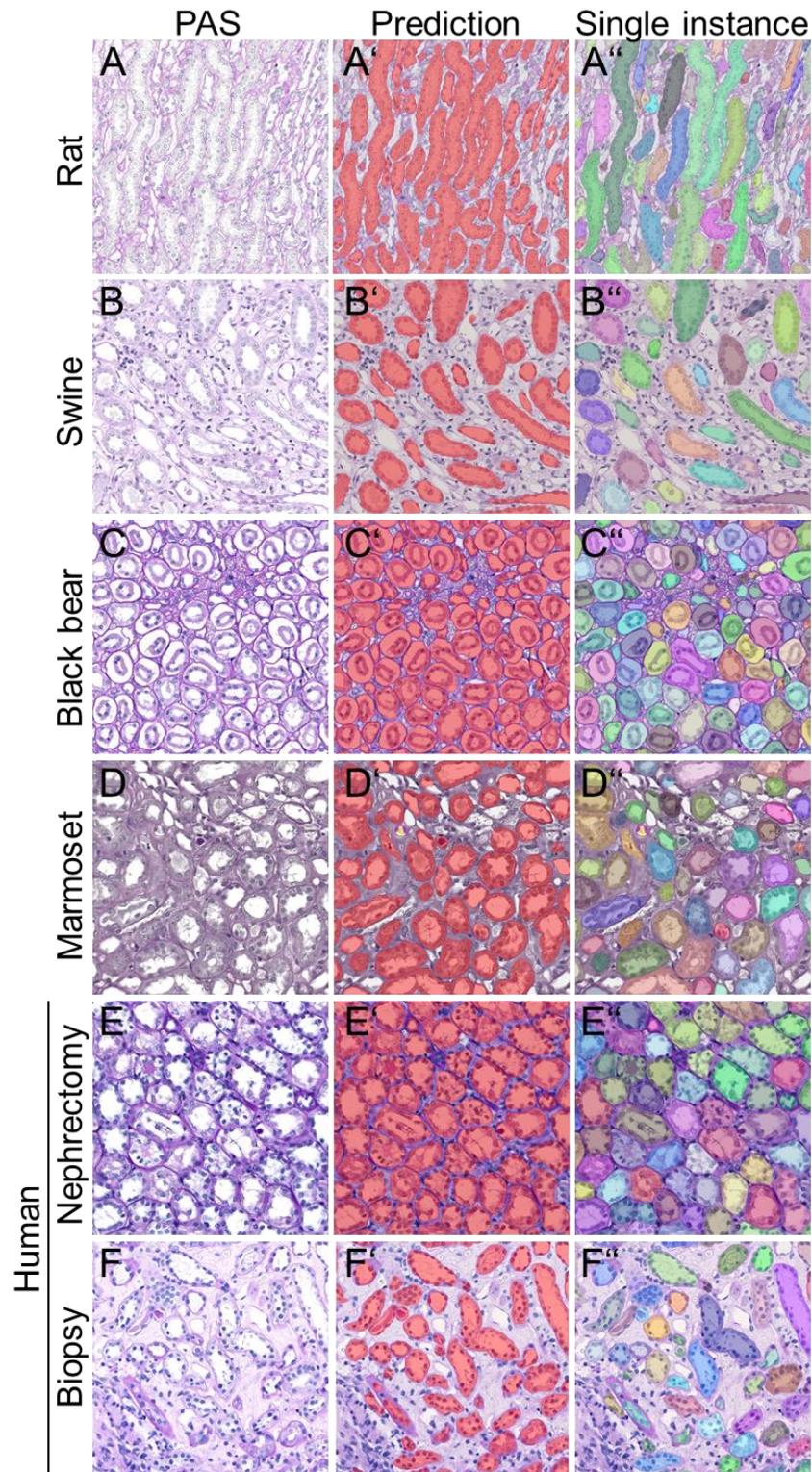
Data are presented in Box plots with median, quartiles and whiskers. IRI = ischemia-reperfusion injury, NTN = nephrotoxic nephropathy, UUO = unilateral ureteral obstruction.



Supp. Fig. 9. Segmentation of non-trained and external murine kidney slides.

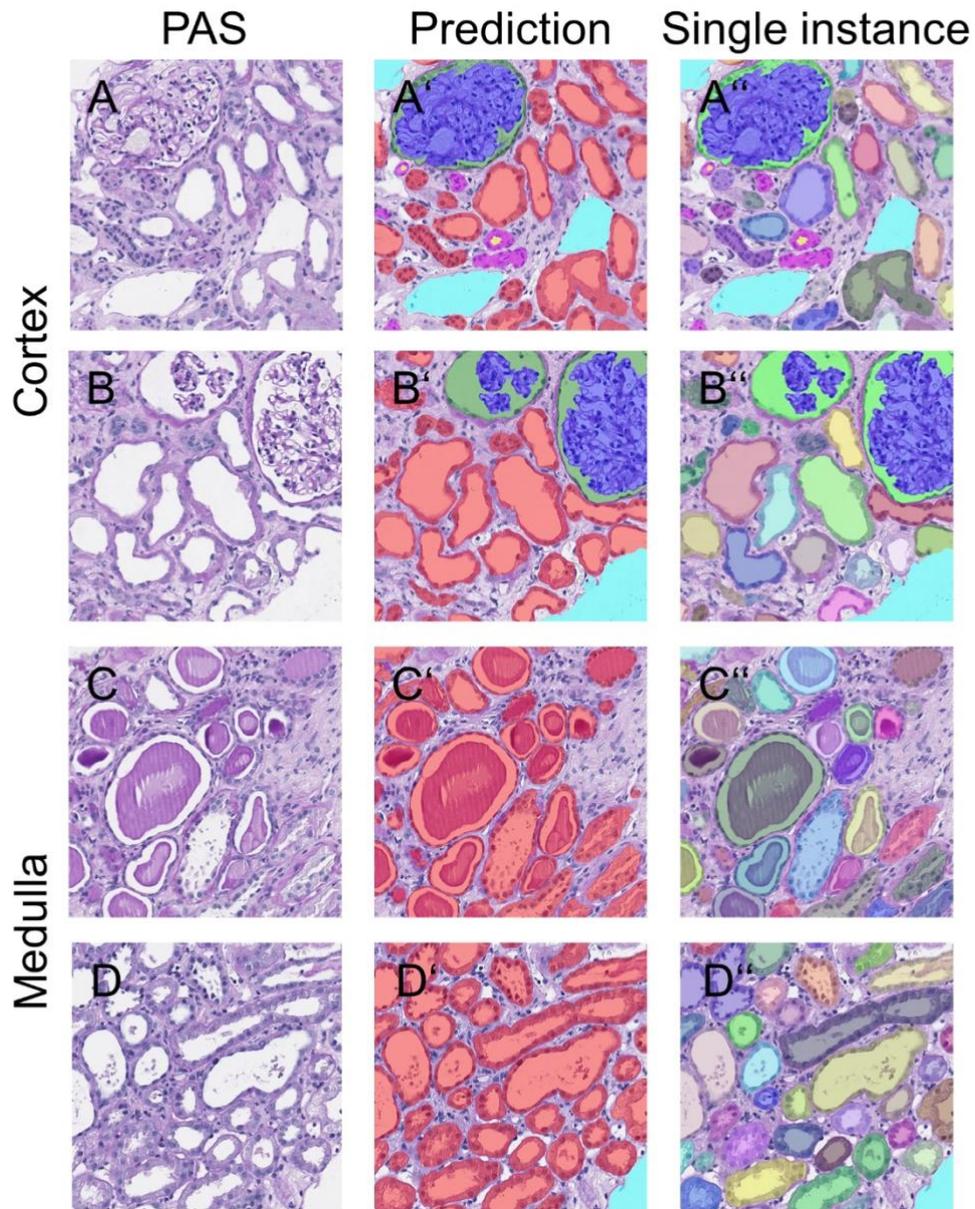
Representative pictures show segmentation results for cortex (A-A'') and medulla (B-B'') for kidneys from db/db mice fed with high fat western diet. Predictions (A', B') depict different classes, while A'' and B'' display segmentation on single instance level. The CNN also accurately segments cortex (C-C'') and medulla (D-D'') from PAS slides of an external UJO cohort. Predictions (C', D') depict different classes, while C'' and D'' display segmentation on single instance level.

UJO = unilateral ureteral obstruction.



Supp. Fig. 10. Automated segmentation of renal medulla in different species.

Representative PAS pictures and the corresponding overlays for segmentation predictions showing either the different classes or every single instance for the medulla of rat (A-A''), pig (B-B''), black bear (C-C''), marmoset (D-D'') and human (E-F'') kidneys. Segmentation is accurate on human nephrectomy (E-E'') as well as on biopsy specimens (F-F'').



Supp. Fig. 11. Automated segmentation of human biopsies presenting with acute tubular damage. Representative PAS-pictures and the respective segmentation prediction overlays from cortex (A-B'') and medulla (C-D'') of human biopsies with acute tubular damage.