THE LANCET
Digital Health

Supplementary appendix

This appendix formed part of the original submission and has been peer reviewed. We post it as supplied by the authors.

Supplement to: Kers J, Bülow RD, Klinkhammer BM, et al. Deep learning-based classification of kidney transplant pathology: a retrospective, multicentre, proof-of-concept study. *Lancet Digit Health* 2021; published online Nov 15. https://doi.org/10.1016/S2589-7500(21)00211-9.

List of Supplementary Figures:

Supplementary Tables:

Supplementary Methods

Selection of the patient cohorts

- Amsterdam UMC cohort (AMS): (inclusion period 01-01-2000 till 01-06-2018): all patients who underwent any kidney transplant biopsy available in the pathology archives of the Amsterdam UMC, location AMC (Academic Medical Centre) were considered for inclusion (N = 3376 biopsies). There was an average of 1·94 and a range of 1 - 12 biopsies per patient available. For the collection of biopsies within the DEEPGRAFT cohort, only the last biopsy was included and digitized, making sure that no acute biopsy-proven events occurred between the moment of biopsy and last follow-up and that biopsies with sufficient chronicity were included, which allows the algorithms to specifically learn to deal with such features. Patients with only implantation biopsies, biopsies from external consultations/revision, biopsies with an oncological indication and biopsies from other organs than the kidney transplant were excluded. This resulted in 1136 eligible biopsies from 1136 individuals. From this set of 1136 biopsies, 59 biopsies were not available from the archives (neither glass sections nor formalin-fixed and paraffin-embedded tissue blocks) leaving 1077 biopsies from 1077 patients. From these 1077 eligible biopsies, 1055 had a stained glass biopsy available for all three stainings (H&E, PAS, Jones' silver), which were digitised with the Philips IntelliSite Ultra Fast scanner. After scanning, manual quality control excluded 64 biopsies from which the WSI of at least one of three stainings was considered insufficient, leaving a total of N = 991 biopsies for analysis. AMS-cohort part two (inclusion period 01-01-2019 till 31-12-2019): all patients who underwent any kidney transplant biopsy available in the pathology archives of the Amsterdam UMC, location AMC were considered for inclusion in this cross-sectional cohort. All 167 biopsies were included for digitisation with the Philips IntelliSite Ultra Fast scanner and 28 were excluded during manual quality control, leaving a total of 139 biopsies in this second part. Scanning took approximately 2 minutes per biopsy section.

The final full AMS cohort used in the study thus consists of 1130 biopsies from 1130 patients. Details on patients and biopsies can be found in Supplementary Tables 1 and 2.

- UMC Utrecht cohort (UMCU): (inclusion period 01-01-2000 till 01-03-2016 for cohort 1 and 01-03-2016 till 31-12-2019 for cohort 2): all patients who underwent any kidney transplant biopsy available in the pathology archives of the UMC Utrecht were considered for inclusion (N = 1704 biopsies). There was an average of 1·82 and a range of 1 - 12 biopsies per patient available. For the collection of biopsies within the DEEPGRAFT cohort, only the last biopsy was included and digitised (see Amsterdam UMC cohort 1). Patients with only implantation biopsies, biopsies from external consultations/revision, biopsies with an oncological indication and biopsies from other organs than the kidney transplant were excluded. This resulted in 877 eligible biopsies from 912 individuals. UMC Utrecht has a fully digital WSI archive scanned at 40x since 01-03-2016: cases before this date (UMCU cohort part 1, N = 608) were processed for rescanning at the Amsterdam UMC and cases after this date (UMCU cohort part 2, N = 269) were extracted directly from the digital WSI archive. From the 608 biopsies in cohort 1, 8 biopsies were not available from the archives to complete the set of three stainings (neither glass sections nor formalin-fixed and paraffin-embedded tissue blocks) leaving 600 biopsies from 600 patients that were scanned and exported at 40x with the Philips IntelliSite Ultra Fast scanner. The 269 cases (cohort 2) were scanned at 40x in the clinical workflow with the Hamamatsu XR scanner. The two cohorts differ in the whole slide scanners used for digitalization of the slides. After scanning, we performed manual quality control and excluded 160 from cohort 1, because the WSI of at least one of three stainings was considered insufficient, leaving a total of N = 448 biopsies for analysis in cohort 1 and 269 biopsies in cohort 2 for analysis.

  The final full UMCU cohort thus consists of 717 biopsies from 717 patients. Details on patients and biopsies can be found in Supplementary Tables 1 and 2.

- Aachen kidney cohort (AC-K, inclusion period 01-01-2019 till 31-12-2020): All kidney allograft biopsies identified during the time-period in the archives of the Institute of Pathology, RWTH Aachen University Hospital Aachen, Germany were considered for eligibility (N = 134). There were an average of 1·33 and a range of 1-3 biopsies available per patient. 29 cases were a second or third biopsy from the same patient and the latest biopsy was digitised, leaving a total of 105 biopsies from 105 patients. These biopsies were digitised using an Aperio AT2-Scanner (Leica Biosystems, Wetzlar, Germany) with the 40x objective. After scanning, manual quality control for staining and scanning quality was performed, evaluating e.g., staining intensity and sharpness of the scans, leading to exclusion of four additional cases.

  The final AC-K cohort consists of 101 biopsies from 101 patients. Details on patients and biopsies can be found in Supplementary Tables 1 and 2.

In all cohorts, adequacy was assessed based on the glomerular count (at least 5) or if a definitive diagnosis could be made (if, e.g., a rejection was evident although only 3 Glomeruli were present in the biopsy).

The following data were not available in this study:

- Transplant recipient characteristics: Initial nephropathy, transplant rank, data on previous episodes of rejection.
- Donor characteristics: Donor type (living or dead), age, cause of death, classification if standard of expanded criteria donor.
- Transplant characteristics: Data on HLA stats (regarding HLA mismatch), AB0-incompatibility, induction therapy, data on initial immunosuppressive therapy.
- Patient characteristics at time of biopsy: Serum-Creatinine, urinary protein/creatinine-ratio, BK viremia, immunosuppressive therapy.

WSI quality control and generation of tiles and pre-processing

All whole slide images (WSIs) were loaded into QuPath[1] - i.e. the most commonly used open-source digital pathology application, and the entire kidney tissue, including both cortex and medulla was annotated manually as regions of interest, in parallel to manual checking for scan and staining quality (e.g. tissue artifacts, scan artifacts, staining artifacts). During annotation, the annotator was blinded for the respective diagnostic class of the tissue. All annotated regions were tessellated into tiles of sizes 256 pixel edge length, corresponding to 128 µm (resolution of 0.5 µm/pixel) from all WSIs using QuPath. Tiles consisting of more than 50% non-tissue area were automatically excluded from all analyses. Images were resized to fit model input sizes using bilinear interpolation. Input sizes of the used ResNets were [224 224], of the shufflenet [256 256] and of the inceptionv3 model [299 299]. All tiles were assigned to the same classes as their parent WSI, and all WSIs inherited their classes from the respective patients. Data augmentation techniques included random flipping and rotation of the tiles to assure rotational independence. Detailed information on data pre-processing steps are available in the "Aachen Protocol for Deep Learning Histology".[2]

Determining the best CNN architecture

Initially we trained several CNN architectures to determine the one best suited for the classification task. We evaluated the performance of ResNets[3] (ResNet18, ResNet50, ResNet101), Inceptionv3[4] and ShuffleNet[5], all pre-trained on the ImageNet database[6] and retrained to classify biopsies into the classes mentioned earlier. The finally used inceptionv3 model is described in detail in Supplementary Table 2.

Cross-validation

Patient cohorts were split into three partitions before training, keeping the target labels balanced between partitions. In each cross-validation step, two partitions were used for training, and one partition was used for testing. All tiles were undersampled, so that tile

numbers per label were equal. Training and test sets were always split on patient level so that no image tiles from the same patient were ever part of a training set and a test set.

Hyperparameters

We used an initial learning rate of $5x10^{-5}$, maximum number of tiles per WSI of 1000 to inhibit over-representation of large biopsies and 30 trainable layers (last 30 layers) in all experiments. Training was performed for four epochs using a mini-batch-size of 512, a learn-rate-factor of 2, and gradient decay factor of 0·9, L2-regularization of $1x10^{-4}$.

Evaluation of CNN performance

Performance of all networks was evaluated on patient-level by pooling tile-predictions as described previously.[7] All performances were evaluated using patient level areas under the receiver-operating-characteristic curve (AUROC)[1] with ten times bootstrapped confidence bounds. We provide areas under the precision recall curves as additional performance metrics. In the single CNN each class is once used as a positive class and distinguished from the two other classes which are collectively used as negative classes.

The optimal operating point of the ROC curve was determined using MATLABs *perfcurve* function which automatically determines a slope (S) using the instance counts of positive (P) and negative classifications (N) and the costs of classifying a negative class falsely as a positive class as follows:

$$S = (Cost(P \mid N) - Cost(N|N))/(Cost(N \mid P) - Cost(P|P)) \times N/P$$

A line with the slope S is moved from the up left corner towards the ROC curve until intersection. The intersection point then is the optimal operating point of the ROC.

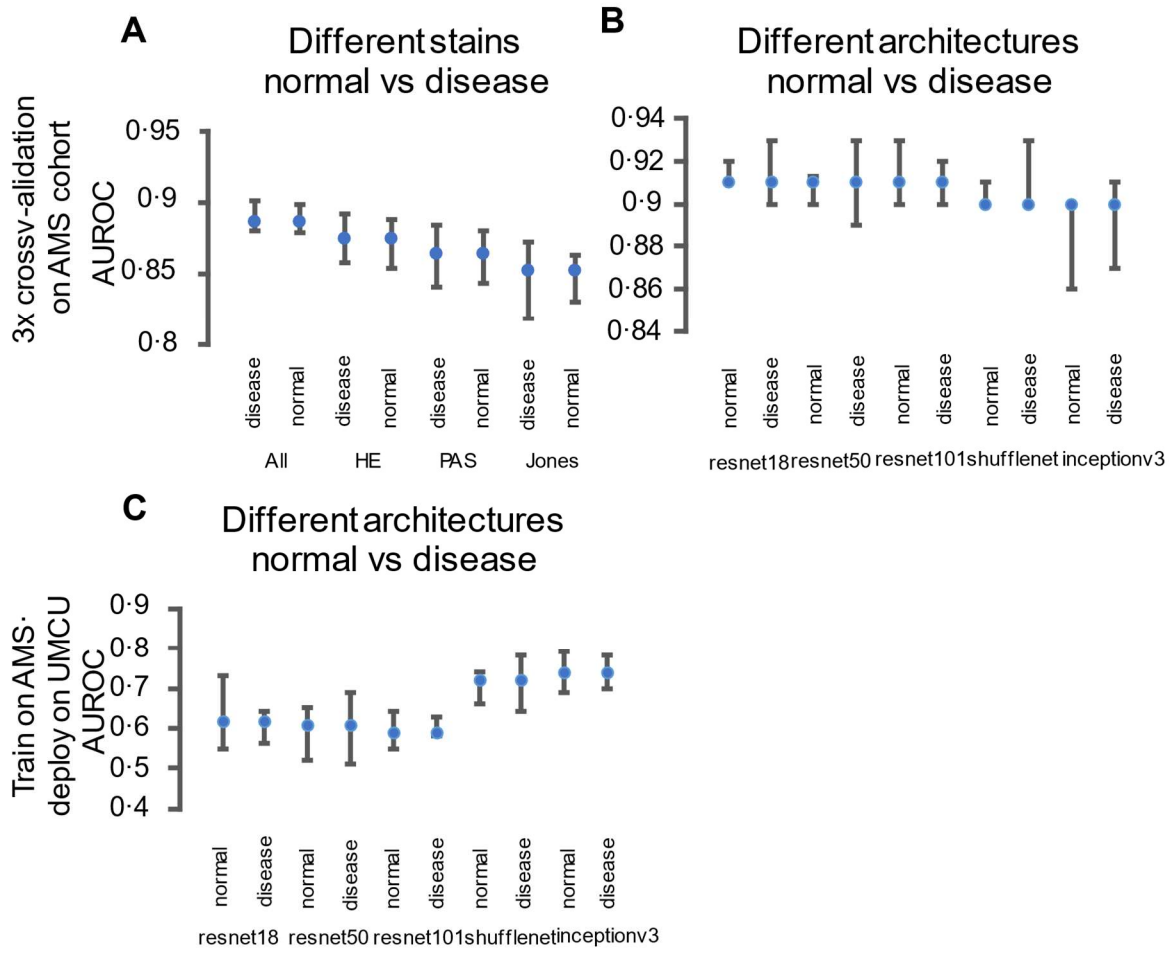
[1] An AUROC value of 0.5 describes a random classifier, while an AUROC of 1.0 describes a perfect classifier.

The optimal classification threshold on patient level was determined at that operating point and sensitivity, specificity, positive and negative predictive values were calculated after classification using this threshold by utilizing MATLABs *bootstrp* function with custom written functions for the calculation of sensitivity, specificity, positive and negative predictive value.

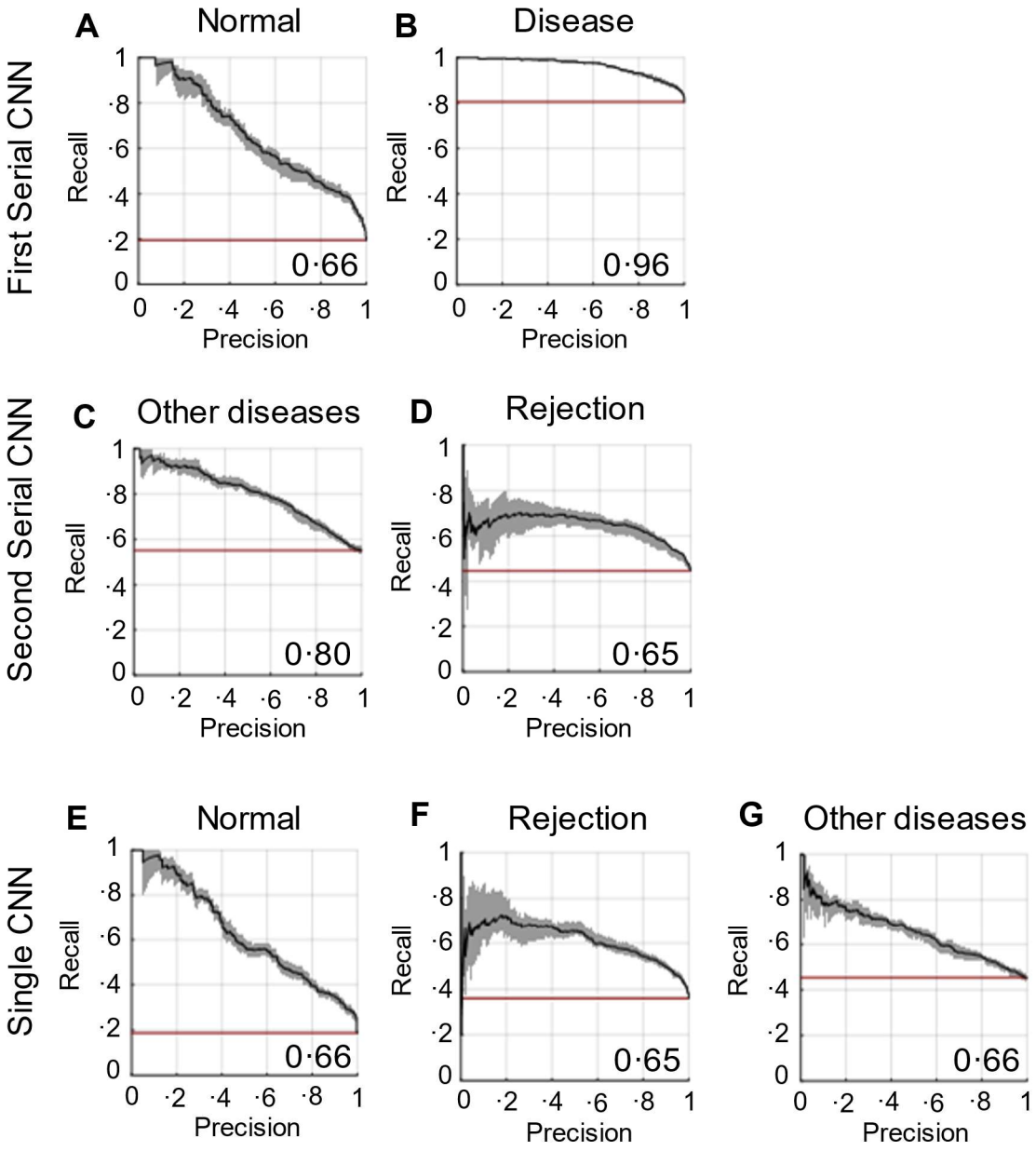
Visualisation of predictions using Occlusion Sensitivity and gradCAM

To visualise and interpret model predictions on the tile level, we used gradient-weighted Class Activation Mapping[8], i.e, a visualisation technique that uses the gradient information in the last convolutional layer to identify parts of an image that are important for classification. We have performed gradCAM analogous to the example provided in the MATLAB documentation[9]. Additionally, we have used Occlusion Sensitivity, i.e., a technique that iteratively occludes image parts and evaluates the impact of this occlusion on the confidence of the neural network for classification of the image. We performed Occlusion Sensitivity using MATLABs in-built Occlusion Sensitivity function.

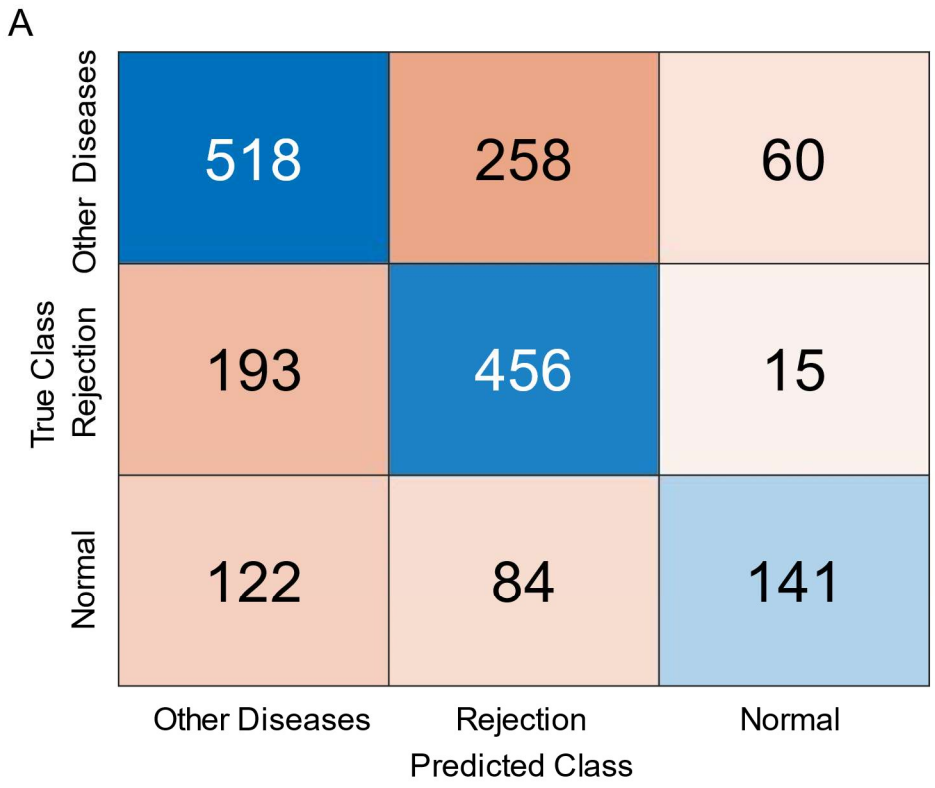
Supplementary Figures



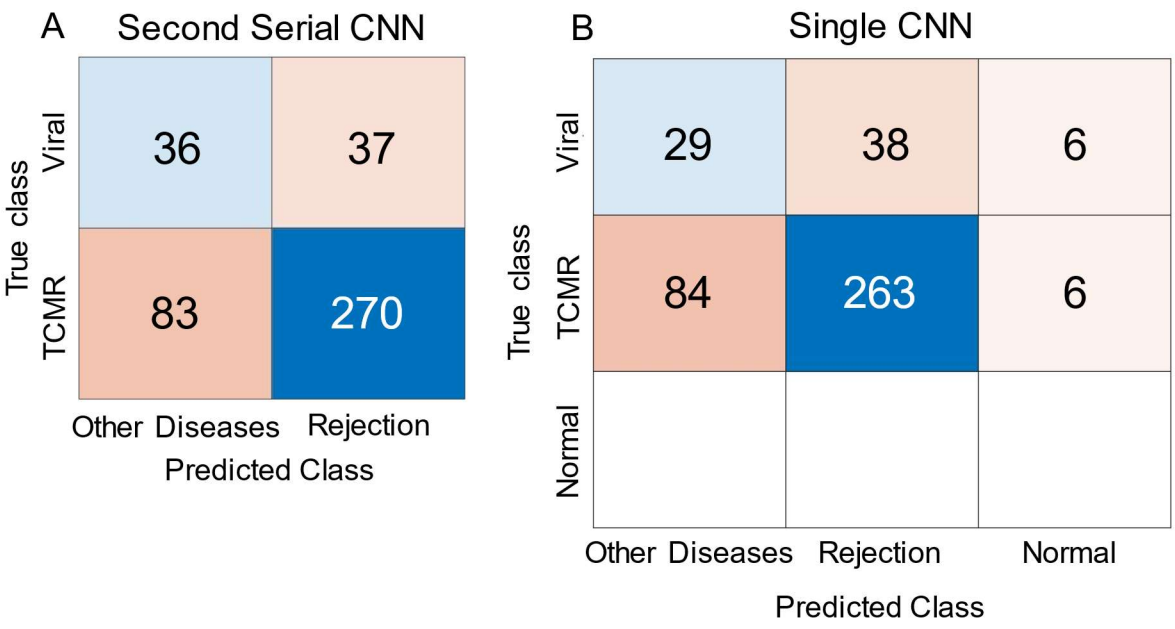
Supplementary Figure 1. Tests for optimal standard architecture. Using a quickly trainable shufflenet, differences in performance were investigated when using only PAS, HE or Jones-staining, or all stainings at once (A). Different standard architectures were used to classify biopsies into normal and disease (first serial CNN) (B). Both (A) and (B) were performed in threefold cross-validation on the AMS cohort. The models trained in (B) were deployed on the other half of the training cohorts, the UMCU cohort to test for generalisability (C). AMS: Amsterdam, UMCU: Utrecht, HE: Hematoxylin & Eosin, PAS: Periodic Acid Schiff, AUROC: Area under the receiver operating characteristic.



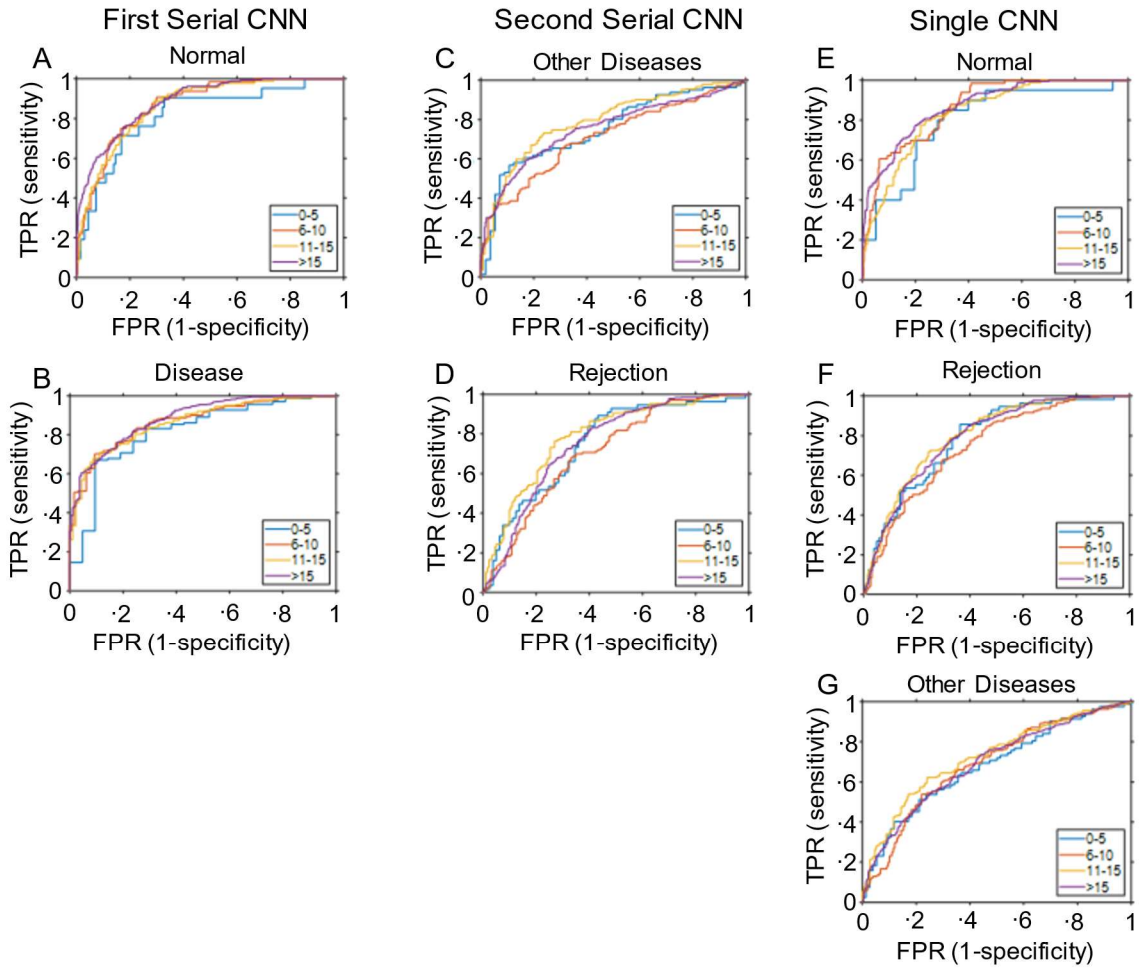
Supplementary Figure 2. Precision Recall Curves of the Serial CNNs and the single CNN in the AMS and UMCU cohorts. Patient level precision recall curves for the labels normal (A) and Disease (B) of the first serial CNN in the AMS and UMCU cohorts (Normal: N=347, Disease: N=1500, total: N=1847). Patient level precision recall curves for the labels other diseases (C) and rejection (D) of the second serial CNN in the AMS and UMCU cohorts (Other diseases: N=836, Rejection: N= 664 total: N=1500). Patient level precision recall curves for the labels normal (E), rejection (F) and other diseases (G) of the single CNN in the AMS and UMCU cohorts (Normal: N=347, Rejection: N=664, Other diseases: N=836, total: N=1847). Numbers in the bottom right indicate mean area under the precision recall curve (AUPRC) in a three-times cross-validated experiment. The red line indicates a random classifier.



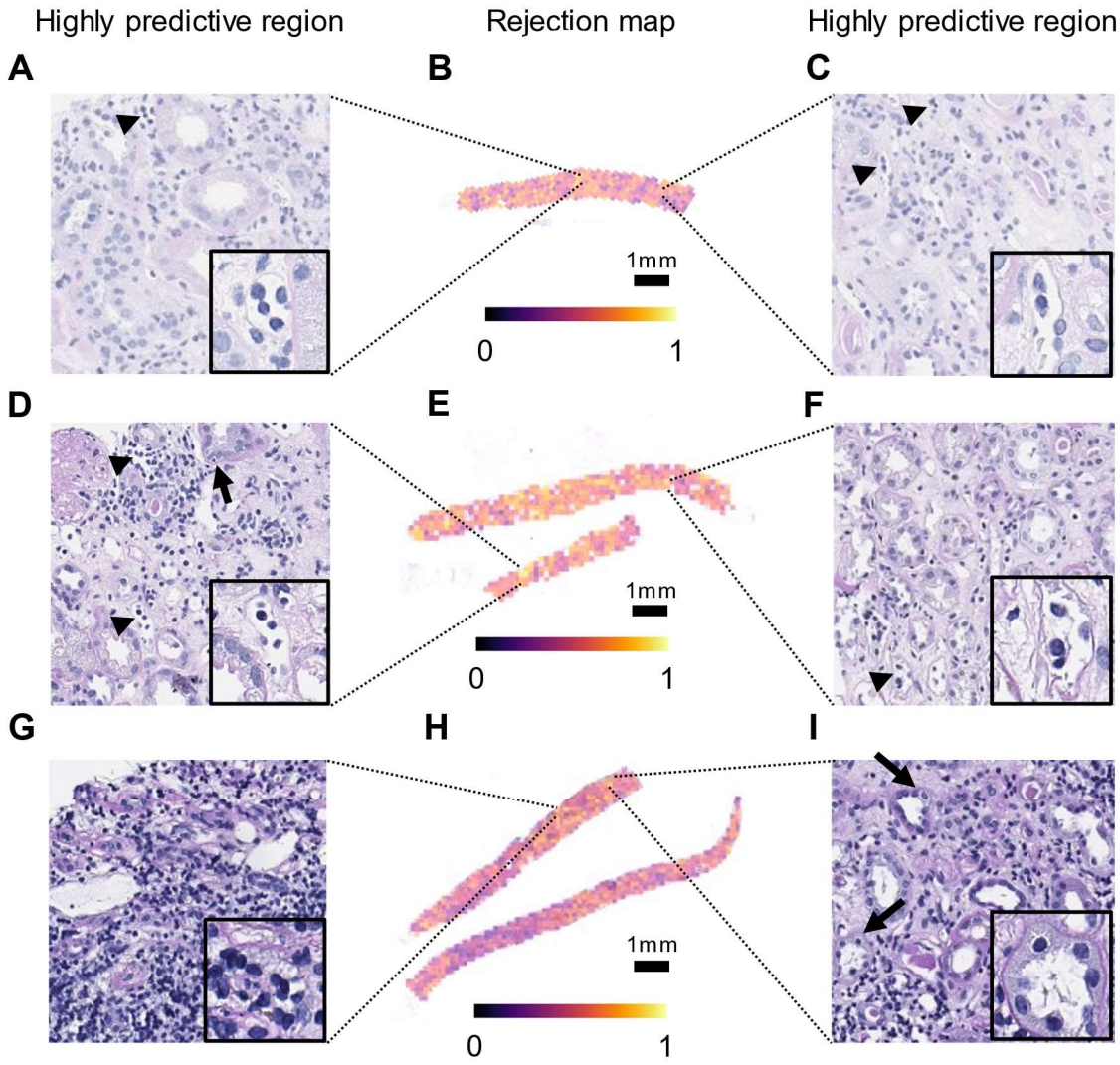
Supplementary Figure 3. Confusion Chart of pathologist and single CNN derived classes. Confusion occurs mainly between the classes rejection and other diseases. All analyses were performed within the combined AMS and UMCU cohort. True Class = Pathologist derived ground truth. Predicted Class = model derived class prediction.



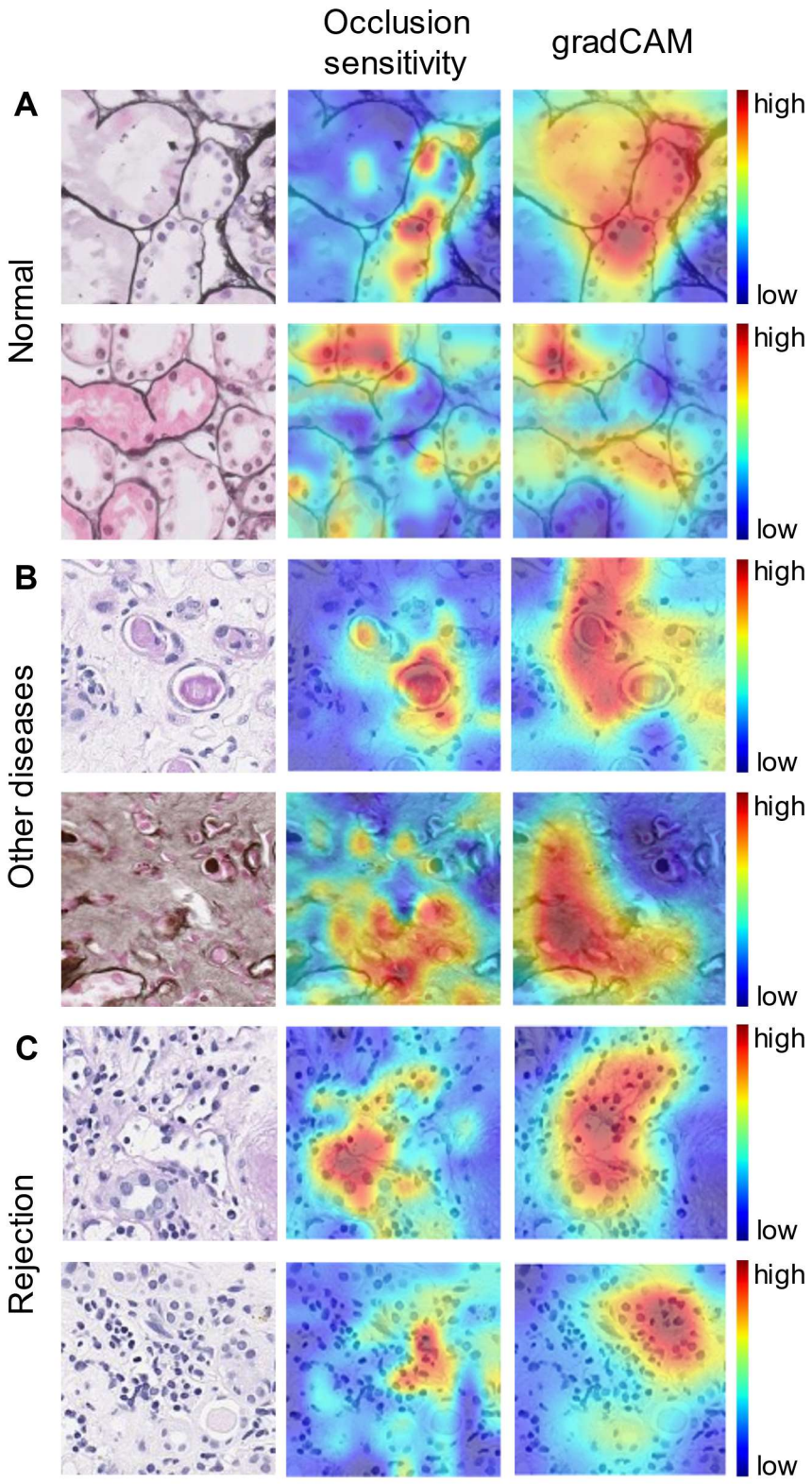
Supplementary Figure 4. Confusion Charts focusing on cases of T-cell mediated rejection and BK polyomavirus nephropathy. (A) Confusion chart for the second serial CNN, (B) confusion chart for the single CNN. All analyses were performed within the combined AMS and UMCU cohort. True Class = Pathologist derived ground truth. Predicted Class = model derived class prediction.



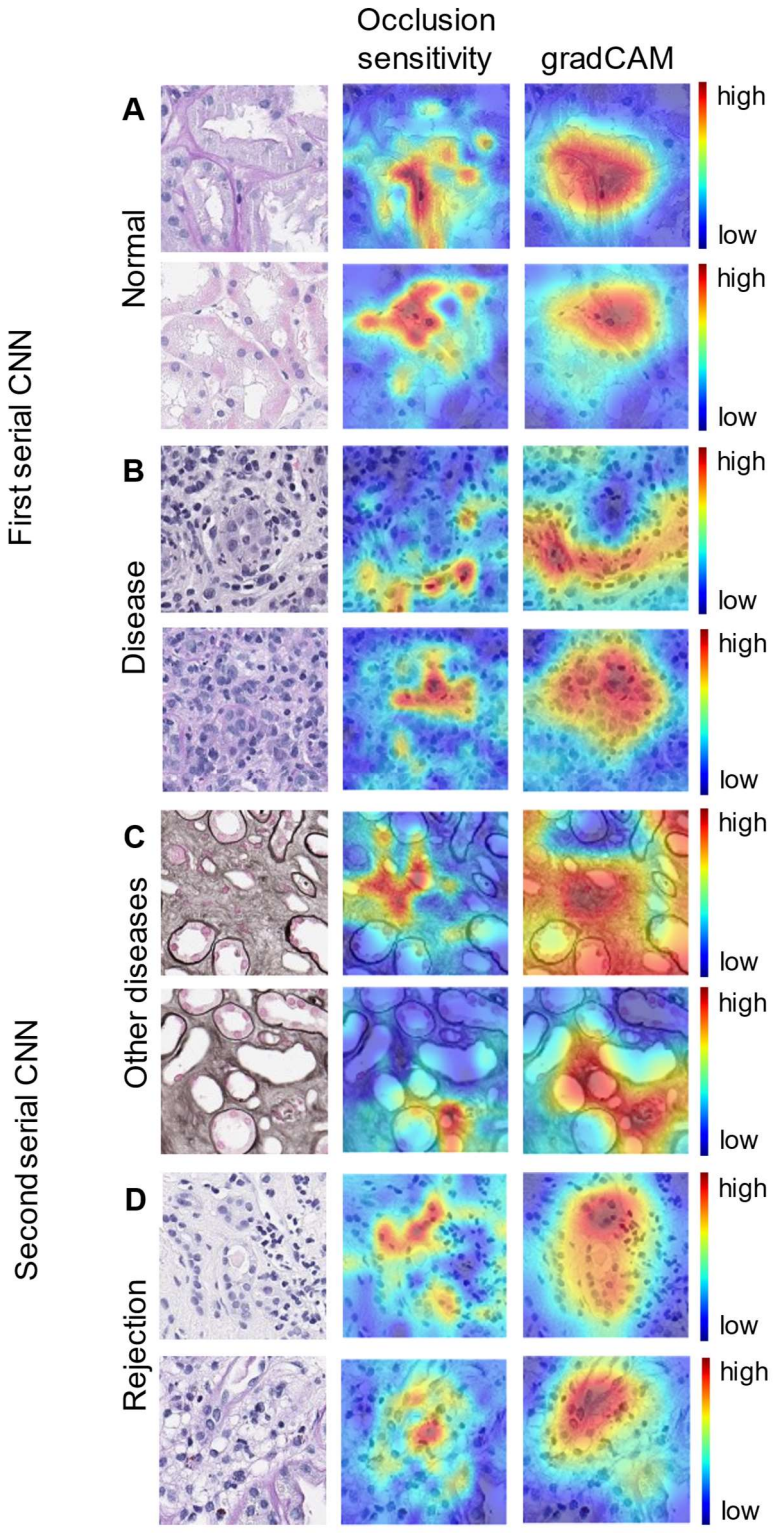
Supplementary Figure 5. Model performance based on glomerular count. (A-B) Performance depicted by the receiver operating characteristic curves (ROC) of the first serial CNN for the classes Normal and Disease within subgroups of biopsies with 0-5, 6-10, 11-15 and >15 glomeruli per biopsy. (C-D) ROCs of the second serial CNN for the classes Other Diseases and Rejection within subgroups with the same glomerular numbers as in A-B. (E-G) ROCs of the single CNN for the classes Normal, Rejection and Other Diseases within subgroups with the same glomerular numbers as in A-B. TPR: True Positive Rate, FPR: False Positive Rate. All analyses were performed within the combined AMS and UMCU cohort.



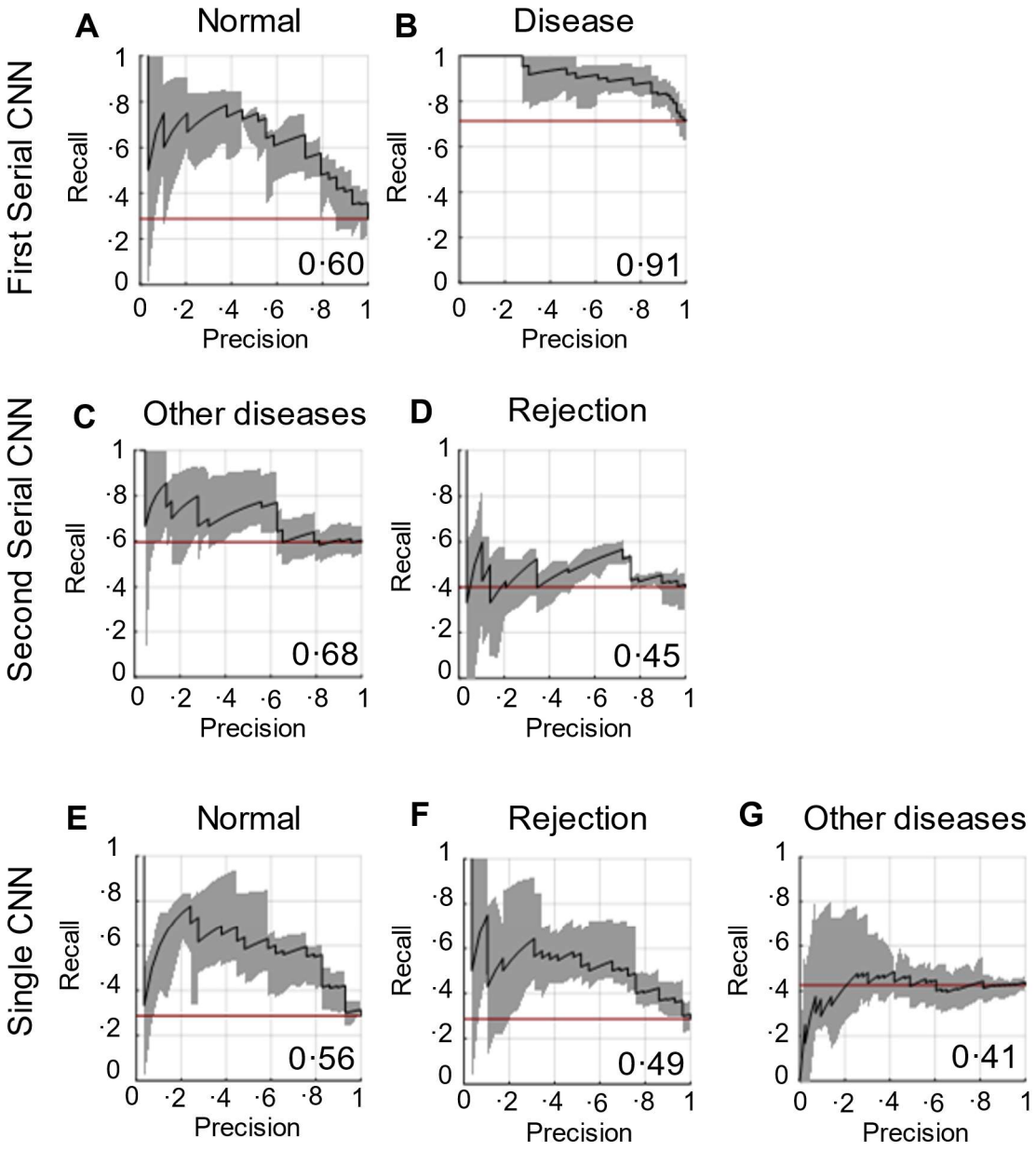
Supplementary Figure 6. Exemplary prediction maps for the class rejection and corresponding highly predictive areas. (A-C) A biopsy revealing peritubular capillaritis (A, C, arrowhead) and interstitial inflammation (A, C) in highly predictive areas identified by mapping the tile level predictions on their parent Whole Slide Image (WSI) (B). (D-F) Another biopsy reveals peritubular capillaritis (D, F, arrowhead), tubulitis (D, arrow) and interstitial inflammation (D, F) in highly predictive areas identified by mapping the tile level predictions on their parent WSI (E). (G-I) Another biopsy revealing tubulitis (I, arrow) and interstitial inflammation (G, I) in highly predictive areas identified by mapping the tile level predictions on their parent Whole Slide Image (WSI) (H). Magnification inlays present the highlighted pathological alterations in more detail.



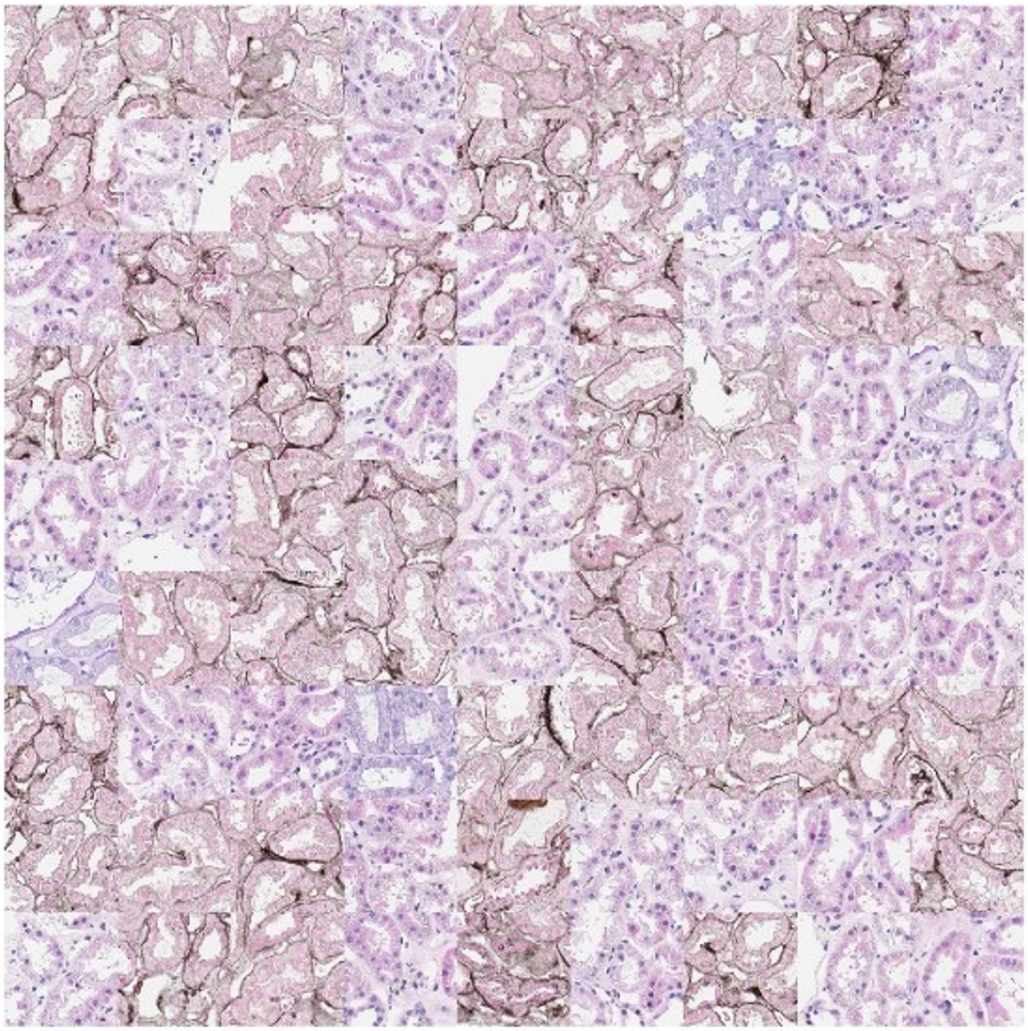
Supplementary Figure 7. Tile-level visualisations of the single CNN. Examples of highly predictive tiles and their tile-level visualisation using Occlusion Sensitivity and gradCAM for the class Normal (A), Other Diseases (B) and Rejection (C). gradCAM: gradient-weighted Class Activation Mapping.



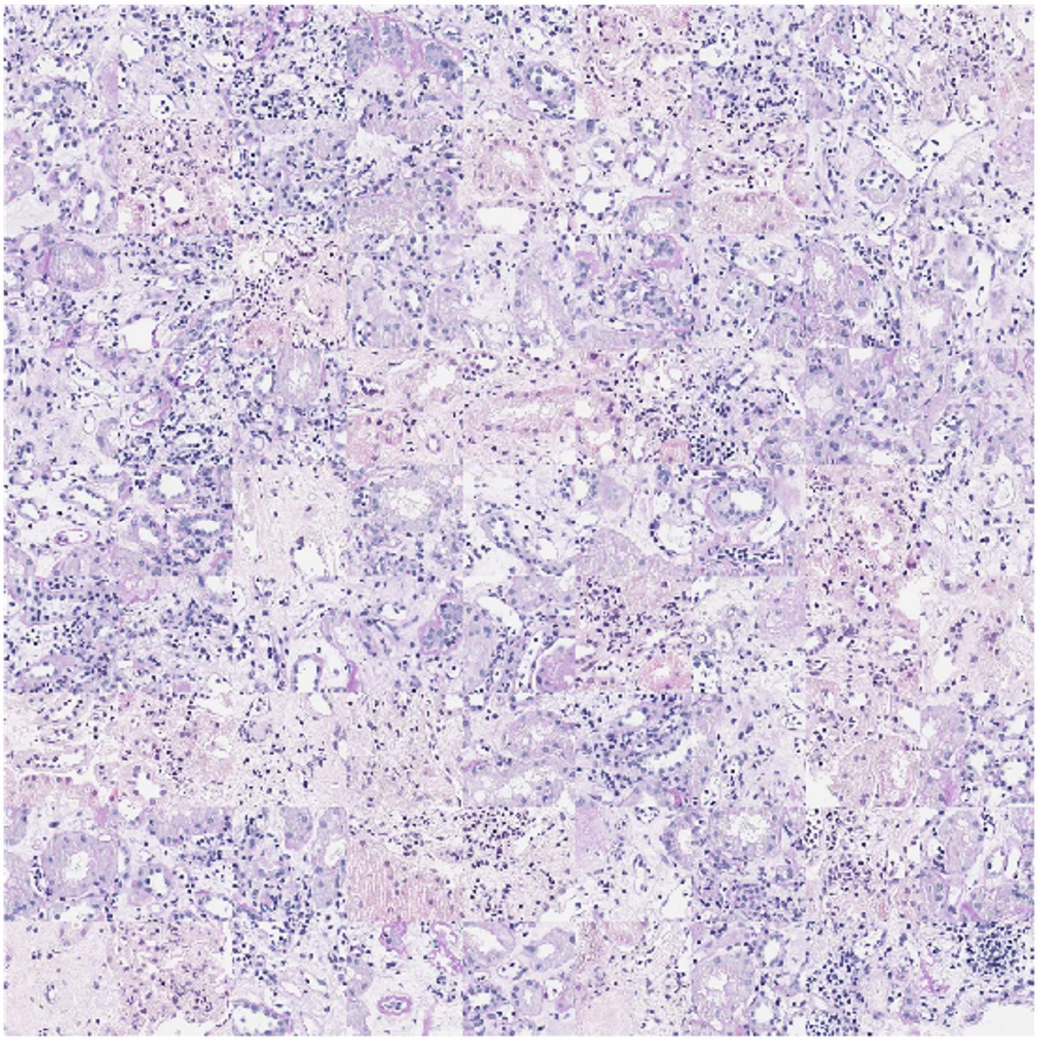
Supplementary Figure 8. Tile-level visualisations of the serial CNNs. Examples of highly predictive tiles and their tile-level visualisation using Occlusion Sensitivity and gradCAM in the first serial CNN classifying tiles as Normal (A) and Diseased (B) and the second serial CNN classifying tiles as Other diseases (C) or Rejection (D). gradCAM: gradient-weighted Class Activation Mapping.



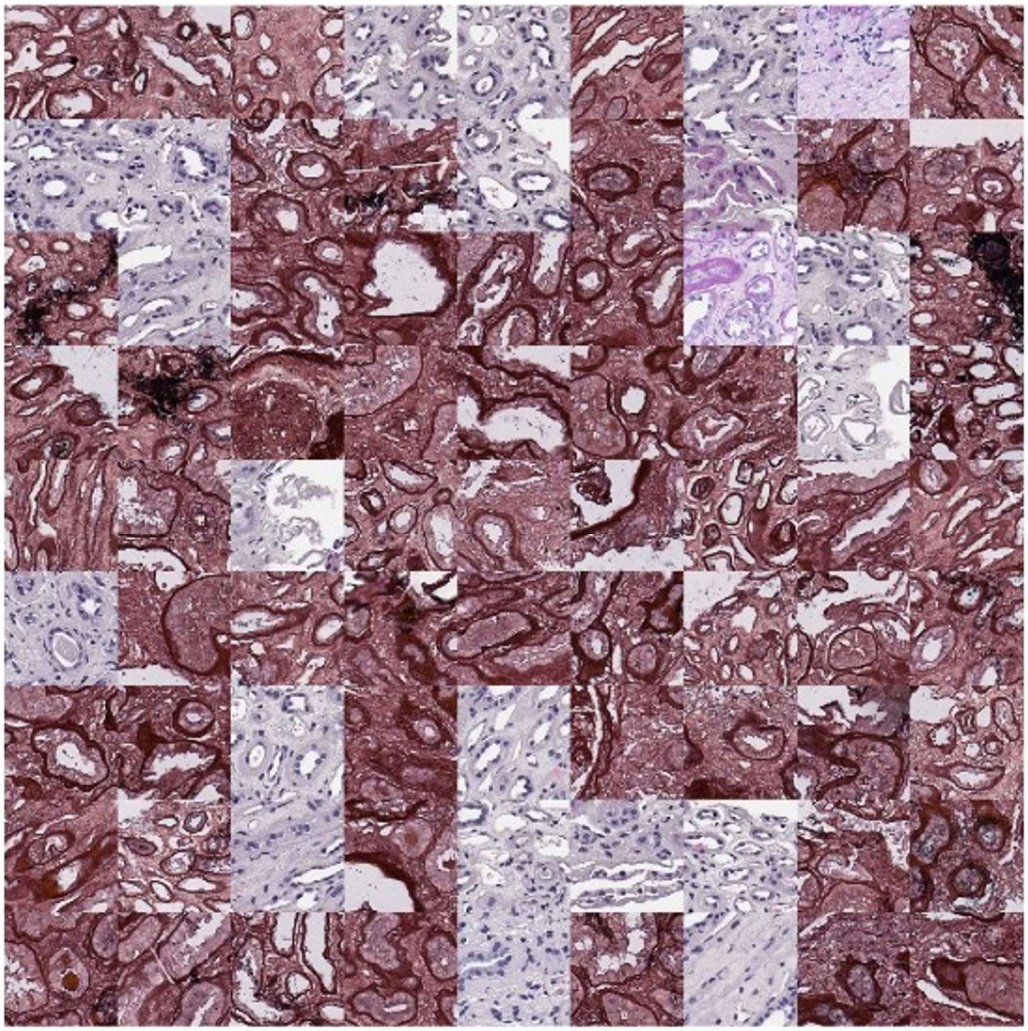
Supplementary Figure 9. Precision Recall Curves of the Serial CNNs and the single CNN in the external AC-K cohort. Patient level precision recall curves for the labels normal (A) and Disease (B) of the first serial CNN in the AC-K cohort (Normal: N=29, Disease: N=72, total: N=101). Patient level precision recall curves for the labels other diseases (C) and rejection (D) of the second serial CNN in the AC-K cohort (Other diseases: N=43, Rejection: N= 29 total: N=101). Patient level precision recall curves for the labels normal (E), rejection (F) and other diseases (G) of the single CNN in the AC-K cohort (Normal: N=29, Rejection: N=29, Other diseases: N=43, total: N=101). Numbers in the bottom right indicate mean area under the precision recall curve (AUPRC) in a three-times cross-validated experiment. The red line indicates a random classifier.



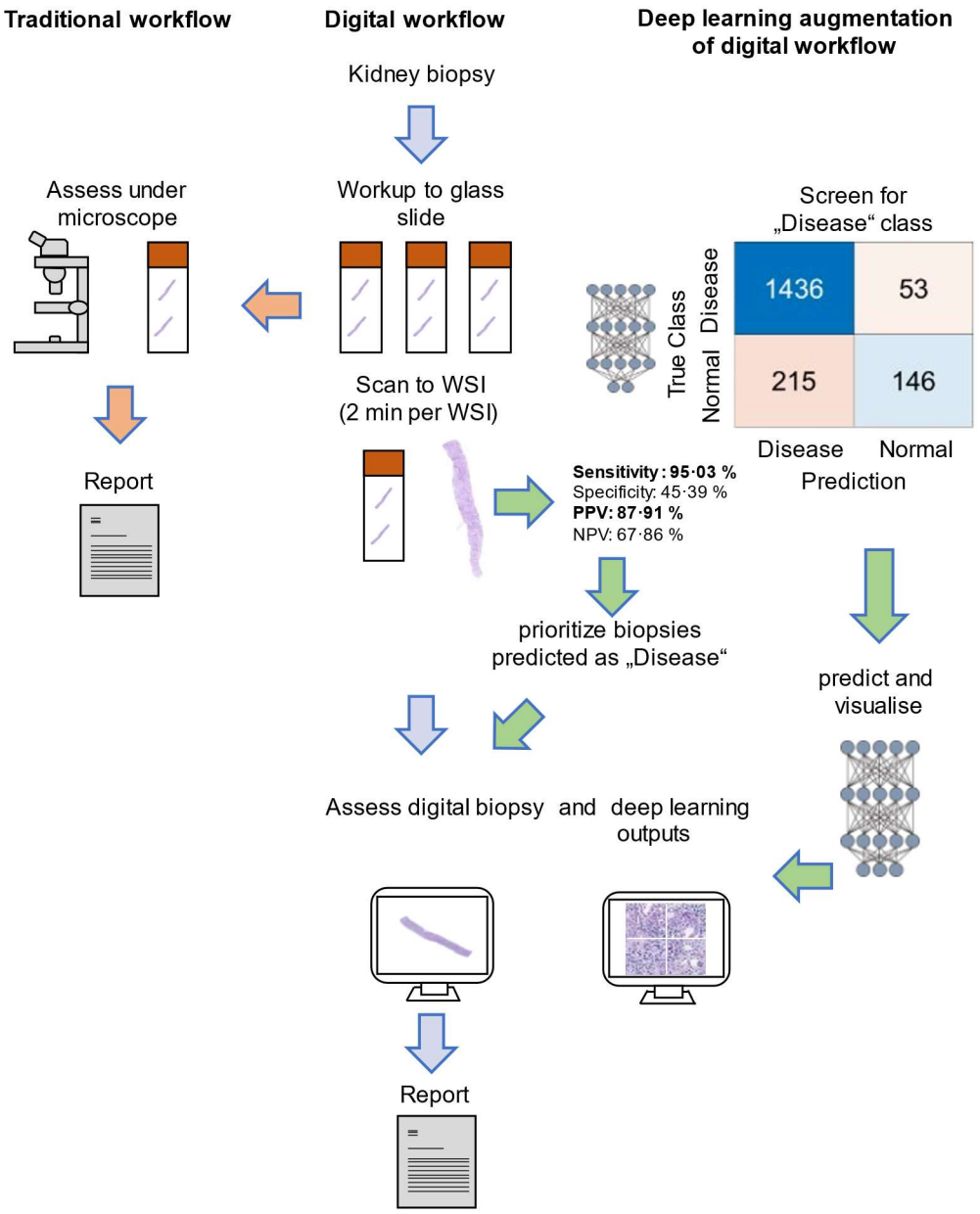
Supplementary Figure 10. Most predictive tiles from a patient correctly classified as belonging to the class "normal". The tiles show healthy kidney tubules with open lumina, brush border and no interstitial inflammation. Tiles from all three stains are included within this patient's most predictive tiles. Each tile has an edge length of 128 µm.



Supplementary Figure 11. Most predictive tiles from a patient correctly classified as belonging to the class “rejection”. The tiles show prominent interstitial inflammation, tubulitis and peritubular capillaries filled with leukocytes. Interestingly, no Jones-stain appears in this patient's most predictive tiles. Each tile has an edge length of 128 µm.



Supplementary Figure 12. Most predictive tiles from a patient correctly classified as belonging to the class “other diseases”. The tiles show interstitial fibrosis and intratubular material, some tiles show medulla. Each tile has an edge length of 128 µm.



Supplementary Figure 13. Potential clinical use scenario of the developed algorithms. Traditionally, kidney biopsies are processed to glass slides and assessed under the microscope by a nephropathologist. In digital pathology laboratories, these glass slides are digitised using whole slide scanners to whole slide images (WSI), which takes approx. 2 minutes per slide, and assessed at an imaging workstation. After scanning, these WSI could be worked up using deep learning approaches as depicted in this study, first identifying biopsies with pathological alterations and then subtyping these (e.g., into rejection or other diseases). Inference of a model takes less than ten minutes per case. Visualisation techniques, such as depicting the most predictive tiles or colour-coding the predictions on the WSI could assist in clinical assessment of the biopsies. PPV: Positive Predictive Value, NPV: Negative Predictive Value.

Supplementary Tables

Supplementary Table S1: Basic patient cohort characteristics.

	AMS	UMCU	AC-K
Recipient age in years (mean ± one standard deviation)	49 ± 15	50 ± 16	53 ± 13
F:M	1 : 1·6	1 : 1·5	1 : 1·24
Days from transplantation to biopsy (Median [Minimum, Maximum])	356 [0, 9960]	241 [1, 12909]	112 [0, 9311]
N included biopsies (WSIs)	1130 (3390)	717 (2151)	101 (303)
Number of Glomeruli per biopsy (mean ± one standard deviation)	16±11	20±12	13±7
N (%) Normal	226 (21)	121 (17)	29 (29)
N (%) Other Diseases	536 (47)	300 (42)	43 (42)
N (%) Rejection	368 (32)	296 (41)	29 (29)

AC-K: Kidney cohort from Aachen, UMCU: Kidney cohort from Utrecht, AMS: Kidney cohort from Amsterdam, F: female, M: male, N: number, WSI: Whole Slide Image.

Supplementary Table S2. Histopathological diagnoses of the included cases.

Diagnostic Class	Diagnosis	AMS	UMCU	AC-K
Other Diseases	ATI	61	64	18
	Chronic damage NOS	281	89	11
	BKVAN	55	18	3
	CMV nephritis	4	1	0
	BKVAN + CMV nephritis	1	0	0
	TMA	33	17	2
	Pyelonephritis/obstruction	13	17	1
	Eosinophilic TIN NOS	4	9	0
	Phosphate nephropathy	0	0	1
	Nephrocalcinosis	4	0	1
	Oxalate nephropathy	1	4	0
	Recurrent GN	44	14	2
	Recurrent FSGS	14	0	0
	CNI toxicity	12	37	0
	Cortical necrosis	2	4	0
	Hypertensive nephropathy	5	9	4
	Diabetic nephropathy	2	13	0
	AL amyloidosis	0	2	0
	PTLD	0	1	0

	Light chain proximal tubulopathy	0	1	0
Rejection	ABMR	91	24	5
	Borderline TCMR	40	101	11
	TCMR grade I	82	55	4
	TCMR grade II/III	54	21	4
	ABMR + TCMR	101	95	5

Number of individual diagnoses within the overarching classes used to train the neural networks. ATI: acute tubular injury, NOS: not otherwise specified, BKVAN: BK polyomavirus nephropathy, CMV: cytomegalovirus, TMA: thrombotic microangiopathy, GN: glomerulonephritis, FSGS: focal and segmental glomerulosclerosis, CNI: calcineurin inhibitor, ABMR: antibody-mediated rejection, TCMR: T cell-mediated rejection.

Supplementary Table S3. Banff 2019 Classification categories and their descriptions as used in this study.

Banff Category	Description
1 Normal biopsy or nonspecific changes	This category includes all biopsies that do not show pathology or only nonspecific changes. We allow nonspecific changes to include cases with minimal chronic damage in any of the kidney compartments (<25% glomerulosclerosis and/or interstitial fibrosis/tubular atrophy and/or maximum Banff cv1/ah1.
2 Antibody-mediated changes (ABMR)	This category includes all cases that show features of active ABMR and/or chronic-active ABMR with microvascular inflammation, tubular necrosis or thrombotic microangiopathy with detectable donor-specific antibodies (CDC and/or luminex single bead assay) and/or C4d positivity on peritubular capillaries in the clinical work-up. Autoantibodies or molecular diagnostics are not part of the routine work-up of biopsies in our hospitals.
3 Borderline (suspicious) for acute T cell-mediated rejection (TCMR)	This category includes cases with any score of tubulitis (t1, t2, t3) with mild inflammation (i1) or t1 tubulitis with moderate to severe interstitial inflammation not meeting the minimal criteria for acute TCMR and/or ABMR and/or other posttransplant diseases like BK polyomavirus nephropathy.
4 T cell-mediated rejection (TCMR)	This category includes acute TCMR grades I, II and III and all forms of chronic active TCMR
5 Polyomavirus nephropathy	This category includes all cases with a biopsy-proven (SV40 immunostaining+) BK polyomavirus nephropathy, not meeting the criteria for TCMR grade II/III and/or ABMR.

Supplementary Table S4. Diagnoses of misclassified cases by the single CNN in the AMS and UMCU cohorts.

Ground Truth Class	Pathology Diagnoses	Predicted as		
		Normal	Rejection	Other Diseases
Normal	Normal	NA	84	122
Rejection	ABMR	6	NA	61
	Borderline TCMR	4	NA	34
	Mixed Rejection	3	NA	48
	TCMR I	2	NA	36
	TCMR II or III	0	NA	14
Other Diseases	ATI	7	67	NA
	BKVAN	6	38	NA
	Chronic Damage NOS	29	59	NA
	CMV Nephritis	0	4	NA
	CNI toxicity	5	14	NA
	Cortical Necrosis	0	6	NA
	Diabetic Nephropathy	1	9	NA
	Hypertensive Nephropathy	0	3	NA
	LCPT	0	1	NA
	Nephrocalcinosis	1	1	NA
	Oxalate Nephropathy	0	3	NA
	PTLD	0	1	NA
	Pyelonephritis	1	17	NA
	Recurrent FSGS	2	1	NA
	Recurrent GN	4	13	NA
	TIN NOS	0	10	NA
	TMA	4	11	NA

ATI: acute tubular injury, NOS: not otherwise specified, BKVAN: BK polyomavirus nephropathy, CMV: cytomegalovirus, TMA: thrombotic microangiopathy, GN: glomerulonephritis, FSGS: focal and segmental glomerulosclerosis, CNI: calcineurin inhibitor, ABMR: antibody-mediated rejection, TCMR: T cell-mediated rejection. LCPT: Light Chain Proximal Tubulopathy. TIN: Tubulointerstitial Nephritis.

Supplementary Table S5. Areas under the receiver operating characteristic curve (AUROC) in subgroups based on glomerular number. All analyses were performed within the combined AMS and UMCU cohort.

Model	Target	0-5 Glomeruli	6-10 Glomeruli	11-15 Glomeruli	>15 Glomeruli
1st Serial	Normal	0·82	0·87	0·87	0·89
1st Serial	Disease	0·82	0·87	0·87	0·89
2nd Serial	Other Diseases	0·75	0·71	0·79	0·75
2nd Serial	Rejection	0·75	0·71	0·79	0·75
Single	Normal	0·80	0·87	0·84	0·87
Single	Other Diseases	0·68	0·69	0·72	0·70
Single	Rejection	0·78	0·75	0·80	0·79

Supplementary Table S6. Used inceptionv3 network structure for the first serial CNN.

LayerNumber	LayerName	LayerType	Description
1	'input_1'	Image Input	299x299x3 images with 'rescale-symmetric' normalization
2	'conv2d_1'	Convolution	32 3x3x3 convolutions with stride [2 2] and padding [0 0 0 0]
3	'batch_normalization_1'	Batch Normalization	Batch normalization with 32 channels
4	'activation_1_relu'	ReLU	ReLU
5	'conv2d_2'	Convolution	32 3x3x32 convolutions with stride [1 1] and padding [0 0 0 0]
6	'batch_normalization_2'	Batch Normalization	Batch normalization with 32 channels
7	'activation_2_relu'	ReLU	ReLU
8	'conv2d_3'	Convolution	64 3x3x32 convolutions with stride [1 1] and padding 'same'
9	'batch_normalization_3'	Batch Normalization	Batch normalization with 64 channels
10	'activation_3_relu'	ReLU	ReLU
11	'max_pooling2d_1'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]

12	'conv2d_4'	Convolution	80 1x1x64 convolutions with stride [1 1] and padding [0 0 0 0]
13	'batch_normalization_4'	Batch Normalization	Batch normalization with 80 channels
14	'activation_4_relu'	ReLU	ReLU
15	'conv2d_5'	Convolution	192 3x3x80 convolutions with stride [1 1] and padding [0 0 0 0]
16	'batch_normalization_5'	Batch Normalization	Batch normalization with 192 channels
17	'activation_5_relu'	ReLU	ReLU
18	'max_pooling2d_2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
19	'conv2d_9'	Convolution	64 1x1x192 convolutions with stride [1 1] and padding 'same'
20	'batch_normalization_9'	Batch Normalization	Batch normalization with 64 channels
21	'activation_9_relu'	ReLU	ReLU
22	'conv2d_7'	Convolution	48 1x1x192 convolutions with stride [1 1] and padding 'same'
23	'conv2d_10'	Convolution	96 3x3x64 convolutions with stride [1 1] and padding 'same'

24	'batch_normalization_7'	Batch Normalization	Batch normalization with 48 channels
25	'batch_normalization_10'	Batch Normalization	Batch normalization with 96 channels
26	'activation_7_relu'	ReLU	ReLU
27	'activation_10_relu'	ReLU	ReLU
28	'average_pooling2d_1'	Average Pooling	3x3 average pooling with stride [1 1] and padding 'same'
29	'conv2d_6'	Convolution	64 1x1x192 convolutions with stride [1 1] and padding 'same'
30	'conv2d_8'	Convolution	64 5x5x48 convolutions with stride [1 1] and padding 'same'
31	'conv2d_11'	Convolution	96 3x3x96 convolutions with stride [1 1] and padding 'same'
32	'conv2d_12'	Convolution	32 1x1x192 convolutions with stride [1 1] and padding 'same'
33	'batch_normalization_6'	Batch Normalization	Batch normalization with 64 channels
34	'batch_normalization_8'	Batch Normalization	Batch normalization with 64 channels
35	'batch_normalization_11'	Batch Normalization	Batch normalization with 96 channels

36	'batch_normalization_12'	Batch Normalization	Batch normalization with 32 channels
37	'activation_6_relu'	ReLU	ReLU
38	'activation_8_relu'	ReLU	ReLU
39	'activation_11_relu'	ReLU	ReLU
40	'activation_12_relu'	ReLU	ReLU
41	'mixed0'	Depth concatenation	Depth concatenation of 4 inputs
42	'conv2d_16'	Convolution	64 1x1x256 convolutions with stride [1 1] and padding 'same'
43	'batch_normalization_16'	Batch Normalization	Batch normalization with 64 channels
44	'activation_16_relu'	ReLU	ReLU
45	'conv2d_14'	Convolution	48 1x1x256 convolutions with stride [1 1] and padding 'same'
46	'conv2d_17'	Convolution	96 3x3x64 convolutions with stride [1 1] and padding 'same'
47	'batch_normalization_14'	Batch Normalization	Batch normalization with 48 channels
48	'batch_normalization_17'	Batch Normalization	Batch normalization with 96 channels
49	'activation_14_relu'	ReLU	ReLU

50	'activation_17_relu'	ReLU	ReLU
51	'average_pooling2d_2'	Average Pooling	3x3 average pooling with stride [1 1] and padding 'same'
52	'conv2d_13'	Convolution	64 1x1x256 convolutions with stride [1 1] and padding 'same'
53	'conv2d_15'	Convolution	64 5x5x48 convolutions with stride [1 1] and padding 'same'
54	'conv2d_18'	Convolution	96 3x3x96 convolutions with stride [1 1] and padding 'same'
55	'conv2d_19'	Convolution	64 1x1x256 convolutions with stride [1 1] and padding 'same'
56	'batch_normalization_13'	Batch Normalization	Batch normalization with 64 channels
57	'batch_normalization_15'	Batch Normalization	Batch normalization with 64 channels
58	'batch_normalization_18'	Batch Normalization	Batch normalization with 96 channels
59	'batch_normalization_19'	Batch Normalization	Batch normalization with 64 channels
60	'activation_13_relu'	ReLU	ReLU
61	'activation_15_relu'	ReLU	ReLU

62	'activation_18_relu'	ReLU	ReLU
63	'activation_19_relu'	ReLU	ReLU
64	'mixed1'	Depth concatenation	Depth concatenation of 4 inputs
65	'conv2d_23'	Convolution	64 1x1x288 convolutions with stride [1 1] and padding 'same'
66	'batch_normalization_23'	Batch Normalization	Batch normalization with 64 channels
67	'activation_23_relu'	ReLU	ReLU
68	'conv2d_21'	Convolution	48 1x1x288 convolutions with stride [1 1] and padding 'same'
69	'conv2d_24'	Convolution	96 3x3x64 convolutions with stride [1 1] and padding 'same'
70	'batch_normalization_21'	Batch Normalization	Batch normalization with 48 channels
71	'batch_normalization_24'	Batch Normalization	Batch normalization with 96 channels
72	'activation_21_relu'	ReLU	ReLU
73	'activation_24_relu'	ReLU	ReLU
74	'average_pooling2d_3'	Average Pooling	3x3 average pooling with stride [1 1] and padding 'same'

75	'conv2d_20'	Convolution	64 1x1x288 convolutions with stride [1 1] and padding 'same'
76	'conv2d_22'	Convolution	64 5x5x48 convolutions with stride [1 1] and padding 'same'
77	'conv2d_25'	Convolution	96 3x3x96 convolutions with stride [1 1] and padding 'same'
78	'conv2d_26'	Convolution	64 1x1x288 convolutions with stride [1 1] and padding 'same'
79	'batch_normalization_20'	Batch Normalization	Batch normalization with 64 channels
80	'batch_normalization_22'	Batch Normalization	Batch normalization with 64 channels
81	'batch_normalization_25'	Batch Normalization	Batch normalization with 96 channels
82	'batch_normalization_26'	Batch Normalization	Batch normalization with 64 channels
83	'activation_20_relu'	ReLU	ReLU
84	'activation_22_relu'	ReLU	ReLU
85	'activation_25_relu'	ReLU	ReLU
86	'activation_26_relu'	ReLU	ReLU
87	'mixed2'	Depth concatenation	Depth concatenation of 4 inputs

88	'conv2d_28'	Convolution	64 1x1x288 convolutions with stride [1 1] and padding 'same'
89	'batch_normalization_28'	Batch Normalization	Batch normalization with 64 channels
90	'activation_28_relu'	ReLU	ReLU
91	'conv2d_29'	Convolution	96 3x3x64 convolutions with stride [1 1] and padding 'same'
92	'batch_normalization_29'	Batch Normalization	Batch normalization with 96 channels
93	'activation_29_relu'	ReLU	ReLU
94	'conv2d_27'	Convolution	384 3x3x288 convolutions with stride [2 2] and padding [0 0 0 0]
95	'conv2d_30'	Convolution	96 3x3x96 convolutions with stride [2 2] and padding [0 0 0 0]
96	'batch_normalization_27'	Batch Normalization	Batch normalization with 384 channels
97	'batch_normalization_30'	Batch Normalization	Batch normalization with 96 channels
98	'activation_27_relu'	ReLU	ReLU
99	'activation_30_relu'	ReLU	ReLU

100	'max_pooling2d_3'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
101	'mixed3'	Depth concatenation	Depth concatenation of 3 inputs
102	'conv2d_35'	Convolution	128 1x1x768 convolutions with stride [1 1] and padding 'same'
103	'batch_normalization_35'	Batch Normalization	Batch normalization with 128 channels
104	'activation_35_relu'	ReLU	ReLU
105	'conv2d_36'	Convolution	128 7x1x128 convolutions with stride [1 1] and padding 'same'
106	'batch_normalization_36'	Batch Normalization	Batch normalization with 128 channels
107	'activation_36_relu'	ReLU	ReLU
108	'conv2d_32'	Convolution	128 1x1x768 convolutions with stride [1 1] and padding 'same'
109	'conv2d_37'	Convolution	128 1x7x128 convolutions with stride [1 1] and padding 'same'
110	'batch_normalization_32'	Batch Normalization	Batch normalization with 128 channels
111	'batch_normalization_37'	Batch Normalization	Batch normalization with 128 channels

112	'activation_32_relu'	ReLU	ReLU
113	'activation_37_relu'	ReLU	ReLU
114	'conv2d_33'	Convolution	128 1x7x128 convolutions with stride [1 1] and padding 'same'
115	'conv2d_38'	Convolution	128 7x1x128 convolutions with stride [1 1] and padding 'same'
116	'batch_normalization_33'	Batch Normalization	Batch normalization with 128 channels
117	'batch_normalization_38'	Batch Normalization	Batch normalization with 128 channels
118	'activation_33_relu'	ReLU	ReLU
119	'activation_38_relu'	ReLU	ReLU
120	'average_pooling2d_4'	Average Pooling	3x3 average pooling with stride [1 1] and padding 'same'
121	'conv2d_31'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'
122	'conv2d_34'	Convolution	192 7x1x128 convolutions with stride [1 1] and padding 'same'
123	'conv2d_39'	Convolution	192 1x7x128 convolutions with stride [1 1] and padding 'same'

124	'conv2d_40'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'
125	'batch_normalization_31'	Batch Normalization	Batch normalization with 192 channels
126	'batch_normalization_34'	Batch Normalization	Batch normalization with 192 channels
127	'batch_normalization_39'	Batch Normalization	Batch normalization with 192 channels
128	'batch_normalization_40'	Batch Normalization	Batch normalization with 192 channels
129	'activation_31_relu'	ReLU	ReLU
130	'activation_34_relu'	ReLU	ReLU
131	'activation_39_relu'	ReLU	ReLU
132	'activation_40_relu'	ReLU	ReLU
133	'mixed4'	Depth concatenation	Depth concatenation of 4 inputs
134	'conv2d_45'	Convolution	160 1x1x768 convolutions with stride [1 1] and padding 'same'
135	'batch_normalization_45'	Batch Normalization	Batch normalization with 160 channels
136	'activation_45_relu'	ReLU	ReLU

137	'conv2d_46'	Convolution	160 7x1x160 convolutions with stride [1 1] and padding 'same'
138	'batch_normalization_46'	Batch Normalization	Batch normalization with 160 channels
139	'activation_46_relu'	ReLU	ReLU
140	'conv2d_42'	Convolution	160 1x1x768 convolutions with stride [1 1] and padding 'same'
141	'conv2d_47'	Convolution	160 1x7x160 convolutions with stride [1 1] and padding 'same'
142	'batch_normalization_42'	Batch Normalization	Batch normalization with 160 channels
143	'batch_normalization_47'	Batch Normalization	Batch normalization with 160 channels
144	'activation_42_relu'	ReLU	ReLU
145	'activation_47_relu'	ReLU	ReLU
146	'conv2d_43'	Convolution	160 1x7x160 convolutions with stride [1 1] and padding 'same'
147	'conv2d_48'	Convolution	160 7x1x160 convolutions with stride [1 1] and padding 'same'
148	'batch_normalization_43'	Batch Normalization	Batch normalization with 160 channels

149	'batch_normalization_48'	Batch Normalization	Batch normalization with 160 channels
150	'activation_43_relu'	ReLU	ReLU
151	'activation_48_relu'	ReLU	ReLU
152	'average_pooling2d_5'	Average Pooling	3x3 average pooling with stride [1 1] and padding 'same'
153	'conv2d_41'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'
154	'conv2d_44'	Convolution	192 7x1x160 convolutions with stride [1 1] and padding 'same'
155	'conv2d_49'	Convolution	192 1x7x160 convolutions with stride [1 1] and padding 'same'
156	'conv2d_50'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'
157	'batch_normalization_41'	Batch Normalization	Batch normalization with 192 channels
158	'batch_normalization_44'	Batch Normalization	Batch normalization with 192 channels
159	'batch_normalization_49'	Batch Normalization	Batch normalization with 192 channels
160	'batch_normalization_50'	Batch Normalization	Batch normalization with 192 channels

161	'activation_41_relu'	ReLU	ReLU
162	'activation_44_relu'	ReLU	ReLU
163	'activation_49_relu'	ReLU	ReLU
164	'activation_50_relu'	ReLU	ReLU
165	'mixed5'	Depth concatenation	Depth concatenation of 4 inputs
166	'conv2d_55'	Convolution	160 1x1x768 convolutions with stride [1 1] and padding 'same'
167	'batch_normalization_55'	Batch Normalization	Batch normalization with 160 channels
168	'activation_55_relu'	ReLU	ReLU
169	'conv2d_56'	Convolution	160 7x1x160 convolutions with stride [1 1] and padding 'same'
170	'batch_normalization_56'	Batch Normalization	Batch normalization with 160 channels
171	'activation_56_relu'	ReLU	ReLU
172	'conv2d_52'	Convolution	160 1x1x768 convolutions with stride [1 1] and padding 'same'
173	'conv2d_57'	Convolution	160 1x7x160 convolutions with stride [1 1] and padding 'same'

174	'batch_normalization_52'	Batch Normalization	Batch normalization with 160 channels
175	'batch_normalization_57'	Batch Normalization	Batch normalization with 160 channels
176	'activation_52_relu'	ReLU	ReLU
177	'activation_57_relu'	ReLU	ReLU
178	'conv2d_53'	Convolution	160 1x7x160 convolutions with stride [1 1] and padding 'same'
179	'conv2d_58'	Convolution	160 7x1x160 convolutions with stride [1 1] and padding 'same'
180	'batch_normalization_53'	Batch Normalization	Batch normalization with 160 channels
181	'batch_normalization_58'	Batch Normalization	Batch normalization with 160 channels
182	'activation_53_relu'	ReLU	ReLU
183	'activation_58_relu'	ReLU	ReLU
184	'average_pooling2d_6'	Average Pooling	3x3 average pooling with stride [1 1] and padding 'same'
185	'conv2d_51'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'

186	'conv2d_54'	Convolution	192 7x1x160 convolutions with stride [1 1] and padding 'same'
187	'conv2d_59'	Convolution	192 1x7x160 convolutions with stride [1 1] and padding 'same'
188	'conv2d_60'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'
189	'batch_normalization_51'	Batch Normalization	Batch normalization with 192 channels
190	'batch_normalization_54'	Batch Normalization	Batch normalization with 192 channels
191	'batch_normalization_59'	Batch Normalization	Batch normalization with 192 channels
192	'batch_normalization_60'	Batch Normalization	Batch normalization with 192 channels
193	'activation_51_relu'	ReLU	ReLU
194	'activation_54_relu'	ReLU	ReLU
195	'activation_59_relu'	ReLU	ReLU
196	'activation_60_relu'	ReLU	ReLU
197	'mixed6'	Depth concatenation	Depth concatenation of 4 inputs
198	'conv2d_65'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'

199	'batch_normalization_65'	Batch Normalization	Batch normalization with 192 channels
200	'activation_65_relu'	ReLU	ReLU
201	'conv2d_66'	Convolution	192 7x1x192 convolutions with stride [1 1] and padding 'same'
202	'batch_normalization_66'	Batch Normalization	Batch normalization with 192 channels
203	'activation_66_relu'	ReLU	ReLU
204	'conv2d_62'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'
205	'conv2d_67'	Convolution	192 1x7x192 convolutions with stride [1 1] and padding 'same'
206	'batch_normalization_62'	Batch Normalization	Batch normalization with 192 channels
207	'batch_normalization_67'	Batch Normalization	Batch normalization with 192 channels
208	'activation_62_relu'	ReLU	ReLU
209	'activation_67_relu'	ReLU	ReLU
210	'conv2d_63'	Convolution	192 1x7x192 convolutions with stride [1 1] and padding 'same'

211	'conv2d_68'	Convolution	192 7x1x192 convolutions with stride [1 1] and padding 'same'
212	'batch_normalization_63'	Batch Normalization	Batch normalization with 192 channels
213	'batch_normalization_68'	Batch Normalization	Batch normalization with 192 channels
214	'activation_63_relu'	ReLU	ReLU
215	'activation_68_relu'	ReLU	ReLU
216	'average_pooling2d_7'	Average Pooling	3x3 average pooling with stride [1 1] and padding 'same'
217	'conv2d_61'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'
218	'conv2d_64'	Convolution	192 7x1x192 convolutions with stride [1 1] and padding 'same'
219	'conv2d_69'	Convolution	192 1x7x192 convolutions with stride [1 1] and padding 'same'
220	'conv2d_70'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'
221	'batch_normalization_61'	Batch Normalization	Batch normalization with 192 channels

222	'batch_normalization_64'	Batch Normalization	Batch normalization with 192 channels
223	'batch_normalization_69'	Batch Normalization	Batch normalization with 192 channels
224	'batch_normalization_70'	Batch Normalization	Batch normalization with 192 channels
225	'activation_61_relu'	ReLU	ReLU
226	'activation_64_relu'	ReLU	ReLU
227	'activation_69_relu'	ReLU	ReLU
228	'activation_70_relu'	ReLU	ReLU
229	'mixed7'	Depth concatenation	Depth concatenation of 4 inputs
230	'conv2d_73'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'
231	'batch_normalization_73'	Batch Normalization	Batch normalization with 192 channels
232	'activation_73_relu'	ReLU	ReLU
233	'conv2d_74'	Convolution	192 1x7x192 convolutions with stride [1 1] and padding 'same'
234	'batch_normalization_74'	Batch Normalization	Batch normalization with 192 channels
235	'activation_74_relu'	ReLU	ReLU

236	'conv2d_71'	Convolution	192 1x1x768 convolutions with stride [1 1] and padding 'same'
237	'conv2d_75'	Convolution	192 7x1x192 convolutions with stride [1 1] and padding 'same'
238	'batch_normalization_71'	Batch Normalization	Batch normalization with 192 channels
239	'batch_normalization_75'	Batch Normalization	Batch normalization with 192 channels
240	'activation_71_relu'	ReLU	ReLU
241	'activation_75_relu'	ReLU	ReLU
242	'conv2d_72'	Convolution	320 3x3x192 convolutions with stride [2 2] and padding [0 0 0 0]
243	'conv2d_76'	Convolution	192 3x3x192 convolutions with stride [2 2] and padding [0 0 0 0]
244	'batch_normalization_72'	Batch Normalization	Batch normalization with 320 channels
245	'batch_normalization_76'	Batch Normalization	Batch normalization with 192 channels
246	'activation_72_relu'	ReLU	ReLU
247	'activation_76_relu'	ReLU	ReLU

248	'max_pooling2d_4'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
249	'mixed8'	Depth concatenation	Depth concatenation of 3 inputs
250	'conv2d_81'	Convolution	448 1x1x1280 convolutions with stride [1 1] and padding 'same'
251	'batch_normalization_81'	Batch Normalization	Batch normalization with 448 channels
252	'activation_81_relu'	ReLU	ReLU
253	'conv2d_78'	Convolution	384 1x1x1280 convolutions with stride [1 1] and padding 'same'
254	'conv2d_82'	Convolution	384 3x3x448 convolutions with stride [1 1] and padding 'same'
255	'batch_normalization_78'	Batch Normalization	Batch normalization with 384 channels
256	'batch_normalization_82'	Batch Normalization	Batch normalization with 384 channels
257	'activation_78_relu'	ReLU	ReLU
258	'activation_82_relu'	ReLU	ReLU
259	'conv2d_79'	Convolution	384 1x3x384 convolutions with stride [1 1] and padding 'same'

260	'conv2d_80'	Convolution	384 3x1x384 convolutions with stride [1 1] and padding 'same'
261	'conv2d_83'	Convolution	384 1x3x384 convolutions with stride [1 1] and padding 'same'
262	'conv2d_84'	Convolution	384 3x1x384 convolutions with stride [1 1] and padding 'same'
263	'average_pooling2d_8'	Average Pooling	3x3 average pooling with stride [1 1] and padding 'same'
264	'conv2d_77'	Convolution	320 1x1x1280 convolutions with stride [1 1] and padding 'same'
265	'batch_normalization_79'	Batch Normalization	Batch normalization with 384 channels
266	'batch_normalization_80'	Batch Normalization	Batch normalization with 384 channels
267	'batch_normalization_83'	Batch Normalization	Batch normalization with 384 channels
268	'batch_normalization_84'	Batch Normalization	Batch normalization with 384 channels
269	'conv2d_85'	Convolution	192 1x1x1280 convolutions with stride [1 1] and padding 'same'
270	'batch_normalization_77'	Batch Normalization	Batch normalization with 320 channels

271	'activation_79_relu'	ReLU	ReLU
272	'activation_80_relu'	ReLU	ReLU
273	'activation_83_relu'	ReLU	ReLU
274	'activation_84_relu'	ReLU	ReLU
275	'batch_normalization_85'	Batch Normalization	Batch normalization with 192 channels
276	'activation_77_relu'	ReLU	ReLU
277	'mixed9_0'	Depth concatenation	Depth concatenation of 2 inputs
278	'concatenate_1'	Depth concatenation	Depth concatenation of 2 inputs
279	'activation_85_relu'	ReLU	ReLU
280	'mixed9'	Depth concatenation	Depth concatenation of 4 inputs
281	'conv2d_90'	Convolution	448 1x1x2048 convolutions with stride [1 1] and padding 'same'
282	'batch_normalization_90'	Batch Normalization	Batch normalization with 448 channels
283	'activation_90_relu'	ReLU	ReLU
284	'conv2d_87'	Convolution	384 1x1x2048 convolutions with stride [1 1] and padding 'same'

285	'conv2d_91'	Convolution	384 3x3x448 convolutions with stride [1 1] and padding 'same'
286	'batch_normalization_87'	Batch Normalization	Batch normalization with 384 channels
287	'batch_normalization_91'	Batch Normalization	Batch normalization with 384 channels
288	'activation_87_relu'	ReLU	ReLU
289	'activation_91_relu'	ReLU	ReLU
290	'conv2d_88'	Convolution	384 1x3x384 convolutions with stride [1 1] and padding 'same'
291	'conv2d_89'	Convolution	384 3x1x384 convolutions with stride [1 1] and padding 'same'
292	'conv2d_92'	Convolution	384 1x3x384 convolutions with stride [1 1] and padding 'same'
293	'conv2d_93'	Convolution	384 3x1x384 convolutions with stride [1 1] and padding 'same'
294	'average_pooling2d_9'	Average Pooling	3x3 average pooling with stride [1 1] and padding 'same'
295	'conv2d_86'	Convolution	320 1x1x2048 convolutions with stride [1 1] and padding 'same'

296	'batch_normalization_88'	Batch Normalization	Batch normalization with 384 channels
297	'batch_normalization_89'	Batch Normalization	Batch normalization with 384 channels
298	'batch_normalization_92'	Batch Normalization	Batch normalization with 384 channels
299	'batch_normalization_93'	Batch Normalization	Batch normalization with 384 channels
300	'conv2d_94'	Convolution	192 1x1x2048 convolutions with stride [1 1] and padding 'same'
301	'batch_normalization_86'	Batch Normalization	Batch normalization with 320 channels
302	'activation_88_relu'	ReLU	ReLU
303	'activation_89_relu'	ReLU	ReLU
304	'activation_92_relu'	ReLU	ReLU
305	'activation_93_relu'	ReLU	ReLU
306	'batch_normalization_94'	Batch Normalization	Batch normalization with 192 channels
307	'activation_86_relu'	ReLU	ReLU
308	'mixed9_1'	Depth concatenation	Depth concatenation of 2 inputs
309	'concatenate_2'	Depth concatenation	Depth concatenation of 2 inputs

310	'activation_94_relu'	ReLU	ReLU
311	'mixed10'	Depth concatenation	Depth concatenation of 4 inputs
312	'avg_pool'	Global Average Pooling	Global average pooling
313	'fc'	Fully Connected	2 fully connected layer
314	'softmax'	Softmax	softmax
315	'classoutput'	Classification Output	crossentropyex with classes 'Disease' and 'Normal'

Supplementary References

1 Bankhead P, Loughrey MB, Fernandez JA, *et al.* QuPath: Open source software for digital pathology image analysis. *Sci Rep* 2017; **7**: 16878.

2 Muti HS, Loeffler C, Echle A, *et al.* The Aachen Protocol for Deep Learning Histopathology: A hands-on guide for data preprocessing. 2020 DOI:10.5281/zenodo.3694994.

3 He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770–8.

4 Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818–26.

5 Zhang X, Zhou X, Lin M, Sun J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6848–56.

6 Deng J, Dong W, Socher R, Li L, Kai Li, Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009: 248–55.

7 Kather JN, Heij LR, Grabsch HI, *et al.* Pan-cancer image-based detection of clinically actionable genetic alterations. *Nature Cancer* 2020; **1**: 789–99.

8 Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int J Comput Vis* 2020; **128**: 336–59.

9 Grad-CAM reveals the why behind deep learning decisions - MATLAB & Simulink - MathWorks Deutschland. https://de.mathworks.com/help/deeplearning/ug/gradcam-explains-why.html (accessed March 9, 2021).