








Software/Web server Article

GNCnn: A QuPath extension for glomerulosclerosis and glomerulonephritis characterization based on deep learning

Israel Mateos-Aparicio-Ruiz ^{a, }, Anibal Pedraza ^{a, }, Jan Ulrich Becker ^{b, }, Nicola Altini ^c,
Jesus Salido ^{a, }, Gloria Bueno ^{a, },*

^a VISILAB Group, Universidad de Castilla-La Mancha, Av. Camilo José Cela, Ciudad Real, 13071, Ciudad Real, Spain

^b Institute of Pathology, University Hospital of Cologne, Cologne, Germany

^c Department of Electrical and Information Engineering (DEI), Polytechnic University of Bari, Via Edoardo Orabona n.4, Bari, 70126, Italy

ARTICLE INFO

Dataset link: <https://github.com/UCLM-VISILAB/qupath-extension-gncnn>

Keywords:

Glomerulosclerosis classification
Glomerulonephritis classification
QuPath
Digital pathology
Deep learning
Artificial intelligence

ABSTRACT

The digitalization of traditional glass slide microscopy into whole slide images has opened up new opportunities for pathology, such as the application of artificial intelligence techniques. Specialized software is necessary to visualize and analyze these images. One of these applications is QuPath, a popular bioimage analysis tool. This study proposes GNCnn, the first open-source QuPath extension specifically designed for nephropathology. It integrates deep learning models to provide nephropathologists with an accessible, automatic detector and classifier of glomeruli, the basic filtering units of the kidneys. The aim is to offer nephropathologists a freely available application to measure and analyze glomeruli to identify conditions such as glomerulosclerosis and glomerulonephritis. GNCnn offers a user-friendly interface that enables nephropathologists to detect glomeruli with high accuracy (Dice coefficient of 0.807) and categorize them as either sclerotic or non-sclerotic, achieving a balanced accuracy of 98.46%. Furthermore, it facilitates the classification of non-sclerotic glomeruli into 12 commonly diagnosed types of glomerulonephritis, with a top-3 balanced accuracy of 84.41%. GNCnn provides real-time updates of results, which are available at both the glomerulus and slide levels. This allows users to complete a typical analysis task without leaving the main application, QuPath. This tool is the first to integrate the entire workflow for the assessment of glomerulonephritis directly into the nephropathologists' workspace, accelerating and supporting their diagnosis.

1. Introduction

Glomeruli are a kidney tissue compartment and the primary filter units generating urine. As a high-pressure capillary filter, they are often involved in autoimmune inflammatory diseases called glomerulonephritis. Depending on the histological pattern and the etiology, glomerulonephritis can be diagnosed in various categories [1–4]. Although nephropathology on standard paraffin histology (including periodic-acid Schiff - PAS staining), immunostaining and electron microscopy, is considered the gold standard for diagnosis, often clinical data and even genetic data are required to reach the correct glomerulonephritis diagnostic category.

Recent advances in digital pathology have revolutionized the field by transforming traditional glass slide microscopy into digital Whole Slide Images (WSIs), enabling advanced image analysis and data management. Applications of this technological advancement include telepathology, education, and image exchange for research [5]. This digital transformation has also opened up new opportunities, such as the application of artificial intelligence (AI) techniques in the analysis of the WSIs [6,7]. In particular, with current hardware advances, deep learning has proven to be a promising subfield of AI.

WSIs present a pyramidal structure, where each layer corresponds to a magnification level. A typical WSI captured with a $\times 20$ magnification could use more than 20 GB of storage if uncompressed, and hundreds of megabytes after compression [8]. Specialized software is

* Corresponding author.

E-mail address: gloria.bueno@uclm.es (G. Bueno).

<https://doi.org/10.1016/j.csbj.2024.11.049>

Received 29 September 2024; Received in revised form 29 November 2024; Accepted 29 November 2024

necessary to visualize and analyze these images [9,10]. One of the first image-processing applications to support microscopy images, and therefore WSIs, was ImageJ [11], a Java-based image processing application that provides extensibility via plugins. Inspired by ImageJ, QuPath represents a 2nd generation application supporting digital pathology and WSI analysis [12,13].

QuPath also provides extensibility via plugins, some of which employ deep learning algorithms. Most of these plugins focus on cell segmentation. For example, Cellpose [14] includes the Cellpose [15,16] and Omnipose [17] algorithms, while the QuANTUM pipeline [18] enables molecular analysis. Additional segmentation tools include SAM [19], which utilizes the Segment Anything Model [20] (also available in a mobile version [21]), and StarDist [22], which localizes nuclei using star-convex polygons.

Other QuPath extensions provide more general functions, such as WSInfer [23], which implements a pipeline to run any patch-based classification models. Conversely, some extensions focus on more specific topics, such as bone marrow cellularity assessment [24] or the integration of spatial transcriptomics information [25] and Large Language Models [26]. Despite the significant impact of digital pathology on this subspecialty [27], to the best of the authors' knowledge, no QuPath extension has been specifically developed for nephropathology. The only existing work is the pipeline called MEScnn (MESc classification by neural network, where MESc stands for mesangial hypercellularity (M), endocapillary hypercellularity (E), segmental sclerosis (S), and active crescents (C)). MEScnn serves as an interface between QuPath and Python. It runs entirely within Python but can save annotations to a specified QuPath project, facilitated through the module QIGS (QuPath Interface for Glomeruli Segmentation) [28]. Thus, MEScnn is not, strictly speaking, a QuPath extension.

In addition to ImageJ and QuPath, there are other WSI analysis software tools with large communities. One of these applications is Digital Slide Archive (DSA) [29], a web-based platform that integrates preprocessing, segmentation and feature extraction algorithms provided by HistomicsTK, a Python toolkit for WSI analysis. Orbit Image Analysis [30] is a Java software that integrates different machine learning algorithms for pixel classification, object segmentation, and object classification. A more complex software is Cytomine [31], which uses many technologies, but its deployment is automated through the use of Docker containers.

Despite the significant progress in general digital pathology tools, a limited number are specifically tailored for nephropathology. For instance, Histo-Cloud [32] provides cloud-based segmentation of glomeruli and other renal structures, while PodoSighter [33] is designed specifically for podocyte segmentation. Although these tools address essential segmentation tasks in nephropathology, they operate as standalone cloud-based applications rather than as components of a multifunctional platform like QuPath, which offers a broader range of functionalities. This gap highlights the need for a comprehensive tool that integrates real-time segmentation and classification within a unified platform. Such a tool would further support the integration of deep learning applications specifically tailored for nephropathology.

Glomeruli segmentation is the most common application of deep learning in nephropathology, particularly through the application of convolutional neural networks (CNNs). For this purpose, in [34,35], the task was considered an instance segmentation problem, and state-of-the-art results were achieved by leveraging Region Proposal Networks (RPNs). In [36], instead, a two-step approach was followed, first detecting bounding boxes, and then segmenting glomeruli within them. Other works not only segment glomeruli, but also classify them. Another usual task in deep learning applications to nephropathology is the classification of glomeruli into sclerotic or non-sclerotic categories. Recently, CNNs have been deployed for this classification [37–39].

Other studies focus on more specific classification schemes, such as the Oxford classification [2–4] for IgAN (immunoglobulin A-associated nephropathy) [28]. Nevertheless, a variety of classifications exist for

glomerular pathologies, and specifically, glomerulonephritis can be further divided into several subtypes. Glomerulonephritis is a glomerular injury with ensuing glomerular inflammation that is characterized by increased glomerular cellularity [40]. Different classifications exist for these pathologies according to histological patterns [41], etiology [40], and other factors such as pathogenesis, activity and chronicity [42]. Therefore, glomerulonephritis is often a source of confusion even among experts. Nephropathologists typically combine conventional microscopy techniques, such as H&E (Hematoxylin and Eosin) or PAS (Periodic Acid–Schiff) staining, with immunofluorescence, electron microscopy, clinical information, and genetic studies to make a final diagnosis. However, access to many of these techniques remains limited for patients around the world.

Despite these advancements, the application of deep learning in glomerulus characterization is not without limitations, particularly concerning dataset availability and quality. The research datasets currently commonly used are small, not publicly available, and come from a single institution [43,44]. This limitation can hinder the generalizability of the trained models and the reliability of their evaluation. A robust model for a limited dataset does not necessarily imply that it provides the best performance compared to other models [45]. Therefore, while some aforementioned models have achieved state-of-the-art performance on reduced datasets, their generalizability might be questioned due to the small size of these datasets.

While the predominant trend in AI applications within nephropathology focuses on deep learning algorithms for characterizing renal structures in PAS-stained WSIs [43], some studies are also leveraging immunofluorescence images to provide a more comprehensive diagnostic assessment [46,47]. Beyond diagnosis, AI is also applied to prognosis prediction, a promising area for anticipating disease progression and patient outcomes. For instance, in [48], CNNs are used to predict 1-, 3-, and 5-year renal survival rates, while estimated glomerular filtration rate is the primary target in [49,50]. Additionally, the development of platforms like Smartpath^k [51], a remote tool to teach glomerulopathies using machine learning, is also supporting the education of nephropathologists.

However, despite these promising advancements, several challenges must be addressed for the adoption of AI in nephropathology, particularly around regulatory issues. As a result, WSI-analysis tools like QuPath are typically limited to research. While the main areas of research using QuPath are oncology, cell biology and pathology, its adoption is often restricted to biomarker quantification [13]. Furthermore, it has recently been applied in the clinical domain [52,53]. In general, AI algorithms face restrictions due to strict regulations from bodies such as the US Food and Drug Administration (FDA). Nevertheless, the deep learning algorithm described in [54] is implemented in Paige Prostate, a software system for the assessment of prostate cancer, which has recently been granted Breakthrough Designation by the US FDA [55]. This approval of an AI-based image analysis algorithm may serve as a model for similar algorithms developed for renal pathology.

To address the need for accessible, efficient and comprehensive tools in nephropathology, we present GNCnn (short for GlomeruloNephritis Classification by neural network), a free plugin specifically developed for QuPath. This tool leverages deep learning methods to automate the preprocessing of PAS-stained WSIs, glomeruli detection and classification, distinguishing between sclerotic and non-sclerotic, and further categorizing non-sclerotic glomeruli into 12 common glomerulonephritis diagnosis. GNCnn addresses limitations in access to advanced diagnostic techniques by relying solely on PAS-stained LM images. By integrating GNCnn as a QuPath extension, we aim to offer nephropathologists a comprehensive solution for the characterization of glomeruli.

Table 1

Renal pathologies (e.g., glomerulonephritis) classified in the second stage of the GNCnn tool. For those with an acronym as their label, the letters forming the acronym are capitalized in the first column.

Complete name	Label
Anti-glomerular Basement Membrane antibody GlomeruloNephritis	AMBGN
Anti-Neutrophil Cytoplasmic Antibody-associated glomerulonephritis	ANCA
C3-GlomeruloNephritis	C3-GN
Cryoglobulinemic GlomeruloNephritis	CryoglobulinemicGN
Dense Deposit Disease	DDD
Fibrillary glomerulonephritis	Fibrillary
Infection-associated GlomeruloNephritis	IAGN
Immunoglobulin A-associated GlomeruloNephritis	IgAGN
Membranous nephropathy	Membranous
idiopathic MembranoProliferative GlomeruloNephritis	MPGN
Proliferative GlomeruloNephritis with Monoclonal Immunoglobulin Deposits	PGNMID
Systemic Lupus Erythematosus-associated GlomeruloNephritis class IV	SLEGN-IV

2. Methods

2.1. Overview

The GNCnn tool is a user-friendly extension for QuPath, designed and implemented to automatically detect and classify glomeruli in WSIs. GNCnn integrates the pre-processing of the WSI, along with the detection and classification of glomeruli, adding them as real-time annotations to the image in QuPath. The classification is made in two stages: first, glomeruli are classified into sclerotic or non-sclerotic categories; lastly, the non-sclerotic glomeruli are classified into 12 of the most common diagnoses of glomerulonephritis, listed in Table 1. The user is allowed to run either the first stage or both stages. Moreover, the user may choose only to detect glomeruli without classifying them. The extension also provides a visualization tool to analyze WSI-level results after detection and classification.

Some examples of glomeruli for each class are shown in Fig. 1. The figure illustrates the variability in texture and color features among glomeruli, even within the same class. It also highlights the similarity in morphological patterns across different classes, which can make distinguishing between them challenging. These subtle morphological differences, combined with the variations in color and texture despite using the same staining technique, underscore the complexity involved in accurate diagnosis.

The main code for the extension is developed in Java, using the “JavaFX” library for the GUI (Graphical User Interface), and asynchronous tasks, represented by Java’s “Task” class, for each of the steps in the detection and classification pipeline. The deep learning methods employed for the detection and classification, as well as the tissue detection algorithm, are implemented as Python scripts. The specific libraries used in each Python script will be detailed in their respective sections below.

The main workflow for GNCnn consists of the following steps, as illustrated in Fig. 2: 1) Tissue detection, 2) Tile exporting, 3) Glomeruli detection, 4) Exporting glomerular crops, 5) Glomeruli classification into a) Sclerotic and Non-Sclerotic, and optionally b) 12 diagnoses of glomerulonephritis for the non-sclerotic glomeruli.

QuPath allows two ways of working with images. Typically, a set of WSIs is grouped into a project. However, it is also possible to work with individual images. GNCnn supports both options. Since a project may consist of a considerable number of images, GNCnn provides an image selection tool that allows detection and/or classification to be applied to only a subset of the images in the project.

2.2. Task management

The glomeruli characterization pipeline implemented in GNCnn is based on a task management system that handles each processing step. Each step is represented by an asynchronous task, leveraging Java’s “Task” class. In this way, the execution of the process steps is decoupled from the main application thread, allowing both QuPath and the GNCnn extension to remain responsive to user interactions. Otherwise, the JavaFX application thread would be blocked.

Task management is centralized in a single class called “TaskManager”, which contains a single-thread executor pool as an attribute. Each step depends on the preceding one, and as a result, the sequential order of tasks must be maintained. This is accomplished by running only one thread at a time. Additionally, “TaskManager” acts as an intermediary layer that abstracts the individual tasks from the UI (User Interface) controller, which focuses exclusively on managing events triggered by user interactions.

In order to manage the connection between the output of one task and the input of another, temporary files are employed. A temporary folder is created within the project’s directory. If the user is working with a single image instead of a project, the temporary folder is created in the same location as the image.

Within the temporary folder, a separate folder is created for each task. Furthermore, for projects involving multiple images, a folder is created for each image inside the respective task folder. Temporary files generated during the process are automatically deleted when they are no longer needed for subsequent tasks. This ensures that the disk storage space used during execution is minimized and fully released upon completion.

Furthermore, some steps require executing Python code using specific libraries tailored to each task: tissue detection is primarily performed with OpenCV [56], glomeruli detection relies on a Detectron2 [57] model for inference on Linux (with a Torchscript [58] equivalent for Windows and macOS), and glomeruli classification is performed using MMClassification [59]. These steps are highlighted in Fig. 2. These scripts are run as processes through a “VirtualEnvironment” class, which provides an additional layer of abstraction. This class is responsible for building the process using Java’s “ProcessBuilder” class, executing the appropriate command based on the operating system, waiting for its completion, and updating the global progress based on the process status.

Finally, each component of GNCnn utilizes a variety of QuPath API functions to interact with the main application. The primary integration point is QuPath’s “QuPathExtension” interface, which GNCnn implements to be recognized as an extension by QuPath. Given the large number of API functions used across different components, a detailed explanation of each interaction would risk overwhelming readers and detracting from the main focus. Instead, a schematic representation of the interaction between GNCnn’s components is provided in Fig. 3, and a complete class diagram is included in Appendix A (Fig. 9).

2.3. Tissue detection

The first image preprocessing step involves detecting tissue pixels to exclude background pixels in subsequent steps, thereby speeding up the process. The tool exports a low-resolution version of the WSI using functions from the QuPath library for Java. This image is then used as input by a Python script, which the tool launches as a separate process. The script utilizes classic image processing techniques, such as morphological operations and Otsu’s thresholding [60], using OpenCV [61].

The steps of the implemented tissue detection algorithm, illustrated in Fig. 4, are as follows:

1. Exporting low-resolution image: QuPath provides functionality to export regions of WSIs, applying a specified downsampling factor. In this step, a downsampling factor of 20 is applied, meaning that

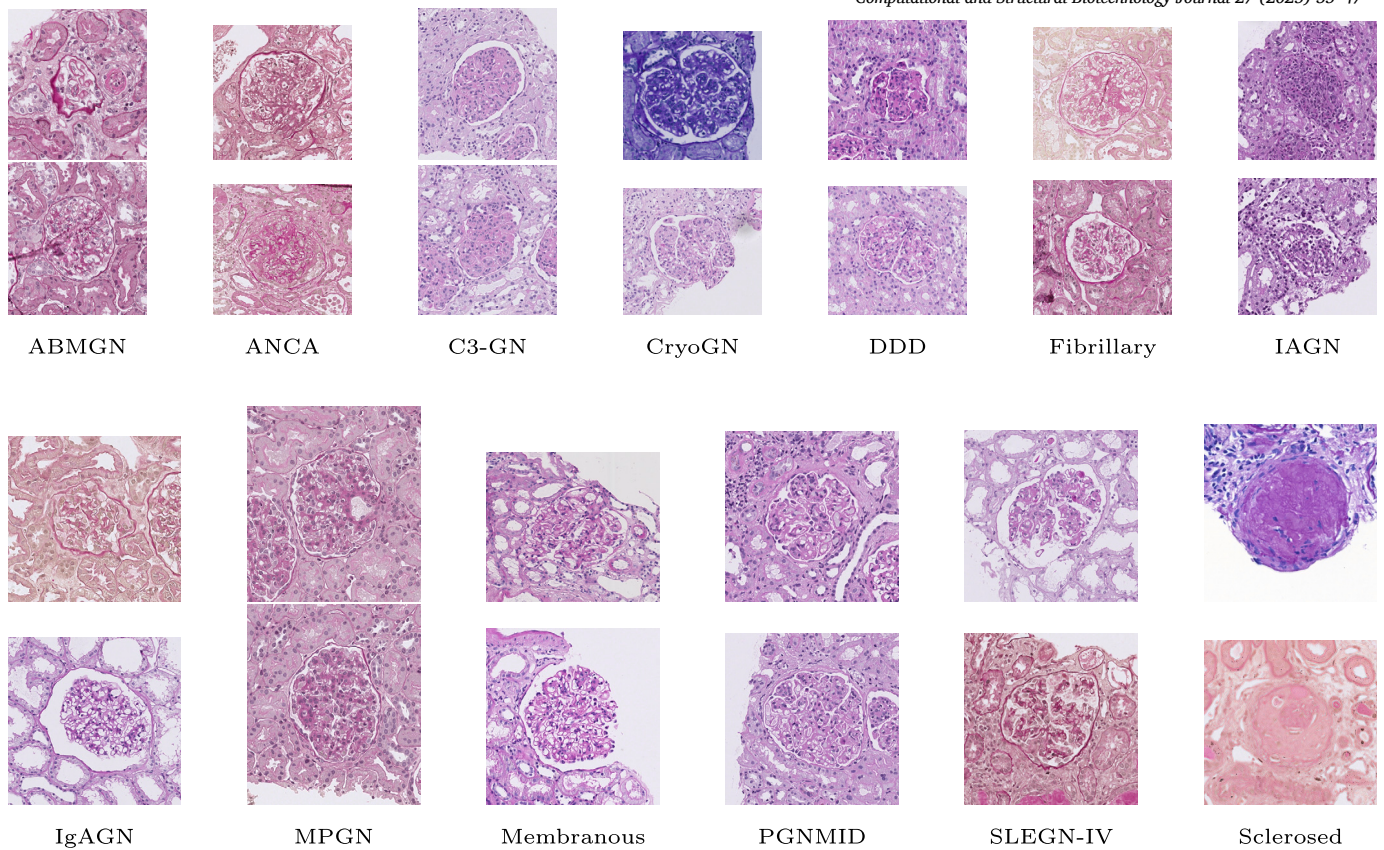


Fig. 1. Example images of each class of glomeruli. Two glomeruli from each class are shown.

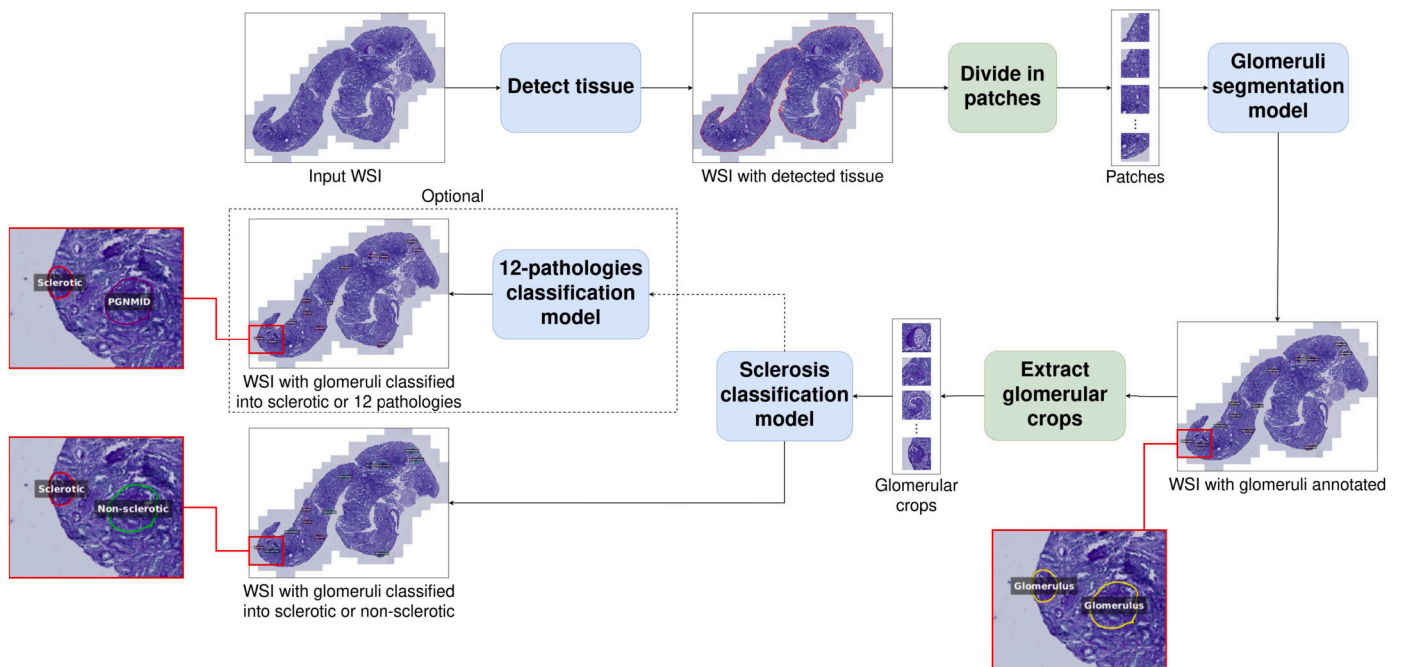


Fig. 2. A schematic representation of the workflow of GNCnn. Steps that involve launching and handling Python processes are highlighted in blue.

the exported image is 20 times smaller compared to the original WSI, making it suitable for further processing in the next steps.

2. Median filtering: it involves replacing each pixel's value with the median value of its neighboring pixels. This step helps to reduce

noise in images and is particularly useful in WSIs that exhibit abrupt changes in background colors.

3. Saturation channel extraction from HSV space: the image, originally in RGB space, is converted to HSV space. Extracting the saturation

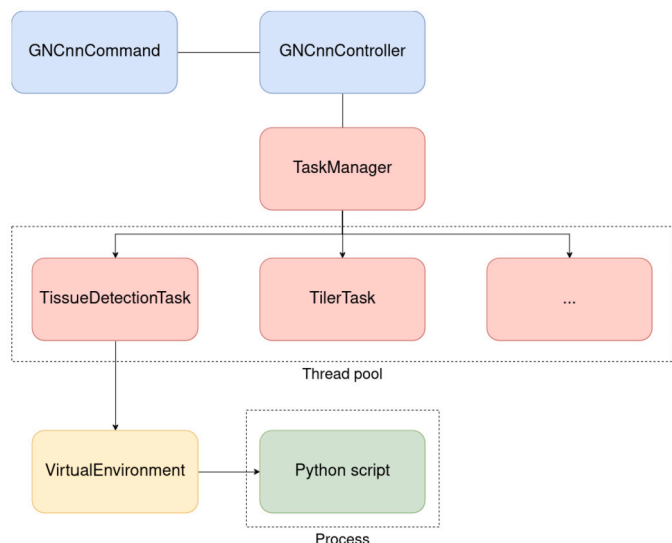


Fig. 3. A schematic representation of the interaction between components of GNCnn.

channel produces a grayscale image, which is needed for the next step.

4. Otsu's method: it is a thresholding technique that separates a grayscale image into two distinct classes. The optimal threshold is determined by minimizing the intra-class intensity variance while maximizing the inter-class variance.
5. Closing: it is a morphological operation that involves applying dilation followed by erosion, in that specific order. Dilation expands the boundaries of objects in an image, making them larger, while erosion contracts these boundaries, making the objects smaller. As a result, closing fills small holes and gaps in the image while preserving the shape and size of larger objects.
6. Rescaling and annotation export as GeoJSON: after all operations have been applied, the coordinates of the tissue segment contours are rescaled to their original size by multiplying them by 20, and then exported as GeoJSON annotations. GeoJSON is a format for encoding geospatial data based on JSON [62]. This format is compatible with QuPath for importing annotations.

After the Python process completes, the output GeoJSON file is imported into QuPath, and "Tissue"-class annotations are added to the image. These annotations will be used in the next step. Once this stage is finished, the temporary files corresponding to the low-resolution images are deleted from the disk.

2.4. Tile exporting

WSIs typically range from tens of thousands to hundreds of thousands of pixels. The computational cost of using images of this size as input for deep learning models is prohibitive. To address this, WSIs are divided into smaller patches using a sliding window approach with QuPath's "TileExporter" function, applied only to regions containing annotations. In our case, the annotations correspond to the Tissue category, generated in the previous step. The patches are 4096×4096 pixels in size, with a 2048-pixel overlap.

The 2048-pixel overlap, while doubling the number of patches, is essential for preserving the integrity of glomeruli that may lie across tile boundaries. Without this overlap, glomeruli located near the edges of a tile could be split across two or more tiles, leading to incomplete or fragmented detections. By using a sliding window approach with overlap, we ensure that glomeruli near the edges are fully contained within at least one tile, supporting accurate detection and minimizing the risk of false negatives.

Overlapping annotations are merged into a single one during the glomeruli detection stage, using a variant of the Non-Maximum Suppression (NMS) algorithm called Non-Maximum-Area Suppression (NMAS) [38]. This approach will be explained in more detail in the glomeruli detection section.

When this stage is finished, the "Tissue" annotations in the WSI are removed. The patches are exported in JPEG format to a temporary directory, which will then be passed as an argument to the glomeruli detection Python script.

2.5. Glomeruli detection

The glomeruli detection stage is carried out by another Python script. Similar to the tissue detection script, the glomeruli detection script is launched as a process from the corresponding "Task" thread in GNCnn. This script is based on the QuPath interface for Python, called QIGS (QuPath Interface for Glomeruli Segmentation) [28].

The model used in this stage is an instance segmentation model, which not only detects but also outlines the glomeruli. The architecture consists of a Cascade Mask R-CNN [63] model employing a ResNet-50 [64] backbone, implemented using Detectron2 [57]. The backbone network extracts deep features, and is followed by RPNs to propose candidate object regions. Cascade Mask R-CNN refines these candidate regions through multiple stages, *i.e.*, a cascade of detection branches, so that each stage outputs progressively more accurate bounding boxes and segmentation masks.

For implementation, we used Detectron2 [57] to develop and train the model. Since Detectron2 does not offer official support for Windows or macOS, we created a TorchScript [58] version of the model for these platforms using TorchScript's tracing method. This approach captures the operations of the model during inference with a sample input. It then generates a portable version of the model that operates independently of the full Detectron2 environment, ensuring compatibility across different systems. The script automatically detects the platform in use and selects the appropriate model and processing logic accordingly.

The model was pre-trained on ImageNet-1k [65] and re-trained on a private dataset from four institutions (the University Hospital of Cologne, the University Hospital of Szeged, the University Hospital of Lille and the University Hospital of Bari) encompassing the 12 types of glomerulonephritis used in the classification stage. The dataset included 587 WSIs from 227 patients for training, 161 WSIs from 57 patients for validation, and 127 WSIs from 47 patients for testing, with 16,571 annotated glomeruli for training, 3,958 for validation, and 2,948 for testing. The model was trained with the following parameters: Optimizer SGDM (Stochastic Gradient Descent with Momentum), Base learning rate 3×10^{-4} , Maximum number of iterations 300,000, and Batch size 2. It achieves an AP_{50} of 0.720 and a Dice coefficient of 0.807 on the test set. Further details of the implemented model can be found in a related paper by one of the authors and collaborators [28].

Although SOTA results in glomeruli detection are often reported with higher performance metrics, these are frequently based on limited, single-institution datasets with restricted diversity, as introduced in Section 1. For example, [34] uses the same method, *i.e.*, Cascade Mask R-CNN, and achieves an AP of 0.962 for PAS-stained glomeruli detection on a private dataset containing 3,017 glomeruli acquired from a single institution, predominantly from IgAGN cases. In contrast, the dataset used in this study contains a more extensive and diverse collection of 23,477 annotated PAS-stained glomeruli, representing 12 types of glomerulonephritis in a roughly balanced distribution and acquired from four institutions. This broader range supports the model's generalizability across varied glomerular pathologies, as opposed to models optimized on narrow datasets that may not perform as reliably on unseen or heterogeneous samples.

To test the model's generalizability on different data types, we used the HuBMAP Kaggle dataset [66], which consists of 15 annotated PAS-stained WSIs from surgical excisions, for external validation. Surgical

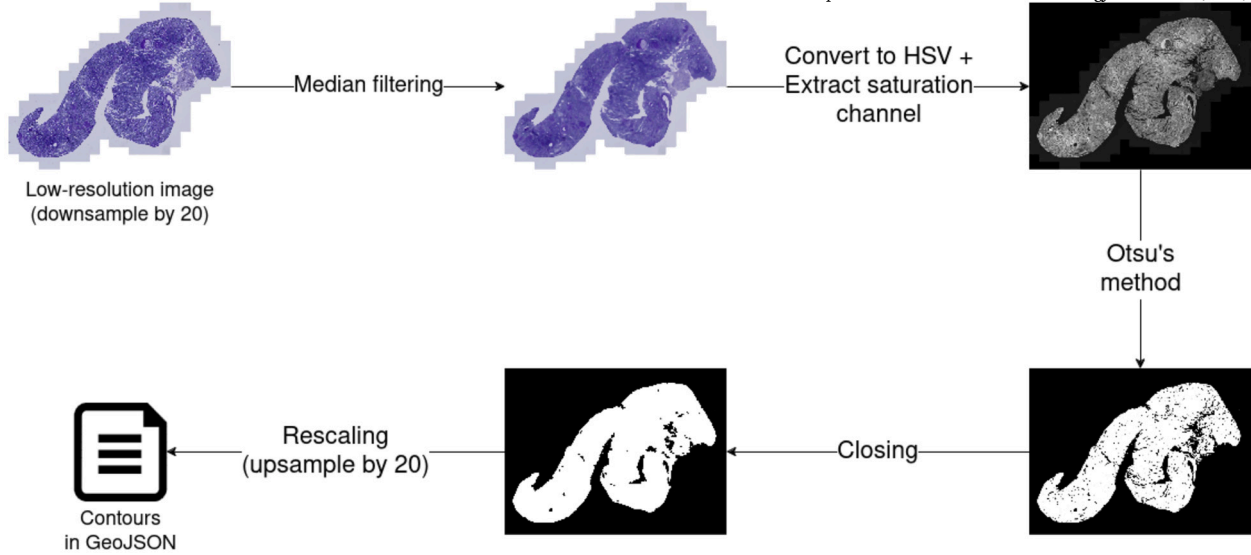


Fig. 4. A schematic representation of the tissue detection algorithm.

excisions differ significantly from needle biopsies, as they contain larger, more diverse tissue regions, with a broader array of glomerular structures and surrounding anatomical features. This dataset was selected to evaluate how such variations impact model performance when applied to a different type of renal tissue sample.

On this dataset, the model achieved an AP_{50} of 0.8630 and a Dice coefficient of 0.7334. The difference in these metrics may arise from the fact that both were calculated at the WSI level, where the large image sizes amplify subtle differences in segmentation boundaries. These findings underscore the model's robustness in detecting glomeruli across diverse sample types, although segmentation precision may vary with increased structural diversity.

Before detailing the detection process, it is important to address the NMAS algorithm, which we implemented to handle overlapping detections more effectively than NMS. In the conventional NMS algorithm, the bounding box with the highest confidence score is selected, and all other bounding boxes that overlap it beyond a predefined Intersection over Union (IoU) threshold are removed. This procedure is applied recursively until all boxes have been processed. However, the area of the detected objects is not taken into account. Consequently, if two bounding boxes are detected for the same object, one inside the other, the algorithm may select the smaller box, even if it only has a slightly higher confidence score.

To address this problem, NMAS considers not only the confidence score but also the area of the bounding box. This is achieved by introducing a parameter $f_j = w_j h_j s_j^2$, where w_j and h_j represent the bounding box width and height, respectively, and s_j is the confidence score. The square of the confidence score is used to penalize lower values. In addition, NMAS incorporates both Intersection over Union (IoU) and Intersection over Minimum (IoM) to detect overlapping boxes. IoM is particularly effective in identifying cases where one bounding box is largely contained within another, a scenario frequently encountered in overlapping sliding window methods. This enhancement improves performance in sliding window approaches [38]. The NMAS algorithm is presented in Algorithm 1.

The detection script performs the following operations:

1. Loads the model, its weights, and the patches to serve as the model's input.
2. Runs inference on the patches, producing bounding boxes and segmentation masks as output.
3. Projects back the patch-level detections to WSI-level detections.

Algorithm 1: Non-Maximum-Area Suppression.

```

input :  $B_i = b_1, \dots, b_{N_i}$ , the  $N_i$  initial detections
          $b_j = (X_j, y_j, w_j, h_j), j = 1, \dots, N_i$ 
          $S_i = s_1, \dots, s_{N_i}$ , the  $N_i$  initial scores
          $T_{iou}$ , the NMAS threshold on  $IoU$ 
          $T_{iom}$ , the NMAS threshold on  $IoM$ 
output:  $B_o = b_1, \dots, b_{N_o}$ , the  $N_o \leq N_i$  final detections
          $S_o = S_{o_1}, \dots, S_{o_{N_o}}$ , the  $N_o \leq N_i$  final scores

1  $B_o = \{ \};$ 
2  $S_o = (w_1 h_1 s_1^2, w_2 h_2 s_2^2, \dots, w_{N_i} h_{N_i} s_{N_i}^2);$ 
3 while  $B_i$  is not empty do
4    $m = \text{argmax}(S_o);$ 
5    $B_o = B_o \cup \{b_m\};$ 
6    $B_i = B_i \setminus \{b_m\};$ 
7   while  $b_j \in B_i$  do
8     if  $iou(B_m, b_j) \geq T_{iou} \vee iom(b_m, b_j) \geq T_{iom}$  then
9        $B_i = B_i \setminus \{b_j\};$ 
10       $S_o = S_o \setminus \{s_j\};$ 
11    end
12  end
13 end

```

4. Reduces overlapping annotations to a single one by applying NMAS, with both thresholds T_{iou} and T_{iom} set to 0.4. These values were determined empirically.
5. Discards all masks with an area smaller than $5000 \mu\text{m}^2$.

In both step 3 and step 4, the area of each predicted mask is taken into account. However, the pixel size in two different WSIs may result in different physical sizes on the slide. Therefore, the area must be calculated in square microns relative to the original specimen, rather than in square pixels in the image. There are two ways to calculate the area in microns: using the WSI magnification or the pixel size. The former depends on the objective lens of the device used to obtain the WSI, so two WSIs at the same magnification may have different pixel-to-micron ratios if they were captured with different devices. Conversely, pixel size is a vendor-neutral descriptor of image quality and is more suitable for this task [67]. Therefore, the area is computed using pixel size instead of magnification level. The pixel size in microns per pixel, for both width and height, can be extracted using functions from the QuPath library

To add the annotations to the WSI in QuPath in real time, the outlines of the output masks are now exported as a GeoJSON file, with the class “Glomerulus” assigned to each. This format can be imported directly within QuPath, allowing the tool to update annotations without leaving the main QuPath application. Once detection is complete, the GeoJSON annotations are read as QuPath Objects and imported into the image hierarchy. Both the GeoJSON files and the patches generated in the previous step are then removed from disk.

2.6. Extracting glomerular crops

For each image, all “Glomerulus”-class annotations must be exported as individual images, which will serve as input for the classification model. Each “Glomerulus”-class annotation is exported with a padding of 300 pixels on each side, as the glomerular crops used to train the classification models included this padding.

Each annotation is exported as a PNG file with a unique identifier, which is included in the filename, along with the X and Y coordinates, and the width and height. This information will be used at the end of the classification stage to trace the results back to each annotation in QuPath.

2.7. Glomeruli classification

The glomeruli classification stage is performed by a Python script based on QIGS. Its main features are as follows:

- A model and its weights are loaded to classify glomerular crops as either sclerotic or non-sclerotic. If the 12-pathologies classification is selected by passing an optional argument to the script, a different model and its weights are loaded to classify the non-sclerotic glomeruli into 12 common diagnoses of glomerulonephritis.
- The top-3 predictions are computed for the classification into the 12 glomerulonephritis diagnoses. The probabilities given by the models for each class are saved and used to determine the 3 most likely pathologies, not only for each individual glomerulus but also for the whole WSI. The process of computing the slide-level top-3 predictions is explained later in this section.
- The output of the script consists of one CSV file per WSI, where each row corresponds to a glomerulus, and both the predicted pathologies and their respective probabilities are provided. Additionally, another CSV file is provided as a summary for all classified WSIs. In this file, each row corresponds to a WSI, and both the predicted pathologies at the slide level and the ratio of glomeruli with that pathology to the total number of glomeruli are provided.
- The model checkpoints are loaded locally. However, they are downloaded and saved to the correct location when the user installs the tool.

Two glomerular crop classification models were developed, one for each classification step. Both models are implemented in MMClassification [59] and pre-trained on ImageNet-1k [65]. They were both re-trained using a private dataset, provided by the four aforementioned institutions (the University Hospital of Cologne, the University Hospital of Szeged, the University Hospital of Lille, and the University Hospital of Bari). This dataset comprises 725 WSIs from 400 patients across 13 classes: 12 types of glomerulonephritis and glomerulosclerosis.

The sclerotic/non-sclerotic classification model uses a Swin Transformer architecture [68], in its Swin-T (Tiny) version, trained and evaluated with a 5-fold cross-validation technique. It was re-trained on 754 glomeruli, validated on 312 glomeruli, and tested on 2,959 glomeruli, Optimizer AdamW, Initial learning rate 0.001, Weight decay 0.05, $\epsilon 10^{-8}$, $\beta_1 0.9$, $\beta_2 0.999$, Learning rate policy: Cyclic with cosine annealing, Warm-up epochs 20, Warm-up type: Linear, Warm-up rate 0.001, Minimum learning rate 0.01, Maximum number of epochs 300, and Batch size 64. The best fold was selected, achieving balanced accuracy

of 0.9846, precision of 0.9784, and an AUC of 0.95 on the test set. Further details of the implemented model can be found in the paper by the authors [69].

The disproportionate sizes of the partitions are due to the fact that this set is a subset of the original dataset, which includes 13 classes, that is, glomerulosclerosis and 12 types of glomerulonephritis. Therefore, all sclerotic glomeruli were used, and a subset of glomeruli from the 12 pathologies was selected to assemble a balanced non-sclerotic class. The remaining glomeruli were included in the test set to ensure robust model evaluation, and an independent dataset of sclerotic glomeruli [70] from the European project AIDPATH¹ was incorporated for balancing, resulting in a larger test set. This AIDPATH dataset consists of 50 WSIs, each from a different patient, with sclerotic glomeruli. The dataset was provided by three institutions: Castilla-La Mancha’s Healthcare Services (Spain), the Andalusian Health Service (Spain), and Vilnius University Hospital Santaros Klinikos (Lithuania).

The 12-label classification model uses a ConvNeXt architecture [71], in its ConvNeXt-B (Base) version, and was also trained and evaluated using a 5-fold cross-validation technique. It was re-trained on 6,511 glomeruli, validated on 1,506 glomeruli, and tested on 1,480 glomeruli, with the following parameters: Optimizer AdamW, Initial learning rate 0.001, Weight decay 0.05, $\epsilon 10^{-8}$, $\beta_1 0.9$, $\beta_2 0.999$, Learning rate policy: Cyclic with cosine annealing, Warm-up epochs 20, Warm-up type: Linear, Warm-up rate 0.001, Minimum learning rate 0.01, Maximum number of epochs 300, and Batch size 64.

The best fold was selected, achieving top-1 values of 0.6676 for balanced accuracy, 0.3848 for precision, and 0.6925 for AUC on the test set. The top-3 predictions improved these metrics to 0.8441 for balanced accuracy and 0.7134 for precision. Further details of the implemented model can be found in the paper by the authors [69].

The results for the classification into 12 pathologies are given as top-3 predictions. The probability scores for each label are generated by both models and added as measurements to the corresponding annotations in QuPath. When classifying a non-sclerotic glomerulus into the 12 pathologies, the class of the corresponding annotation is set to the 3 most probable labels, separated by vertical bars and listed in decreasing order.

For example, “ABMGN | ANCA | DDD” would mean that the most probable predicted labels for the glomerulus are: 1st) ABMGN, 2nd) ANCA, and 3rd) DDD. Top-3 predictions were used instead of top-1 predictions to provide pathologists with an optimal guide for diagnosis, as these achieve higher performance.

Once the classification is complete, all remaining temporary directories and files are removed from the disk.

2.8. Operational requirements

GNCnn supports both GPU and CPU for inference. For GPU use, only NVIDIA units with CUDA 11.1 are supported. Python 3.8 or 3.9 is required for the Python scripts. The tool was developed using OpenJDK 17 but should be compatible with Java 8+ versions. GNCnn was developed and tested with QuPath version 0.5.0. The extension has been tested on the following operating systems: Ubuntu 20.04 and 22.04, Windows 10, and macOS Big Sur 11.4. Additionally, automatic installation scripts are provided for each of the three platforms.

The minimum hardware requirements for GNCnn align with those recommended for QuPath, as it functions as a QuPath plugin. The software is compatible with 64-bit Windows, Linux, and macOS systems. Optimal performance is achieved with a fast multicore processor (e.g., Intel Core i7 or higher) and at least 16 GB of RAM. These specifications help prevent memory limitations and performance slowdowns during resource-intensive tasks, such as analyzing large numbers of objects,

¹ <https://aidpath.eu/>.

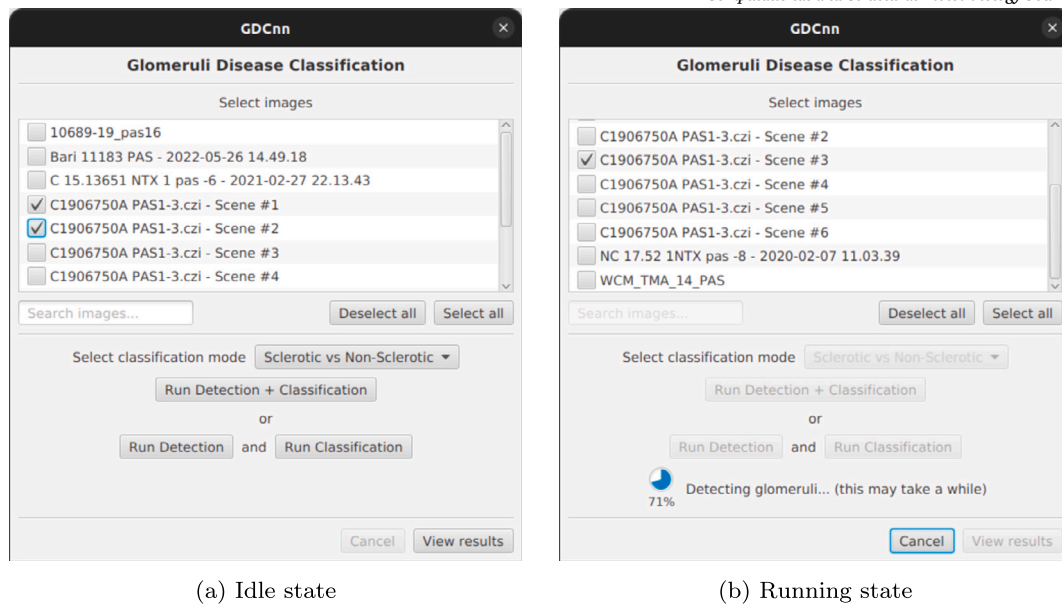


Fig. 5. Main window for the plugin, both in a) idle and b) running states.

processing multiplexed data, or handling non-pyramidal images and z-stacks of WSIs.

While GNCnn can operate on a CPU, inference times for glomeruli detection and classification are significantly reduced with an NVIDIA GPU. For optimal processing efficiency, we recommend an NVIDIA GPU with at least 4 GB of memory.

3. Results and discussion

The main window of the plugin is shown in Fig. 5. This window contains the following elements:

- Image selector: The images in the project are displayed in a list of checkboxes. Two buttons allow the user to select or deselect all images in the project. Additionally, a search bar is provided to enable the user to filter the images shown in the list.
- Detection and classification buttons: First, a choice box offers two options: running sclerotic/non-sclerotic classification or performing sclerotic classification along with classification of non-sclerotic glomeruli into 12 pathologies. Furthermore, the user can choose to run the entire detection and classification pipeline or only one of these processes.
- Progress indicator: A progress percentage is displayed while the plugin is running, as shown in Fig. 5b.
- Canceling and results buttons: If the user presses the “Cancel” button, the process is terminated, and all temporary files, directories, or annotations are removed. The “View results” button opens a WSI-level results window for the selected images, as shown in Fig. 6.

The main window is carefully designed to meet the specific needs of nephrologists. QuPath projects often involve handling a large number of WSIs, making efficient image management a priority. To address this, the image selector is equipped to handle high volumes of images, allowing users to quickly filter, select, or deselect images as needed.

In addition to image management, the detection and classification buttons provide flexibility for a variety of tasks. These buttons are designed to accommodate multiple use cases, ensuring that users can either focus solely on detection or classification, or run both processes sequentially depending on their workflow. This versatility allows for customization based on specific project requirements, such as focusing on

glomerular detection, refining annotations, or performing classification based on pre-existing annotations. Thus, the detection and classification buttons handle the following use cases:

- The user may have images that are already annotated with glomeruli and may only want to run the classification.
- The user may want to run only the detection in order to count glomeruli and manually analyze each one for potential diseases.
- The detection model may generate false positives in the image, so the user may choose to run only the detection, manually revise the generated annotations, and then proceed with the classification.

The previously mentioned WSI-level results window consists of three main elements:

- Main table: The results are shown in a table, with each row corresponding to a WSI. A low-resolution version of the image and its name identify the WSI. The table includes counts for total, per-label, and non-classified glomeruli, along with the most frequently predicted class. This most predicted class could be empty if there are no glomeruli annotated in the image; a single label, if none of the glomeruli are classified into the 12 pathologies for non-sclerotic glomeruli; or the three most probable classes, if there are glomeruli classified into any of the 12 pathologies. These three most probable classes are computed based on the probabilities added as measurements to each glomerulus. Sclerotic and non-sclerotic probabilities add up to 1, and the probabilities for the 12 classes also sum to 1. Since the 12 pathologies correspond to non-sclerotic glomeruli, these probabilities are multiplied by the non-sclerotic probability to calculate the actual score. For annotations classified only as sclerotic or non-sclerotic, the probabilities for the 12 pathologies are set to 0. The probabilities for each label are summed across all annotations in a WSI and ordered in decreasing order. The top three labels are then selected, determining the three most probable classes for the WSI, listed in decreasing order.
- Search bar: The user can filter images by entering a text string, which is applied to any column in the results table. This functionality allows for quick and efficient searching, enabling the user to locate specific images based on criteria such as name, classification results, or other attributes associated with the results.

Thumbnail	Image	Most predicted class	Glomeruli	Non-sclerotic	Sclerotic	ABMGN	ANCA	C3-GN
	C1906750A PAS1-3.czi - Scene #1		0	0	0	0	0	0
	C1906750A PAS1-3.czi - Scene #3	Non-classified	4	0	0	0	0	0
	NC 17.52 1NTX pas -8 - 2020-02-07 11.03.39	Non-sclerotic	22	19	3	0	0	0
	WCM_TMA_14_PAS	CryoglobulinemicGN Sclerotic SLEGN-IV	37	0	7	0	0	0

Fig. 6. WSI-level results window.

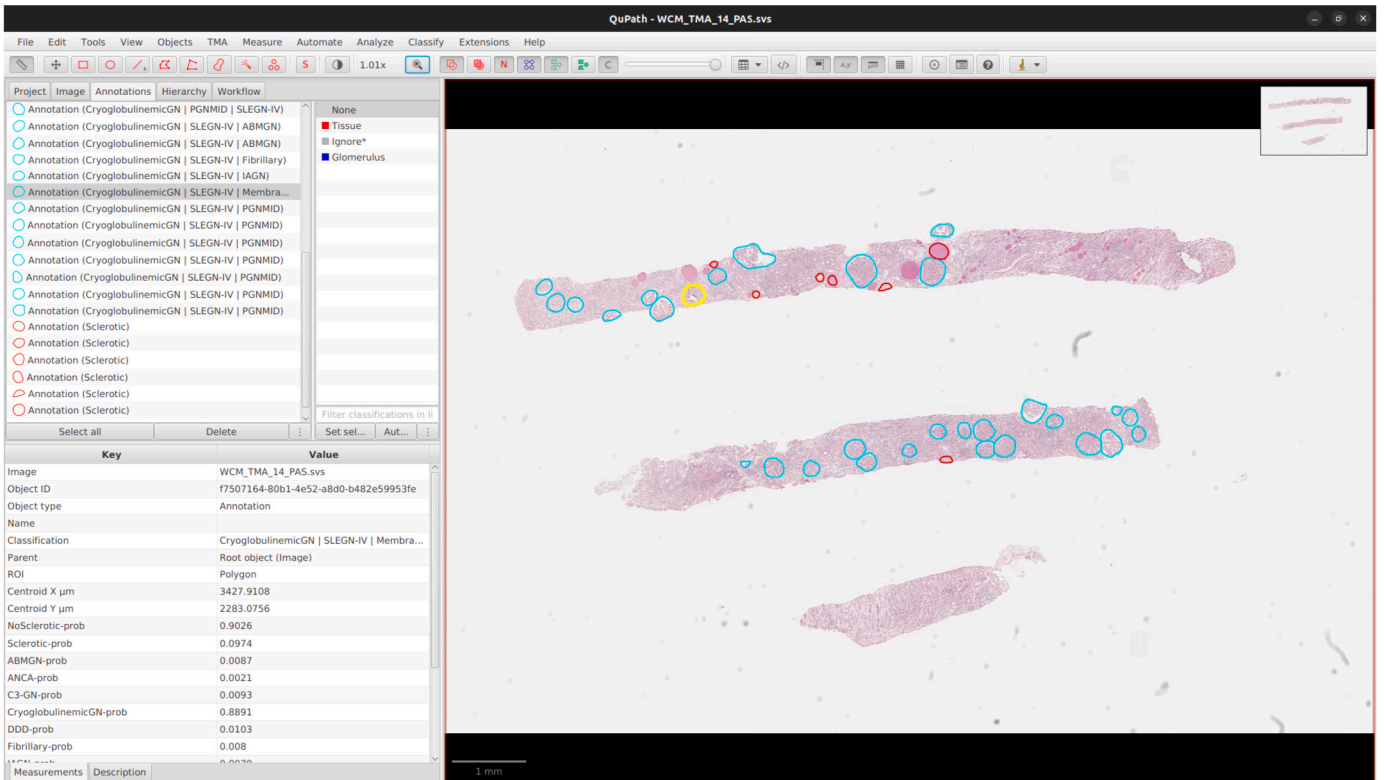
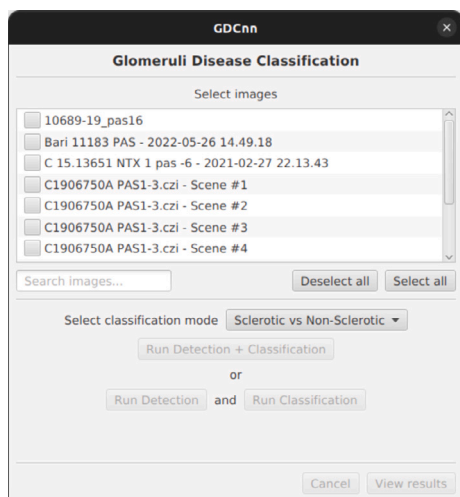
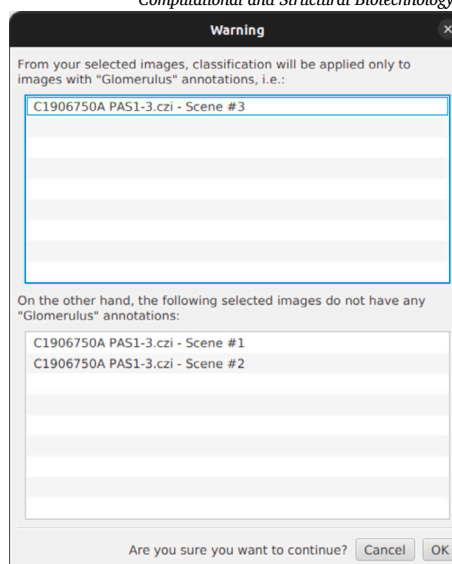


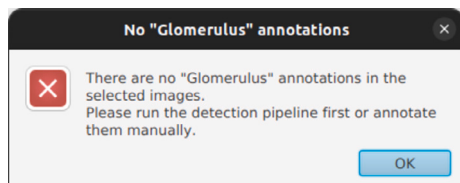
Fig. 7. Result of running the plugin on a WSI.



(a) Disabled elements



(b) Classification warning on some WSIs



(c) Classification warning on all WSIs



(d) Cancellation warning when closing

Fig. 8. Error handling and warnings in the QuPath plugin.

- “Save” button: The user can save the results to a CSV file for further analysis or documentation. This allows the user to export all relevant data, making it easy to review or share the findings outside of the QuPath environment.

Glomerulus-level results are updated on the image in real time. An example is shown in Fig. 7.

Furthermore, potential errors are handled, and warnings are issued when necessary. For example, if no image is selected or no project is open, interface elements are disabled (Fig. 8a). If the user tries to run the classification on images without annotated glomeruli, the extension either prevents the classification from running (Fig. 8c) or warns the user that the classification will only run on images with annotated glomeruli (Fig. 8b). Moreover, if the user attempts to close the main window while a process is running, a confirmation window is displayed (Fig. 8d).

Finally, an analysis of the execution time of the plugin for the complete process was conducted, using the classification into sclerotic and 12 pathologies for non-sclerotic glomeruli. For this purpose, 10 WSIs of varying sizes and amounts of tissue were used, and the execution times for each operation were compared. This analysis was performed for both GPU and CPU inference to evaluate performance across different hardware configurations. Model inference was performed on an NVIDIA RTX A6000 (with 48 GB of memory) GPU and an Intel Core i7-9800X (with 8 cores) CPU. The results are shown in Table 2. The analysis reveals that glomeruli detection is the most time-consuming task, accounting for an average of 72.64% of the total time with GPU inference and 89.31% with CPU-only inference, followed by tile exporting, which takes 10.81% of the time on the GPU and 4.46% on the CPU.

The total execution time primarily depends on the glomeruli detection stage, which is influenced by the number of patches generated. The number of patches depends not only on the amount of tissue in the image, but also on how the tissue is distributed within the image.

For a WSI, glomeruli are efficiently delineated and classified within an average time of 2 minutes and 53 seconds on a GPU, and approximately 8 minutes and 29 seconds on a CPU. They are categorized as either sclerotic or into the three most probable pathologies out of a possible 12. While GPU acceleration provides significant speed advantages, CPU-based processing times remain practical for specific workflows. For example, the CPU mode can be utilized to process batches of WSIs while the user focuses on other clinical tasks, enabling pathologists to later review pre-evaluated slides with preliminary diagnostic results.

Moreover, a comprehensive WSI-level diagnosis is generated, providing not only an overview of the most likely pathologies but also a detailed count of glomeruli classified by pathology type. The system also offers the total number of glomeruli and a breakdown per pathology, making it a powerful tool to significantly accelerate the diagnostic process. By automating these time-consuming tasks, this tool can enhance the efficiency and accuracy of nephropathologists, enabling them to focus on more complex cases and improving patient outcomes.

4. Conclusions

In this study, we introduce GNCnn, a user-friendly extension for QuPath that automatically detects and classifies glomeruli using only PAS-stained images from renal biopsies. GNCnn first classifies glomeruli as sclerotic or non-sclerotic, and then further classifies non-sclerotic cases into 12 common diagnoses of glomerulonephritis. The extension is

Table 2

Plugin execution times comparison. Each row represents a different WSI, except for the last row, which presents the average values of the previous rows.

Dimensions (pixels)	Tissue amount (pixels)	Generated patches	Tile exporting (s)		Tissue detection (s)		Glomeruli detection (s)		Exporting glomerular crops (s)		Classification (s)		Total time (s)	
			GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU
51791 x 33575	9.88×10^7	171	20.55	20.23	0.68	0.68	83.65	345.26	18.26	10.35	14.33	15.76	137.46	392.28
37780 x 26858	1.42×10^7	55	4.65	8.80	2.54	0.91	30.26	234.00	2.91	14.90	11.83	13.67	52.18	272.28
42240 x 32256	7.73×10^7	95	6.30	5.54	0.71	0.72	54.67	207.76	8.53	24.45	12.63	13.89	82.83	252.36
85043 x 51681	1.75×10^8	419	45.05	32.07	2.15	2.14	193.37	608.06	28.27	17.95	14.98	15.23	283.82	675.45
49920 x 32256	1.10×10^5	140	6.87	7.40	0.84	0.83	74.84	299.22	6.79	5.76	12.47	13.72	101.79	326.93
72960 x 34560	5.80×10^7	173	9.94	10.12	1.13	1.16	87.52	367.81	9.10	6.76	12.35	13.58	120.03	399.43
50970 x 135142	1.78×10^8	793	34.75	85.96	4.66	4.74	376.93	1472.79	19.18	32.47	14.27	15.70	449.78	1611.66
31855 x 23861	2.91×10^8	109	15.21	15.80	0.65	0.68	57.57	204.54	20.41	18.48	12.05	13.26	105.89	252.76
119040 x 64512	8.80×10^7	414	25.93	28.69	2.87	1.92	206.44	476.48	14.85	16.73	15.33	15.30	265.42	539.12
65185 x 38178	4.70×10^7	196	17.62	12.22	1.33	1.02	90.68	330.61	7.83	10.43	12.34	14.11	129.80	368.39
Mean		257	18.69	22.68	1.76	1.48	125.59	454.65	13.61	15.83	13.26	14.42	172.90	509.07

designed with a strong emphasis on usability and accessibility, featuring an intuitive interface specifically tailored for nephrologists.

To our knowledge, GNCnn is the first tool to integrate a comprehensive pipeline for the characterization of glomeruli within QuPath. By leveraging deep learning algorithms, GNCnn offers a promising solution to support and streamline the routine, time-consuming tasks of nephrologists.

GNCnn allows nephrologists to execute the entire pipeline or individual parts of it, depending on the specific study they need to perform. Moreover, results are updated on the image in QuPath in real time, eliminating the need for external tools. All results are provided directly within the main application.

These results are presented at the glomerulus level, as annotations on the images, and at the slide level, offering a visualization tab that includes a quantification of glomeruli in the image and a diagnosis for the entire slide. The slide-level diagnosis is computed as the most predicted class for sclerotic and non-sclerotic glomeruli, or as the three most probable classes, based on the probabilities given for each pathology at the glomerulus level.

Glomeruli detection achieves an AP_{50} of 0.720 and a Dice coefficient of 0.807 on the test set. An external validation was performed on images from surgical excisions, a sample distinct from the needle biopsies used during training, to evaluate the model's generalizability. The model achieved an AP_{50} of 0.8630 and a Dice coefficient of 0.7334. The sclerotic and non-sclerotic classification reaches a balanced accuracy of 0.9846, while classification into 12 common diagnoses of glomerulonephritis achieves a balanced accuracy of 0.8441, considering up to the top-3 predictions.

Furthermore, a computational time study was conducted, showing that the extension takes an average of 2 minutes and 53 seconds to complete the entire pipeline using an NVIDIA GPU, compared to 8 minutes and 29 seconds for CPU-only inference. These results underscore the software's potential to support nephrologists in their diagnostic workflow.

The GNCnn extension is not without limitations. The morphological similarities among the 12 types of glomerulonephritis hinder the performance of the model designed for this classification. This limitation has been addressed by incorporating top-3 predictions, presenting these results as a guide to the most probable diagnoses.

Additionally, while external validation was performed on 15 WSIs to evaluate the detection model, the limited availability of large, multi-site, and publicly accessible datasets in nephrology constrains comprehensive validation for both detection and classification tasks. This limitation highlights the need for future studies using broader datasets to further validate and enhance the model's generalizability and performance.

Finally, while the tool operates efficiently on systems equipped with high-performance GPUs, processing times on CPU-only systems are significantly longer. This performance gap may limit its usability in environments restricted to CPU-based hardware. To address this limitation, future improvements could include model quantization or server-based inference, making GNCnn more accessible in clinical settings with limited hardware resources.

CRediT authorship contribution statement

Israel Mateos-Aparicio-Ruiz: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Anibal Pedraza:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Conceptualization. **Jan Ulrich Becker:** Writing – review & editing, Validation, Resources, Formal analysis, Data curation. **Nicola Altini:** Validation, Software, Data curation. **Jesús Salido:** Writing – review & editing, Validation, Supervision. **Gloria Bueno:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Funding

The authors acknowledge funding support from the HANS project, funded by the Spanish Ministry of Science, Innovation and Universities, as well as by the European Union NextGenerationEU/PRTR (Ref. PID2021-127567NB-I00). Additional support was provided by the ARTE project (Ref. 2022-GRIN-34352), funded by the University of Castilla-La Mancha.

Declaration of competing interest

The authors declare no conflicts of interest.

Acknowledgements

We would also like to acknowledge the collaboration of Jean-Baptiste Gibier from the Pathology Institute, Lille University, France; Sándor Turkevi-Nagy from the Department of Pathology, Albert Szent-Györgyi Health Center, University of Szeged, Hungary; Avi Z. Rosenberg from The Johns Hopkins Hospital, Baltimore, USA; and Surya V. Seshan from Weill Cornell Medicine, New York, USA, for their assistance with validation and data provision.

Appendix A. Additional diagrams

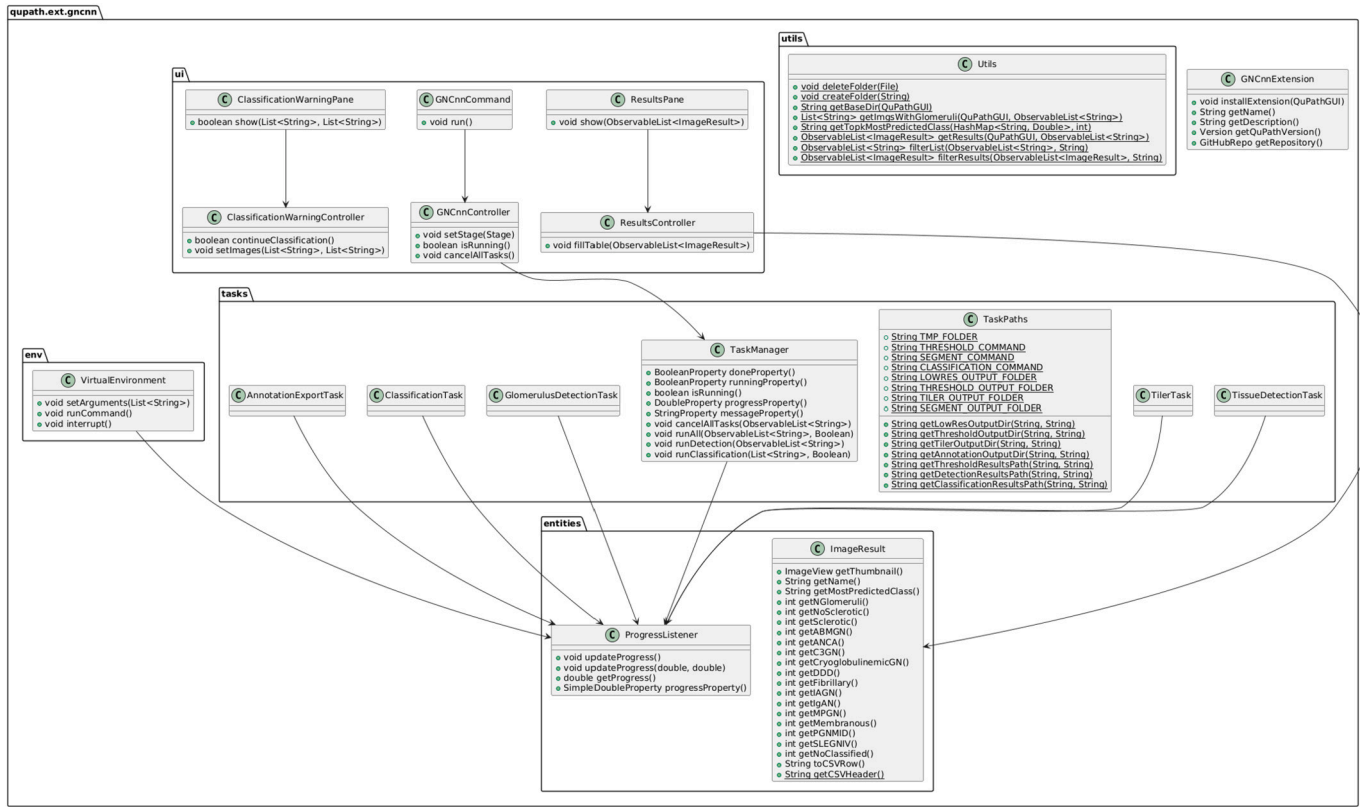


Fig. 9. Class diagram of GNCnn. For clarity, only public attributes and methods are shown.

Data availability

GNCnn is available open source at <https://github.com/UCLM-VISILAB/qupath-extension-gncnn>.

References

[1] Racusen LC, Solez K, Colvin RB, Bonsib SM, Castro MC, et al. The Banff 97 working classification of renal allograft pathology. *Kidney Int* 1999;55(2):713–23.

[2] Roberts IS, Cook HT, Troyanov S, Alpers CE, Amore A, et al. The Oxford classification of IgA nephropathy: pathology definitions, correlations, and reproducibility. *Kidney Int* 2009;76(5):546–56. <https://doi.org/10.1038/ki.2009.168>.

[3] Cattran DC, Coppo R, Cook HT, Feehally J, Roberts IS, et al. The Oxford classification of IgA nephropathy: rationale, clinicopathological correlations, and classification. *Kidney Int* 2009;76(5):534–45. <https://doi.org/10.1038/ki.2009.243>.

[4] Trimarchi H, Barratt J, Cattran DC, Cook HT, Coppo R, et al. Oxford Classification of IgA nephropathy 2016: an update from the IgA Nephropathy Classification Working Group. *Kidney Int* 2017;91(5):1014–21. <https://doi.org/10.1016/j.kint.2017.02.003>.

[5] Kumar N, Gupta R, Gupta S. Whole slide imaging (WSI) in pathology: current perspectives and future directions. *J Digit Imag* 2020;33(4):1034–40.

[6] Poalelungi DG, Neagu AI, Fulga A, Neagu M, Tutunaru D, et al. Revolutionizing pathology with artificial intelligence: Innovations in immunohistochemistry. *J Person Med* 2024;14(7):693.

[7] Jain E, Patel A, Parwani AV, Shafi S, Brar Z, et al. Whole slide imaging technology and its applications: Current and emerging perspectives. *Int J Surg Pathol* 2024;32(3):433–48.

[8] Romero Lauro G, Cable W, Lesniak A, Tseytlin E, McHugh J, et al. Digital pathology consultations—a new era in digital imaging, challenges and practical applications. *J Digit Imag* 2013;26:668–77.

[9] Lajara N, Espinosa-Aranda JL, Deniz O, Bueno G. Optimum web viewer application for dicom whole slide image visualization in anatomical pathology. *Comput Methods Programs Biomed* 2019;179:104983.

[10] Gorman C, Punzo D, Octaviano I, Pieper S, Longabaugh WJ, et al. Interoperable slide microscopy viewer and annotation tool for imaging data science and computational pathology. *Nat Commun* 2023;14(1):1572.

[11] Collins TJ. ImageJ for microscopy. *BioTechniques* 2007;43(S1):S25–30.

[12] Bankhead P, Loughrey MB, Fernández JA, Dombrowski Y, McArt DG, et al. Qupath: Open source software for digital pathology image analysis. *Sci Rep* 2017;7(1):1–7.

[13] Humphries M, Maxwell P, Salto-Tellez M. Qupath: The global impact of an open source digital pathology system. *Comput Struct Biotechnol J* 2021;19:852–9.

[14] Burri O, Sobolewski P, Fehlmann T. BIOP/qupath-extension-cellpose: Parallel object reader, normalization bugfix gpu disable option and more (v0.9.3). Available from: <https://doi.org/10.5281/zenodo.11203598>, 2024.

[15] Stringer C, Wang T, Michaelos M, Pachitariu M. Cellpose: a generalist algorithm for cellular segmentation. *Nat Methods* 2021;18(1):100–6.

[16] Pachitariu M, Stringer C. Cellpose 2.0: how to train your own model. *Nat Methods* 2022;19(12):1634–41.

[17] Cutler KJ, Stringer C, Lo TW, Rappetz L, Stroustrup N, et al. Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation. *Nat Methods* 2022;19(11):1438–48.

[18] L’Imperio V, Cazzaniga G, Mannino M, Seminati D, Mascadri F, et al. Digital counting of tissue cells for molecular analysis: the quantum pipeline. *Virchows Arch* 2024;1–10.

[19] Sugawara K. Training deep learning models for cell image segmentation with sparse annotations. *BioRxiv* 2023. <https://doi.org/10.1101/2023.06.13.544786>.

[20] Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, et al. Segment anything. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*; 2023. p. 4015–26.

[21] Zhang C, Han D, Qiao Y, Kim JU, Bae SH, et al. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint*. Available from: [arXiv:2306.14289](https://arxiv.org/abs/2306.14289), 2023.

[22] Schmidt U, Weigert M, Broaddus C, Myers G. Cell detection with star-convex polygons. In: *Medical image computing and computer assisted intervention—MICCAI 2018: 21st international conference, Granada, Spain, September 16–20, 2018, proceedings, part II 11*. Springer; 2018. p. 265–73.

[23] Kaczmarzyk JR, O’Callaghan A, Inglis F, Gat S, Kurc T, et al. Open and reusable deep learning for pathology with WSInfer and QuPath. *NPJ Precis Oncol* 2024;8(1):9.

[24] Sarkis R, Burri O, Royer-Chardon C, Schyrr F, Blum S, et al. MarrowQuant 2.0: a digital pathology workflow assisting bone marrow evaluation in experimental and clinical hematology. *Mod Pathol* 2023;36(4):100088.

[25] Huang CH. QuST: QuPath Extension for Integrative Whole Slide Image and Spatial Transcriptomics Analysis. *arXiv preprint*. Available from: [arXiv:2406.01613](https://arxiv.org/abs/2406.01613), 2024.

[26] Huang CH. QuST-LLM: Integrating Large Language Models for Comprehensive Spatial Transcriptomics Analysis. *arXiv preprint*. Available from: [arXiv:2406.14307](https://arxiv.org/abs/2406.14307), 2024.

- [27] Cazzaniga G, Rossi M, Eccher A, Girolami I, L'Imperio V, et al. Time for a full digital approach in nephropathology: a systematic review of current artificial intelligence applications and future directions. *J Nephrol* 2024;37(1):65–76.
- [28] Altini N, Rossini M, Turkevi-Nagy S, Pesce F, Pontrelli P, et al. Performance and limitations of a supervised deep learning approach for the histopathological Oxford Classification of glomeruli with IgA nephropathy. *Comput Methods Programs Biomed* 2023;242:107814.
- [29] Gutman DA, Khalilia M, Lee S, Nalisnik M, Mullen Z, et al. The digital slide archive: a software platform for management, integration, and analysis of histology for cancer research. *Cancer Res* 2017;77(21):e75–8.
- [30] Stritt M, Stalder AK, Vezzali E. Orbit image analysis: an open-source whole slide image analysis tool. *PLoS Comput Biol* 2020;16(2):e1007313.
- [31] Marée R, Rollus L, Stévens B, Hoyoux R, Louppe G, et al. Collaborative analysis of multi-gigapixel imaging data using cytomine. *Bioinformatics* 2016;32(9):1395–401.
- [32] Lutnick B, Manthey D, Becker JU, Ginley B, Moos K, et al. A user-friendly tool for cloud-based whole slide image segmentation with examples from renal histopathology. *Commun Med* 2022;2(1):105.
- [33] Govind D, Becker JU, Miecznikowski J, Rosenberg AZ, Dang J, et al. Podosighter: a cloud-based tool for label-free podocyte detection in kidney whole-slide images. *J Am Soc Nephrol* 2021;32(11):2795–813.
- [34] Jiang L, Chen W, Dong B, Mei K, Zhu C, et al. A deep learning-based approach for glomeruli instance segmentation from multistained renal biopsy pathological images. *Am J Pathol* 2021;191(8):1431–41.
- [35] Yang CK, Lee CY, Wang HS, Huang SC, Liang PI, et al. Glomerular disease classification and lesion identification by machine learning. *Biomed J* 2022;45(4):675–85.
- [36] Kawazoe Y, Shimamoto K, Yamaguchi R, Nakamura I, Yoneda K, et al. Computational pipeline for glomerular segmentation and association of the quantified regions with prognosis of kidney function in iga nephropathy. *Diagnostics* 2022;12(12):2955.
- [37] Bueno G, Fernandez-Carrobles MM, Gonzalez-Lopez L, Deniz O. Glomerulosclerosis identification in whole slide images using semantic segmentation. *Comput Methods Programs Biomed* 2020;184:105273. <https://doi.org/10.1016/j.cmpb.2019.105273>.
- [38] Altini N, Cascarano GD, Brunetti A, De Feudis I, Buongiorno D, et al. A deep learning instance segmentation approach for global glomerulosclerosis assessment in donor kidney biopsies. *Electronics* 2020;9(11):1768.
- [39] Salvi M, Mogetta A, Gambella A, Molinaro L, et al. Automated assessment of glomerulosclerosis and tubular atrophy using deep learning. *Comput Med Imaging Graph* 2021;90:101930. <https://doi.org/10.1016/j.compmedimag.2021.101930>.
- [40] Sethi S, Fervenza FC. Standardized classification and reporting of glomerulonephritis. *Nephrol Dial Transplant* 2019;34(2):193–9.
- [41] Chadban SJ, Atkins RC. Glomerulonephritis. *Lancet* 2005;365(9473):1797–806.
- [42] Romagnani P, Kitching AR, Leung N, Anders HJ. The five types of glomerulonephritis classified by pathogenesis, activity and chronicity (gn-ac). *Nephrol Dial Transplant* 2023;38(Supplement_2):ii3–10.
- [43] Feng C, Liu F. Artificial intelligence in renal pathology: current status and future. *Biomol Biomed* 2023;23(2):225.
- [44] Hermesen M, Smeets B, Hilbrands L, van der Laak J. Artificial intelligence: is there a potential role in nephropathology? *Nephrol Dial Transplant* 2022;37(3):438–40.
- [45] Althnian A, AlSaeed D, Al-Baity H, Samha A, Dris AB, et al. Impact of dataset size on classification performance: an empirical evaluation in the medical domain. *Appl Sci* 2021;11(2):796.
- [46] Ligabue G, Pollastri F, Fontana F, Leonelli M, Furci L, et al. Evaluation of the classification accuracy of the kidney biopsy direct immunofluorescence through convolutional neural networks. *Clin J Am Soc Nephrol* 2020;15(10):1445–54.
- [47] Zhang L, Li M, Wu Y, Hao F, Wang C, et al. Classification of renal biopsy direct immunofluorescence image using multiple attention convolutional neural network. *Comput Methods Programs Biomed* 2022;214:106532.
- [48] Kolachalama VB, Singh P, Lin CQ, Mun D, Belghasem ME, et al. Association of pathological fibrosis with renal survival using deep neural networks. *Kidney Int Rep* 2018;3(2):464–75.
- [49] Ledbetter D, Ho L, Lemley KV. Prediction of kidney function from biopsy images using convolutional neural networks. arXiv preprint. Available from: arXiv:1702.01816, 2017.
- [50] Lee J, Warner E, Shaikhouni S, Bitzer M, Kretzler M, et al. Unsupervised machine learning for identifying important visual features through bag-of-words using histopathology data from chronic kidney disease. *Sci Rep* 2022;12(1):4832.
- [51] Aldeman NLS, de Sá Urtiga Aita KM, Machado VP, da Mata Sousa LCD, Coelho AGB, et al. Smartpathk: a platform for teaching glomerulopathies using machine learning. *BMC Med Educ* 2021;21(1):248.
- [52] Humphries MP, McQuaid S, Craig SG, Bingham V, Maxwell P, et al. Critical appraisal of programmed death ligand 1 reflex diagnostic testing: current standards and future opportunities. *J Thorac Oncol* 2019;14(1):45–53.
- [53] Humphries MP, Bingham V, Abdullahi Sidi F, Craig SG, McQuaid S, et al. Improving the diagnostic accuracy of the pd-11 test with image analysis and multiplex hybridization. *Cancers* 2020;12(5):1114.
- [54] Campanella G, Hanna MG, Geneslaw L, Miraflor A, Werneck Krauss Silva V, et al. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nat Med* 2019;25(8):1301–9.
- [55] da Silva LM, Pereira EM, Salles PG, Godrich R, Ceballos R, et al. Independent real-world application of a clinical-grade automated prostate cancer detection system. *J Pathol* 2021;254(2):147–58.
- [56] Bradski G. The OpenCV Library. *Dr Dobb's J Softw Tools* 2000;120:122–5.
- [57] Wu Y, Kirillov A, Massa F, Lo WY, Girshick R. Detectron2. Available from: <https://github.com/facebookresearch/detectron2>, 2019.
- [58] Contributors P. TorchScript. Available from: <https://pytorch.org/docs/stable/jit.html>, 2023.
- [59] Contributors M. OpenMMLab's Image Classification Toolbox and Benchmark. Available from: <https://github.com/open-mmlab/mmlclassification>, 2020.
- [60] Otsu N, et al. A threshold selection method from gray-level histograms. *Automatica* 1975;11(285–296):23–7.
- [61] Bradski G, Kaehler A. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc.; 2008.
- [62] Butler H, Daly M, Doyle A, Gillies S, Schaub T, et al. The GeoJSON Format. RFC 7946. Available from: <https://doi.org/10.17487/RFC7946>, 2016.
- [63] Cai Z, Vasconcelos N. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Trans Pattern Anal Mach Intell* 2019;43(5):1483–98.
- [64] 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. p. 770–8.
- [65] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, et al. Imagenet large scale visual recognition challenge. *Int J Comput Vis* 2015;115:211–52.
- [66] Howard A, Lawrence A, Sims B, Tinsley E, Kazmierczak J, et al. Hubmap - hacking the kidney. Available from: <https://kaggle.com/competitions/hubmap-kidney-segmentation>, 2020.
- [67] Sellaro TL, Filkins R, Hoffman C, Fine JL, Ho J, et al. Relationship between magnification and resolution in digital pathology systems. *J Pathol Inform* 2013;4(1):21.
- [68] Liu Z, Lin Y, Cao Y, Hu H, Wei Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF international conference on computer vision*; 2021. p. 10012–22.
- [69] Bueno G, Mateos-Aparicio I, Pedraza A, Vo HQ, Altini N, et al. Classification of glomerulonephritis with CNN and self-attention networks in individual glomeruli in nephropathology. In: *IEEE-EMBS International Conference on Biomedical and Health Informatics*; 2024. Available from: <https://openreview.net/pdf?id=ak0LppzDaM>.
- [70] Bueno G, Gonzalez-Lopez L, Garcia-Rojo M, Laurinavicius A, Deniz O. Data for glomeruli characterization in histopathological images. *Data Brief* 2020;29:105314.
- [71] Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, et al. A convnet for the 2020s. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*; 2022. p. 11976–86.