Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-022-06608-9

RESEARCH ARTICLE-ELECTRICAL ENGINEERING

Glomerulus Semantic Segmentation Using Ensemble of Deep Learning Models

Ye Gu[1] · Ruyun Ruan[1] · Yan Yan[1] · Jian Zhao[1] · Weihua Sheng[2] · Lixin Liang[1] · Bingding Huang[1]

Received: 25 August 2021 / Accepted: 13 January 2022

Abstract
Quantification and classification of tissue features such as glomerulus are important elements of the histopathologic assessment of renal tissue. To fulfill this task, glomerulus segmentation is required. In this paper, we propose a multi-stream glomerulus segmentation framework based on three signature models: FCN, Deeplabv3 and Unet. Resnet is combined with FCN and Deeplabv3 model to enhance the encoding process. Meanwhile, Unet is upgraded to use EfficientNet as the backbone for feature extraction. Each individual model will output a local decision. On top of these base learners, ensemble approaches are proposed for robust performance through prediction aggregation. Among all the ensemble methods, the Bayesian voting method performs best and achieves F-score of 91.5%

Keywords Glomerulus segmentation · Deep learning · Ensemble learning · Convolutional neural network

1 Introduction

Glomerulus are clusters of capillaries which are responsible for expelling substances composed of waste and extra fluids unnecessary for the human body [1]. In daily practice, each renal biopsy should undergo a quantification of the total number of glomerulus found in each cut. The application of this work is to automate the renal histology analysis by using deep learning models.

Detection of glomerulus in digitized histological images has been approached using a variety of methods. The majority of studies incorporate domain-specific morphometric or texture-based techniques to search for and define glomerular boundaries. Many of these have demonstrated boundary detection for small image patches containing isolated glomerulus [2–4]. Translating detection techniques to whole-slide images (WSI) containing numerous glomerulus is a necessary but more difficult undertaking.

CNN-based semantic segmentation recently is one mainstream approach for glomerulus analysis. Semantic segmentation is a task of pixel-level classification. Most CNN-based semantic segmentation models adopt an encoder-decoder architecture. The encoder is in charge of the feature extraction, shrinking the spatial dimensions meanwhile increasing the depth. The decoder recovers the spatial information from the output of the encoder. Fully convolutional network (FCN) [5] is the first milestone of CNN based image segmentation model. It uses the backbone designed by Visual Geometry Group (VGG) [6] to extract features. Then, the features are upsampled to recover the input image size. However, due to the high upsampling rate, it is very challenging to extract precise details. Like FCN, Unet [7] also has a encoder-decoder architecture. The encoder consists of a stack of convolution, activation and pooling layers to capture the context in the input image. The decoder enables precise localization with transposed convolutions. The process combines the high level features and spatial information by a sequence of upconvolutions and concatenation with corresponding feature maps from the encoding path, as the low level feature maps from encoder carry better spatial information. And the abundance of feature channels in the upsampling process allows the network to propagate context information to higher resolution layers. On the other hand, Deeplabv3 [8] model is the first segmentation model using atrous spatial pyramid pooling modules. The atrous convolution can increase the receptive field without increasing the parameters. The Atrous Spatial Pyramid Pooling (ASSP) module

✉ Bingding Huang
huangbingding@sztu.edu.cn

1 Shenzhen Technology University, Shenzhen, Guangdong, China

2 Department of Advanced Intelligent Sensing, Shenzhen Academy of Robotics, Shenzhen, Guangdong, China

Published online: 08 February 2022

applies four atrous convolution with different dilation rate to combine multi-scale features.

The development of deep neural networks is facilitating more advanced digital analysis of histopathologic images. Marsh [9] et al. propose a CNN based model for identifying non-sclerosed and sclerosed glomerulus whole-slide images of donor kidney frozen section biopsies. A patch-based and a fully convolutional model is proposed. Blob detection is implemented as a post-processing step to generate the position and shape of each individual glomerulus. Hermsen [10] et al. train a convolutional neural network for multiclass segmentation of digitized kidney tissue sections stained with periodic acid–Schiff (PAS). The dataset is divided into five subsets and five individual Unet models are trained using each subset. The final result is the ensemble of the five models. Yao [11] et al. utilize a GoogleNet model with batch normalization to recognize different categories of glomerulus. Bayesian optimization is used for hyper parameters tuning. Kanan [12] et al. select Inception-v3 to recognize three classes: no glomerulus, NPS glomerulus and GS glomerulus. The input image is divided into small slices for detection purpose. Then, a traditional morphology technique is used to identify each glomerulus based on the generated heatmap. Altini et al. [13] introduce a Computer-Aided Diagnosis (CAD) system to assess global glomerulosclerosis. SegNet and DeepLab v3+ are used for semantic segmentation. Morphology operator is applied afterward to identify each individual instance. Heckenauer et al. [14] use YOLOv3 for real-time detection of glomerulus at different scales. The YOLOv3 model is pre-trained on the COCO dataset and fine tuned to detect glomerulus. Wetzer et al. [15] present a deep learning approach for glomerulus detection using two architectures, VGG16 (with batch normalization) and ResNet50. Three kinds of fusion approaches are implemented. The late fusion approach gives the best output. Xu et al. [16] propose a combination of Unet paired with optimal threshold detection via OSTU thresholding. The proposed method outperforms both Hessian-based blob detection and Unet with standard thresholding. Bueno et al. [17] first use Unet and SegNet for semantic segmentation. Then Alexnet is used for glomerulosclerosis detection. Gallego et al. [18] applies Alexnet model for glomerulus detection. The results indicate that the method is suitable for glomerulus detection in whole slide images. In summary, most of the existing work use a single or single type CNN model for glomerulus detection or segmentation. The potential of ensemble of various segmentation models has not been fully explored yet. The ensemble model may outperform each individual model through fusion mechanism. The contribution of our work is that we propose an effective ensemble of state-of-the-art segmentation models to fulfill glomerulus segmentation task. Three powerful semantic segmentation models which are FCN, Deeplabv3 and Unet are chosen for pixel-wise segmentation. FCN and Deeplabv3 models are upgraded by using Resnet backbone for feature extraction. Meanwhile, Unet is promoted to use EfficientNet backbone during encoding phase. Each of these models has its own advantages. We try to seek the complementary performance by using ensemble strategies.

2 Methodology

Semantic segmentation is also called pixel-level recognition. The semantic model usually has an encoding and an decoding process. The encoder learns useful features meanwhile downsampling the input image size. The decoder upsamples the features to have the dense prediction.

2.1 Data Preprocessing

In this work, the input image is the WSI images which have huge pixel numbers. Therefore, in order to feed the images to the segmentation model, we divided each image into small patches using a sliding window. The sliding window size is set as 1024×1024. It starts from the top left corner and slides till the right bottom corner. All the image patches are further downsampled. The image patches do not contain any glomerulus will be discarded. Meanwhile, in order to increase the quantity and diversity of the training data, we apply data augmentation technique. Three kinds of techniques are applied. First, images are flipped horizontally and vertically. Second, random contrast, random gamma and random brightness effects are applied. Third, elastic transform, grid distortion and optical distortion are added to create the variations.

2.2 Model

Glomerulus in high-resolution images are usually distributed randomly with vague borders and irregular shapes and sizes. Therefore, a multi-stream semantic segmentation model is proposed to handle this challenging task.

Three signature models are used in our model, the FCN-Resnet model, the Deeplabv3-Resnet model and the EfficientUnet model. The advantages of using Resnet [19] as backbone are as follows: 1. The model can have deeper convolutional layers. 2.The skip connection can maintain the background context of the image. 3.Resnet module has low amount of parameters. 4.Resnet module can solve the degradation problem.

The FCN-Resnet architecture is shown in Fig. 1a. The input is a rgb image. It starts with a 7×7 convolutional layer, a batch normalization layer and a maxpooling layer. Then, the model is followed by a stack of bottleneck modules for feature extraction. In each bottleneck module, a batch normalization layer is attached to each convolution layer

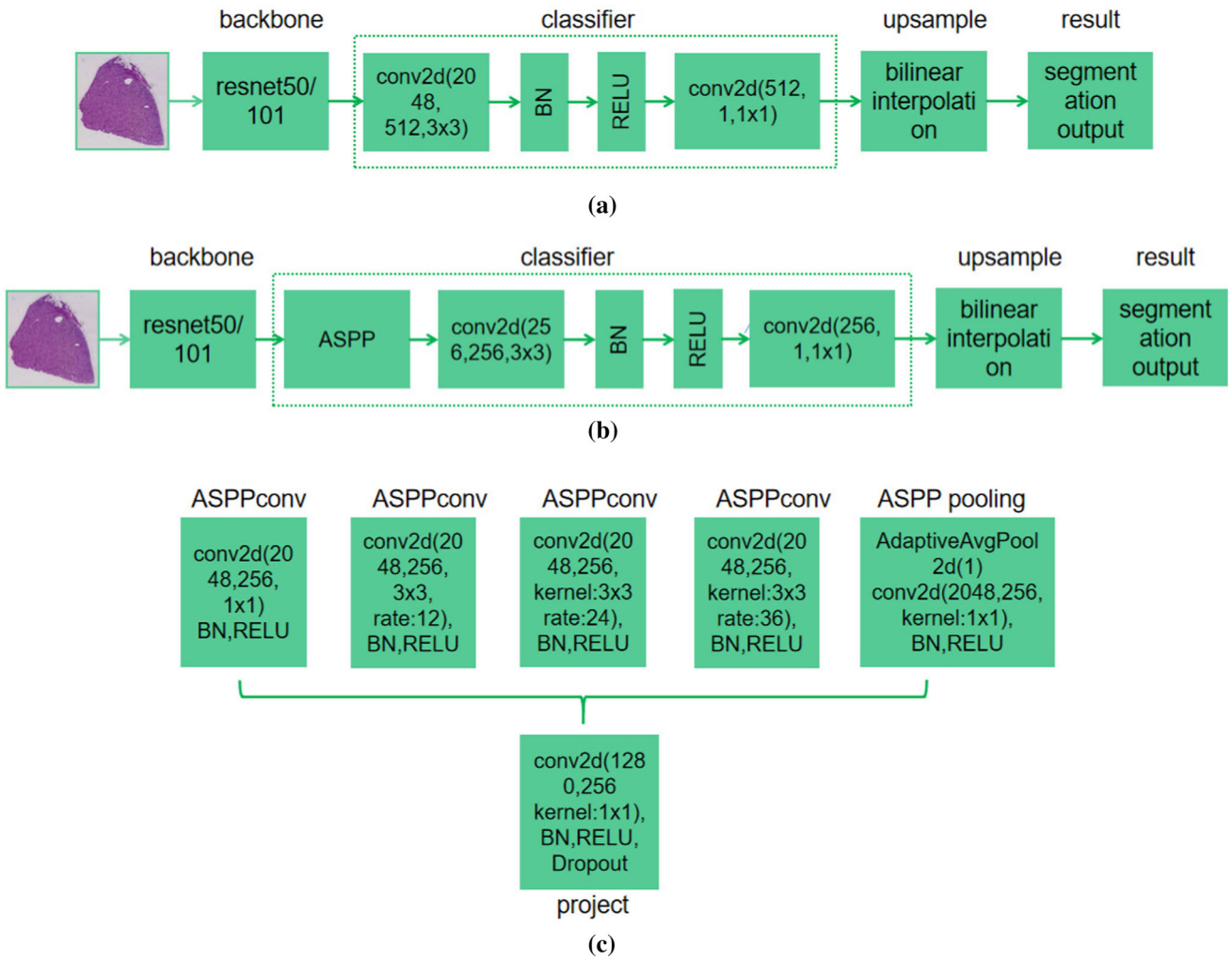


Fig. 1 **a** FCN-Resnet architecture. **b** Deeplabv3-Resnet architecture. **c** Atrous spatial pyramid pooling structure

Table 1 Resnet-50 and Resnet-101 backbone architectures

Layer name	50-layer	101-layer
conv1; bn	7×7, 64, stride 2; weights:64	
relu; 2d max pool	inplace; 3×3	
Layer1	$\begin{pmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{pmatrix} \times 3$	$\begin{pmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{pmatrix} \times 3$
Layer2	$\begin{pmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{pmatrix} \times 4$	$\begin{pmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{pmatrix} \times 4$
Layer3	$\begin{pmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{pmatrix} \times 6$	$\begin{pmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{pmatrix} \times 23$
Layer4	$\begin{pmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{pmatrix} \times 3$	$\begin{pmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{pmatrix} \times 3$

to regulate the data distribution. In this work, we consider two Resnet variants: Resnet-50 and Resnet-101. The configuration of these two is shown in Table 1. Each of them contains four layers. Each layer is a multiple repetition of a bottleneck module. The differences between Resnet-50 and Resnet-101 are the layer3, where the repetition numbers of

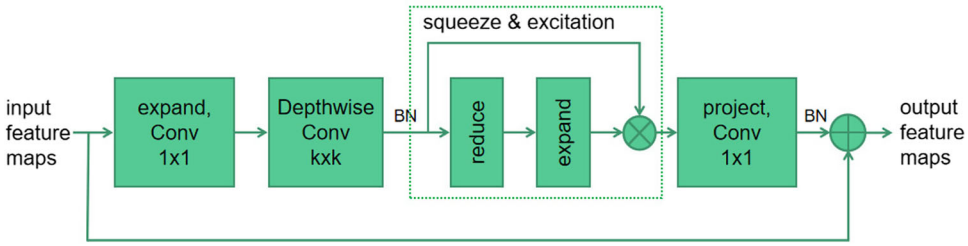


Fig. 2 MBconv building block

bottlenecks are 6 and 23, respectively. The output feature of the backbone has 2048 channel. The classifier diminishes the channel size to the number of the segmentation class using two convolution operators. Finally, the output of the classifier is upsampled to the size of the input through bi-linear interpolation.

The Deeplabv3-Resnet architecture is shown in Fig. 1b. Similar to FCN-Resnet, Deeplabv3-Resnet also uses Resnet as backbone. Deeplabv3 is the first model uses atrous convolution (dilation convolution) [20]. It increases the receptive field of the convolution by inserting multiple zero rows and columns in between convolution kernels. The output of the backbone which has 2048 channels is fed to the ASPP module. The structure of the ASPP is given in Fig. 1c. There are five branches: three dilation convolutions, one regular convolution with kernel size of 1×1 and a pooling. Each of the three dilation convolutions has kernel size of 3×3 generate features with 256 channels. The dilation rate is 12, 24 and 36, respectively. The output of these five blocks is concatenated to generate an output with 1280 channels. The projection layer takes the input with 1280 channels and output a feature with 256 channels. The output of the ASPP is further mapped to the number of classes using two convolution operators. Finally, the output is upsampled to have the dense prediction.

On the other hand, the Unet can also be combined with EfficientNet [21] backbone. The EfficientUnet uses an EfficientNet backbone as the encoder. EfficientNet is a compound scaling method which uniformly adjusts the network depth, width and resolution with a fixed set of scaling factors. There are eight variants of the EfficientNets, namely EfficientNetB0 to EfficientNetB7. The EfficientNet is fundamentally a stack of mbconv blocks. The mbconv block structure is shown in Fig. 2. The initial 1×1 convolution can either have 1 channel or 6 channels. The depthwise convolution can have kernel size of 3×3 or 5×5. The SE (squeeze and excitation) is to learn the correlation between channels which ends up giving each channel a weight. The channel weights multiply the output features of the depthwise convolution. Finally, the projection output and the input features are added to form the output features of the mbconv block.

The feature output from the backbone is fed into the decoder, where the features are upsampled four times using transpose convolution. After each transpose convolution, the features are concatenated with the corresponding encoder features. Then, a double convolution is applied. At the end, the output channel size equals the number of class defined. And the output size is the same as the input image.

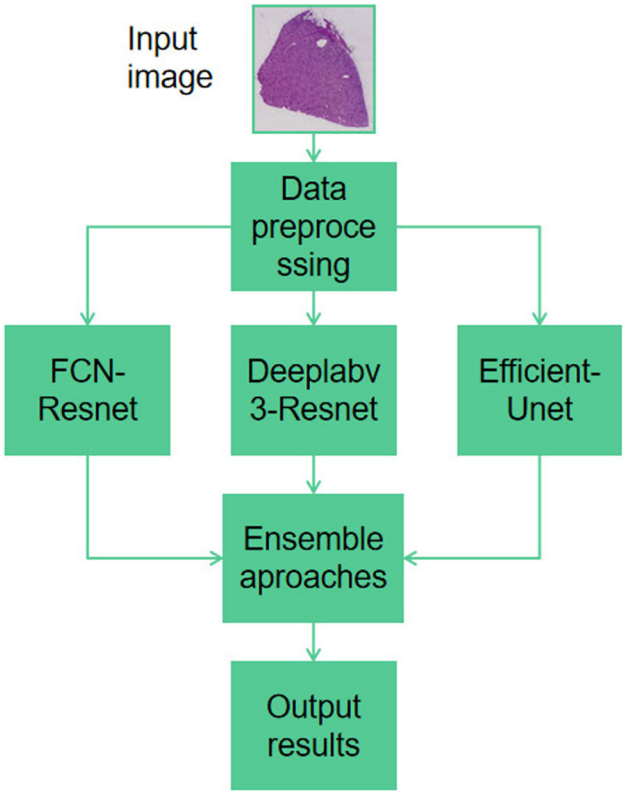


Fig. 3 Ensemble flowchart

2.3 Ensemble Approaches

The ensemble flowchart is shown in Fig. 3. The prediction of each segmentation model is a probability map. We apply three methods to combine the predictions of the segmentation models: (a) average voting, (b) major voting, (c) Bayesian voting. Average voting takes unweighted average of the probability maps for all the base learners and reports it as the

predicted score/probability [22].

$$P(y) = \text{average}\left(\sum_i P(y|i)\right) \tag{1}$$

Here, $P(y)$ is the final average voting output. i stands for the ith segmentation model; $P(y|i)$ is the probability map from model i.

Majority voting is similar to unweighted averaging. But instead of averaging over the output probability, it counts the votes of all the predicted labels from the base learners and makes a final prediction using label with most votes [23].

$$P(y) = \max\left(\sum_i P(y|i)\right) \tag{2}$$

In the Bayesian voting approach, the goal is to calculate the maximum posterior probability (MAP), $P(y|x)$, where x is the input image and y is the pixel-wise prediction. According to Bayesian rule:

$$P(y|x) = \sum_i P(y, i|x) \propto \sum_i P(y|i) * p(i) \tag{3}$$

$P(i)$ is the prior of the ith model which is learned from the training dataset.

2.4 Loss Function

Here, two kinds of loss functions are combined, the BCE (binary cross entropy) loss [24] and the Dice loss.

$$\text{BCE loss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)) \tag{4}$$

where N is the number of pixel of a training image. y_i is the prediction confidence of the ith pixel. $\hat{y}_i$ is the ground truth of the ith pixel. The BCE loss focuses on the pixel-wise performance. Meanwhile, we also include a global loss measure, Dice loss [25] which is defined as follows:

$$\text{Dice loss} = \frac{2\,|A \cap B|}{|A| + |B|} \tag{5}$$

$|A \cap B|$ is the element-wise multiply of the prediction map A and the ground truth map B. $|A| + |B|$ is the element-wise addition of the prediction and the ground truth. To take into both local and global loss into account, we have the final loss:

$$\text{Final loss} = 0.8 \times \text{BCE} + 0.2 \times \text{Dice} \tag{6}$$

3 Experiments and Results

Semantic segmentation applying FCN-Resnet, Deeplabv3-Resnet and EfficientUnet is performed categorizing kidney tissue into glomerulus and background. FCN-Resnet has two variants: FCN-Resnet50 and FCN-Resnet101. Similarly, Deeplabv3-Resnet50 and Deeplabv3-Resnet101 are considered. EfficientNetUnet has 8 variants which use EfficientNet B0 to B7 as backbone.

3.1 Hardware and Software Platform

The backend server has Intel Xeon(R) Gold 6230R CPU with 2.10GHz frequency. The server equips with eight Nvidia Tesla V100s GPUs. Each GPU has 32GB memory. To fully take the advantages of the GPU resources, distributed data parallel [26] mechanism is applied. This technique replicates the model on every computational resource to generate gradients independently and then communicates those gradients at each iteration to keep model replicas consistent. The programming is implemented using Pytorch framework.

3.2 Dataset

Just as the Human Genome Project mapped the entirety of human DNA, the Human BioMolecular Atlas Program (HuBMAP) [27] is a major endeavor. Sponsored by the National Institutes of Health (NIH), HuBMAP is working to catalyze the development of a framework for mapping the human body at a level of glomerulus functional tissue units for the first time in history. There are over 600, 000 glomerulus in each human kidney. The Human BioMolecular Atlas Program (HuBMAP) data used in this paper include 15 Formalin Fixed Paraffin Embedded (FFPE) PAS (Periodic Acid Schiff) [28] kidney images. Glomerulus FTU annotations exist for all 15 tissue samples. The annotations are in both RLE-encoded and unencoded (JSON) forms. The annotations denote segmentations of glomerulus. The individual image size ranges from 500MB to 5GB. One of the samples is shown in Fig. 4. From the sample, it is showed that glomerulus structures present high variability in terms of size, shape and color. The reasons of this high variability in samples are multiple: the relative positions of the glomerulus inside the renal section, the heterogeneity in immunohistochemistry staining or the presence of internal biological processes. In a healthy kidney before sectioning, glomerulus present a spherical shape with fixed size (diameter ranges between 350 and 100 m), but its aspect can change due to the presence of medical diseases. For instance: glomerulus can present a swell aspect under hypertension [29] or diabetes [30] conditions. After sectioning, the presence of pathologies affects the appearance. Besides, different glomerulus sizes observed

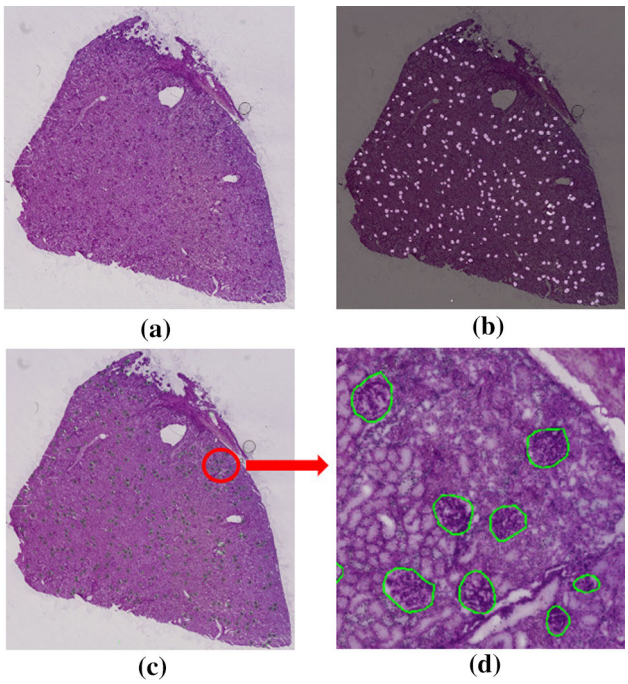


Fig. 4 One data sample. The image has 38, 160 × 39, 000 pixels. **a** The raw image. **b** The image and the labeled mask. **c** The image and the labeled contours. **d** The enlarged labeled contours

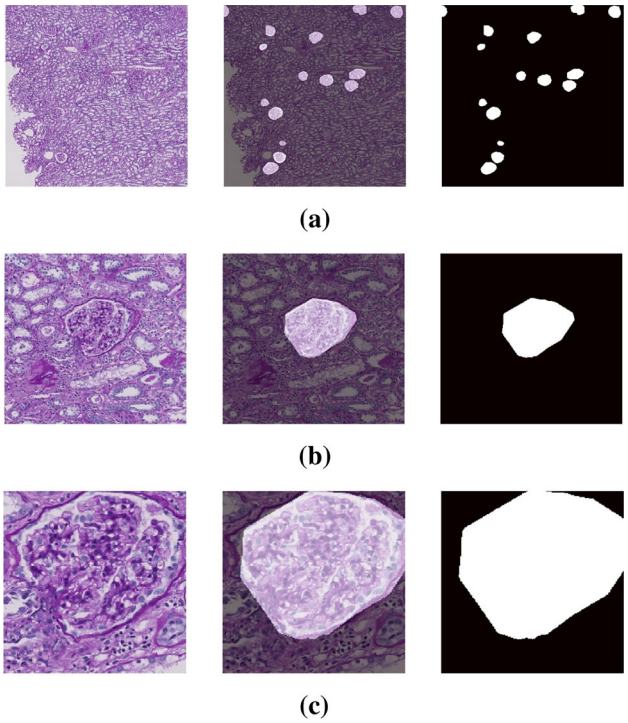


Fig. 5 Image patch, the image patch with masks and the binary masks. **a** 2500 × 2500 pixels. **b** 470 × 500 pixels. **c** 200 × 200 pixels

could vary depending on where the cross section was taken with respect to each glomerulus sphere.

The patched images and masks with different sizes are shown in Fig. 5. The dataset also includes anatomical structure annotation to label various parts of the tissue.

3.3 Training Process

Transfer learning from natural image datasets using standard large models and corresponding pretrained weights has become a popular method for deep learning applications to medical imaging [31]. Therefore, transfer learning approach is used in this work. These segmentation models are pretrained using Pascal VOC dataset [32]. Then, all the models are trained using the WSI images. The training procedure of each model is described in Algorithm 1.

There are totally 15 WSI images which can generate around 18000 image patches; 4500 of them contain glomerulus. Four Fifths of them are used for training and the remaining one fifth are used for testing. Cross-validation is implemented. One seventh of the training data is used for validation. The optimization algorithm is AdamW [33] which adds L-2 regularization and weight decay elements on top of Adam. The learning rate is 0.0001. The weight decay is 0.001. The model which has the best validation loss will be saved. The following parameters are tuned,

Algorithm 1 glomerulus segmentation model training process

Input: img: WSI images; $best\_vloss$: minimum validation loss;
Output: binary labels of each WSI image;
data augmentation;
create image patches from original WSI images;
divide the image patches into training set and validation set;
define the loss function;
load the pretrained weights;
set training parameters: $epochs$, $batch\_size$, LR, $weight\_decay$, $optimizer$;
repeat
  feed the training patches to the model;
  update the weights using the optimizer;
  calculate the validation loss $vloss$ using the updated weights;
  if "$vloss < best\_vloss$" **then** "$vloss = best\_vloss$, save the model weights;"
  end if
until $epochs = 0$

- image size: The image patch size used for training and testing. Segmentation model can have arbitrary input image size. However, for training efficiency purpose, two sizes are considered 256 and 512.
- overlap: the overlap of two adjacent image patches. To eliminate the edge effects during image slicing, adjacent patches maintains certain overlaps. The overlaps are 32 and 64 pixels for image of size 256×256 and 512×512, respectively.

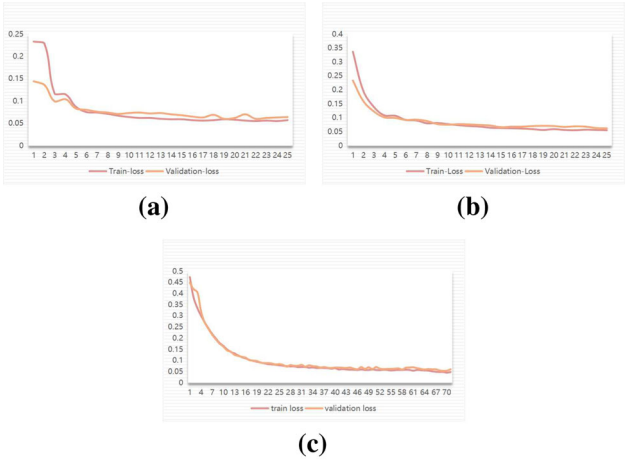


Fig. 6 **a** Training curve of model (**a**) FCN-Resnet-50. **b** FCN-Resnet-101. **c** EfficientUnet-B7

- batch size: The batch size is an important parameter of training. Small batch size may lead to unstable training while big batch size may slow down the training process drastically. Therefore, given the training dataset, two batch sizes are considered 16 and 32.

Each model is trained using two sets of parameters. Figure 6-a shows one training curve of FCN-Resnet50; it is shown that both the training and validation loss converges at epoch 17. The training is stopped after 25 epochs. The minimum training loss is 0.056 while the minimum validation loss is 0.061. On the other hand, the minimum training loss of FCN-Resnet101 is 0.055 and the minimum validation loss is 0.060. The average training time of one epoch is 1.67 minutes and 1.89 minutes, respectively, since the FCN-Resnet101 has more parameters. Compared to FCN-Resnet, the Deeplabv3-Resnet50 and Deeplabv3-Resnet101 have similar training curves. The best training loss of these two model is 0.053 and 0.052. The best validation loss of these two models is 0.058 and 0.057.

Figure 6c shows the training curves of EfficientUnet-B7 which take longer to converge compared to FCN-Resnet and Deeplabv3-Resnet. The training curve becomes stable after 40 epochs. The best training and validation losses are 0.047 and 0.051. Table 2 lists all the training results using different parameters. The table shows that all the models are trained successfully. Both the training and validation loss converge after certain epochs. The FCN-reset50 and FCN-Resnet101 models have comparable training curves. The EfficientUnet-B7 has the best validation loss of 0.051. From all training

Table 2 Training and validation loss with different parameters

Model	Image size	Epoch	Overlap	Best train loss	Best validation loss
FCN-Resnet50	256 × 256	25	32	0.057	0.061
	512 × 512	25	64	0.056	0.061
FCN-Resnet101	256 × 256	25	32	0.055	0.060
	512 × 512	25	64	0.056	0.063
Deeplabv3-Resnet50	256 × 256	25	32	0.054	0.058
	512 × 512	25	64	0.053	0.059
Deeplabv3-Resnet101	256 × 256	25	32	0.052	0.057
	512 × 512	25	64	0.058	0.062
EfficientUnet-B0	256 × 256	70	32	0.054	0.062
	512 × 512	70	64	0.055	0.063
EfficientUnet-B1	256 × 256	70	32	0.055	0.062
	512 × 512	70	64	0.060	0.065
EfficientUnet-B2	256 × 256	70	32	0.058	0.060
	512 × 512	70	64	0.059	0.063
EfficientUnet-B3	256 × 256	70	32	0.059	0.061
	512 × 512	70	64	0.061	0.063
EfficientUnet-B4	256 × 256	70	32	0.056	0.058
	512 × 512	70	64	0.059	0.062
EfficientUnet-B5	256 × 256	70	32	0.050	0.055
	512 × 512	70	64	0.052	0.056
EfficientUnet-B6	256 × 256	70	32	0.048	0.052
	512 × 512	70	64	0.049	0.053
EfficientUnet-B7	256 × 256	70	32	0.047	0.051
	512 × 512	70	64	0.048	0.053

Table 3 Performance metrics applied

Metric	Equation
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$
Recall	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
F-score	$\frac{2*Precision*Recall}{Precision+Recall}$
MCC	$\frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}}$

results, it is shown that the model trained with 256 × 256 patches performs better with its counterpart model trained with 512 × 512 patches. Therefore, all the models trained with 256 × 256 patches are used in the testing phase.

3.4 Results

In the segmentation process, these models are tested on 3 WSI images. The testing process of each model is described in Algorithm 2. The CNN output is fed to the sigmoid function to have the final score. If the score is larger than 0.5, the output is 1 (glomerulus), otherwise 0 (background). The final score map has the size of the input. Therefore, the patched segmentation output is merged together to generate the score map of each WSI image. The average computational time of the segmentation model is 382s per WSI image.

There are totally four possible results: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Based on these values, the performance metrics shown in Table 3 are calculated. These metrics averages are calculated over cross-validation runs.

Algorithm 2 glomerulus segmentation model testing process

Input: img: WSI images;
Output: performance metrics;
1: create image patches from testing WSI images;
2: feed testing patches to the model and obtain the results.
3: calculated the performance metrics based on the model outputs and the ground truth.

The calculated metrics are shown in Table 4. All the models have high accuracies. The reason is the unbalance data distribution, since in each image patch, there are much more TN pixels than TP pixels. The F-score which is harmonic mean of precision and recall is the most important metric. According to the F-score, FCN-Resnet101 and Deeplabv3-Resnet101 outperform FCN-Resnet50 and Deeplabv3-Resnet101, respectively. Since the deeper network introduces more non-linearity, it has stronger representative capability. Similarly, as EfficientUnet goes deeper from B0 to B7, the performance keeps improving. Among all the individual models, the EfficientUnet B7 has the best metric scores. The reasons are as follows: 1. All the training images are in same scales. 2. Although the edges of the glomerulus is very complicated, the Unet takes both low-level and high level information using concatenation. The low-level information offers raw position of the target clusters. Meanwhile, the high-level info provides clues for precise segmentation. 3. The EfficientNet-b7 as the backbone can elevate the representative capability of the encoder. The ROC curve of the Bayesian model is drawn in Fig. 7. The area under curve (AUC) is 0.834.

Further, the results of the ensemble methods are given. To fuse the decision of all the base learners, the top 5 best models are selected to eliminate redundancies. The average voting

Table 4 Calculated metrics. The top 5 models (labeled in bold) are used in ensemble approaches

Model	Accuracy	Recall	Precision	F-score	MCC
FCN-Resnet50	0.952	0.872	0.855	0.863	0.862
FCN-Resnet100	0.953	0.881	0.867	0.874	0.873
Deeplabv3-Resnet50	0.962	0.885	0.870	0.877	0.876
Deeplabv3-Resnet100	0.963	0.894	0.882	0.888	0.885
EfficientUnet-B0	0.963	0.855	0.840	0.847	0.845
EfficientUnet-B1	0.965	0.862	0.837	0.849	0.850
EfficientUnet-B2	0.962	0.866	0.842	0.854	0.853
EfficientUnet-B3	0.965	0.872	0.863	0.867	0.866
EfficientUnet-B4	0.964	0.881	0.866	0.873	0.874
EfficientUnet-B5	0.963	0.902	0.880	0.890	0.889
EfficientUnet-B6	0.967	0.913	0.892	0.902	0.901
EfficientUnet-B7	0.968	0.918	0.895	0.906	0.906
Average voting	0.972	0.920	0.900	0.910	0.911
Major voting	0.974	0.918	0.898	0.908	0.907
Bayesian voting	0.975	0.922	0.908	0.915	0.914

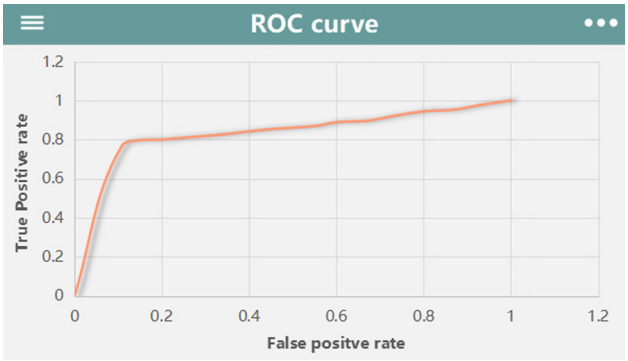


Fig. 7 ROC curve of the Bayesian voting model

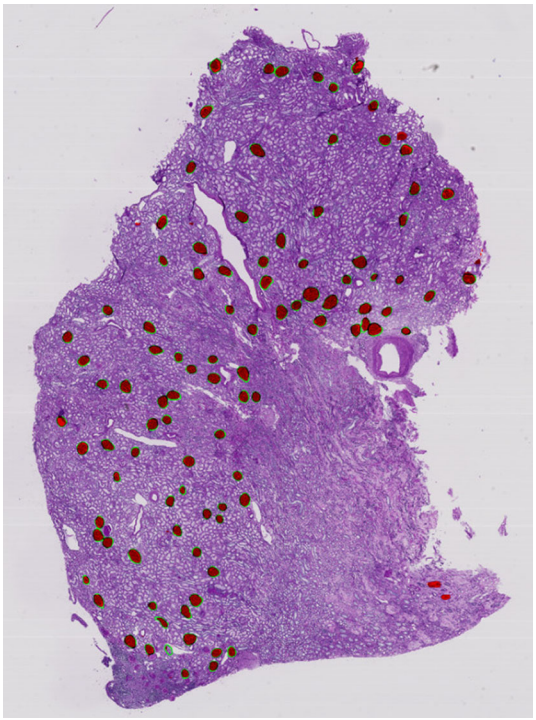


Fig. 8 One testing result. The ground truth is given by green contours. The segmented glomerulus pixels are displayed in red. Image ID: *aaa6a05cc*

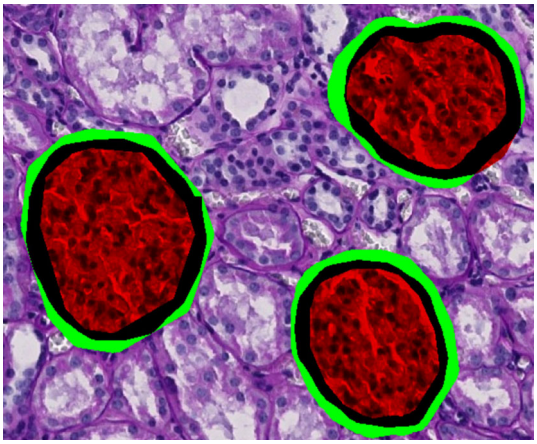


Fig. 9 Successful glomerulus segmentation from image ID: *aaa6a05cc*

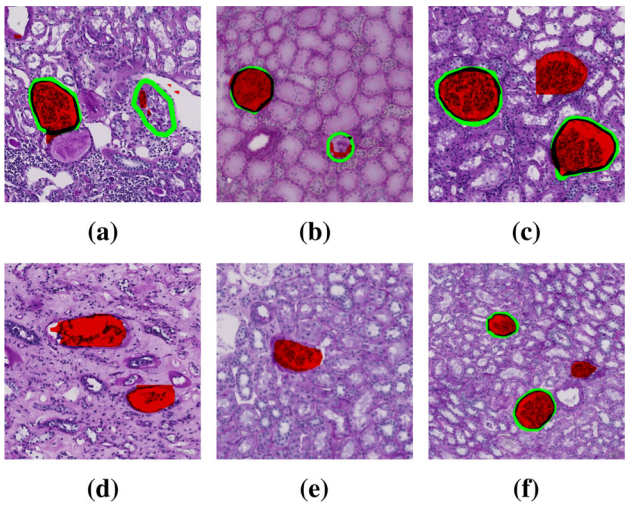


Fig. 10 Failure cases. **a**, **b** False negatives. **c**–**f** False positives

and major voting have comparable outputs, and both of them outperform the individual models. Since taking the average of multiple networks reduces the variance, CNNs have high variance and low bias. Compared to naive averaging, majority voting is less sensitive to the output from a single network. In the Bayesian voting approach, each base learner is viewed as

Table 5 Performance comparison with state of the art

Model and year	Task	Performance metrics
Yolov3 [14], 2020	Glomerulus detection	F-score: 48%
Inception V3 [12], 2019	3 classes recognition	F-score: 62.3%
Unet [10], 2019	10 classes segmentation,	F-score: 84%
FCN+blob detection [9], 2018	3 classes segmentation	F-score: 91.4%
GoogLeNet-BN-Bayesian model [11], 2020	3 classes recognition	F-score: 92.7%
SegNet-VGG19 [17], 2019	Glomerulus segmentation	F-score: 99.24%
Our approach ensemble model	Glomerulus segmentation	F-score: 91.5%

an hypothesis made on the functional form of the conditional distribution of y given x. The final output can be interpreted as a weighted voting sum from each base learner. The weight is the prior distribution of each model learned from the training.

One of the segmentation results of Bayesian voting is given in Fig. 8. It shows that the ensemble approach can segment majority of the target glomerulus successfully. Figure 9 gives an enlarged segmentation of three glomerulus. The segmentations are successful however, not perfect. The contours of the pixels belong to glomerulus are not strictly aligned with the ground truth. Figure 10a and b gives two false negative cases. The iregular shapes and textures of the glomerulus may cause this failure. Compared to false negatives, there are more false positive cases, since certain parts of the background look like glomerulus. More testing results are presented in Fig. 11 in Appendix. On the other hand, the performance of our work is compared with recent related works. Unlike object detection realm, there are very limited open source glomerulus datasets available. Although each of the related works provides performance metric F-score which is the harmonic mean of precision and recall, all of these related works use their own private dataset for model training. Both glomerulus detection and segmentation models are included in the comparison. However, glomerulus semantic segmentation task (pixel-wise labeling) is more challenging than glomerulus detection task (bounding box regression). Table 5 shows that the proposed model outperform most of the existing work. Although [11] has F-score of 92.7%, it is only a detection model. The segmentation task we are handling is more challenging. [17] has F-score of 99.24%. The model is trained using 47 WSI images. In comparison, our training dataset only has 15 WSI images. Therefore, our model is more training efficient. Overall, our proposed model is among the best performance models.

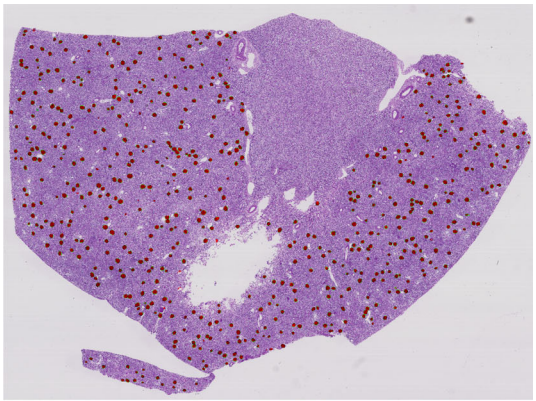
4 Conclusions and Future Works

This paper has presented a methodology to segment glomerulus effectively. Several previous studies have applied deep learning methodologies for glomerulus detection but, as far as the authors know, none of them have been focused on using FCN-Resnet, Deeplabv3-Resnet and EfficientUnet segmentation models together, which makes this paper a novel contribution in this field. Further, we construct ensemble strategies of these models, with the aim of decreasing the variance and with its model- and configuration-specific behaviors. The experimental results prove the effectiveness of each individual model. Further, the ensemble model outperforms each individual model. The Bayesian voting approach gives the best performance with F-score 91.5%.

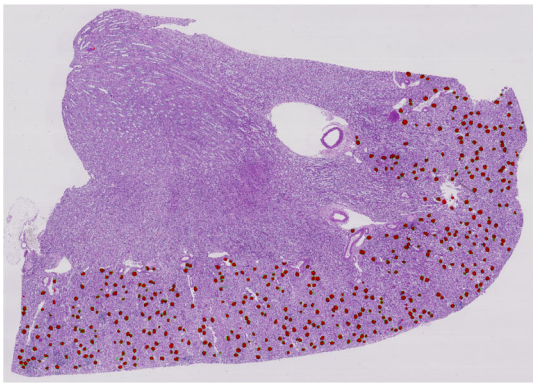
The future works are as follows: First, currently, each segmentation model is trained separately. We will try to build an end to end model so that these models can be trained simultaneously. Secondly, other than the pixel-wise labels, the positions of each individual glomerulus will be produced by our future model. Thirdly, the model will be further enhanced by adding more glomerulus labels, such as sclerosed glomerulus and normal glomerulus.

Funding This project is supported by the National Natural Science Foundation of China (No. 61906123), The Fundamental Research Funds from Shenzhen Technology University, Natural Science Foundation of Top Talent of SZTU (Grant No. 2018010801008).

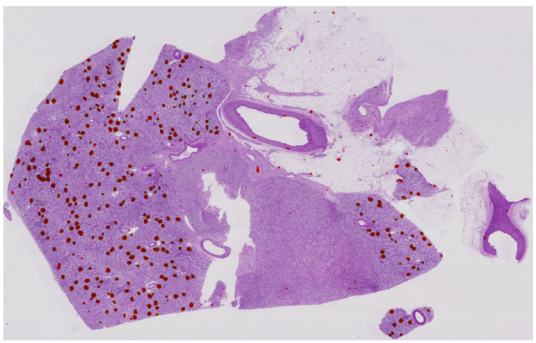
5 Appendix



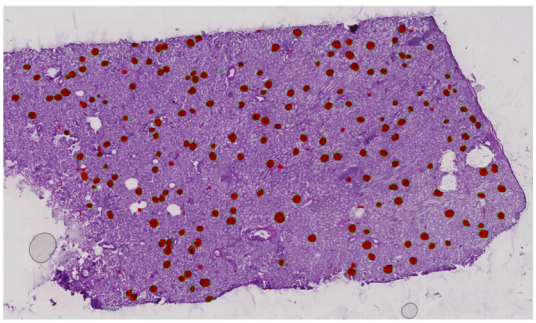
(a)



(b)



(c)



(d)

Fig. 11 More testing results. The ground truth is given by green contours. The segmented glomerulus pixels are displayed in red. Image ID: **a** 8242609fa. **b** $b9a3865fc$. **c** $cb2d976f4$. **d** $e79de561c$

References

1. Wilbur, D.C.; Smith, M.L.; Cornell, L.D.; Andryushkin, A.; Pettus, J.R.: Automated identification of glomeruli and synchronized review of special stains in renal biopsies by machine learning and slide registration: a cross-institutional study. Histopathology (2021)
2. Deng, S.; Zhang, X.; Yan, W.; Chang, I.C.; Xu, Y.: Deep learning in digital pathology image analysis: a survey. Front. Med., no. 6 (2020)
3. Samsi, S., Jarjour, W.N., Krishnamurthy, A.: Glomeruli segmentation in h and e stained tissue using perceptual organization. In: Signal Processing in Medicine and Biology Symposium (2013)
4. Ma, J.; Zhang, Hu, J.: Glomerulus extraction by using genetic algorithm for edge patching. In: IEEE Congress on Evolutionary Computation (2009)
5. Long, J.; Shelhamer, E.; Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2015)
6. Simonyan, K.; Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Computer Science (2014)
7. Ronneberger, O.; Fischer, P.; Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. Springer, Cham (2015)
8. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2018)
9. Marsh, J.N.; Matlock, M.K.; Kudose, S.; Liu, T.C.; Stappenbeck, T.S.; Gaut, J.P.; Swamidass, S.J.: Deep learning global glomerulosclerosis in transplant kidney frozen sections. IEEE Trans. Med. Imaging **37**(12), 2718–2728 (2018)
10. Hermsen, M.; Bel, T.; Boer, M.; Steenbergen, E.; Kers, J.; Florquin, S.; Roelofs, J.; Stegall, M.; Alexander, M.; Smith, B.; Smeets, B.; Hilbrands, L.; van der Laak, J.: Deep learning-based histopathologic assessment of kidney tissue. J. Am. Soc. Nephrol. 30, ASN.2019020144, 09 (2019)
11. Yao, X.; Wang, X.; Karaca, Y.; Xie, J.; Wang, S.: Glomerulus classification via an improved googlenet. IEEE Access **8**, 176916–176923 (2020)
12. Kannan, S.; Morgan, L.A.; Liang, B.; Cheung, M.; Kolachalama, V.B.: Segmentation of glomeruli within trichrome images using deep learning. Kidney International Reports, 4(7) (2019)
13. Altini, N.; Cascarano, G.D.; Brunetti, A.; Marino, F.; Rocchetti, M.T.; Matino, S.; Venere, U.; Rossini, M.; Pesce, F.; Gesualdo, L.; Bevilacqua, V.: Semantic segmentation framework for glomeruli detection and classification in kidney histological sections. Electronics, 9(3) (2020)
14. Heckenauer, R.; Weber, J.; Wemmert, C.; Feuerhake, F.; Forestier, G.: Real-time detection of glomeruli in renal pathology. In: 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS) (2020)
15. Wetzer, E.; Lindblad, J.; Sintorn, I.M.; Hultenby, K.; Sladoje, N.: Towards Automated Multiscale Imaging and Analysis in TEM: Glomerulus Detection by Fusion of CNN and LBP Maps: Munich, Germany, September 8–14, 2018, Proceedings, Part VI. Computer Vision—ECCV 2018 Workshops (2019)
16. Xu, Y., Gao, F., Wu, T., Bennett, K.M.., Sarkar, S.: U-net with optimal thresholding for small blob detection in medical images. In: 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE) (2019)
17. Gb, A.; Fc, A.; Gl, B.; Od, A.: Glomerulosclerosis identification in whole slide images using semantic segmentation. Comput. Methods Progr. Biomed., **184** (2019)

18. Jaime, G.; Anibal, P.; Samuel, L.; Georg, S.; Lucia, G.; Arvydas, L.; Gloria, B.: Glomerulus classification and detection based on convolutional neural networks. J. Imaging **4**(1), 20–20 (2018)
19. He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition (2015)
20. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H.: Rethinking atrous convolution for semantic image segmentation. CoRR (2017)
21. Tan, M.; Le, Q.: "EfficientNet: rethinking model scaling for convolutional neural networks. In: Chaudhuri, K, Salakhutdinov, R. (Eds.) Proceedings of the 36th International Conference on Machine Learning vol. 97 of Proceedings of Machine Learning Research, pp. 6105–6114, PMLR, 09–15 (Jun 2019)
22. Yazdizadeh, A.; Patterson, Z.; Farooq, B.: Ensemble convolutional neural networks for mode inference in smartphone travel survey. IEEE Trans. Intell. Transp. Syst. **21**(6), 1–8 (2019)
23. Kamnitsas, K.; Bai, W.; Ferrante, E.; Mcdonagh, S.; Sinclair, M.; Pawlowski, N.; Rajchl, M.; Lee, M.; Kainz, B.; Rueckert, D.: Ensembles of Multiple Models and Architectures for Robust Brain Tumour Segmentation. Springer, Cham (2017)
24. Ruby, U.; Yendapalli, V.: Binary cross entropy with deep learning technique for image classification. Int. J. Adv. Trends Comput. Sci. Eng., **9**(4) (2020)
25. M. F, Navab, N.; Ahmadi, S. A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV) (2016)
26. Li, S.; Zhao, Y.; Varma, R.; Salpekar, O.; Noordhuis, P.; Li, T.; Paszke, A.; Smith, J.; Vaughan, B.; Damania, P.; Chintala, S.: Pytorch distributed: experiences on accelerating data parallel training. CoRR (2020). arXiv:2006.15704
27. Snyder, M.P.; Lin, S.; Posgai, A.; Atkinson, M.; Regev, A.; Rood, J.; Rosen, O.; Gaffney, L.; Hupalowska, A.; Satija, R.: The human body at cellular resolution: the nih human biomolecular atlas program. Nature **574**(7777), 187–192 (2019)
28. Crist, H.; Hennessy, M.; Hodos, J.; Mcginn, J.; White, B.; Payne, S.; Warrick, J.I.: Acute invasive fungal rhinosinusitis: Frozen section histomorphology and diagnosis with pas stain. Head Neck Pathol. (2019)
29. Hughson, M.; Puelles, V.G.; Hoy, W.E.; Douglas-Denton, R.N.; Mott, S.A.; Bertram, J.F.: Hypertension, glomerular hypertrophy and nephrosclerosis: the effect of race. In: Nephrology, Dialysis, Transplantation: Official Publication of the European Dialysis and Transplant Association - European Renal Association, no. 7, p. 1399 (2014)
30. Rasch, R.; Lauszus, F.; Thomsen, J.S.; Flyvbjerg, A.: Glomerular structural changes in pregnant, diabetic, and pregnant-diabetic rats. Apmis Acta Pathol. Microbiol. Immunol. Scand. **113**(7–8), 465–472 (2010)
31. Raghu, M.; Zhang, C.; Kleinberg, J.; Bengio, S.: Transfusion: understanding transfer learning for medical imaging. CoRR (2019).
32. Shetty, S.: Application of convolutional neural network for image classification on pascal VOC challenge 2012 dataset. CoRR (2016). arXiv:1607.03785
33. Loshchilov, I.; Hutter, F.: Fixing weight decay regularization in ADAM. CoRR (2017). arXiv:1711.05101