

Automated Detection of Segmental Glomerulosclerosis in Kidney Histopathology

Ikaro Campos de Araújo¹, Leizer Schnitman¹, Angelo Amancio Duarte², and Washington LC dos-Santos³

¹ Institute of Mathematics and Polytechnic School,
Federal University of Bahia, Brazil,

`ikarocampos75@gmail.com`, `leizer@ufba.br`

² Department of Technology,
State University of Feira de Santana, Brazil,
`angeloduarte@uefs.br`

³ Instituto Gonçalo Moniz,
Fundação Oswaldo Cruz, Brazil,
`wluis@bahia.fiocruz.br`

Abstract. This paper presents the current results in the detection of segmental glomerulosclerosis by analyzing histological images of kidney biopsies, stained using hematoxylin and eosin (H&E) or periodic acid-Schiff (PAS) techniques. The work is part of the development of the PathoSpotter-K system, which aims the detection of elemental lesions in histological images of kidney. Currently, PathoSpotter-K accuracy for detecting segmental glomerulosclerosis is 84.8% for H&E stained samples and 81.3% for PAS stained samples. Such rates are similar to that reported for most of the analogous systems used for histological lesions detection in other organs and diseases.

1 Introduction

The kidney diseases have a global incidence, affecting millions of people around the world. A relevant fraction of these diseases affect the glomeruli (glomerulopathies), extending throughout the nephron, which constitutes the functional unit of the kidney. Often, glomerulopathies affect a considerable number of glomeruli, leading to chronic renal failure.

Glomerular diseases are the third main cause of end stage renal disease in Brazil[21], and are associated with 30%-60% of the cases of renal failure in Asia and Oceania, and 10%-15% of the cases of renal failure in Europe and United States of America [19].

Glomerulopathies are associated with a variety of structural changes in the glomeruli. These changes, defined as lesion pattern, frequently combine different elemental lesions.

A frequent type of lesion pattern is the segmental glomerulosclerosis, which is characterized by an increase in the glomerular extra-cellular matrix, causing the obliteration of the glomerular capillary lumen/lumina.

Segmental glomerulosclerosis is the morphological substrate of the two most frequent glomerulopathies: Focal Segmental Glomerulosclerosis (FSGS) and IgA Nephropathy. In both diseases, the glomerulosclerosis has a progressive behavior, becoming global and leading to chronic renal failure [6, 1].

As in any medical diagnostic based on histological images, glomerulosclerosis detection is strongly dependent on the pathologist's expertise. Developments in computational image analysis made computer-assisted diagnostic in pathology a promising research field in medicine and computing [9, 4]. These systems became of great interest in pathology since the beginning of histological image digitalization and subsequent building of digital histological image libraries. Such context made it possible to apply computer image analysis systems on searching for histological lesions based on structural tissue patterns [9, 23]. Since then, the computer-aided diagnosis systems have been envisaged as an important tool for decision support and research in pathology [15].

Most of the literature on computer-aided diagnosis systems deals with cancer related images [4] [23] [16]. There are few studies dedicated to renal histology, such as Kato and colleagues [12], who proposed a new descriptor, referred as Segmental Histogram of Oriented Gradients (Segmental HOG), for glomeruli detection in kidney histopathological images, achieving a precision of 87,4%, and Barros and colleagues [3] working on the PathoSpotter-K project, which proposed a computational tool that achieves an accuracy of 88.3% +/- 3.6% for automatic identification of proliferative glomerular lesions in histological images from kidney biopsies.

In this paper, we present the current developments in the PathoSpotter-K software, making it possible the computational detection of segmental glomerular sclerosis in H&E- and PAS-stained kidney sections. The PathoSpotter-K system is currently under development and aims to be a research and decision support tool for pathologists, enabling large-scale clinical-pathological associations and training of young pathologists. PathoSpotter-K is the resulting work of an interdisciplinary team, composed of pathologists and computer scientists.

2 Methodology

Our methodology used a classical image processing and pattern recognition approach to detect segmental glomerulosclerosis lesions in digital histological images of kidneys.

One glomerulus is characterized by some structures as shown in Fig. 1(a) and Fig. 1(b).

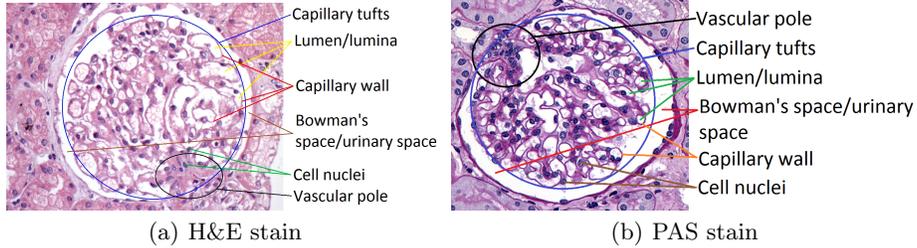


Fig. 1. Glomerular structure in a histological sections stained with (a) H&E and (b) PAS.

Glomerulus with sclerosis is characterized by the increase of the glomerular extra-cellular matrix, causing the obliteration of the glomerular capillary lumen/lumina, as shown in Fig. 2(a) and Fig. 2(b).

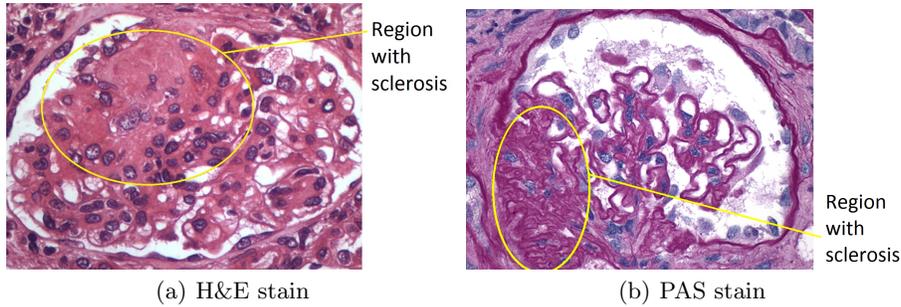


Fig. 2. Glomerulosclerosis histological images stained in (a) H&E and (b) PAS techniques

We started by applying the methods successfully used for analysing other diseases, according to similar works, that used texture identification algorithms to extract features such as those present in glomerulosclerosis. These methods were:

- Haralick texture features [10, 7, 11, 14, 13, 15, 2];
- Gabor filter [7, 4, 26];
- Local Binary Pattern (LBP) [22, 25];
- First-order histogram statistics [4, 15, 16, 20].

The code was developed in Python 2.7 using the packages: scikit-image [24], scikit-learn [18] and mahotas [5].

2.1 Dataset

The dataset was built by pathologists from the Gonçalo Moniz Institute from Oswald Cruz Foundation (FIOCRUZ/IGM) at Brazil, and consists of 499 glomeruli

images stained in H&E technique, being 267 with glomerulosclerosis and 232 of normal glomeruli, and 248 glomeruli images stained in PAS technique, being 155 with glomerulosclerosis and 93 of normal glomeruli. All images have 1024x768 pixels, RGB color space, and represent a single glomerulus, as shown in Fig.1.

2.2 System Architecture

The system architecture follows the typical structure of a digital image processing with pattern recognition system [8] and consists of three steps: preprocessing, feature extraction and classification/validation (see Fig. 3).

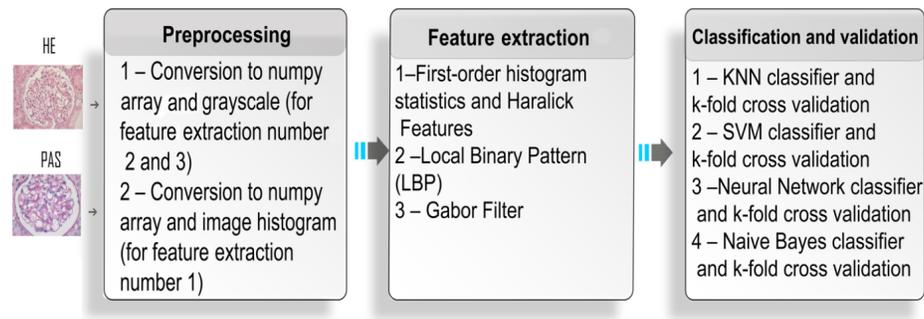


Fig. 3. System architecture

Preprocessing In this step, the images were prepared to facilitate the feature extraction process. Three operations were performed: conversion to numpy array and to gray scale, and computing the image histogram.

The conversion to numpy array allows the representation of the pixels of the image in a Python numpy array, to be processed and analyzed by the feature extraction methods.

The conversion to gray scale facilitates the image processing, because it reduces the operations with the color channels from three (Red, Blue and Green) to one (Gray). This operation is also important, because the Haralick texture features are applied over the Gray-Level Co-Occurrence Matrix (GLCM), preparing the image for feature extraction.

The image histogram is computed to be calculated the first-order histogram statistics in the next step.

Feature Extraction The feature extraction step consists of extracting four features independently: Haralick texture features, Gabor filter, LBP and first-order histogram statistics. The extracted features were stored in a numpy array, to be used as input of the classifier in the next step. We explore the features individually and combined to evaluate the combination that would yield the minimum detection error.

Haralick features, proposed by Haralick, Shanmughan and Dinstein (1973) [10], are a set of 28 textural features which can be extracted from each gray-tone spatial-dependence matrices. There are 14 features, and each one of these features are computed with one displacement vector in four directions (0° , 45° , 90° and 135°), then are calculated the mean and the range, averaged over these directions, comprising all the 28 set of features. The Haralick features are calculated using the *mahotas* package implementation for Python[5].

Gabor filter analyzes whether there are any specific frequency content in the image in specific orientations (theta). The images are filtered by a convolution with the real parts of a Gabor filter kernel, and the mean and variance of the filtered images are used as features for classification. The Gabor filter kernel is a Gaussian kernel modulated by a complex harmonic function which has the following parameters: a spatial frequency of the harmonic function, a theta orientation of analysis, a bandwidth captured by the filter, a standard deviation for x and y directions (*sigma_x* and *sigma_y*) of the Gaussian kernel, and a linear size of the kernel defined in *n_std*s standard deviations [24]. The method was computed by the *gabor_kernel* function implemented in *scikit-image* package [24]. In the tests, several combinations of frequency, bandwidth, theta orientation, *sigma_x* and *sigma_y*, and *n_std*s standard deviation were evaluated to find the parameters which minimize the detection error. The frequency and bandwidth ranged from 0.1 to 1 (steps of 0.1); the theta ranged from 0° to 360° (steps of 45°); *sigma_x* and *sigma_y* from 1 to 5 (steps of 1) and the standard deviation from 2 to 5 (steps of 1).

LBP is a method that computes the relationship between each pixel and their neighborhood. Each pixel is referenced and a pattern is computed over the relationship between the value of the referenced pixel and its neighbors (defined by a radius in pixels and a number of points to consider) [17]. This method was computed by the *lbp* function of the *mahotas* package [5]. In the tests, we ranged the radius (in pixels) and the number of points taken into consideration. The radius was ranged from 1 to 4 and the number of points to be taken into consideration varied from 8 to 16 (both in steps of 1).

Finally, the first-order histogram statistics were computed using the *iah2stat* function from Python collection *ia636*. This function computed eleven histogram statistics: mean (in grayscale value), variance (in grayscale values), skewness, kurtosis, entropy, mode (gray scale value with largest occurrence), percentile 1%, percentile 10%, percentile 50% (the median gray scale value), percentile 90% and percentile 99%.

Classification and Validation In this step, *numpy* arrays with the extracted features from the previous step were used as input to four classifiers. We tested the four classifiers (and their parameterizations) to verify which of them would have the minimum detection error solving our problem.

The parameterization of a classifier (also known as model selection) is a task that requires an empirical analysis through several tests to choose the parameters that minimize the detection error. There are different methods that help in this

choice, among them, the grid search (an exhaustive and brute force parameter search) and the randomized search (that implements a randomized search over parameters) [18]. We have chosen to do a brute force search, with parameter ranges based on scikit-learn’s documentation and references [18].

We tested the classifiers K-nearest neighbors (KNN), Support Vector Machine (SVM), Neural Network (NN) and the Gaussian Naive Bayes. The validation was made using the K-fold cross validation method. All methods were implemented using the scikit-learn package [18].

The KNN method find a predefined number of training samples closest in distance and predict the label from these [18]. The KNN classifier parameterization was made adjusting the number of neighbors, the weights and the distance metric. On the scikit-learn implementation, the best algorithm to compute the nearest neighbor is chosen automatically between ball tree, kd tree or brute force algorithms. The number of neighbors was chosen after tests ranging from 3 to 100. The weights were tested between uniform and distance (weight points by the inverse of their distance). Finally, the distance metric was tested with the Manhattan distance, Euclidean distance and Minkowski distance.

The SVM classifier represents the samples as points in space, mapped so that the samples of different categories are divided by a clear gap that is as wide as possible. The parameterization of this classifier was made with changes in the penalty parameter (from 0.1 to 5), gamma (from 0.0005 to 3), the kernel (poly, rbf, linear or sigmoid) and the independent term in kernel function (for poly and sigmoid function, ranging 0 to 1 in steps of 0.1). The stop tolerance was maintained in 0.0000005 and the class weight was selected as balanced.

The Neural Network used was a multilayer perceptron with the solver algorithm limited Broyden–Fletcher–Goldfarb–Shanno (lbfgs). This algorithm is indicated by scikit-learn documentation as faster and best performing for small datasets (without thousands of training samples or more) [18]. In the tests, the Neural Network was parameterized ranging the maximum number of iterations from 100 to 1000, the learning rate from 0.1 to 5, the penalty from 0.000002 to 0.05, and the hidden layers from 1 to 2, with the first layer ranging from 3 to 100 neurons and when using a second layer, it ranged from 2 to 60.

For the Gaussian Naive Bayes no parameterization was needed since there are no specific parameters for this method. The Gaussian Naive Bayes apply the Bayes’ theorem with the “naive” assumption of independence between every pair of features [18]. The algorithm assumes the likelihood of the features as Gaussian, and the parameters are estimated using maximum likelihood.

All classifiers were validated using the K-fold cross validation method. We used k equals to 10, what means a dataset divided in ten subsets and each subset used as a validation set in one of the ten evaluation processes, while the remaining subsets are used as a training set in the same process. This method results in ten evaluations with each one of the ten subsets being used as a validation set in some moment, and the final performance is measured by the average of the values computed in the evaluations.

The final classifiers parameterization will be presented in the next section.

2.3 Ethical Considerations

This work was conducted in accordance with resolution No. 466/12 of the Brazilian National Health Council. To preserve confidentiality, the images (including those shown in the paper) were separated from other patient’s data. No data presented herein allows patient identification. All the procedures were approved by the Ethics Committee for Research Involving Human Subjects of the Gonçalo Moniz Institute from the Oswaldo Cruz Foundation (CPqGM/FIOCRUZ), Protocols No. 188/09 and No. 1817574.

3 Results

The H&E and PAS stain techniques yield some different characteristics for the glomerulosclerosis problem. In H&E images only the increased glomerular extra-cellular matrix is highlighted (see Fig.2(a)), while the PAS images present the increased glomerular extra-cellular matrix with a strong reaction (stronger color) in the sclerosed region (see Fig. 2(b)). Because of such differences, we made one pipeline for each staining technique.

Except for the combination between the Haralick and first-order histogram features, no improvement was found in the combination of the other extraction methods presented with respect to the final detection accuracy. The other combinations either generated the same results or generated worse results than the other features used individually.

Table 1 shows the best results for H&E stained images in each classifier used. The Haralick features and the first-order histogram statistics are shown together because these features presented better results combined than individually.

Methods	KNN	SVM	NN	Naive Bayes
Haral.+Hist.	77.8%	76.8%	79.4%	73.0%
Gabor	83.8%	83.6%	84.8%	67.3%
LBP	79.6%	78.8%	82.6%	80.2%

Table 1. Detection results for H&E stained images

As shown in table 1, the best result for the samples stained in H&E was 84.8% using a neural network as classifier and the Gabor filter as feature extractor.

The Gabor filter used a frequency of 0.4 and a bandwidth of 0.8. The other parameters used the standard scikit-image settings.

The classifiers were used with the following parameters:

- SVM: penalty = 0.5, kernel = Radial basis function kernel (RBF), kernel coefficient (gamma) = 0.0025, tolerance for stopping criterion = 0.0000005 and class weight = balanced.
- KNN: 11 neighbors, weights = uniform and Manhattan distance.
- Neural Network: Multilayer perceptron with two hidden layers, the first with thirteen neurons and the second with six neurons. Activation function for the

hidden layer = rectified linear unit function (relu), solver = lbfgs, regularization term = 0.06, maximum number of iterations = 400, seed for random number generator = 2 and tolerance = 0.0000001.

- Naive Bayes: No parameters.

In the results using Haralick features and first-order histogram statistics have been chosen the mode, percentile 90% and percentile 99% from the first-order histogram statistics. The Haralick features were used together with the first-order histogram statistics and alone, but there were no improvements in the results. The classifiers that received these features as input were parameterized so:

- SVM: penalty = 0.5, kernel = Radial basis function kernel (RBF), kernel coefficient (gamma) = 0.0025, tolerance for stopping criterion = 0.0000005 and class weight = balanced.
- KNN: 9 neighbors, weights = uniform and Minkowski distance.
- Neural Network: Multilayer perceptron with two hidden layers, the first with five neurons and the second with six neurons. Activation function for the hidden layer = rectified linear unit function (relu), solver = lbfgs, regularization term = 0.06, maximum number of iterations = 400, seed for random number generator = 2 and tolerance = 0.0000001.
- Naive Bayes: No parameters.

The LBP method was set with radius in pixels = 1 and number of points to consider = 8 on mahotas package. The classifiers that received the features as input were parameterized so:

- SVM: penalty = 1, kernel = linear, kernel coefficient (gamma) = 0.0025 and class weight = balanced.
- KNN: 9 neighbors, weights = uniform and Manhattan distance.
- Neural Network: Multilayer perceptron with two hidden layers, the first with forty eight neurons and the second with four neurons. Activation function for the hidden layer = rectified linear unit function (relu), solver = lbfgs, regularization term = 0.06, maximum number of iterations = 400, seed for random number generator = 2 and tolerance = 0.0000001.
- Naive Bayes: No parameters.

Table 2 shows the best results of the analyzed features for the samples stained with PAS technique in each used classifier. The Haralick features and the first-order histogram statistics features are shown together because the combination of these features presented better results than they separated.

Methods	KNN	SVM	NN	Naive Bayes
Haral.+Hist.	77.2%	73.3%	81.3%	68.5%
Gabor	75.7%	70.8%	79.4%	73.7%
LBP	73.8%	65.9%	76.6%	74.6%

Table 2. Detection results for PAS stained samples

As shown in table 2, the best result for the samples stained in PAS was 81.3% using a neural network with Haralick features and first-order histogram statistics

as feature extractors. For the results of each method and classifier shown in the table, the following settings were used:

The Gabor filter was set with a frequency of 0.4 and a bandwidth of 0.8. The other parameters were used in the standard scikit-image settings. The classifiers that received the features as input were parameterized so:

- SVM: penalty = 0.5, kernel = Radial basis function kernel (RBF), kernel coefficient (gamma) = 0.0025, tolerance for stopping criterion = 0.0000005 and class weight = balanced.
- KNN: 27 neighbors, weights = uniform and Manhattan distance.
- Neural Network: Multilayer perceptron with two hidden layers, the first with twenty two neurons and the second with twenty one neurons. Activation function for the hidden layer = rectified linear unit function (relu), solver = lbfgs, regularization term = 0.06, maximum number of iterations = 400, seed for random number generator = 2 and tolerance = 0.0000001.
- Naive Bayes: No parameters.

In the Haralick features and first-order histogram statistics have been chosen the mean and percentile 50% from the first-order histogram statistics, and contrast, sum average and difference entropy from Haralick features. The classifiers that received these features as input were parameterized so:

- SVM: penalty = 0.5, kernel = Radial basis function kernel (RBF), kernel coefficient (gamma) = 0.0025, tolerance for stopping criterion = 0.0000005 and class weight = balanced.
- KNN: 23 neighbors, weights = uniform and Manhattan distance.
- Neural Network: Multilayer perceptron with two hidden layers, the first with twenty one neurons and the second with fifty one neurons. Activation function for the hidden layer = rectified linear unit function (relu), solver = lbfgs, regularization term = 0.06, maximum number of iterations = 400, seed for random number generator = 2 and tolerance = 0.0000001.
- Naive Bayes: No parameters.

The LBP method was set with radius in pixels = 1 and number of points to consider = 8 on mahotas package. The classifiers that received the features as input were parameterized so:

- SVM: penalty = 0.8, kernel = linear, kernel coefficient (gamma) = 0.0025 and class weight = balanced.
- KNN: 38 neighbors, weights = uniform and Manhattan distance.
- Neural Network: Multilayer perceptron with two hidden layers, the first with ten neurons and the second with one neuron. Activation function for the hidden layer = rectified linear unit function (relu), solver = lbfgs, regularization term = 0.06, maximum number of iterations = 400, seed for random number generator = 2 and tolerance = 0.0000001.
- Naive Bayes: No parameters.

At the end, the best results presented were chosen to be part of the PathoSpotter-K system: 84.8% for H&E stained samples and 81.3% for PAS stained samples. The accuracy of PathoSpotter-K is similar to equivalent works applied on different organs and diseases using the same methods, as shown in Table 3. However,

Pathospotter-K was the only dedicated work to detect glomerular lesions that was found.

Authors	Methods	Results	Problem	Image Dataset (stain technique)
Doyle et al.[7]	textural features	84%, 83% and 76%	Prostate cancer	100 with lowest, intermediate and highest levels (H&E)
Irshad, Roux and Racoceanu [11]	textural, statistical and morphological features	70%	Mitosis in Breast Cancer	326 (H&E)
Kothari et al.[14]	shape and textural features	77%	Renal cell carcinoma	103 (H&E)
Tashk et al.[22]	textural features	70.9% and 70.1%	Mitosis in Breast Cancer	326 (H&E)
Kim et al.[13]	textural features	75%, 100%, 66.6% and 72.7%	Renal cell carcinoma	8, 23, 13, 22 respectively (H&E)
McCarthy, O’Hurley and Cunningham [15]	textural and morphological features	77.5%	Prostate Carcinoma	20 (H&E)
Zheng et al.[26]	texture features and pLSA models	94.4%	Breast Cancer	600 from 5 categories (H&E)
Barker et al.[2]	shape and textural features	93.1%	Brain tumor	604 (H&E)
Sengar et al.[20]	textural features	81%	Colorectal cancer	165 (H&E)
Wan et al.[25]	textural and architectural features	92%, 77%, 76% and overall 69%	Breast Cancer	106; 24 low grade, 62 intermediate grade and 20 high grade (H&E)
Our method	textural features	84.8% and 81.3%	Glomerular lesions	499 (H&E) and 248 (PAS), respectively

Table 3. Comparison between similar works

4 Conclusion

In this paper, we presented the current developments in the PathoSpotter-K software to automatically detects segmental glomerular sclerosis lesions on H&E- and PAS-stained kidney sections.

We used a combination of classical image processing and pattern recognition methods to create a pattern detection system that is part of a wider system in development to be used by students and pathologists to learn and research about glomerular lesions. The current results achieves an accuracy of 84.8% for H&E and 81.3% for PAS stained samples. Such performance is equivalent to other similar computer-aided diagnosis systems in pathology.

As future works, we intend to improve the accuracy with additional classification algorithms and to increase the number of samples in the dataset to improve the robustness with more training images. A web service of the PathoSpotter-K software is been developed to be tested and used by pathologists and computer scientists that may help to improve the system.

Acknowledgment

The PathoSpotter project is partially funded by the Bahia State Research Founding Agency (FAPESB), grant No. TO P0008/15. Ikaro Campos received a scholarship from FAPESB. The image dataset used in this work was built with support from the Supporting Program for the Diagnosis of Hepatic and Renal Diseases, founded by the Gonçalo Moniz Institute, from the Oswaldo Cruz Foundation, Ministry of Health, Brazil.

References

1. ARIAS, L.F., JIMENEZ, C.A., ARROYAVE, M.J.: Variantes histológicas da glomeruloesclerose segmentar e focal primária: casuística e evolução clínica. *Brazilian Journal of Nephrology* 35(2), 112–119 (2013)
2. Barker, J., Hoogi, A., Depeursinge, A., Rubin, D.: Automated classification of brain tumor type in whole-slide digital pathology images using local representative tiles. *Medical Image Analysis* 30, 60–71 (2016)
3. Barros, G.O., Navarro, B., Duarte, A., dos Santos, W.L.C.: Pathospotter-k: A computational tool for the automatic identification of glomerular lesions in histological images of kidneys. *Scientific Reports* 7, 46769 (4 2017), <http://dx.doi.org/10.1038/srep46769>
4. Belsare, A.D., Mushrif, M.M.: Histopathological image analysis using image processing techniques: an overview. *Signal & Image Processing : An International Journal (SIPIJ)* 3(4) (2012)
5. Coelho, L.: Mahotas: Open source software for scriptable computer vision. *Journal of Open Research Software* 1(1) (2013)
6. D’Agati, V., Fogo, A., Bruijn, J., Jennette, J.: Pathologic classification of focal segmental glomerulosclerosis: a working proposal. *American Journal of Kidney Disease* 43(2), 368–382 (2004)
7. Doyle, S., Feldman, M., Tomaszewski, J., Madabhushi, A.: A boosted bayesian multiresolution classifier for prostate cancer detection from digitized needle biopsies. *IEEE Transactions on Biomedical Engineering* 59(5) (2012)
8. Gonzalez, R., Woods, R.: *Digital Image Processing*. Pearson (2007)
9. Gurcan, M.N., Boucheron, L.E., Can, A., Madabhushi, A., Rajpoot, N.M., Yener, B.: Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering* 2 (2009)
10. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics SMC-3*(6), 610–621 (1973)
11. Irshad, H., Roux, L., Racoceanu, D.: Multi-channels statistical and morphological features based mitosis detection in breast cancer histopathology. *Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society* pp. 6091–6094 (2013)

12. Kato, T., Relator, R., Ngouv, H., Hirohashi, Y., Takaki, O., Kakimoto, T., Okada, K.: Segmental hog: new descriptor for glomerulus detection in kidney microscopy image. *BMC Bioinformatics* 16(316) (2015)
13. Kim, T., Cho, N., Jeong, G., Bengtsson, E., Choi, H.: 3d texture analysis in renal cell carcinoma tissue image grading. *Computational and Mathematical Methods in Medicine* (2014)
14. Kothari, S., Phan, J.H., Young, A.N., D., W.M.: Histological image classification using biologically interpretable shape-based features. *BMC Medical Imaging* 2013 13(9), 1–16 (2013)
15. McCarthy, N., Cunningham, P., OHurley, G.: The contribution of morphological features in the classification of prostate carcinoma in digital pathology images. *22nd International Conference on Pattern Recognition* (2014)
16. Mosquera-Lopez, C., Agaian, S., Velez-Hoyos, A., Thompson, I.: Computer-aided prostate cancer diagnosis from digitized histopathology: A review on texture-based systems. *IEEE Reviews in Biomedical Engineering* 8, 98–113 (2015)
17. Ojala, T., Pietikainen, M., Maenpaa, T.: Gray scale and rotation invariant texture classification with local binary patterns. *Lecture Notes in Computer Science* (Springer) pp. 404–420 (2000)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, 2825–2830 (2011)
19. Queiroz, M.M., Júnior, G.B.S., do Socorro R. Lopes, M., Nogueira, J.O.L., Correia, J.W., Jerônimo, A.L.C.: Estudo das doenças glomerulares em pacientes internados no hospital geral césar cals - fortaleza, ceará, brasil. *Brazilian Journal of Nephrology* 31(1), 6–9 (2009)
20. Sengar, N., Mishra, N., Dutta, M.K., Prinosil, J., Burget, R.: Grading of colorectal cancer using histology images. *39th International Conference on Telecommunications and Signal Processing (TSP)* (2016)
21. Sesso, R.C., Lopes, A.A., Thomé, F.S., Lugon, J.R., Martins, C.T.: Inquérito brasileiro de diálise crônica 2014. *Jornal Brasileiro de Nefrologia (JBN)* 38(1), 54–61 (2016)
22. Tashk, A., Helfroush, M.S., Danyali, H., Akbarzadeh, M.: An automatic mitosis detection method for breast cancer histopathology slide images based on objective and pixel-wise textural features classification. *5th Conference on Information and Knowledge Technology* (2013)
23. Veta, M., Pluim, J., Van, D.P., Viergever, M.: Breast cancer histopathology image analysis: A review. *IEEE Transactions on Biomedical Engineering* 61(5), 1400–1411 (2014)
24. van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Guillard, E., Yu, T., the scikit-image contributors: scikit-image: image processing in Python. *PeerJ* 2, e453 (6 2014), <http://dx.doi.org/10.7717/peerj.453>
25. Wan, T., Cao, J., Chen, J., Qin, Z.: Automated grading of breast cancer histopathology using cascaded ensemble with combination of multi-level image features. *Neurocomputing* 229, 34–44 (2016)
26. Zheng, Y., Jiang, Z., Shi, J., Ma, Y.: Retrieval of pathology image for breast cancer using pls model based on texture and pathological features. *IEEE International Conference on Image Processing (ICIP)* (2014)