Generating density maps for convolutional neural network-based cell counting in specular microscopy images

Journal of Physics: Conference Series **1547** (2020) 012019 doi:10.1088/1742-6596/1547/1/012019

Generating density maps for convolutional neural network-based cell counting in specular microscopy images

J S Sierra[1], J Pineda[1], E Viteri[2,3,4], A Tello[2,3,4], M S Millán[5], V Galvis[2,3,4], L A Romero[6], and A G Marrugo[1]

[1] Facultad de Ingeniería, Universidad Tecnológica de Bolívar, Cartagena de Indias, Colombia
[2] Centro Oftalmológico Virgilio Galvis, Floridablanca, Colombia
[3] Fundación Oftalmológica de Santander Clinica Carlos Ardila Lulle, Floridablanca, Colombia
[4] Facultad de Salud, Universidad Autónoma de Bucaramanga, Bucaramanga, Colombia
[5] Departamento de Óptica y Optometría, Universidad Politécnica de Cataluña, Terrassa, España
[6] Facultad de Ciencias Básicas, Universidad Tecnológica de Bolívar, Cartagena de Indias, Colombia

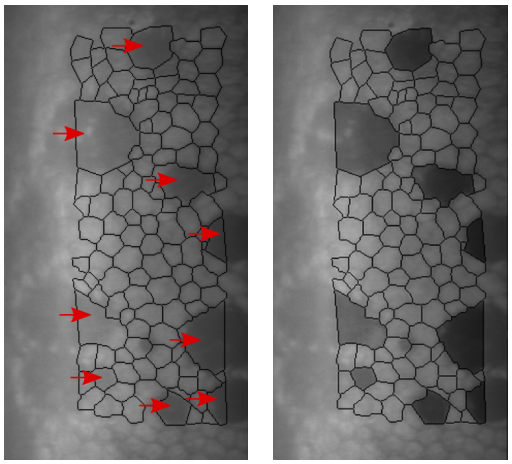
E-mail: juansierrabravo@outlook.com

Abstract. Accurate endothelial cell density with specular microscopy is essential for correct clinical assessment of the cornea. Commercial specular microscopes incorporate automated cell segmentation methods to estimate cell density. However, these methods are prone to false cell detections in pathological corneas. This project aims to obtain a reliable automated cell density from specular microscopy images of both healthy and pathological corneas with convolutional neural networks. Convolutional neural networks require labeled datasets. Thus, we developed custom software for producing a curated dataset of labeled ground-truth images and cell density maps. In this paper, we implemented a fully convolutional regression network to predict the cell density map from the input microscopy image. Encouraging preliminary results show the potential of the method. This approach may pave the way for dealing with the variability of corneal endothelial cell images.

1. Introduction

Automated cell density (CD) estimation in in-vivo specular microscopy images is a challenging task [1], especially in situations where conventional single-cell segmentation methods fail due to pathological conditions that affect the cornea, like in the case of cornea guttata [2]. The specular microscope is used to assess the corneal endothelial cells (CEC) of a patient's cornea. The microscope software automatically obtains the cell segmentation, the cell numbers, and the cell sizes. Often, inaccurate segmentations are obtained, especially in pathological corneas [3], as the one shown in Figure 1(a). The automatic segmentation can usually be modified with editing tools in the microscope, for instance, to draw or remove cells. However, removing erroneous detections may lead to over- or under-estimation of cell parameters, such as CD, cell size, among others.

The goal of this project is to obtain reliable automated CD and morphological values from specular microscopy images of both healthy and pathological corneas with a supervised learning algorithm [4]. These algorithms require labeled datasets that are difficult to obtain in these scenarios. For this reason, we developed custom software that allows the examiner to manually modify the endothelial cell segmentation of images obtained with a Topcon SP-3000P specular microscope to obtain a curated database of images. In this paper, we show preliminary results of the ground-truth dataset generation and a prototype implementation using a fully convolutional regression network (FCRN) [5] with CEC images from porcine eyes [6].



(a) (b)

Figure 1. (a) Original automatic segmentation of a pathological cornea obtained from the specular microscope software. The red arrows indicate regions of inaccurate cell segmentation. (b) The ophthalmologist selected the pathological regions shown in black with the developed software.

1.1. Background

The problem of automated CD estimation in specular microscopy has been studied using different methods. For instance, Ruggeri, *et al.*, [1] proposed to use a 2D discrete Fourier transform (DFT) technique to obtain the spatial frequency of CEC images to obtain an estimation of the CD. Alternatively, other works are based on cell segmentation and morphology analysis [2, 7–12]. Several authors analyze both cell morphology and guttae morphology in corneas with pathologies [2,12]. These methods must detect precisely the cell contour to generate a proper segmentation. However, they can fail in the pathological corneas, where the images typically have low quality.

More recently, several authors have begun using convolutional neural networks (CNNs) for the cell counting task. In reference [12], the authors assess the performance of U-Net (a neural network for biomedical image segmentation) in healthy and pathological CEC images with different image qualities. In references [5,13], the authors used two fully convolutional regression networks (FCRNs) to regress a cell spatial density map in microscopy images, where the integral on a region of interest (ROI) results in the number of cells in that region. In other words, the cell counting problem is cast as a supervised learning problem that tries to learn a mapping between an image $I(x)$ and a density map $D(x)$, denoted as $F : I(x) \rightarrow D(x)(I \in R^{m\times n}, D \in R^{m\times n})$ for a $m \times n$ pixel image. This approach has been used for counting objects [14], such as cells in microscopy images [5] or people in surveillance videos of crowded scenes [15]. Here we adopt the same approach.

2. Method

The objective is to create a curated database of CEC images for developing an automated cell counting system based on CNNs. The considered approach [5] requires the generation of cell density maps from ground-truth data. We used a Topcon SP-3000P specular microscope. Its

software exports a two-channel tagged image file format (TIFF) file that contains the CEC image and the calculated segmentation. An expert ophthalmologist used the developed software called segmentation corrector to import the TIFF file and make the necessary corrections to the segmentation to obtain real data for a correct assessment of the patient's cornea. In the following, we explain the segmentation correction stage and the generation of cell density maps.

2.1. Segmentation correction

We implemented our software using the Python-based Tkinter library to design the GUI shown in Figure 2. This software features several tools to modify the segmentation manually. When the segmentation is modified, the user can see the recalculated parameters and save the new segmentation.

The two-channel TIFF file contains the CEC image, shown at the right side of the GUI, and the automatic segmentation from the specular microscope overlayed on top of the CEC image. This feature allows for easy visualization of necessary modifications in the segmentation. The software contains four tools: to draw borders to include important information in the segmentation, to delete the border between two regions, and also to select a region that corresponds to guttae or cell. The tools were implemented using morphological operations. After each modification, the software recalculates areas, number of cells, number of guttae, cell density, guttae density, and area ratio for cells and guttae. The new segmentation can be saved as a new TIFF image of three channels. In Figure 1(a), we show an example of an image before being processed. The red arrows indicate regions of inaccurate cell segmentation. In Figure 1(b), several regions were marked as guttae.

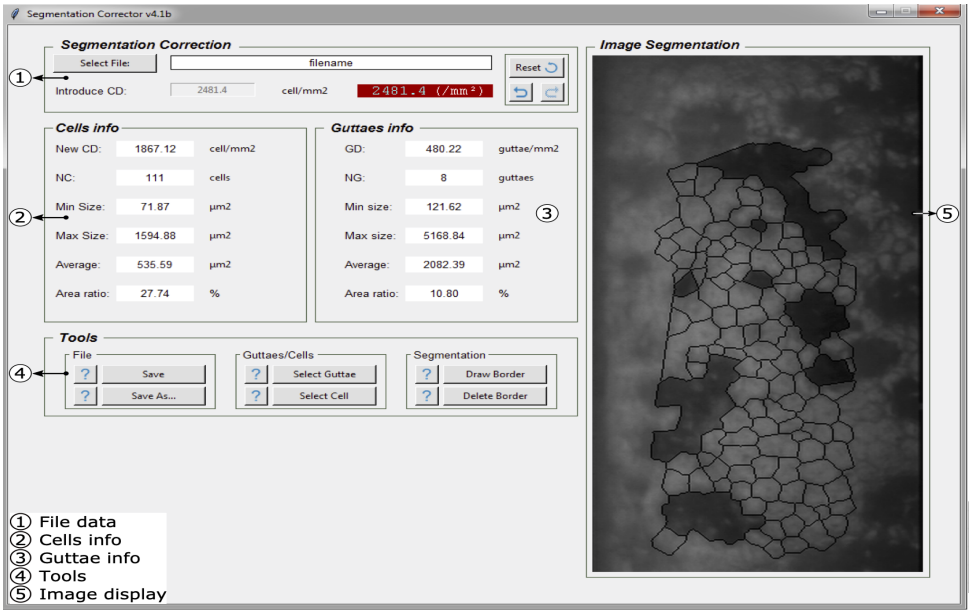


Figure 2. Graphical user interface (GUI) of the segmentation corrector software.

2.2. Density map generation

The training and testing density maps images were generated using the corrected segmentations and the corresponding CEC images. For this task, the segmentation must be a binary image, and each cell is processed individually, as shown in Figure 3(a). The cell orientation is obtained from the region bounding box, as shown in Figure 3(b). The probability density function is created from a Gaussian function that matched the analyzed cell normalized to sum to 1, as shown in Figure 3(c). The Gaussian function is given by Equation (1).

$$f(x,y) = A\exp\{-(a(x-x_o)^2 + 2b(x-x_o)(y-y_o) + c(y-y_o)^2))\} \ ,$$
$$a = \frac{cos^2\theta}{2\sigma_X^2} + \frac{sin^2\theta}{2\sigma_Y^2} \ , \ b = -\frac{sin2\theta}{4\sigma_X^2} + \frac{sin2\theta}{4\sigma_Y^2} \ , \ c = \frac{sin^2\theta}{2\sigma_X^2} + \frac{cos^2\theta}{2\sigma_Y^2} \ , \quad (1)$$

where A is a normalization constant, x and y are the coordinates of each pixel, x_o and y_o are the origin coordinates, θ is the Gaussian orientation, and σ_X and σ_Y are the dimensions of the Gaussian function.

(a) (b) (c)

Figure 3. (a) A binary region corresponding to a cell. (b) Orientation obtained from the bounding box. (c) the Gaussian probability density function used to represent the cell in a cell density map.

3. Experiments and results

3.1. Image corpus and data curation

The software segmentation corrector was used by two expert ophthalmologists to obtain real data from images of corneas with pathologies. In Figure 1 is shown an example of segmentation correction, the calculated data before and after the image been corrected are shown in Table 1, where we can see there was an overestimation of initial cell density and number of cells. The software also shows additional information, like the number of guttae, guttae size and guttae density, which can be used by the ophthalmologist to value the health of the patient's cornea.

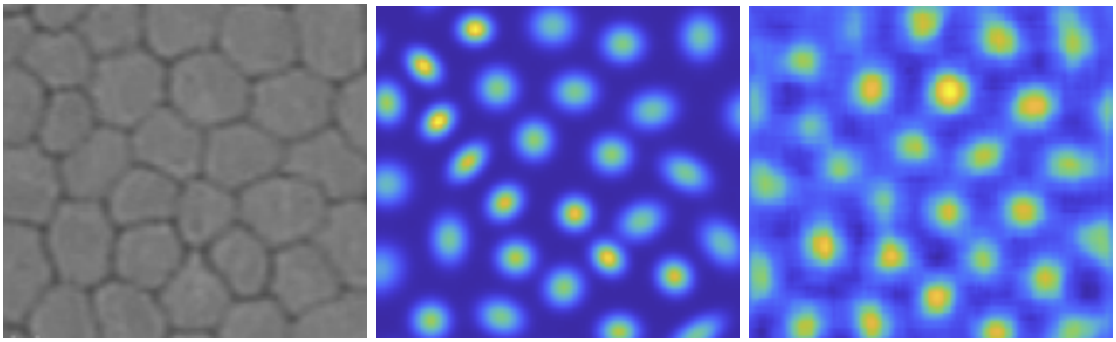
Table 1. Calculated data from the segmentation before and after been corrected.

	Original	Corrected
Number of cells	119.0	110.00
Average cell size (μm)	535.9	579.72
Cell density (cells/mm^2)	1866.1	1724.97
Number of guttae	-	9.00
Average guttae size (μm)	-	1793.66
Guttae density (guttae/mm^2)	-	141.13

3.2. Fully convolutional regression networks implementation

To validate if the method is efficient in corneal endothelial cell images, we implemented a FCRN architecture based on one of the proposed in reference [5]. We have implemented this neural network using the Python-based Keras library with a Tensor flow backend [16]. We used the

dataset from reference [6] to generate the training and testing images. It contains CEC images acquired from porcine eyes. Figure 4 shows one of the results obtained by the implementation of the FCRNs. Figure 4(a) is the CEC image; Figure 4(b) shows the ground-truth density map, where the integral over the image is 29.9979; and Figure 4(c) is the predicted density map using the FCRN, where we obtained a result of 30.4864. A negligible difference between predicted and ground-truth data of 0.4885 confirms the validity of the proposed approach. Nevertheless, further validation is needed.



(a) (b) (c)

Figure 4. FCRNs implementation. (a) Corneal endothelial cell image. (b) Density map (ground truth), integral: 29.9979. (c) Predicted density map by the FCRNs, integral: 30.4864

4. Conclusions

Reliable endothelial cell counting is difficult due to the inherent variability in corneas, both healthy and pathological. Machine learning methods, like CNNs, provide a means to learn from annotated data and output accurate results. To this end, we implemented custom software to annotate corneal endothelial cell images to train a FCRN. The preliminary results show that the proposed method may yield accurate results without requiring cell segmentation, which is especially challenging in pathological corneas. The software annotation also produced additional clinical data like guttae size and density that could be used in future clinical studies.

Acknowledgment

This work has been partly funded by the Centre de Cooperació per al Desenvolupament (CCD) at the Universitat Politècnica de Catalunya under project ref. CCD 2019-B004. Juan S. Sierra thanks Universidad Tecnológica de Bolívar for a Masters degree scholarship.

References

[1] Ruggeri A, Grisan E and Jaroszewski J 2005 *British Journal of Ophthalmology* **89(3)** 306
[2] Giasson C J, Graham A, Blouin J F, Solomon L, Gresset J, Melillo M and Polse K A 2005 *Eye & Contact Lens* **31(4)** 158
[3] Maruoka S, Nakakura S, Matsuo N, Yoshitomi K, Katakami C, Tabuchi H, Chikama T and Kiuchi Y 2017 *International Ophthalmology* **35** 1
[4] Russell S J and Norvig P 2016 *Artificial Intelligence: A Modern Approach* (Malaysia: Pearson Education Limited)
[5] Xie W, Noble J A and Zisserman A 2018 *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* **6(3)** 283
[6] Ruggeri A, Scarpa F, De Luca M, Meltendorf C and Schroeter J 2010 *British Journal of Ophthalmology* **94(5)** 643
[7] Al-Fahdawi S, Qahwaji R, Al-Waisy A S, Ipson S, Ferdousi M, Malik R A and Brahma A 2018 *Computer Methods and Programs in Biomedicine* **160** 11

[8] Scarpa F and Ruggeri A 2016 Automated morphometric description of human corneal endothelium from in-vivo specular and confocal microscopy *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Orlando: IEEE) pp 1296–1299
[9] Scarpa F and Ruggeri A 2016 *Cornea* **35(9)** 1222
[10] Piorkowski A, Nurzynska K, Boldak C, Reska D and Gronkowska-Serafin J 2015 *Journal of Medical Informatics & Technologies* **24** 155
[11] Piorkowski A, Nurzynska K, Gronkowska-Serafin J, Selig B, Boldak C and Reska D 2017 *Computerized Medical Imaging and Graphics* **55** 13
[12] Daniel M C, Atzrodt L, Bucher F, Wacker K, Böhringer S, Reinhard T and Böhringer D 2019 *Scientific Reports* **(1)** 4752
[13] Weidi X, Noble J A and Zisserman A 2015 Microscopy cell counting with fully convolutional regression networks *1st Deep Learning Workshop, Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (Munich: The Medical Image Computing and Computer Assisted Intervention Society)
[14] Lempitsky V and Zisserman A 2010 Learning to count objects in images *Advances in Neural Information Processing Systems 23th International Conference on Neural Information Processing Systems (NIPS)* (Canada: Neural Information Processing Systems Foundation Inc.) pp 1324–1332
[15] Kang D, Ma Z and Chan A B 2018 *IEEE Transactions on Circuits and Systems for Video Technology* **29(5)** 1408
[16] Chollet F *et al.* 2015 *Keras: Machine Learning in Python* (San Francisco: GitHub Inc.)