HAL
open science

Computationally Efficient Blood Vessels Segmentation in Fundus Image on Shared Memory Parallel Machines

Yaroub Elloumi, Mohamed Akil

To cite this version:

Yaroub Elloumi, Mohamed Akil. Computationally Efficient Blood Vessels Segmentation in Fundus Image on Shared Memory Parallel Machines. SPIE Real-Time Image and Video Processing, Apr 2018, Orlando, United States. hal-01796765

HAL Id: hal-01796765

https://hal.science/hal-01796765

Submitted on 22 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computationally Efficient Blood Vessels Segmentation in Fundus Image on Shared Memory Parallel Machines

YaroubElloumi[a,b], Mohamed Akil[a]

[a]GaspardMonge Computer Science Laboratory, ESIEE-Paris, University Paris-Est Marne-la-Vallée, France.
[b]Medical Technology and Image Processing Laboratory, Faculty of medicine, University of Monastir, Tunisia.

Abstract:

Blood vessels segmentation in fundus image is a requiring step in order to detect retinopathies. A higher performing segmentation was been proposed in [12]. It consists at three dependent stages: Provide two binary images to extract wide vessels, compute features of the remaining pixels on binary images in order to extract fine vessels, and then combine both wide and fine vessels. The segmentation execution time is about 3-12 seconds when it is performed with fundus image having resolutions between 768*584 and 999*960. These latest resolutions are quite smaller than ones provided by actual retinograph, which leads to a higher rise on execution time.

In this paper, we propose a parallelism strategy of the segmentation approach for implementation in Shared Memory Parallel Machine (SMPM). First, both binary images are provided in parallel. Thereafter, features processing is split according to their computational complexities. At the later stage, wide vessels and fine vessels images are subdivided adequately in the objective of a parallel combination.

The parallel strategy is implemented using OpenCV and then assessed on STARE public data sets. Experimental analyses of execution time and efficiency are presented and discussed.

Keywords: Fundus image, blood vessels, parallelism, Shared Memory Parallel Machine, OpenCV.

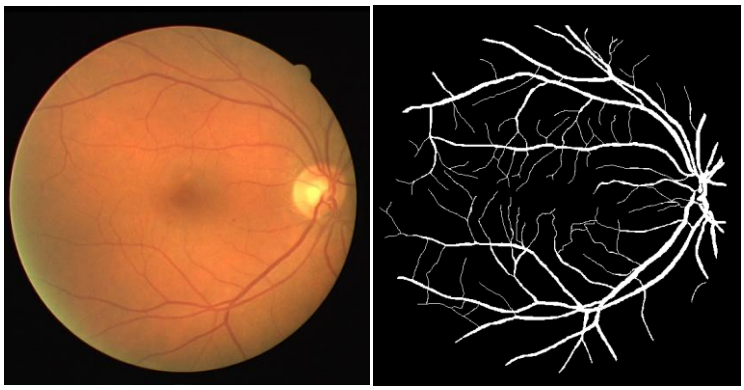
Introduction

Fundus images have been extensively used to detect abnormal eye conditions such as diabetic retinopathy. For example, in [1], Kar et al. provided an approach to detect neovascularization and in [2], Tan et al. presented an approach for detecting lesions that lead to diabetic retinopathy. Other examples include the work described in [3] which addressed the Age-related Macular Degeneration (AMD) condition by examining the anatomy of fundus images and the work in [4] by Prakash et al. extracting the macula in order to detect diabetic. Also, the work in [5] discussed an approach for detecting glaucoma.

In several works, blood vessels segmentation represents a crucial step to detecting pathology from fundus images. On the one hand, some pathologic level leads to the formation of new vessels such as the wet AMD of the macula and the neovascularization of the optic disk in the case of DR. Hence, the blood vessels segmentation in a specific anatomic component leads to directly detect the pathology. On the other hand, some lesions have similar characteristics in terms of brightness, color and shape than blood vessels. Thus, vessel graph segmentation leads to identify lesions and hence pathology detection. Other works proceed to locate anatomical component of retina with correspondence to the vessels. The work described in [15] leads to detect optic disk basing on the convergence of major vessels within, such as shown in fig.1. The approach proposed in [16] describes the way to detect the macula based on the elliptical shape formed by the major vessel grid and the optic disk location.

Several works are focused on extracting blood vessels in fundus images, which can be categorized on non-supervised and supervised approach. In the first category, the work proposed in [7] presents a pipeline processing based on morphological operator which aims to extract separately the major and thin vessels respectively, where execution time is about 5 seconds. Roychowdhury et al [10] proposed a non-supervised method to extract the major vessels and hence

exploring iteratively the fundus image to extract thin vessel with decreased width. Elaheh et al. [14] leads to extract the bloods vessels and the lesions separately. Therefore, a third steps consists of merging results in order to drop the confused pixels. The work described on [13] aims to apply a thresholding to detect major vessels and the Gabor Wavelet (GW) to extract the thin ones, and hence combining their results.



(a) (b)

Figure 1. (a) Fundus image; (b) the segmented blood vessels

For the second category, several supervised approaches were proposed with different classifiers. The work described in [5]and [11] classify pixels using C-mean and fuzzy processing, respectively. Aslani et al. propose to use the radon forest using 17 features [6], where testing time is about 1 minute. Wang et al. [8] aim to extract vessels by combining Convolutional Neural Network and Random Forest classifiers. The work described in [9] employs an Extreme Learning Machine (ELM) with 39 features, which is evaluated using DRIVE and an “RIS” local database.

We deduce that supervised approaches are more performing in terms of segmentation quality compared to the non-supervised one. However, they are characterized by raised execution times. In this context, we distinguish a segmentation approach proposed in [12] which combines morphological operator and a computational efficiency classifier. The vessels are segmented on two steps for thick and thin vessels, respectively. First, the major vessels are extracted by generating and intersecting two binary images. This step allows reducing about 90% of time complexity compared to the classification the same pixels. The second step consists of classifying the thin vessels using the Gaussian Mixture Model (GMM) with optimal 8-feature set. The proposed algorithm achieves blood vessel segmentation with a higher accuracy in average execution times between 3-12 seconds on three public databases.

However, fundus image resolution increases continuously, which exceeds widely the public database image resolutions. Therefore, the execution time still increases similarly. Moreover, the medical context imposes execution time constraint when employing automatic detection of pathologies or retinal components, whose are based on vessel extraction. Consequently, it leads to imposing computational constraint to the vessel segmentation.

In this paper, we propose a parallelism strategy of the segmentation approach proposed in [12] for implementation in Shared Memory Parallel Machine. Both binary images are scheduled in parallel. Thereafter, features processing and classification are computed in concurrency with respect to their computational complexities. The rest of the paper is organized as follows. In Section II, we describe the blood vessel segmentation approach. In section III, we propose our parallelism strategy. The experimental results are presented in Section IV, followed by the conclusion in Section V.

Blood Vessel Segmentation approach

The green component ofthe fundus image is selected. Then,a fundus mask is applied in order to extract the retina and a contrast adjustmentis processed in order to enhance vessels where result is untitled Ie.Alow-pass median filter with window size [20x20]is applied and then subtracted from the Ie which result is saved in H. A second step consists of inversing the Ie image. Next, a Top Hat reconstruction is applied with 12 linear structures having 15 pixel lengths with a difference angles of 15° steps. Then, all highest intensity pixels are selected to provide the image T.Both H and T images are thresholdedwith value equal 0.2 and intersected in order to provide the major vessels in image V.

The remaining pixels in each H and T images, which are not selected in V, are saved in two binary images H1 and T1, respectively, and then combined in image C.The pixels inC are classified as vessel or non-vessel using a Gaussian Mixture Model (GMM) whichis trained using theDRIVE DB images.First, each pixel '(x, y)' in C is placed in the center of a (9*9) square window Ws, where the square size is discussed in [12]. Then, the first 4 features correspond respectively to the mean, standard deviation, maximumpixel intensity and minimum pixel intensity among all pixels extracted in the Ws.The fifth new feature (f5) is similar to the number of pixels which do not correspond to bright regions. The f6 and f7 features are extracted from the square windowed-image from the Top Hat reconstructed imageT, which correspond to the second-ordermoments invariant-based features. The f8 feature is similar to the pixel intensity Ie(x, y). The classified pixels correspond to the fine vessels, which are saved in V'. Thereafter, V and V' pixels are combined to provide the final segmented blood vessels in Vf.The retained vessels are those having surface greater than a parameter 'a', where different values are proposed for each database. The whole processing pipeline is modeled in fig. 2.

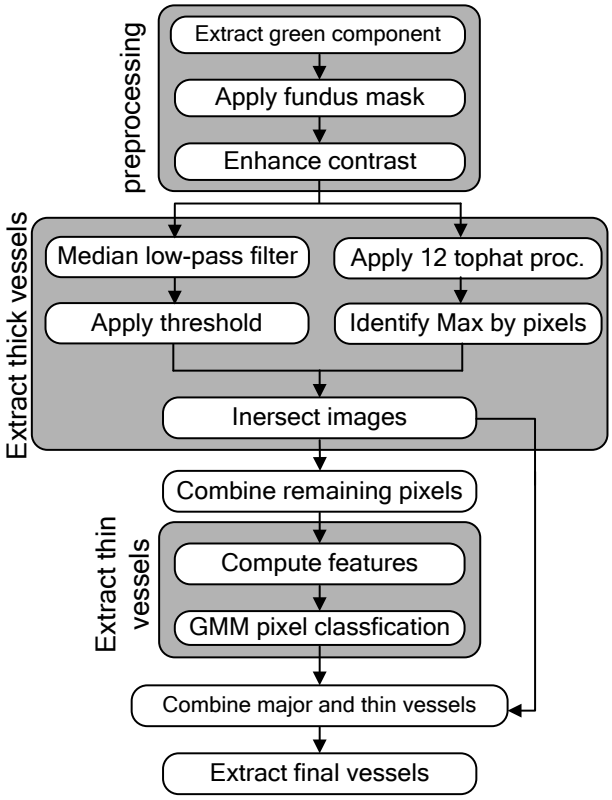


Figure 2.Processing pipeline of the blood vessel segmentation approach [12]

Parallel strategy of Blood Vessel Segmentation

Our parallelism strategy consists at implementing the steps that requires important execution times on the GPU architecture. The choice of architecture is motivated by the repetitive aspect of the segmentation processing, which requires a SIMD programming on shared memory. We proceed to evaluate each processing step in terms of execution time and its performance on the GPU implementation.

The execution time of the preprocessing block is too little that CPU-GPU communication requires more than computational execution. We proceed to compare the CPU and the GPU execution times of each step modeled in fig.2. The experiment is performed using STARE and DRIVE database images. We deduced that each step which corresponds adding, subtracting or multiplying per-element matrix, is adequate to be executed on CPU rather than GPU, such as "apply threshold", "intersect images", "combine remaining pixels", "combine major and thin vessels" and "extract final vessels".

The median low-pass filter leads to a separate processing for each pixel p, which consists at extract a sub-window SW, where p pixel is centered, and multiply the sub-window to existing matrix M, where SW and M have the same size. Likewise, the Top Hat reconstruction consists at extracting a sub-window and then multiplying it 12 times with a separate morphologic structure. Similarly, each pixel feature consists at computing the criterion based on (9*9) sub-window. Thereafter, the GMM processing leads to combining the 8 features in order to classify the corresponding pixel. Such processing are characterized by a higher computational complexity and thereby, we proceed to implement then on GPU architecture, as illustrated in fig. 3.

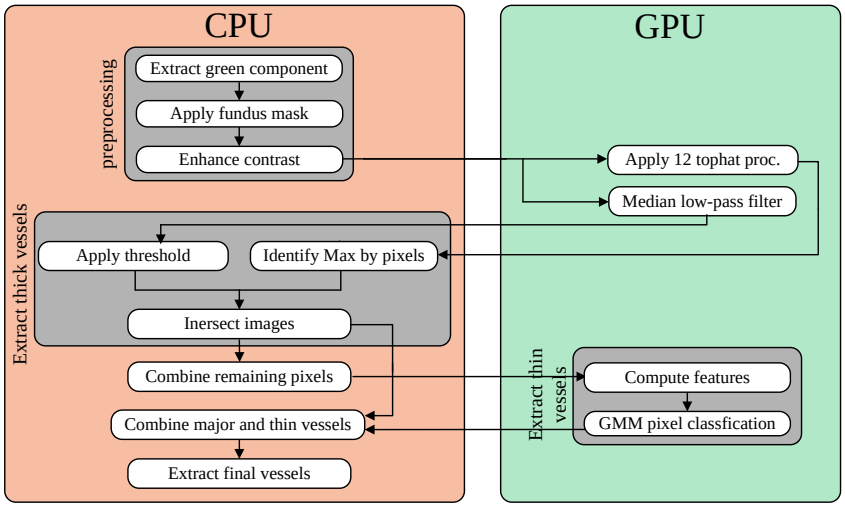


Figure 3. CPU/GPU scheduling of the blood vessel segmentation steps

The median low-pass and the 12 Top Hat reconstructions are called for each pixel of input image Ie. Therefore, the whole corresponding processing of one pixel is affected in the same thread, on order to access to the input image once per pixel. Each thread output parameteris affected respectively to the Top Hat reconstructions and the median matrices, as described in figure 4. The parallelism strategy follows the same principle to computing features, where f1to f5 features are defined jointly since the explore data from the same C image, while f6 and f7 are based on maximal matrix of Top Hat reconstructions.

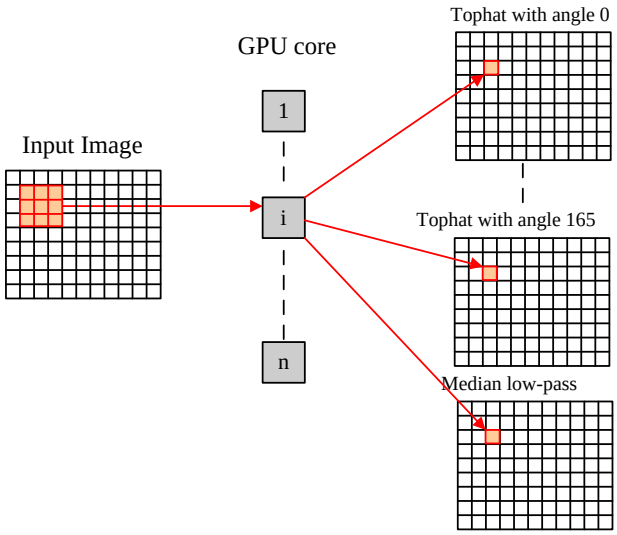


Figure 4. GPU scheduling of Top Hat and median low-pass filter processing

Implementation environment

The blood vessels segmentation was evaluated via a desktop implementation having a Quad-core i7-4790 CPU cadenced by 3.6GHz and equipped by 8 GB of RAM. The optimized code is processed in the “GeForce GTX 950" GPU architecture described in Table.1.

Table.1. GPU main features

Features	Characteristics
CUDA Cores	768
Multiprocessors	6
GPU Max Clock rate	1.25 GHz
Global memory	1988 MB
Registers available per block	65536 B
Constant memory	65536 B
Shared memory per block	49152 B

The processing was implemented using Open Computer Vision(OpenCV) which is an open source library for computer vision and machine learning written natively in C++ [2][3]. It is one of the most employed libraries by image and video application implemented on mobile devices. OpenCV provides more than 2500 optimized algorithms which are able be employed in camera movements tracking augmented reality, etc. Most algorithms are suitable for real-time implementation where performance improvement was demonstrated [4]. Therefore, the blood vessel segmentation is to be developed with mixed-programming involves OpenCV and C++.

The contrast enhancement was processed using CLAHE algorithm in order to insure a balanced brightness in the whole retina. Thus, the CLAHE is configured with a sub-window (12*12) and a cliplimit equal to 1.5 (createCLAHE (clipLimit = 1.5, tileGridSize = (12, 12))). The segmentation approach consists at performing the Top Hat reconstructions in 12 angles using a 15 pixel line. Thus, a linear structure is created (STR_Line_angle0 = numpy.ones ((15, 1), numpy.uint8)). Hence, the 11 other linear structures are provided by rotating the first one. The rotation processing requires identifying the structure center P, the rotation angle “ang” and a rotation function ROT, as described in the following code:

```
P = Point( 8, 8);
ROT = getRotationMatrix2D( P, ang, 1);
warpAffine(STR_Line_angle0, STR_Line_angle15, ROT, (15*15)  );
```

Thereafter, the Top Hat reconstruction is processed using the provided kernel (cv.morphologyEx (img, cv.MORPH_TOPHAT, STR_Line_angle0) ;). The segmentation approach requires selecting the maximal intensity of each pixel from all tophat matrices. Thus, a pyramidal maximization processing is done using max function between both images (max (tophat_0, tophat_15, result) ;). To classifier pixels using GMM, an EM object is created and cluster number is affected. Hence, the training is processed using the fundus image (FI) and the corresponding segmented blood vessels image (SB) for each DRIVE database image (train (FI, SB) ;). Thereafter, the test step is done for each image T separately (predict (T) ;).

Experimental Results

Our experimentation consists at evaluating the optimized implementation execution time compared to the sequential one. For this purpose, we select randomly 8 images from the STARE databases. We aim to evaluate the execution time improvement for each processing in run in GPU and the whole execution time. Therefore, figure 5 illustrates the execution time in sequential and GPU implementation in “top-hat& Low-pass Filter”, “Features processing” and “GMM classification”, where we deduce speedup values equals to 6.025, 5.826 and 6.165 for each processing respectively.

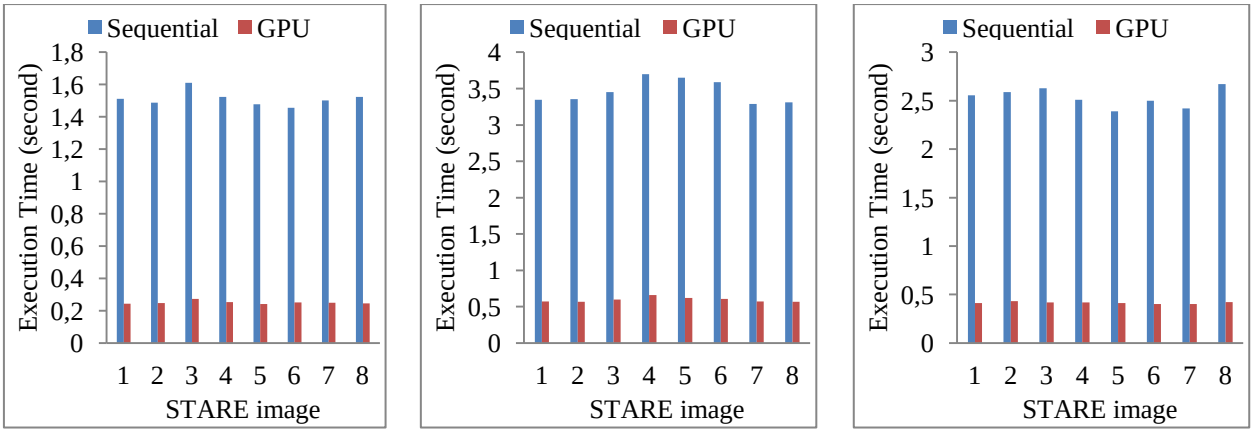


Figure 5. Execution time of sequential and optimized implementations in terms of STARE database images: (a)top-hat& Low-pass Filter; (b) Features processing; (c) GMM classification

Thereafter, we proceed to evaluate the whole execution time of the optimized implementation compared to the sequential one, where values are illustrated in figure 6. The proposed optimization allows a speedup ratio equal to 3.88.

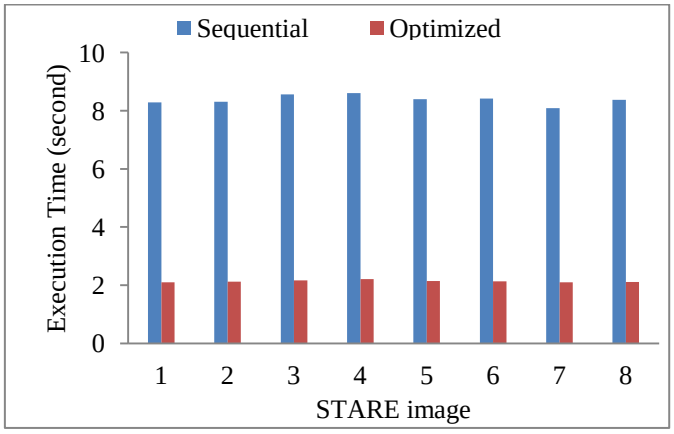


Figure 6.Execution time of sequential and optimized implementations in terms of STARE database images

Conclusion and future works

In this paper, a blood vessel segmentation approach in fundus image was implemented in GPU as shared memory architecture. The approach specificity consists on combining morphological operator and classifier in order to achieve a performing segmentation quality in a reduced execution time. However, a computational optimization is required due to the permanent rise of fundus image resolution and the timing constraint imposed by medical application.

In this purpose, we have identified the processing steps requiring higher execution times. Then, we implemented them on GPU architecture where experiment indicates a speedup ratio of 3.88 compared to the sequential scheduling. We aim in our future works to propose a blood vessel segmentation approach that allows increasing parallelism level in order to achieve a higher speedup.

REFERENCES

[1] Guohui Wang, Blaine Rister, and Joseph R. Cavallaro., "WorkloadAnalysis and Efficient OpenCL-basedImplementation of SIFT Algorithm on a Smartphone," GlobalSIP. (2013).

[2] Adrian R.G. Harwood and Alistair J. Revell., "Parallelisation of an interactive lattice-Boltzmann method on an Android-powered mobile device," Advances in Engineering Software 104, 38–50 (2017).
[3] Maria De Marsico, Chiara Galdi, Michele Nappi and Daniel Riccio., "FIRME: Face and Iris Recognition for Mobile Engagement," Image and Vision Computing 32, 1161–1172 (2014).
[4]Wenkai Wu, Qing He, Yongkang Wang, Youpan Hu, Baopu Li, Bin Leng, Guan Guan, Haibin Wang and HehuiZou., "An Improved Method of Vibe for Motion Detection Based on Android System," ROBIO (2013).
[5] Jyotiprava Dash, NilamaniBhoi., "A thresholding based technique to extractretinalblood vessels from funds image," Future Computing and Informatics Journal, 1-7(2017).
[6] ShahabAslani andHaldunSarnel., "A new supervised retinal vessels segmentation method based on robust hybrid features," Biomedical Signal Processing and Control 30, 1–12 (2016).
[7] Zhexin Jiang, Juan Yepez, Sen An andSeokbumKo., "Fast, accurate and robust retinal vessels segmentation system," biocybernatics and biomedicine engineering, 1-10(2017).
[8] Shuangling Wang, Yilong Yin, Guibao Cao, Benzheng Wei, YuanjieZheng andGongping Yang., "Hierarchical retinal blood vessel segmentation based on feature and ensemble learning," Neurocomputing, 1-10 (2014).
[9] ZHU Chengzhang, ZOU Beiji, XIANG Yao, CUI Jinkai and WU Hui., "An Ensemble Retinal Vessel Segmentation Based on Supervised Learning in Fundus Images," Chinese Journal of Electronics, 503-511(2016).
[10] SohiniRoychowdhury, Dara D. Koozekanani, and Keshab K. Parhi., "Iterative Vessel Segmentation of Fundus Images," IEEE Transactions on Biomedical Engineering, 1-12(2015).
[11] SudeshnaSilKar and Santi P. Maity., "Retinal blood vessel extraction using tunable bandpass filter and fuzzy conditional entropy," Computer Methods and Programs in Biomedicine, 1-22 (2016).
[12] SohiniRoychowdhury, Dara D. Koozekanani, and KeshabK.Parh., "Blood Vessel Segmentation of Fundus Images by Major Vessel Extraction and Sub-Image Classification," IEEE Journal of Biomedical and Health Informatics, 1-11(2014).
[13] Aziah Ali, AiniHussain, Wan Mimi Diyana and Wan Zaki., "vessel extraction in retinal image using automatic thresholding and GABOR WAVELET," Engineering in Medicine and Biology Society (EMBC), 1-4 (2017).
[14] ElahehImani, MaliheJavidi and Hamid-Reza Pourreza., "Improvement of Retinal Blood Vessel Detection Using Morphological Component Analysis," Computer Methods and Programs in Biomedicine, 1-23(2015).
[15] Ivo Soares, Miguel Castelo-Branco, and Antonio M. G. Pinheiro,"Optic Disc Localization in Retinal Imagesbased on Cumulative Sum Fields," IEEE Journal of Biomedical and Health Informatics, 574-585 (2016).
[16] A. García-Floriano et al, “A machine learning approach to medical image classification: Detecting age-related macular degeneration in fundus images”, Computers & Electrical Engineering,1-12(2017).