Journal of Engg. Research Vol. 7 No. (3) September 2019 pp. 161-177

Optic disc detection using fish school search algorithm based on FPGA

Sa'ed Abed*, Dalal Al-Oraifan and Amna Safar

Computer Engineering Department, College of Engineering and Petroleum, P.O. Box 5969, Safat 13060, Kuwait University, Kuwait
**Corresponding Author: s.abed@ku.edu.kw*

ABSTRACT

Many people worldwide suffer from Diabetic Retinopathy (DP). This health ailment affects their vision throughout the years, as they get older. The fundus image is examined for detecting diabetic diseases that could affect the retina such as the DP. Correctly detecting the optic disc is required to discover the disease. Several methods have been proposed to improve the detection of the optic disc with respect to different performance metrics. In this work, we investigate the performance, mainly the power consumption and the computational time, of the Fish School Search (FSS) technique. We detect the optic disc by using contrast enhanced multi-step pre-processing technique to improve the color fundus image. The pre-processing steps used in this work improve the quality of the colored image by filtering out the noises, smoothing the image, and masking out the regions where it is guaranteed that the optic disc is not located in. The FSS algorithm is applied to find the brightest pixel in the pre-processed image, which is marked as the optic disc. The algorithm is also implemented in the FPGA to benefit from the parallel processing power of the FPGA. The algorithm is tested on DRIVE, STARE, and RiaRetDB1 databases and compared to other methods in the literature. The accuracy of the FSS was 100%, 95.7%, and 99.92% when using DRIVE, STARE, and RiaRetDB1 databases, respectively. Moreover, the running time of the FPGA implementation was found to be 1.605 ms with a total power dissipation of 121.818 mW.

Keywords: Diabetic retinopathy; fish school search techniques; optic disc detection; pre-processing; FPGA; speed and accuracy.

INTRODUCTION

Diabetes is one of the most prevalent diseases in the world. By 2013, it was estimated that 2-4% of people were suffering from Diabetic Retinopathy (DP) as a complication (Pires *et al*., 2013). Diagnosing the patients' eyes in early stages can prevent blindness caused by the disease. Fundus photography is the process of capturing an image of the back of the eye, which is called fundus. Figure 1 shows an example of fundus image. These images can be used to diagnose the symptoms affecting the retina of diabetic patients.

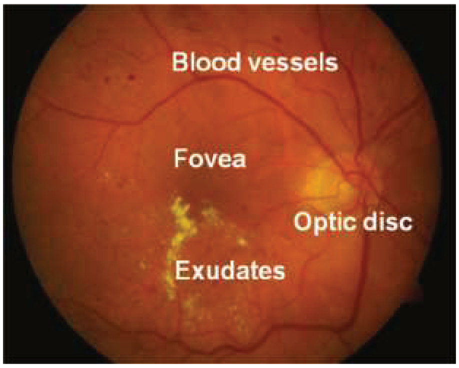


Fig. 1. Optic disc in fundus image (Jaafar *et al*., 2011).

For a successful diagnosis, the fundus image should be segmented to distinguish the structures of the retinal from any abnormalities in the eye fundus. Optic disc is a bright yellow disc that serves as an entry point for the retinal blood vessels as shown in Figure 1. Correctly detecting the optic disc is helpful as it can show the changes that are due to the diabetic retinopathies infection.

It is very hard and time consuming to detect the optic disc manually from the fundus image. Manually detection could also lead to false localization of the optic disc in many cases when the fundus image is affected by noise. Therefore, the process of detecting the optic disc has been automated. There are many proposed contributions using different image processing algorithms for optic disc detection that have good performance results in terms of accuracy and running time. Recent researches have focused on the remarkable results of the swarm intelligence techniques (Abed *et al*., 2016). Swarm intelligence techniques are based on the natural behavior of self-organized systems. Usually, the natural behavior of some creatures is based on simple rules and conditions to solve specific problems. This behavior is based on random individual interaction and can be used to write simple and powerful algorithms for solving very difficult and complicated problems.

Motivated from the outperformance of the previously used swarm intelligence algorithms in the field of segmenting the fundus image, we apply the Fish School Search (FSS) technique along with multiple pre-processing techniques to explore its efficiency in terms of running time and accuracy in locating the optic disc. The authors in Lima *et al*. (2013) proposed the FSS algorithm in which it imitates the behavior of the fish schools searching food. We compare the performance of our method with a similar work in the literature using different swarm intelligence techniques (Abed *et al*., 2016). In the time of conducting this work, the fish schooling technique has not been used in the literature to detect the optic disc. We implement FSS algorithm on Field Programmable Gate Array (FPGA) to minimize running time and reduce the power needed to detect optic disc area. Before starting to detect the optic disc in the fundus image, the fundus image must be pre-processed using several steps. We introduce an improvement on the pre-processing steps used in Abed *et al*. (2016) to enhance the fundus image before applying the selected retinal image-processing algorithm. First, the fundus images retrieved from the public databases for fundus optic disc images undergo several median filtering, contract enhancement, and masking the areas where optic disc cannot be found. Then, the FSS algorithm is applied to find the peak pixel in the resultant image that represent the optic disc. Figure 2 shows an outline of the proposed method.

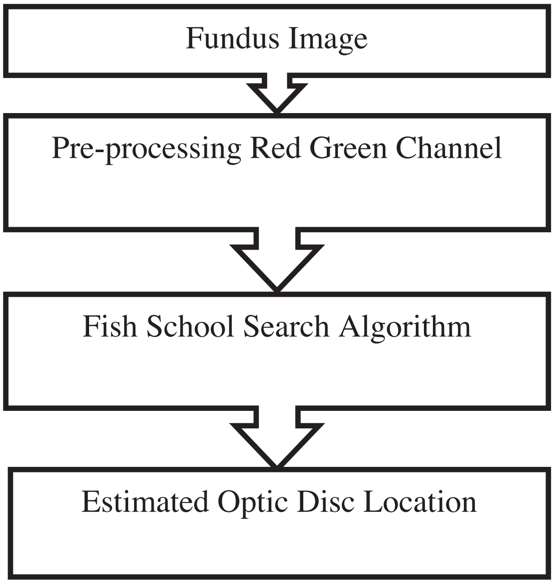


Fig. 2. Outline of the proposed methodology.

The detection accuracy of the FSS algorithm was measured on DRIVE, STARE and RiaDetDB1 databases and found to be 100%, 95.7%, and 99.92%, respectively. The computational time of the proposed FPGA implementation of the FSS algorithm is 1.605 ms, with extra 0.254 seconds needed to pre-process an image from DRIVE, 0.1597 seconds needed for an image from STARE dataset, and 0.261 seconds for images from RiaDetDB1. The FPGA implementation runs with a total thermal power dissipation of 121.818 mW.

The outline of this paper is as follows. Section 2 gives a literature review. Section 3 explains the proposed methodology for detecting the optic disc. Section 4 presents the experimental results and the discussion of the work. Finally, section 5 concludes the paper and presents some future trends.

RELATED WORKS

The literature shows that detecting the optic disc is still an interesting research topic. Several papers have been published to propose new methods and techniques that could achieve superior results in terms of efficiency and accuracy. The authors in Zhang *et al*. (2016) located the horizontal and the vertical coordinate of optic disc using vessel distribution and directional characteristics, respectively. They considered three characteristics of the vessel distribution represented by local vessel density, compactness, and uniformity. To increase the accuracy of the method, the vessel direction is studied to obtain the vertical coordinate of the optic disc by applying Hough transformation.

The authors in Zahoor *et al*. (2017) proposed an approach to locate and segment the optic disc by applying morphological operations to eliminate the retinal vasculature. Afterwards, they applied Circular Hough transform and Polar transformation as optic disc locators and to get the region of interest. Finally, the optic disc boundary is identified using Adaptive thresholding.

The work in Soares *et al*. (2016) derived a new vessel enhancement method that was used along with the morphological operators. The cumulative sum was computed in the vertical and horizontal vessel orientations to find the initial optic disc position. High vasculature convergence and high intensity values features were used for the final optic disc localization.

In Manjiri *et al*. (2015), a system was proposed, named Automated Diabetic Retinopathy detection System that is made up of five stages. The third stage is concerned with the optic disc localization after the fundus image has been pre-processed and masked in the preceding two stages. Initially, the green channel in the fundus image is enhanced by removing the noise from background and applying histogram equalization. Next, the red channel of the image is applied to a thresholding operation for mask detection and identifying the boundary. Speed-up robust features were used to locate the optic disc in the green channel.

The authors in Ponnaiah *et al*. (2013) used the blue and the green channel intensity estimation that was applied in the fitness function of the genetic algorithm for detecting the optic disc, a nonlinear optimization technique, repeatedly applying the objective function to find the best location of the optic disc. They designed a direct search based algorithm to speed up the process of locating the potential regions.

In Qureshi *et al*. (2012), groups of different optic disc detection algorithms were used such as the pyramidal decomposition, edge detection, entropy filter, Hough transformation, and feature vector and uniform sample grid. Each algorithm will provide a potential optic disc candidate, and the region that encloses most of the algorithms result will be selected as the optic disc.

To locate the optic disc, illumination correction technique was used in Hsiao *et al*. (2012). This gives a differentiation between the optic disc and the background as the illumination operator will keep the bright area of the image. For optic disc boundary segmentation, the Canny edge detector and the Hough transform are used to obtain an initial contour image that is fed into the next stage of the proposed algorithm. In addition, a combination of the supervised gradient vector flow snake model and supervised classification is used to remove the optic disc boundary. Obtaining results iteratively will update the snakes operators to get better edge detection.

Information captured from the vessels in the fundus image can be used for optic disc detection. The authors in Wigdahl *et al*. (2017) converted the retinal image into a graph to calculate the edge weights. The edge weights calculation is carried out by applying shortest path algorithm, namely, Dijkstra, between pair of points. The optic disc is found where we can find a part in the graph with the maximum number of shortest paths using template matching and the vertical edge detection.

The work in Wyawahare *et al*. (2014) compared between five methods for optic disc segmentation including distance regularized level set, Otsu thresholding, region growing, particle swarm optimization, and Generalized Regression Neural Network (GRNN). The experiments were done on the maximum pixels intensity of the V component of the HSV fundus image. The study shows that the GRNN outperforms other methods with low computational time.

In Alshayeji *et al*. (2017), the authors used multi-step pre-processing including anisotropic diffusion of a gray-scaled fundus image to improve the image before detecting the optic disc. Then, it was followed by the edge detection inspired by the gravitational law based algorithm. Thresholding was used in the post-processing steps after the unneeded areas were removed for simplicity. Finally, the optic disc was selected among possible candidates based on a novel proposed technique called candidate selection.

Recently, the swarm intelligence techniques have received much attention for the detection of the optic disc. These techniques are nature-inspired algorithms that provide advancement in accuracy and computational time. The authors of Pereira *et al*. (2013) detected the optic disc using anisotropic diffusion followed by ant colony optimization algorithm. The paper of Devasia *et al*. (2015) proposed a method using histogram based particle swarm optimization technique. The authors in Abed *et al*. (2016) proposed a new pre-processing scheme called Background Subtraction-based Optic Disc Detection (BSODD) along with a study of the performance of several swarm intelligence algorithms for optic disc detection. The used algorithms are artificial bee colony, particle swarm optimization, bat algorithm, cuckoo search, and firefly algorithm. The pre-processing scheme in Abed *et al*. (2016) shows an enhancement of the performance of the swarm intelligence algorithms especially with the firefly algorithm.

For enhancement of feature extraction, the authors in Kimori (2016) applied mathematical morphology techniques on medical image processing. With the variety choices of algorithms to apply for detection, still most of the algorithms and methods take high execution time to compute the results due to the high complexity. To improve the average computation time, researchers started to implement their algorithms based on hardware architecture especially using FPGA. Being able to have a parallel processing and other features that VLSI implementation offers will have a positive impact on the performance of the method. The authors in Majeed *et al*. (2015) implemented the Sobel edge detection and segmentation method using the FPGA to detect the exudates.

The paper of Hadj Fredj *et al*. (2015) improved the performance of the Speckle Reducing Anisotropic Diffusion (SRAD), an anisotropic diffusion technique. They applied FPGA based implementation of the SRAD on a fundus image to remove the noise. The results showed an improvement over the MATLAB implementation.

Our methodology is based on benefiting from the efficient detection of the optic disc using the swarm optimization techniques. We apply the FSS algorithm to detect the optic disc in a shorter time than the other proposed methods in the literature. To improve the quality of the applied algorithm, we apply multi-step pre-processing techniques on the green and red channels of a colored fundus image.

PROPOSED METHODOLOGY

There is no one optimal algorithm that can be used for optic disc detection and most of the algorithms can lead to an extensive consumption of time. In this work, we introduce an improvement of optic disc detection in terms of computational time and energy by implementing the FSS algorithm on the FPGA. The methodology is divided into two steps; first, the fundus image initially passes through many pre-processing steps where we enhance the pre-processing techniques proposed in Abed *et al*. (2016) to improve contrast and remove false peaks from the fundus image. The pre-processing procedures that we follow are rescaling and conversion to Red Green channels, median filtering, Adaptive Histogram Equalization enhancement, and masking operations, which have been proven experimentally that they are essential to exclude the regions that the optic disc cannot be found. The second step of the method is applying the FSS algorithm to detect the optic disc in fundus image based on FPGA. The following subsections provide more detailed information about the pre-processing techniques and the FSS algorithm.

Pre-processing steps

The authors in Abed *et al*. (2016) proposed the Background Subtraction based Optic Disc Detection (BSODD). It is a series of operations used to smooth the fundus image and improve the accuracy of the used algorithm when detecting the highest grayscale value (peak) in an image, which represents the possible optic disc location. In this work, the process of pre-processing the fundus image starts with rescaling the size of the image to a moderate size 128x128. The rescaling process of the fundus image reduces the processing time without affecting the accuracy of optic disc detection. Then, the rescaled image is transformed to Red and Green channels image by removing the blue channel, as the information proposed in the latter channel is nearly low to non-useful in the process of detecting the optic disc. After that, the noise is removed by using median filtering of 12x12 pixel kernel. Each pixel is represented by the median value that is calculated by sorting the pixel values of the entire surrounding neighborhood (neighboring), in this case a square of 12x12, then choose the median of these sorted values. The window size is experimentally proven by Abed *et al*. (2016) that it is the best size to remove noise and smoothen the image without affecting the edges of fundus image. Median filter is applied on Red and Green channel images separately.

Next, Adaptive Histogram Equalization (AHE) is applied on the Red channel image and the Green channel image separately to improve image contrast. It uses the distribution of light and dark values of the image. It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image.

Afterwards, a median filtering of 4x4 and mean filtering of 5x5 are applied to smooth the Red and Green channels images. Before applying masking, the Red channel and Green channel images are concatenated.

To boost the performance of the applied algorithm for detecting the optic disc, the pixels where there is a probability that the optic disc is not located there are masked. Referencing a 128x128-pixel image, and a circular area as the region of interest, the masking eliminates pixels that are in the range of 10xWidth/128 from the edge of the region of interest, 5xWidth/128 from the center of the image, and 25xHeight/128 of height of the image. Figure 3 shows the flowchart of the pre-processing steps. The authors of Abed et al. (2016) proved that the optic disc could not be found in the region outside the Region of Interest (ROI). The result will be a rectangular image cropped by 25 pixels from the height of the image and 10 pixels from the width. Add to that a circular region with radius of 5 pixels from the center of the image also will be excluded.

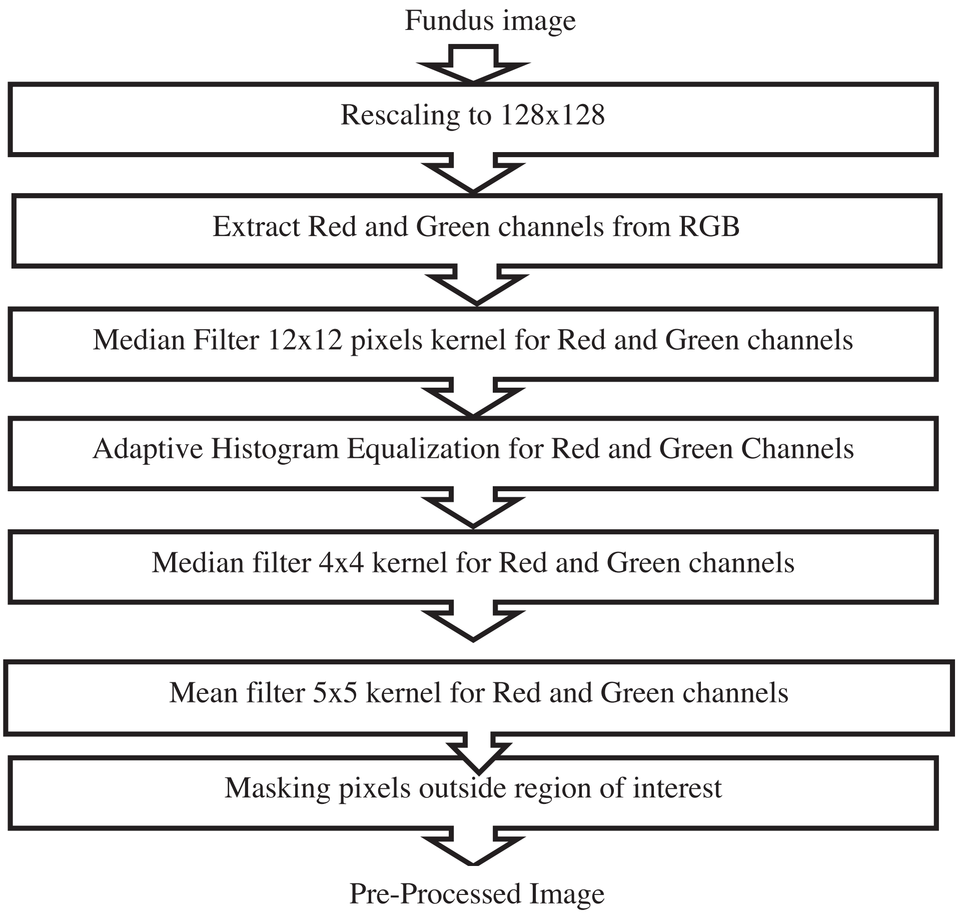


Fig. 3. Series of steps used in the pre-processing.

Figure 4 shows the original fundus image and the image after BSODD pre-processing that were introduced by the authors of Abed *et al*. (2016). Figure 5 displays the colored image after using the pre-processing steps, which are presented in this work.

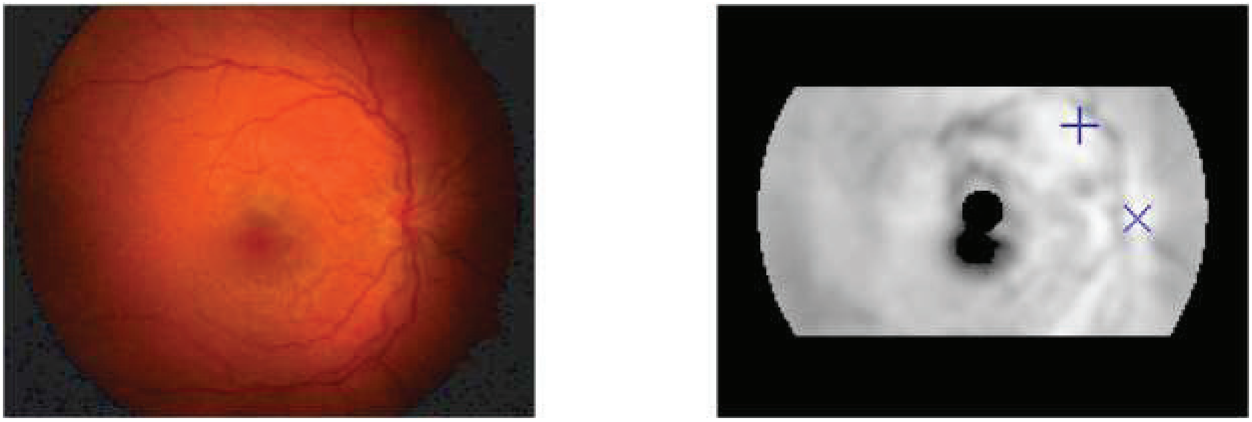


Fig. 4. The resultant image after applying BSODD (Abed *et al*., 2016).

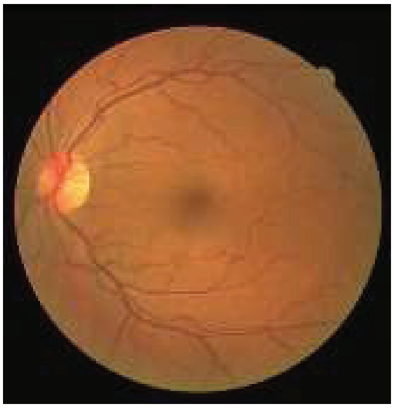
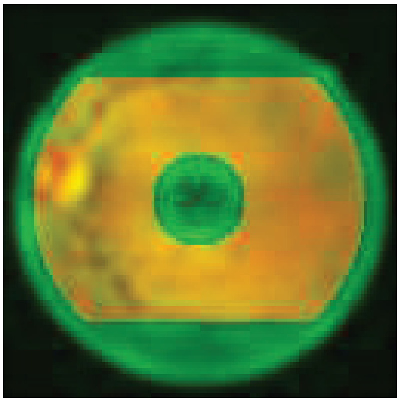
 

Fig. 5. The original image on the left and the image after the proposed pre-processing steps on the right.

Fish school search algorithm

Many latest studies nowadays use algorithms that are based on the natural behavior of some living species. One of the efficient search algorithms that are inspired by the fish behavior is Fish School Search (FSS) (Lima *et al*., 2013). FSS is mainly inspired by the behavior of swimming fishes to collect their food. The FSS algorithm uses operators that can be divided into two groups: feeding and swimming.

We use the FSS Algorithm to detect the optic disc in fundus images as it excels when applied on searching problems. In addition, FSS considered a new method to detect the optic disc, and no proposed similar work was found in the literature.

FSS has three swimming operators, individual movement, collective instinctive movement, and collective volitive movement operator. Based on the size of the searching problem and difficulty, one can use one or more of these swimming operators to achieve the desired result. In this work, we explore the performance of the individual swimming operator in detecting the optic disc location. Algorithm 1 represents the pseudo-code of the FSS algorithm. Initially, each fish holds a spot that represents a possible solution to the problem, a pixel that represents the optic disc location. In each iteration, each fish inspects the neighborhood to find its new position as shown in Equation (1) (see line 1).

$$n_i(t) = x_i(t) + s(t)\, rand \qquad \ldots\ldots\ldots\ldots\ldots\ldots \qquad (1)$$

where n_i is the neighbor position, x_i current position, s is the individual step, and rand is random value. Individual movement only occurs if the estimated new position has a better fitness function value than the current position as shown in line 3. The fitness function value is based on the pixel value as specified in line 2. The greater that value is, the more likely the pixels represent the optic disc. After a fair number of iterations (line 4), the fishes start to be around bay center where the optic disc is located.

The advantage of using the FSS Algorithm over the other swarm intelligence algorithms is that the FSS algorithm uses much less time because it can detect the bay center and move toward it quickly, which leads it to move to the solution faster.

Algorithm 1: *The pseudo-code of the FSS algorithm*
Initialize all fish positions;
While *itr the stopping criterion is not met* **do**
For *each fish* **do**
Begin
1. Find the next neighboring position using Equation (1);
2. Evaluate fitness of the expected new position;
3. Execute individual movement operator, only if new position has better fitness value;
End for
4. Increment Individual Step by 1;
End while

Algorithm 1. The used Fish School Search Algorithm pseudo-code.

In order for a fish to determine if its current position is more likely to be the possible optic disc location or close to it, the neighbor position is needed to be inspected as described in Equation (1). The individual step is offering the distance of how far the new position will be from the current position of the fish. Due to the size of the problem, the individual step is evaluated to be equal to one. The random value is used to determine the direction of the new position of the fish. With the aid of Equation (1), the fish will determine if it is better to move to the neighbor position, which is one of the 8 pixels neighboring the current pixel, or stay in the current place during the currently running iteration.

A simple example to explain how the Fish Search School work is as follows: Assume the problem area is represented by 4X4 array and number of fishes is 2. Initially, the positions of the fishes can be randomly selected or initialized to specific locations. Each location has a pixel value; the value of the pixel represents the brightness of the image. In this example, the address of the pixels is expressed as small numbers on the bottom right side of the pixels in the array. The optic disk is viewed as the brightest pixel; so during the search, the fish will change their positions and swim toward the locations where the pixels have larger values. The movement of the fish is based on the individual swimming operator. Equation (1) is used as described above.

$$\begin{bmatrix} 58_0 & 73_1 & 84_2 & 92_3 \\ 70_4 & 83_5 & 98_6 & 100_7 \\ 95_8 & 120_9 & 175_{10} & 203_{11} \\ 105_{12} & 166_{13} & 190_{14} & 215_{15} \end{bmatrix}$$

At any positions in the fish school, after a couple of iterations the new positions of the fish will be near the highest pixel value, if not already pointing to it.

For instance, assuming one of the fish in a location with an address equals 10 and holding pixel value equals to 175. In the first iteration, Equation (1) will be evaluated and the new address is found to be 11 and the pixel value is 203. This new position holds a better pixel value than the current one; therefore, the fish will update its location to the new calculated address. We keep undergoing Equation (1) in the next iterations, but this time any new surrounding positions obtained will not be accepted by this fish due to the pixel value it holds (203), which is greater than its neighbors' pixel values, except one new neighbor position whose location is 15 with a pixel value equal to 215. When the individual swimming operator provides this fish with the address of the neighbor located in 15, the fish will evaluate and accept its new position using the fitness function. This means the fish may stay in its position (11) for a couple of iterations before it moves to the new position (15).

The input of the algorithm is the pre-processed fundus image 128x128 pixels. The empirical parameters used in the FSS algorithm are shown in Table 1. We experimentally found that these parameters lead to efficient results.

Table 1. FSS algorithm parameters.

Parameter	Value
Number of fishes	8
Number of iterations, itr	10
Initial value for individual step	1

FPGA implementation

We introduce an FPGA implementation to improve the computational time and the power consumption of the FSS algorithm. An overview of the proposed design is shown in Figure 6.

Fig. 6. Proposed design overview.

The data path of the design is presented in Figure 7. More details can be found in Appendix A. Initially, the pre-processed image is stored in the Random Access Memory (RAM). Then the image is divided into a number of batches, and FSS algorithm is applied on each batch to detect the best possible pixel that represent the optic disc. The design undergoes four iterations; in each iteration, the number of batches tested is less than the previous iteration. Finally, the last iteration examines two batches and results in two possible optic disc locations, which are both found to be within the center of the optic disc area. Each batch holds eight pixels and the number of batches in each iteration is 1257, 157, 19, and 2, respectively. These values are calculated based on the number of analyzed pixels.

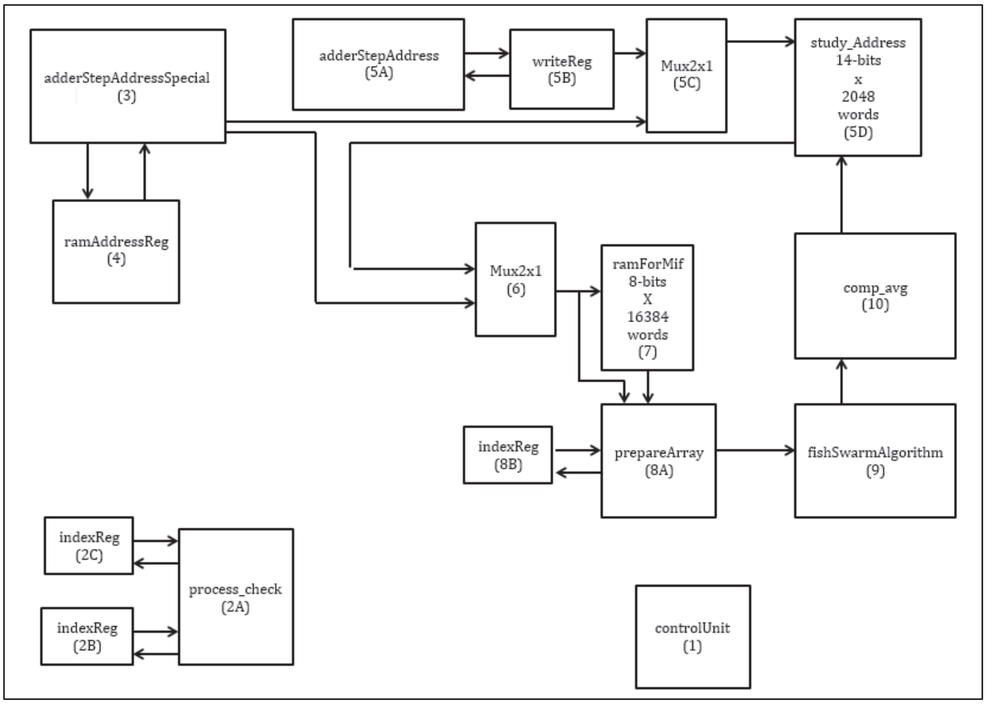


Fig. 7. The datapath of the proposed design.

The resulted value represents the optic disc location in the stored image in RAM, to get the exact location on the image. Equations (2) and (3) are used.

$$x\text{-}axes = address\ of\ pixel\ in\ RAM\ \mathbf{mod}\ width\ of\ the\ image \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

$$y\text{-}axes = address\ of\ pixel\ in\ RAM\ \mathbf{div}\ width\ of\ the\ image \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

where mod is the modulus operator, while div is the division operator. The width of the image is 128.

EVALUATION RESULTS AND DISCUSSIONS

Three public datasets are used in this work: DRIVE, STARE, and DiaRetDB1. DRIVE (Digital Retinal Images for Vessel Extraction) is a public database that has 40 retinal images to be used by researchers to detect blood vessels and optic disc detection (Staal *et al*., 2004). STARE (Structured Analysis of the Retina) is a database of 20 different retinal images for public research projects (Hoover *et al*., 2000). DiaRetDB1 (Standard Diabetic Retinopathy Database Calibration level 1) is a database of 89 images; 84 of these images contain symptoms of DR, and the remaining 5 do not have any symptoms of the disease (Kauppi *et al*., 2007). First, the images are examined manually and the optic disc location and radius are marked using "paint" application in windows. After that, the images are pre-processed using MATLAB with the previously mentioned procedures. MATLAB procedures were run on Intel® Core™ i7 CPU L 640 @ 2.13 GHz 4 GB RAM, and an operating system of 64-bit Windows 8.

The pre-processing steps for DRIVE database images took 0.254 seconds per image, 0.1597 seconds per image for STARE database images, and 0.261 seconds for DiaRetDB1 database. Compared to the pre-processing steps in Abed *et al*. (2016), DRIVE pre-processing steps take 0.695 seconds per image, 0.461 seconds per image for STARE, and 0.711 seconds per image for DiaRetDB1. Figure 8 shows the difference between the different pre-processing steps.

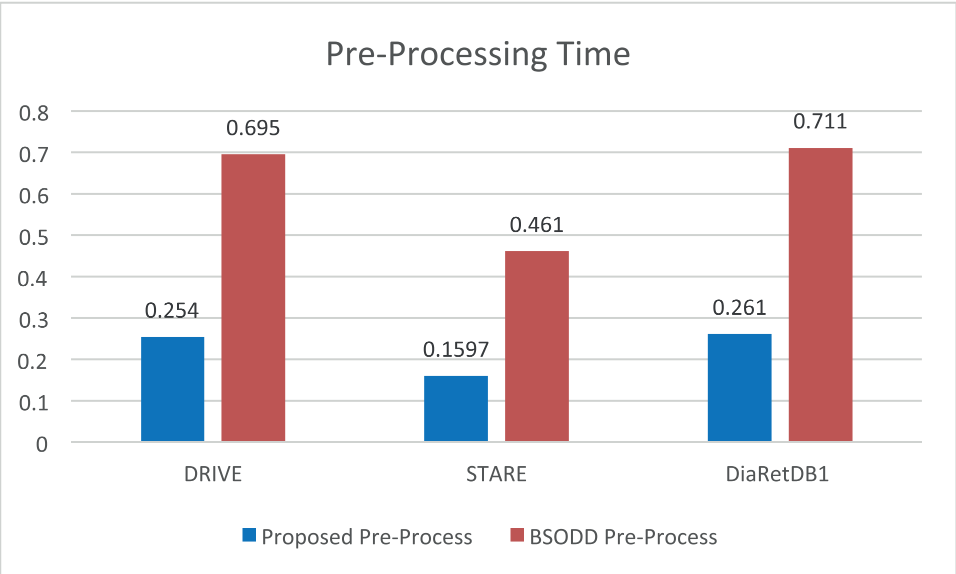


Fig. 8. Pre-processing time for BSODD and the proposed pre-processing steps.

The implementation of the FSS algorithm is written in VHDL. The selected device family is Cyclone II. The simulation is carried out using Quartus II on Intel® Core™ i7 CPU M 620 @ 2.67 GHz, 2 GB RAM, and operating system of 64-bit Windows 8. Table 2 shows a comparison between previous works reported in the literature and our proposed methodology in terms of running time.

Table 2. Comparison between the running times of detecting the optic disc.

Algorithm Name	Running time (s)		
	DRIVE	STARE	DiaRetDB1
Particle Swarm Optimization	2.05	2.08	2.06
Artificial Bee Colony	11.12	10.65	11.23
Bat Algorithm	6.34	6.03	5.87
Cuckoo Search	1.74	1.52	1.84
Firefly Algorithm	1.72	1.49	1.81
Proposed FSS (using Matlab)	1.69	1.48	1.76
Proposed FSS (using FPGA)	$0.254+1.6053*10^{-3}$	$0.1597+1.6053*10^{-3}$	$0.261+1.6053*10^{-3}$

In terms of running time, the FSS algorithm implemented in FPGA was the fastest with extra 0.254 seconds needed to pre-process an image from DRIVE, 0.1597 seconds from STARE, and 0.261 seconds needed to pre-process an image from RiaRetDB1. The firefly and cuckoo algorithms running time was close to the FSS algorithm implemented in Matlab. The ABC algorithm was the slowest.

The total thermal power dissipation is found to be 121.818 mW, and the core dynamic thermal power dissipation is 56.112 mW. The core static thermal power dissipation is 2.69 mW, and 65.436 mW is the input/output thermal power dissipation.

Accuracy percentage is computed with respect to the previously marked radius of the optic disc to find the error rate by dividing the distance between the estimated optic disc location and the actual location. The estimated value is accepted when the error rate is less than one. The mean of the accuracy results was compared to other results as shown in Table 3.

Table 3. Detection accuracy results compared with previously published work.

Algorithm Name	Detection Accuracy (%)		
	DRIVE	STARE	DiaRetDB1
Particle Swarm Optimization	100	91.25	98.60
Artificial Bee Colony	100	90	100
Bat Algorithm	100	93.75	98.88
Cuckoo Search	100	92.5	99.44
Firefly Algorithm	100	95	100
Proposed FSS (using Matlab)	100	95.7	99.92
Proposed FSS (using FPGA)	100	95.7	99.92

In terms of detection accuracy, the FSS algorithm had the highest accuracy, followed by the firefly algorithm. Both ABC algorithm and cuckoo search algorithm had achieved the same detection performance. The PSO had the worst detection performance.

CONCLUSION

This work represents detecting the optic disc from a colored fundus images using the Fish School Search (FSS) algorithm. We proposed multi step pre-processing techniques to enhance the colored image followed by the selected swarm intelligence technique, the FSS algorithm. These pre-processing steps remove the variation of pixels and make the optic disc area more clear. The hardware FPGA implementation of the FSS algorithm gives marvelous results in terms of execution time and accuracy when detecting the optic disc. We propose a new power efficient methodology for locating the optic disc. Comparing the proposed implementation of the FSS algorithm and the other previously used algorithms in the literature, the running time of the hardware implementation is much faster than using a software implementation of the algorithms. In future work, we will introduce a new hardware design that supports parallelization and more resource optimization. We will also compare the performance of the other swimming operators in the FSS algorithm when using all together to detect the optic disc.

REFERENCES

Abed, S., Al-Roomi, S.A. & Al-Shayeji, M. 2016. Effective optic disc detection method based on swarm intelligence techniques and novel pre-processing steps. Applied Soft Computing **(49)**: 146-163.

Alshayeji, M., Al-Roomi, S.A. & Abed, S. 2017. Optic disc detection in retinal fundus images using gravitational law-based edge detection. Medical & biological engineering & computing **55**(6): 935-948.

Devasia, T., Jacob, P. & Thomas, T. 2015. Automatic Optic Disc Localization and Segmentation using Swarm Intelligence. World of Computer Science & Information Technology Journal **5**(6): 92-97.

Hadj Fredj, A., Ben Abdallah, M., Malek, J. & Azar, A.T. 2015. Fundus image denoising using FPGA hardware Architecture. International Journal Computer Applications in Technology **54**(1): 1-13.

Hsiao, H.K., Liu, C.C., Yu, C.Y., Kuo, S.W. & Yu, S.S. 2012. A novel optic disc detection scheme on retinal images. Expert Systems with Applications **39**(12): 10600-10606.

Hoover A, Kouznetsova V & Goldbaum M. 2000. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. IEEE Transactions on Medical Imaging **19**(3): 203–210.

Jaafar, H.F., Nandi, A.K. & Al-Nuaimy, W. 2011. Decision support system for the detection and grading of hard exudates from color fundus photographs. Journal of Biomedical Optics **16**(11): 116001.

Kauppi, T., Kalesnykiene, V., Kamarainen, J.-K., Lensu, L., Sorri, I., Raninen A., Voutilainen R., Uusitalo, H., Kälviäinen, H. & Pietilä, J. 2007. Diaretdb1 diabetic retinopathy database and evaluation protocol. In British Machine Vision Conference – BMVC, University of Warwick: 252–261.

Kaur, S. 2013. Contrast enhancement techniques for images-A visual analysis. International Journal of Computer Applications **64**(17): 20-25.

Kimori, Y. 2011. Mathematical morphology-based approach to the enhancement of morphological features in medical images. Journal of clinical bioinformatics **1**(1): 33-40.

Lima, A.C. & de Castro, L.N. 2013. An Enhanced Fish School Search Algorithm. Brics Congress on Computational Intelligence & 11th Brazilian Congress on Computational Intelligence: 152-157.

Majeed, N.A. & Rao, S. 2015. Hardware Implementation of Retinal Image Processing Algorithm on FPGA. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering **3**(1): 157-161.

Manjiri, P., Ramesh, M., Yogesh, R., Manoj, S. & Neha, D. 2015. Automated localization of optic disk, detection of microaneurysms and extraction of blood vessels to bypass angiography. In Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA). Advances in Intelligent Systems and Computing: 579-587.

Pereira, C., Gonçalves, L. & Ferreira, M. 2013. Optic disc detection in color fundus images using ant colony optimization. Medical & biological engineering & computing **51**(3): 295-303.

Pires, R., Jelinek, H.F., Wainer, J., Goldenstein, S., Valle, E. & Rocha, A. 2013. Assessing the need for referral in automatic diabetic retinopathy detection. IEEE Transactions on Biomedical Engineering **60**(12), 3391-3398.

Ponnaiah, G.F.M. & Baboo, S.S. 2013. GA based automatic optic disc detection from fundus image using blue channel and green channel information. International Journal of Computer Applications **69**(2):23-31.

Qureshi, R.J., Kovacs, L., Harangi, B., Nagy, B., Peto, T. & Hajdu, A. 2012. Combining algorithms for automatic detection of optic disc and macula in fundus images. Computer Vision and Image Understanding **116**(1): 138-145.

Soares, I., Castelo-Branco, M. & Pinheiro, A.M. 2016. Optic disc localization in retinal images based on cumulative sum fields. IEEE journal of biomedical and health informatics **20**(2): 574-585.

Staal J, Abràmoff MD, Niemeijer M, Viergever MA & van Ginneken B. 2004. Ridge-based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging **23** (4): 501–509.

Wigdahl, J., Guimaraes, P. & Ruggeri, A. 2017. A shortest path approach to optic disc detection in retinal fundus images. Journal for Modeling in Ophthalmology **1**(4): 29-42.

Wyawahare, M.V. & Patil, P.M. 2014. Performance Evaluation of Optic Disc Segmentation Algorithms in Retinal Fundus Images: an Empirical Investigation. International Journal of Advanced Science and Technology **(69)**: 19-32.

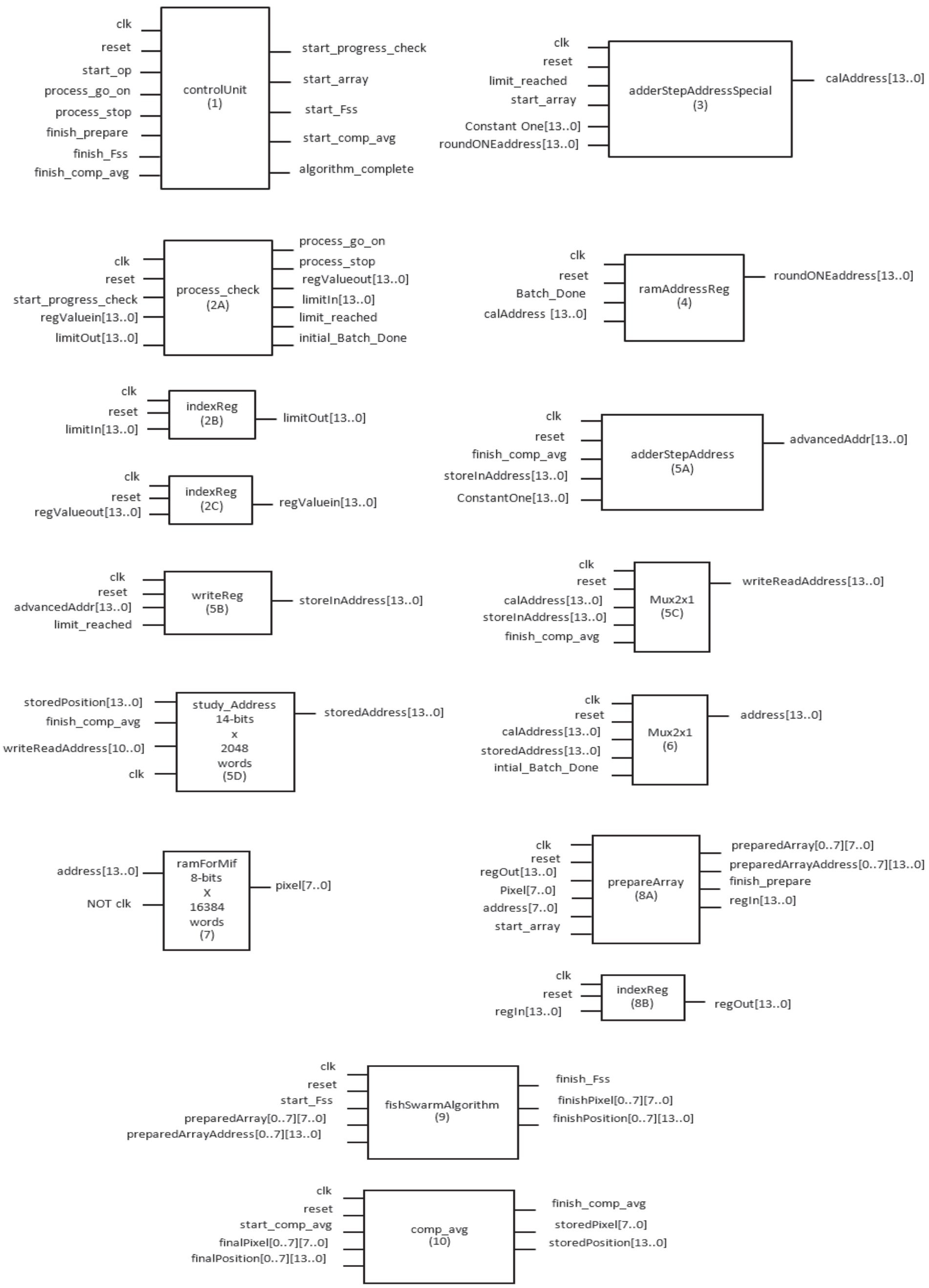
Zahoor, M.N. & Fraz, M.M. 2017. Fast Optic Disc Segmentation in Retina Using Polar Transform. IEEE Access **(5)**: 12293-12300.

Zhang, D. & Zhao, Y. 2016. Novel accurate and fast optic disc detection in retinal images with vessel distribution and directional characteristics. IEEE journal of biomedical and health informatics **20**(1): 333-342

Submitted: 08/10/2017
Revised: 20/02/2018
Accepted: 06/05/2018

APPENDIX A:

Fig. 9. Close up look into the input/output of each component of the design



Each component in the proposed design shown in Figure 9 is explained next:

1) Control Unit.

The control unit is responsible of starting and monitoring the other components of the design. Based on the received input signal, it triggers the required next component to start.

2A) Process check.

This component plays a role key in detecting the progress of detecting the optic disc location. It keeps track of the iteration number by communicating with (indexReg 2B), and the batch number from (indexReg 2C). Upon reset, it initializes the limit of the first iteration to 1257. Then the iteration limits will decrease to 157, 19, and 2. If the limit is not reached, the batch number will be incremented by 1. process_go_on signal is generated if the limit is still not reached, while process_stop signal is generated at the end of the last iteration.

3) adderStepAddressSpecial.

This adder is responsible of advancing the address pointer in a single batch based on the performance of the reading operation held by (prepareArray 9A) and (indexReg 9B). Upon reset the register is initialized to point to address 3098, all addresses before this point are empty that is having no pixel value but zero.

4) ramAddressReg.

This register holds the next address that will be used in (adderStepAddressSpecial).

5A) adderStepAddress.

Used to advance the address that points to where the write operation will be performed on (study_Address 5D), the address is retrieved from (writeReg 5B). This component is activated when finish_comp_avg is set to 1.

6B) writeReg.

This register holds the next address that will be used in the write operation.

5C) Mux2x1.

A regular multiplexer, used to select between two different addresses for (study_Address 5D) RAM, one used for write operation and the other for read. The writing address is retrieved from (writeReg 5B), while the reading address is retrieved from (adderStepAddressSpecial 3). The select signal is based on the value of finish_comp_avg signal. When the value is 0, the operation is read operation, otherwise it is write operation.

5D) study_Address.

RAM memory that is used to store the best values (possible optic disc location) that were detected by (comp_avg 10). The stored data is the address of the pixels. There are 2048 words in this RAM with a capacity of 14-bits each.

6) Mux2x1.

The first iteration of the design examines the main RAM memory (ramForMif 7) sequentially. Then, all other iterations will retrieve the addresses from (study_Address 5D) memory. To allow this operation, a multiplexer is used in this stage. The select signal is based on the value of initial_Batch_Done signal, raised from (process_check 2A). Where the low signal indicates that the initial iteration is still going on, and the high signal indicates that the first iteration is completed.

7) ramForMif.

The main RAM memory, in which the mif file of the pre-processed image is stored. The pixels are represented with 8-bits.

8A) prepareArray.

This component prepares an array of 8 pixels and their addresses from the (ramForMif 7) memory to be inputted to the FSS algorithm. For tracking the indices, (indexReg 8B) is used.

8B) indexReg.

A register that is used to record the progress of storing the pixels in (prepareArray 8A).

9) fishSwarmAlgorithm.

Used to execute the FSS algorithm. It takes an array of 8 pixels and performs the FSS algorithm on it. The resultant arrays show the positions and the pixel values that the “fishes” of the FSS algorithm are selecting as the best possible solutions.

10) comp_avg.

In this stage, the results of the FSS algorithm are studied and only one value is selected to be the best possible solution found in a batch that may represent the optic disc location. The address of the selected candidate is written in (study_Address 5D).

الكشف عن القرص البصري باستخدام خوارزمية سرب الأسماك (FSS) استناداً على FPGA

سائد عبد، دلال العريفان وآمنة صفر
قسم هندسة الحاسوب، كلية الهندسة والبترول، جامعة الكويت، الكويت

الخلاصة

يعاني كثير من الناس في جميع أنحاء العالم من اعتلال الشبكية السكري. ويؤثر هذا المرض على رؤيتهم على مر السنين، كما أن المشكلة تتفاقم مع التقدم في السن. يتم فحص صورة القاع للكشف عن أمراض السكري التي يمكن أن تؤثر على شبكية العين مثل اعتلال الشبكية السكري. المطلوب هو الكشف الصحيح عن القرص البصري لاكتشاف المرض. تم اقتراح عدة طرق لتحسين الكشف عن القرص البصري فيما يتعلق بمقاييس الأداء المختلفة. نقوم في هذا العمل بدراسة الأداء، وتحديداً استهلاك الطاقة والوقت الحسابي لتقنية أسراب الأسماك (FSS). ونقوم بالكشف عن القرص البصري باستخدام تقنية التباين المُحسن متعدد الخطوات قبل المعالجة لتحسين لون صورة القاع. خطوات المعالجة المسبقة المُستخدمة في هذا العمل تحسن من جودة الصورة الملونة عن طريق تصفية الضوضاء، وتسوية الصورة، وإخفاء المناطق التي يضمن فيها عدم وجود القرص البصري. تم تطبيق خوارزمية FSS للعثور على البكسل الأكثر إضاءة في الصورة المُعالجة مسبقاً، والذي يتميز بوجود القرص البصري. تم تنفيذ الخوارزمية أيضاً في FPGA للاستفادة من قوة المعالجة الموازية. تم اختبار الخوارزمية على قواعد بيانات DRIVE، STARE وRiaRetDB1 وتم مقارنتها مع الطرق الأخرى. لقد كانت دقة FSS 100%، 95.7%، و99.92% عند استخدام قواعد بيانات DRIVE، STARE وRiaRetDB1 على التوالي. وعلاوة على ذلك، فإن وقت التشغيل لتنفيذ FPGA كان 1.605 مللي / ثانية مع إجمالي استهلاك طاقة مقداره 121.818 ميغاواط.