Hardware implementation of Sobel edge detection system for blood cells images-based field programmable gate array

Ahmed Khazal Younis[1], Basma MohammedKamal Younis[1], Mohammed Sabah Jarjees[2]
[1]Department of Computer Technology Engineering, Engineering Technical College, Northern Technical University, Mosul, Iraq
[2]Department of Medical Instrumentation Technology Engineering, Engineering Technical College, Northern Technical University, Mosul, Iraq

Article Info

Article history:

Received Oct 17, 2021
Revised Jan 24, 2022
Accepted Feb 7, 2022

Keywords:

Blood cells images
Edge detection
FPGA
HLS
Medical image processing

ABSTRACT

The microscopic-blood image has been used to diagnose various diseases according to the morphological specifications of red and white blood cells. However, the manual analysis and procedures are not accurate due to the human error. Therefore, several studies conducted to find new techniques to perform this analysis using computer algorithms. The complexity of these algorithms led to thinking in simpler ways or to the hardware solutions. On the other hand, edge detection is a mathematical procedure that play an essential role in the field of medical image processing. It is considered as one of the foundations' processes for other procedures, such as the segmentation and the classification of the image. The Sobel filter is one of the conventional methods that is used to perform the edge detection process. It is based on finding the local contrast for the level of intensity of the image. This paper presents a proposed and a new method for detecting the edges of cells in the microscopic blood images using Sobel filter and its hardware implementation on the field programmable gate array (FPGA) chip. Three different techniques are proposed: MATLAB, OpenCV standard code, and FPGA customize code which give the best visual results, minimum timing results than the others.

Corresponding Author:

Ahmed Khazal Younis
Department of Computer Technology Engineering, Engineering Technical College
Northern Technical University
Mosul, Iraq
Email: ahmedkhazal@ntu.edu.iq

1. INTRODUCTION

Recently, the computer aided diagnosis (CAD) is widely spread which it enhances the understanding of the medical images. Edge detection (ED) plays the important roles in the field of image processing particularly in the segmentation and clustering procedures. It is based on calculating the changes of the intensity for the neighbour pixels in the image. ED has been widely used in the field of medical image processing [1]–[4]. In the paper, Anas *et al.* studied the performance of several edge detection methods on different X-ray images [1]. They stated that the best Gaussian noise removing algorithm in X-ray medical images was that median filter. Study by Chucherd introduced a generalized gradient vector flow method as an edge detection procedure to detect a breast cancer in ultrasound images [2]. Chucherd mentioned that the proposed method produced higher segmentation accuracy comparing with as Sobel and Canny segmentation methods. The neural network with the aid of conventional edge detection method was also used for segmentation blood vessels of retina [3]. Fuzzy geometrical approach has also been used to enhance the contrast of digital images for better classification [5]. In this study, the conventional edge detection filters such as Sobel and canny filters were used to extract the neural network algorithm features. Most of the state of art studies used the Sobel operator as a tool for edge detection and image segmentation [2], [3].

The Sobel filter is one of the most effective and popular methods in ED and image segmentation [6]. It is based to detect the vertical and horizontal edges in the image. There are several platforms, programs and languages that can be used to implement the Sobel operator such as MATLAB Software [7], field programmable gate array (FPGA) [7], [8], and OpenCV based FPGA [9], [10]. In terms of FPGA, there are various FPGA studies have implemented the Sobel edge-detection using different algorithms. Nausheen *et al.* proposed an implementation of the Sobel filter using very high speed integrated circuit (VHSIC) hardware description language (VHDL) [11], [12]. Moreover, Zhang *et al.* designed a Sobel operator using VHDL [13]. Menaka *et al.* also implemented an ED system based on FPGA [14]. All these mentioned studies aimed to faster the performing processes and to reduce the spatial complexity of the FPGA [11], [13], [14]. Sobel filter has been also used by Solanki and Tailor to design a video streaming architecture using high level synthesis [15]. In Solanki and Tailor study, zybo board (based also on ZYNQ 7000) was used, which has provided adequate peripherals for implementation. Although, the Sobel filter has been used in most of the edge detection studies, these studies used the standard codes for implementation of the Sobel filter. This study aims to parallel implementation of the Sobel operator based on the customization code of FPGA and to compare the results in terms of quality of edge detection, spatial complexity and time consuming with state-of-art studies.

The organization of this research is as follows: methods and material which highlights the theoretical aspect of Sobel filter mathematics. It also depicts the platform used in implementing of the hardware part of the system. Practical system section includes the practical part for implementing the proposed techniques with their hardware architectures that are used to detect the edges of the cells in blood images. Result and discussion section provides empirical results and performance analysis obtained by implementing methods used in the proceeding sections. Finally, conclusion section presents the conclusions of the study.

2. METHODS AND MATERIAL

The first part in this section mainly presents the edge detection techniques used in this paper. Generally, edge detection techniques are based on the concept of 2D spatial filtering operation. Spatial filtering operation refers to periodicity with which pixel values change. Just contrary to other techniques of the edge detection, the algorithms of first difference-based operator are the simplest to implement on hardware. Based on that, starting from the upper left side of the image, horizontal and vertical masks slide all over the image. Also, the masks move in the right direction on the image until they arrive at the end of each row. Then, they start again from the left most next row of the image [16]. To include all image pixels by the filter kernel, it is to a usual procedure to pad the image by the number of rows and columns depending on kernel dimensions. This had not been followed here because the main information dose not reside in the image edges but anywhere else. The second part in this section shows the materials that are used in implementation of this filter to detect the edges of the blood cells images.

2.1. Sobel edge detector

Sobel operator is a classical first order edge detection operator. It is used to compute the approximation of the image intensity function. At every point in the image, the result of the Sobel operator is the corresponding norm of this gradient vector. It makes use of two 3×3 masks convolved with the original image and calculates the image approximation. Both convolution masks are designed in such a way to detect the edge in both horizontal and vertical directions. The following (1) and (2) are used to calculate the G_x and G_y masks (the gradient component of each orientation G_x and G_y) [16].

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (1)$$

$$G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2)$$

The separate output gradients is combining in order to obtain the absolute magnitude of the gradient at every points of the image. In (3) illustrates the gradient magnitude |G|. The direction of the gradient is calculated by (4), while Figure 1 illustrates the flowchart of the steps followed in the design of this filters.

$$|G| = \sqrt{{G_x}^2 + {G_y}^2} \tag{3}$$

$$\theta = \text{arc}\tan\left(\frac{G_x}{G_y}\right) \tag{4}$$

Finally, binary images can be created by changing the threshold value. The threshold value is obtained by replacing each pixel with either a black or white one. The replacing of black pixel is dnoe when the image intensity does not exceed some fixed constant T (that is, I (i, j) < T), while white pixel whose intensity exceeds that constant.

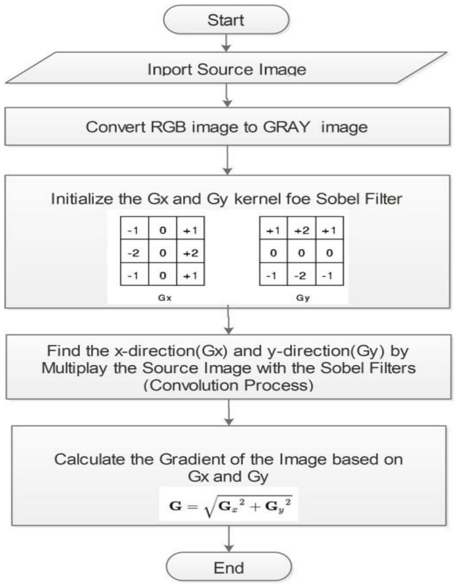


Figure 1. Flowchart of Sobel filter algorithm

2.2. Implementation platforms

The following platforms are used in this study:

- MATLAB: the first software platform used in this work is matrix laboratory (MATLAB). The image processing toolbox (IPT) is being the most relevant to this study [17]. MATLAB (version 2019) is used to design Sobel filter which is used to manipulate image processing and to produce edge detection for the original image.
- Xilinx OpenCV: this software platform is a template library optimized for FPGA high-level synthesis (HLS) which is used to create the image processing pipelines simply because it contains common algorithms such as: color conversion, image resizing, border and edge detection algorithms (Canny, Sobel), warp transformation, hog transform, matrix-matrix operations and others. It is completely possible to integrate Xilinx OpenCV to the Vivado HLS tool for accelerator-oriented projects. This library is used to build the first architecture that implement the Sobel edge detector [18]–[20].
- Vivado Design Suite: it is a software suite produced by Xilinx for synthesis and analysis of HDL designs. Xilinx Vivado 2018.3 is used to design the second architecture for detecting the edge of the image using Sobel filter [20].
- FPGA: The ZC702 board or Zynq-7000 XC7Z020 AP SoC is used to implement the proposed design of this study as a hardware platform. It comprises of a processing system (PS) organized around a dual-core ARM Cortex-A9 processor, and programmable logic (PL), that equivalent of traditional FPGA with additional new features such as integrated memory, variety of peripherals, and high-speed communications interfaces [21]–[24]. Both architectures designed (the first one using Xilinx OpenCV and the second one using Vivado Design Suite) are implemented using this hardware platform.

2.3. Proposed system

This section is divided into three parts: the first part deals with design the Sobel filter to detect edges using the MATLAB program, this technique is used for the purpose of comparing the results with the second and third methods. The second part presents our first architecture design using the OpenCV library from Xilinx, where the build-in functions are used in the design of Sobel filter. The third part, where the Sobel filter algorithm has been programmed using HLS to produce a new special architecture built to detect edges in blood cells of microscopic blood samples images.

2.3.1. Matlab based Sobel edge detection system

In order to evaluate the proposed hardware system, function simulations have been done using MATLAB package and the results have been examined before the designs have been implemented on the real device. The Sobel filter algorithm is programmed manually step by step without using build-in functions, this is done for fair comparison with the proposed architectures in next sections.

2.3.2. Xilinx OpenCV based Sobel edge detection system

An architecture is built to detect the edges of cells in blood microscopic images. The architecture is creaed by using OpenCV library built-in functions. Figure 2 illustrates the flowchart of the Sobel filter implementation process using OpenCV.

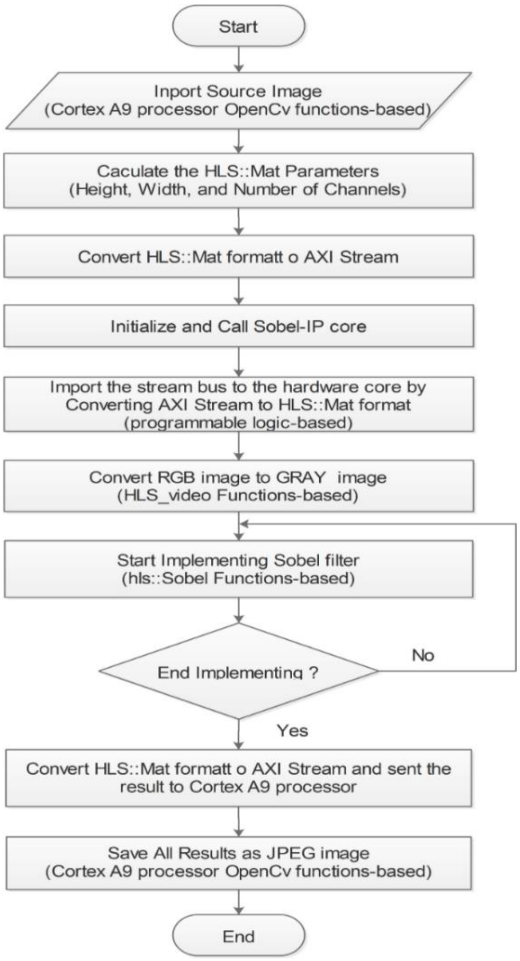


Figure 2. Flowchart of the Sobel filter implementation process using OpenCV

At this point, it is worth noting that the final gradient calculation after each step of the convolution process (G in (3) using square root) is done using the special addition function of this library (named *AddWeighted*) where different weights are given to images so that it gives a feeling of blending. Figure 3 shows the block diagram obtained from implementing of this algorithm on the ZYNQ hardware platform. As shown in this Figure 3 the edge detection system designed using OpenCV consists of the Sobel IP core which is designed inside PL to detect the edges of an input source image and the ZYNQ processor inside PS which is used to connect this core through automated X-ray inspection (AXI) bus to input output peripherals.

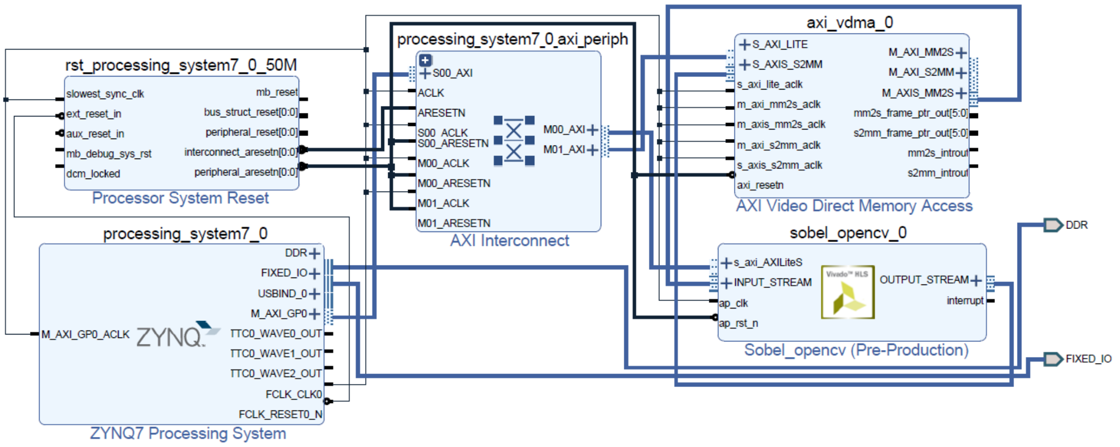


Figure 3. Implementation of the Sobel algorithm on the Zynq chip using OpenCV

2.3.3. Customized parallel HLS based Sobel edge detection system

This part of the research presents the customized parallel HLS algorithm. Figure 4 shows the Flowchart of the Sobel filter parallel implementation process using HLS. It can be noticed that the code is written to work in parallel to find the gradient for each direction of the image. The gradient component of each orientation G_x and G_y are calculated simultaneously then final G is calculated to reach our goal for finding the edges of the image faster. The parallelism process is illustrated in Figure 5, where the designed architecture consists of two IP core (named Sobel-Ix and Sobel-Iy) of the side PL. The other blocks of this design complete the work as in previous design; ZYNQ processor of PS, AXI buses and control.

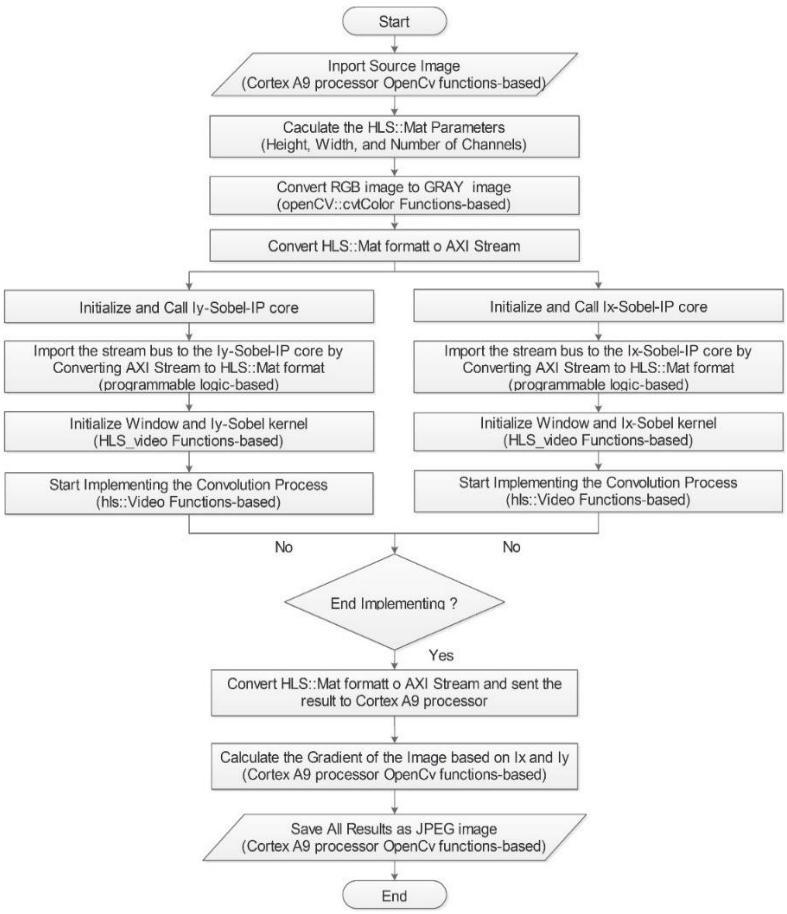


Figure 4. Flowchart of the Sobel filter parallel implementation process using HLS

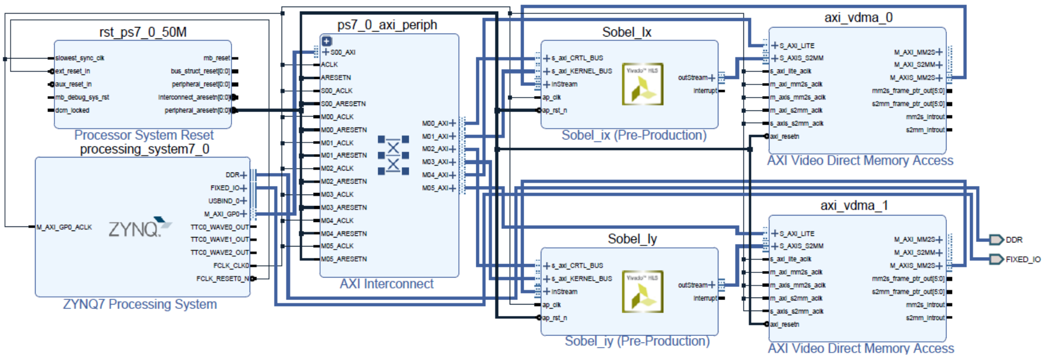


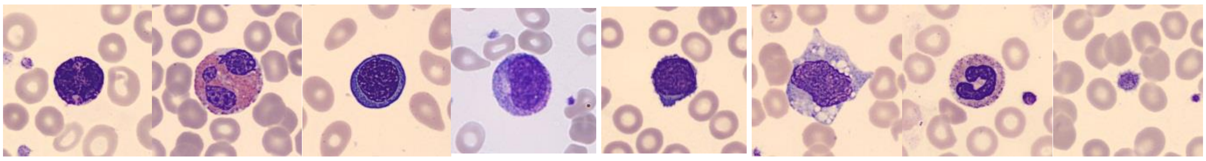
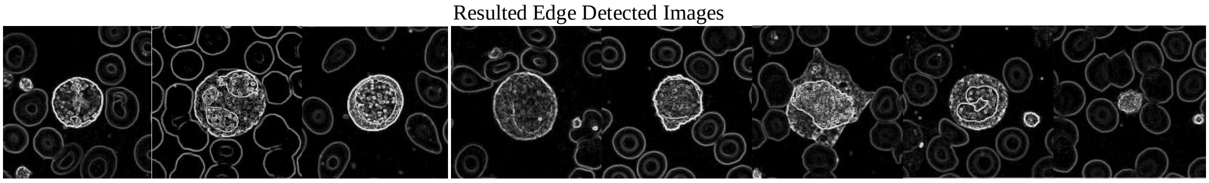
Figure 5. Implementation of the Sobel algorithm on the Zynq chip using Customized parallel HLS

3. RESULTS AND DISCUSSION

3.1. MATLAB based Sobel edge detection system results

Table 1 represents the results obtained using MATLAB for microscopic blood images. Where the upper images represent the original input image. The lower images used in this table are represent the resulted edge detected images using MATLAB method.

Table 1. MATLAB based Sobel Edge detection system results

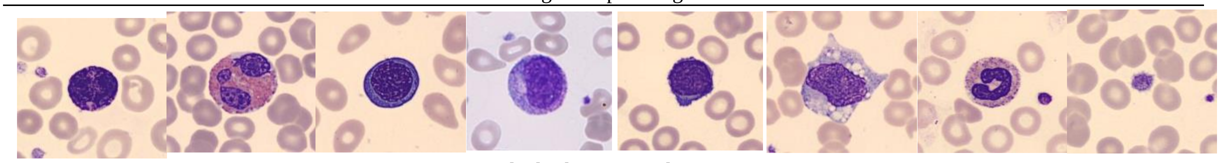
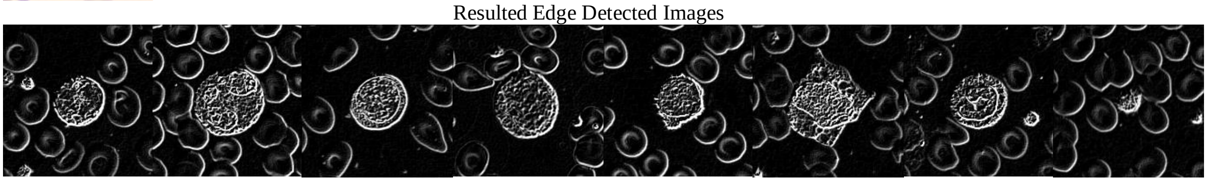
Original Input Image

Resulted Edge Detected Images


3.2. Xilinx OpenCV based Sobel edge detection system results

Table 2 represents the results that were obtained using OpenCV library for microscopic blood images. Where the upper images represent the original input image. Images in the lower part of the table represent the resulted edge detected images using Xilinx OpenCV method.

Figure 6 shows the overall system hardware utilization resources that used to implement the hardware Sobel Edge detection system. The examining resources are based on Zynq-7000 XC7Z020 board. It can be noticed that the maximum utilization is 18% go for the DSP block which is used to for convolution process.

Table 2. Xilinx OpenCV based Sobel Edge detection system results

Original Input Image

Resulted Edge Detected Images


Resource	Utilization	Available	Utilization %
LUT	6491	53200	12.20
LUTRAM	172	17400	0.99
FF	7411	106400	6.97
BRAM	7	140	5.00
DSP	39	220	17.73

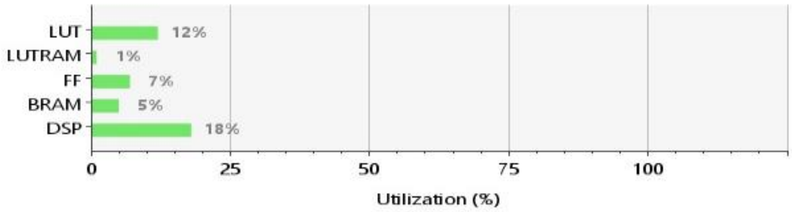


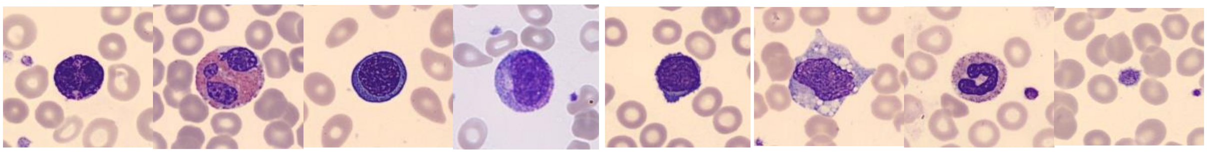
Figure 6. OpenCV based Sobel Edge detection system hardware utilization resources

3.3. Customized parallel HLS based Sobel edge detection system results

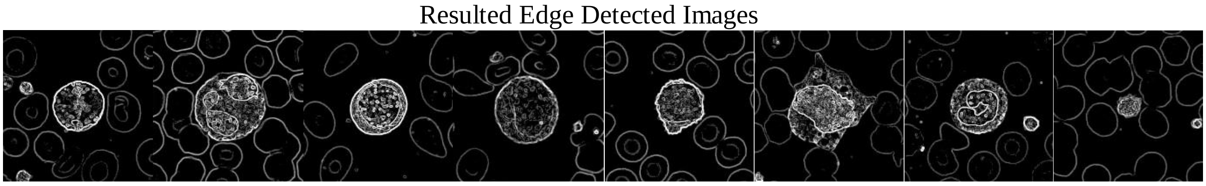
Table 3 represents the results obtained using customized parallel HLS algorithm. Where the upper images represent the original input image, while the lower images show the resulted edge detected images using customized parallel HLS method. Figure 7 shows the customized parallel HLS based Sobel Edge detection system hardware utilization resources of Zynq-7000 XC7Z020 board that are used to implement this design. By examining this figure, we can see that the maximum utilization is 14% go for the LUTs.

Table 3. The ccustomized parallel HLS based Sobel Edge detection system results

Original Input Image



Resulted Edge Detected Images



Resource	Utilization	Available	Utilization %
LUT	7331	53200	13.78
LUTRAM	274	17400	1.57
FF	7609	106400	7.15
BRAM	11	140	7.86

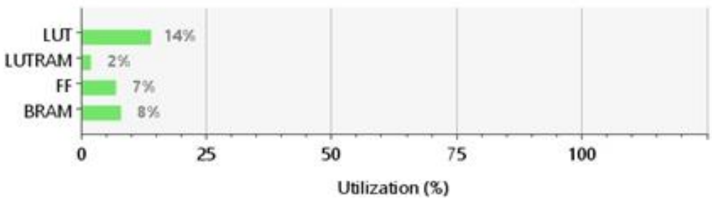


Figure 7. Customized parallel HLS based Sobel Edge detection system hardware utilization resources

3.4. Performance evaluation results

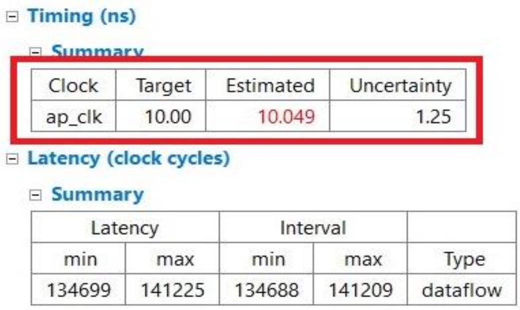
The performance evaluation of the proposed system has been performed by calculating the mean square error (MSE) and peak signal-to-noise ratio [25], [26]. According to the obtained mean square error (MSE) results demonstrated in Table 4, all techniques demonstrate very similar results, but MSE obtained from the OpenCV based method was better performance than the others. The same conclusion can be noticed in the Table 5 which represents the peak signal-to-noise ratio in decibels (dB). Finally, Figures 8 and 9 show the estimated time of the core using OpenCV and customized parallel HLS, respectively. It can be noticed that the built-in functions algorithm has higher estimated time, which is critical and leads to a delay of obtaining the results, compare to the estimated time of the proposed customized algorithm with no or smaller delay.

Table 4. The mean square error (MSE)× 10^4

Image No. / Technique	1	2	3	4	5	6	7	8
Matlab	3.7	2.9	3.9	3.7	3.6	3.4	3.7	3.7
OpenCV	3.6	2.7	3.8	3.6	3.6	3.4	3.7	3.7
Customized	3.9	2.9	4.0	3.8	3.8	3.6	3.9	3.9

Table 5. The peak signal-to-noise ratio in decibels (dB)

Image No. / Technique	1	2	3	4	5	6	7	8
Matlab	2.4	3.4	2.1	2.4	2.5	2.7	2.3	2.4
OpenCV	2.5	3.7	2.2	2.5	2.5	2.7	2.4	2.5
Customized	2.2	3.4	2.0	2.2	2.3	2.4	2.1	2.1

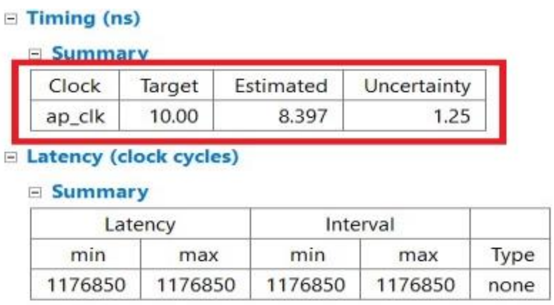

Timing (ns)
Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	10.049	1.25

Latency (clock cycles)
Summary

Latency		Interval		
min	max	min	max	Type
134699	141225	134688	141209	dataflow

Figure 8. OpenCV based Sobel Edge detection system performance estimation


Timing (ns)
Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.397	1.25

Latency (clock cycles)
Summary

Latency		Interval		
min	max	min	max	Type
1176850	1176850	1176850	1176850	none

Figure 9. Customized parallel HLS based Sobel Edge detection system performance estimation

4. CONCLUSION

The microscopic blood images play the important roles to diagnose different biological diseases. However, the diagnostic error can be produced due to human error. Therefore, this study aimed to implement an automated edge detection system for microscopic blood images. It proposed a new method of parallel implementation for Sobel operator based on the customization code of FPGA. It is observed that the FPGA customized codes give the best visual and less timing results than the ready to used algorithms such as MATLAB, using OpenCV standard code. The proposed system can not only apply to the microscopic blood images but also to other medical images, such as X-ray, computer tomography and magnetic resonance images.

REFERENCES

[1] R. Anas, H. A. Elhadi, and E. S. Ali, "Impact of edge detection algorithms in medical image processing," *World Scientific News*, vol. 118, no. January, pp. 129–143, 2019, Accessed: Feb. 08, 2022. [Online]. Available: www.worldscientificnews.com.
[2] S. Chucherd, "Edge detection of medical image processing using vector field analysis," in *2014 11th Int. Joint Conf. on Computer Science and Software Engineering: "Human Factors in Computer Science and Software Engineering" - e-Science and High Performance Computing: eHPC, JCSSE 2014*, May 2014, pp. 58–63, doi: 10.1109/JCSSE.2014.6841842.
[3] B. Saha Tchinda, D. Tchiotsop, M. Noubom, V. Louis-Dorr, and D. Wolf, "Retinal blood vessels segmentation using classical edge detection filters and the neural network," *Informatics in Medicine Unlocked*, vol. 23, p. 100521, 2021, doi: 10.1016/j.imu.2021.100521.
[4] D. Alhelal, A. K. Younis, and R. H. A. Al-Mallah, "Detection of brain stroke in the MRI image using FPGA," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 14, no. 5, pp. 1307–1315, Aug. 2021, doi: 10.12928/TELKOMNIKA.v19i4.18988.
[5] IEEE Signal Processing Society. Malaysia Chapter and Institute of Electrical and Electronics Engineers, *IEEE 2015 International Conference on Signal and Image Processing Applications (ICSIPA) : proceedings : 19-21 October 2015, Pullman Bangsar, Kuala Lumpur, Malaysia*. 2015.
[6] T. Sanida, A. Sideris, and M. Dasygenis, "A heterogeneous implementation of the Sobel edge detection filter using openCL," *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, Sep. 2020, doi: 10.1109/MOCAST49295.2020.9200249.
[7] A. Pujare, P. Sawant, H. Sharma, and K. Pichhode, "Hardware implementation of Sobel edge detection algorithm," *ITM Web of Conferences*, vol. 32, p. 03051, 2020, doi: 10.1051/itmconf/20203203051.
[8] R. Mehra and R. Verma, "Area efficient FPGA implementation of Sobel edge detector for image processing applications," *International Journal of Computer Applications*, vol. 56, no. 16, pp. 7–11, Oct. 2012, doi: 10.5120/8973-3086.
[9] G. Ravivarma, K. Gavaskar, D. Malathi, K. G. Asha, B. Ashok, and S. Aarthi, "Implementation of Sobel operator based image edge detection on FPGA," *Materials Today: Proceedings*, vol. 45, pp. 2401–2407, 2021, doi: 10.1016/j.matpr.2020.10.825.
[10] I. Yasri, N. H. Hamid, and V. V. Yap, "Performance analysis of FPGA based Sobel edge detection operator," *2008 International Conference on Electronic Design*, Dec. 2008, doi: 10.1109/ICED.2008.4786751.
[11] N. Nausheen, A. Seal, P. Khanna, and S. Halder, "A FPGA based implementation of Sobel edge detection," *Microprocessors and Microsystems*, vol. 56, pp. 84–91, Feb. 2018, doi: 10.1016/j.micpro.2017.10.011.
[12] A. A. K. Rawther and L. Mary, "The design of low power Sobel edge detection in FPGA," in *AIP Conference Proceedings*, 2020, vol. 2222, doi: 10.1063/5.0004201.
[13] K. Zhang, Y. Zhang, P. Wang, Y. Tian, and J. Yang, "An improved sobel edge algorithm and FPGA implementation," *Procedia Computer Science*, vol. 131, pp. 243–248, 2018, doi: 10.1016/j.procs.2018.04.209.
[14] D. R. Menaka, D. R. Janarthanan, and D. K. Deeba, "FPGA implementation of low power and high speed image edge detection algorithm," *Microprocessors and Microsystems*, vol. 75, p. 103053, Jun. 2020, doi: 10.1016/j.micpro.2020.103053.
[15] N. Solanki and N. Tailor, "Sobel edge detection on zynq based architecture with vivado," *International Research Journal of Engineering and Technology*, pp. 1068–1070, 2019, Accessed: Feb. 08, 2022. [Online]. Available: www.irjet.net.
[16] H. A. T. Abdullah and B. M. K. Younis, "FPGA based bone fracture detector," *IOP Conference Series: Materials Science and Engineering*, vol. 745, no. 1, p. 12052, Feb. 2020, doi: 10.1088/1757-899X/745/1/012052.
[17] G. K. Ijemaru *et al.*, "Image processing system using matlab-based analytics," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2566–2577, Oct. 2021, doi: 10.11591/eei.v10i5.3160.
[18] M. A. Dávila-Guzmán, R. G. Tejero, M. Villarroya-Gaudó, D. S. Gracia, L. Kalms, and D. Göhringer, "A Cross-platform openVX library for FPGA accelerators," in *Proceedings-29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2021*, Mar. 2021, pp. 75–83, doi: 10.1109/PDP52278.2021.00020.
[19] X. Inc., "Xilinx openCV user guide (v2018.2)," vol. 1233, pp. 1–302, 2018, Accessed: Feb. 08, 2022. [Online]. Available: www.xilinx.com.
[20] T. Li, "Accelerating OpenCV applications with Zynq-7000 all programmable soc using Vivado HLS video librarie," *Xilinx Wiki*, vol. 1167, p. 1, 2013, Accessed: Feb. 08, 2022. [Online]. Available: http://www.wiki.xilinx.com/XAPP1167.
[21] B. M. K. Younis and A. K. Younis, "Hardware accelerator for anti-aliasing Wu's line algorithm using FPGA," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 672–682, Apr. 2021, doi: 10.12928/TELKOMNIKA.v19i2.18158.
[22] Xilinx and Inc, "Zynq-7000 All programmable soc data sheet: overview (DS190)," 2012, Accessed: Feb. 08, 2022. [Online]. Available: www.xilinx.com.
[23] Xilinx Inc. and Xilinx, *Vivado Design Suite User Guide: Using the Vivado IDE*. 2020.
[24] X. Inc. and Xilinx, "Vivado design suite user guide: implementation," *UG904*, vol. 2021, no. 1, pp. 1–208, 2021, Accessed: Feb. 08, 2022. [Online]. Available: www.xilinx.com.
[25] B. Magnier, "Edge detection: a review of dissimilarity evaluations and a proposed normalized measure," *Multimedia Tools and Applications*, vol. 77, no. 8, pp. 9489–9533, Sep. 2018, doi: 10.1007/s11042-017-5127-6.
[26] H. Abdulrahman, B. Magnier, and P. Montesinos, "From contours to ground truth: How to evaluate edge detectors by filtering," *Journal of WSCG*, vol. 25, no. 2, pp. 133–142, 2017.

BIOGRAPHIES OF AUTHORS

Ahmed Khazal Younis works at Northern Technical University/Engineering Technical College-Mosul-Iraq/Department of Computer Technology Engineering. He Completed his undergraduate education (1998-2002), Master degree (2005-2008) at Computer Technical Engineering Department in Technical College/North Technical University and Ph.D. degree in computer engineering/embedded system design from Yasar University in Izmir/Turkey in 2019. His research interests include: Neural networks, Neuro-fuzzy system, Microcontrollers, and FPGA devices. He can be contacted at email: ahmedkhazal@ntu.edu.iq.

Basma MohammedKamal Younis works at Northern Technical University/Engineering Technical College-Mosul-Iraq/Department of Computer Technology Engineering. She received B.Sc. degree in electrical engineering/Electronic and Communication Engineering from University of Mosul-Iraq in 1995, M.Sc. degree in computer engineering from University of Mosul-Iraq in 2000 and Ph.D. degree in computer engineering/Microprocessors Architecture from University of Mosul-Iraq in 2007. Her research interests and activity are Embedded System Design using FPGA and Computer Graphics. Now, she teaches Advanced Digital Electronics, Microprocessors Based Systems and Embedded Systems for postgraduate and undergraduate students. She can be contacted at email: dr_bmk@ntu.edu.iq.

Mohammed S. Jarjees PhD in Biomedical Engineering, University of Glasgow, Glasgow, United Kingdom, 2017. Technical Master Degree in Medical Instrumentation Technology Engineering, Technical Engineering College of Mosul, Mosul, IRAQ. 2006. B. Eng. In medical Instrumentation Technology Engineering 2002. Head of Medical Instrumentation Techniques Engineering Department, Technical Engineering. College of Mosul, Northern Technical University, Mosul, IRAQ 2017-Now. He can be contacted at email: mohammed.s.jarjees@ntu.edu.iq.