Hindawi
IET Circuits, Devices & Systems
Volume 2023, Article ID 8820773, 12 pages
https://doi.org/10.1049/2023/8820773

Research Article

Implementation of Image Enhancement and Edge Detection Algorithm on Diabetic Retinopathy (DR) Image Using FPGA

Mumtahina Orthy,[1] **Sheikh Md. Rabiul Islam**,[1] **Faijah Rashid**,[1] **and Md. Asif Hasan**[2]

[1]*Department of Electronics and Communication Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh*
[2]*Department of Biomedical Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh*

Correspondence should be addressed to Sheikh Md. Rabiul Islam; robi@ece.kuet.ac.bd

Received 9 June 2023; Revised 24 October 2023; Accepted 1 November 2023; Published 11 December 2023

Academic Editor: Nandha Kumar Thulasiraman

Diabetic retinopathy (DR) is an ocular ailment that may lead to loss of vision and eventual blindness among individuals diagnosed with diabetes. The blood vessels of the retina, a layer of light-sensitive tissue located at the posterior aspect of the ocular globe, are adversely impacted. The identification of DR entails the utilization of retinal fundus images. The detection of any form of abnormality in the eye through raw fundus images poses a significant challenge for medical practitioners. Hence, it is imperative to engage in the processing of fundus images. This paper delineates several image processing techniques for DR images, including but not limited to, manipulation of brightness levels, application of negative transformation, and utilization of threshold operations. It focuses on elucidating the enhancement techniques that pertain to DR images, which aim to optimize the visual quality of said images in order to facilitate more facile disease detection. The process of detecting edges within DR images is also executed by Sobel edge detection algorithm. In order to successfully execute the aforementioned algorithms, expedient and contemporaneous systems are favored to account for the intricacies of the image processing calculations. The exclusive utilization of software techniques in order to fulfill the prerequisites of advanced algorithms presents a significant challenge, owing to the multifarious processes that are involved in their computation, coupled with an exigent requirement for high processing speeds. The proposed model is utilized to articulate a proficient model for the design and execution of field programable gate array (FPGA)-based image enhancement processes along with the Sobel edge detection algorithm upon DR images. Finally, a Internet Protocol chip is developed that can combine multiple image enhancement operations into a single framework with less complexity.

1. Introduction

Image processing plays a vital role in medical contexts, facilitating the timely identification, interpretation, and remediation of various disorders. The objective of the given undertaking is to systematically carry out processing and analysis of the retrieved images. The principal aim of this procedure is to isolate, quantify, and elucidate the data pertaining to the structures under investigation. This methodological approach is necessary to glean a comprehensive understanding of the anatomy as well as the functionality of the organ systems being scrutinized. With regard to the context in which it is applied, the essential objective of utilizing enhancement operations in image processing is to attain an outcome that surpasses the quality of the original image. Various techniques are utilized by image enhancement methods to enhance the visual aesthetics of an image. Concurrently, the concept of image enhancement entails a course of action intended to augment the visual quality of an image through the amplification of select attributes or by decreasing the degree of indistinctness between disparate regions of that image.

Spatial domain techniques are primarily concerned with the direct manipulation of individual image pixels. In order to achieve the intended enhancement, modifications are made to the values of individual pixels. The present work pertains to the spatial domain category within the domain of image processing. Such category encompasses diverse operations including, but not limited to, image negatives, gray level transformation, threshold operation, and brightness control.

The process of edge detection is a customary methodology employed in the realm of medical image analysis, serving as a valuable tool for the recognition and identification of human organs. Edge detection is capable of pinpointing the exact locations where shadows are present within an image, in addition to detecting any distinguishable alternations in image luminosity that may have been induced by extraneous factors. Presently, the predominant method employed for image edge detection is through the utilization of computer software. This approach yields a superior outcome, nevertheless, it is dependent upon the reading and execution of instructions by the computer. A crucial concern that arises when working with voluminous image data is the processing speed. Image edge detection in hardware is capable of performing data processing tasks at a faster rate, as it is not dependent on instruction operations. A hardware-based approach has been employed to implement image edge detection, yielding the present outcome. The Sobel edge detection algorithm [1] is a prevalent technique employed in processing image edge detection within the field of computer vision. The conventional Sobel edge detection method, which relies on personal computer implementation, has been observed to exhibit a limited capacity for processing images at a satisfactory rate while concurrently delivering satisfactory gradient levels. As a result, its suitability for application to modern, intricate images is significantly impaired. Field programable gate arrays (FPGAs) [2] exhibit remarkable parallel processing capabilities and exceptional reliability, attributable to their extensive employment and integration. Hence, the utilization of FPGA technology is viable for the purpose of executing image edge detection.

Diabetic retinopathy (DR) is an aftermath of diabetes, characterized by detriment to the posterior portion of the eye, specifically the retina, caused by heightened levels of glucose in the blood. In the event that it remains untreated and misdiagnosed, consequential vision impairment may occur [3]. The progression of DR, however, commonly requires several years before reaching a critical stage that poses a significant threat to an individual's ability to see. In the incipient stages, DR tends to be asymptomatic. Retinal fundus images are commonly employed in clinical settings to detect and evaluate the pathology of DR. Diagnosis of DR from raw retinal fundus images presents a significant challenge for physicians. In the proposed approach of implementation of image enhancement and edge detection on DR image, the main objective is to develop a Internet Protocol (IP) chip for image enhancement that can combine multiple operations of image enhancement to solve the problem of erroneous biomedical appliances. Consequently, digital radiography images necessitate postprocessing procedures. The processing of raw retinal fundus images poses a considerable challenge for machine learning algorithms due to the relatively protracted processing time. The most contemporary means of improving image quality and identifying edges in digital radiography images have been attributed to the FPGA approach. This approach can be customized utilizing the Verilog programing language to cater to a range of applications and boasts swift processing capabilities, allowing for real-time imaging. Furthermore, its employment is particularly useful in identifying abnormalities within the image data. Consequently, the present study employs field-programable gate array-based methodologies for image enhancement and Sobel edge detection algorithms to preprocess the unprocessed retinal fundus images.

2. Literature Review

We discovered what other researchers had done in this field while researching our own. Cavinato et al. [4] outlined the preprocessing techniques implemented for retinal fundus imagery and the corresponding feature extraction methods for feature prioritization. This encompasses optic disc elimination, exudate removal, green channel isolation, hemorrhage identification, and contrast enhancement.

The utilization of machine learning algorithms is employed for the purpose of preprocessing diabetic retinal fundus images and subsequently extracting notable features. The digital image processing software package in MATLAB was utilized to carry out preprocessing methodologies such as the extraction of the green channel, equalization of the histogram, and scaling.

Sisodia et al.'s [1] study explicates an investigative study on a Sobel edge detection structure implemented on a FPGA. As hardware devices possess greater potential for parallelism compared to software, the decision was made to utilize a hardware-based implementation of the Sobel edge detection technique. Conversely, the Sobel edge detection algorithm is adept at functioning under conditions of elevated noise with decreased decline in performance. Drawing upon the aforementioned methodologies, a concise investigation was also undertaken. The main trade off issues in the aforementioned literature reviews are inefficiencies in the detection algorithm, time consumption, and design complexity. Higher amount of power consumption both during process execution and postprocessing is another important phenomenon to develop a more effective IP chip in the field of biomedical appliances.

In Yildirim and Cinar's [2] study, the validation and examination of FPGAs involve the application of a variety of prevalent edge detection methodologies as documented in the literature, including the Sobel, Prewitt, and Canny algorithms. The system generator digital signal processing (DSP) blocks produced by Xilinx have the capability of interfacing with the software packages of MATLAB or Simulink, enabling the creation of necessary program files to be utilized in a FPGA setting.

Hasan et al.'s [3] study presents a review of recent research endeavors pertaining to the application of image processing algorithms for the detection of DR features. Numerous image processing algorithms have been deliberated for the detection of DR. The outcomes of various techniques for detecting lesions in DR have been documented as well. The aforementioned outcomes are employed in the assessment of techniques pertaining to image processing. In the present study, an examination was conducted into

the implementation of the Sobel edge detection algorithm on a digital radiography image. Additionally, the execution of the edge detection algorithm via a FPGA was also investigated.

Cavinato et al.'s [4] study dedicated to the implementation of image enhancement techniques, encompassing brightness control, contrast correction, and histogram equalization, utilizing FPGA technology. FPGA has emerged as a viable alternative for high-performance digital image or signal processing applications. The present study employs the MATLAB Simulink programing tool for the purpose of enacting proficient image-enhancing algorithms upon a FPGA.

Khan's [5] study demonstrated the appropriateness of FPGAs in the realm of biomedical applications through the provision of a case study which encompassed the development of a vascular segmentation algorithm. When compared to the software-based approach, the empirical findings obtained from the analysis of the DRIVE and STARE databases reveal notable improvements in regard to both operational speed and energy efficiency. The hardware proposal exhibits superior performance in comparison to previous works while maintaining the overall sensitivity measurements and accuracy of the system.

3. Implementation Tool

3.1. Field Programable Gate Array. A FPGA is a type of semiconductor device that consists of programable logic blocks capable of executing a specified set of operations. The programable logic blocks are interconnected through employment of an interconnect matrix. The interconnects establish connections between the various logic blocks and enable the propagation of signals throughout the semiconductor substrate. The said structure comprises logic blocks, interconnects, and input/output (I/O) blocks, which collectively facilitate an arrangement in the form of a 2D array. This arrangement enables the connection of the structure to the input and output signals. A logic block comprises three primary elements, namely a look-up table (LUT), a flip–flop (FF), and a multiplexer. FPGAs have the distinguishing characteristic of being reprogramable, allowing for their functionality to be modified by loading new code onto the FPGA [7–15].

3.2. Xilinx Vivado. The Vivado plan suite developed by Xilinx is a comprehensive software suite utilized for the integration and evaluation of hardware description language (HDL) designs. This software offers additional functionality for the development of systems-on-a-chip and high-level synthesis, whereby it holds potential to serve as a viable replacement for Xilinx ISE. Vivado presents a versatile platform for developers to conduct a range of design tasks, including synthesis of plans, timing analysis, evaluation of register transfer level (RTL) charts, emulation of design responses to various alterations, and collaboration with systems engineers to construct the target device. Vivado constitutes a planning software platform specifically designed for Xilinx FPGAs and is interdependently linked with the layouts of said chips. Consequently, it is unsuitable for deployment with FPGAs furnished by alternative vendors [13–19]

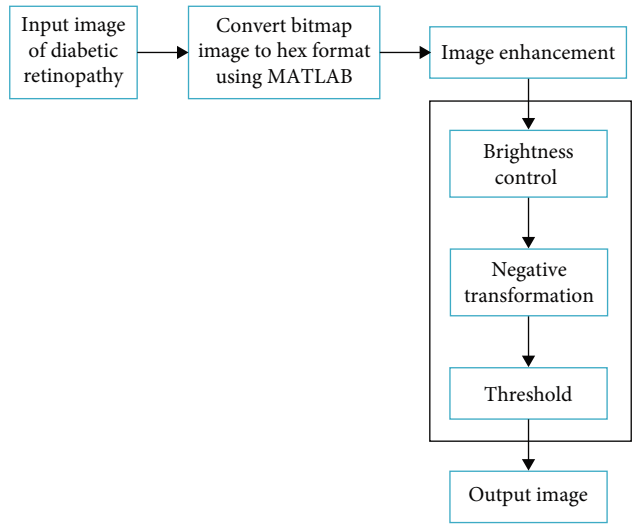


Figure 1: Steps of image enhancement [6].

3.3. Verilog Code for Image Processing. The objective of this FPGA undertaking is to explicate the intricacies involved in the manipulation of an image utilizing Verilog [13]. The comprehensive process entails reading an input DR image in Verilog and processing the image, culminating in the subsequent composition of the processed result in Verilog to yield a refined image. It is not possible for Verilog to directly interpret pictorial data. In order to access the .bmp image within a Verilog environment, it is necessary to convert the image from its bitmap format to a hexadecimal format. The conversion of a bitmap image into a hex file may necessitate the utilization of a MATLAB programing code. The study of the hexadecimal information record for image processing in Verilog involves the utilization of the command $readmemh, or alternatively, $readmemb in the case where the image data are available in binary content format.

4. Methodology

4.1. Data Collection. The data collection was taken among DR image on diabetes mellitus from Khulna BNSB Eye Hospital, Bangladesh [3].

4.2. Image Enhancement Process. Figure 1 shows the steps of image enhancement that has been illustrated throughout the proposed approach. To begin with the raw input image of DR has been converted to hexadecimal format using MATLAB. After getting the required hexadecimal values, the image enhancement operations have been conducted that include brightness control, negative transformation, and threshold operation. Finally, after the adequate image enhancement operations mentioned in Figure 1, the desired output image was found. In this research, the image enhancement operations mentioned in Figure 1 have been delineated in the following sections.

4.2.1. Brightness Control. Brightness adjustment refers to the application of a technique that involves augmenting or lowering the intensity level of individual pixels within an image by way of the addition or subtraction of a constant value. This technique is commonly used to remedy issues of overly bright or insufficient brightness within an image. The luminosity of a dim visual representation may be easily enhanced through the incorporation of a constant quantity to every individual pixel. The addition operation applied with a constant factor will result in a rightward shift of the histogram. In contrast, the operation of subtraction will result in a shift of the histogram toward the lower end of the luminance spectrum. In the context of image manipulation, it is imperative to exercise caution while applying methods aimed at modifying an image's brightness. It is essential to undertake a meticulous adjustment of the constant, ensuring that the full spectrum of intensity values lies within the prescribed range of 0–255. In the event that the ultimate value of a pixel exceeds 255, the data become irretrievable.

This is how the algorithm works:

$$P(t) + g, \text{if } P(t) + g \leq 255, \tag{1}$$

$$K(t) = P(t) + g, \tag{2}$$

$$P(r) - g, \text{if } P(r) + g \geq 0, \tag{3}$$

$$K(r) = P(r) - g, \tag{4}$$

where g is a constant value ($g > 0$), $P(t)$ is the intensity level of input pixel (t), and $K(t)$ is the intensity level of output pixel (t) after the brightness adjustment.

4.2.2. Negative Transformation. The following function is used to obtain the negative transformation of the digital image:

$$S = L - 1 - r, \tag{5}$$

where L is the maximum intensity levels, S is the output intensity, and r is the input intensity at any point (x, y) in the image. The goal is to reverse the order from white to black or vice versa, so that the transformed image's intensity drops as the input intensity increases.

4.2.3. Threshold Operation. The threshold of a photo refers to the transformation of all pixels into only two values. It is a sort of quantization in which the pixel values are compared to a predetermined threshold value, g, which is generally a constant. All pixel values must be mapped to one of two fixed intensity values, g_0 or g_1, with $0 < g_{th} < g_{max}$ using the given threshold function f threshold (g). The threshold operation is carried out by scanning each pixel value from the input image and replacing it at the destination. Using $g_0 = 0$ and $g_1 = 255$ as the output image within the Verilog testing code, the threshold value can be set.

4.3. Implementation of Image Enhancement Process in FPGA. Figure 2 is the representation of the conceptual framework of

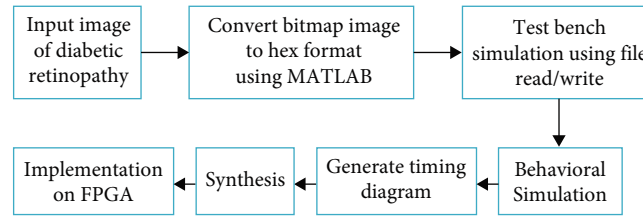


FIGURE 2: Conceptual framework of image enhancement process in FPGA [7].

image enhancement process in FPGA that includes multiple steps. It has been depicted in Figure 2 that the digital radiography image was preserved in the bitmap (.bmp) file format initially. In order to facilitate the interpretation of the image in Verilog, it is necessary to convert the bitmap representation to its corresponding hexadecimal format. The conversion of the .bmp image into a hexadecimal file is performed through the utilization of MATLAB code. The input image has a resolution of 768×512 pixels and is stored in a hexadecimal file format that includes data on the red (R), green (G), and blue (B) components of the respective pixel values in the bitmap image. Then, in Figure 2, it has been mentioned that a Verilog-based test bench along with file read and write operation is required for simulation and timing diagram generation purpose. This Verilog code comprises of a total of three modules. The initial module serves the purpose of image retrieval and preprocessing. The Verilog language employs the $readmemb or $readmemh command to access and interpret hexadecimal image files. The image data in RGB format were stored in memory and subsequently utilized for further analysis subsequent to the retrieval of the image's hexadecimal file. The operation pertaining to image processing was elected within the file named "parameter.v" utilizing proper approach. The ultimate component involved in producing the .bmp image entailed composing its written format. The header data pertaining to the .bmp image holds significant pertinence. In the absence of header data, the visual depiction cannot be presented accurately. The $fwrite statement is employed within the Verilog HDL to facilitate the outputting of data to a file. An imperative is to author a test bench code for the purpose of verifying the efficacy of image processing procedures, as well as to visually evaluate the outcome of each respective processing technique. The duration of the simulation is contingent upon the overall duration necessitated for the reading, processing, and writing of the resultant image. Upon execution of a simulation, an evaluation of the temporal sequence can be visually analyzed via a timing diagram, and validation of the resulting output can be conducted correspondingly. Upon obtaining adequate results from the simulation process, the design or code may be subjected to synthesis for eventual implementation on a FPGA. This allows for synchronous observation of operations that enable real-time enhancement of images.

4.4. Flowchart of Sobel Edge Detection Algorithm. Figure 3 shows the flowchart of the Sobel edge detection algorithm implemented on a DR image. It is a gradient-based edge detection algorithm which demonstrates the edges using

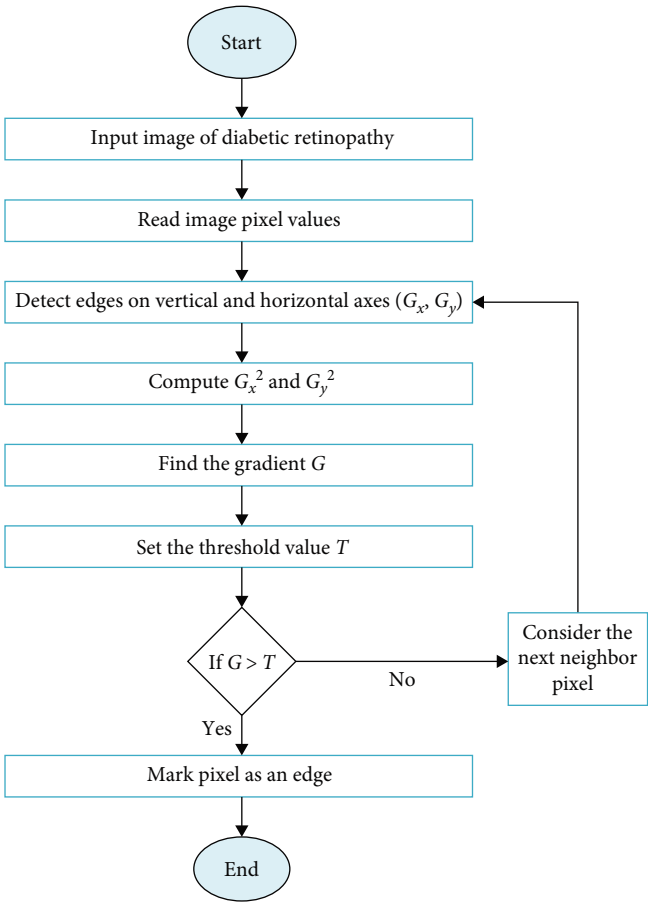


Figure 3: Flowchart of Sobel edge detection algorithm.

vertical mask (VM) and horizontal mask (HM). One mask is the transpose of the other as follows:

$$\mathrm{VM} = [-1\ -2\ -1\ 0\ 0\ 0\ 1\ 2\ 1];\ \ \mathrm{HM} = [-1\ 0\ 1\ -2\ 0\ 2\ -1\ 0\ 1]. \tag{6}$$

For the convolution process, the DR image is scanned from top to bottom and left to right using HM and VM independently. Convolution is the procedure of multiplying each pixel value in an image with its local neighbors and then weighting the result with the mask. Let $P = 3 \times 3$ represents a subwindow of an-image matrix as follows:

$$P = [P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8]. \tag{7}$$

G_x is the horizontal gradient and G_y is the vertical gradient. We have considered for finding the pixel value P_4 by the equations as follows:

$$G_x \approx f_1 - f_2, \tag{8}$$

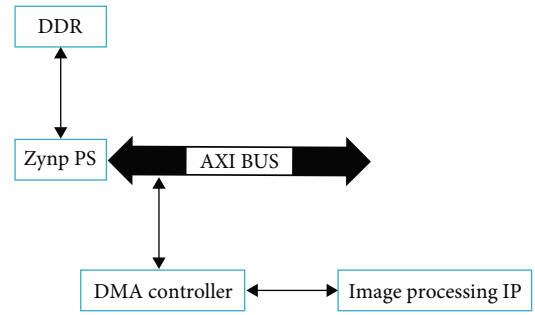


Figure 4: Block diagram of the proposed system architecture.

where:

$$f_1 = P_6 + 2P_7 + P_8, f_2 = P_0 + 2P_1 + P_2. \tag{9}$$

$$G_x \approx f_3 - f_4, \tag{10}$$

where:

$$f_3 = P_2 + 2P_5 + P_8, f_4 = P_0 + 2P_3 + P_6. \tag{11}$$

The resultant gradient of the center pixel of the subwindow is given by the following equation:

$$G = \sqrt{G_x^2 + G_y^2}, \tag{12}$$

which is convoluted with the HM and VM. Let the horizontal gradient be.

The gradient is compared to a predetermined threshold value to determine whether it is an edge pixel or not. The pixel is deemed an edge pixel if its gradient is bigger than the threshold value (denoted as 1), otherwise, it is handled as a nonedge pixel (denoted as 0). The following equation illustrates this:

$$\text{Pixel edge} = \begin{cases} 1, \text{ if } G > T \\ 0, \text{otherwise} \end{cases}. \tag{13}$$

4.5. System Architecture for Sobel Edge Detection Algorithm. The overall system architecture is shown in Figure 4. The description of this system is given below.

4.5.1. Verilog Code for the Proposed Architecture. Initially, the digital radiography image is required to undergo grayscale conversion via the utilization of MATLAB. The resolution of the image is 512 pixels in width and 512 pixels in height.

4.5.2. Line Buffer module. Four line buffers are created for better performance, and three pixels from each line buffer

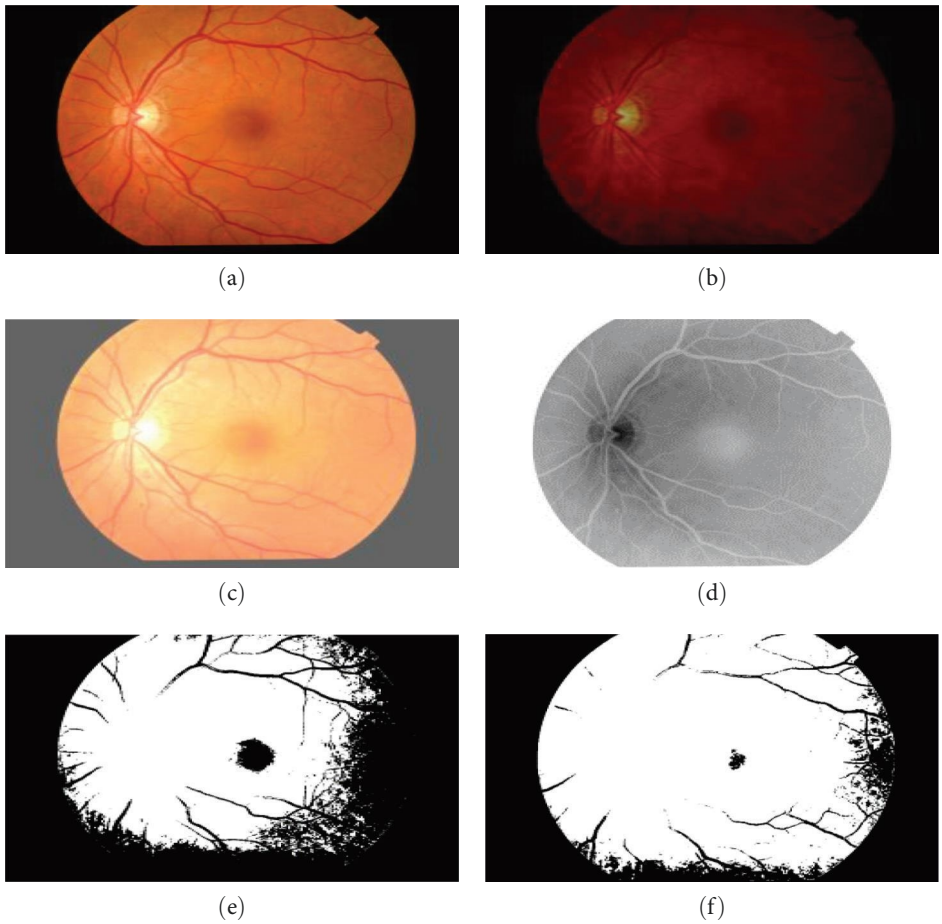


FIGURE 5: (a) Original DR image. (b) After reducing the brightness. (c) After increasing the brightness. (d) After inversion or negative transformation. (e) After threshold operation when threshold value is greater than 90 and (f) when threshold value is 80.

will be read in one shot. So, the size of each line buffer will be 24 bits.

4.5.3. MAC (Multiply and Accumulate) Module. The pixels form line buffer will be multiplied by the Sobel kernel and then added together.

4.5.4. Image Control IP. The control logic pertaining to image processing is explicated in this manuscript. At this juncture, four line buffers are instantiated. This will facilitate the assignment of incoming data from the external environment to a specific line buffer through a determination process. Upon the completion of the initial filling phase of the first three line buffers, the convolutional process commences, whereby pixels are concurrently extracted from said line buffers. Upon the culmination of the convolution process utilizing the initial three line buffers, the fourth buffer shall subsequently be replenished, thereby facilitating the streaming of the forthcoming pixel for convolution purposes.

4.5.5. Image Processing Top Module. The current phase of the project will involve the integration of the three predesigned modules, culminating in the finalization of the IP module dedicated to image processing. A first-in-first-out (FIFO) buffer is employed as an output mechanism for the purpose of managing any disparities that may arise between the input and output streams.

4.5.6. Test Bench. The image will be subjected to a process of reading, and subsequently transmitted to the Internet Protocol (IP). Upon completion of the image data processing, input will be received from the IP, subsequently resulting in the generation of an output. The instantiation of the top module for image processing is initiated at this juncture, and a new file is generated for the storage of the processed image.

5. Result Analysis

5.1. Simulated Output for Image Enhancement Operation. The original image obtained from the dataset may be observed in Figure 5, representing the DR dataset. The dimensions of the image in question are 768 pixels in width and 512 pixels in height. Prior to its enhancement through various algorithms, the image underwent conversion into a hexadecimal file format. The aforementioned transformation was executed utilizing the software program, MATLAB. Subsequent to its conversion into hexadecimal format, the file is subsequently subjected to inspection for additional manipulation within Xilinx Vivado.

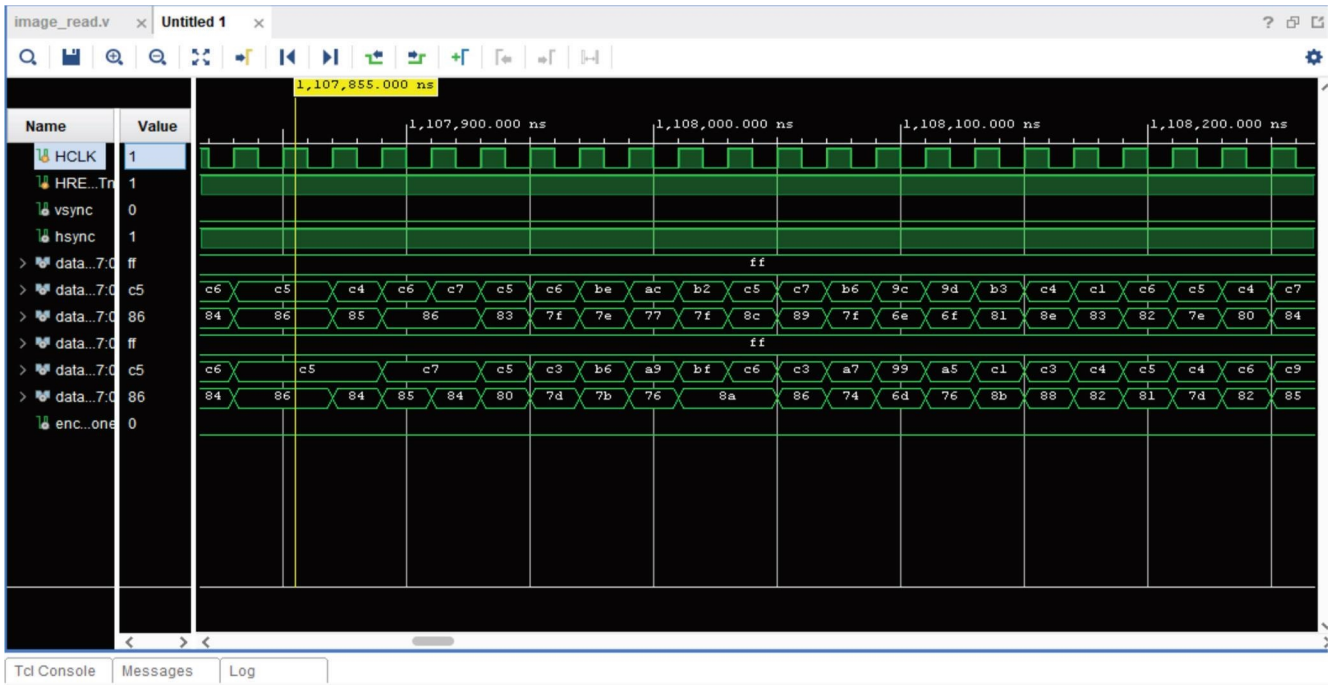


FIGURE 6: Timing diagram after the simulation of image enhancement process.

The presented data in Figure 5(b) depict the digital radiography image subsequent to a decrease in luminance. The brightness operation was chosen from the file titled "parameter.v." The original value of each pixel in the DR image was decremented by 100. The output generated by simulation indicates a perception of a comparatively darker image.

Figure 5(c) displays the digital radiography image subsequent to an augmentation in luminosity. The brightness operation was chosen from the "parameter.v" file. In the event of a brightness increase, the corresponding SIGN value was assigned a numerical value of 1 within the code. In the digital radiography image, a linear scaling operation was performed where the value of each individual pixel was uniformly increased by a scalar factor of 100 from its original value. The value is subject to alteration in accordance with one's preferences. The results obtained through simulation demonstrated a visibly enhanced brightness in the output image.

The DR image subsequent to the application of inversion or negative transformation is depicted in Figure 5(d). The process involves subtracting every pixel value of an image from 255 to produce a negative version of the same. The purpose of performing this task is to extract comprehensive information from the obscured areas of the dynamic range image, thereby facilitating the doctors' ability to readily detect any aberrations. The transformation has resulted in a noticeable alteration of luminosity, wherein the previously brighter region underwent a diminution in brightness and the formerly darker area experienced an increase in luminosity.

Figure 5(e) depicts the DR image subsequent to the execution of a threshold procedure. The aforementioned is additionally chosen from the file entitled "parameter.v" in an academic context. The threshold value was established as 90 in this instance. The majority of the image region exhibits low light intensities, wherein the preponderance of pixel values within the dynamic range image surpassed 90. The DR image subsequent to threshold operation at a value of 80 is depicted in Figure 5(f).

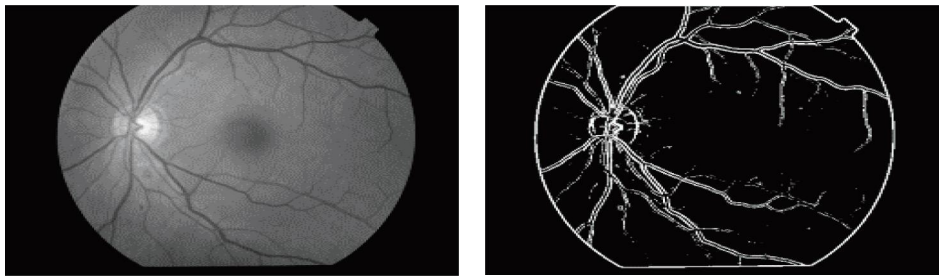
Figure 6 depicts the timing diagram observed during simulation execution. The current observation reveals the process of reading image pixels. In the interest of expediting processing time, the procedure called for the concurrent reading of two adjoining pixels. Each pixel relayed a total of six values from the hexadecimal file, comprising three distinct RGB values. These concurrent values are feed into line buffer for enhancement operation. As the values are read from hexadecimal file, therefore, the values in the timing diagram is displayed in hexadecimal format. The above timing diagram (Figure 6) displayed the data flow maintaining the order as original DR image where the clock pulse indicates the concurrent data reading of pixels, after reducing brightness, after increasing the brightness, after negative transformation, and after threshold operation. In between two operations each, the value, ff, in the timing diagram indicates the file iteration index.

Figure 6 shows consecutive two pixels and RGB values of the converted hexadecimal file of DR image. Indicating that the read and write operation of two pixels are ongoing simultaneously.

Data [7:0] R0,G0,B0-pixel 1 (horizontal)

Data [7:0] R1,G1,B1-pixel 2 (vertical)

The two pixels for each case have six values that have been depicted in Figure 6. For pixels 1 and 2, ff, c5, and 86 are memory values as well as file iteration index for the current



(a) (b)

Figure 7: (a) Converting DR image into grayscale and (b) edge detection by Sobel edge detection algorithm.

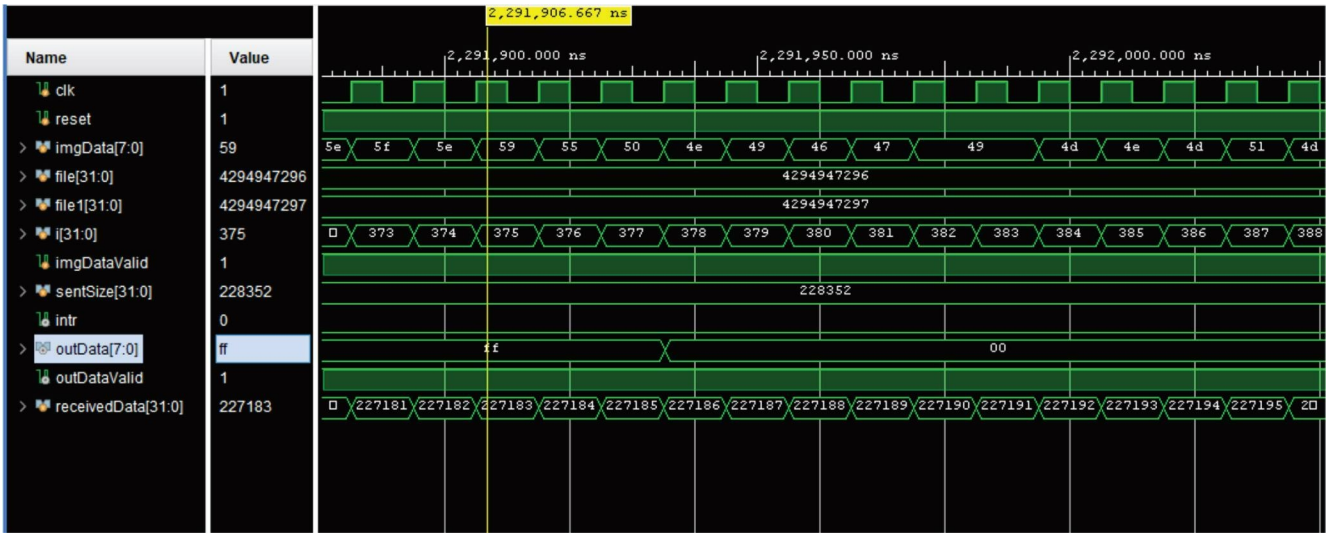


Figure 8: Timing diagram analysis of Sobel edge detection.

iteration. A clear understanding can be found that at a time two pixels will be converted, read and write in the memory that is one is horizontal and another one is vertical. Thus, the framework has lined up a total of six values for each initial case. The last two data stream indicates the enhanced values of pixels after the enhancement operation such as after reducing brightness, after increasing the brightness, after negative transformation, after threshold operation both horizontal and vertical, respectively. Again, looking at Figure 6, if we compare the initial and enhanced pixels, it can be clearly found that the dilated version in the timing diagram indicates brightness enhancement of the DR image and vice versa. Similarly, negative transformation and threshold operation are also depicted in the timing diagram.

Another point to be mentioned is that "parameter.v" file is only used to define the initial input and output taking in consideration that when to select a particular operation of image enhancement. So, "parameter.v" does not have any direct relation with Data [7:0] R0,G0,B0-pixel 1 (horizontal) and Data [7:0] R1,G1,B1-pixel 2 (vertical) pixel data that have been shown in the timing diagram.

5.2. Simulated Output for Sobel Edge Detection. The process of performing Sobel edge detection involved the transformation of the DR image into a grayscale image, as exemplified in Figure 7.

Upon successful implementation of the system architecture, whereby the requisite code was written, efficient detection of the edge of the DR image was achieved. This outcome has been visually demonstrated in Figure 7. In light of this, medical professionals can readily identify the compromised blood vessel within the retinal fundus.

In the simulation process, the timing diagram of the Sobel edge detection algorithm is depicted in Figure 8. It is possible to observe the processed input and output data at distinct intervals in the course of its processing. When outDataValid attained a value of 0, it signaled the cessation of available data to read and the subsequent termination of processing activity.

5.3. RTL Diagram Analysis for the Proposed Method. The diagram depicted in Figure 9 provides an overview of the envisaged system architecture for conducting sobel edge detection. The comprehensive RTL diagram is comprised of three distinct modules. The hierarchical representation of all modules for the proposed algorithm is present herein. The imageControl and conv modules are integrated within the imageProcessTop module, resulting in the production of

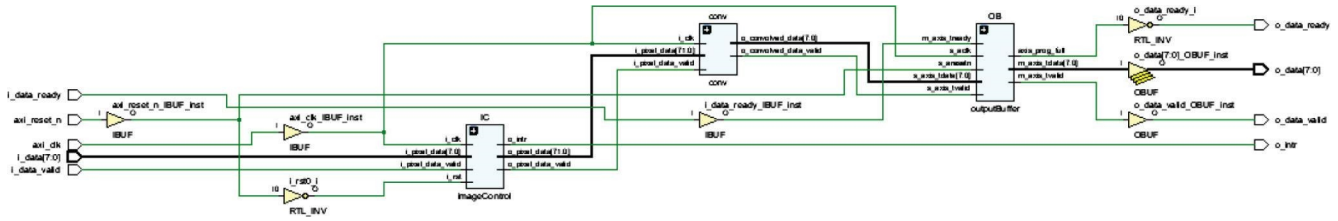


Figure 9: Overall RTL diagram for the proposed system.

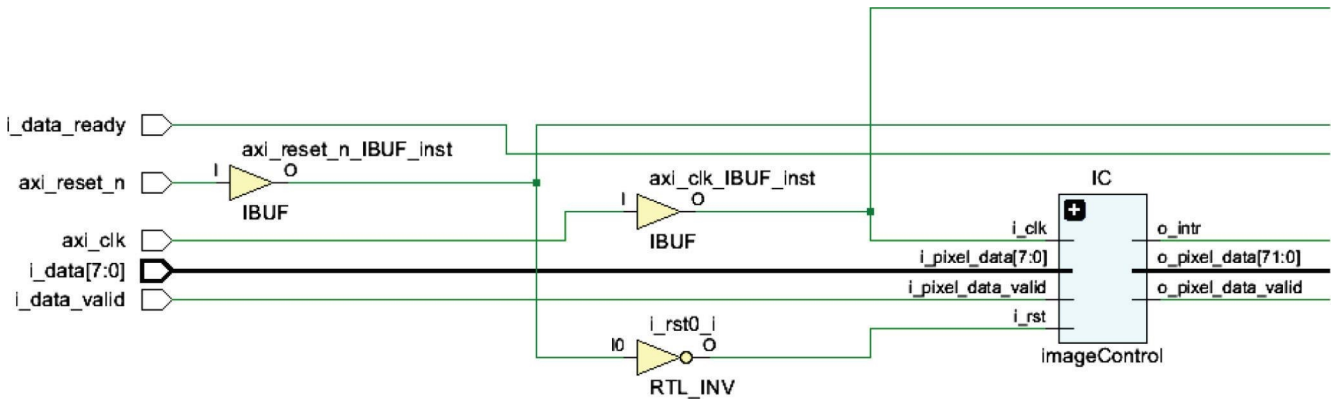


Figure 10: RTL diagram of image control module.

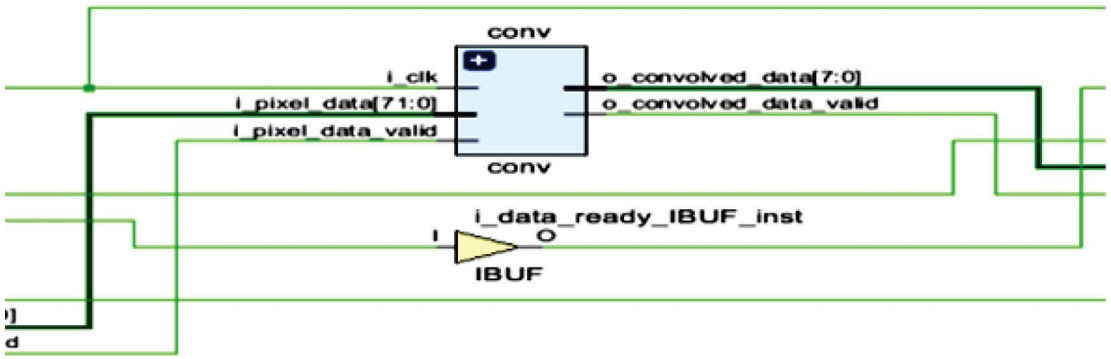


Figure 11: RTL diagram of convolution module.

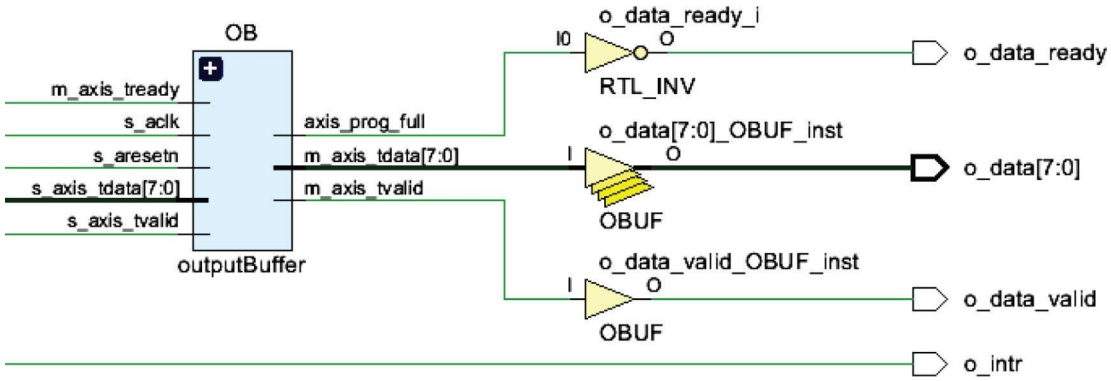


Figure 12: RTL diagram of top module.

Table 1: Simulation time required in MATLAB vs. FPGA.

Transformation	MATLAB (s)	FPGA (ms)
Brightness control	5.2484	5.5828
Inverting	1.4041	5.5828
Threshold	18.5613	5.3620
Sobel edge detection	2.0903	2.652

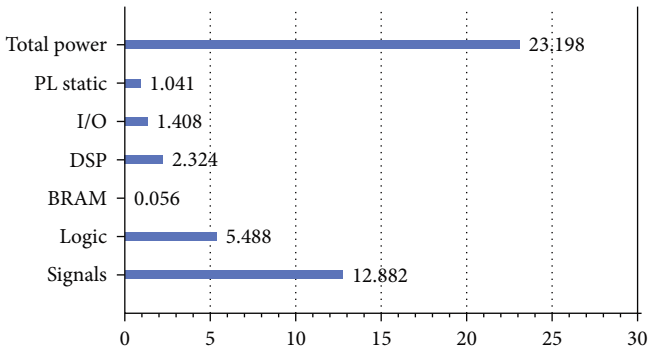


Figure 13: Power consumption of the proposed system.

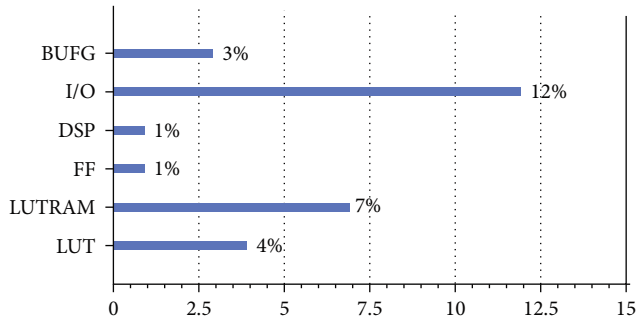


Figure 14: Power utilization of postsynthesis.

the outputBuffer. This amalgamation results in an effective image processing mechanism.

The image control module of the proposed framework is shown in Figure 10. Image information are studied to convoluted afterward in this module. The internal input and output ports can be seen in Figure 10. Four line buffers are instantiated here. It will determine which line buffer will get the data from the external world.

Figure 11 depicts the RTL of the convolution module. In this context, the line buffer composed of pixels will undergo a multiplication process by the Sobel kernel, followed by an addition process, leading to the execution of a convolution operation. The aforementioned task is to be carried out in an academic manner of writing. The determination of the pixels to be convoluted is processed by the image control module.

In Figure 12, the RTL diagram of top module is appeared where all other modules are combined and a FIFO as outputBuffer is generated through which the processed pixels can be compared with the input pixels. Finally, by observing the o_data [7:0] in the timing diagram, the value of the pixel after implementation of sobel edge algorithm can be observed.

Utilization	Postsynthesis | Postimplementation

Graph | Table

Resource	Estimation	Available	Utilization (%)
LUT	1,876	53,200	3.53
LUTRAM	1,153	17,400	6.63
FF	202	106,400	0.19
DSP	2	220	0.91
IO	23	200	11.50
BUFG	1	32	3.13

Figure 15: Utilization statistics of postsynthesis.

Table 2: Comparison of synthesis result.

Resources	Sree and Rao's [7] study	Nausheen et al.'s [6] study	Proposed method
Slice LUTs	2,201 (7%)	7,444 (27%)	1,876 (3.53%)
Flip–flop	1,404 (4%)	358	202 (0.19%)
RAM	2,369 (8%)	—	1,153 (6.63%)
Bounded IOB	17 (6%)	7 (2%)	23 (11.50%)
Slice register	1,401 (9%)	411 (1%)	532 (0.5%)

5.4. Comparison of Analysis. Table 1 shows the comparison of preparing time required between MATLAB and FPGA. The time required for brightness operation was 5.2484 s, for Sobel edge detection was 2.0903 s, and so on. In the case of FPGA, the values were 5.5828 and 2.652 ms. The time was in millisecond (ms) extend for FPGA which is much lesser than MATLAB. This demonstrates the speedier handling capacity of FPGA.

In Figure 13, the total power utilization of the proposed framework is appeared where the total power is 23.198 W. The signals expend most of the power 12.882 W while the BRAM expends the slightest. These insights are got after synthesizing the design. In Figure 14, the source utilization of the framework is appeared. BUFG, I/O, DSP, FF, LUTRAM, and LUT are utilized 3%, 12%, 1%, 1%, 7%, and 4% separately from the accessible assets in FPGA, respectively.

Figure 15 shows the source utilization in terms of approximated number as well as percentage. It can be seen that utilization is comparatively less than desired. A comparison is made in tabular form with our proposed algorithm to the available algorithms below.

Table 2 shows the comparison considers of the proposed framework with others for Sobel edge detection calculation. The LUT device, in this demonstration, is utilized 3.53% whereas in Sree and Rao's [7] study and Nausheen et al.'s [6] study, it is 7% and 27%, respectively. The FF utilization is additionally less than the other strategies specified. But the rate of Bounded I/O in this strategy is comparatively higher than others. Otherwise rest of the measurements appears that the proposed system is not effect in the system for successful operation.

Figure 16: Pin layout of proposed method.

The pin layout of the proposed system for Sobel edge detection is presented in Figure 16. The segment highlighted in light blue signifies the resources that have been employed within the entirety of the FPGA.

6. Conclusion

The importance of image processing in all facets of digitization exceeds imagination, especially within the medical domain. The identification of any pathological condition is predicated upon restorative imaging techniques and the requisite preparatory measures preceding the acquisition of such an image. The condition known as DR is acknowledged as a bona fide complication of diabetes, which poses a significant risk to one's visual function. The prompt identification of ocular conditions may alleviate the risk of patients experiencing visual impairment. Consequently, the production of diagnostic radiography images has been undertaken throughout the course of research proposal activities. The enhancement of images and the identification of edges algorithm are interlinked with the digital radiography image. Thereafter, the resultant outcomes are observed through a comparative analysis. The employment of FPGA technology for simulation and management tasks has been found to necessitate a reduced amount of time in comparison to conventional program approaches. The proposed framework design for Sobel edge detection is deemed highly effective, as it minimizes the utilization of computational resources in comparison to extant models. In future research endeavors, the preprocessing of DR images may be leveraged in conjunction with machine learning techniques to effectively extract salient features and recognize instances of DR through FPGAs.

Data Availability

The data were collected from Khulna BNSB Hospital, Bangladesh [3].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

MO, SMRI, FR, and MAH conceived the presented idea, developed the theory, performed the computations, and verified the analytical methods. All the authors presented the idea, data collection, method developed, and the manuscript edited. All authors discussed the results and contributed to the final manuscript.

Acknowledgments

The authors thank the Department of Electronics and Communication Engineering staff and all professors for supporting equipment and other facilities.

References

[1] D. S. Sisodia, S. Nair, and P. Khobragade, "Diabetic retinal fundus images: preprocessing and feature extraction for early detection of diabetic retinopathy," *Biomedical and Pharmacology Journal*, vol. 10, no. 2, pp. 615–626, Jun. 2017.

[2] M. Yildirim and A. Cinar, "Simultaneously realization of image enhancement techniques on real-time FPGA," in *International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–6, IEEE, Malatya, Turkey, 2019.

[3] D. M. A. Hasan, D. S. M. R. Islam, and D. M. A. K. Azad, "A hospital-based cohort study on risk factors for diabetic retinopathy among patients with type 2 diabetes mellitus," *Informatics in Medicine Unlocked*, vol. 38, Article ID 101219, 2023.

[4] L. Cavinato, I. Fidone, M. Bacis, E. D. Sozzo, G. C. Durelli, and M. D. Santambrogio, "Software implementation and hardware acceleration of retinal vessel segmentation for diabetic retinopathy screening tests," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1226–1229, IEEE, Jeju, Korea (South), 2017.

[5] W. Khan, "Diabetic retinopathy detection using image processing: a survey," vol. 13, 2023, Accessed: Apr. 13 [Online] https://www.academia.edu.

[6] 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, 2018.

[7] V. K. Sree and P. S. Rao, "Hardware implementation of enhancement of retinal fundus image using Simulink," in *2013 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, pp. 239–244, IEEE, Visakhapatnam, India, 2013.

[8] M. I. AlAli, K. M. Mhaidat, and I. A. Aljarrah, "Implementing image processing algorithms in FPGA hardware," in *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pp. 1–5, IEEE, Amman, Jordan, 2013.

[9] T. Imsaengsuk and S. Pumrin, "Feature detection and description based on ORB algorithm for FPGA-based image pocessing," in *2021 9th International Electrical Engineering Congress (iEECON)*, vol. 1, pp. 420–423, IEEE, Pattaya, Thailand, 2021.

[10] H. El Khoukhi and M. A. Sabri, "Comparative study between HDLs simulation and Matlab for image processing," in *International Conference on Intelligent Systems and Computer Vision (ISCV)*, pp. 1–6, 2018.

[11] Chiuchisan, "Implementation of medical image processing algorithm on reconfigurable hardware," in *2013 E-Health and Bioengineering Conference (EHB)*, vol. 1, pp. 1–4, Iasi, Romania, 2013.
[12] 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.
[13] N. Tabassum, S. M. R. Islam, and F. Bulbul, "Brain tumor detection from brain MRI using soft IP core on FPGA," *Circuits, Systems, and Signal Processing*, vol. 42, no. 2, pp. 724–747, 2023.
[14] J. R. Kim and J. W. Jeon, "Sobel edge-based image template matching in FPGA," in *2020 IEEE International Conference on Consumer Electronics–Asia (ICCE–Asia)*, vol. 1, pp. 1-2, IEEE, Seoul, Korea (South), 2020.
[15] S. Yaman, B. Karakaya, and Y. Erol, "Real time edge detection via IP-core based sobel filter on FPGA," in *International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*, pp. 1–4, IEEE, Elazig, Turkey, 2019.
[16] P. Chen, Y. Wang, and J. Xing, "Research on design and verification of sobel image edge detection based on high level synthesis," in *2019 6th International Conference on Dependable Systems and Their Applications (DSA)*, vol. 1, pp. 491-492, 2020.
[17] P. Kumar Charu and K. Singh, "Hardware model for efficient edge detection in images," in *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, vol. 1, pp. 798–802, IEEE, Greater Noida, India, 2020.
[18] C.-Y. Lee, Y.-H. Kim, and Y.-H. Lee, "Design of real-time hardware for edge detection," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, vol. 1, pp. 348–353, IEEE, Las Vegas, NV, USA, 2018.
[19] S. Eetha, S. Agrawal, and S. Neelam, "Zynq FPGA based system design for video surveillance with sobel edge detection," in *2018 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, pp. 76–79, IEEE, Hyderabad, India, 2018.