International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 12, No. 1, March 2023, pp. 19~28
ISSN: 2089-4864, DOI: 10.11591/ijres.v12.i1.pp19-28

A systematic literature review on hardware implementation of image processing

Zul Imran Azhari, Samsul Setumin, Anis Diyana Rosli, Siti Juliana Abu Bakar
Centre for Electrical Engineering Studies, Universiti Teknologi MARA Cawangan Pulau Pinang, Permatang Pauh, Malaysia

Article Info

Article history:

Received Apr 22, 2022
Revised Aug 8, 2022
Accepted Aug 26, 2022

Keywords:

Field programmable gate array
Hardware implementation
Image enhancement
Image processing
Systematic literature review

ABSTRACT

Image processing has become under the spotlight recently and leads to a significant shift in various fields such as biomedical, satellite images, and graphical applications. Nevertheless, the poor quality of an image is one of the noticeable limitations of image processing as it restricts efficient data extraction to be conducted. Conventionally, the image was processed via software applications such as MATLAB. In spite of the software's ability to cater to the data extraction of low-quality image issues, it still suffers from the time-consuming issue. As the ability to obtain a rapid outcome is a favorable feature of efficient image processing, the use of hardware in image processing is deemed to keep the addressed issue at bay. Thus, the image enhancement techniques using hardware have gradually rising interest among researchers with numerous approaches such as field programmable gate array (FPGA). In this study, 25 different research papers published from 2016 to 2021 are studied and analyzed to focus on the performance of FPGA as hardware implementation in image processing techniques.

Corresponding Author:

Samsul Setumin
Centre for Electrical Engineering Studies, Universiti Teknologi MARA Cawangan Pulau Pinang
Permatang Pauh, 13500 Pulau Pinang, Malaysia
Email: samsuls@uitm.edu.my

1. INTRODUCTION

Digital image processing (DIP) is widely employed in diverse applications such as communication, multimedia services, arts, medicine, space exploration, surveillance, automated industry, robotics, aerospace as well as in education areas [1], [2]. Technology advances in digital image processing are influenced by several major goals. For instance, one goal is to enhance human perception by enhancing visual data. Consequently, any image that sought to enhance its quality signifies that it intends to enhance the quality of the original image [3]. Implementation of real-time IP algorithms on sequential processors has significant drawbacks for large image sizes with high resolution. Several initiatives were taken to execute it in hardware, which allows for optimized and parallel methods. Field programmable gate array (FPGA) technology, on the other hand, is viewed to offer a reliable alternative. In paper [1], the real-time biomedical image enhancement (BIE) method using FPGA is proposed to improve the quality of biomedical images for human viewing. In this paper, brightness control, contrast stretching, and threshold enhancement is implemented to observe human veins.

Enhancing the quality of an image for human interpretation and analysis is the primary goal of image processing [4]. Various image processing applications, such as segmentation, object detection, and recognition, utilize image enhancement techniques in the pre-processing step. These methods either improve the image to make it better for the human eye or improve the algorithm of automatic computer applications [5]. For a complex analysis and image display, the primary goal is to draw attention to certain characteristics that are significant in an image. Considering the acquisition process was less than optimal, this technique was developed to improve the visual quality of an image. For a particular application, the results of the image processing are preferable to the original image [5].

Running real-time image processing methods on serial processors turns out to have additional drawbacks for the huge number of images with high resolution. However, there have been numerous attempts to implement it on hardware, where it is possible to use more parallel and optimized techniques. An appropriate substitute is the usage of FPGA technology [1]. FPGAs have recently been acknowledged to have significantly improved in terms of size and functionalities. This has raised interest in adopting it as a platform for image processing applications, particularly those that need real-time processing [1]. Moreover, a haze is typically the root cause of an image that lacks contrast. The image appears blurry because it reduces visibility, especially for far-off scenes [6]. Therefore, contrast enhancement of an image is important to remove the haze and increase the visibility of an image. There are a lot of algorithms proposed by researchers to enhance images that suffer from low contrast and low exposure. *Horé and Pecht* proposed 2D filters based on a general approximation utilizing only the power of two that produced a very good image [7].

FPGAs have recently advanced massively in both size and functionalities. This has piqued interest in adopting it as a development platform for IP applications, particularly in an application where real-time processing is essential [8]. In an FPGA, a matrix of programmable logic cells is linked via a grid of coupled lines and switches. These input/output cells are positioned around the chip's edge and serve as a link between the chip's interconnection lines and its outer pins. Note that, it is necessary to define the logic function of each cell as well as the switches for joining lines in order to set up an FPGA. FPGA implementation of low-light image enhancement does have a significant impact on the processing speed compared with software implementation while pricing comparable image results [9].

In this paper, we provide a thorough analysis and systematic literature review (SLR) to aid scholars in the field of DIP. The survey includes image processing algorithm hardware implementation research from 2016 to 2021. The implementation of image processing on hardware is the main focus of this review. We are inspired to perform this SLR since there are no other SLR papers covering hardware implementation of image processing. A total of 263 different research publications have been gathered, 25 of which focused on the hardware implementation of AI and ML algorithms. We used stringent exclusion criteria for this study to consider as many papers as we could in order to represent a diversity of perspectives on the relevant topic.

The remaining sections are arranged as follows: section 2 discusses a literature review of related studies. The methodology is then covered in section 3, which explains the criteria that were utilized to categorize and assess the quality of the gathered research papers. We go over the analyses and answers to the research questions that were posed in the methodology section in section 4. Section 5 presents the work's conclusion in its final form.

2. LITERATURE REVIEW

The investigation of the use of hardware in image processing techniques becoming progressively popular in recent years. It has been the subject of a number of studies. The attention occurred as image processing algorithm demands vast computational resources, especially in real-time processing [10]. as very large-scale integration (VLSI) technology progresses, hardware implementation is believed to be a viable choice in image processing to attain more rapid and effective outcomes [10]. Consequently, there are numerous hardware implementations for various applications are discovered. Over the last five years, the image processing areas have seen many studies published, focusing on hardware-software optimizations and their implementation methods.

The primary goal of image enhancement was to improve the brightness, contrast, and improved visual quality of an image for human viewing [11], [12]. Image enhancement evolves in a wide range of applications such as biomedical [8], and haze removal [6], [12]. Dong *et al*. [13] proposed a high-speed, effective algorithm for enhancing a dim light video. In this paper, the enhancement was done by inverting input with dim-lighting video and employing an optimized image de-hazed algorithm on the inverted video. On the other hand, Ren *et al*. [14] proposes another method for improving the visual quality of an image with dim light by adjusting the illumination to be uniform and strong in intensity.

An array of programmable logic gates can be found on the FPGA chip (PLA). FPGA does not have a chip architecture that has been rebuilt. The programmers used logic gates to construct the hardware architecture of the application to program the FPGAs. Two hardware description languages (HDLs) such as VHDL and Verilog are employed in FPGA hardware architecture. On the other hand, "C, Java, system C" and domain-specific languages (DSL) are used to program FPGAs [10]. However, 'C' type languages are incompatible for non-sequential programming applications, whilst DSLs are appropriate for certain applications only [15].

When developing an effective algorithm, it is entirely beneficial to figure out the FPGA hardware thoroughly. High-level design exploration is carried out using high-level languages, and it begins with algorithmic verification. The programmer must first verify the algorithm prior to creating a code that will execute well [15]. Only software coding can provide decent, optimized code. As it is flexible in nature, it faces two challenges: sequential execution and operational serial memory. Due to these two factors, reading and writing image frames require a long time.

The most recent state-of-the-art solutions map applications with substantial data processing; this can be accomplished by employing hardware parallelism with multicore or FPGA-based designs. The key advantages of employing FPGAs as hardware platforms are their low cost, low power consumption [16], and a short time to market. Several optimization strategies may be used to extract parallelism in hardware, resulting in a significant increase in processing speed compared to software. The data from software is transferred to hardware, manually or automatically depending on the designer's preferences. For this process, the term high-level synthesis is used. A wide range of commercial tools, such as Vivado HLS, Catapult, and others able to convert algorithmic design (SW) to hardware design (HW) [15]. These tools advise the designer to use optimizations such as loop unrolling, array splitting, and so on, resulting in a time and resource trade-off. As an embedded system design suggests, a hierarchy-structured IP core is built using the mentioned tools and will interact with the processor's local bus.

Image processing applications, in general, take a long time to process since the applications require a large data set and a complex algorithm. An image processing algorithm with improved performance and shorter time to market can be achieved via configurable hardware and system-level programming languages [17]. For hardware implementation, a variety of technologies are available. Reconfigurable devices such as application specific integrated circuits (ASICs) and FPGAs are able to assist parallelism and pipeline techniques [17]. The use of reconfigurable hardware to construct image processing algorithms reduces time-to-market costs and allows for quick prototyping with simplified debugging and verification steps. As a result, reconfigurable devices appear to be an excellent option for implementing image processing algorithms.

This work adds to the current body of knowledge and helps to provide a complete picture of image processing. This survey's contributions can be summarized as follows:

- To give the most relevant information on this issue, the paper uses an SLR technique.
- The survey examines projects completed in the last five years, from 2016 to 2021.
- The survey compares and contrasts existing FPGA hardware.
- The study includes papers on image processing algorithms at the software level.

3. METHOD

The SLR covers a wide range of hardware implementations of image processing. The employed method is based on Kitchenham and Charter's method [18], presented in Figure 1. Firstly, research questions for this SLR are identified which is important to set the focus of this research. Next, search terms and exclusion criteria are determined to gather relevant papers. Database library and year range will also be specified before performing the search. Paper selection and inclusion criteria have also been identified to screen relevant papers from overall searched papers. The remaining papers will then be categorized and applied for selection and filtration to further screen the papers to ensure the remaining papers are the most relevant papers for this research. Technical metrics and parameters will then be extracted from the papers and processed afterward. The aim of this survey is to obtain the answer to several research questions concerning:

RQ 1: Application perspective: what are the most common applications that make use of image processing in hardware? This inquiry aims to discover an image processing applications that works via hardware implementation.

RQ 2: Image processing perspective: what image processing algorithms and tools are utilized to implement them in hardware? The goal of this topic is to look at the most common algorithm used in image processing.

RQ 3: Hardware perspective: what are the employed hardware platforms for image processing acceleration? The goal of this is to identify the type of hardware used.

3.1. Search strategy

The next step in performing this review is to determine the search terms and gather relevant papers to address the research questions. The terms "image processing" and "image enhancement" are employed, as well as "hardware implementation" and "FPGA". Using the stated search criteria, the following digital libraries are utilized to obtain all relevant items (journals and conference papers): Scopus and Web of Science.

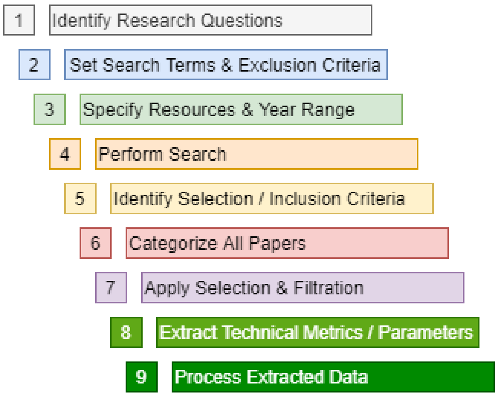
1 Identify Research Questions
2 Set Search Terms & Exclusion Criteria
3 Specify Resources & Year Range
4 Perform Search
5 Identify Selection / Inclusion Criteria
6 Categorize All Papers
7 Apply Selection & Filtration
8 Extract Technical Metrics / Parameters
9 Process Extracted Data

Figure 1. SLR method

3.2. Study selection

Initially, 263 papers were gathered using the previously acknowledged search terms. Next, the papers were grouped into more specific categories. In the selection and filtering process, the following steps are taken into consideration:

Step 1: Remove articles that are duplicated or have several versions.
Step 2: Use inclusion and exclusion criteria to exclude any papers that are not relevant.
Step 3: Remove review papers from the collected papers.
Step 4: Apply quality assessment rules to include qualified papers that best address the research questions.
Step 5: Using reference lists of the collected papers, look for new similar publications and repeat the process.

The following are the applied inclusion and exclusion criteria:

a. Inclusion criteria:
– Date from 2016 to 2021.
– Only journals and conference papers that discuss the optimization or implementation of image processing in hardware.

b. Exclusion criteria:
– Non-journal and non-conference articles.
– Articles that do not discuss image processing application of image enhancement and edge detection.

3.3. Quality assessment rules

The final stage in determining the list of articles to be examined in this review is to apply the quality assessment rules (QARs). The QARs are crucial in ensuring that the quality of research publications is properly assessed. As a result, 10 QARs are assigned, each earning one point out of ten. Each QAR's score is determined by the following formula: "completely answered" = 1, "above average" = 0.75, "average" = 0.5, "below average" = 0.25, and "not answered" = 0. The score of each item is the sum of the marks achieved for the 10 QARs. Furthermore, the article is considered if the result is 5 or greater; otherwise, it is excluded. The following list is the QARs that are taken into account in this study:

QAR 1: Are the research objectives well-defined?
QAR 2: Are image processing techniques clearly defined and deliberated?
QAR 3: Is there any evidence of the proposed technique being used in practice in the paper?
QAR 4: Is there any evidence that the proposed technique has been tested in practice?
QAR 5: Are the experiments well-thought-out and well-supported?
QAR 6: Is the suggested image processing design thoroughly validated using common-standard test cases?
QAR 7: Is the result accurately reported?
QAR 8: Is the proposed image processing design compared to others?
QAR 9: Are the approaches for analyzing the data appropriate?
QAR 10: Does the overall study contribute significantly to the image processing area of research?

3.4. Data extraction strategy

In this stage, the final selection of papers will be evaluated in order to gather crucial data for answering the research questions. To begin, general information about each document is extracted, such as the paper number, title, publication year, and publishing type. Then more particular data is required, such as algorithm models and hardware accelerator categories, such as FPGA, GPU, ASIC, or a mix of these. Finally, any information that is directly connected to the study topics is sought. Due to the underlying unstructured nature of the required data, extracting such details is difficult. For example, some articles [10] proposed the compilation of software into a hardware descriptive language that could be implemented on FPGAs. It is worth noting that not all of the papers responded to the addressed research questions.

3.5. Synthesis of extracted data

After gathering information from the selected publications, numerous approaches were employed to collect and construct the desired responses that were in line with the research questions. To get the necessary information for RQ 1, RQ 2, and RQ 3, a qualitative synthesis is used.

4. RESULTS AND DISCUSSION

The results of the predetermined research questions are discussed in this section. A study of image processing hardware implementations between 2016 and 2020 based on the selected papers is viewed. The first subsection provides a summary of the articles collected, followed by subsections that go through each study issue in detail, highlighting key conclusions and observations.

4.1. Study overview

As previously stated, this study identifies and discusses 25 out of 263 research publications on FPGA-based implementation. It was particularly valuable for research to create and test image processing because of its rapid prototyping environment and ability to give an easy prototype environment with significant computational capabilities. As a result, methods such as edge detection, Gaussian filtering, histograms, and many more are rapidly evolving. FPGA is seen to be a feasible alternative for implementing these algorithms because of its considerable flexibility. Furthermore, current FPGA design trends have made it more affordable and have attracted substantial attention for deep learning research.

Figure 2 shows that there is almost the same number of research papers collected from 2016 to 2021. The equal distribution signifies the relevancy of the extracted information over time. In other words, the study fairly examined the papers within these past 6 years.

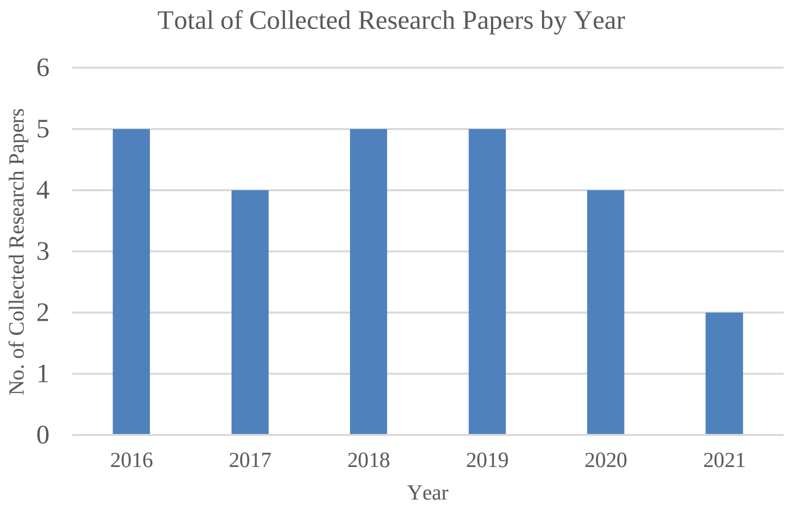

Figure 2. Collected research papers between 2016 and 2021

4.2. RQ 1: Application perspective: what are the most common applications that make use of image processing in hardware?

The collected research papers are categorized according to the implemented application. Three categories are identified based on the implemented applications as follows: image enhancement and edge detection. The field of image processing is rapidly expanding. Picture processing techniques are used in numerous new applications, including computer graphics, biomedical imaging, satellite imaging as well as underwater image restoration and enhancement. As shown in Figure 3, 79% of the papers are image enhancement-related applications. Most research articles on image processing tend to focus on image enhancement. Image enhancement has become increasingly popular due to its close ties to computer vision and image analysis. It has a wide range of applications, but it's most commonly used to improve the quality of an image for better interpretation. Other image processing applications and edge detection have been studied for implementation in hardware in recent years. These applications grew in prominence as a result of their potential to automate numerous real-time decision-making systems.

Alareqi *et al.* [8] explained the real-time biomedical image enhancement (BIE) method using FPGA is proposed to improve the quality of biomedical images for human viewing. In this paper, brightness control, contrast stretching, and threshold enhancement is implemented to observe human veins. Ngo *et al.* [12] proposed an image enhancement application for image haze removal. Due to the unavoidable adverse effects of poor weather conditions, images or videos taken outdoors typically suffer from an apparent loss of contrast and detail. The technique proposed by the authors aims to eliminate undesirable effects and restore clear visibility. Alex *et al.* [19] showed image enhancement application enhances the underwater image using contrast limited adaptive histogram equalization (CLAHE). This technique is able to enhance contrast and improves the quality of an image that suffers from poor lighting conditions, such as an underwater image.

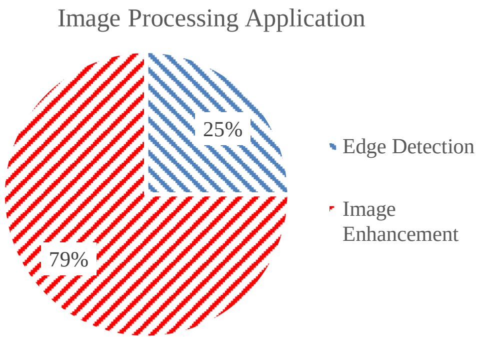

Figure 3. Comprehensive distribution of all image processing applications

4.3. RQ 2: Image processing perspective: what image processing algorithms and tools are utilized to implement them in hardware?

This section presents an algorithmic analysis of the collected research papers in order to identify the most widely used algorithms in image enhancement and edge detection applications. Most of the collected research papers discuss the implementation of image enhancement. Table 1 shows all algorithms used in the collected papers. There are various algorithms proposed for image enhancement. The most proposed algorithm for image enhancement is contrast enhancement as it aids to enhance and improve the quality of an image. Histogram-based algorithms are also used in image enhancement. The visual appearance of an image can be improved by using the histogram. The dynamic range of pixels, contrast, and many other issues that arise during the acquisition of an image is shown in the histogram [11].

The algorithm proposed by Ngo *et al.* [12] is a single image haze removal algorithm that aims to improve the quality of the images and restore clear visibility. The proposed algorithm includes a detail enhancement algorithm in the pre-processed to restore the faded detail of the input image. Weight maps are first calculated concerning dark channels to accurately blend haze-free areas into the fused picture and then normalized to avoid the out-of-range issue [12]. As a result of fusing a series of under-exposed images, the hazy input is lighter than the resultant image. For this reason, adaptive tone remapping is used to increase luminance and highlight the chrominance. Soma and Jatoth [6] propose another image enhancement application using the de-hazing algorithm. Two methods for hardware implementation of the image de-hazing algorithm are discussed in this paper. The authors propose the pixel-wise and grey image-based de-hazing algorithm. The hazy input image is converted into two images that are grey images and a minimum image. These two images were then converted into an average image, significantly reducing the halos and artifacts present in the final dehazed images. Timarchi *et al.* [20] propose a novel algorithm based on ConText, called the modified context (MCT), to attain a high-quality stego-image. The proposed algorithm is based on threshold level, which can lead to faster and more power-efficient implementation.

Whilst, for edge detection application, two papers employed the Sobel operator algorithm [20], [21]. Aside from that, others used algorithms for edge detection are Harris, smallest univalue segment assimilating nucleus (Susan) [22], Canny [23], Robert, Prewitt, and Laplacian of Gaussian (LoG) operator [24]. Edge detection is an approach of an image processing technique to distinguish the edge of objects within the image. The Sobel operator operates based on differentiation and is used to detect edges. As a comparison to conventional edge detection operators, the Sobel operator is more applicable due to its simplicity and higher precision [20]. Using the Sobel operator, the paper [21] was able to identify diseases in hevea tree leaves by computing the gradient of each pixel in the image.

Table 1. Summary of the algorithm in each category proposed in the collected papers

Type of Implementation	Image processing application	
	Image enhancement	Edge detection
Algorithm	Average, Gaussian, sharpening filter, grayscale and edge detection [25]	Harris and susan [22]
	Histogram (brightness, contrast enhancement, region of interest) [11]	Sobel Operator [21], [26]
	De-hazing [6], [12]	Robert, Prewitt, Sobel, and Laplacian of Gaussian (LoG) operator masks [24]
	Power-of-two terms [24]	Canny edge detection [23]
	Gaussian-based halo-reducing filter [7], [27]	Biomedical image enhancement (BIE) [8]
	Modified context (MCT) [20]	
	Median filter [28]	
	Adding image [1]	
	Adaptive histogram equalization (AHE) [19]	
	2D adaptive DIP [29]	
	Boundary discriminative noise detection (BDND) [30]	
	Guided image filtering and Halide [31]	
	Discrete wavelet transform (DWT) [2]	
	Contrast, brightness enhancement, image inverting, and threshold operation [32]	
	Gaussian-based smoothing filter [33]	
	Particle swarm [34]	
	Medical image algorithm [35]	

4.4. RQ 3: Hardware perspective: what are the employed hardware platforms for image processing acceleration?

This section discusses the hardware-implemented image enhancement based on standard figures of merit as depicted in Figure 4. 13 papers discussed hardware implementation for image enhancement. 10 papers discussed hardware implementation using Xilinx board and the remaining 3 papers implemented hardware in image processing via Altera board.

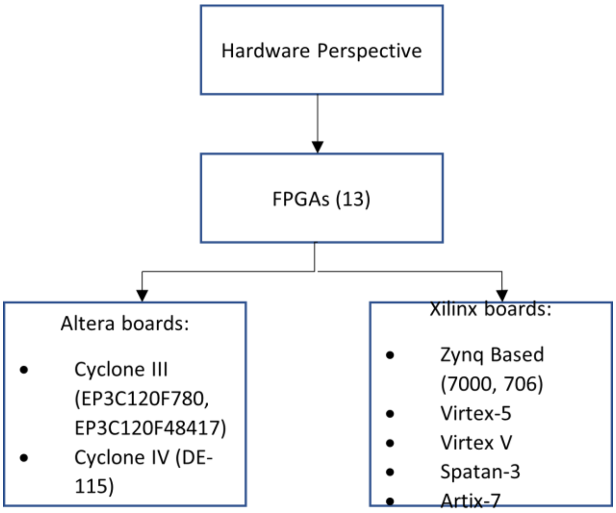

Figure 4. Summary of hardware perspective section's outcomes

Out of 13 papers that implemented image enhancement in hardware, only papers [12] and [6] can be compared as only these papers employs similar algorithm. Soma and Jatoth [6] proposed hardware implementation via the use of the Xilinx Zynq-706 FPGA board, while the paper [12] proposes the Xilinx Zynq-7000 FPGA board for its hardware implementation. Paper [12] uses a video file, while paper [6] uses an image file as an input. Therefore, it is impossible to perform an accurate comparison due to these distinct inputs. Table 2 compare the setup between these two papers.

Table 2. Comparison between hardware mapping of haze removal technique

Parameter	[12]	[6]
FPGA	Zynq-7000	Zynq-706
Max. clock (MHz)	242	933
Latency (ns)	4.12	1.07
LUT available	218600	218600
LUT utilized	30676	2664

5. CONCLUSION

This study compares existing image processing with hardware implementation in an SLR. Twenty-five papers from 2016 to 2021 focusing on hardware implementation in image processing were analyzed. The study found that most of the papers analyzed implemented image enhancement applications in their paper. This shows that image enhancement is an interesting application whose main goal is to improve the image so a piece of useful information can be extracted from it. Image processing algorithm for both image enhancement and edge detection applications has been identified and listed. For image enhancement, a de-hazing algorithm is commonly used in image processing to increase the visibility of an image. All hardware employed in those reviewed papers has been listed from a hardware perspective. However, since only two papers use the same algorithm, a thorough analysis in terms of FPGA resource usage can not be done. In-depth analyses of the various implementation platforms, tools, and strategies used throughout the previous five years are presented in this paper. Finally, we hope that this SLR paper can provide future research direction to researchers for hardware implementation, particularly on image processing aspects.

ACKNOWLEDGEMENTS

The authors would like to thank Universiti Teknologi MARA, Cawangan Pulau Pinang for supporting the research work, especially the Centre for Electrical Engineering Studies, by providing a platform to gather the required papers and so on.

REFERENCES

[1] H. el Khoukhi and M. A. Sabri, “Comparative study between HDLs simulation and MATLAB for image processing,” *2018 International Conference on Intelligent Systems and Computer Vision, ISCV 2018,* vol. 2018-May, pp. 1–6, 2018, doi: 10.1109/ISACV.2018.8354046.

[2] N. Chervyakov, P. Lyakhov, D. Kaplun, D. Butusov, and N. Nagornov, “Analysis of the quantization noise in discrete wavelet transform filters for image processing,” *Electronics (Switzerland),* vol. 7, no. 8, p. 135, 2018, doi: 10.3390/electronics7080135.

[3] M. Sreenivasulu and T. Meenpal, “Efficient hardware implementation of 2D convolution on FPGA for image processing application,” in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT),* Feb. 2019, pp. 1–5. doi: 10.1109/ICECCT.2019.8869347.

[4] R. C. Gonzalez and R. E. Woods, *Digital image processing,* 3rd ed. Pearson International Edition prepared by Pearson Education, 2008.

[5] B. H. Ramyashree, R. Vidhya, and D. K. Manu, “FPGA implementation of contrast stretching for image enhancement using system generator,” in *12th IEEE International Conference Electronics, Energy, Environment, Communication, Computer, Control: (E3-C3), INDICON 2015,* 2016, pp. 1–6. doi: 10.1109/INDICON.2015.7443730.

[6] P. Soma and R. Jatoth, “Implementation of a novel, fast and efficient image De-Hazing algorithm on embedded hardware platforms,” *Circuits Syst Signal Process,* vol. 40, pp. 1–17, 2021, doi: 10.1007/s00034-020-01517-4.

[7] A. Horé and O. Yadid-Pecht, “On the design of optimal 2D filters for efficient hardware implementations of image processing algorithms by using power-of-two terms,” *J Real Time Image Process,* vol. 16, 2019, doi: 10.1007/s11554-015-0550-2.

[8] M. Alareqi *et al.,* “Design and FPGA implementation of real-time hardware co-simulation for image enhancement in biomedical applications,” in *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS),* Apr. 2017, pp. 1–6. doi: 10.1109/WITS.2017.7934601.

[9] X. Peng, X. Li, S. Geng, J. Wang, and F. Nie, “Low-light image enhancement based on FPGA; low-light image enhancement based on FPGA,” in *2021 IEEE 15th International Conference on Anti-counterfeiting, Security, and Identification (ASID),* 2021, pp. 66–69. doi: 10.1109/ASID52932.2021.9651721.

[10] M. A. Talib, S. Majzoub, Q. Nasir, and D. Jamal, “A systematic literature review on hardware implementation of artificial intelligence algorithms,” *J Supercomput,* vol. 77, no. 2, pp. 1897–1938, 2021, doi: 10.1007/s11227-020-03325-8.

[11] R. Chinchwadkar, V. Ingale, and A. Gokhale, “Hardware implementation of histogram-based algorithm for image enhancement,” in *Applied Computer Vision and Image Processing*, 2020, pp. 60–68. doi: 10.1007/978-981-15-4029-5_6.

[12] D. Ngo, S. Lee, Q.-H. Nguyen, T. Ngo, G.-D. Lee, and B. Kang, “Single image haze removal from image enhancement perspective for real-time vision-based systems,” *Sensors (Basel)*, vol. 20, no. 18, p. 5170, 2020, doi: 10.3390/s20185170.

[13] X. Dong *et al.*, “Fast efficient algorithm for enhancement of low lighting video,” in *2011 IEEE International Conference on Multimedia and Expo*, Jul. 2011, pp. 1–6. doi: 10.1109/ICME.2011.6012107.

[14] Y. Ren, Z. Ying, T. H. Li, and G. Li, “LECARM: low-light image enhancement using the camera response model,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 968–981, 2019, doi: 10.1109/TCSVT.2018.2828141.

[15] P. Soma and R. K. Jatoth, “Hardware implementation issues on image processing algorithms,” in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, Dec. 2018, pp. 1–6. doi: 10.1109/CCAA.2018.8777564.

[16] P. R. Schaumont, *A Practical Introduction to Hardware/Software Codesign*. Springer US, 2013. doi: 10.1007/978-1-4419-6000-9.

[17] I. Chiuchisan, M. C. Cerlinca, A. D. Potorac, and A. Graur, “Image enhancement methods approach using verilog hardware description language,” in *11th International Conference on Development and Application Systems*, 2012, pp. 144–148.

[18] D. Budgen and P. Brereton, “Performing systematic literature reviews in software engineering,” in *28th international conference on Software engineering*, 2006, pp. 1051–1052.

[19] R. S. M. Alex, S. Deepa, and M. H. Supriya, “Underwater image enhancement using CLAHE in a reconfigurable platform,” in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–5. doi: 10.1109/OCEANS.2016.7761194.

[20] S. Timarchi, M. A. Alaei, and H. Koushkbaghi, “Novel algorithm and architectures for high-speed low-power ConText-based steganography,” in *2017 19th International Symposium on Computer Architecture and Digital Systems (CADS)*, Dec. 2017, pp. 1–6. doi: 10.1109/CADS.2017.8310733.

[21] N. M. Yusoff, I. S. A. Halim, N. E. Abdullah, and A. A. A. Rahim, “Real-time hevea leaves diseases identification using sobel edge algorithm on FPGA: A preliminary study,” *2018 9th IEEE Control and System Graduate Research Colloquium, ICSGRC 2018 - Proceeding*, no. August, pp. 168–171, 2019, doi: 10.1109/ICSGRC.2018.8657603.

[22] C. Torres-Huitzil, “A review of image interest point detectors: from algorithms to FPGA hardware implementations,” in *Image Feature Detectors and Descriptors*, vol. 630, 2016, pp. 47–74. doi: 10.1007/978-3-319-28854-3_3.

[23] D. Sangeetha and P. Deepa, “An efficient hardware implementation of canny edge detection algorithm,” in *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, Jan. 2016, pp. 457–462. doi: 10.1109/VLSID.2016.68.

[24] G. B. Reddy and K. Anusudha, “Implementation of image edge detection on FPGA using XSG,” in *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Mar. 2016, pp. 1–5. doi: 10.1109/ICCPCT.2016.7530374.

[25] A. Rupani, P. Whig, G. Sujediya, and P. Vyas, “Hardware implementation of IoT-based image processing filters,” in *Proceedings of the Second International Conference on Computational Intelligence and Informatics*, 2018, pp. 681–691.

[26] S. Taslimi, R. Faraji, A. Aghasi, and H. R. Naji, “Adaptive edge detection technique implemented on FPGA,” *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, pp. 1–12, 2020.

[27] P. Ambalathankandy, A. Horé, and O. Yadid-Pecht, “An FPGA implementation of a tone mapping algorithm with a halo-reducing filter,” *J Real Time Image Process*, vol. 16, no. 4, pp. 1317–1333, 2019, doi: 10.1007/s11554-016-0635-6.

[28] M. Ismaeil, K. Pritamdas, K. J. K. Devi, and S. Goyal, “Performance analysis of new adaptive decision based median filter on FPGA for impulsive noise filtering,” in *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)*, Apr. 2017, pp. 1–5. doi: 10.1109/IEMENTECH.2017.8076990.

[29] E. Kalali and I. Hamzaoglu, “Low complexity 2D adaptive image processing algorithm and its hardware implementation,” *IEEE Transactions on Consumer Electronics*, vol. 63, no. 3, pp. 277–284, Aug. 2017, doi: 10.1109/TCE.2017.014996.

[30] S. Sadangi, S. Baraha, and P. K. Biswal, “Efficient hardware implementation of switching median filter for extraction of extremely high impulse noise corrupted images,” in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, Oct. 2019, pp. 1601–1605. doi: 10.1109/TENCON.2019.8929543.

[31] A. Ishikawa, N. Fukushima, A. Maruoka, and T. Iizuka, “Halide and GENESIS for generating domain-specific architecture of guided image filtering,” in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2019, pp. 1–5. doi: 10.1109/ISCAS.2019.8702260.

[32] N. Mohammed, M. Salih, R. A. A. Raof, Q. Hussein, and N. A. Khalid, “Design and implementation image processing functional unit using spatial parallelism on FPGA,” *ARPN Journal of Engineering and Applied Sciences*, vol. 13, pp. 4514–4520, 2018.

[33] L. Kabbai, A. Sghaier, A. Douik, and M. Machhout, “FPGA implementation of filtered image using 2D Gaussian filter,” *International Journal of Advanced Computer Science and Applications*, vol. 7, 2016, doi: 10.14569/IJACSA.2016.070771.

[34] H. Jing and X. Xiaoqiong, “Sports image detection based on FPGA hardware system and particle swarm algorithm,” *Microprocess Microsyst*, vol. 80, p. 103348, 2021, doi: https://doi.org/10.1016/j.micpro.2020.103348.

[35] P. G. Patel, A. Ahmadi, and M. Khalid, “Implementing an improved image enhancement algorithm on FPGA,” in *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2021, pp. 1–6. doi: 10.1109/CCECE53047.2021.9569049.

BIOGRAPHIES OF AUTHORS

Zul Imran Azhari received the B. Eng. degree (Hons.) in Electrical and Elcetronic engineering from the Universiti Teknologi Mara (UiTM), in 2021. He is now an engineer at Intel Microelectronics (M) Sdn. Bhd. He can be contacted at email: zulimran9@gmail.com.

Samsul Setumin received the B.Eng. degree (Hons.) in electronic engineering from the University of Surrey, in 2006, and the M.Eng. degree in electrical-electronic and telecommunication from the Universiti Teknologi Malaysia, in 2009. He obtained his Ph.D. degree from Universiti Sains Malaysia in 2019 in the imaging field. Since 2010, he has been a Lecturer with the Universiti Teknologi MARA, Malaysia. He was a Test Engineer with Agilent Technologies (M) Sdn. Bhd., and Intel Microelectronics (M) Sdn. Bhd., for a period of one year. His research interests include computer vision, image processing, pattern recognition, and embedded system design. He can be contacted at email: samsuls@uitm.edu.my.

Anis Diyana Rosli received her first honour degree from UiTM Shah Alam in 2007 and Msc from University of New South Wales in 2009. She is now a part time student in Doctorate program and her research is on heavy metal detection via electrochemical sensor. At the same time, she is also a senior lecturer in Universiti Teknologi MARA, Pulau Pinang and have served for 11 years since 2010.She can be contacted via email at anis.diyana@uitm.edu.my.

Siti Juliana Abu Bakar Received her B.Eng degree (Hons.) in electronic engineering from UTeM, Malaysia in 2019 and Master ESDE (Electronic System Design Engineering) from USM, Malaysia, in the year 2015. Completed her Ph.D. in the field of Automation & Control System from Universiti Sains Malaysia, Engineering Campus in 2020. She is currently working as senior lecturer at Centre for Electrical Engineering Studies, Universiti Teknologi MARA Campus Pulau Pinang Malaysia. Prior working as a senior lecturer at UiTM, she worked at Intel Product (M) Sdn.Bhd more than 5+ years as Validation Engineer. She can be contacted at email: sitijuliana@uitm.edu.my.