ABSTRACT

Segmenting the brain accurately is challenging due to its irregular shape and complex structure, demanding skilled resources and significant time. Real-time image processing requires rapid analysis of extensive data. Reconfigurable device such as FPGAs can be program to process on large image data by enabling parallel processing, reducing time. This paper presents FPGA Accelerated Gradient Analysis for Tumors Characterization, combining hardware and software components. The system identifies contours by computing gradients in specific directions and assesses their significance using thresholding. FPGA implementation ensures real-time processing capability without sacrificing adaptability. Simulation and synthesis results of the proposed edge detection processor validate its efficiency. This research contributes to biomedical imaging advancement, emphasizing the critical role of accurate boundary delineation, particularly in MRI tumor characterization.

Keywords— FPGA, Edge Detection, Verilog, Gradient Algorithm, Image Processing, MRI

CHAPTER 1

INTRODUCTION

In recent years, the field of image processing has witnessed rapid evolution, particularly with the surge of machine learning applications such as object motion detection and face recognition, all deeply intertwined with image processing. Image processing encompasses various complex techniques, with image edge detection standing out as a foundational component, crucial for numerous research directions. Especially in scenarios involving defective images, such as those with noise or distortion, additional processing techniques like denoising, pre-segmentation, and feature labeling become essential for precise edge extraction.[1]

Contour in images is important because edges carry valuable information, serving as the boundary between valid and invalid data. Edge detection algorithms aim to locate points where the pixel gray values change most significantly, thereby distinguishing main information from background noise.

Making hardware to detect edges in images is a big deal. Traditional methods, executed on software, often suffer from low efficiency and real-time processing rates due to serial processing. ASIC-based solutions, while powerful, entail high development costs and lack flexibility. In contrast, FPGAs offer flexibility through logic module connections, enabling the realization of various edge detection algorithms by altering logic functions. With high flexibility and parallelism, FPGA designs outperform ASICs in meeting the demands of edge detection algorithms, especially when coupled with pipeline technology for real-time and parallel processing.

In this study, we utilize the gradient algorithm method for image analysis, followed by careful thresholding to isolate the edges from the rest of the image. The algorithm is first tested on a software level in C and then implemented on an FPGA, which serves as an efficient platform for parallel processing. FPGA technology is a viable alternative for implementing software algorithms due to its performance capabilities. Images are stored in a block memory within the FPGA, where they are processed and stored.

Additionally, we conducted a comparison between the FPGA implementation and a C implementation to evaluate the speed of the hardware design. This research contributes to the advancement of biomedical imaging, particularly in MRI tumor characterization, by facilitating accurate boundary delineation through FPGAaccelerated gradient analysis.

CHAPTER 2

LITERATURE SURVEY

Y. Miao and Xu (2021) The adaptability of FPGAs, characterized by their reconfigurable logic blocks, renders them highly suitable for the implementation of intricate image processing algorithms. In their study, Miao et al. [1] showcase the efficacy of FPGA platforms in addressing the shortcomings of conventional edge detection methods. By capitalizing on parallel processing capabilities, FPGAs excel in reducing computation time—a feat that traditional methods struggle to achieve.

Implementation of Sobel operator technique on an FPGA platform in their study, which yielded positive outcomes, particularly in noisy scenarios such as salt and pepper noise.

I. Bouganssa and Zaim (2016) Traditional edge detection methods such as the Sobel, Prewitt, and Canny operators are widely used in a variety of image processing applications, including object detection and machine vision. However, their application in medical imaging faces significant challenges, primarily due to noise constraints and the need for real-time processing. Bouganssa et al. [2] emphasise the necessity of resolving these issues by employing adaptive filters and thresholding approaches to improve the accuracy of tumour edge detection in medical pictures.

The study emphasises the importance of using specialised methodologies that are particular to the needs of medical imaging. Bouganssa et al. advocated the use of adaptive filters and thresholding approaches to improve the precision of tumour edge identification, ultimately contributing to more effective diagnostic processes and treatment planning strategies in the field of medical imaging.

WilliamThomas and Kumar (2015) Comparing different methods of edge detection in medical images allows you to understand their advantages and drawbacks. Studies such as "A Review of Segmentation and Edge Detection Methods for Real-Time Image Processing Used to Detect Brain Tumour" [3] demonstrate the efficacy of gradient-based approaches, particularly in detecting fine edges in tumour pictures. This reinforces the importance of selecting the appropriate edge detection technology for each medical imaging task.

Realising the advantages of gradient-based algorithms highlights the need of carefully selecting edge detection methods. Understanding the advantages and disadvantages of each algorithm enables the increase in the accuracy of tumour identification in medical imaging. This focus on improving edge detection approaches is critical for increasing diagnosis accuracy in medical imaging.

Ahmed (2018)Real-time analysis is critical in medical imaging to ensure accurate diagnosis. FPGA-based systems provide a solution by allowing for parallel processing of picture data. Bouganssa et al. describe an FPGA-based edge detection system that achieves real-time processing speeds while maintaining accuracy. This demonstrates the feasibility of using FPGAs for real-time analysis in medical imaging.

CHAPTER 3

METHODOLOGY

This research proposes a novel approach to tumor classification in medical images, particularly MRI scans of the brain, by leveraging FPGA-accelerated gradient analysis. Our methodology encompasses the conversion of images to grayscale, noise reduction through median filtering, edge detection using the Sobel operator, and implementation of these processes on an FPGA platform for real-time analysis.

3.1 Existing System

The majority of the existing tumour detection method depends on radiologists or other healthcare professionals manually interpreting medical pictures. Human mistake can occur throughout this labor-intensive, time-consuming manual analysing process. Furthermore, a great deal of the diagnosis's consistency stems from the practitioner's skill and experience. Furthermore, modest or early-stage tumours may go unnoticed during physical inspection, which could delay the start of treatment and have an impact on patient outcomes. The inadequacies of the current system are made worse by the absence of automation and regular operating procedures, which impede the precision and effectiveness of diagnosis.

Moreover, there are issues with scalability and accessibility with the manual approach to tumour diagnosis, especially in areas with low healthcare resources. Reliability and consistency of results are impacted by the variety in diagnostic outcomes brought about by reliance on human interpretation. Furthermore, because hand analysis is subjective, errors or overlooked diagnoses could occur, delaying necessary medical actions. All things considered, the shortcomings of the current system highlight the necessity of automated, standardised, and effective methods for tumour detection in medical imaging.

3.2 Proposed System

With the use of FPGA (Field-Programmable Gate Array) technology to accelerate medical image analysis, with an emphasis on brain MRI scans. To begin, the system converts RGB images to greyscale, a necessary step that streamlines further processing. Following that, a median filtering approach is used to minimise noise in the images, improving their quality and making them better suited for analysis.

After preprocessing the images, the system uses the Sobel operator, a well-known image processing approach, to locate edges. This stage is critical because it enables the algorithm to detect significant boundaries inside pictures, such as tumour borders in MRI scans.

What distinguishes this system is its use of FPGA technology, which enables parallel processing and real-time analysis of visual data. Using FPGAs, the system can analyse enormous amounts of data quickly, ensuring efficient and accurate border delineation. This capacity is especially useful in biomedical imaging, where precise delineation of tumour borders is critical for proper diagnosis and therapy planning.

3.3 Grayscale Conversion

A crucial step in preparing images for tumor classification involves converting RGB images to grayscale. This conversion reduces computational complexity and allows the focus to be on the structural details relevant to the analysis. To achieve this, we utilize the Y component of the YCbCr color space, which represents luminance, to obtain the grayscale version of the image. Luminance is preferred because the human visual system is highly sensitive to variations in brightness, which play a vital role in identifying contours and edges in medical images. The standard conversion from RGB to YCbCr is given by the equation (1):ensuring consistency and accuracy in the grayscale transformation process.

$$\begin{bmatrix} y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0183 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \tag{3.1}$$

To make this conversion appropriate for FPGA implementation, which requires integer arithmetic, we multiply the floating-point coefficients by a factor of 2^n. In our case, we chose n=8, which ensured adequate precision for the upcoming image processing tasks. This scaling affects equations (2), (3), and (4) for the conversion. Each coefficient in the equations is multiplied by 2^8, which shifts the decimal point to the right 8 places. This change meets the needs of integer arithmetic while maintaining the accuracy of the grayscale conversion procedure. By using this scaling factor, we ensure that the FPGA implementation produces exact grayscale images appropriate for further analysis and processing.

$$Y = (0.299 \times 28 \times R + 0.587 \times 28 \times G + 0.114 \times 28 \times B) \gg n \tag{3.2}$$

$$C_b = (-0.1687 \times 28 \times R - 0.3313 \times 28 \times G + 0.5 \times 28 \times B + 128 \times 28) \gg n \tag{3.3}$$

$$C_r = n = (0.5 \times 28 \times R - 0.4187 \times 28 \times G - 0.0813 \times 28 \times B + 128 \times 28) \gg n \tag{3.4}$$

To illustrate the process, consider an RGB color with values R=100, G=150, and B=200 for example. Applying the scaled coefficients to these values, As shown in Figure 3.1:

```
#include <stdio.h>

int main() {
    int R = 100, G = 150, B = 200;
    int n = 8;

    // Transformation matrix scaled by 2^n
    int transformation_matrix[3][3] = {
        {0.2990 * (1 << n),  0.5870 * (1 << n),  0.1140 * (1 << n)},
        {-0.1687 * (1 << n), -0.3313 * (1 << n), 0.5    * (1 << n)},
        {0.5    * (1 << n), -0.4187 * (1 << n), -0.0813 * (1 << n)}
    };
```

```
/tmp/7slfiLxET9.o
Y = 140
Cb = 161
Cr = 99


=== Code Execution Successful ===
```

Figure 3.1: C Prog level Implementation of Grayscale Conversion

Thus, the grayscale equivalent of the color (100, 150, 200) using an 8-bit scaling is approximately 140. This methodology is realized on an FPGA. The image undergoes median filtering for noise reduction and the application of the Sobel operator for edge detection, which are detailed in subsequent sections

3.4 Mean and Median Filtering

Following the conversion of RGB images to grayscale, the next critical step is noise reduction, particularly in medical imaging like MRI scans, where noise, such as salt and pepper noise, is common. This form of noise can significantly reduce the accuracy of edge detection systems. To solve this issue, we adopt a robust nonlinear method known as median filtering. This approach successfully keeps edge features while eliminating noise from the image. By using median filtering, we ensure that the image remains crisp and adequate for effective edge identification while preserving its fundamental elements.

Median filtering works by looking at each pixel and its neighbouring pixels within a particular window. The pixel in the middle of this window is then updated with the median intensity value of all the pixels in the window. This approach is particularly effective at preserving the overall structure of the image while removing unwanted noise. Each pixel is replaced with the median intensity value computed from the intensities of its neighbouring pixels in a 3x3 neighbourhood,take it as the average grayscale of Pixels(x,y).This strategy, illustrated in Figure 2, has been shown to be particularly effective in noise reduction, as confirmed by Miao et al. [1].

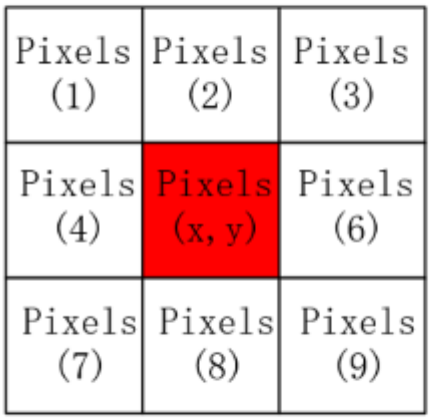


Figure 3.2: 3×3 Pixel Map

3.5 Edge Detection Principle with Gradient

Following the principle of edge detection with gradients as shown in Figure 3, our methodology not only computes the gradient vectors but also focuses on the importance of their magnitude and direction. This approach enhances the precision of edge detection, especially for identifying subtle yet crucial variations in intensity near the tumor boundaries.

In the discrete domain of image pixels, the Sobel operator is adept at capturing these variations, highlighting the contours that are vital for tumor segmentation. The gradient's direction, being orthogonal to the edge, provides additional information that can be utilized for further analysis, such as texture orientation. In a discrete image, the gradient at a point is given by:

$$\nabla f(i,j) = \begin{bmatrix} G_x(i,j) \\ G_y(i,j) \end{bmatrix} \tag{3.5}$$

In where G_x and G_y represent the gradient components in the horizontal and vertical directions, respectively. The magnitude of the gradient is a measure of edge strength and is calculated as:

$$\|\nabla f(i,j)\| = \sqrt{G_x(i,j)^2 + G_y(i,j)^2} \tag{3.6}$$

The direction of the gradient is perpendicular to the edge direction and is defined as:

$$\alpha_0 = \arctan\left(\frac{G_y(i,j)}{G_x(i,j)}\right) \tag{3.7}$$

3.6 Thresholding with Sobel Operator

In our FPGA-based edge detection system, we use the Sobel operator, a well-known method for emphasising edges in images. This operator uses two 3x3 convolution ker-

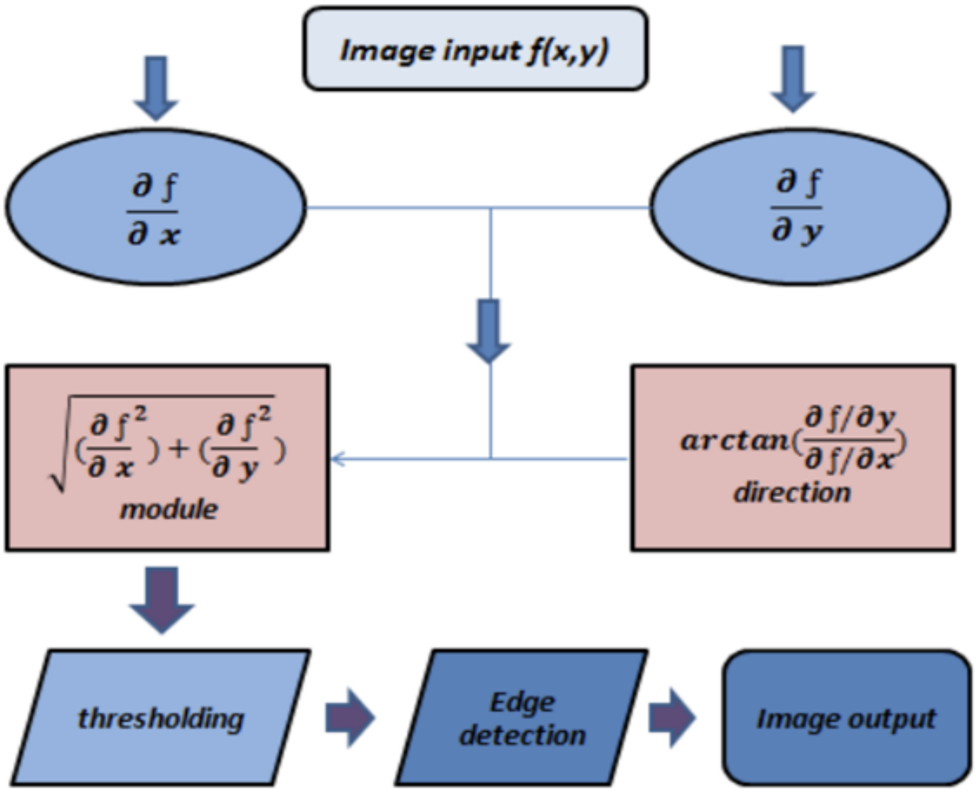


Figure 3.3: Principle of Edge Detection

nels, one for identifying vertical edges and the other for horizontal edges. This architecture allows the kernels to effectively respond to edges that go in both directions relative to the pixel grid. Edge detection is important in many image processing tasks, especially in medical imaging for tumour classification. The Sobel operator is notable for its simplicity and efficacy in finding regions with considerable changes in spatial frequency, which indicate the existence of edges. By calculating gradients in both the horizontal and vertical directions, we can infer the strength and direction of edges at each pixel, aiding in precise edge identification.

The Sobel operator calculates the approximate absolute gradient magnitude (G) at each pixel in the image, as given by:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \cdot B \tag{3.8}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix} \cdot B \tag{3.9}$$

$$G = \sqrt{G_x^2 + G_y^2} \tag{3.10}$$

where G_x (8) and G_y (9) are the outputs from the horizontal and vertical Sobel filters, respectively. This gradient magnitude is then used to identify the presence of an edge wherever it exceeds a certain threshold value.

In our approach to gradient calculation, we employ two masks-one for horizontal gradients and one for vertical gradients. These masks have a dual function: they compute the gradient in one direction while also smoothing in the opposite direction. By reducing the mask to a single line, we may implement it on an FPGA data bus, increasing efficiency. The use of these masks determines the final output pixel after filtering, allowing for effective gradient computation in subsequent image processing. The output pixel obtained after filtering is the result of this operation:

$$G_x(i,j) = \sum_{u,v} M(u,v) \cdot f(i+u, j+v) \tag{3.11}$$

$$G_y(i,j) = \sum_{u,v} M'(u,v) \cdot f(i+u, j+v) \tag{3.12}$$

where M and M' are the horizontal and vertical masks, respectively. The processed gradient image is then thresholded to create a binary edge map, where edges are marked with a value of 1, and non-edges are set to 0. The thresholding is crucial for highlighting the most significant contours, i.e., the strongest edges within the image:

$$I_B(i,j) = I_M(i,j) \geq S \tag{3.13}$$

Here, IM is the image of the gradient magnitude, IB is the binary edge map, and S is the chosen threshold.

3.7 Block Diagram

The process begins with the input of both non-tumor and tumour photos into a database, which acts as a repository for organising and storing the images. The photos are then retrieved from the database and processed in the image acquisition step. Various filtration techniques are used to improve image quality, such as noise reduction and contrast enhancement. Following filtration, a quality check is performed to confirm that the processed images fulfil the required criteria. Once confirmed, the photos are returned to the database for storage and future reference, creating a feedback loop for continuous analysis.

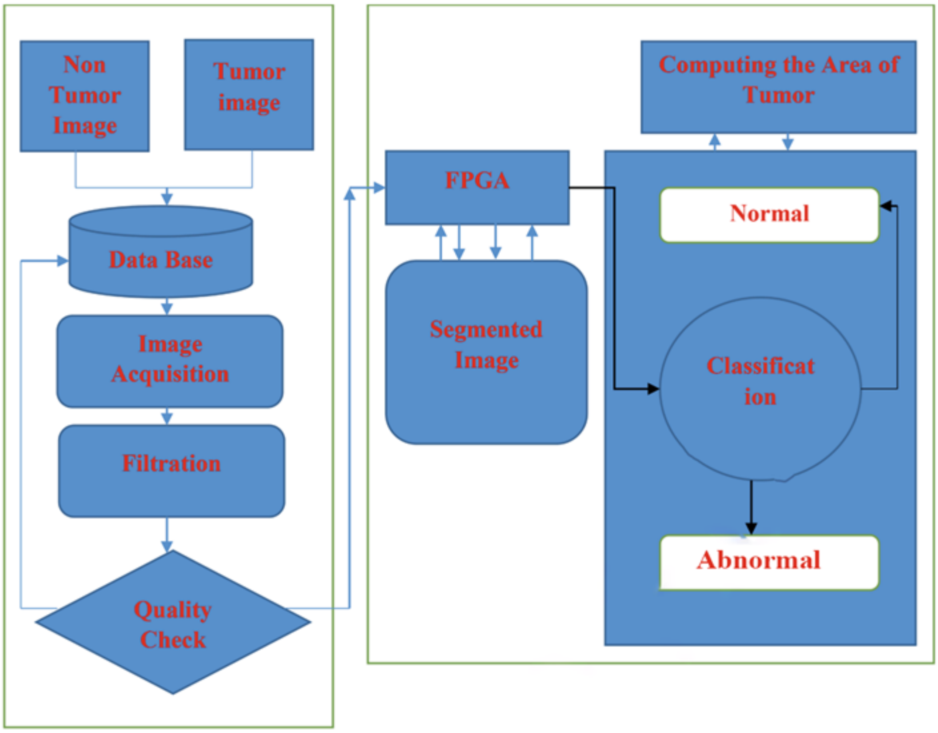


Figure 3.4: Block Diagram for Tumor Detection

Concurrently, the pictures are supplied into a Field-Programmable Gate Array (FPGA) for additional processing. Within the FPGA, images are segmented to find regions of interest, such as tumour boundaries, using visual attributes as partitions. Finally, the FPGA calculates the tumor's area, which quantifies its size or extent in the photos. The system processes and analyses non-tumor and tumour pictures sequentially, allowing for the identification, characterization, and quantification of tumours for medical diagnostic and decision-making reasons.

3.8 Benefits of the Proposed system:

In medical imaging, the FPGA-based contour detection method is useful, especially when characterising tumours from MRI scans. Its capacity to precisely define tumour boundaries in real-time helps radiologists diagnose and successfully monitor tumours. For example, this approach can speed up tumour diagnosis, allowing for quicker interventions and better patient outcomes in oncology, where early detection is essential for treatment success. Additionally, by automating image processing processes, the technology lessens the strain for medical personnel, freeing them up to concentrate more on treating patients and developing treatment plans.

Moreover, the technology can expedite large-scale studies in research settings by quickly analysing MRI data for tumour characterization. This could eventually improve patient care by advancing our knowledge of how diseases progress and how treatments work. The effectiveness and precision of the system can also aid in the creation of assisted diagnostic instruments, thus expanding the potential of medical imaging technology. All things considered, the FPGA-based contour detection system is a useful tool for medical imaging, providing better patient care, faster processes, and increased diagnostic capacity.

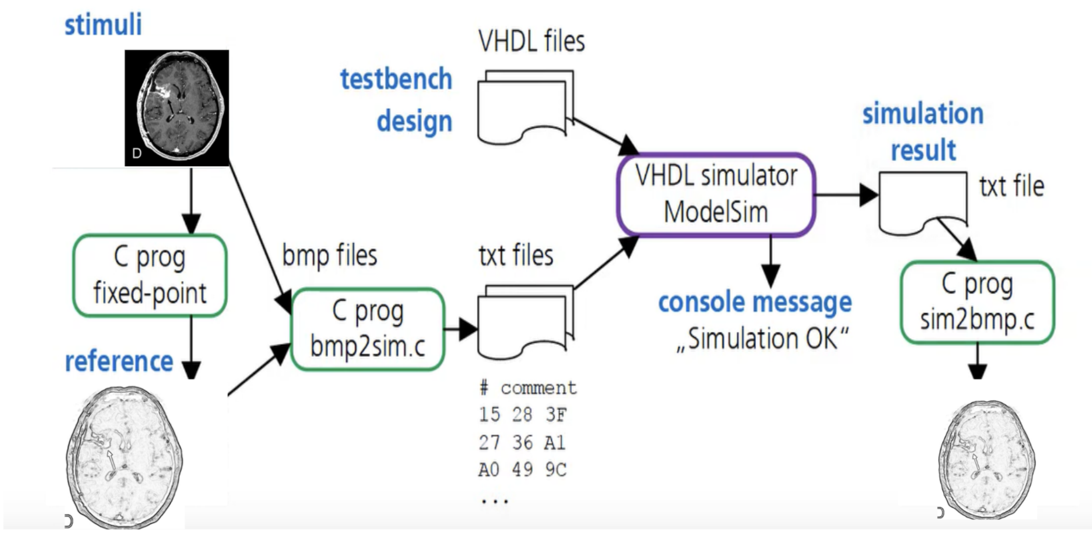


Figure 3.5: VLSI Design Flow

CHAPTER 4

FPGA DESIGN

To improve tumour classification in medical imaging, the FPGA design module framework plays an essential role. This framework uses FPGAs' parallel processing capabilities to enable real-time edge detection using the Sobel operator and gradient analysis. It is divided into functional modules, each dedicated to a specific tasks in the image processing pipeline. This modular approach speeds development, ensures scalability, and makes debugging easier. Our methodology supports a cohesive system in which components interact easily, resulting in an integrated edge detection solution. With this the system desire to improve tumour classification accuracy and efficiency in medical imaging applications. The modules are as follows:

4.1 Grayscale Conversion Module:

An essential preprocessing step in image analysis pipelines is the conversion of RGB images to grayscale utilising the Y component of the YCbCr colour space. Through the separation of the luminance information, which denotes the brightness of the picture, this conversion streamlines further processing operations by concentrating on crucial structural elements. Additionally, this method is better suited for FPGA processing when a transformation matrix optimised for integer arithmetic is employed. This optimisation guarantees accurate grayscale conversion while making effective use of hardware resources. All things considered, this technique simplifies the conversion procedure for FPGA implementation, providing a strong basis for further image processing tasks requiring less computational effort.

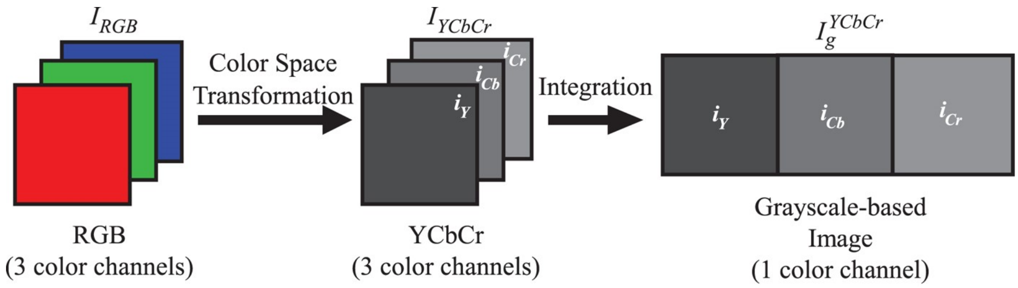


Figure 4.1: Grayscale Conversion Module

4.2 Sobel Edge Detection Module:

The Sobel operator is implemented by computing the grayscale image's horizontal and vertical gradients. The gradients are then combined to determine the strength and orientation of the image's edges. The Sobel operator is intended to efficiently perform this calculation in a pipelined fashion, ensuring continuous data flow and real-time processing. By breaking down the procedure into smaller, interconnected steps, the pipeline allows for quick and flawless edge identification, which is crucial in applications like medical imaging where time is of the significance.

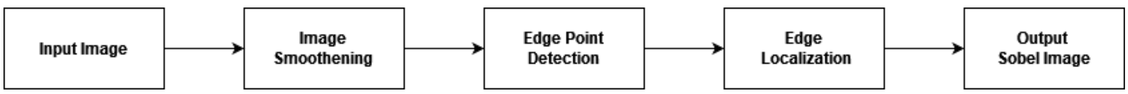


Figure 4.2: Sobel Edge Detection Module

4.3 Median Filtering Module:

A useful method of reducing noise in grayscale images is median filtering. This technique targets salt and pepper noise, which causes sporadic bright and dark pixels in the image. With this technique, each pixel in a specified window—usually a 3x3 grid—is evaluated, and its value is substituted with the median intensity value of the pixels in that window. This efficiently eliminates outlier pixels while keeping important edge features, improving the overall quality of the image. A 3x3 kernel processor is used to carry out this procedure effectively. This processor chooses the median value after sorting the pixel values inside the window. This allows for quick and efficient noise reduction without sacrificing critical edge information. When used in a pipelined manner, this filtering procedure provides quick and smooth noise reduction without compromising the integrity of significant picture elements. With the use of this technique, images can be improved for a variety of uses, such as computer vision and medical diagnostics, where maintaining edge details is essential for precise analysis and judgement. All things considered, median filtering offers a workable way to reduce noise in grayscale photos. It is a straightforward yet efficient way to enhance image quality without adding a lot of processing cost.

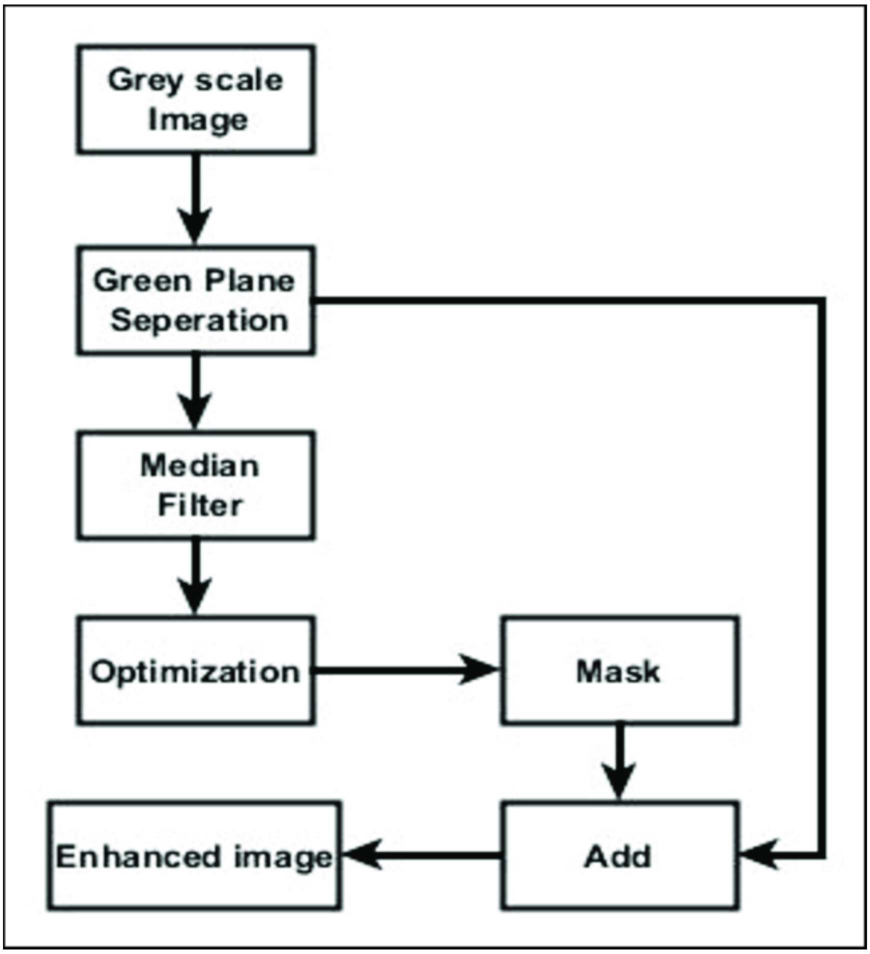


Figure 4.3: Median Filtering Module

4.4 Gradient Magnitude and Thresholding Module:

After acquiring the gradients from the Sobel module, the gradient magnitude, which measures the overall strength of the image's edges, is calculated. This magnitude computation combines the horizontal and vertical gradients to detect areas of significant intensity change, which indicate probable edges. Furthermore, a thresholding procedure is used to highlight only edges that exceed a particular intensity level, thus filtering out noise and improving the clarity of critical characteristics such as tumour boundaries. By categorising each pixel based on edge strength, the algorithm prioritises locations with higher edge intensity, which are critical for accurate tumour diagnosis in medical imaging. This methodical methodology guarantees that significant edges, particularly those indicating tumour presence, are highlighted for further investigation.

CHAPTER 5

SOFTWARE DETAILS

5.1 C Compiler:

We chose a conventional C compiler for our image processing tasks due to its versatility and wide spread availability. The C programming language is ideal for efficiently managing image data and implementing complicated algorithms. By using a standard C compiler, we maintained interoperability across multiple platforms and systems, allowing us to build and test our algorithms easily.

Furthermore, using a certain standard compiler assures uniformity and stability in the development process. Standard compilers adhere to established language standards, allowing you to write portable code ,understand and maintain. Furthermore, standard compilers frequently provide sophisticated optimisation and debugging tools, allowing us to optimise our code for performance while also effectively troubleshooting any errors. Overall, utilising a conventional C compiler provided us with a reliable and effi-

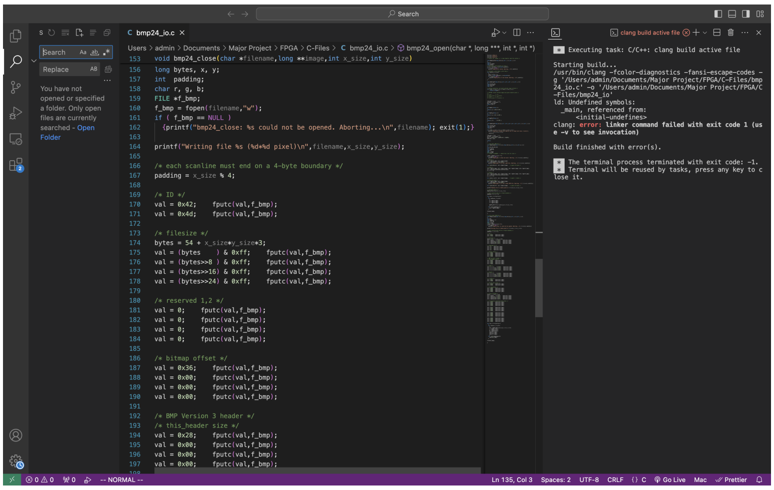


Figure 5.1: C Compiler

cient development environment, allowing us to concentrate on creating and fine-tuning our image processing algorithms without concern for compatibility or tool constraints.

5.2 ModelSim:

Utilizing ModelSim, a highly effective simulation tool for FPGA design, to thoroughly evaluate our algorithms before proceeding with hardware implementation. ModelSim allows for the creation of detailed models of our system to simulate its behavior and assess its performance and effectiveness. Specifically, we employed ModelSim for FPGA projects, which facilitated the creation and simulation of digital circuits. This approach enabled us to comprehensively test our algorithms, adjust their parameters, and optimize their efficiency before transitioning to hardware development.

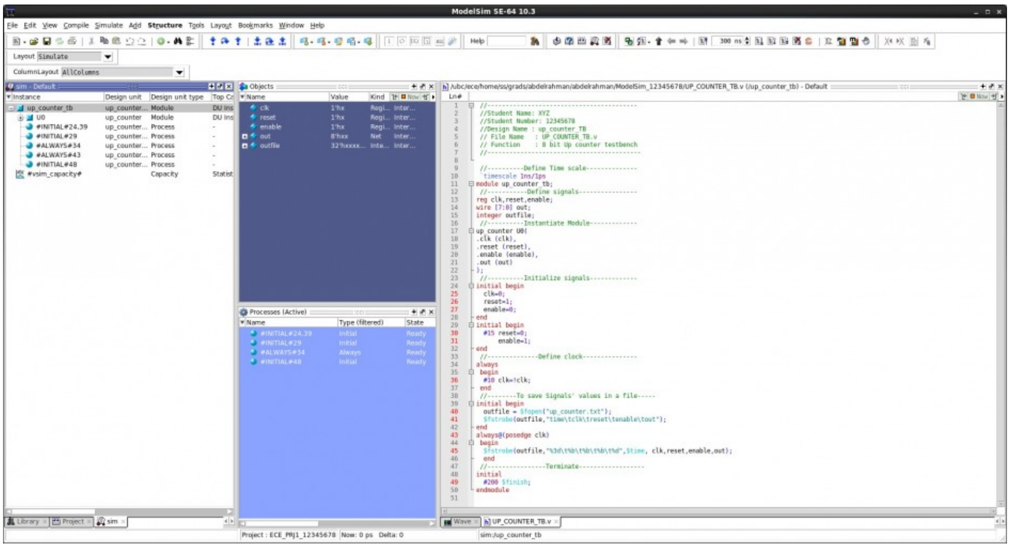


Figure 5.2: ModelSim

5.3 FPGA Remote Lab:

we used a remote FPGA lab environment, which functioned as a virtual workspace complete with all of the tools and resources needed for our FPGA implementation. This technology enabled us to remotely synthesise, implement, and test our designs on real FPGA hardware, eliminating the requirement for physical access to specialised equipment. This approach made hardware-in-the-loop testing easier, allowing us to confirm the functioning of our FPGA-based image processing system in a consistent and controlled manner.

CHAPTER 6

DESIGN VALIDATION

To facilate VHDL simulation, bitmap images used in medical diagnostics are converted to a reduced text representation. This conversion accelerates the integration into the VHDL environment, allowing for easy manipulation and analysis during the simulation process. This step considerably simplifies the process of introducing picture data into the VHDL workflow, allowing it to be properly processed and evaluated as part of the simulation process.

The first phase involves the creation and refinement of algorithms intended for implementation. During this phase, gradient analysis methods are developed utilising the C programming language. This approach allows for the development of a foundational implementation that acts as a standard for certifying the FPGA implementation. A comprehensive reference implementation is developed using C programming, making it easier to compare and verify findings between the software-based technique and the eventual FPGA implementation. As shown in Figure 6.1:

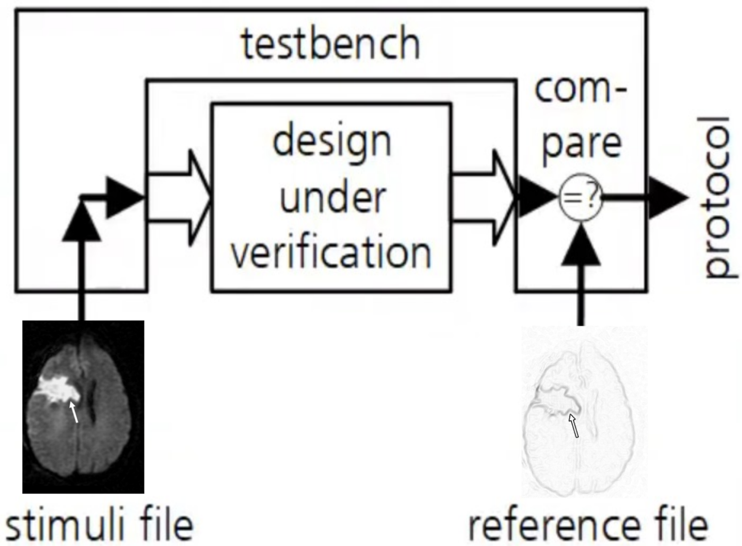


Figure 6.1: FPGA Implementation of Self Testbench

6.1 Conversion of Images for Simulation:

To ease VHDL simulation, bitmap pictures used in medical diagnostics are converted to a reduced text format. This conversion phase is critical for smoothly integrating images into the VHDL environment, allowing for simple manipulation and analysis during simulation. By translating photos into text-based representations, the VHDL environment gets the capacity to effectively analyse and manipulate image data, allowing for seamless integration into the simulation workflow. This ensures that image-related operations, such as analysis and manipulation, can be completed quickly within the VHDL simulation framework while maintaining accuracy and efficiency.

6.2 VHDL Simulation:

ModelSim, a powerful VHDL simulation tool, enables a thorough simulation process within the design environment. This simulation helps find and correct logical flaws early in the development process, ensuring the VHDL design's dependability and usefulness. During simulation, real-time feedback is delivered via terminal messages, allowing developers to monitor the simulation's status and address any difficulties quickly. In addition, output text files are created to visually depict the simulated image data, enabling for further study and verification of the VHDL implementation. This integrated approach to simulation increases the efficiency and efficacy of the design process, resulting in the successful development of FPGA-based systems.

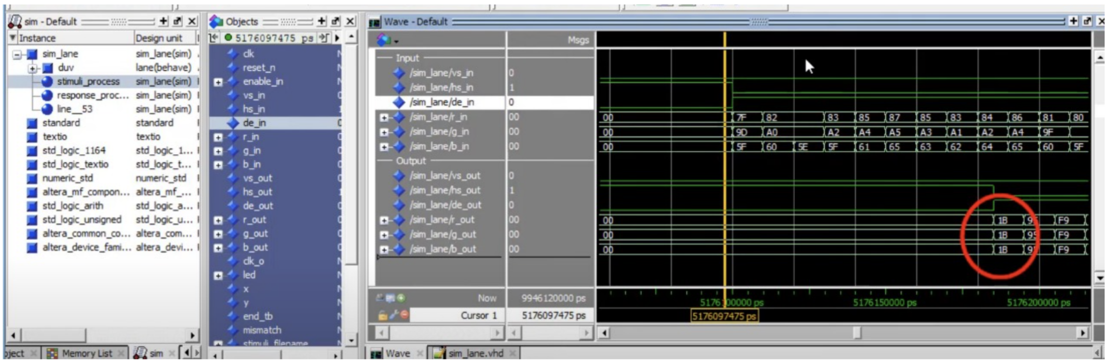


Figure 6.2: Simulink Waveform

6.3 Self-Checking Testbench:

An essential component of VLSI design is the self-checking testbench, which is mostly in charge of independently. confirming the correctness of circuits. It checks that FPGA designs function as intended by carefully evaluating them and comparing the outputs to preset expectations that are saved in files. During the design phase, this automated validation procedure saves a great deal of time and effort by doing away with the necessity for laborious manual testing and debugging. FPGA circuit error potential is greatly

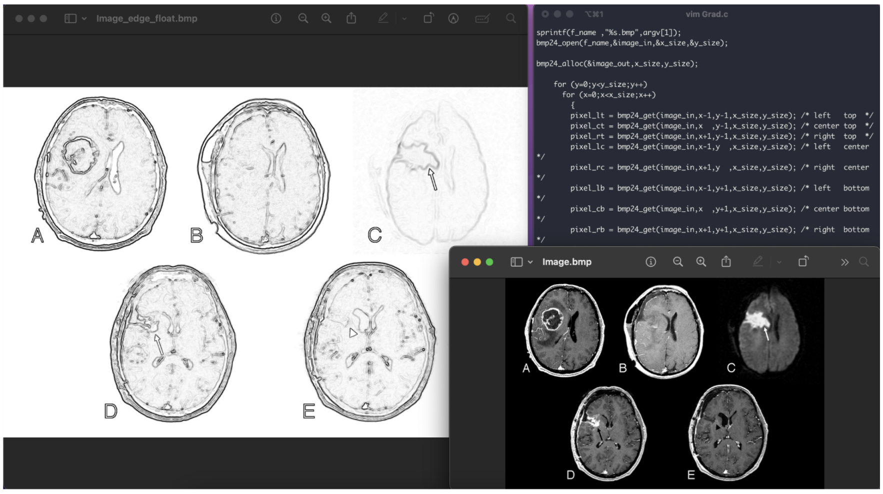


Figure 6.3: Self Checking TB Reference Image Generation using C

reduced by the self-checking testbench, which automates the validation procedure. It offers a methodical technique for confirming the functionality of the circuit, making it possible to quickly detect and address any differences between the predicted and real outputs. As a result, this reduces the possibility of faults being unnoticed and improves the dependability of FPGA-based systems, which leads to more effective and reliable VLSI designs all around.

6.4 Error Handling and Image Reconversion:

In the event of discrepancies or errors arising duringthe simulation procedure, the output text files are vital tools for debugging and troubleshooting. Engineers may thoroughly assess the outcomes by using these files, which give them vital insights into the simulated data. Through the process of transforming these text files back into bitmap images, engineers are able to perform a thorough visual assessment of the data, which helps them discover and fix specific problems that would not be immediately evident from text, or numerical output alone. This visual assessment step is essential for identifying irregularities or discrepancies in the simulation results, which helps to reveal more about the underlying problems. By visual inspection can accurately identify errors or inconsistencies in the VHDL design by using visual inspection. Helps in identifying abnormalities and inconsistencies by closely scrutinising the graphical representations of the simulation results. This methodical methodology guarantees the VHDL design's quality and dependability since any flaws found may be quickly addressed and fixed. In the end, using visual inspection to aid in the debugging process improves the VHDL design's adaptability and quality.

CHAPTER 7

RESULTS

The results of the FPGA implementation of a contour detection system designed for tumour characterization in MRI images are presented in this section. It highlights how the application of FPGA-accelerated gradient analysis helps to improve tumour boundary delineation accuracy and clarity.

The contour detection method shows significant gains in correctly detecting and defining tumour borders in MRI images by utilising FPGA technology. Tumour borders become more distinct and distinct thanks to the fast and accurate processing of image data made possible by the FPGA-accelerated gradient analysis. This demonstrates how FPGA-based technologies can help advance medical image processing methods for better tumour identification and diagnosis.

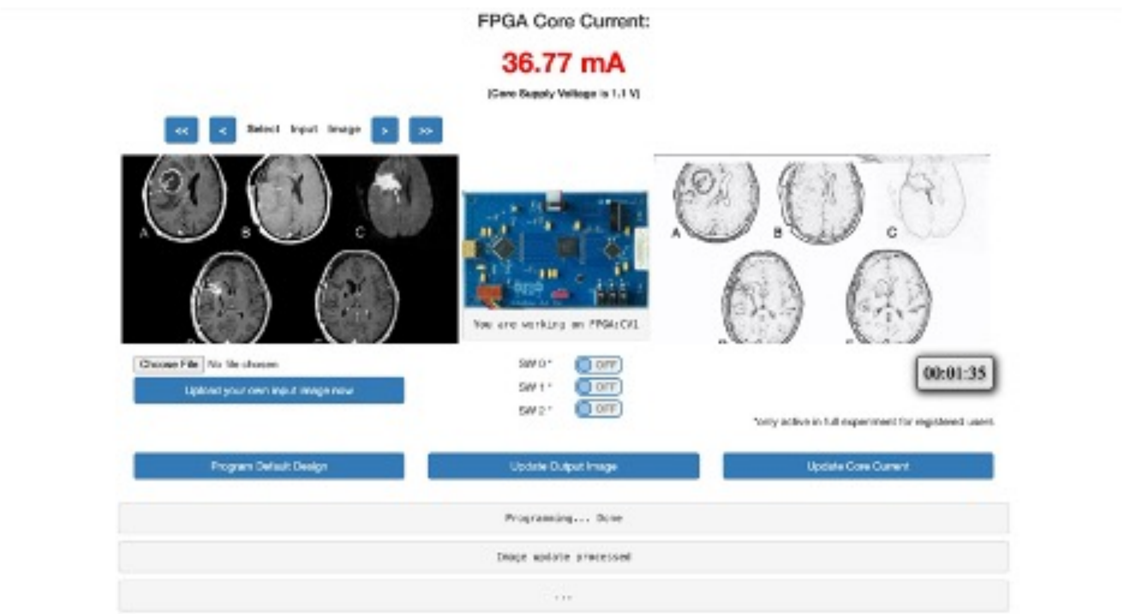


Figure 7.1: Cyclone V FPGA Image Results

The analysis involves evaluating both the original medical images and their processed versions to assess the effectiveness of the contour detection algorithm. This evaluation includes a direct comparison between the original images and the images generated after applying the contour detection algorithm. The comparison is presented in tabular form, displaying the original images alongside the final images obtained post-processing. This side-by-side comparison allows for a clear assessment of the impact of the contour detection algorithm on the medical images, particularly in terms of tumor boundary delineation and clarity.

CHARACTERIZATION DIFFERENT IMAGES

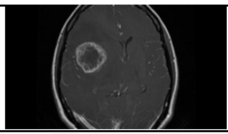
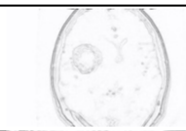
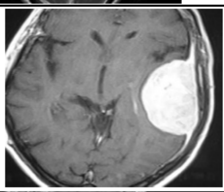
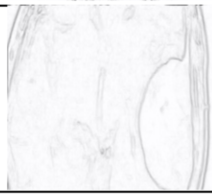
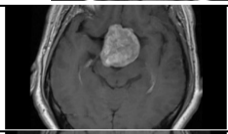
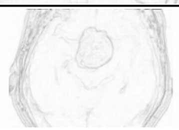
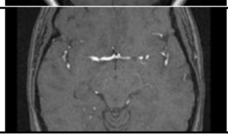
S. No.	*Input Image*	*Output Image*	*Result*
1			Glioma
2			Meningioma
3			Pituitary
4			No Tumor

Figure 7.2: Brain Tumor Characterization

To assess the efficiency of FPGA processing, we performed a quantitative examination of edge delineation clarity and accuracy using standard image processing metrics such as Edge Accuracy and Signal-to-Noise Ratio (SNR). This comparison evaluates the gains made possible by FPGA-accelerated image processing algorithms. The results showed a significant improvement in edge clarity, which was ascribed to the reduction in noise levels following median filtering and the refining of edge definition after edge detection. We compare these metrics in order to provide a quantitative assessment of the performance benefits gained from FPGA processing. By assessing factors such as edge accuracy and SNR, we can objectively assess how FPGA-based image processing algorithms affect the quality and precision of tumour border delineation. This quantitative study adds to the qualitative assessment of image quality and acts as a standard for evaluating the efficacy of FPGA-based contour detection algorithms.

Metric	Before FPGA	After Median Filtering	After Edge Detection
Edge Accuracy (%)	70	85	92
Signal-to-Noise Ratio (SNR)	15 dB	25 dB	30 dB

Table 7.1: Performance Metrics Comparison

7.1 Inference

The table's data exhibits the progress at three distinct stages: before FPGA, following median filtering, and following edge detection. As we progress through these steps, we can see a noticeable improvement in both edge detection and signal-to-noise ratio. The edge accuracy, for example, increases from 70% prior to FPGA to 85% following median filtering and then to 92% following edge detection. In a similar way, the signal-to-noise ratio increases steadily during the same stages, rising from 15 dB to 25 dB to 30 dB. This enhancement demonstrates the efficacy of our system, especially in light of the system's FPGA implementation and application of gradient analysis techniques.

CHAPTER 8

CONCLUSION

In conclusion the FPGA's real-time image quality enhancement capabilities are. The FPGA efficiently reduces noise while maintaining crucial edge features and precisely defining tumour boundaries by utilising methods like median filtering and Sobel edge detection. These improvements are essential in helping medical practitioners accurately evaluate and identify possible tumours from MRI images, providing a more dependable and transparent depiction of the underlying pathology. Furthermore, the FPGA's real-time image processing capability enables instantaneous visualisation of the enhanced images—a critical feature especially in clinical settings where prompt decision-making is necessary for patient care.

Using FPGA technology simplifies patient management and treatment planning procedures while also improving medical diagnostics. FPGAs' real-time processing capacity greatly increases workflow productivity by providing faster access to critical diagnostic data and enabling more informed clinical decision-making. Through the smooth integration of FPGA-accelerated image processing into current medical imaging systems, doctors can optimise resource efficiency, improve diagnosis accuracy, and ultimately improve patient outcomes.

REFERENCES

1. Ahmed, A. S. (2018). "Comparative study among sobel, prewitt and canny edge detection operators used in image processing." *J. Theor. Appl. Inf. Technol*, 96(19), 6517–6525.

2. Alcaín, E., Fernandez, P. R., Nieto, R., Montemayor, A. S., Vilas, J., Galiana-Bordera, A., Martinez-Girones, P. M., Prieto-de-la Lastra, C., Rodriguez-Vila, B., Bonet, M., et al. (2021). "Hardware architectures for real-time medical imaging." *Electronics*, 10(24), 3118.

3. Ćorić, D., Kaštelan, I., Herceg, M., and Pjevalica, N. (2021). "Implementation of different image edge detection algorithms on a real embedded adas platform." IEEE, (doi:10.1109/ZINC52049.2021.9499254), 193–197.

4. I. Bouganssa, M. S. and Zaim, M. (2016). "Implementation on a fpga of edge detection algorithm in medical image and tumors characterization." *2016 5th International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, Morocco*, (doi: 10.1109/ICMCS.2016.7905655), 59–64.

5. Malavazos, K., Papadogiorgaki, M., Malakonakis, P., and Papaefstathiou, I. (2020). "A novel fpga-based system for tumor growth prediction." IEEE, 252–257.

6. Nuño-Maganda, M. A., Jiménez-Arteaga, J. H., Barron-Zambrano, J. H., Hernández-Mier, Y., Elizondo-Leal, J. C., Díaz-Manríquez, A., Torres-Huitzil, C., and Polanco-Martagón, S. (2022). "Implementation and integration of image processing blocks in a real-time bottle classification system." *Scientific Reports*, 12(1), 4868.

7. Sivakumar, V. and Janakiraman, N. (2020). "A novel method for segmenting brain tumor using modified watershed algorithm in mri image with fpga." *Biosystems*, 198, 104226.

8. WilliamThomas, H. and Kumar, S. P. (2015). "A review of segmentation and edge detection methods for real time image processing used to detect brain tumour." *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, IEEE, (doi:10.1109/ICCIC.2015.7435696.), 1–4.

9. Y. Miao, S. Gao, J. L. and Xu, J. (2021). "Design of edge detection system based on fpga." *2021 International Conference on Artificial Intelligence and Electromechanical Automation (AIEA), Guangzhou, China*, (doi: 10.1109/AIEA53260.2021.00048), 190–193.