https://doi.org/ 10.33472/AFJBS.6.10.2024.5505-5514

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Innovative Cell Boundary Detection in Microscopic Images: Leveraging Intensity Patterns for Hardware Efficiency

Saravanakumar C[1*], Murugesan D[1], Pandian A[1], Marirajan S[1]

[1] Department of Electronics and Communication Engineering, SRM Valliammai Engineering College, Chengalpattu District -603 203, Tamilnadu, India

Corresponding Author[*]

Saravanakumar C

Department of Electronics and Communication Engineering, SRM Valliammai Engineering College, Chengalpattu District -603 203, Tamilnadu, India

kanchi.saravana@gmail.com

ORCID id: 0000-0003-2505-1317

Article History

Volume 6,Issue 10, 2024

Received:25 May 2024

Accepted : 01 Jun 2024

doi: 10.33472/AFJBS.6.10.2024.5505-5514

Abstract

Objective: The aim of this study is to develop and implement a straightforward algorithm for identifying cell boundaries in microscopic images, particularly focusing on the unique intensity patterns often present at cell membranes. This algorithm is designed for deployment on a hardware chip to evaluate its performance in terms of both segmentation accuracy and hardware efficiency.

Methods: This research addresses the challenge of cell segmentation in microscopic images by leveraging a simple, yet effective algorithm tailored to detect cell boundaries. Unlike conventional approaches that predominantly utilize the gradient field of the image, our method capitalizes on distinct intensity patterns at cell membranes. The algorithm is integrated into a hardware chip to experimentally test its efficacy. To optimize the use of hardware resources, a resource-sharing technique is applied, aiming to minimize the required area while maintaining accurate boundary identification.

Results: The proposed algorithm effectively isolates cell boundaries within microscopic images. When implemented on the hardware chip, it not only maintains high segmentation accuracy but also achieves a significant reduction in the area occupied by the hardware. Through the resource-sharing strategy, we successfully decrease the hardware area requirement by 60%, ensuring that the algorithm remains efficient and accurate in identifying cell boundaries.

Keywords: Cell segmentation, Microscopic imaging, Boundary detection, Intensity patterns, Algorithm development

Introduction:

Cells are the cornerstone of all living organisms, serving as the essential building blocks of human system. Across different domains, ability for image and analyze cells is critical, from understanding science behind cells and to facilitate discovery of drugs. Techniques used so far, are extensively employed to capture detailed structures of cells for in-depth analysis. One of the key processes in analyzing these images is cell image segmentation, which involves distinguishing cell regions from the background[1]. This segmentation is crucial for applications like cell tracking, which is integral to studying cellular behaviors such as guided movement and morphological changes[2].

In pictures containing few cells, the segmentation can be performed either manually or automatically. Given the enormous data volume in high-throughput studies, manual segmentation is often impractical due to its time-consuming nature[3]. This has led to a growing reliance on automated segmentation methods, spurred by advances in computational algorithms and machine learning in the fields of computer vision.

From a technical perspective, automatic cell image segmentation encompasses two main tasks: cell localization and cell border detection. Cell localization identifies the positions of cells within an image, a vital step for research areas like cell migration studies[4]. Meanwhile, cell border detection involves delineating the contours of cells as accurately as possible. High precision in identifying cell boundaries is essential for studies focusing on cell morphology[5]. Numerous techniques, such as active contour models and thresholding methods, have been developed to automate the segmentation process[6]. These methods often incorporate a range of algorithms with processing of image, like the GK convolution and BT transform, to enhance segmentation accuracy[7].

Despite these advances, challenges persist in achieving accurate cell segmentation[8]. Errors in segmentation can lead to extra segmentation, so that a single cell is erroneously divided into multiple segments, or less segmentation, where multiple cells are mistakenly combined into one[9]. Therefore, developing robust methodologies for precise cell boundary detection remains a critical area of research.

In this work, we introduce an innovative approach that leverages distinct intensity patterns at cell membranes for boundary detection in microscopic images. Unlike conventional methods that predominantly rely on the image's gradient field[10], our algorithm exploits these unique intensity characteristics to improve segmentation accuracy. Furthermore, this approach is implemented on a hardware platform using Field Programmable Gate Arrays (FPGAs) to test its performance and efficiency.

The increasing complexity of hardware design, particularly with programmable devices like FPGAs, poses significant challenges due to the rising number of transistors and the intricacy of the design process. To address this, we employ model-based hardware design tools that streamline the development process by translating high-level algorithmic descriptions directly into Hardware Description Language (HDL). This approach not only broadens the applicability of FPGA implementations but also enhances design productivity.

In our experiments, we program the proposed algorithm into FPGA chips and demonstrate its effectiveness in accurately isolating cell boundaries while significantly reducing hardware resource usage. By utilizing resource-sharing techniques, we achieve substantial reductions in the hardware area without compromising the accuracy of cell boundary identification.

The following sections of this paper are organized to detail the materials and methods used, present the results of our experiments, and discuss the implications and potential applications of our findings.

2. Materials and Methods:

2.1 Materials:

2.1.1. Image Data:

Cell images were sourced from the publicly available Kaggle dataset archive. This dataset provided a diverse range of cell images necessary for robust testing and evaluation of the segmentation algorithm.

2.1.2. Hardware:

The Kintex-7 Evaluation Board, a Field Programmable Gate Array (FPGA) kit, was used for the implementation and testing of the algorithm. This board is available through the Modernization and Removal of Obsolescence (MODROBS) program in the VLSI Lab of our institution.

2.1.3. Software:

A personal computer equipped with Xilinx System Generator software was employed to simulate the algorithm and interface with MATLAB’s Simulink environment.

2.2 Methods:

2.2.1. Image Pre-processing:

The cell images from the Kaggle dataset were loaded into the MATLAB Simulink Simulator via the Xilinx System Generator interface. To ensure consistency in analysis, all images were resized to a standard dimension. This step standardized the images, which varied in size across different sources, facilitating uniform processing and analysis.

2.2.2. Image Transformation:

The 2-dimensional cell images were converted into a 1-dimensional format using a 2D to 1D converter block. This transformation was achieved through a Frame Conversion block, which adjusted the sampling mode of the input signals based on the specified parameters. The block converted an Ai by Bi input matrix into a one by Bi output vector by unbuffering the rows sequentially.

Subsequently, the Unbuffer block converted the frame-based input signal into a sample-based output, producing individual time samples from each row of the input matrix. This conversion ensured that the sample period remained consistent between input and output signals, preparing the data for further processing.

2.2.3. Data Interpretation and Conversion:

The transformed 1-dimensional signal was passed to the processing unit through the Gateway-In block. This block interpreted the incoming signal as an empirical value and converted it to a signed fixed-point data type with a length of 8 bits. This conversion was crucial for aligning the data format with the requirements of the arithmetic processing unit.

2.2.4. Processing and Resource Optimization:

The core processing unit applied arithmetic operations (addition and subtraction) to the converted image data to identify the cell boundaries. To enhance the efficiency of these operations, a resource-sharing approach was implemented. This technique minimized the number of arithmetic operators required by sharing resources between similar operations that do not occur simultaneously. As a result, the use of multiplexers and other resources was significantly reduced.

In practice, the adders, subtractors, and multipliers were configured to share resources whenever possible, thereby optimizing the hardware design and reducing the overall complexity and area requirements on the FPGA.

2.2.5. Data Output and Reshaping:

The processed data, after arithmetic operations, was converted back to a Simulink-compatible data type using the Gateway-Out block. This block allowed the Fixed-Point Block set data type to be back propagated to the required Simulink data type for further processing.

To restore the processed data to an image format, a Reshape block was used. This block adjusted the dimensionality of the input signal to match the desired output format of 128 x 128 pixels. The final image was then buffered with a specified size of 128 x 128 to complete the transformation and prepare the output for further analysis or display.

2.2.6. Implementation and Verification:

The entire algorithm, from pre-processing to final reshaping, was programmed into the Kintex-7 FPGA. The implementation on the FPGA allowed for the experimental verification of the algorithm's performance in terms of both segmentation accuracy and hardware resource efficiency. The resource-sharing technique applied during the arithmetic operations significantly reduced the hardware area requirement while maintaining the effectiveness of cell boundary detection.

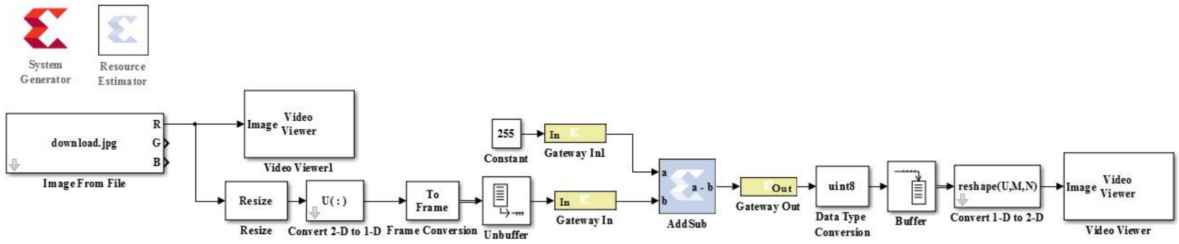


Figure 1: Block Diagram of Proposed Work

This structured approach ensured that the algorithm not only accurately identified cell boundaries but also operated efficiently within the hardware constraints, making it suitable for high-throughput cell imaging applications.

3. Results and Discussion:

The proposed algorithm and its hardware implementation were rigorously tested through simulations and practical experiments to evaluate their effectiveness in identifying cell boundaries and optimizing hardware resource usage.

3.1. Simulation and Algorithm Implementation:

The image processing workflow was first simulated using MATLAB Simulink. This environment provided a comprehensive platform to design, simulate, and verify the cell boundary detection algorithm.

The core arithmetic unit of the algorithm was developed and implemented in Verilog HDL, a hardware description language, to facilitate its integration into FPGA hardware. The resource-sharing optimization was incorporated into the arithmetic unit, and the unit was subsequently simulated using Xilinx Vivado 2019 software to evaluate its performance and resource efficiency.

3.2. Cell Image Processing:

A sample cell image from Shutterstock was processed to demonstrate the algorithm's ability to clearly delineate cell boundaries. Figure 2 shows the input and output images, where the boundary detection algorithm successfully identified and highlighted the cell's contours.

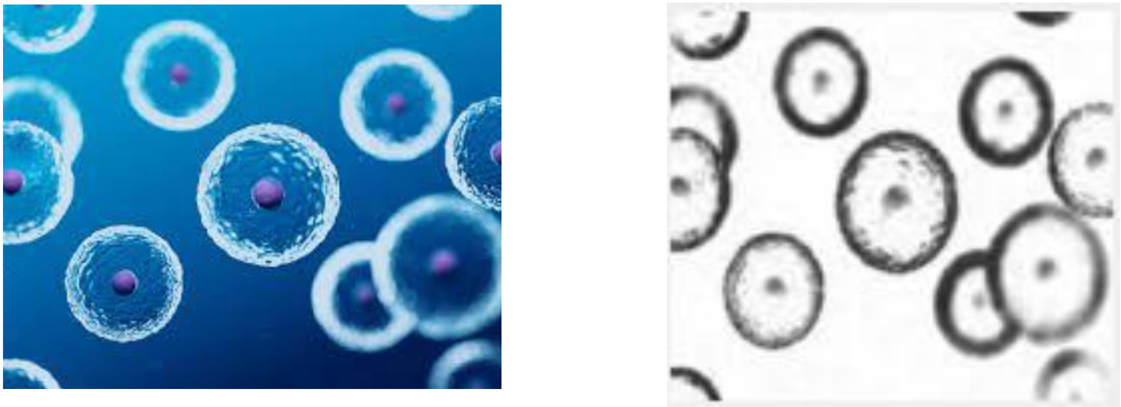


Figure 2: Input and output images of the cell from the proposed system.

Further testing was conducted using a cell image from the Kaggle dataset, specifically an Electron tomography image of African Green Monkey epithelial cells infected with SARS-coronavirus. The algorithm performed consistently, accurately detecting the cell boundaries as depicted in Figure 3.

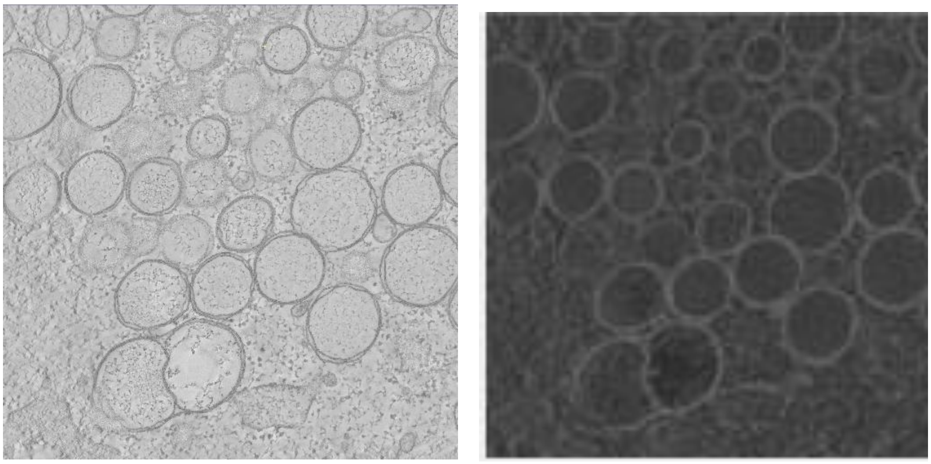


Figure 3: Input and output images of the African Green Monkey epithelial cells.

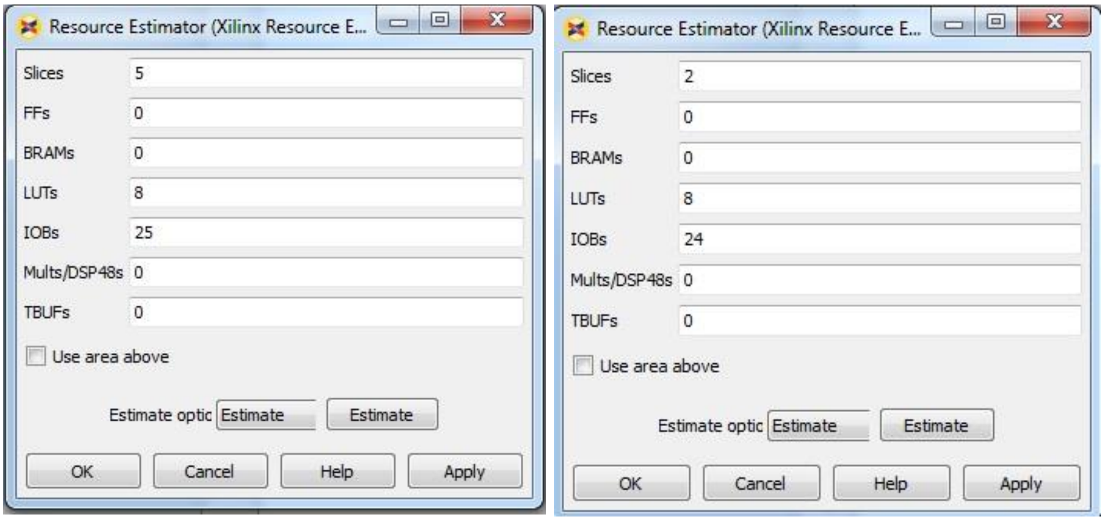
3.3. Hardware Resource Utilization:

The resource utilization was evaluated by comparing the hardware resources required for the algorithm with and without the resource-sharing technique. Table 1 provides a detailed comparison of the resources used.

Table 1: Comparison of Resources Utilized

Parameter	Existing	Proposed
Slices	5	2
LUTs	8	8
IOBs	25	24

As illustrated in Figure 4, the implementation with resource sharing significantly reduced the number of slices required, achieving a 60% reduction compared to the existing approach. This optimization demonstrates the efficiency of the proposed system in minimizing hardware resources while maintaining functional performance.



Resource Estimator (Xilinx Resource E...

Slices	5
FFs	0
BRAMs	0
LUTs	8
IOBs	25
Mults/DSP48s	0
TBUFs	0

□ Use area above

Estimate optic Estimate

Resource Estimator (Xilinx Resource E...

Slices	2
FFs	0
BRAMs	0
LUTs	8
IOBs	24
Mults/DSP48s	0
TBUFs	0

□ Use area above

Estimate optic Estimate

Figure 4: Resource utilization results of the existing and proposed system.

3.4 Discussion:

The proposed system effectively combines a simplified image processing algorithm with a resource-efficient hardware implementation to identify cell boundaries in microscopic images. This approach not only maintains high segmentation accuracy but also significantly reduces the hardware footprint.

3.4.1. Efficiency of Resource Sharing:

The adoption of a resource-sharing technique in the arithmetic unit proved to be highly effective. By allowing multiple operations to share the same hardware resources when not used simultaneously, the number of required slices in the FPGA was reduced by 60%. This optimization was achieved without compromising the precision or accuracy of cell boundary detection.

3.4.2. Accuracy and Consistency:

The algorithm demonstrated consistent performance across different types of cell images, from various sources like Shutterstock and Kaggle. This versatility is crucial for applications in diverse biological studies, where image characteristics can vary widely. The processed images clearly revealed cell boundaries, validating the algorithm's capability to handle different intensity patterns and image conditions effectively.

3.4.3. Practical Applications:

This approach is particularly valuable in high-throughput biological imaging, where large volumes of cell images need to be processed quickly and accurately. The reduced hardware requirements also make the system suitable for portable and cost-effective imaging solutions, potentially benefiting research and clinical applications.

3.4.4. Future Directions:

Future work could focus on further optimizing the algorithm for real-time processing capabilities and extending the approach to more complex and varied cell imaging scenarios. Integrating advanced machine learning techniques could also enhance the robustness and adaptability of the segmentation algorithm.

4. Conclusion:

The field of cell image segmentation faces considerable challenges due to the intricate and varied nature of cellular structures. Accurate boundary delineation is essential for numerous applications, including drug discovery and morphological analysis. Existing methods often focus on cell localization and struggle with precise boundary detection, especially in densely packed or overlapping cells, leading to issues like over-segmentation or under-segmentation. These methods also tend to be computationally intensive and require extensive manual adjustments, making them less suitable for high-throughput environments.

Our proposed system addresses these challenges by enhancing boundary detection and optimizing hardware resource usage. The algorithm leverages distinct intensity patterns at cell membranes, improving the accuracy of boundary identification compared to traditional methods that rely on gradient fields. Implemented on an FPGA, the system uses a resource-sharing technique, reducing the number of slices required by 60%, resulting in a smaller chip area and lower power consumption. This makes the system scalable and portable for various applications.

Testing on diverse cell images from sources like Shutterstock and Kaggle demonstrated the system’s robustness and consistency in producing clear and accurate boundaries. This performance is crucial for applications in diverse biological research settings. Moreover, the optimized resource usage led to a reduction in operating speed without compromising accuracy, which is vital for high-throughput applications where rapid and reliable segmentation is needed.

Future enhancements could include integrating machine learning to recognize a broader range of boundary patterns, potentially improving accuracy and enabling real-time processing for dynamic applications like live cell imaging. Additionally, expanding the system to handle more complex imaging modalities, such as 3D microscopy, could significantly broaden its impact.

In conclusion, our system represents a significant advancement in cell image segmentation, combining a simple yet effective algorithm with efficient hardware implementation. This approach promises to improve accuracy and resource efficiency, supporting a wide range of applications in quantitative single-cell biology and beyond.

5. References

1. Wang Y. Zhang Z. Wang H. Bi S Segmentation of the Clustered Cells with Optimized Boundary Detection in Negative Phase Contrast Images. PLoS ONE. 2015; 10(6): e0130178.doi.org /10.1371 / journal.pone.0130178

2. Sotiris Dimopoulos. Christian E Mayer. Fabian Rudolf. Joerg Stelling Accurate cell segmentation in microscopy images using membrane pattern. Bioinformatics. 2014; 30(18):2644–2651doi.org/10.1093/bioinformatics/btu302

3. Al-Kofahi Y. Zaltsman A. Graves R. et al A deep learning-based algorithm for 2-D cell segmentation in microscopy images. BMC Bioinformatics. 2018; 19(365)doi.org/10.1186/s12859-018-2375-z

4. M Syahrir. A Kuncoro. A Ari Kurniawan. M Arief Soeleman and G Fajar Shidik.Image Enhancement Segmentation and Edge Detection in MRI for Mammogram Disease. International Seminar on Application for Technology of Information and Communication .2018;515-521doi: 10.1109/ISEMANTIC.2018.8549801.

5. Chandrasekaran S. Nageswaran U B A mutated addition–subtraction unit to reduce the complexity of FFT. Applied Nanoscience. 2023;doi.org/10.1007/s13204-021-02278-5

6. B Ronak and S A Fahmy Multipumping Flexible DSP Blocks for Resource Reduction on Xilinx FPGAs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.2017;36(9):1471-1482 doi: 10.1109/TCAD.2016.2629421.

7. M G Vayada. H R Patel and B R Muduli Hardware software co-design simulation modeling for image security concept using Matlab-Simulink with Xilinx system generator. 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS). 2017;134-137 doi: 10.1109/SSPS.2017.8071580.

8. B S Durgakeri and G N Chiranjeevi. Implementing Image Processing Algorithms using Xilinx System Generator with Real Time Constraints. 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT). 2019; 230-234, doi: 10.1109/RTEICT46194.2019.9016962.

9. V Chaitanya Deepthi.P Surya Prasad. Medical Image Fusion using Xilinx System Generator in FPGA. International Journal of VLSI System Design and Communication Systems.2016;4(10), 990-993

10. Šušteršič, T. Milovanović V. Ranković V. Filipović N. Peulić A. Medical Image Processing Using Xilinx System Generator. In: Filipovic, N. (eds) Computational Bioengineering and Bioinformatics. ICCB 2019. Learning and Analytics in

Intelligent Systems.2020;11. Springer, Cham. https://doi.org/10.1007/978-3-030-43658-2_10

6. Conflict of Interest: No

7. Author Contribution: CSK conducted the investigation, DM collected data, and AP wrote the manuscript following statistical analysis. SM helped develop the topic.