JoCoSiR
Journal of Computer Science Research
ISSN: 2986-2337(Online)

Implementation and Performance Analysis of Basic and Advanced Image Processing Techniques Using Python and OpenCV

Syaiful Rahman Lubis [a], Muhammad Iqbal [b]

[a,b] *Magister Teknologi Informasi, Universitas Pembangunan Panca Budi, Jl. Gatot Subroto, Kec. Medan Sunggal, Kota Medan, Sumatera Utara 20122, Indonesia*

email: [a] syaifulrahmanlubis51@gmail.com, [b] wakbalpb@yahoo.co.id

ARTICLE INFO

Keywords: *Digital Image Processing, Opencv, Grayscale, Morphological Operation, Convolution*

IEEE style in citing this article: [citation Heading]
F. Fulan and F. Fulana, "Article Title," *JoCoSiR: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 7, no. 1, pp. 1-10, 2021. [Fill citation heading]

ABSTRACT

Abstract— Digital image processing plays a crucial role in artificial intelligence and computer vision, with widespread applications in healthcare, agriculture, security, industry, and transportation. This research focuses on implementing both basic and advanced image processing methods using Python and the OpenCV library within a desktop application. The main problem addressed is the lack of an integrated, structured approach that bridges basic and advanced techniques, limiting users' comprehensive understanding of image processing workflows. The objective is to design a complete system that allows step-by-step processing, starting from grayscale conversion, binarization, arithmetic and logical operations, to convolution and morphological transformations such as Sobel edge detection and erosion. The proposed application utilizes Tkinter for the user interface, enabling users to upload images, apply various processing techniques, and analyze results interactively. The system also includes histogram visualization and equalization to enhance contrast. Findings show that the implemented methods effectively transform images in accordance with theoretical expectations, such as edge enhancement and shape simplification. The integration of these methods into a single, user-friendly platform supports both educational and applied uses. The contribution of this research lies in its practical demonstration of digital image processing techniques, providing a comprehensive and accessible reference for developers, researchers, and students. Despite its achievements, the system lacks advanced segmentation and real-time capabilities, which are suggested for future development through integration of adaptive methods and machine learning techniques.

Copyright: Journal of Computer Science Research (JoCoSiR) with CC BY NC SA license.

1. INTRODUCTION

Digital image processing is an important field in artificial intelligence and computer vision that is widely used in various sectors, such as health (for tumor detection or radiological image analysis), agriculture (for crop classification and pest detection), security (for facial recognition and detection of dangerous objects), industry (for product quality inspection), to transportation (for autonomous vehicles and navigation systems) [1] ; [2]. The object of this study is the implementation of basic and advanced methods in digital image processing using the Python programming language and OpenCV library, which is known to be flexible and rich in visual processing functions.

The image processing methods that have been used previously are generally divided into two major groups, namely the basic method and the advanced method [3]. Basic methods include color conversion (e.g. RGB to grayscale), thresholding for simple segmentation, filtering (such as Gaussian blur and median filters) for noise reduction, as well as edge detection using operators such as Sobel, Prewitt, and Canny [4] ; [5]. This method is relatively simple to implement and has a high process speed because it does not require large computing resources. However, its disadvantages lie in its sensitivity to lighting, noise, and the complexity of objects, making it less accurate for complex real-life cases [6].

Meanwhile, advanced methods in image processing use more complex techniques such as contour-based segmentation, object detection using the Haar Cascade Classifier, and the application of deep learning-based models such as Convolutional Neural Networks (CNN) for image classification and object detection [7] ; [8]. This advanced method has advantages in terms of accuracy and flexibility to variations in image data, as well as the ability to recognize complex patterns [9]. However, the implementation of this method requires large amounts of datasets, time-consuming model training, and high-performance hardware [10] ; [11].

The main problem raised in this study is the lack of a comprehensive and structured approach that bridges basic and advanced methods in one integrated platform. Many studies or projects address only a small part of image processing methods, making it difficult for beginners, researchers, and practitioners to gain a comprehensive understanding of the actual implementation of the entire spectrum of image processing techniques.

As a solution to this problem, this study proposes a systematic approach in the form of designing and implementing various digital image processing methods using Python and OpenCV. This implementation includes pre-processing methods, image transformation, edge detection, segmentation, pattern recognition, and deep learning-based classification.

http://doi.org/10.65126/jocosir.v2i2.60
Journal of Computer Science Research (JoCoSiR) with CC BY NC SA license.

ISSN 2502-3357 (Online)

Each method is tested in real-world cases, such as object detection in environmental imagery, background and main object separation, and simple shape classification. The results of the implementation are studied in terms of effectiveness, efficiency, and complexity of computing, so as to provide a comprehensive picture that is useful for both academic and industrial purposes.

This research aims to produce a technical documentation that not only explains the theory but also demonstrates the practical steps of applying image processing methods, from the most basic to the most advanced stages, which can be used as a reference for the development of digital image-based artificial intelligence applications in an efficient and structured manner.

2. LITERATURE REVIEW

Digital image processing is a technique used to modify, analyze, and extract information from digital images. With the development of computing technology, the Python programming language and the OpenCV library have become key tools in the research and development of various image-based applications. The following is a literature review related to the features used in this study.

2.1 Grayscale

Grayscale conversion is one of the most basic pre-processing steps in image processing. The goal is to reduce the color image (RGB) to a single intensity channel so that it is easier to process. Research by [12] states that converting to grayscale simplifies the process of segmentation and edge detection. However, color information is missing, which is an obstacle in color-based classification. The standard conversion formula to grayscale is:

$$I_{(x,y)} = 0{,}2989 \text{ x } R + 0{,}5870 \text{ x } G + 0{,}1140 \text{ x } B \qquad (1)$$

2.2 Thresholding

Binary conversion converts the grayscale image into two pixel levels: 0 and 255. This operation is used in object segmentation. Deep [13] suggests that simple thresholding methods are less effective for images with uneven lighting. Adaptive thresholding or Otsu can be a solution. The basic formula where T is the threshold value:

$$I_{\text{binary}}(x,y) = \begin{cases} 255, \text{if } I(x,y) > T \\ 0, \text{if } I(x,y) \le T \end{cases} \qquad (2)$$

2.3 Arithmetic operations

Arithmetic operations are used to combine two images or perform pixel value transformations such as addition, subtraction, multiplication, and division. In implementing arithmetic operations in change detection between two satellite images [14].

Addition:

$$I_{\text{result}}(x,y) = I_1(x,y) + I_2(x,y) \qquad (3)$$

Reduction

$$I_{\text{result}}(x,y) = I_1(x,y) - I_2(x,y) \qquad (4)$$

2.4 Logic Operations

Logic operations are used to manipulate images based on binary logic such as AND, OR, XOR, and NOT [15]. This method is commonly used in object masking, especially to isolate specific areas of the image based on a specific pattern [16].

AND:

$$I_{\text{result}}(x,y) = I_1(x,y) \wedge I_2(x,y) \qquad (5)$$

OR:

$$I_{\text{result}}(x,y) = I_1(x,y) \vee I_2(x,y) \qquad (6)$$

2.5 Logic Operations

A histogram is a graphical representation of the pixel intensity distribution. It can be used for analysis and contrast enhancement (histogram equalization). Research by [17] Indicates that histogram equalization is effective in improving medical imaging.

$$s_k = (L-1)\sum_{j=0}^{k}\frac{n_j}{n} \qquad (7)$$

http://doi.org/10.65126/jocosir.v2i2.60
Journal of Computer Science Research (JoCoSiR) with CC BY NC SA license.

ISSN 2502-3357 (Online)

2.6 Convolution Filter

Convolution is used for the application of spatial filters in image processing, such as edge detection, smoothing, and sharpening where w(i,j) is the filter kernel. Research [18] Indicates that convolution filters are very effective for edge feature extraction in character recognition systems.

$$I_{result}(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(i,j)\ x\ (I(x+i, y+j) \quad (8)$$

Example of the Sobel kernel for horizontal edge detection:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (9)$$

2.7 Morphological Operations

Morphological operations such as dilation, erosion, opening, and closing are used in binary imagery to eliminate noise, fill holes, or enlarge objects [19]. Research [20] use it in object shape segmentation for highway surveillance imagery.

Dilation:

$$A \oplus B = \{z \mid (B)_z \cap A \neq \emptyset\} \quad (10)$$

Erosion:

$$A \oplus B = \{z \mid (B)_z \subseteq A\} \quad (11)$$

3. METHODOLOGY

This study applies the stages of digital image processing using the Python programming language with OpenCV library and Tkinter-based user interface. The main focus of this method is to provide basic and advanced features of image processing interactively. The method is designed with systematic steps described as follows:

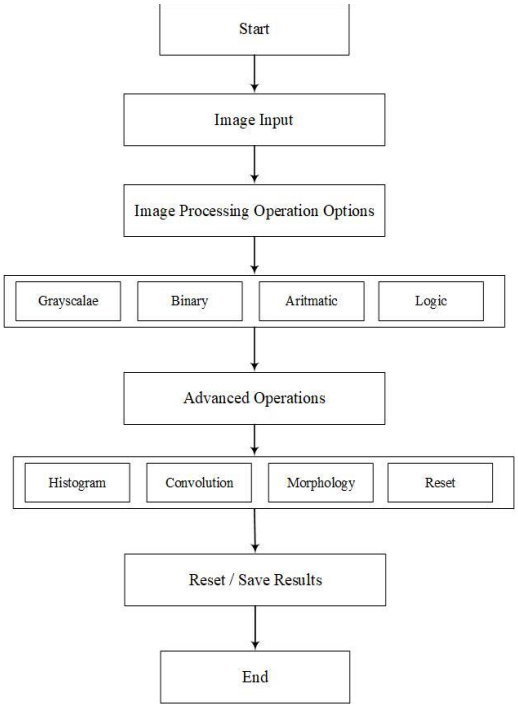

Figure 1. Research Flowchart

This study implements digital image processing methods which are divided into two main categories, namely basic surgery and advanced surgery. The entire process is done in a desktop application built with Python using the Tkinter library for the user interface and OpenCV for the image manipulation process.

The first step is to perform image input from the user's local storage. The selected image will be displayed in the app and used as the main object to be processed. Furthermore, the image can be converted into grayscale form. The goal of this conversion is to simplify the colors in the image to just gray scale, making advanced processing processes such as binerization or edge detection easier to do. After that, the grayscale image can be converted into a binary image. This process is done by specifying a specific threshold value, so that each pixel will be categorized as black or white. This operation is essential for applications such as object segmentation or shape analysis.

http://doi.org/10.65126/jocosir.v2i2.60
Journal of Computer Science Research (JoCoSiR) with CC BY NC SA license.

ISSN 2502-3357 (Online)

Next up is arithmetic operations, which allow users to perform addition, subtraction, multiplication, or division operations between two images. This operation is useful for stitching together information from two images or adjusting the exposure and contrast. Then, there are also logical operations such as AND, OR, XOR, and NOT that are used for further manipulation of binary images. This logic operation is very useful in the process of masking or merging specific areas of the image. In the advanced operations section, the system provides a histogram feature that displays the distribution of pixel intensity values in the image. In addition, there is also a histogram equalization function that functions to automatically increase image contrast by flattening the distribution of pixel intensity.

The next operation is convolution, which is the process of applying a filter or kernel into an image to produce a specific effect such as edge detection or smoothing. Some of the types of filters used include Sobel, Laplacian, and Gaussian. This operation is important in applications such as object edge detection and image quality improvement. Next is the morphological operation, which is applied to binary images to improve the shape or structure of an existing object. This surgery includes erosion, dilation, opening, and closing. For example, dilation is used to increase the thickness of an object, while erosion reduces the thickness to remove noise.

As an additional feature, the app also provides a reset button that allows users to restore the image to its original state before any process is performed. This makes it easy for users to retry the process or move to another method without having to reload the image from scratch. By implementing these stages, the system is able to provide a complete and easy-to-use range of image processing processes, both for basic and advanced image analysis needs.

4. RESULT AND DISCUSSION

This research resulted in a desktop-based application for digital image processing that is equipped with basic and advanced operating features. The application is built using the Python programming language, with support for the Tkinter library as a graphical interface and OpenCV for image processing. The test was carried out using different types of images, both color and grayscale, to observe the effectiveness of each feature that had been implemented.

Figure 2. Image Input

4.1 Basic Operations Implementation Results

Basic operations consist of grayscale conversion, binary conversion, arithmetic operation, and logic operation. The results of the grayscale conversion showed that the initially colored image was successfully converted to a grayscale image with the appropriate pixel intensity level. This shows that the cvtColor function of OpenCV works well to reduce the color dimension without omitting key visual information.

In the binary conversion process, the results show that the use of thresholds successfully groups pixels into two values, namely black and white. This is especially helpful for separating the main object from the background, especially in high-contrast images.

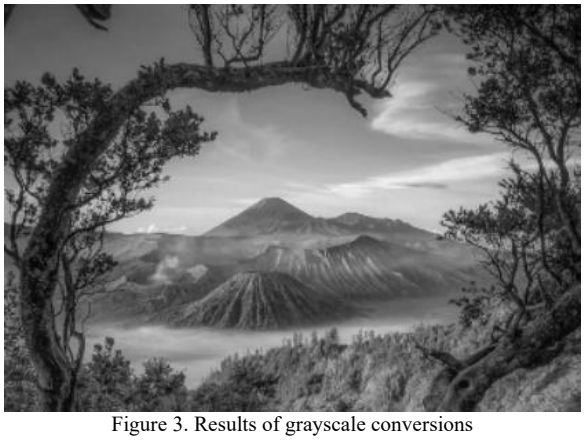

Figure 3. Results of grayscale conversions

http://doi.org/10.65126/jocosir.v2i2.60
Journal of Computer Science Research (JoCoSiR) with CC BY NC SA license.

ISSN 2502-3357 (Online)

Figure 4. Results of binary conversions

Arithmetic operations such as addition and subtraction between two images show significant differences in image exposure and contrast. For example, the addition of two images can increase the intensity of the pixels so that the image becomes brighter, while the subtraction can highlight the difference between the two images.

For logic operations, the results show that the AND, OR, XOR, and NOT functions can be used to manipulate shapes or structures in binary images. For example, an AND operation between two binary images successfully maintains an equally white area, while OR can unite the two objects in a single image.

Figure 5. Result of arithmetic operation with icrease brightness (+50)

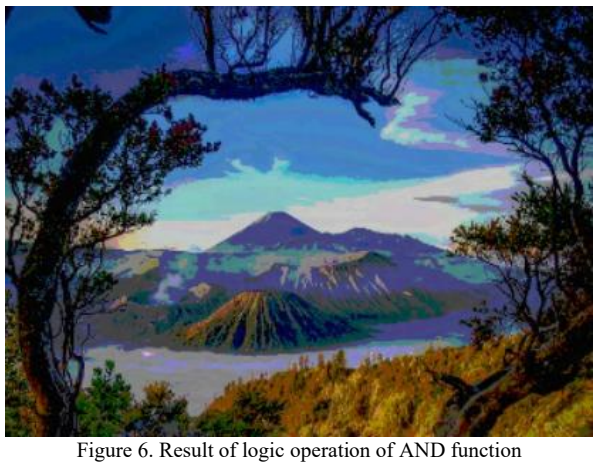

Figure 6. Result of logic operation of AND function

4.2 Advanced Operations Implementation Results

In the advanced operations section, the image histogram is successfully displayed visually and shows the distribution of pixel intensity values in the image. Histogram equalization provides more optimal results in improving image contrast, especially for images that have less than even lighting.

http://doi.org/10.65126/jocosir.v2i2.60
Journal of Computer Science Research (JoCoSiR) with CC BY NC SA license.

ISSN 2502-3357 (Online)

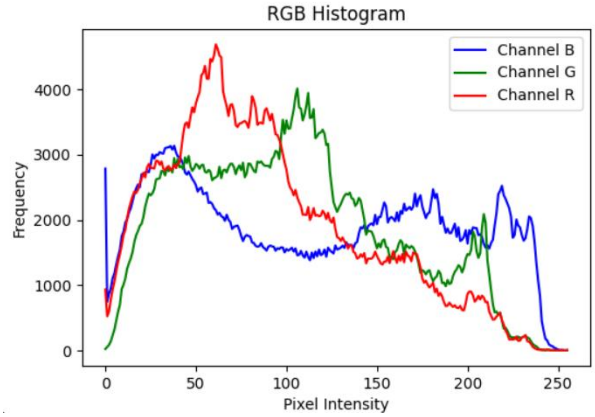

Figure 7. Image histogram results

Image convolution using Sobel and Prewitt filters provides results that are consistent with the theory and literature of digital image processing. The Sobel filter is used to detect edges by combining the first derivatives in the horizontal (x) and vertical (y) directions. The results of the application of the Sobel filter show a clear boundary of the object in the image, especially in areas with significant changes in intensity. Meanwhile, the Prewitt filter is also used for a similar purpose, but has a slightly different approach to gradient calculation than Sobel. Although the results were similar, Prewitt was simpler in its kernel structure and more computationally faster, but not as accurate as Sobel in handling noise. Both filters are effective for initial segmentation and extraction of object edge features in images.

Figure 8. Image Convolution Results using Sobel filters

Figure 9. Image Convolution Results using Prewitt filters

The morphological operations implemented in this study focus on the erosion process, including the use of rectangular structuring elements. The test results show that the erosion process successfully shrinks the area of the object in the binary image by scraping the edges of the object. Erosion is very effective for removing small noises and separating adjacent objects. By using rectangular structural elements, erosion results tend to follow the geometric shape of the element, which influences the orientation and final shape of the processed object. This technique is particularly useful in the image preprocessing process, especially to simplify the structure of objects before further analysis such as detection or classification.

http://doi.org/10.65126/jocosir.v2i2.60
Journal of Computer Science Research (JoCoSiR) with CC BY NC SA license.

Figure 10. Results of Erosion Morphology Operations

Figure 11. Results of Erosion Morphology Operations

5. CONCLUSION

This research successfully implements basic and advanced image processing methods using Python and OpenCV, including grayscale, binary, arithmetic, logic, convolution (Sobel and Prewitt), and morphology (erosion with rectangular structural elements). The results show that the methods used are able to provide image transformations in accordance with the theory, such as edge detection and simplification of object shapes. These findings support the research goal of building an interactive image processing system that can be used for further learning and application development. The main contribution of this research is the practical application of image processing methods integrated in a simple user interface. However, this study still has limitations, such as the lack of implementation of advanced segmentation and real-time image processing. Therefore, further research is recommended to develop these systems with more complex and adaptive methods, including the integration of machine learning techniques.

6. REFERENCES

[1] X. Chen *et al.*, "Recent advances and clinical applications of deep learning in medical image analysis," *Med. Image Anal.*, vol. 79, p. 102444, 2022.
[2] K. Khairunnisa *et al.*, *Image Processing*. PT. Green Pustaka Indonesia, 2025.
[3] R. Dijaya and H. Setiawan, "Buku Ajar Pengolahan Citra Digital," *Umsida Press*, pp. 1–85, 2023.
[4] W. Burger and M. J. Burge, *Digital image processing: An algorithmic introduction*. Springer Nature, 2022.
[5] E. A. Jalil, P. Wahyuningsih, N. Umar, M. Risal, and A. E. F. Anatasya, *Buku Ajar Pengolahan Citra Berbasis Open Source*. PT. Sonpedia Publishing Indonesia, 2024.
[6] S. Suganyadevi, V. Seethalakshmi, and K. Balasamy, "A review on deep learning in medical image analysis," *Int. J. Multimed. Inf. Retr.*, vol. 11, no. 1, pp. 19–38, 2022.
[7] C.-H. Choi, J. Kim, J. Hyun, Y. Kim, and B. Moon, "Face detection using haar cascade classifiers based on vertical component calibration," *Human-centric Comput. Inf. Sci.*, vol. 12, no. 11, pp. 1–17, 2022.
[8] M. D. A. HASAN, T. Bhargav, V. SANDEEP, V. S. A. I. REDDY, and R. AJAY, "Image classification using convolutional neural networks," *Int. J. Mech. Eng. Res. Technol.*, vol. 16, no. 2, pp. 173–181, 2024.
[9] Y. Liu, H. Pu, and D.-W. Sun, "Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices," *Trends Food Sci. Technol.*, vol. 113, pp. 193–204, 2021.
[10] S. Cong and Y. Zhou, "A review of convolutional neural network architectures and their optimizations," *Artif. Intell. Rev.*, vol. 56, no. 3, pp. 1905–1969, 2023.
[11] A. E. Ilesanmi and T. O. Ilesanmi, "Methods for image denoising using convolutional neural network: a review," *Complex Intell. Syst.*, vol. 7, no. 5, pp. 2179–2198, 2021.
[12] R. A. Saputra, R. Reskal, and F. M. Wahyuni, "Segmentasi pada plat kendaraan dinas dengan metode deteksi tepi canny, prewitt, sobel, & roberts," *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 6, no. 1, pp. 328–339, 2022.
[13] H. Fitriyah and R. C. Wihandika, *Dasar-Dasar Pengolahan Citra Digital*. Universitas Brawijaya Press, 2021.
[14] W. Andriyani *et al.*, *PENGANTAR TEKNOLOGI KOMPUTER*. Penerbit Widina, 2025.

http://doi.org/10.65126/jocosir.v2i2.60
Journal of Computer Science Research (JoCoSiR) with CC BY NC SA license.

ISSN 2502-3357 (Online)

[15] M. Malau, S. Hutahaean, and G. Siboro, "Metode Steganografi EOF untuk Penyisipan Pesan Teks Tersembunyi," *J. QUANCOM QUANTUM Comput. J.*, vol. 3, no. 1, pp. 18–24, 2025.
[16] N. Nurhadi *et al.*, *BUKU AJAR LOGIKA & ALGORITMA*. PT. Sonpedia Publishing Indonesia, 2023.
[17] M. R. Pratama, S. Z. Hidayat, A. R. Nuruddin, H. W. Niamaputri, and F. T. Anggraeny, "Optimasi Peningkatan Kontras Gambar Menggunakan Interval-Valued Intuitionistic Fuzzy Sets dan Contrast Limited Adaptive Histogram Equalization (CLAHE)," in *Prosiding Seminar Implementasi Teknologi Informasi dan Komunikasi*, 2025, pp. 307–317.
[18] A. M. Ahmad and C. A. Sari, "Perbandingan Kinerja Hough Transform, Watershed dengan Gabor Wavelet Filter dalam Pengenalan Iris Mata," *J. Apl. Teknol. dan Komputasi*, vol. 1, no. 1, pp. 33–40, 2025.
[19] S. Bhutada, N. Yashwanth, P. Dheeraj, and K. Shekar, "Opening and closing in morphological image processing," *World J. Adv. Res. Rev.*, vol. 14, no. 3, pp. 687–695, 2022.
[20] Y. Apridiansyah, R. Toyib, and A. Wijaya, "Metode Otsu dan Mathematical Morphology Dalam Segmentasi Region Karakter Plat Nomor Kendaraan," *J. Appl. Comput. Sci. Technol.*, vol. 3, no. 1, pp. 134–143, 2022.

http://doi.org/10.65126/jocosir.v2i2.60
Journal of Computer Science Research (JoCoSiR) with CC BY NC SA license.