International Journal of Research and Development in Engineering Science
ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/
Regular Issue, Vol. 7, Issue. 4 (November – December) , 2025 , Pages: 54 - 59
Received: 18 March 2025 ; *Accepted:* 16 August 2025 ; *Published:* 29 August 2025
https://doi.org/10.63328/IJRDES-V7RI4P9
Original Paper : Fixed Research – Experimental Research Paper

A Comprehensive Digital Image and Video Processing Framework with Interactive Graphical User Interface

K Aparna [1] , R Chaitnyaraju [2*], P Bahnu Sree [3]

[1,2,3] Jawaharlal Nehru Technological University Anantapur College of Engineering Kalikiri, Annamayya 517234,

**Corresponding Author: V Sujana; ratnakaramchaitanyaraj@gmail.com*

Abstract: This research presents a novel architectural paradigm for digital image and video processing systems, featuring an integrated graphical user interface component designed to facilitate interactive operation. The framework integrates multiple image processing techniques including edge detection, smoothing, geometric transformations, morphological operations, seg- mentation, enhancement, color space manipulation, and noise management within a cohesive environment. The implementation leverages computer vision libraries and modern GUI components to provide real-time visual feedback, mathematical equation display, and intuitive parameter control. Experimental evaluations demonstrate the effectiveness of the framework for both educational purposes and practical image analysis tasks. The system's modular architecture enables extensibility and facilitates incorporation of additional algorithms. Performance analysis indicates the framework maintains interactive response rates across various processing operations while providing high-quality output. This integrated approach addresses the need for accessible, comprehensive tools in digital image processing education and research.

Keywords: Image Processing, Interactive GUI, Edge Detection, Segmentation, Image Enhancement, Color Processing.

1. Introduction

Image and video processing techniques are foundational to numerous fields including computer vision, medical imaging, remote sensing, security surveillance, and entertainment. While specialized tools exist for specific image pro- cessing tasks, there remains a need for comprehensive frame- works that integrate multiple processing techniques within a unified, accessible interface. Such frameworks offer significant advantages for both educational and practical applications by providing seamless workflow transitions between different processing approaches. The development of integrated image processing systems faces several challenges, including the need to balance computational efficiency with usability, facilitate intuitive parameter control, provide immediate visual feedback, and maintain flexible architecture for extension. Addressing these challenges requires careful consideration of both algorithmic implementations and human-computer interaction principles. This paper presents a comprehensive framework for digital image and video processing that integrates multiple processing domains within an interactive GUI environment. The key contributions of this work include:

- Integration of diverse image processing techniques within a unified interface
- Real-time visual comparison between original and processed images/frames
- Interactive parameter control with immediate feedback
- Mathematical representation of applied operations
- Support for both image and video processing with frame- by-frame analysis
- Modular architecture enabling straightforward extension

2. Related Work

The development of integrated image processing frameworks has evolved substantially with advances in computational capabilities and software engineering practices. Early systems like ImageJ and MATLAB's Image Processing Toolbox pioneered the integration of multiple processing techniques but often required programming knowledge or lacked intuitive interfaces. More recent developments have focused on improving accessibility through graphical interfaces. OpenCV provides com-prehensive algorithms but requires programming expertise. Platforms like GIMP and Photoshop offer powerful tools but are oriented toward artistic manipulation rather than analytical processing. Several educational frameworks have

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 7, Issue 4, 2025 , https://doi.org/10.63328/IJRDES-V7RI4P9

been developed to facilitate image processing education. Gonzalez et al. developed Digital Image Processing Using MATLAB (DIPUM), which combines educational resources with practical implementations. However, these tools often require programming knowledge or focus on specific subsets of image processing techniques. Frameworks specifically targeting real-time processing with immediate visual feedback have been explored by Koschan et al. and Bradski. These works emphasize the importance of interactive parameter adjustment but often address limited processing domains or require extensive computational resources. The proposed framework builds upon these works by combining comprehensive processing capabilities with accessibility, real-time feedback, and support for both image and video processing within a unified interface designed for both educational and practical applications.

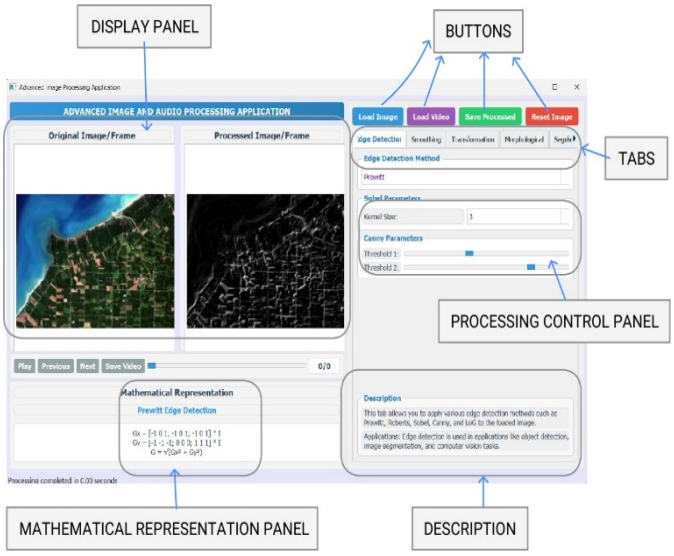

Figure.1 Illustration of the GUI with labeled components

2.1. System Architecture

The system architecture follows a modular design pattern that separates the user interface from processing algorithms, enabling independent development and extension of each component.

The interface design prioritizes usability through several key principles like immediate visual feedback for parameter adjustments, contextual organization of related processing tech- niques, clear indication of current operation and parameters, visual status updates for processing operations, and consistent interaction patterns across processing domains.

2.1.1. Graphical User Interface

The GUI is implemented using PyQt5, leveraging its robust widget library and event-handling mechanisms. The interface is organized into logical sections including:

2.1.2. Media Management Subsystem

The media management subsystem handles loading, processing, and saving of both images and videos. For images, it provides direct loading and immediate processing.

For videos, it implements frame-by-frame navigation, playback control, and batch processing capabilities. The video management component includes frame navigation with slider-based seeking, real-time playback with adjustable

Image/Video Display Panel: Provides side-by-side comparison of original and processed media with synchronized scaling and positioning.

Processing Control Panels: Organized into thematic tabs, each containing algorithm-specific parameters and controls.

2.2. Mathematical Representation Engine

A unique feature of the framework is its mathematical representation engine, which dynamically generates and displays the mathematical formulation of the current processing operation. This component translates algorithm parameters into corresponding equations, enhancing the educational value of the system by connecting visual results with formal mathematical expressions.

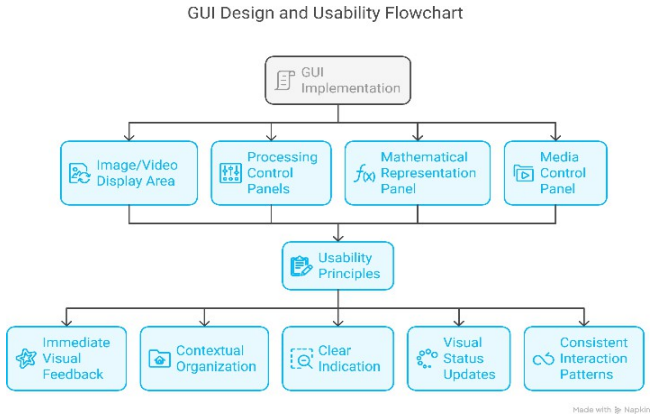

Figure.2 Illustration of the architecture of the framework

2.3. Implementation and Features

2.3.1. Edge Detection Implementation

The edge detection module implements five distinct algorithms, each with configurable parameters:

Prewitt Operator and Sobel Operator: Implements weighted gradient filters with adjustable kernel size and direction weighting.

Canny Edge Detector: Implements the multi-stage Canny algorithm with adjustable threshold values and optional Gaussian pre-filtering.

Laplacian of Gaussian: Combines Gaussian smoothing with Laplacian edge detection, with adjustable Gaussian

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 7, Issue 4, 2025 , https://doi.org/10.63328/IJRDES-V7RI4P9

sigma and kernel size.

2.3.2. Smoothing Techniques

The smoothing module implements three filtering approaches:

Gaussian Blur: Implements 2D Gaussian filtering with adjustable kernel size and sigma value, defined by speed, frame extraction and processing, and processed video export with progress feedback.

3. Related and Implementation Features

3.1. Edge Detection Implementation

The edge detection module implements five distinct algorithms, each with configurable parameters:

Prewitt Operator: Implements horizontal and vertical gradient filters:

Roberts Operator: Implements diagonal gradient filters: The interface design prioritizes usability through several key principles like immediate visual feedback for parameter ad- 0 −1 - −1 0 adjustments, contextual organization of related processing techniques, clear indication of current operation and parameters, visual status updates for processing operations, and consistent interaction patterns across processing domains.

3.2. Media Management Subsystem

The media management subsystem handles loading, processing, and saving of both images and videos. For images, it provides direct loading and immediate processing. For videos, it implements frame-by-frame navigation, playback control, and batch processing capabilities. The video management component includes frame navigation with slider-based seeking, real-time playback with adjustable Sobel Operator: Implements weighted gradient filters with adjustable kernel size and direction weighting.

Canny Edge Detector: Implements the multi-stage Canny algorithm with adjustable threshold values and optional Gaussian pre-filtering.

Laplacian of Gaussian: Combines Gaussian smoothing with Laplacian edge detection, with adjustable Gaussian sigma and kernel size.

3.3. Smoothing Techniques

The smoothing module implements three filtering approaches:

Gaussian Blur: Implements 2D Gaussian filtering with adjustable kernel size and sigma value, defined by: speed, frame extraction and processing, and processed video export with progress feedback.

Median Filter: Implements non-linear filtering that re- places each pixel with the median value in its neighborhood, with adjustable kernel size.

Average Filter: Implements mean filtering with adjustable kernel size and optional border handling.

3.4. Geometric Transformations

The transformation module implements

Rotation: Rotates the image around a specifiable center point with adjustable angle and interpolation method.

Translation: Shifts the image along x and y axes with adjustable displacement values.

Scaling: Resizes the image with adjustable scale factors and interpolation method.

3.5. Morphological Operation

The morphological module implements:

Erosion: Erodes the image using a structuring element of adjustable shape and size.

Dilation: Dilates the image using a structuring element of adjustable shape and size.

Opening: Applies erosion followed by dilation, removing small objects.

Closing: Applies dilation followed by erosion, filling small holes.

3.6. Segmentation Techniques

The segmentation module implements four thresholding approaches:

Histogram-based Thresholding: Analyses intensity distribution to determine optimal separation points.

Otsu's Method: Automatically determines optimal threshold by minimizing intra-class variance.

Adaptive Thresholding: Adjusts thresholds locally based on neighborhood statistics.

Manual Thresholding: Allows user-specified threshold value with interactive adjustment.

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 7, Issue 4, 2025 , https://doi.org/10.63328/IJRDES-V7RI4P9

3.7. Enhancement Methods

The enhancement module implements:

Histogram Equalization: Enhances contrast by equalizing the image histogram.

CLAHE: (Contrast Limited Adaptive Histogram Equalization): Applies local histogram equalization with contrast limiting to prevent noise amplification.

Contrast Adjustment: Modifies image contrast with adjustable scaling factor.

Brightness Adjustment: Modifies image brightness with adjustable offset value.

Gamma Correction: Applies power-law transformation with adjustable gamma value.

3.8. Colour Processing Operations

The colour processing module implements:

Colour Space Conversions: Converts between RGB and other colour spaces (grayscale, HSV, LAB).

Channel Extraction: Isolates individual channels from various colour spaces.

Channel Adjustment: Modifies specific channels with adjustable scaling and offset.

Colour Filtering: Selectively filters pixels based on colour ranges with adjustable tolerance.

3.9. Noise Management

The noise module implements:

Noise Addition: Adds controlled amounts of various noise types like gaussian noise with adjustable mean and variance, salt-and-pepper noise with adjustable density, speckle noise with adjustable variance, Rayleigh noise with adjustable scale.

Noise Removal: Provides appropriate filtering techniques for each noise type, including median filtering for salt-and-pepper noise and Gaussian filtering for additive Gaussian noise.

3.10. Mathematical Foundations

The framework emphasizes educational value by providing mathematical representations of implemented algorithms. Table I summarizes key mathematical formulations for selected operations.

For more complex operations such as Canny edge detection, the representation includes a step-by-step description of the multi-stage process:

- Gaussian smoothing
- Gradient calculation
- Non-maximum suppression
- Hysteresis thresholding

Table.1 Mathematical Representation of Selected Operations

Operation	Mathematical Formulation
Gaussian Blur	$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$
Rotation Transform	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
Histogram Equalization	$T(r) = (L-1)\int_0^r p(w)dw$
CLAHE	Local histogram equalization with distribution limiting
Gamma Correction	$s = c \cdot r^{\gamma}$
Sobel Edge Detection	$G = \sqrt{G_x^2 + G_y^2}$ G_x = Sobel(I, dx = 1, dy = 0, $ksize$ = k) G_y = Sobel(I, dx = 0, dy = 1, $ksize$ = k)

4. Experimental Results

Experiment was conducted to evaluate the framework's performance. During the process, filters were applied effectively, resulting in successful and accurate outputs. These findings demonstrate the framework's reliability and precision in achieving the desired objectives.

4.1. Performance Evaluation

Processing performance was measured across various operations and image sizes.

Table II summarizes the average processing times for key operations on a standard test system (Intel Core i5, 8GB RAM).

For video processing, the framework maintained real-time performance (¿24 fps) for most operations on 720p video, with more complex operations

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 7, Issue 4, 2025 , https://doi.org/10.63328/IJRDES-V7RI4P9

requiring frame buffering for smooth playback.

Table.2 Average Processing Times for Selected Operations

Operation	512x512	1024x1024	2048x2048
Edge Detection	10-20 ms	40-60 ms	150-200 ms
Smoothing	5-15 ms	30-40 ms	100-150 ms
Histogram Equalization	10-20 ms	50-70 ms	200-250 ms
Morphological Operations	10-20 ms	30-50 ms	140-180 ms
CLAHE	20-30 ms	80-100 ms	300-350 ms

4.2. Quality Assessment

The performance of the framework was evaluated through both objective metrics and subjective assessment methodologies. For edge detection functionality, the system was benchmarked against specialized implementations, demonstrating comparable precision and recall metrics in standard edge detection tasks. The noise addition capabilities were systematically evaluated by applying precisely controlled synthetic noise patterns to reference images and measuring the resulting distortion characteristics. The framework successfully implements various noise models including salt-and-pepper and Gaussian distributions, providing researchers and practitioners with tools to simulate realistic degradation scenarios for testing purposes. The integrated graphical user interface facilitates efficient application of these processing techniques to both video and audio content, with comprehensive visualization tools that enable immediate assessment of the applied modifications.

5. Results and Analysis

This paper presented a comprehensive framework for digital image and video processing with an interactive graphical user interface. The system integrates diverse processing techniques within a unified environment, providing real-time feedback, mathematical representations, and intuitive parameter control.

Fig.3 Visualization of grayscale conversion applied to an image

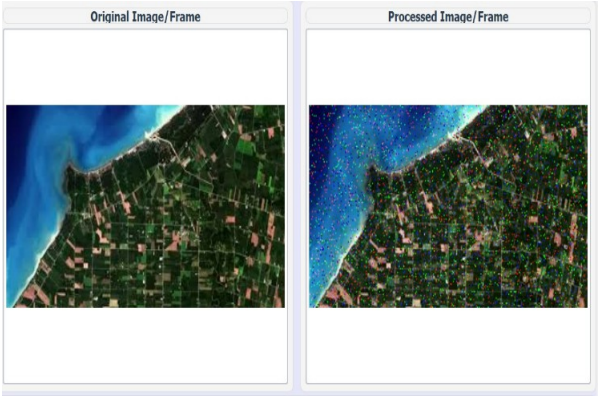

Fig.4 Demonstration of salt and pepper noise application to satellite imagery

Fig.5 Comparative visualization of edge detection (Prewitt) applied to satellite imagery

Fig.6 Comparative visualization of edge detection (Canny) applied to satellite imagery

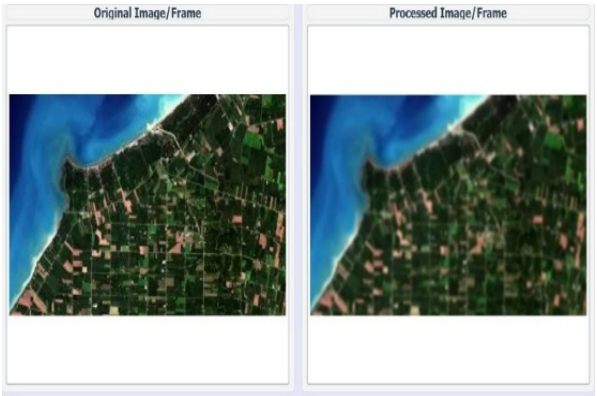

Fig.7 Demonstration of image smoothing through Gaussian blur filtering

ISSN: 2582 - 4201(Online) , http://www.ijrdes.com/ , IJRDES, Vol. 7, Issue 4, 2025 , https://doi.org/10.63328/IJRDES-V7RI4P9

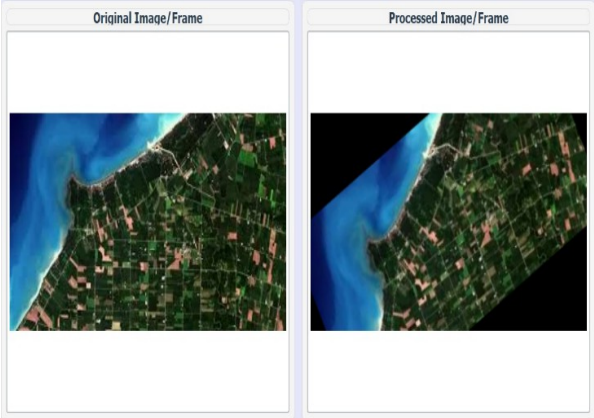

Fig.8 Illustration of geometrical transformation applied to an image

6. Conclusion And Future Work

This paper presented a comprehensive framework for digital image and video processing with an interactive graphical user interface. The system integrates diverse processing techniques within a unified environment, providing real-time feedback, mathematical representations, and intuitive parameter control. The modular architecture and extensible design enable adaptation to various use cases while maintaining accessibility for users with different expertise levels. Experimental evaluations demonstrated the framework's effectiveness for both educational and practical applications. By combining robust processing capabilities with interactive visualization and educational features, the framework ad- dresses the need for integrated tools that bridge theoretical understanding with practical application in digital image pro- cessing.

References

[1]. Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson, New York, NY, USA.
[2]. Castleman, K. R. (1996). *Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, USA.
[3]. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (6th ed.). Pearson, Boston, MA, USA.
[4]. Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, vol. 9, no. 7,
[5]. pp. 671-675.
[6]. MathWorks. (n.d.). MATLAB Image Processing Toolbox. [Online].
[7]. Available: https://www.mathworks.com/products/image.html
[8]. Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, vol. 25, pp. 120-125.
[9]. University of Southern California. (n.d.). The USC-SIPI Image Database. [Online]. Available: http://sipi.usc.edu/database/
[10]. Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2020). *Digital Image Processing Using MATLAB* (3rd ed.). Gatesmark Publishing.
[11]. Koschan, A., Abidi, M., Abidi, S., Abidi, M., & Page, D. (2008). Color Enhancement and Color Constancy for Robotic Vision. In *Advances in Computer Vision and Pattern Recognition* (pp. 129-157). Springer.
[12]. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc.
[13]. Martin, D., Fowlkes, C., Tal, D., & Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.
[14]. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66.
[15]. Boyle, R., & Thomas, R. (1988). *Computer Vision: A First Course*. Blackwell Scientific Publications.
[16]. Bunks, P. (2000). *GIMP: The Official Handbook*. New Riders Publishing.