

A Comprehensive Image Processing Pipeline for Noise Simulation, Enhancement, Segmentation, and Feature Evaluation

Sakthivel T

Department of Computer Science and Engineering
Sri Sivasubramaniya Nadar College of Engineering,
Chennai, India
sakthivel2310758@ssn.edu.in

Saravanan E

Department of Computer Science and Engineering
Sri Sivasubramaniya Nadar College of Engineering,
Chennai, India
saravanan2310681@ssn.edu.in

Abstract—This paper presents a comprehensive and systematic pipeline for digital image analysis — beginning from noise modeling, filtering, and contrast enhancement, to segmentation and feature extraction. The methodology incorporates multiple techniques including Gaussian, Speckle, and Salt-and-Pepper noise, followed by Median and Gaussian filtering, Histogram Equalization, and Contrast Stretching. The segmented images are obtained using Otsu's Thresholding and K-Means clustering, and evaluated using quantitative metrics such as PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index).

Experimental results show that the proposed approach effectively restores image details and improves segmentation accuracy. Comparative results and visual analysis demonstrate the trade-offs between global and local enhancement methods and highlight how different filters respond to distinct noise characteristics. The extracted features, including area, centroid, and color histograms, were analyzed to illustrate their relevance for object recognition and classification.

Index Terms—Image Enhancement, Noise Filtering, Histogram Equalization, Segmentation, Feature Extraction, PSNR, SSIM, K-Means, Otsu Threshold.

I. INTRODUCTION

Digital images are often degraded due to various environmental and sensor-based factors, including lighting variations, transmission noise, and quantization errors. The purpose of image processing is to enhance and extract meaningful information from such imperfect data.

This paper constructs a **multi-stage image processing pipeline** integrating key techniques from spatial filtering and frequency domain enhancement to intelligent segmentation and feature evaluation. The pipeline allows controlled experimentation to understand how each stage influences the next — ultimately improving visual perception and quantitative accuracy.

Input Image → Noise Addition → Filtering → Enhancement → Segmentation → Feature Extraction

This structured approach is suitable for domains such as medical imaging, remote sensing, robotic vision, and surveil-

lance systems, where reliable preprocessing directly determines analytical success.

II. TASK 1 — IMAGE ACQUISITION AND INITIAL ANALYSIS

A. What We Did

The input image was acquired in RGB format and converted into grayscale for computational processing. The intensity distribution was analyzed through histograms to assess brightness and contrast spread.

B. Why

Histogram inspection identifies whether the image suffers from overexposure, underexposure, or poor contrast. This step defines the parameters for the enhancement stage.

C. Implementation

Source code available at: [GitHub Repository](#)

```
img = cv2.imread("input.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.hist(gray.ravel(), 256, [0,256])
```

D. Mathematical Model

For an image $I(x, y)$ of size $M \times N$,

$$\mu = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N I(x, y), \quad \sigma = \sqrt{\frac{1}{MN} \sum (I(x, y) - \mu)^2}$$

E. Impact

Understanding the tonal distribution aids in selecting proper enhancement operations later.

III. TASK 2 — NOISE MODELING AND ADDITION

A. What We Did

Three distinct noise models were introduced:

- 1) Gaussian Noise – simulates sensor inaccuracies.
- 2) Salt-and-Pepper Noise – models impulsive data corruption.
- 3) Speckle Noise – common in radar and ultrasound imagery.

B. Why

Adding noise enables controlled evaluation of filter robustness.

C. Implementation

```
def add_gaussian_noise(img, mean=0, var=0.01):
    sigma = var ** 0.5
    noise = np.random.normal(mean, sigma, img.shape)
    noisy = np.clip(img + noise*255, 0, 255)
    return noisy.astype(np.uint8)
```

D. Mathematical Model

$$I_n(x, y) = I(x, y) + n(x, y)$$

where $n(x, y) \sim \mathcal{N}(0, \sigma^2)$ for Gaussian, and $n(x, y) = \{0, 255\}$ with probability p for Salt-and-Pepper noise.

E. Impact

The addition of controlled noise creates degraded test cases that mimic real-world imaging conditions.

IV. TASK 3 — DENOISING AND FILTERING

A. What We Did

Applied Median and Gaussian filters to noisy images to observe their smoothing and edge preservation performance.

B. Why

Filtering is essential to remove unwanted variations while maintaining structural integrity.

C. Implementation

```
median = cv2.medianBlur(noisy_img, 3)
gaussian = cv2.GaussianBlur(noisy_img, (5, 5), 1)
```

D. Mathematical Formulation

Median Filter:

$$g(x, y) = \text{median}\{f(s, t) \mid (s, t) \in S_{xy}\}$$

Gaussian Filter:

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Quantitative Metrics:

$$MSE = \frac{1}{MN} \sum (I - K)^2, \quad PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right)$$

E. Quantitative Comparison

Noise Type	Filter	PSNR (dB)	SSIM
Gaussian	Gaussian	29.1	0.92
Salt & Pepper	Median	31.4	0.95

V. TASK 4 — HISTOGRAM-BASED ENHANCEMENT

A. What We Did

Performed Histogram Equalization (HE) and Contrast Stretching (CS) to enhance global contrast.

B. Implementation

```
he = cv2.equalizeHist(gray)
cs = cv2.normalize(gray, None, 0, 255, cv2.
    ↪ NORM_MINMAX)
```

C. Formulas

Histogram Equalization:

$$s_k = (L - 1) \sum_{j=0}^k p_r(r_j)$$

Contrast Stretching:

$$I' = \frac{I - I_{min}}{I_{max} - I_{min}} \times 255$$

VI. TASK 5 — SEGMENTATION AND OBJECT ISOLATION

Used Otsu Thresholding for grayscale segmentation and K-Means Clustering for color-based region segmentation.

```
_, otsu = cv2.threshold(gray, 0, 255, cv2.
    ↪ THRESH_BINARY+cv2.THRESH_OTSU)
kmeans = cv2.kmeans(data, 3, None, criteria, 10, cv2.
    ↪ .KMEANS_RANDOM_CENTERS)
```

Otsu's Thresholding:

$$t^* = \arg \max_t \sigma_b^2(t)$$

$$\sigma_b^2(t) = q_1(t)q_2(t)[\mu_1(t) - \mu_2(t)]^2$$

K-Means:

$$J = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$$

VII. TASK 6 — FEATURE EXTRACTION

Extracted object-level features: area, centroid, bounding box, and mean color histogram.

```
M = cv2.moments(mask)
cx = int(M['m10']/M['m00'])
cy = int(M['m01']/M['m00'])
```

$$C_x = \frac{M_{10}}{M_{00}}, \quad C_y = \frac{M_{01}}{M_{00}}$$

$$\text{Area} = \sum I(x, y)$$

VIII. TASK 7 — COMPLETE PIPELINE VISUALIZATION

All stages were integrated to form a consistent and unified workflow, as illustrated in Fig. 1. The process follows the sequence: **Input** → **Noise** → **Filtering** → **Enhancement** → **Segmentation** → **Feature Extraction**.



Fig. 1. Comprehensive visualization of the image processing pipeline showing noise addition, filtering, enhancement, segmentation, and feature extraction stages.

IX. DISCUSSION AND COMPARATIVE INSIGHTS

Each stage of the pipeline was critically analyzed both qualitatively and quantitatively using PSNR and SSIM metrics. The following table summarizes the strengths, limitations, and ideal use cases for each method within the workflow.

TABLE I
COMPARATIVE ANALYSIS OF TECHNIQUES USED IN THE IMAGE PROCESSING PIPELINE

Stage	Technique	Strength	Limitation	Best Use
Noise Filtering	Median	Removes impulse noise	Slight blurring	Salt & Pepper
Enhancement	Contrast Stretch	Preserves tone	Limited for dark regions	Natural scenes
Segmentation	K-Means	Works on colors	Needs tuning	Multi-object
Feature Extraction	Centroid	Simple and effective	No texture data	Recognition

X. CONCLUSION

This study implemented a modular and data-driven image processing pipeline covering noise addition, filtering, enhancement, segmentation, and feature analysis. Each stage was critically analyzed both qualitatively and quantitatively using PSNR and SSIM metrics.

Key Results:

- Median filtering outperforms for impulse noise, while Gaussian filtering is more effective for continuous noise.

- Contrast stretching preserves color fidelity better than histogram equalization.
- K-Means clustering delivers superior segmentation in multi-region images, while Otsu's method remains optimal for high-contrast grayscale segmentation.
- Extracted features such as centroid and area proved reliable for basic classification and tracking tasks.

Professional Reflection: This project enhanced understanding of how low-level pixel operations influence high-level object detection and analysis. It also highlighted that, for small datasets, combining classical image processing techniques can yield results comparable to computationally intensive deep-learning methods.

XI. FUTURE SCOPE

- 1) Integrate CLAHE for localized enhancement.
- 2) Employ adaptive filtering based on noise estimation.
- 3) Extend to real-time video stream processing.
- 4) Combine with CNN-based segmentation (e.g., U-Net).
- 5) Implement feature selection using PCA or LDA.

REFERENCES

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Pearson, 2018.
- [2] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
- [3] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. SMC*, 1979.
- [4] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. 5th Berkeley Symposium on Math Statistics*, 1967.
- [5] Z. Wang and A. C. Bovik, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Process.*, 2004.