Edge-Preserving and Contrast-Enhancing Image Processing Tool

فرح جوده علي الشبيني 202200506
لبني محمد ناجي احمد 202203571
سلمى عصام الدين عباس 202203649
علا عبدالرازق نعيم 202207065
كيرلس محسن الفونس لويس 202203682
مازن عمرو عبدالرحمن محمد 202202644
عبدالرحمن اشرف عبدالرحمن محمد 202207123

Lab Instructor(s): TA. Ahmed Khaled
Information Systems and Computer Science Faculty
October 6 University

May 13, 2025

1. Introduction

Image quality is frequently degraded by noise during acquisition, compression, or transmission. Common types of noise include Gaussian, salt-and-pepper, and speckle, all of which can obscure important image details and affect downstream processing tasks. Traditional spatial domain filters—such as median, Gaussian, and bilateral filters—have been widely adopted for their simplicity and ability to reduce noise while preserving structural features. Additionally, metrics like Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) are essential for quantitatively evaluating denoising performance and perceptual image quality [Buades et al., 2005].

This project focuses on applying and comparing classical filtering techniques to enhance images degraded by different types of noise and exposure issues. It simulates common noise conditions, applies specific enhancement methods like CLAHE and histogram equalization, and then evaluates the results using visual comparison, PSNR, SSIM, brightness, and contrast. The goal is to demonstrate the strengths and limitations of each classical method in restoring image quality in varied scenarios.

2. Problem Definition

In digital imaging, the quality of an image can be significantly affected by a variety of degradation sources. These include sensor limitations, poor lighting conditions, compression artifacts, and noise introduced during image acquisition or transmission. Such degradations not only reduce the aesthetic quality of the image but also hinder subsequent image analysis, recognition, and interpretation tasks in fields like medical imaging, satellite image processing, and computer vision.

The most common types of image degradation include:

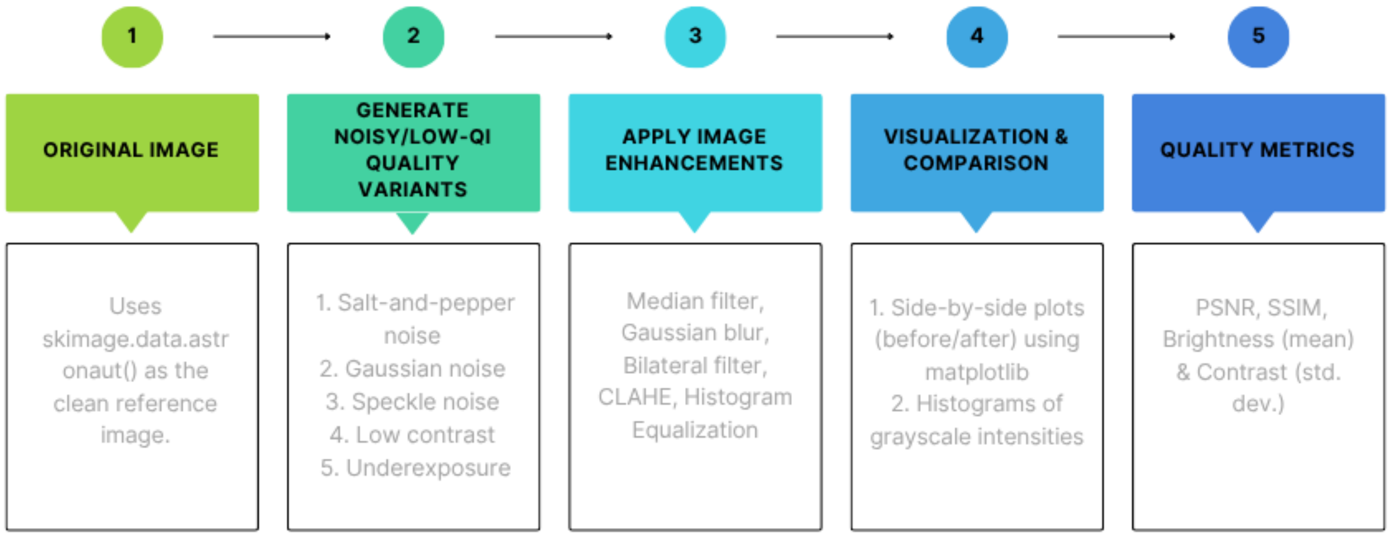
- **Salt-and-pepper noise**: Appears as random white and black pixels scattered throughout the image, often caused by faulty sensors or transmission errors.
- **Gaussian noise**: Results from electronic circuit noise or sensor imperfections, appearing as grainy distortions across the image.
- **Speckle noise**: Typically arises in radar and ultrasound imaging and manifests as granular noise patterns.
- **Low contrast**: Can result from inadequate lighting or limited dynamic range of the sensor, making it difficult to distinguish features.
- **Underexposure**: Occurs when the image sensor receives insufficient light, leading to darker images with poor detail visibility.

Addressing these issues is essential for enhancing the quality and usability of images. Poor-quality images can mislead automated systems and reduce the accuracy of visual tasks.

The objective of this project is to:

- Simulate a variety of real-world image degradation scenarios.
- Apply a set of classical enhancement and denoising techniques—including median filtering, Gaussian and bilateral filtering, CLAHE, and histogram equalization—to address these problems.
- Evaluate the performance of each method using visual comparison and quantitative metrics such as PSNR, SSIM, contrast, and brightness.
- Provide insights into the effectiveness of each technique based on the type of noise or degradation, helping inform future method selection for image restoration tasks.

3. Proposed Methodology: System Data Flow



4. Solution Definition & Core Functionality Examples

Each noise type or degradation is handled with a specific enhancement:

Degradation Type	***Enhancement Applied***	***Technique Description***
Salt & Pepper	Median Filter	Removes sparse noise by median replacement
Gaussian Noise	Gaussian Blur	Smooths image while preserving edges
Speckle Noise	Bilateral Filter	Preserves edges while reducing noise
Low Contrast	CLAHE	Adaptive histogram equalization for local contrast
Underexposure	Histogram Equalization	Redistributes intensities for better visibility

Figure1: Image Quality Evaluation using PSNR and SSIM Function

```
def quality_metrics(original, processed, title):
    gray_orig = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY)
    gray_proc = cv2.cvtColor(processed, cv2.COLOR_BGR2GRAY)
    psnr = peak_signal_noise_ratio(gray_orig, gray_proc)
    ssim = structural_similarity(gray_orig, gray_proc)
    print(f"{title} -> PSNR: {psnr:.2f}, SSIM: {ssim:.4f}")
```

Fig.1. This function computes two key metrics—**Peak Signal-to-Noise Ratio (PSNR)** and **Structural Similarity Index (SSIM)**—to assess the quality of an enhanced image compared to its original version. The images are first converted to grayscale before calculating these metrics, which provide insights into fidelity and perceptual similarity, respectively.

Example: Bilateral Filter

```
bilateral_result = apply_bilateral(img3)
results.append(("Bilateral Filter", img3, bilateral_result))
```

Results:

Figure 2&3: Bilateral Filter — Before and After / Grayscale Histogram — Before and After Bilateral Filtering

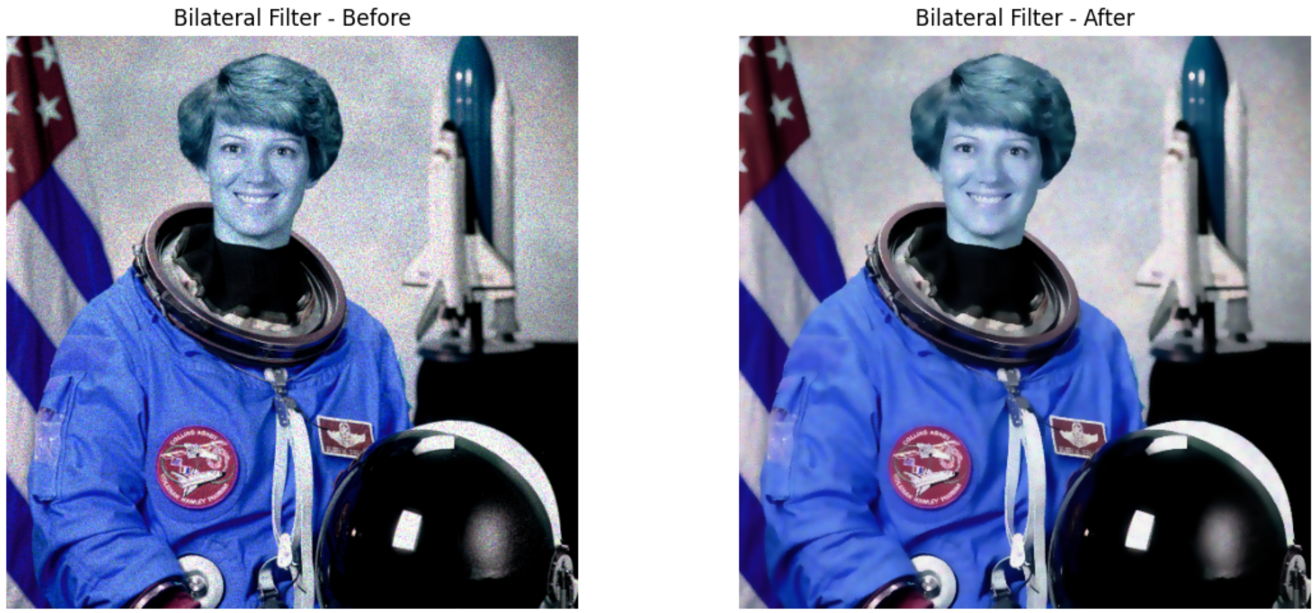


Fig.2. This side-by-side comparison illustrates the effect of applying a **bilateral filter** on an image corrupted with **speckle noise**. The filter effectively smooths the noise while preserving edges, making it ideal for scenarios where noise reduction must be achieved without blurring important image features such as facial outlines or uniform textures.

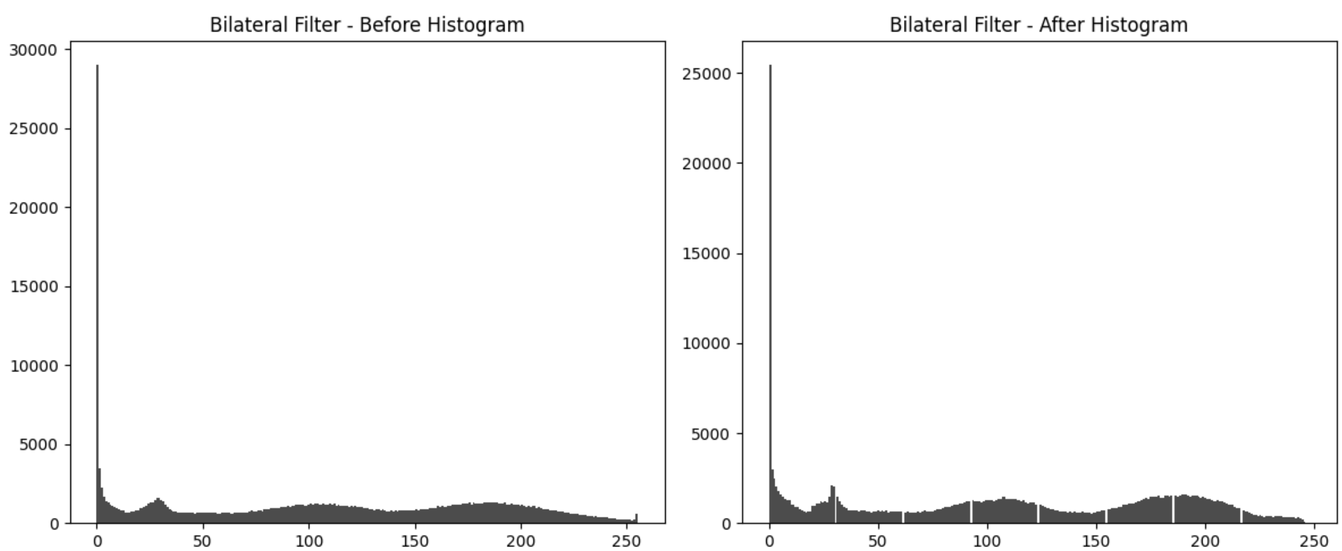


Fig.3. The histograms show reduced noise and a more uniform pixel intensity distribution after applying the bilateral filter, indicating improved image quality.

Example: CLAHE Enhancement

```
clahe_result = enhance_clahe(img4)
results.append(("CLAHE Enhancement", img4, clahe_result))
```

Results:

Figure 4&5: CLAHE Enhancement — Before and After / Grayscale Histogram — Before and After CLAHE

Fig.4. This visual comparison demonstrates the effect of applying **Contrast Limited Adaptive Histogram Equalization (CLAHE)** on a low-contrast image. CLAHE enhances local contrast and reveals details in darker regions—such as the astronaut suit and background—while preventing over-amplification of noise.

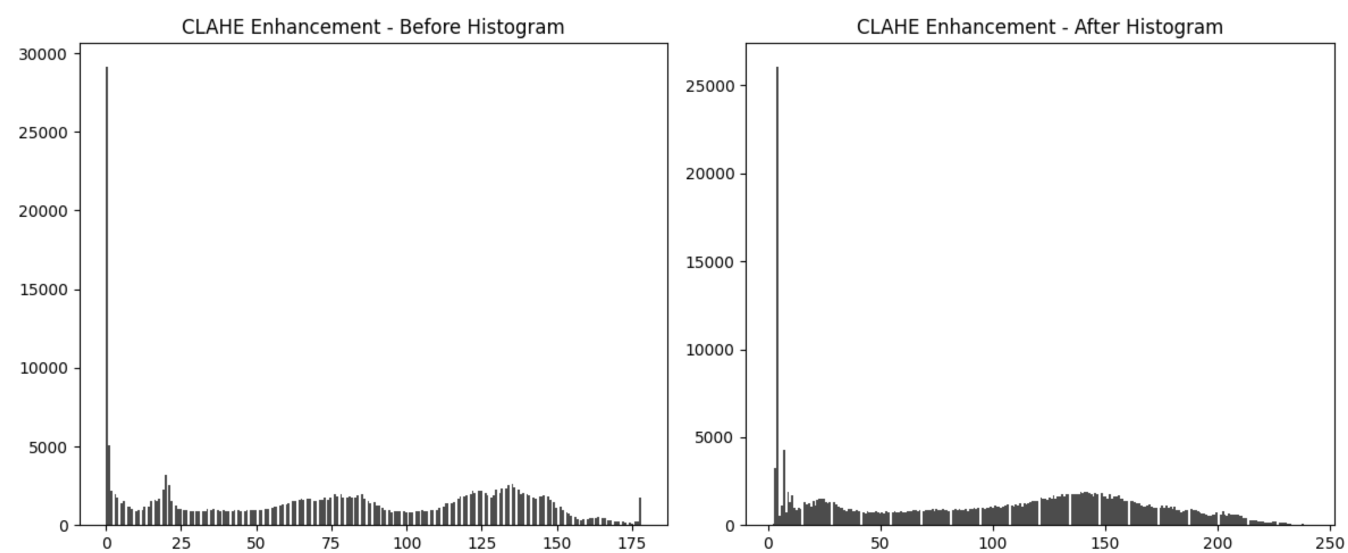


Fig.5. The histograms reflect improved contrast and intensity spread after CLAHE, with pixel values distributed more evenly across the grayscale range.

5. Code & Libraries Used

- **Language**: Python 3.x
- **Core Libraries**:
    - OpenCV (cv2) – For image filtering, CLAHE, histogram equalization, color conversions, and saving images.
    - NumPy – For handling image arrays, brightness/contrast calculations.
    - Matplotlib – For visualizing image comparisons and grayscale histograms
    - scikit-image (skimage) – Used to:
        - Load sample image (data.astronaut)
        - Add synthetic noise (util.random_noise)
        - Evaluate quality (peak_signal_noise_ratio, structural_similarity)
- **GitHub Link**: https://github.com/kirollosmohsen/IMAGE_PROCESSING_PRJ.git

6. Reference

- **Buades, A., Coll, B., & Morel, J. M. (2005).** A non-local algorithm for image denoising. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2, 60–65. https://doi.org/10.1109/CVPR.2005.38