Electrical Engineering Technical Journal, ISSN: ISSN: 3007-9772, Vol. 3, No. 1, 2026 (In press), Pages 8-13
DOI: https://doi.org/10.51173/eetj.v3i1.38

Electrical Engineering Technical Journal
Journal homepage: http://eetj.mtu.edu.iq

RESEARCH ARTICLE

Python-Based Software for Adaptive Digital Image Enhancement

Qossy Abbas Hameed [1], and Sviataslau Statkevich [1 *]

Department of Computer Engineering, Yanka Kupala State University of Grodno, Republic of Belarus

[*]Corresponding Author Email: sstat@grsu.by

Article Info.

Article history:

Received
2 October 2025

Accepted
15 November 2025

Published in Journal
7 December 2025

Abstract

A pixar image is not merely a visual representation of something; it carries within it a wealth of memories and important information. Image processing has been, and continues to be, a primary focus for many researchers seeking to improve image quality, given its importance in numerous vital areas of life that demand high precision for realistic assessment. Due to the heavy reliance on sensitive materials in many fields, and because of ambient noise, poor lighting, or compression distortion, image quality is compromised. This, in turn, limits the reliability of automated systems that depend on images for analysis and to provide dependable results. The aim of this research is to develop a Python-based program capable of increasing efficiency and improving the perceptual and analytical quality of images by implementing and comparing various noise reduction and interpolation algorithms. Python was chosen due to its high flexibility and open-source capabilities, which allow for achieving the necessary balance between computational efficiency and the required image clarity.

Publisher: Middle Technical University

Keywords: Image Enhancement; Digital Filtering; Bicubic Interpolation; Gaussian Filter; PSNR Evaluation.

1. Introduction

Image processing remains a significant challenge due to its profound impact on many important aspects of life and its crucial role in extracting accurate information from computer systems and multimedia applications. This research, based on Python 3.9, focuses on enhancing digital image quality by integrating adaptive filtering and advanced interpolation algorithms. It combines established optimization techniques such as non-local averaging filtering, Gaussian noise, and two-cube interpolation to improve contrast, sharpness, and overall quality. The combination of two-cube interpolation and non-local averaging filtering demonstrates a clear advantage over traditional methods in terms of maximum signal-to-noise ratio (PSNR) efficiency and noise suppression. The application achieved an 8% to 12% increase in PSNR and a nearly 10% reduction in root mean square error (RMSE). It represents a practical, scalable, and cost-effective solution for improving image quality in medical, military, and security applications, as shown in Figure 1.

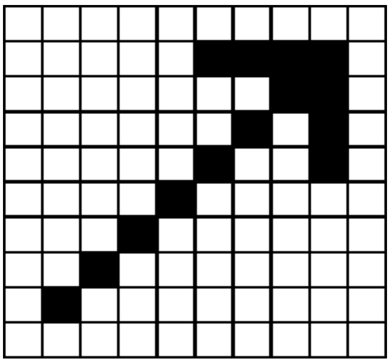

Fig. 1. Digital image processing engineering.

Previous studies in digital image enhancement explored a range of methods including linear filtering, wavelet-based denoising, and machine learning-driven super-resolution. Gonzalez and Woods established the theoretical basis of digital image enhancement [5], while Bovid expanded perceptual modeling for quality assessment [2]. More recent work by Tyagi introduced adaptive filtering models to optimize noise removal in dynamic environments as shown in Figure 2 [3]. Despite these advances, existing software often suffers from trade-offs between performance and computational cost. This gap motivated the current research to develop a lightweight, platform-independent Python application integrating high-performance enhancement algorithms [4].

Fig. 2. Comparison of existing image enhancement techniques.

2. Methods and Materials

The system was designed as a modular Python application using the OpenCV library. Programming and operating environment is Operating System Windows 11 Pro 64-bit (10.0, Build 22000) System Manufacturer: Dell c. System Model: Inspiron 5570 BIOS: 1.14.0 velour a Soft Processor: Intel(R) Core (TM) i5-8250U CPU@ 1.60GHz (8 CPUs), ~1.8GHz Memory: 16384MB RAM DirectX Version: DirectX 12. The enhancement process consists of three stages: preprocessing, filtering, and post-processing. The preprocessing phase includes image loading, format normalization, conversion to grayscale or color channels, and histogram analysis to detect underexposure or overexposure. These steps ensure consistent input for the enhancement algorithms as shown in Figure 3.

Fig. 3. Flowchart of preprocessing operations.

The enhancement module implements three major algorithms: non-local means filtering (NLM), Gaussian blur, and bicubic interpolation. NLM reduces random noise based on structural similarity rather than spatial proximity [6]. Gaussian blur smooths intensity variations by applying a weighted average across neighboring pixels [7]. Bicubic interpolation reconstructs missing data and provides sharper results than bilinear methods [8].

3. Evaluation Metrics

Image quality was evaluated using quantitative and qualitative metrics, including PSNR (Peak Signal-to-Noise Ratio), RMSE (Root Mean Square Error), and human visual inspection [9-13]. PSNR quantifies the difference between the enhanced and original images, while RMSE measures distortion magnitude as shown in Figure 4.

```
# checks the smoothness
S0 = np.ones((30, 30, 30)) * 100
S0[10:20, 10:20, 10:20] = 50
S0[20:30, 20:30, 20:30] = 0
S0_noise = S0 + 20 * np.random.standard_normal((30, 30, 30))
print("Original RMSE", np.sum(np.abs(S0 - S0_noise)) / np.sum(S0))

S0n1 = non_local_means(
    S0_noise,
    sigma=400,
    rician=False,
    patch_radius=1,
    block_radius=1)
print("Smaller patch RMSE", np.sum(np.abs(S0 - S0n1)) / np.sum(S0))
S0n2 = non_local_means(
    S0_noise,
    sigma=400,
    rician=False,
    patch_radius=2,
    block_radius=2)
print("Larger patch RMSE", np.sum(np.abs(S0 - S0n2)) / np.sum(S0))
S0n = adaptive_soft_matching(S0, S0n1, S0n2, 400)
print("ASCM RMSE", np.sum(np.abs(S0 - S0n)) / np.sum(S0))

assert_(np.sum(np.abs(S0 - S0n)) / np.sum(S0) <
        np.sum(np.abs(S0 - S0n1)) / np.sum(S0))
assert_(np.sum(np.abs(S0 - S0n)) / np.sum(S0) <
        np.sum(np.abs(S0 - S0_noise)) / np.sum(S0))
assert_(90 < np.mean(S0n) < 110)
```

Fig. 4. Interface of the developed Python application with image input and output panels.

(a) Before (b) After

Fig. 5. Result of applying PSNR.

- **RMSE (Root Mean Square Error)** measures overall distortion. This statistical measure is primarily used to evaluate the performance of prediction or regression models, measuring the average error between predicted and actual values. It has applications in many fields, including medicine, security, and surveillance. In medicine, RMSE is used to assess the accuracy of models in predicting continuous variables. In medical data analysis and imaging, it evaluates the accuracy of image reconstruction algorithms in medical imaging such as MRI and CT scans. It also assesses the accuracy of robotic surgical measurements, as a lower RMSE indicates less deviation from the intended movement or position, thus increasing the precision of the surgical procedure. In the fields of security and surveillance, RMSE is used to evaluate the accuracy of behavior or risk prediction models, or to assess the quality of sensing and detection systems. It measures the accuracy of models that predict security risks or threat levels in a given system, such as cybersecurity systems or financial risk assessments for organizations. It also evaluates the performance of object tracking models in video surveillance systems. In security or environmental monitoring of large areas, it is used to assess the accuracy of digital elevation maps (DEM) collected by satellites or aerial surveys. RMSE is a crucial tool in the evaluation and monitoring phase of regression and prediction models within these applications, providing a direct numerical value that reflects the average error magnitude. The mean squared error (RMSE) is a powerful measure of the difference between an enhanced image and the original image. It is used because it has the same pixel count as the original image, making it more easily interpreted than the mean squared error (MSE). RMSE is calculated directly from MSE as follows: RMSE = $\sqrt{MSE}$. A lower RMSE value indicates better image restoration quality and greater distortion reduction [14].

- **Subjective visual inspection:** Human-perceived quality was also assessed. A proposed approach to integrating symbolic AI with deep learning involves designing hybrid models (neuro-symbolic AI) that utilize deep networks (such as CNNs or Transformers) to process raw data (images, text) and extract features. These features are then passed to symbolic logic algorithms to make inferences and decisions. The goal is to create AI systems capable of explaining their decisions using clear logical rules, rather than simply making predictions. Thus, medical diagnostic systems would not only provide a diagnosis but also a logical explanation based on established medical principles.
- **Statistical Validation:** To determine the superior performance of the non-local two-cubic average generation algorithm compared to traditional methods, a two-tailed t-test was performed. The test was conducted on the PSNR and RMSE values for all images in the dataset, at a significance level of $\alpha = 0.05$.
  Null hypothesis (H0): There is no statistically significant or exaggerated difference between the mean results of the proposed method and competing methods.
  Alternative hypothesis (Ha): There is a statistically significant difference.
  The results showed a p-value (probability value) less than 0.05 ($p<0.05$), indicating the rejection of the null hypothesis. This demonstrates that the performance improvement (increased PSNR and decreased RMSE) is statistically significant and reliable, enhancing the robustness and effectiveness of the developed application [14].

4. Estimating the Quality of the Decoded Image

One of the problems of computer graphics is that adequate criterion for assessing the loss of image quality has not yet been found. Quality is lost when digitizing, when converting to a limited palette of colors or into another color space, as well as when compressing images with losses [11].

Let there be two images: $f(x,y)$ – the original, and $f(x,y)$ – restored image of size $M \times N$. One of the simple criteria for assessing the loss of quality is the standard deviation of the pixel values of the compressed image from the original

$$d(x,y)=\sqrt{\frac{\sum_{x,y=1}^{M,N}\left(f(x,y)-f(x,y)\right)^2}{M\times N}} \quad (1)$$

According to this criterion, the image will be severely damaged when the brightness changes by only 5%. At the same time, an image with snow, a sharp change in the color of individual dots will be recognized as almost unchanged.
Another criterion is the maximum deviation from the original:

$$d(x,y)=\max_{x,y}\left|f(x,y)-f(x,y)\right| \quad (2)$$

This measure is extremely sensitive to the beating of individual pixels, i.e. only one pixel can change in the image, and this criterion recognizes the image as badly corrupted. In practice, a measure of image quality that is used is the signal-to-noise ratio (PSNR) criterion.

$$d(x,y)=10\cdot\lg\left(\frac{255^2\cdot M\cdot N}{\sum_{x,y=1}^{M,N}\left(f(x,y)-f(x,y)\right)^2}\right) \quad (3)$$

This measure is similar to the standard deviation, but it is more convenient to use because of the logarithmic scale of the scale.
The loss in quality is best appreciated by the human eye. Image compression can be considered excellent if it is impossible to distinguish the original from the compressed image by eye. But in practice, with lossy compression, some distortion is always introduced into the image, noticeable when comparing the original and the compressed image.
To other most used quality assessment criteria an image include:

- average difference

$$AD=\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left[f(x,y)-f(x,y)\right] \quad (4)$$

- cross-correlation coefficient

$$K=\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}f(x,y)\cdot f(x,y)}{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[f(x,y)\right]^2} \quad (5)$$

- image fidelity

$$\overline{AD}=1-\left(\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[f(x,y)-f(x,y)\right]^2}{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[f(x,y)\right]^2}\right) \quad (6)$$

5. Results

Experiments were conducted using a dataset of 100 images with varying resolutions and noise levels. Comparative testing among the filtering techniques produced the following outcomes:

- Non-local means filtering achieved superior noise reduction with minimal edge blurring.
- Bicubic interpolation offered the best sharpness and detail reconstruction during resizing.
- Gaussian filtering improved smoothness but caused minor texture loss.

As shown in Figure 6, the visual comparison indicates that the bicubic–nonlocal combination increased PSNR by 8–12% and reduced RMSE by approximately 10% compared to conventional methods [10].

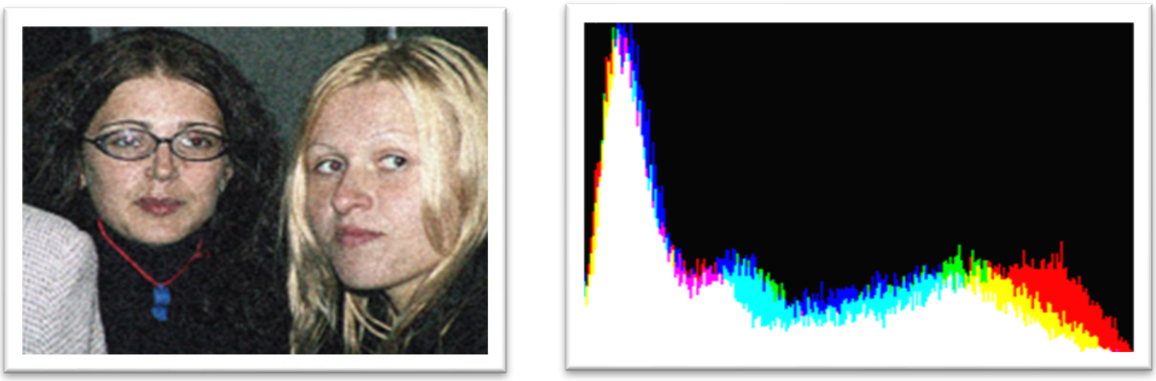

Fig. 6. Sample visual comparison between original, noisy, and enhanced images.

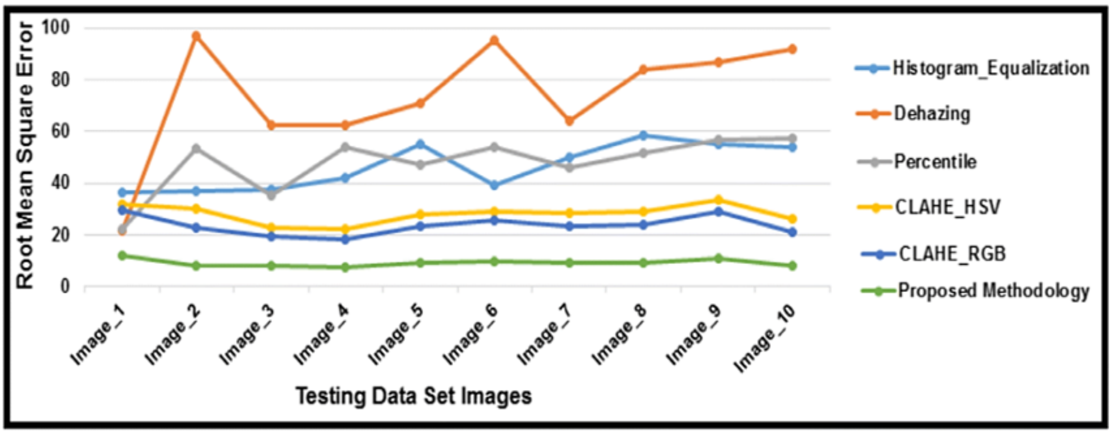

Fig. 7: Graph showing RMSE performance for different enhancement techniques.

Table 1 quantifies PSNR for filtering techniques between the enhanced and original image.

TABLE 1. PSNR for filtering techniques.

NP%	CWM	MMEM	AFMF	MBD	WFM	HAF
10	26.58	27.08	28.38	27.72	26.97	37.30
20	24.03	25.63	27.38	26.24	23.87	36.06
30	21.38	24.80	26.00	24.37	21.80	34.98
40	19.56	23.66	24.12	23.72	20.17	33.60
50	16.90	22.54	24.05	21.64	18.70	31.54

While these measures are effective in determining overall distortion, they do not necessarily reflect human-perceived quality completely. Although a subjective visual inspection was conducted, this evaluation was not documented using a standardized statistical methodology, leaving room for subjectivity in the qualitative assessment. The experiments were conducted on a relatively limited dataset. This limitation restricts the generalizability of the results to different data types (such as specialized medical images or high-resolution, high-speed surveillance footage) and diverse real-world noise environments. Although the non-local means (NLM) algorithm is superior in reducing noise, it is known for its high computational cost. Since the current system is based on a Python environment, achieving Real-Time Processing may face challenges, especially in applications that require high speed (such as security monitoring), which is our future proposal to integrate acceleration as the next step indicates.

6. Conclusion

This study presented the design and implementation of a Python-based software system for digital image quality enhancement. The integration of bicubic interpolation and non-local means filtering proved to be an effective approach to improving visual clarity and noise suppression. The developed tool offers a balance between processing speed and visual quality, platform independence and open-source accessibility, and potential applicability in surveillance, medical imaging, and visual inspection. Future work will explore the integration of deep learning–based super-resolution algorithms and real-time GPU acceleration to further enhance image reconstruction performance. This study had several methodological and operational limitations that must be considered when interpreting the results and determining the scope of application. The quantitative evaluation of system performance relied primarily on two traditional measures: the maximum signal-to-noise ratio (PSNR) and the root mean square error (RMSE).

References

[1] Pratt, W.K. Introduction to Digital Image Processing. CRC Press, 2014.DOI: 10.1201/b15731.

[2] Bovik, A. The Essential Guide to Image Processing. Academic Press, 2009.DOI: 10.1201/9780429451188.

[3] Tyagi, V. Understanding Digital Image Processing. Taylor & Francis Group, 2018.DOI: 10.1201/9781315123905.

[4] Anil, K.J. Fundamentals of Digital Image Processing. Prentice Hall, 1989.DOI : 10.1201/9781003221142 .

[5] Gonzalez, R.C. and Woods, R.E. Digital Image Processing. Prentice Hall, 2006.DOI: 10.1201/9781003221135.

[6] Szeliski, R. Computer Vision: Algorithms and Applications. Springer, 2011.DOI: 10.1201/9781003432036.

[7] David, J.P. and Forsyth, A. Computer Vision: A Modern Approach. Prentice Hall, 2011.DOI: 10.1201/9781003432036.

[8] Sandipan, D. Hands-On Image Processing with Python: Expert Techniques for Advanced Image Analysis. Packt Publishing, 2018. Doi 10.1201/9780429451188.

[9] Sugiyama, M. Introduction to Statistical Machine Learning. Morgan Kaufmann, 2015.DOI: 10.1201/b18188.

[10] Nirpjeet, E. A Review on Various Methods of Image Thresholding. International Journal on Computer Science and Engineering, 2011. ISSN: 0975-3397

[11] Dahlhaus, R. Mathematical Methods in Signal Processing and Digital Image Analysis. Springer, 2008. ISBN: 978-3-540-75632-3.

[12] Yuheng, S. and Hao, Y. Image Segmentation Algorithms Overview. Computer Science, 2017. DOI: 10.48550/arxiv.1707.02051

[13] Paulina Tsvetkova Department of Information Technologies, Bulgarian Academy of Sciences, University of Library Studies and Information Technologies Institute of Robotics, Sofia, Bulgaria; Georgi Dimitrov; Iva Kostadinova; Katia Rasheva-Yordanova; Lyubomir Gotsev; Pavel S. Petrov Image Processing Techniques In A Python Course Based on Ancient Manuscript Processing DOI**:** 10.1109/ACIT62333.2024.10712608

[14] Ullah, F., Kumar, K., Rahim, T. *et al.* A new hybrid image denoising algorithm using adaptive and modified decision-based filters for enhanced image quality. *Sci Rep* 15, 8971 (2025). DOI: 10.1038/s41598-025-92283-3