

IMPLEMENTASI TEKNIK UNSHARP MASK UNTUK PENAJAMAN CITRA DIGITAL MENGGUNAKAN OPENCV

Rivandi¹, Devaro Brahmansyah P.I²

^{1,2}Universitas Singaperbangsa Karawang; JL. HS. Ronggo Waluyo, Puseurjaya, Teluk Jambe Timur, Karawang, Jawa Barat; Telp(0267) 641177

Keywords:

Unsharp Mask, OpenCV, Penajaman Citra

Correspondent Email:

rivandi29@gmail.com

Abstrak. Penajaman citra digital merupakan salah satu proses penting dalam pengolahan citra untuk meningkatkan kejelasan visual, terutama dalam bidang fotografi, pengawasan, dan aplikasi medis. Penelitian ini mengimplementasikan teknik Unsharp Mask menggunakan pustaka OpenCV dalam bahasa pemrograman Python untuk mempertajam citra digital. Tahapan meliputi persiapan data, perancangan program, validasi, uji coba, hingga analisis hasil dengan metrik PSNR dan SSIM. Gambar uji diberikan efek Gaussian Blur sebelum diterapkan Unsharp Mask dengan nilai parameter alpha 1.5 dan blur sebesar 3. Hasil menunjukkan bahwa nilai PSNR mencapai 28.99 dB dan SSIM sebesar 0.9569, yang mengindikasikan kualitas visual citra yang dipertajam tetap tinggi dan struktur citra terjaga dengan baik. Penelitian ini memberikan bukti bahwa teknik Unsharp Mask yang diimplementasikan secara sederhana dapat meningkatkan kualitas citra secara signifikan dan layak digunakan dalam berbagai aplikasi pengolahan citra.



JITET is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Abstract. Image sharpening is a crucial process in digital image processing aimed at enhancing visual clarity, especially in photography, surveillance, and medical applications. This study implements the Unsharp Mask technique using the OpenCV library in Python to enhance digital image sharpness. The research stages include data preparation, program design, validation, testing, and result analysis using PSNR and SSIM metrics. The test image was first blurred using Gaussian Blur, then enhanced using the Unsharp Mask technique with an alpha value of 1.5 and a blur value of 3. The results show a PSNR value of 28.99 dB and an SSIM of 0.9569, indicating that the sharpened image maintains high visual quality and structural similarity to the original. This study demonstrates that the Unsharp Mask technique, implemented in a simple manner, can significantly improve image quality and is suitable for various image processing applications.

1. PENDAHULUAN

Penajaman citra digital merupakan proses penting dalam pengolahan citra untuk meningkatkan kejernihan dan menonjolkan detail visual, yang banyak digunakan dalam fotografi, pengawasan, dan pengolahan citra

medis. Salah satu teknik penajaman citra adalah *Unsharp Masking* yang bekerja dengan mengurangi versi kabur atau blur dari citra untuk memperjelas tepi. Meskipun telah banyak digunakan, implementasi praktis teknik ini menggunakan pustaka OpenCV dengan

pengaturan parameter yang optimal belum banyak dibahas secara sistematis. Oleh karena itu, penelitian ini bertujuan mengkaji efektivitas teknik *Unsharp Mask* dalam meningkatkan ketajaman gambar melalui eksperimen pada berbagai jenis citra, serta memberikan panduan implementasi sederhana dan efisien yang dapat dimanfaatkan dalam berbagai aplikasi pengolahan citra.

2. TINJAUAN PUSTAKA

2.1 *OpenCV*

OpenCV atau biasa disebut dengan Open Computer Vision merupakan pustaka perangkat lunak yang bersifat open-source. *Library* ini berguna untuk kebutuhan pengolahan citra digital seperti analisis gambar dan video. [1][2][3][4]

2.2 Citra Digital

Pengolahan citra digital adalah cabang ilmu yang mempelajari teknik-teknik untuk memproses gambar atau video secara digital menggunakan komputer. Citra yang diolah dapat berupa gambar diam maupun bergerak, dan representasi warnanya biasanya menggunakan model RGB (*Red, Green, Blue*), yang memungkinkan komputer memproses warna dalam bentuk nilai numerik secara universal.[5]

2.3 *Unsharp Mask*

Unsharp Mask adalah metode peningkatan ketajaman citra dengan menambahkan tepi dari versi kabur yang dikurangi, menghasilkan peningkatan detail visual. Meskipun teknik ini telah lama digunakan, penelitian *modern* seperti penggunaan *Unsharp Mask* untuk filter terpandu menunjukkan bahwa variasi algoritmik masih aktif dieksplorasi.[6]

2.4 *Python*

Python merupakan bahasa pemrograman tingkat tinggi yang bersifat interpretatif dan memiliki sintaks yang sederhana serta mudah dipahami. Python banyak digunakan dalam berbagai bidang seperti pengembangan web, otomatisasi, analisis data, kecerdasan buatan, hingga pembelajaran mesin. Salah satu keunggulan utama Python adalah dukungan komunitas yang luas dan ketersediaan pustaka (*library*) yang sangat beragam, sehingga

memudahkan pengembang dalam membangun berbagai jenis aplikasi secara efisien dan fleksibel. [7][8]

Dalam konteks pengolahan citra digital, Python menjadi salah satu bahasa yang sangat populer karena didukung oleh pustaka-pustaka khusus seperti OpenCV, Pillow, dan scikit-image. Pustaka-pustaka ini menyediakan fungsi-fungsi lengkap untuk manipulasi citra seperti pembacaan, konversi warna, filter spasial, segmentasi, hingga deteksi objek. Kelebihan Python dalam pengolahan citra terletak pada kemudahannya dalam mengakses fungsi kompleks hanya dengan beberapa baris kode, sehingga sangat cocok untuk keperluan penelitian maupun pengembangan prototipe dalam bidang visi komputer dan pengolahan citra. [9][10]

2.5 Gaussian Blur

Gaussian Blur adalah teknik blur yang menggunakan fungsi matematis Gaussian (distribusi normal) untuk melakukan weighted averaging pada piksel dan sekitarnya. Dalam praktiknya, sebuah kernel Gaussian (misalnya ukuran 3×3 atau 5×5) ditempatkan di tiap piksel, dan nilai piksel baru dihitung berdasarkan bobot dari piksel tengah dan tetangganya—dengan bobot berkurang seiring jarak dari pusat kernel—sehingga menghasilkan efek blur yang halus dan alami. Keunggulan utama teknik ini dibanding box blur adalah kemampuannya untuk menjaga tepi dengan lebih baik, karena bobot yang menurun secara gradual cenderung mempertahankan struktur citra asli.

Dalam ranah pengolahan citra, Gaussian Blur populer sebagai *low-pass filter*—untuk meredam *high-frequency noise* tanpa menghilangkan struktur (*struktur low-frequency*) secara drastis—sehingga sering digunakan sebelum menjalankan operasi lanjutan seperti deteksi tepi (misalnya Laplacian atau Canny). Selain itu, properti separabilitas kernel Gaussian mempercepat konvolusi, sangat berguna ketika diaplikasikan pada citra resolusi tinggi atau video. Referensi penelitian modern juga sudah membahas hubungan antara parameter blur (σ) dan respons tepi citra, serta penerapannya dalam augmentasi data untuk *deep learning*. [11][12]

2.6 PSNR (Peak Signal-to-Noise Ratio)

PSNR (Peak Signal-to-Noise Ratio) merupakan metode klasik untuk mengukur perbedaan antara gambar asli dan gambar hasil dengan menghitung rasio antara nilai puncak sinyal dan noise berdasarkan Mean Squared Error (MSE). Nilai PSNR dipresentasikan dalam satuan desibel (dB), di mana umumnya nilai di atas 30 dB menunjukkan hasil yang baik dalam konteks kompresi atau pemulihan citra. [13]

2.7 SSIM (Structural Similarity Index Measure)

SSIM mengevaluasi kualitas citra berdasarkan luminansi, kontras, dan struktur, sehingga lebih sesuai dengan persepsi manusia. Versi klasiknya masih luas digunakan, namun penelitian terbaru telah menyempurnakan parameter dan modelnya—seperti SSIM berbasis data dan versi robust terhadap noise non-Gaussian.

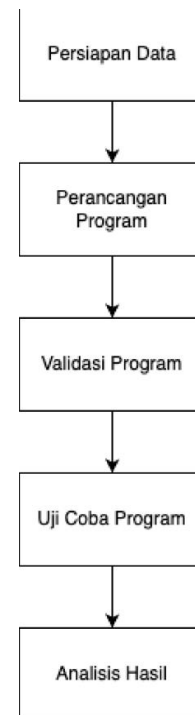
Jika $\alpha = \beta = \gamma = 1$ dan $c_3 = c_2/2$, rumus SSIM sederhana adalah:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Nilai SSIM berkisar antara -1 hingga 1, di mana 1 menunjukkan kesamaan sempurna. [14][15]

3. METODE PENELITIAN

Dalam penelitian ini, ada beberapa tahapan yang dilakukan:



Gambar 1. Tahapan Penelitian

3.1 Persiapan Data

Penelitian ini di mulai dengan mempersiapkan kebutuhan program. Hal-hal yang perlu disiapkan antara lain, menginstall *library* Python dan objek gambar.

“pip install opencv-python”

“pip install scikit-image”

Library di atas akan digunakan dalam penulisan kode program nanti. Tidak lupa juga untuk menyiapkan objek gambar untuk menjadi uji coba pemrograman.

3.2 Perancangan Program

Setelah mempersiapkan semua yang dibutuhkan, program telah siap untuk dibuat. Berikut merupakan Flowchart sistem yang akan dibuat.



Gambar 2. Flowchart Sistem

Library yang digunakan adalah OpenCV yang akan memuat berbagai properti atau command yang akan memproses citra digital.

3.3 Validasi Program

Di tahap ini, sistem akan membaca dan memvalidasi objek gambar yang akan digunakan. Tahap ini juga merupakan pengecekan *error* agar objek gambar dapat terbaca dengan jelas.

3.4 Uji Coba Program

Sistem akan dibuat agar dapat menguji objek gambar menggunakan gaussian blur dan juga teknik *unsharp mask*. Nantinya, sistem akan diatur agar gambar bisa menjadi lebih tajam dan jelas dengan berbagai bobot nilai.

Setelah mendapatkan citra yang telah diuji, citra akan disimpan dan siap untuk dianalisa dengan PSNR dan SSIM.

3.5 Analisa Hasil

Analisa hasil akan menggunakan PSNR (Peak Signal-to-Noise Ratio) dan SSIM (Structural Similarity Index Measure). Dengan ini, akan dapat diketahui hasil perbandingan antara objek gambar dan juga citra yang telah dipertajam dengan teknik *Unsharp Mask*.

4. HASIL DAN PEMBAHASAN

Dari tahapan metodologi dan Flowchart sistem yang telah dibuat, berikut merupakan hasil dan pembahasan pada penelitian ini.

```

[8]: import cv2

def apply_unsharp_mask(image_path, alpha=1.5, gaussian_blur_value=3):

    # Load image
    image = cv2.imread(image_path)

    # Validating if image is loaded or not
    if image is None:
        raise ValueError("Image not loaded properly")

    # Gaussian blur to the image
    gaussian_blur = cv2.GaussianBlur(image, (0, 0), gaussian_blur_value)

    # Add the blurred image to the original image with the given alpha weight
    unsharp_image = cv2.addWeighted(image, alpha, gaussian_blur, 1 - alpha, 0)

    # Save the result
    output_path = image_path.replace('.jpeg', '_hd.jpeg')
    cv2.imwrite(output_path, unsharp_image)

    return output_path

# Apply the function to the uploaded image
image_path = 'pencil.jpeg' # Replace with your actual image path
sharpened_image_path = apply_unsharp_mask(image_path)

print(f"The sharpened image is saved as {sharpened_image_path}")
The sharpened image is saved as pencil_hd.jpeg
  
```

Gambar 3. Kode Program Unsharp Mask

Program di atas merupakan program yang telah dibuat untuk penelitian ini. Dimulai dari impor *library* yaitu OpenCV sebagai *library* utama dari penelitian ini.

Selanjutnya akan didefinisikan sebuah fungsi yang berisi 3 parameter antara lain *image_path* untuk menentukan objek gambar, *alpha* untuk mengatur bobot ketajaman dalam penelitian ini *value* berjumlah 1.5 lalu ada parameter *gaussian blur* untuk mengontrol berapa luas efek blur yang diterapkan.

Lalu akan ada validasi program yang bertujuan agar gambar yang telah didefinisikan benar-benar dapat dibaca oleh program. Jika tidak, maka program akan dijalankan ulang.

Lalu, dibuatkan *variable* untuk membaca objek gambar dan gambar tersebut akan diterapkan *gaussian blur*.

Selanjutnya adalah tahap implementasi dari *Unsharp Mask* yaitu menggabungkan Gambar asli dan blur sesuai bobot yang telah ditentukan. Gambar akan disimpan dan siap untuk diuji.

Untuk percobaan, penulis menggunakan gambar “pencil.jpeg”



Gambar 4. Objek Gambar

Lalu berikut ini adalah citra yang telah dipertajam dengan *unsharp mask*.



Gambar 5. Citra setelah dipertajam

Dengan ini, maka sudah bisa dianalisa menggunakan PSNR dan juga SSIM. Berikut merupakan kode program yang dibutuhkan.

```
[12]: from skimage.metrics import peak_signal_noise_ratio, structural_similarity
import cv2

def evaluate_image_quality(original_path, result_path):
    # Baca gambar asli dan hasil
    original = cv2.imread(original_path)
    result = cv2.imread(result_path)

    if original is None or result is None:
        raise ValueError("Salah satu gambar tidak berhasil dimuat.")

    # Ubah ke grayscale untuk SSIM
    original_gray = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY)
    result_gray = cv2.cvtColor(result, cv2.COLOR_BGR2GRAY)

    # Hitung PSNR
    psnr_value = peak_signal_noise_ratio(original, result)

    # Hitung SSIM
    ssim_value, _ = structural_similarity(original_gray, result_gray, full=True)

    print(f"PSNR: {psnr_value:.2f} dB")
    print(f"SSIM: {ssim_value:.4f}")

    return psnr_value, ssim_value

# Contoh pemanggilan fungsi
evaluate_image_quality("pencil.jpeg", "pencil_hd.jpeg")

PSNR: 28.99 dB
SSIM: 0.9569
[12]: (28.989394695929853, 0.9568602300818316)
```

Gambar 6. Analisis Program

Dengan nilai PSNR sebesar 28.99 dB mengartikan bahwa citra yang telah dipertajam sedikit berubah dan terlihat berbeda. Namun, SSIM yang bernilai 0.9569 mengartikan bahwa gambar sangat mirip.

5. KESIMPULAN

- Teknik *Unsharp Mask* dengan parameter α 1.5 dan gaussian blur value 3 berhasil meningkatkan ketajaman visual gambar tanpa menyebabkan distorsi yang signifikan.
- Nilai Nilai PSNR sebesar 28.99 dB menunjukkan bahwa gambar hasil masih memiliki tingkat kemiripan yang cukup baik dengan gambar asli, meskipun terdapat sedikit perbedaan pada detail tertentu.
- Nilai SSIM sebesar 0.9569 menandakan bahwa struktur dan kualitas visual gambar secara keseluruhan tetap terjaga dengan sangat baik setelah diproses, sehingga metode ini efektif digunakan untuk peningkatan kualitas citra digital.

UCAPAN TERIMA KASIH

Terima kasih atas semua pihak yang telah membantu, mulai dari rekan, pembimbing, hingga para penulis lain yang tercantum pada daftar pustaka. Kami berharap agar penelitian ini dapat berguna untuk masa yang akan datang

DAFTAR PUSTAKA

- [1] Al-Abaji, M., & Al-Ameen, Z. (2024). SHARPNESS IMPROVEMENT OF MAGNETIC RESONANCE IMAGES USING A GUIDED-SUBSUMED UNSHARP MASK FILTER. *Applied Computer Science*, 20(4), 192–210.
- [2] Divanny, S. R. (2024). PENERAPAN OPENCV DAN FUZZY LOGIC CONTROLLER UNTUK LINE FOLLOWER BERBASIS KAMERA PADA SIMULASI ROBOT E-PUCK DI WEBOTS. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(3).
- [3] Gafari, A., Ramadhanu, A., Padang, Y., Raya, J., Begalung, L., Xx, N., Lubuk, K., Kota, B., & Sumbar, P. (2025). KLASIFIKASI CITRA DIGITAL BATU KERIKIL DAN BATU KALI DENGAN METODE K-MEANS DAN TEKNIK PENGOLAHAN CITRA. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 9, Issue 2).
- [4] Muthia, T. (2025). Monitoring Pertumbuhan Vertikal Tanaman Hidroponik Menggunakan Webcam dan Pengolahan Citra Digital. *Electrician : Jurnal Rekayasa Dan Teknologi Elektro*, 19, 166–171.
- [5] Mahanani, A. (2023). Penerapan Unsharp Mask Pada Sekuen Diffusion Weighted Image Kasus Stroke Terhadap SNR dan CNR. *Informasi Dan Promosi Kesehatan*, 2(2), 107–112.
- [6] Qisthiano, M. R., & Pratiwi, A. O. (2025). DETEKSI TEPI PADA CITRA OBJEK BENDA MENGGUNAKAN ALGORITMA SOBEL DAN PREWITT DENGAN PYTHON. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(2).
- [6] Putri, D.D., Nama, G.F. and Sulistiono, W.E., 2022. Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier. *Jurnal Informatika dan Teknik Elektro Terapan*, 10(1).
- [7] Phandany, J., Sambul, A., & Lumenta, A. S. M. (2022). Studi Perbandingan Algoritma Kompresi Optimal Citra Digital Menggunakan Python. *Jurnal Teknik Elektro Dan Komputer*, 11, 23.

- [8] Saragih, S. (2025). Implementasi Algoritma Haar Cascade Menggunakan Pengolahan Citra Digital untuk Absensi Deteksi Wajah dan Nama Menggunakan Python. *Jurnal Sosial Teknologi*, 5, 789–798.
- [9] Sastra, R., Musyafa, N., & Wijonarko, B. (2025). Optimalisasi Sistem Presensi Berbasis Face Recognition dengan Python dan OpenCV. *INSANtek*, 6, 36–42.
- [10] Sastra, R., Musyafa, N., & Wijonarko, B. (2025). Optimalisasi Sistem Presensi Berbasis Face Recognition dengan Python dan OpenCV. *INSANtek*, 6, 36–42.
- [11] Andono, P., & Sari, A. (2022). Remove Blur Image Using Bi-Directional Akamatsu Transform and Discrete Wavelet Transform. *Scientific Journal of Informatics*, 9, 179–188.
- [12] Yuadi, I., Yudistira, N., Habiddin, H., & Nisa, K. (2025). A Comparative Study of Image Processing Techniques for Javanese Ancient Manuscripts Enhancement. *IEEE Access*, PP, 1.
- [13] Li, Y., Zhang, Y., Jia, D., Gao, S., & Zhang, M. (2025). Noise Impact Analysis in Computer-Generated Holography Based on Dual Metrics Evaluation via Peak Signal-to-Noise Ratio and Structural Similarity Index Measure. *Applied Sciences*, 15, 6047.
- [14] Winderlich, K., Dalelane, C., & Walter, A. (2024). Classification of synoptic circulation patterns with a two-stage clustering algorithm using the modified structural similarity index metric (SSIM). *Earth System Dynamics*, 15, 607–633.
- [15] Rani, N., Ratini, N., Gunawan, A., Sutapa, G., Suyanto, H., & Kasmawan, I. G. (2025). Prediction of Radiotherapy Dose Distribution for Glioblastoma Using Convolutional Neural Network Model. *Asian Journal of Medicine and Health*, 23, 1–8.