Image Manipulation

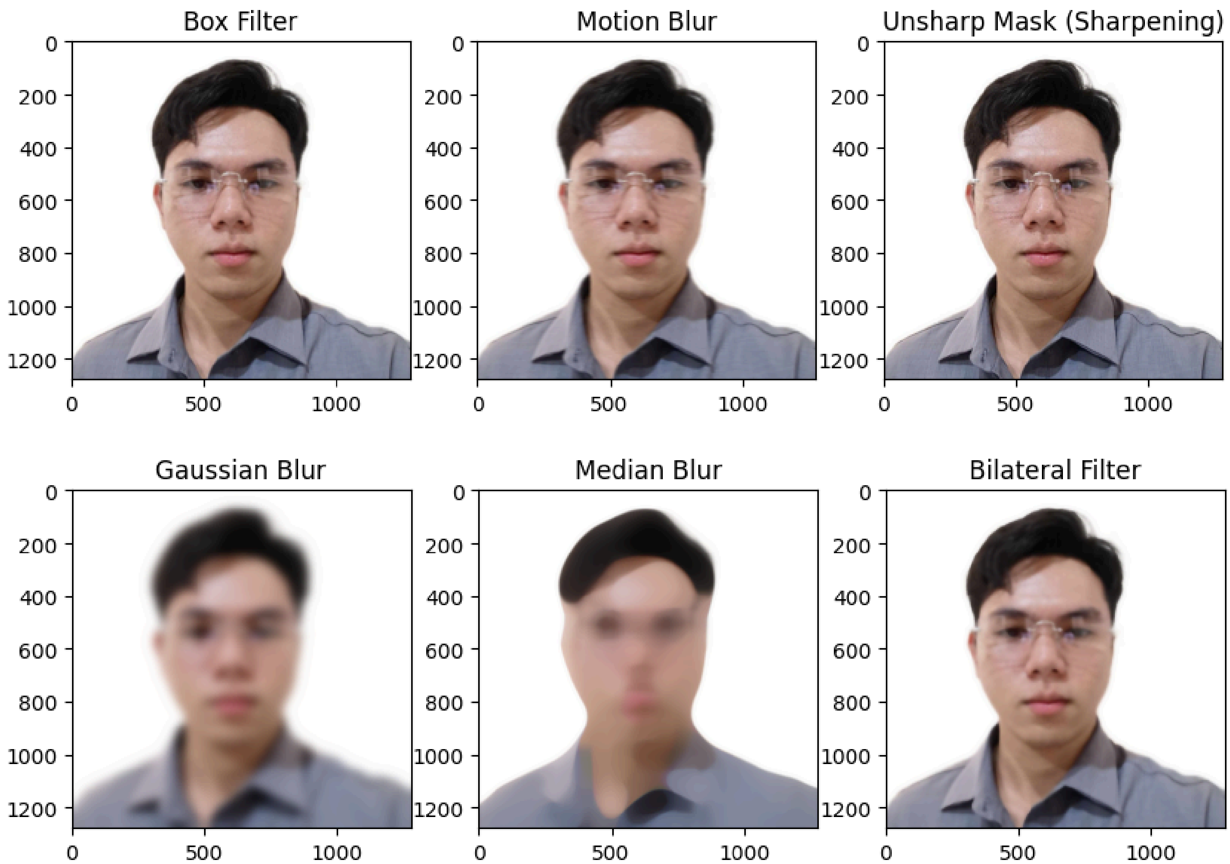
Original Image

Rotated Image 45°

50% Scaled Image

In this Exercise, I used OpenCV for manipulating the image in two ways: rotation and scaling. OpenCV is a powerful library used for image processing, computer vision, and machine learning.

Comparison of Blurring Techniques



Technique	Blurring (0-100%)	Noise Reduction (0-100%)	Edge Preservation (0-100%)	Artistic Effects (0-100%)	Sharpening (0-100%)
Box Filter	70%	50%	30%	20%	10%
Motion Blur	60%	20%	20%	70%	10%
Unsharp Mask	10%	10%	90%	10%	90%
Gaussian Blur	80%	70%	50%	60%	10%
Median Blur	85%	90%	60%	50%	20%
Bilateral Filter	30%	85%	90%	80%	20%

Blurring: Measures how much of the image is blurred (higher = more blurred). Median Blur has the strongest blurring effect at 85%, while Bilateral Filter has minimal blurring at 30%.

Noise Reduction: Measures the technique's ability to reduce image noise (higher = more noise reduction). Median Blur and Bilateral Filter excel in noise reduction with 90% and 85% respectively.

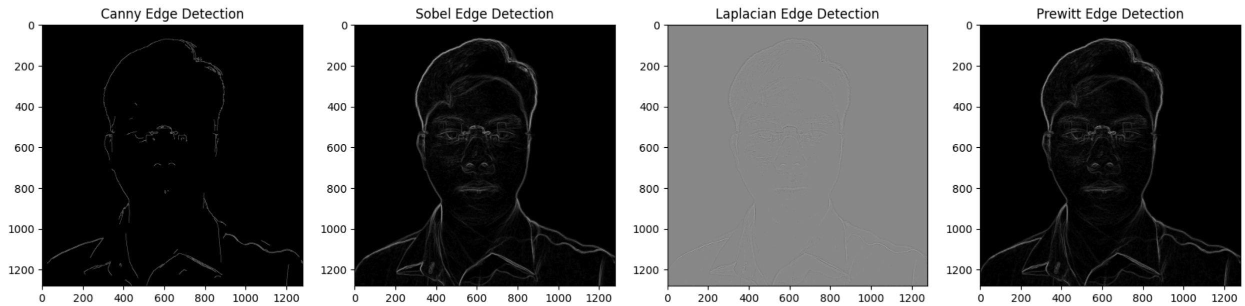
Edge Preservation: Rates how well the method preserves edges (higher = better edge retention). Bilateral Filter and Unsharp Mask are strong in this area with 90%, while Box and Motion Blur perform poorly.

Artistic Effects: Reflects the method's ability to create artistic effects like soft-focus or motion blur (higher = more artistic). Motion Blur and Bilateral Filter score higher for their stylized outputs.

Sharpening: Measures the effectiveness in enhancing image sharpness (higher = more sharpening). Unsharp Mask is the only method designed for sharpening, scoring 90%, while other techniques are primarily for smoothing or blurring.

Blurring techniques vary in terms of their impact on image quality, noise reduction, and edge preservation. The Box Filter applies a basic uniform blur by averaging pixel values within a neighborhood, creating a blocky effect that may feel unnatural. While it reduces noise to some extent, it also blurs edges and lacks artistic appeal or sharpening capabilities. Gaussian Blur offers a smoother and more natural blurring effect by averaging pixels based on a Gaussian function. It is commonly used for noise reduction but sacrifices edge sharpness, making it suitable for soft-focus effects. Motion Blur simulates the appearance of movement by creating directional blur along a certain axis. While it's useful for adding dynamic motion effects, it does not reduce noise effectively and tends to blur edges. Unsharp Mask, on the other hand, focuses on enhancing edges to sharpen images, making details crisper. However, it can amplify noise because it accentuates all details, including unwanted artifacts. Median Blur excels at reducing noise, particularly salt-and-pepper noise, by replacing each pixel with the median of surrounding values. It preserves edges better than Gaussian or Box filters, though it creates a stronger blurring effect. The Bilateral Filter is highly effective at reducing noise while preserving edge details, as it smooths areas with similar intensity without averaging across edges. This technique is ideal for scenarios where edge preservation is crucial, and it can also create stylized, artistic effects by smoothing textures while keeping edges sharp. In summary, Gaussian Blur and Box Filter are ideal for general smoothing, with Gaussian being more effective at noise reduction. Median Blur and Bilateral Filter excel in noise reduction and edge preservation, with Bilateral Filter offering the best results in maintaining edge detail. Unsharp Mask is optimal for sharpening, while Motion Blur adds an artistic directional effect but blurs edges in the process.

Comparison of Edge Detection Techniques



Technique	Sensitivity to Noise (0-100%)	Edge Thinness (0-100%)	Edge Continuity (0-100%)	Computational Efficiency (0-100%)
Canny Edge Detection	50%	90%	90%	60%
Sobel Edge Detection	80%	60%	60%	80%
Laplacian Edge Detection	90%	40%	40%	90%
Prewitt Edge Detection	80%	60%	60%	80%

Sensitivity to Noise: Measures how sensitive the technique is to noise (higher = more sensitivity). Laplacian has the highest sensitivity at 90%, meaning it amplifies noise the most, while Canny reduces noise with a Gaussian filter, resulting in lower sensitivity at 50%.

Edge Thinness: Rates how thin and sharp the detected edges are (higher = thinner, sharper edges). Canny excels in producing thin, sharp edges at 90%, while Laplacian struggles with only 40% thinness.

Edge Continuity: Measures how continuous the edges are (higher = more connected edges). Canny is again the best at maintaining edge continuity with 90%, whereas Laplacian struggles due to its noise amplification, scoring only 40%.

Computational Efficiency: Rates the speed and resource efficiency of the technique (higher = more efficient). Laplacian and Prewitt are computationally efficient at 90% and 80%, respectively, while Canny is slower at 60% due to its more complex process.

Edge detection techniques vary in noise sensitivity, edge thinness, continuity, and computational efficiency. Canny is moderately sensitive to noise due to its use of Gaussian

smoothing and produces thin, continuous edges through non-maximum suppression and edge linking. It’s slower because of its multi-step process. Sobel, highly sensitive to noise, produces thicker edges and breaks them into segments, but it’s computationally faster. Laplacian amplifies noise significantly, leading to thick, discontinuous edges, though it’s computationally efficient. Prewitt, like Sobel, is fast but sensitive to noise, generating thicker, segmented edges.
In summary, Canny excels in edge quality but is slower, while Sobel, Prewitt, and Laplacian are faster but less effective in noisy environments and edge continuity.