Exercise 1: Scaling and Rotation

Scaled Image (50%)

Rotated Image (45°)

Comparison:

Purpose: Scaling is typically used to resize images while maintaining content integrity and aspect ratio, whereas rotation is used to change the orientation of an image.

Image Integrity: Scaling preserves the entire content of the image, just at a different size. Rotation can lead to portions of the image being clipped, depending on the rotation angle and image boundaries.

Usage Context: Scaling is often used in UI/UX design, web development, and other fields where image size needs to be adjusted. Rotation is used in photo editing, graphic design, and any scenario where changing the image's orientation is necessary.

Exercise 2: Blurring Techniques

Gaussian Blur (5x5)

Median Blur (5x5)

Comparison

Purpose: Both blurring techniques are used to reduce noise, but Gaussian blur aims for a smooth overall effect, while median blur focuses on noise reduction while preserving edges.

Noise Reduction: Gaussian blur provides a more general smoothing effect, which can be useful for reducing various types of noise. Median blur is particularly effective at removing specific types of noise, such as salt-and-pepper noise.

Edge Preservation: Gaussian blur tends to smooth out edges along with noise, leading to a loss of detail. Median blur, on the other hand, preserves edges more effectively, maintaining the structure of the image.

Visual Impact: Gaussian blur creates a softer, more diffused appearance, which can make the image appear more aesthetically pleasing in some cases. Median blur retains more detail and sharpness, which can be crucial for applications requiring edge detection and preservation.

Basic Image Processor (Interactive)

Comparison:

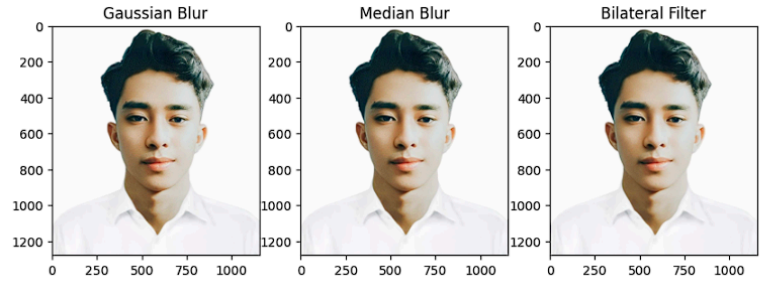
Purpose: The original image serves as the complete visual representation, whereas the Canny processed image isolates and highlights the edges, useful for further analysis.

Detail: The original image contains all the details, including textures and colors. The Canny processed image abstracts these details, showing only the structural edges.

Noise Handling: The Canny algorithm includes a noise reduction step, making it effective in identifying true edges while minimizing the impact of noise.

Applications: The original image is suitable for visual tasks where full detail is needed. The Canny edge detection is useful for applications such as object detection, image segmentation, and computer vision, where understanding the structure and boundaries of objects is essential.

Comparison of Filtering Techniques



'\nExplanation: This displays the images processed by different filtering techniques (Gaussian,\nMedian, and Bilateral) sid

Comparison:

Gaussian Blur:

Applies a Gaussian kernel to smooth the image, which reduces noise and detail.

It's a linear filter that averages surrounding pixels using a Gaussian function, leading to a softer blur effect.

Best used for reducing overall noise in the image but may cause a loss of edge sharpness.

Median Blur:

Replaces each pixel's value with the median value of the pixels in a surrounding neighborhood.

Great for removing salt-and-pepper noise without affecting the edges as much as Gaussian blur.

Preserves edges better than Gaussian Blur but can introduce some artificial texture.

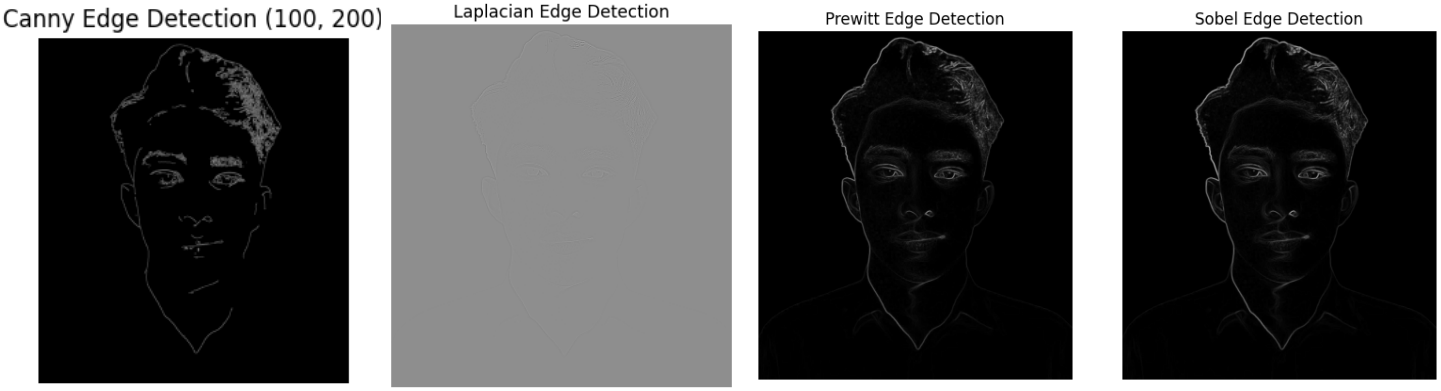
Bilateral Filter:

Smooths images while preserving edges by considering both spatial distance and pixel intensity difference.

Reduces noise without losing sharp edges, which makes it particularly effective for facial image processing.

More computationally expensive but provides the best balance between noise reduction and edge preservation.

Comparison of Edge Detection Techniques



Technique	Purpose	Operation	Use Case
Sobel Edge Detection	Detect edges in the image	Computes the gradient magnitude of the image using horizontal and vertical filters (Sobel kernels).	Edge detection, feature extraction, computer vision applications
Laplacian Edge Detection	Detect edges in the image	Uses the second derivative of the image, highlighting regions where the intensity changes rapidly.	Edge detection, image sharpening, identifying regions of rapid intensity change
Prewitt Edge Detection	Detect edges in the image	Similar to Sobel, uses horizontal and vertical filters (Prewitt kernels) with simpler convolution masks.	Edge detection, feature extraction, computationally efficient edge detection
Canny Edge Detection	Detect edges with precision	Uses a multi-stage algorithm to detect edges by finding the intensity gradient and applying non-maximum suppression and edge tracking by hysteresis.	Precise edge detection, reducing noise, applications requiring detailed and accurate edge maps

Sobel Edge Detection: Provides a robust method for edge detection with smoothing, making it effective in noisy images.
Laplacian Edge Detection: Highlights rapid intensity changes, offering a more isotropic edge detection approach. Ideal for applications needing precise edge detection and image sharpening.
Prewitt Edge Detection: A simpler, computationally efficient method for edge detection, suitable for real-time applications and scenarios with limited processing power.
Canny Edge Detection: A precise edge detection method that reduces noise and provides detailed edge maps, suitable for applications requiring high accuracy.

Comparison of Blurring Techniques

Bilateral Filter

Gaussian Blur (5x5)

Motion Blur

Median Blur (5x5)

Unsharp Mask (Sharpening)

Box Filter

Technique	Purpose	Operation	Use Case
Gaussian Blur	Reduce noise and detail	Applies a Gaussian function to blur the image, effectively reducing noise and detail.	Smoothing, noise reduction, pre-processing for edge detection
Median Blur	Reduce noise	Replaces each pixel's value with the median value of the neighboring pixels.	Removing salt-and-pepper noise, preserving edges while reducing noise
Bilateral Filter	Reduce noise while preserving edges	Applies a Gaussian blur that considers pixel intensity differences, preserving edges.	Denoising while keeping edges sharp, used in facial and detail-preserving smoothing
Box Filter	Smooth image	Computes the average of all the pixels in the kernel area, resulting in a uniform blur.	Simple smoothing, reducing image detail
Motion Blur	Simulate motion	Blurs the image in a specific direction to simulate the appearance of motion.	Creating motion effects, simulating camera movements
Unsharp Mask	Sharpen image	Subtracts a blurred version of the image from the original, enhancing edges and details.	Sharpening, enhancing edges, increasing contrast for better visual clarity