Computer Vision

Class 06

Raquel Frizera Vassallo

Class 06

01 Image Binarization

Global Thresholding
Adaptive Thresholding
Otsu's Binarization

02 Image Smoothing and Blurring

Average Blurring
Gaussian Blurring

03 Edge Detection

Sobel
Canny

04 Morphological Operations

Dilation
Erosion
Opening
Closing

01

Image Binarization

Global Thresholding
Adaptive Thresholding
Otsu's Binarization

Image Binarization

Is the process of converting an image into a binary image where each pixel is either:

- Black (0)
- White (255)

Binarization simplifies an image, making it easier to perform tasks such as:

- Object detection
- Text recognition (OCR)
- Edge detection
- Blob counting and measurement

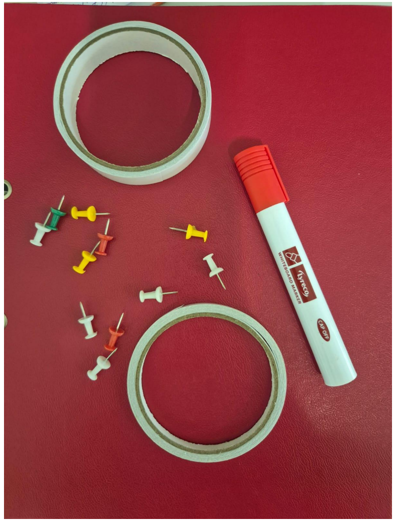


Image Binarization

Global Thresholding

- Pixels above the threshold are set to white (foreground).
- Pixels below the threshold are set to black (background).

Adaptive Thresholding

- Computes local thresholds for different regions of the image, useful under uneven lighting conditions.
- Different thresholds for different regions of the same image.
- Adaptive Mean Thresholding:
  - The threshold is the mean of the neighbourhood area minus a constant C
- Adaptive Gaussian Thresholding
  - The threshold is a gaussian-weighted sum of the neighbourhood values minus the constant C

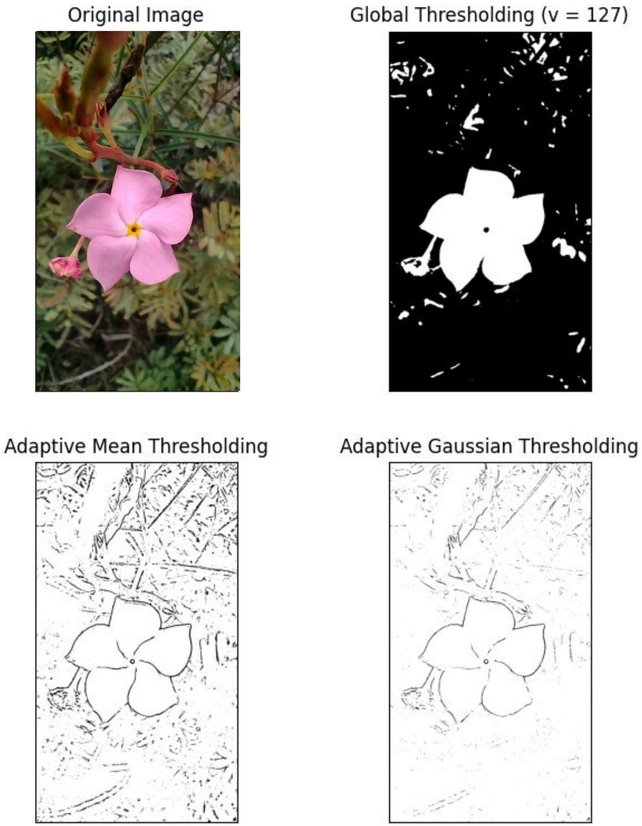


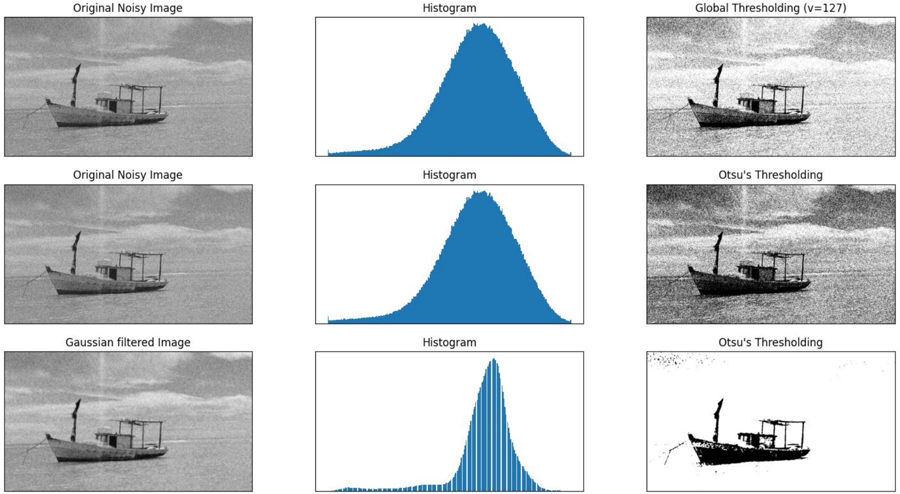
Image Binarization

Otsu's Binarization

- It is a global thresholding, but Otsu's method determines it automatically.
- An optimal global threshold is determined from the image histogram.

Example: Noisy image.

- First: global thresholding (127).
- Second: Otsu's thresholding is applied directly.
- Third: filter with a 5x5 gaussian kernel to remove the noise, then Otsu's thresholding is applied.



Noise filtering is a good practice before performing Image Binarization and Edge detection.

02

Image Smoothing and Blurring

Average Blurring
Gaussian Blurring

Image Smoothing and Blurring

Happens when you get a picture out of focus.
Sharper regions in the image lose their detail.

We can blur or smooth and image on purpose by applying a low-pass filter to the image.

Why?

- To reduce the amount of noise and detail in an image.
- Smaller details in the image are smoothed out and we are left with more of the structural aspects of the image.
- That helps many image processing operations.

Each pixel in the image is mixed in with its surrounding pixel intensities.
That is why the pixel and the image become blurred.

Image Smoothing and Blurring

Simple Average Blurring

- Takes an area surrounding a central pixel, averages all these pixels, and replaces the central pixel with the average.
- Reduces noise but blurs edges.
- Done by convolution using a kernel.
- The kernel slides from left-to-right and from top-to-bottom for each and every pixel in our input image.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} /9$$

kernel

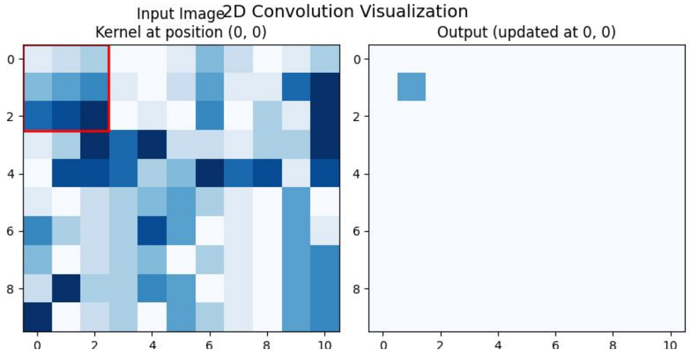


Image Smoothing and Blurring

Gaussian Blurring

- Weights neighboring pixels according to a Gaussian distribution.
- Pixels closer to the central pixel contribute more “weight” to the average.
- Reduces high-frequency noise while preserving edges better than averaging.

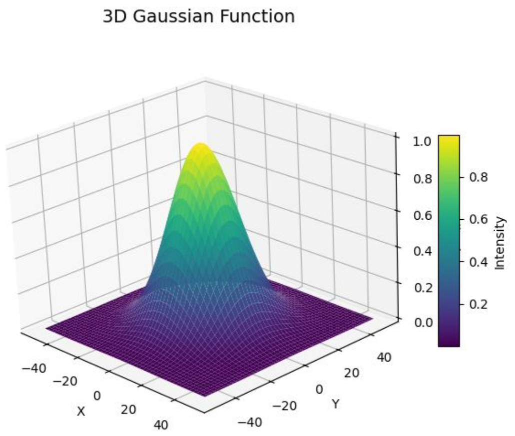


Image Smoothing and Blurring

Other methods:

- **Median Filtering**
  - Replace each pixel with the median of neighboring pixel values.
  - Very effective for salt-and-pepper noise.
  - Preserves edges better.
- **Bilateral Filtering**
  - Combines spatial proximity and intensity similarity to blur while preserving edges.
  - Smooths regions but keeps edges sharp.
  - Ideal for denoising without losing details.
- **Non-Local Means Denoising**
  - Averages similar patches across the whole image, not just local neighbors.
  - Excellent denoising for textures and details.
  - Computationally heavier.
- **Custom Kernel Convolution**
  - Apply a custom convolution kernel.
  - You can design your own smoothing or sharpening filters.

Image Smoothing and Blurring

Method	Preserves Edges	Speed	Best For
Box / Averaging	❌ No	✅ Fast	Simple blur
Gaussian	⚪ Partial	✅ Fast	Noise reduction
Median	✅ Yes	⚪ Medium	Salt & pepper noise
Bilateral	✅ Yes	⚪ Medium-Slow	Edge-preserving smooth
Non-Local Means	✅ Yes	❌ Slow	High-quality denoising

03

Edge Detection

Sobel
Canny

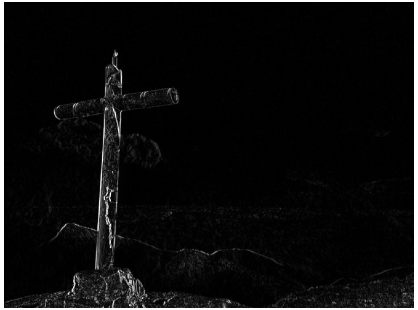
Edge Detection

Sobel Edge Detection

- Highlights edges in an image by measuring the intensity gradient in both the horizontal (x) and vertical (y) directions.
- G_x enhances vertical edges.
- G_y enhances horizontal edges.
- Simple and fast – good for general edge detection.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



+

=

Edge Detection

Canny Edge Detection

- Robust and flexible.
- Considered the gold standard in classical edge detection.
- Follows a four-stage process:
    - Noise Reduction
    - Calculating the Intensity Gradient of the Image
    - Suppression of False Edges (non-maximum suppression)
    - Hysteresis Thresholding (smaller and larger thresholds)

Sobel

Canny

Edge Detection

Other methods:

- **Prewitt Operator**
  - Similar to Sobel but uses a simpler averaging of differences.
  - Slightly less accurate but computationally lighter.

- **Roberts Cross Operator**
  - Uses 2×2 kernels to estimate the gradient.
  - Detects edges at diagonal orientations.
  - Works well for simple, high-contrast images.

- **Scharr Operator**
  - A more accurate version of Sobel, optimized for rotational symmetry.
  - Provides better gradient estimation and edge direction accuracy.

- **Laplacian Operator**
  - Based on the second derivative (measures rate of change of gradient).
  - Detects edges in all directions but is sensitive to noise.
  - Often used after smoothing (e.g., Gaussian blur).

Edge Detection

Method	Accuracy	Noise Sensitivity	Edge Thinness	Speed	Comments / Use Cases
Sobel	⭐⭐☆	⚠️ Moderate	⚪ Medium	⚡ Fast	Simple, good for basic edge detection tasks.
Prewitt	⭐⭐☆	⚠️ Moderate	⚪ Medium	⚡ Fast	Similar to Sobel, slightly less accurate.
Roberts	⭐⭐☆	⚠️ High	⚪ Medium	⚡⚡ Very fast	Detects diagonal edges; good for small images.
Scharr	⭐⭐⭐	⚠️ Low	✅ Thin	⚡ Fast	Improved version of Sobel, better gradient estimation.
Laplacian	⭐⭐☆	❌ High	⚪ Medium	⚡ Fast	Detects edges in all directions; often used after Gaussian blur.
Canny	⭐⭐⭐⭐	✅ Low	✅ Thin & Continuous	⚡ Medium	Gold standard in classical edge detection.

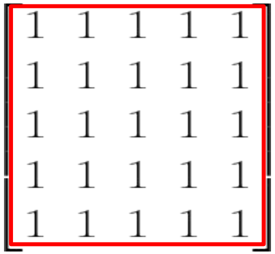
04

Morphological Operations

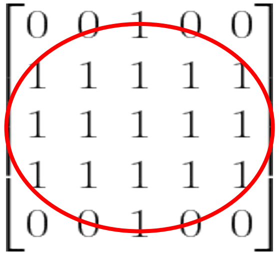
Dilation
Erosion
Opening
Closing

Kernel or Structuring Element

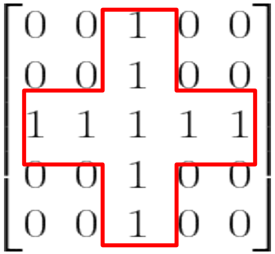
- Small matrix used for morphological operations like dilation, erosion, opening, and closing.
- It defines:
  - The shape.
  - The size of the neighborhood that affects each pixel during processing.
- The kernel slides (is convolved) over the image.
- At each position, it determines how the pixels under it should be modified — for example, whether to add or remove pixels (in dilation or erosion).
- The center of the kernel corresponds to the pixel currently being processed.



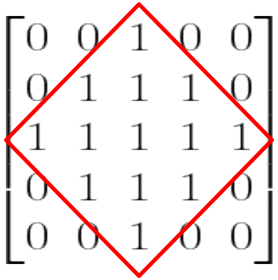
Rectangular
5x5



Elliptical
5x5



Cross-shaped
5x5



Diamond-shaped
5x5

Morphological Operations

Dilation

- Expands the bright (white) regions in a binary image.
- The kernel slides over the image and adds pixels to object boundaries **wherever it touches a white pixel.**
- Fills small holes and connects nearby objects.
- Used for emphasizing features, closing small gaps, joining broken parts.

Morphological Operations

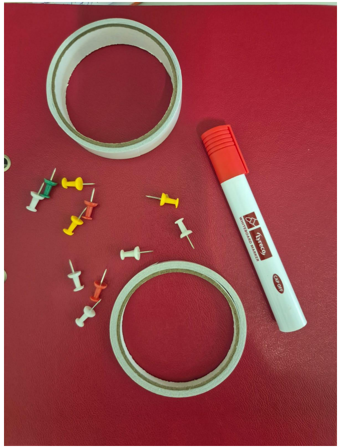
Erosion

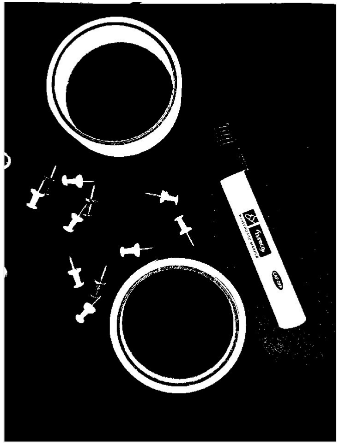
- Shrinks the bright (white) regions in a binary image.
- The kernel slides over the image and removes pixels from object boundaries wherever it **doesn't fully fit inside the white region.**
- Removes small white noise and separates touching objects.
- Used for cleaning up small artifacts, reducing object size.

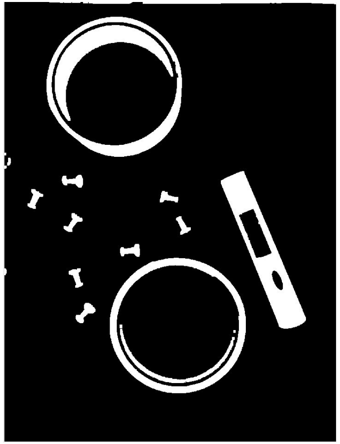
Morphological Operations

Opening

- Erosion followed by dilation.
- Removes small noise or thin protrusions while keeping the main shape of larger objects.
- Used for noise removal without losing important details and enlarging small gaps or holes inside objects or between objects.







Morphological Operations

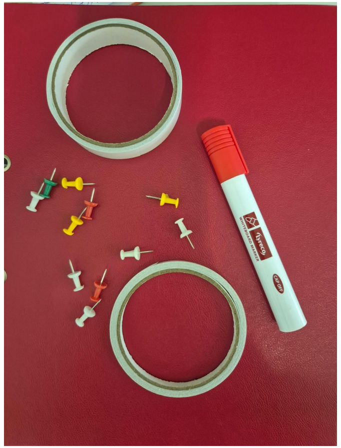
Closing

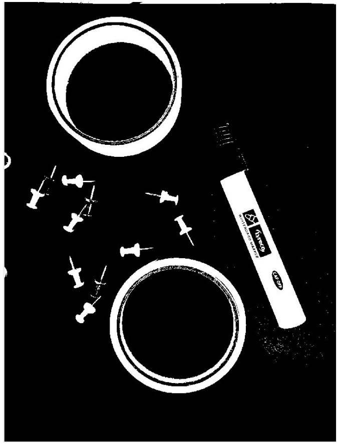
- Dilation followed by erosion.
- Fills small holes or gaps inside objects while preserving overall size.
- Used for closing small black regions inside white objects.

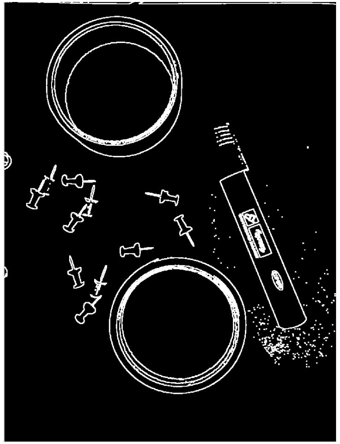
Morphological Operations

Morphological Gradient

- Difference between dilation and erosion of an image.
- The result will look like the outline of the object.







Credits

Open CV:
https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html
https://opencv.org/blog/edge-detection-using-opencv/

PyImage Search:
http://pyimagesearch.com/2021/04/28/opencv-smoothing-and-blurring/

LearnOpenCV (by BigVision):
https://learnopencv.com/edge-detection-using-opencv/

Python Geeks:
https://pythongeeks.org/dilation-and-erosion-in-opencv/