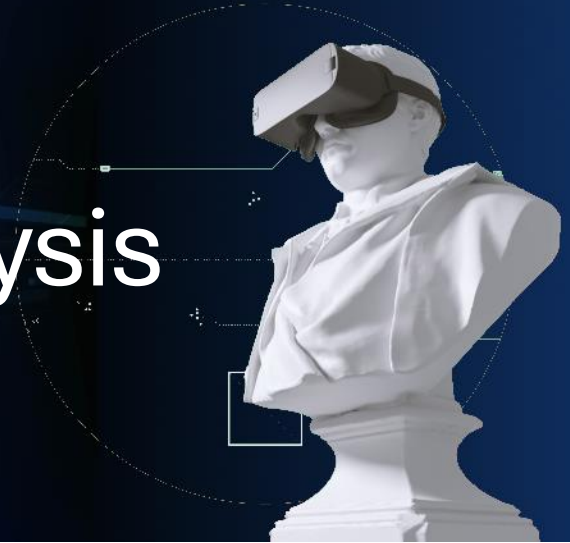


# MSc in Digital Humanities



## Image Processing/Analysis



## Image segmentation

George Pavlidis, Paraskevi Tzouveli, Christos Tsinos

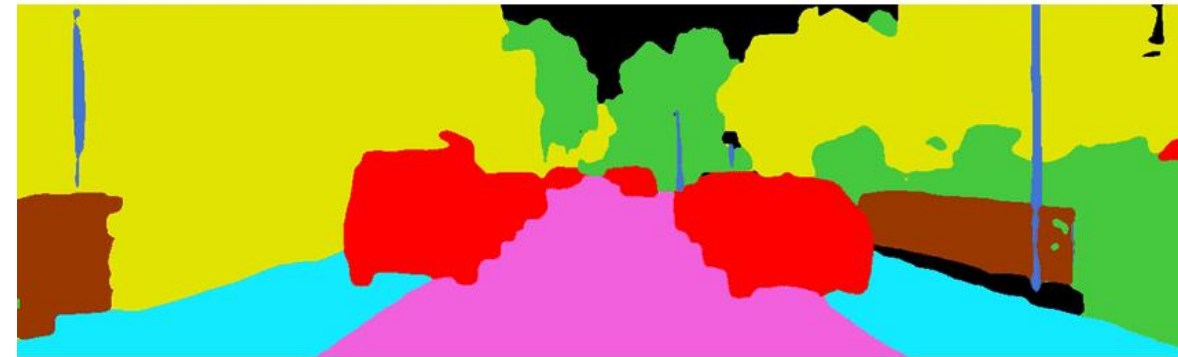
[https://colab.research.google.com/drive/1EAnRU\\_JBdhbCfHeS2lc2xg2YDwfl-zG4?usp=sharing](https://colab.research.google.com/drive/1EAnRU_JBdhbCfHeS2lc2xg2YDwfl-zG4?usp=sharing)

# Introduction



# Introduction

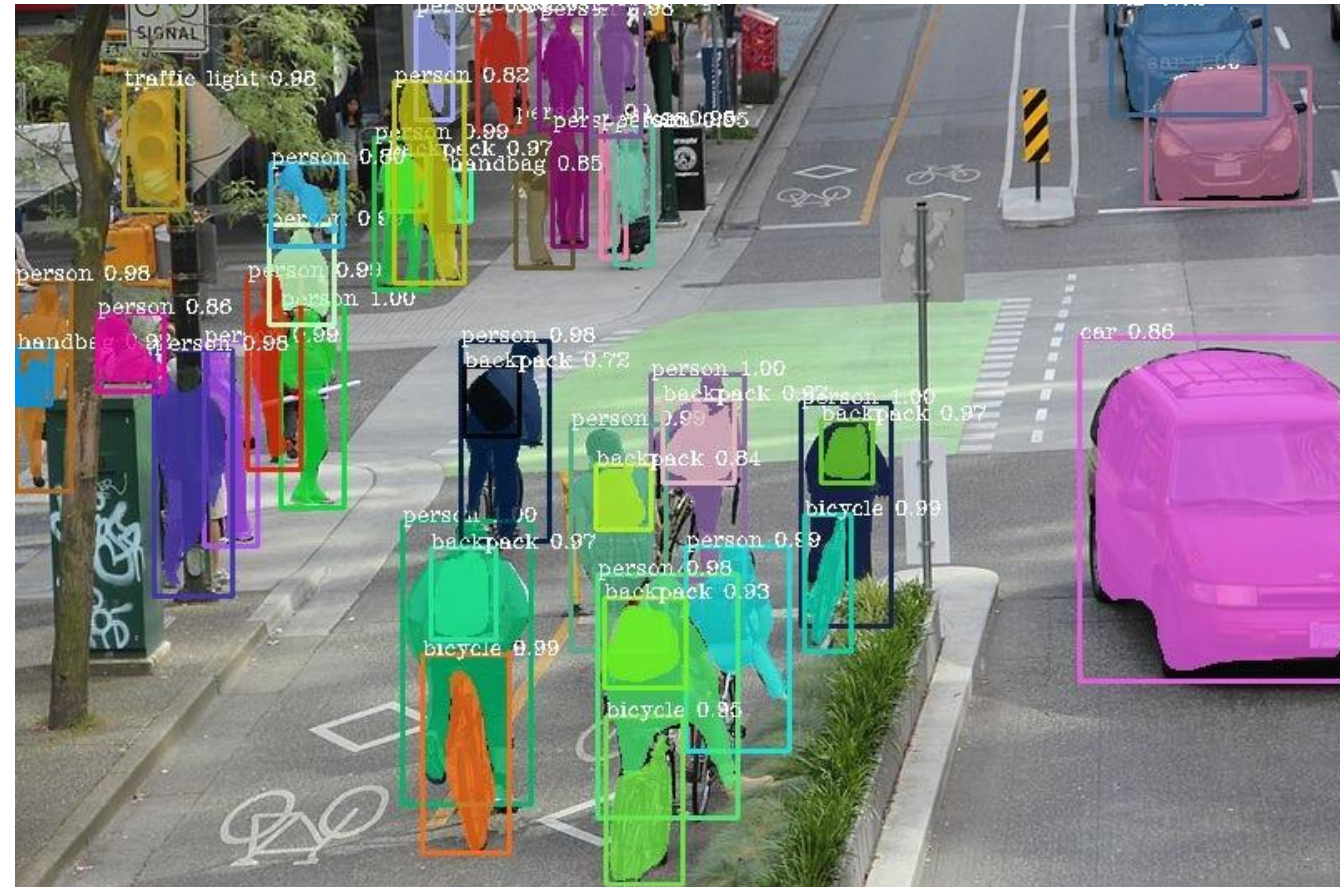
- Definition
  - Image segmentation is the process of partitioning a digital image into multiple segments to simplify or change its representation into something more meaningful and easier to analyze
- Purpose
  - Identifying objects, regions, or boundaries within an image
- Relevance
  - Crucial for applications in computer vision, medical imaging, and automated systems



|                                                                                            |                                                                                                  |                                                                                                |                                                                                               |
|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
|  Road |  Sidewalk   |  Building |  Fence   |
|  Pole |  Vegetation |  Vehicle  |  Unlabel |

# Introduction

- Importance of segmentation
  - Segmentation is fundamental in image analysis, enabling the identification of regions of interest
- Challenges
  - Complex scenes with overlapping or occluded objects
  - Variability in lighting, texture, and shapes
- Role of Segmentation
  - Improves analysis by isolating relevant structures
  - Enables accurate object detection and classification



# Edge Detection - The Foundation of Segmentation



- Why Edges
  - Edges represent **boundaries of regions** or **boundaries of objects**
  - **Boundaries of “parts”** within an object
  - Represent **boundaries of shadowing or lighting changes** in an image
  - **Carry the majority of information** in an image (critical for HVS)
- Definition
  - An edge is a set of pixels where intensity changes sharply
- Applications
  - Region identification
  - Data reduction while preserving content

# Edge Detection - The Foundation of Segmentation

- Carry the majority of information in an image ???

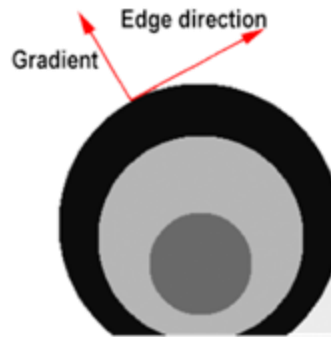


Figure 3.1. Gradient and direction of an edge

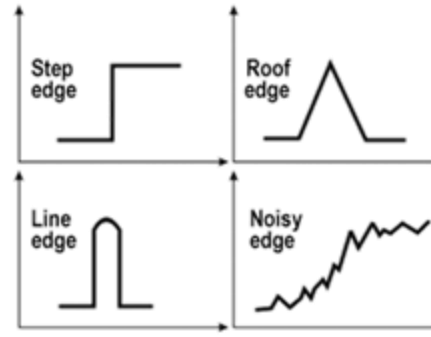


Figure 3.2. Typical edge profiles in images

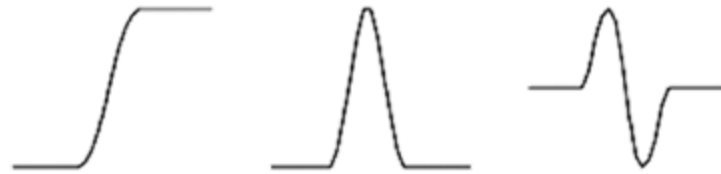
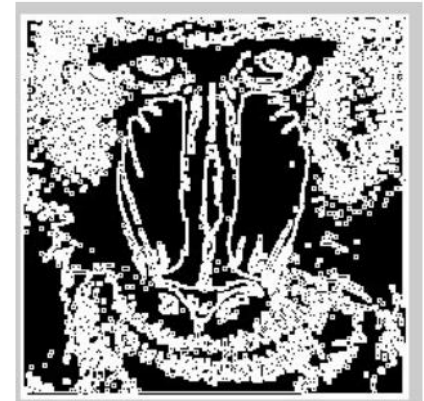
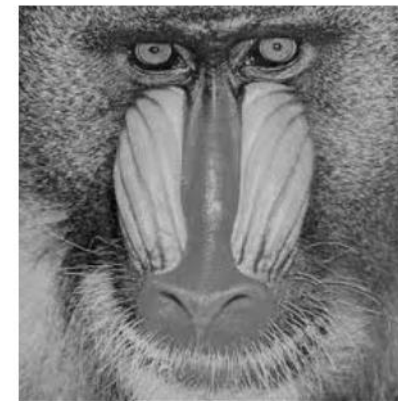
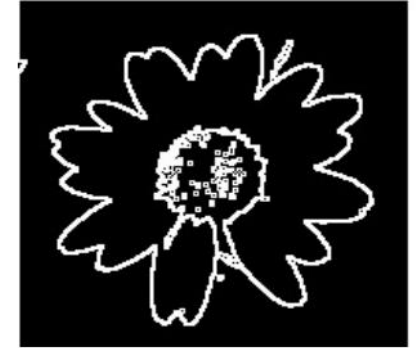


Figure 3.3. Representation of (a) an edge, (b) its first and (c) its second derivative



# Methods of Edge Detection



- Gradient Methods
  - Calculate the magnitude and orientation of changes in intensity
  - Examples: Prewitt, Sobel, and Roberts-Cross methods
- Second Derivative Methods
  - Use the Laplacian operator to find zero-crossings
  - Example: Marr-Hildreth method (LoG)
- Optimal Method
  - Canny Edge Detector (multi-step process with smoothing and non-maximal suppression)

# Gradient methods for segmentation

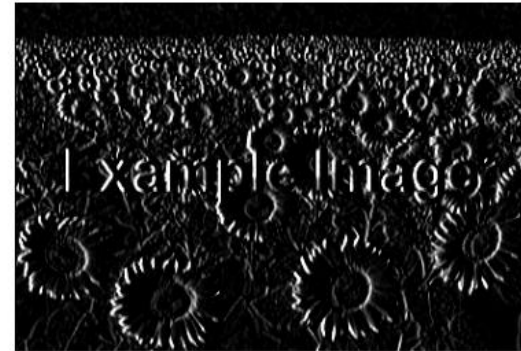


- Gradient methods and Canny

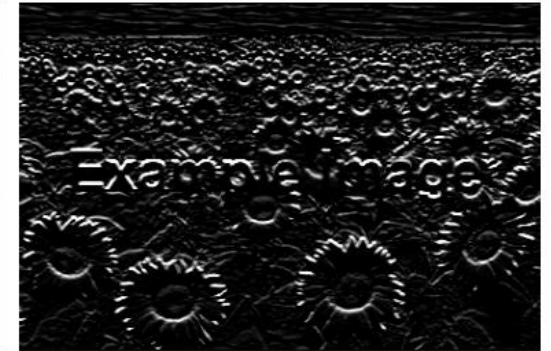
Original Image



Prewitt - X



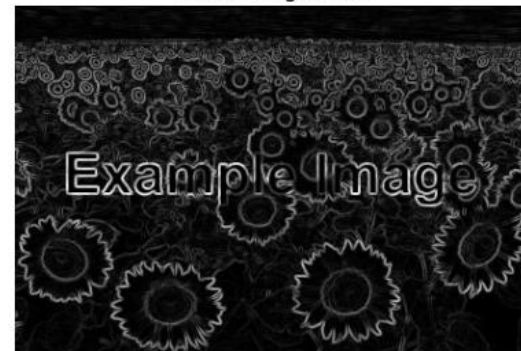
Prewitt - Y



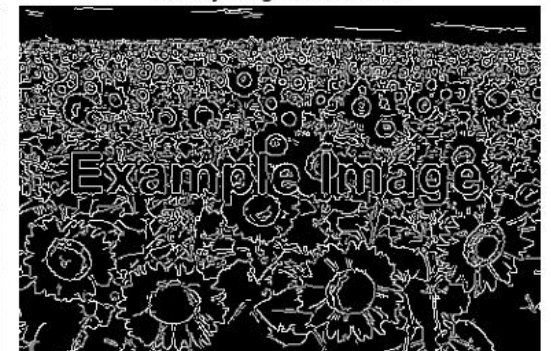
Prewitt Combined



Sobel Magnitude



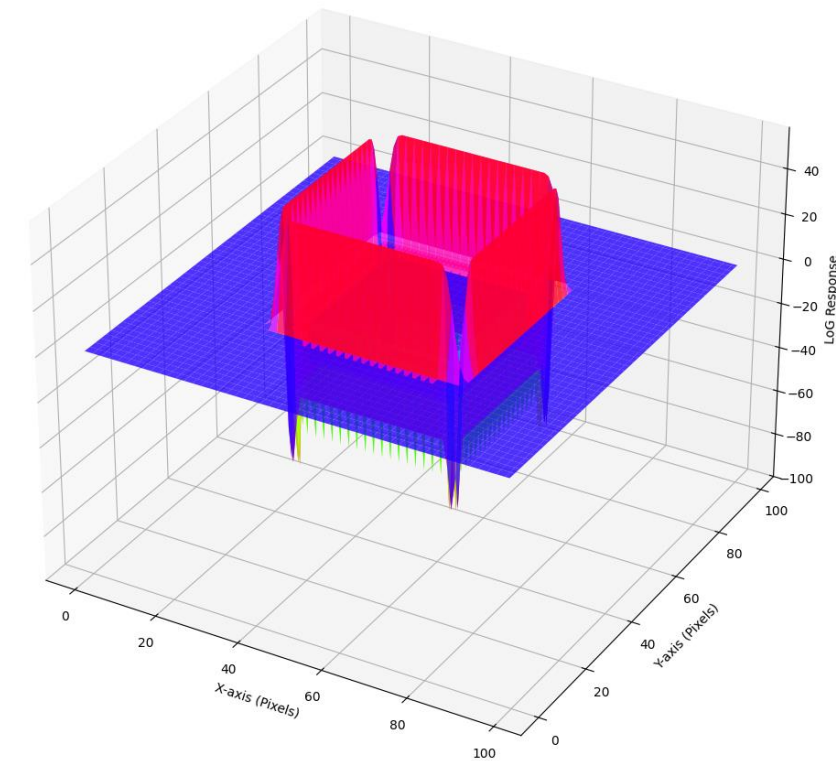
Canny Edge Detection



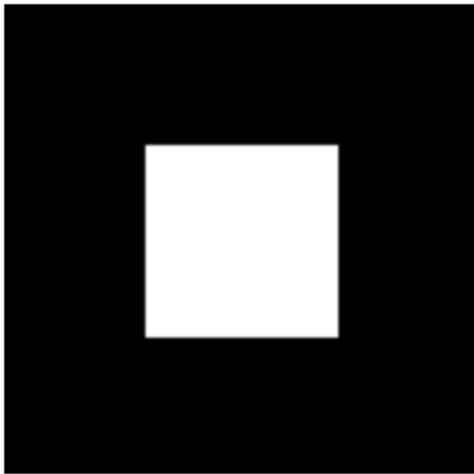
# Gradient methods for segmentation

- LoG and zero crossings

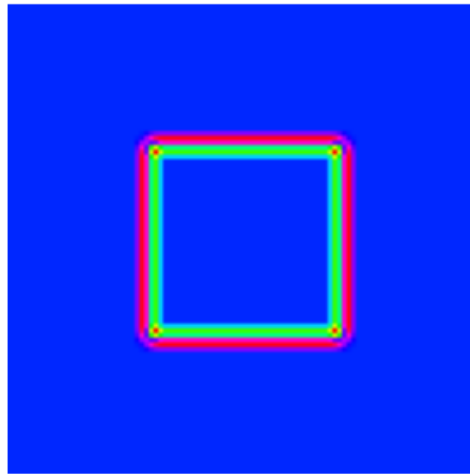
3D Visualization of Laplacian of Gaussian and Zero Crossings



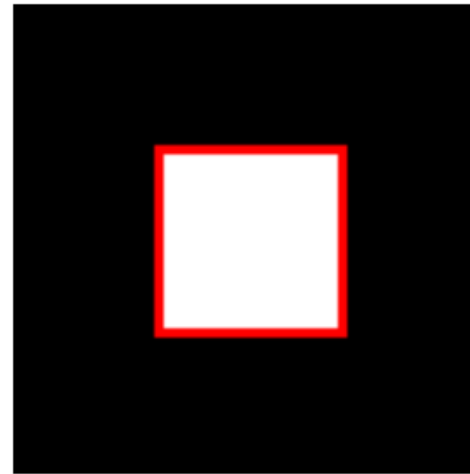
Original Image



Laplacian of Gaussian (LoG)



Zero Crossings



# Segmentation based on thresholding



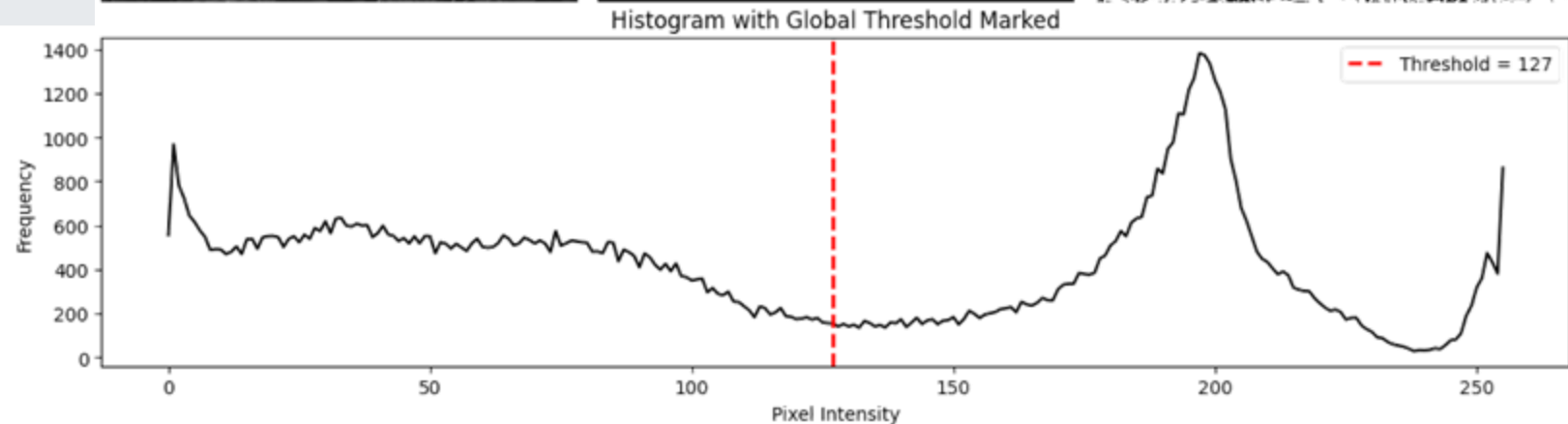
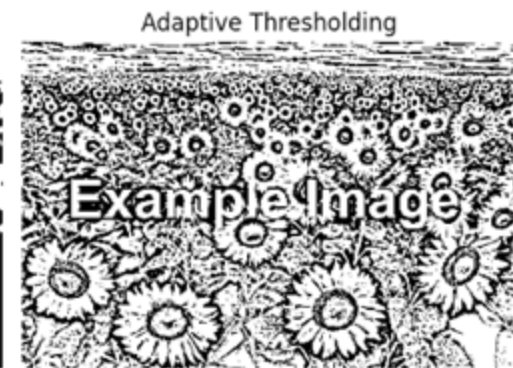
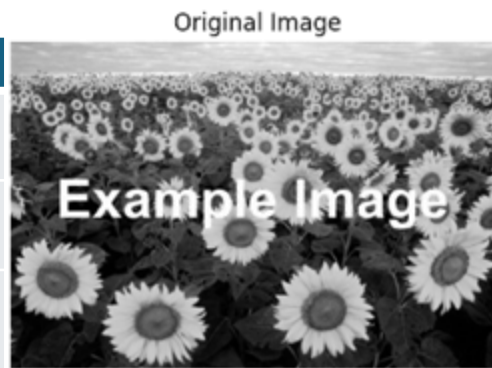
- Simplest segmentation method
  - Segments an image by converting grayscale values into binary regions based on a threshold
- Types
  - Global thresholding
  - Adaptive thresholding (region-specific)
- Applications
  - Object-background separation

# Segmentation based on thresholding



- Thresholding methods

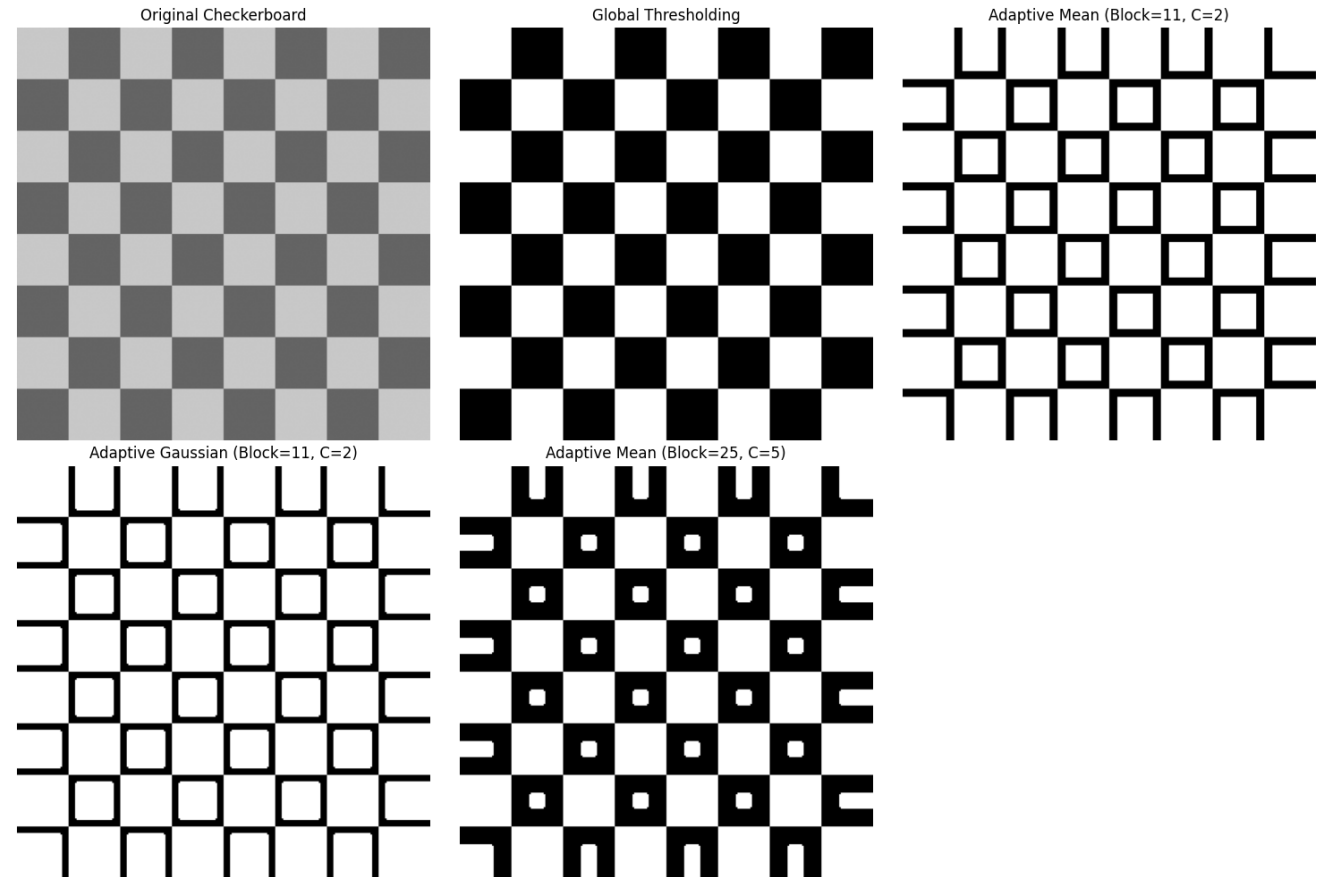
| Method              | Threshold Type                                     | Best For                                          |
|---------------------|----------------------------------------------------|---------------------------------------------------|
| Global Thresholding | Single fixed value (e.g., 127)                     | Simple images with uniform lighting               |
| Adaptive Mean       | Uses <b>average of neighborhood pixels</b>         | Handles moderate lighting variations              |
| Adaptive Gaussian   | Uses <b>Gaussian-weighted sum</b> of nearby pixels | Best for images with strong shadows or highlights |



# Segmentation based on thresholding



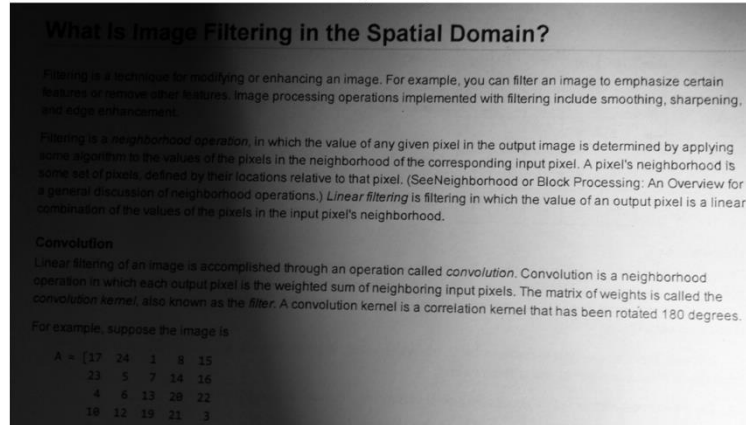
- Global and adaptive thresholding
  - Global Thresholding
    - global threshold=127
  - Adaptive Thresholding Methods
    - **AM, Block=11, C=2**
      - Uses the mean of the neighborhood area with a block size of 11 and a constant C=2
    - **AG, Block=11, C=2**
      - Uses a Gaussian-weighted sum of the neighborhood area with a block size of 11 and a constant C=2
    - **AM, Block=25, C=5**
      - Uses the mean of the neighborhood area with a larger block size of 25 and a constant C=5



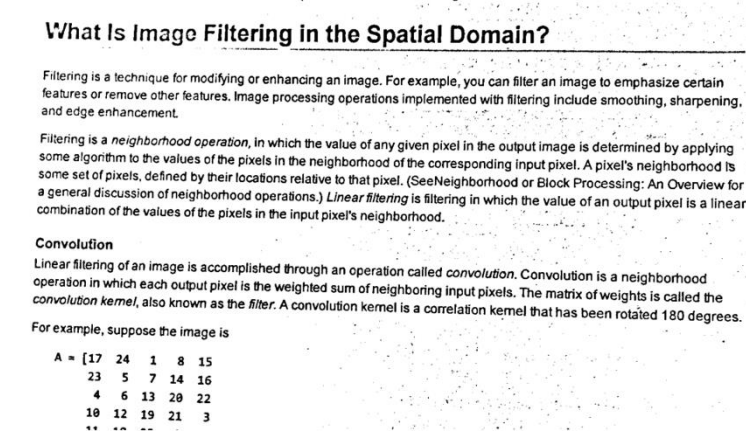
# Segmentation based on thresholding

- Global and adaptive thresholding
  - Global Thresholding
    - global threshold=127
  - Adaptive Thresholding Methods
    - **AM, Block=25, C=5**
      - Uses the mean of the neighborhood area with a block size of 11 and a constant C=2
    - **AG, Block=11, C=2**
      - Uses a Gaussian-weighted sum of the neighborhood area with a block size of 11 and a constant C=2

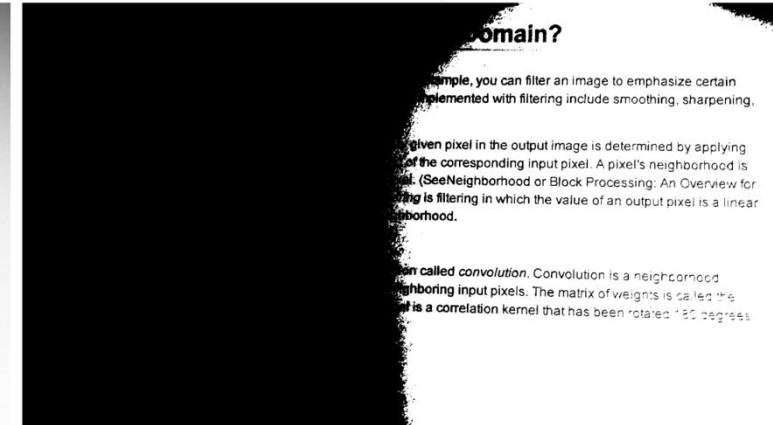
Original Image



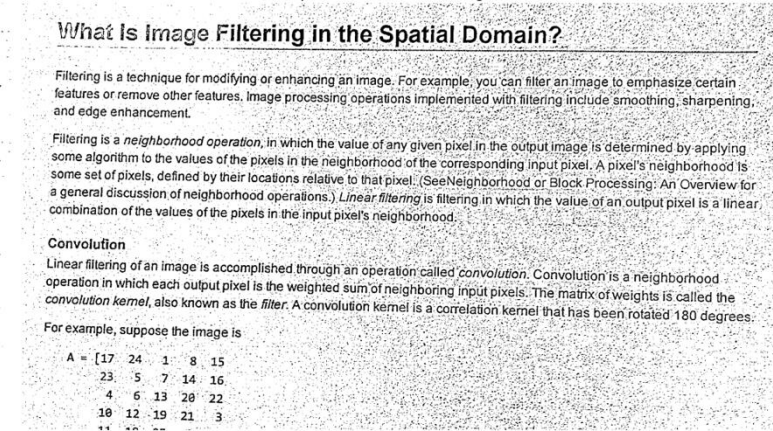
Adaptive Mean Thresholding (25,5)



Global Thresholding (Fails)



Adaptive Gaussian Thresholding (11,2)

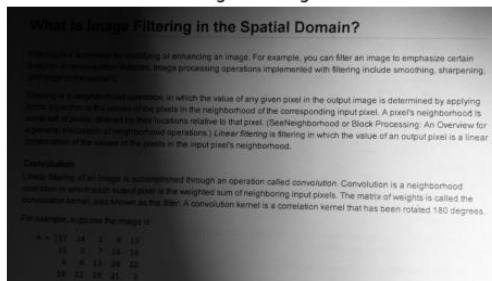


# Advanced Thresholding Techniques

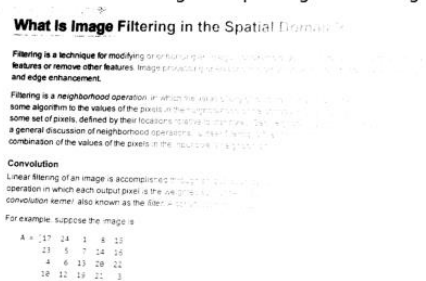


- Multi-thresholding
  - Produces multiple regions instead of binary
- Semi-thresholding
  - Retains parts of the image while removing the background
- Applications
  - Complex scenes with varying intensities

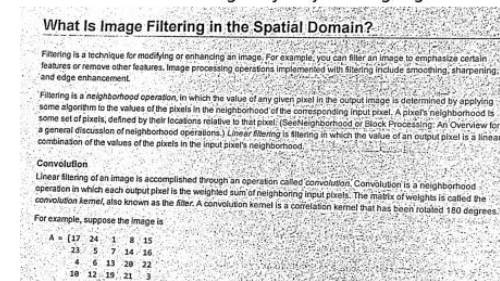
Original Image



Multi-Thresholding: Multiple Segmented Regions



Semi-Thresholding: Key Objects Highlighted



# Region-Based Segmentation



- Concept
  - Groups pixels into regions based on similarity
  - Utilizes homogeneity criteria like luminance, texture, or color
- Methods
  - Region growing (bottom-up approach)
  - Region splitting (top-down approach)
  - Split-and-merge methods

# Region-Based Segmentation



- Region growing and split & merge

Region Growing: Pseudocolor Regions

**What Is Image Filtering in the Spatial Domain?**

Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement.

Filtering is a *neighborhood operation*, in which the value of an output pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel. (See *Neighborhood or Block Processing: An Overview for a general discussion of neighborhood operations*.) *Linear filtering* is filtering in which the value of an output pixel is a linear combination of the values of the pixels in the input pixel's neighborhood.

Linear filtering of an image is accomplished through an operation called *convolution*. Convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels. The matrix of weights is called the *convolution kernel*, also known as the *filter*. A convolution kernel is a correlation kernel that has been rotated 180 degrees.

Split and Merge: Hierarchical Regions

**What Is Image Filtering in the Spatial Domain?**

Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement.

Filtering is a *neighborhood operation*, in which the value of an output pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel. (See *Neighborhood or Block Processing: An Overview for a general discussion of neighborhood operations*.) *Linear filtering* is filtering in which the value of an output pixel is a linear combination of the values of the pixels in the input pixel's neighborhood.

**Convolution**

Linear filtering of an image is accomplished through an operation called *convolution*. Convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels. The matrix of weights is called the *convolution kernel*, also known as the *filter*. A convolution kernel is a correlation kernel that has been rotated 180 degrees.

For example, suppose the image is

$$A = \begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 18 & 12 & 19 & 21 & 3 \end{bmatrix}$$

# Watershed Segmentation



- Inspiration
  - Derived from topographical concepts (watersheds and basins)
- Method
  - Intensity values treated as elevations
  - Local minima represent basins, and edges are watersheds
- Challenges
  - Tends to over-segment without additional constraints.
- Visuals:
  - - Pseudocolor representation of segmented regions in a complex image.

# Watershed Segmentation



- Presudolored regions

Watershed Segmentation: Pseudocolor Regions

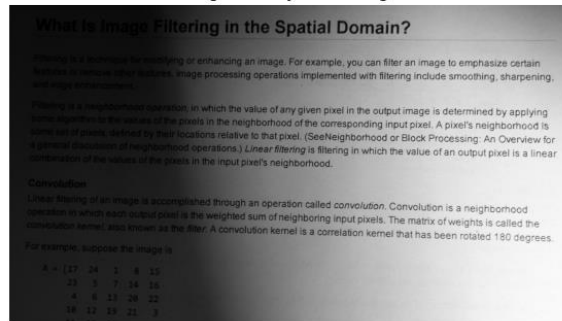


# Watershed Segmentation

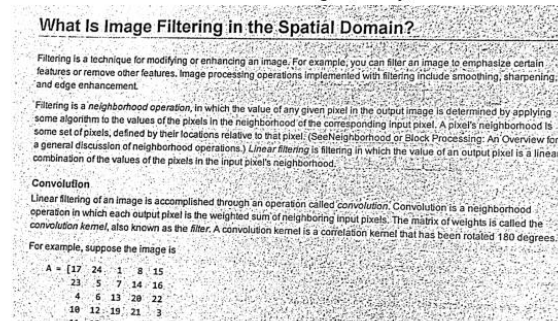


- Application in complex lighted document image

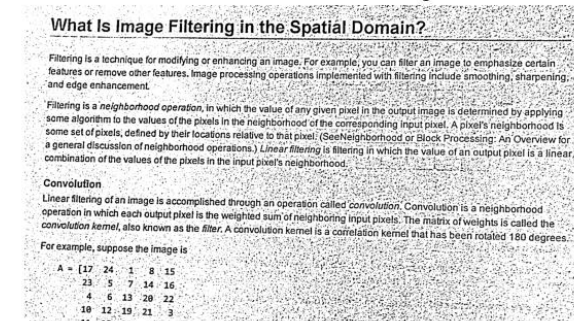
Original Grayscale Image



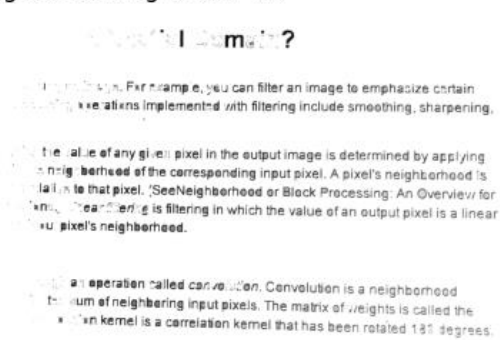
Thresholded Image (Binary)



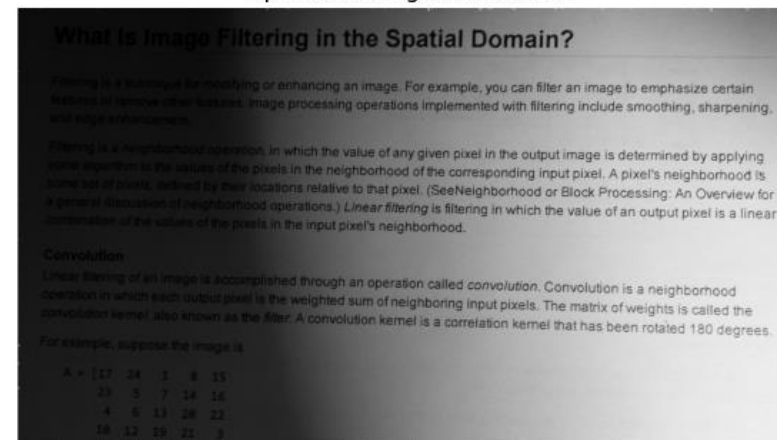
Contours on Thresholded Image



Region Growing (Flood-fill)



Split and Merge (Contours)



# Clustering-Based Segmentation



- Definition
  - Groups pixels into clusters based on similarity.
- Popular Techniques
  - K-means clustering
  - Fuzzy C-means clustering
- Advantages
  - Works well for natural images

# Clustering-Based Segmentation



- Example application in mixed document image
  - Clustering in the color space domain

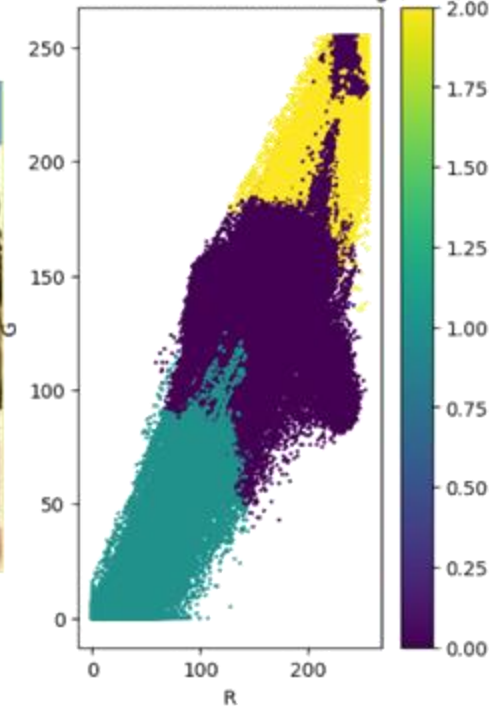
Original Image



K-means Clustering



Scatter Plot of K-means Clustering (RGB)



Pseudocolored K-means Segmentation



# Probabilistic and Bayesian Segmentation



- Overview
  - Utilizes statistical models to describe pixel likelihoods
- Key Techniques
  - Maximum likelihood estimation (MLE)
  - Bayesian inference for assigning pixels to regions
- Applications
  - Medical imaging, object detection

# Probabilistic and Bayesian Segmentation



- Example on a mixed document image

- Likelihood map: using the probability of each pixel belonging to the second Gaussian component
- Segmentation: by thresholding the likelihood map

Original Image



Pixel Likelihoods (GMM)



Segmentation Using Bayesian Inference



2 classes

GMM Segmented Image



3 classes

GMM Segmented Image



8 classes

# Neural Network-Based Segmentation



- Concept
  - Uses machine learning to learn features and segment images
- Advantages
  - Adaptive to complex patterns and textures
  - Handles multi-class segmentation efficiently
- Applications
  - Autonomous vehicles, medical diagnostics

# Neural Network-Based Segmentation

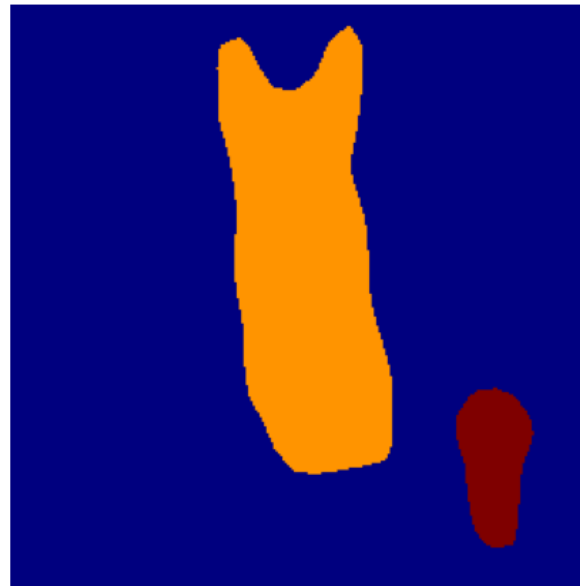


- Example
  - Application of a pre-trained (imagenet) **DeepLabV3** segmentation model
    - for multi-class image segmentation → 3 classes recognized

Original Image



Segmentation Mask



Overlay of Segmentation



# Graph-Based Segmentation



- Definition
  - Models images as graphs with nodes (pixels) and edges (relationships)
- Approach
  - Partition the graph to minimize similarity across groups
- Applications
  - Object recognition and contour detection

# Graph-Based Segmentation



- Example

- Sobel filter to detect edges in the grayscale image
- SLIC (Simple Linear Iterative Clustering) algorithm to segment the image into superpixels



# Hough Transform for Segmentation



- Definition
  - Detects shapes like lines and circles in images
- Key Idea
  - Maps image points to a parameter space to find objects
- Applications
  - Line detection in documents, circle detection in medical imaging

# Hough Transform for Segmentation

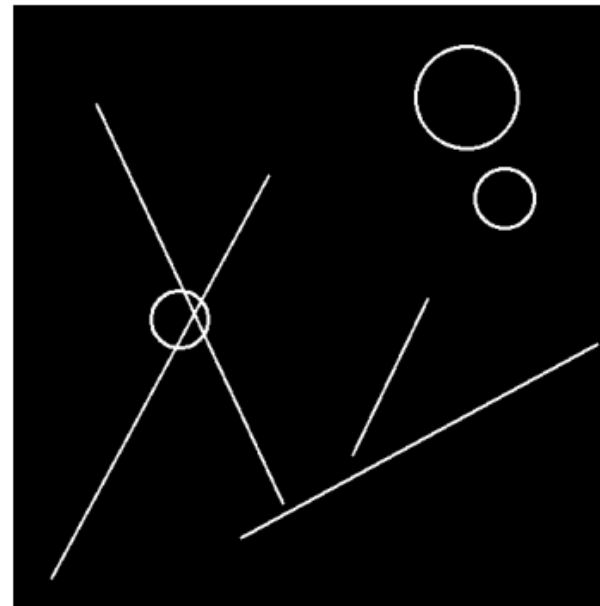


- Example

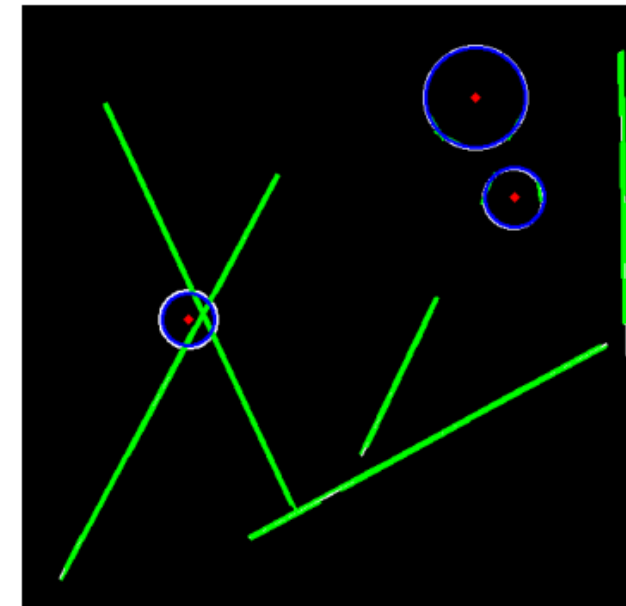
- Apply Canny edge detection to highlight significant edges in the image (in this case not needed)
- Convert edge pixels to the Hough space and identify the most significant lines
- Apply Hough Gradient Method to detect circular patterns in the image

| Feature         | Line Detection                         | Circle Detection              |
|-----------------|----------------------------------------|-------------------------------|
| Equation        | $\rho = x \cos \theta + y \sin \theta$ | $(x - a)^2 + (y - b)^2 = r^2$ |
| Parameters      | $(\rho, \theta)$                       | $(a, b, r)$                   |
| Accumulator Dim | 2D $(\rho, \theta)$                    | 3D $(a, b, r)$                |
| Complexity      | Lower                                  | Higher (requires more votes)  |

Original Synthetic Image



Detected Lines & Circles



# Advanced Region-Based Techniques



- Semantic Region Growing
  - Incorporates semantic properties for better results
- Hybrid Methods
  - Combines region growing with edge detection for precision
- Applications
  - Texture segmentation, feature-based analysis

# Advanced Region-Based Techniques



- Example

- Apply basic intensity-based region growing based
  - Seed placed in the gray area, and the intensity similarity threshold only allowed growth within this region
- Apply hybrid region growing with edge awareness
  - Applied Canny edge detection to refine segmentation
  - Prevented the region from crossing strong edges

Original Image



Basic Region Growing



Hybrid Region & Edge Segmentation



# Tree-Based Representations for Segmentation



- Concept
  - Uses hierarchical structures (e.g., Region Adjacency Graphs)
  - Represents regions and their relationships
- Advantages
  - Captures spatial relationships and hierarchical dependencies
- Applications
  - Scene segmentation, video analysis

# Tree-Based Representations for Segmentation



- Examples

- Convert to Grayscale & Compute Edges (Sobel)
- Apply Watershed Segmentation
- Construct Region Adjacency Graph (RAG)
  - represents how different segmented regions connect to each other (connectivity is based on 4-neighbor adjacency)

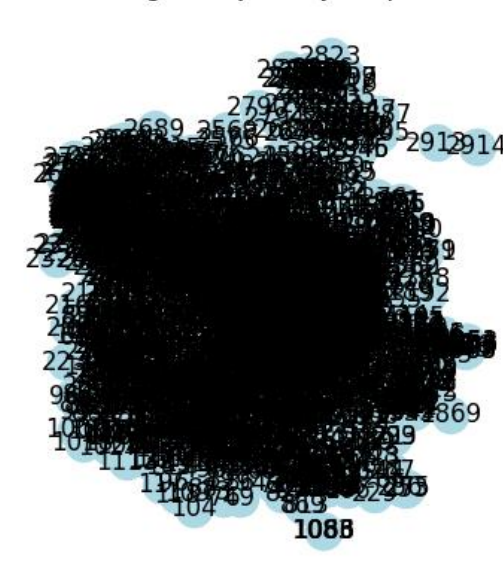
Original Image



Segmented Image



Region Adjacency Graph



# Dynamic Programming for Boundary Detection



- Overview
  - Finds optimal paths for boundaries using cost minimization
- Key Principles
  - Monotonic and separable objective functions
  - Recursive computation of optimal paths
- Applications
  - Medical imaging (e.g., lung boundary detection)

# Dynamic Programming for Boundary Detection



- Example
  - Edge Detection
    - Apply the Canny edge detector to highlight boundaries
  - Contour Extraction
    - Identify the most prominent boundaries using `cv2.findContours()`

# Snake Algorithms (Active Contours)



- Concept
  - Contours evolve under constraints to fit object boundaries
- Advantages
  - Handles smooth, continuous boundaries well
- Challenges
  - Sensitive to initialization and noise

# Snake Algorithms (Active Contours)



- Example

- Active Contours (Snakes) are used to detect object boundaries
  - by evolving an initial contour under the influence of external and internal forces
  - **External forces:** Attract the contour to image edges (e.g., via edge detection)
  - **Internal forces:** Keep the contour smooth and prevent it from distorting too much

Initial Contour and Image



Fitted Contour (Snake Algorithm)



# AI-Based Segmentation with Transformers



- Concept
  - Vision Transformers (ViTs) for segmentation learn global context using self-attention
- Advantages
  - Captures long-range dependencies effectively
- Applications
  - Semantic segmentation tasks in medical imaging and autonomous systems

# AI-Based Segmentation with Transformers



- Example
  - DETR (Detection Transformer)
    - utilizes transformer-based architecture for object detection rather than traditional convolution-based models



# Multi-Scale and Pyramid Networks



- Overview
  - Multi-scale processing captures information at different resolutions
- Examples
  - Feature Pyramid Networks (FPNs)
  - DeepLab (uses atrous convolutions for context capture)
- Applications
  - Satellite imaging, urban planning

# Multi-Scale and Pyramid Networks



- Example
  - DeepLabV3+ leverages atrous convolutions (dilated convolutions) to capture multi-scale features efficiently
  - Feature Pyramid Networks (FPNs) are another example, where feature maps at multiple resolutions are combined to improve detection accuracy



# Semi-Supervised Learning for Segmentation



- Concept
  - Leverages limited labeled data with a larger set of unlabeled data
- Approach
  - Combines supervised learning with consistency regularization
- Advantages
  - Reduces labeling effort significantly

# Semi-Supervised Learning for Segmentation



- Example

- Use a pre-trained segmentation model (e.g., DeepLabV3)
  - to generate segmentation predictions (masks) for the unlabeled image
- Predictions act as pseudo-labels for the image
  - treated as if they were actual labeled data

Original Image



Segmentation Mask (Pseudo-label)



Overlay (Mask + Original)



# Generative Adversarial Networks for Segmentation



- Concept
  - GANs generate realistic predictions by learning from adversarial training
    - Real vs fake segmentation text
- Approach
  - Generator predicts segmentation; discriminator evaluates its accuracy
- Applications
  - Synthetic data generation, domain adaptation

# Self-Supervised Learning for Segmentation



- Concept
  - Learns useful features from data without explicit labels
- Techniques
  - Contrastive learning, clustering-based pretraining
- Advantages
  - Effective in data-scarce environments

# Conclusion

- Segmentation is foundational for many image analysis applications
- Various methods are suited for different scenarios
- Future Trends
  - Integration with AI and deep learning
  - Real-time segmentation in dynamic environments

# Bibliography

