Two-Dimensional Histogram Based Bi-level Image Segmentation

Submitted

By

CHARSIMA S BU21EECE0100356

SIREESHA P BU21EECE0100210

SUCHITHRA C BU21EECE0100426

Under the Guidance of

Dr. Pankaj Khandway

(Assistant Professor)

(Duration: 01/July/2024 to 28/March/2025)

Department of Electrical, Electronics and Communication Engineering

GITAM School of Technology

GITAM

(DEEMED TO BE UNIVERSITY)

(Estd. u/s 3 of the UGC act 1956)

NH 207, Nagadenehalli, Doddaballapur taluk, Bengaluru-561203 Karnataka, INDIA.

DECLARATION

I/We declare that the project work contained in this report is original and it has been done by me under the guidance of my project guide.

Name:

Date: **Signature of the Student**

Department of Electrical, Electronics and Communication Engineering
GITAM School of Technology, Bengaluru-561203

CERTIFICATE

This is to certify that (P.Sireesha,S.Charisma,C.Suchithra) bearing (Regd. No.BU21EECE0100210,BU21EECE0100356,BU21EECE0100426)has satisfactorily completed Mini Project Entitled in partial fulfillment of the requirements as prescribed by University for VIIIth semester, Bachelor of Technology in "**Electrical, Electronics and Communication Engineering**" and **submitted this report during the academic year 2024-2025.**

[Signature of the Guide] **[Signature of HOD]**

Table of contents

Chapter 1: Introduction

Image segmentation is a fundamental process in computer vision and image processing, where an image is divided into distinct regions based on specific characteristics. Traditional segmentation methods, such as Otsu's and Kapur's thresholding techniques, primarily rely on pixel brightness to determine segmentation boundaries. However, these methods often struggle with noisy images or images containing complex textures, leading to inaccurate segmentation.

To overcome these challenges, this project explores a Two-Dimensional Histogram-Based Bilevel Image Segmentation approach. Unlike conventional methods, this technique considers not only the brightness of a pixel but also its relationship with surrounding pixels through Local Contextual Information (LCI). By incorporating LCI, the segmentation process captures contextual information, leading to more precise results, especially in noisy environments and edge-detection scenarios.

This research aims to develop and analyze the proposed segmentation method by comparing its performance with existing techniques. The study focuses on optimizing the algorithm for improved segmentation accuracy, making it suitable for applications such as medical imaging, defect detection, and remote sensing.

1.1Overview of the problem statement

Traditional thresholding methods in image segmentation primarily use brightness information, leading to inaccurate segmentation, particularly in noisy or complex images. To improve segmentation accuracy, it is necessary to account for the contextual information between pixels. This project explores a new thresholding method that constructs a two-dimensional histogram using the brightness of a pixel and the local relative entropy Local Contextual Information (LCI). of its neighboring pixels. By incorporating the LCI, which measures brightness differences between a pixel and its neighbors, more accurate segmentation can be achieved.

1.2 Objectives and Goals

To develop an improved image segmentation method based on two-dimensional histograms incorporating brightness and local relative entropy.
To compare the accuracy of the new method with traditional thresholding techniques such as Otsu and Kapur.
To optimize the method for effective image segmentation in noisy environments.

Chapter 2: Literature Review

Two-Dimensional Histogram-Based Segmentation:
To address the shortcomings of traditional methods, researchers have explored two-dimensional (2D) histograms for bilevel image segmentation, incorporating additional pixel information beyond simple brightness. Previous studies have demonstrated that including features such as edge details, gradients, or local texture information in the histogram can enhance segmentation accuracy. A two-dimensional histogram represents two different aspects of pixel information, where one axis typically represents pixel intensity (gray level), and the other axis can represent additional information, such as gradient magnitude, edge details, or local contextual information.

s.no	TITLE	AUTHOR	YEAR OF PUBLISHING	DRAWBACKS
1.	Two-Dimensional Histogram-Based Bi-Level Image Segmentation	Xu Ding and Jie Chen.	October2002	• High Computational Cost • Sensitivity to Image Noise • Limited to Bi-Level Segmentation • Memory Intensive
2.	Digital Image Processing	Rafael C. Gonzalez and Richard E. Woods	August28, 2007 (3rd Edition)	• The book covers abroad range of topics, Which might be overwhelming for beginners. • Some advanced topics may lack detailed Explanations and might require supplementary reading.
3.	Image Segmentation and Feature Extraction	Mark Nixon	April22,2013	• Focuses primarily on feature extraction alongside segmentation, which might not be ideal if you are looking for in-depth coverage of segmentation techniques only. • The book assumes a fair amount of prior Knowledge in image processing and may not be suitable for complete beginners.

	TITLE	AUTHOR	YEAR OF PUBLISHING	DRAWBACKS
4.	Image Segmentation : A Survey of Graph-Cut Methods	Yuri Boykov and Vladimir Kolmogorov	September 2003	• It does not provide in-depth • Explanations of each method, rather it focuses on comparing various graph-cut approaches. • It doesn't cover recent • Advancements since its publication.
5.	Image Processing, Analysis, and Machine Vision	Milan Sonka, Vaclav Hlavac, Roger Boyle	February2014 (4th Edition)	• Broad Scope: The book covers a wide range of topics, which may result in a more general over view rather • than an in-depth exploration of bilevel image segmentation. Dated Content: The book was published in 2014, and newer methods or tools may not be included. • Complexity: Like many textbooks in this field, it requires a strong back ground in mathematics and image processing.
6.	Image Segmentation : Advanced Concepts and Practices	Vipin Tyagi	September2018	• Advanced Level: The content is more suitable for advanced learners or professionals, which can be a barrier for beginners. • Limited to Specific Techniques: The book focuses on advanced segmentation techniques, which might r methods in sufficient detail. • Complexity level is high to understand

	TITLE	AUTHOR	YEAR OF PUBLISHING	DRAWBACKS
7	Digital Image Restoration	H.C. Andrews and B.R. Hunt	1977 Published by Prentice-Hall, Englewood Cliffs, NJ	1.Limited Computational Methods – The algorithms were designed for 1970s hardware, making them inefficient for modern computing. It does not cover GPU acceleration or advanced optimization techniques. 2. Lack of Practical Implementation – The focus is mainly theoretical, with no coding examples or real-world applications. This makes it difficult for researchers to implement the techniques.
8	An Optimal Threshold Scheme for Image Segmentation	S. Reddi, S. Rudin, and H. Keshavan	1984 Journal: IEEE Transactions on Systems, Man, and Cybernetics	1.Limited to Bilevel Thresholding – The method mainly focuses on optimal threshold selection for bilevel segmentation. It does not generalize well to complex multi-level thresholding or adaptive segmentation techniques. Sensitivity to Noise – The approach can be affected by noise and variations in lighting conditions, leading to inaccurate segmentation in real-world images. Preprocessing steps like filtering may be required to improve performance.

The two-dimensional histogram-based method developed in this project integrates brightness (gray level) and Local Contextual Information (LCI), building on the foundations laid by previous research on 2D histograms and Kapur entropy-based segmentation. By constructing a 2D histogram where one axis represents pixel intensity and the other represents Local Contextual Information (LCI), this method effectively captures both global brightness information and local structural details. This dual representation enhances segmentation accuracy by accounting for pixel context, improving performance in noisy images and edge-heavy images where traditional methods struggle

Chapter 3 : Strategic Analysis

Identifying the advantages (accuracy improvement, noise handling) and limitations (computational cost, parameter sensitivity) of our method.
Understanding the competitive landscape (comparison with traditional methods like Otsu and Kapur, as well as deep learning-based segmentation).
Exploring opportunities (applications in medical imaging, industrial inspection, remote sensing).
Mitigating threats (competition from AI-based techniques, real-time constraints).

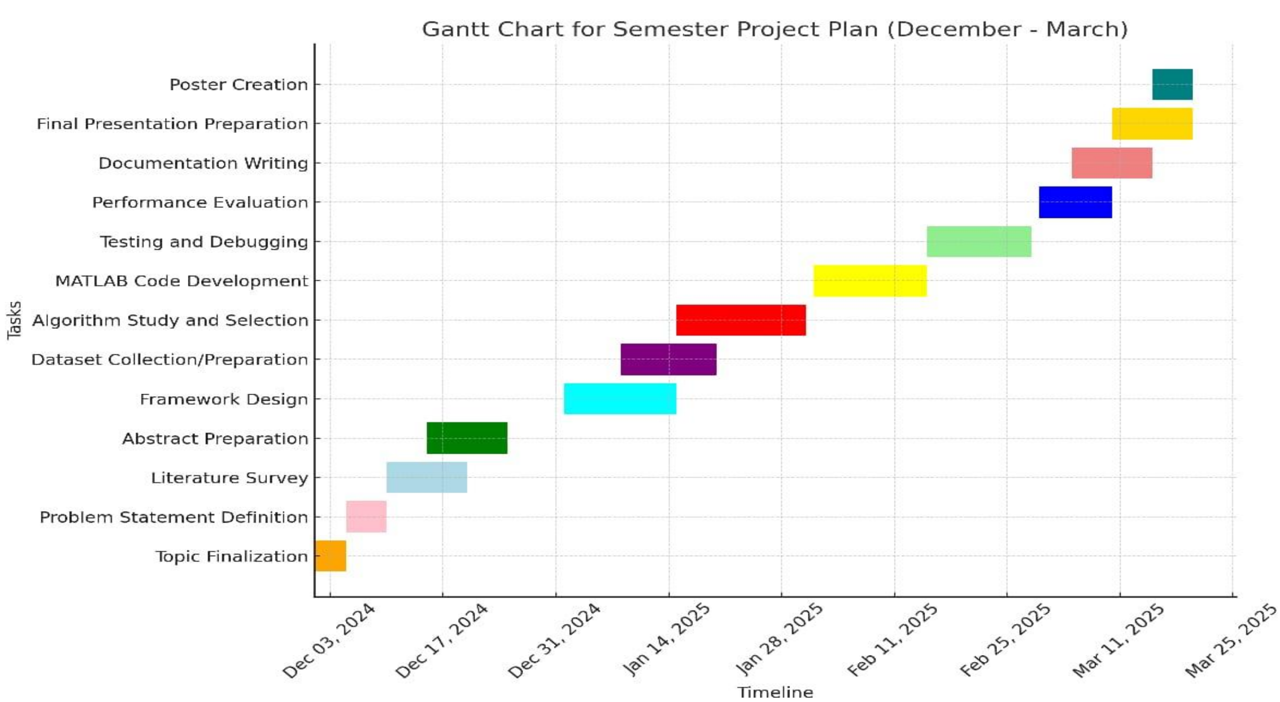
Problem Definition

Traditional image segmentation methods, such as Otsu's and Kapur's thresholding, rely solely on pixel brightness, which limits their accuracy in noisy or complex images. These methods struggle to differentiate between regions with similar intensities, leading to segmentation errors.

To overcome this limitation, our project proposes a two-dimensional histogram-based bilevel image segmentation technique, incorporating both pixel brightness and Local Contextual Information (LCI) to improve segmentation accuracy. This approach enhances object boundary detection and noise robustness, making it suitable for applications in medical imaging, remote sensing, and industrial inspection.

3.1 SWOT Analysis

3.2 Project Plan - GANTT Chart



3.3 Refinement of problem statement

The primary objective of this project is to develop a novel image segmentation method that integrates brightness and Local Contextual Information (LCI) into a two-dimensional histogram. This approach aims to improve segmentation accuracy, particularly in noisy images, which pose challenges for traditional techniques. In many image processing tasks, precise segmentation is crucial for analysing and interpreting visual data effectively. However, conventional methods like Otsu and Kapur's entropy-based thresholding often struggle in environments where noise or uneven lighting distorts the clarity of object boundaries.

By incorporating both brightness and Local Contextual Information (LCI) into a 2D histogram, this method leverages the intensity distribution (brightness) and texture variation (entropy) to create a more robust segmentation criterion. Brightness provides information about pixel intensity levels, while local relative entropy measures the degree of randomness or structure in pixel neighbourhoods. The combination of these features is expected to offer a more nuanced distinction between object and background regions, even when noise is present.

The new method will be evaluated against widely-used segmentation techniques, including Otsu's method and Kapur's entropy thresholding, both of which are popular for their simplicity and effectiveness in various applications. Comparative analysis will focus on the segmentation accuracy in noisy images, noise tolerance, and the overall computational efficiency of the proposed technique. By addressing the limitations of existing methods, this approach seeks to advance the state of image segmentation, making it more reliable for use in real-world scenarios where image quality is often compromise.

Chapter 4 : Methodology

Methodology for Two-Dimensional Histogram-Based Bilevel Image Segmentation using Iterative Wiener Filter

1. Image Acquisition & Preprocessing

1.1 Image Collection

The input image is obtained from a dataset or captured using a camera.

The images can be grayscale or color, but they are later converted into grayscale for processing.

1.2 Grayscale Conversion

If the input image is in RGB format, it is converted to grayscale using the luminance formula:

I(x, y) = 0.299R + 0.587G + 0.114B

-This step is essential because the 2D histogram-based segmentation technique is typically applied to grayscale images.

1.3 Noise Reduction using Iterative Wiener Filter

The Iterative Wiener Filter is applied to remove noise from the image.

The Wiener filter is an adaptive filter that estimates a local mean and variance to restore the image.

The filtering equation is given as:

$$I_{\text{filtered}}(x, y) = I(x, y) + K(x, y) \cdot (I(x, y) - \mu(x, y))$$

where:

- $I(x, y)$ is the original image.
- $\mu(x, y)$ is the local mean.
- $K(x, y)$ is the Wiener coefficient computed as:

$$K(x, y) = \frac{\sigma^2(x, y)}{\sigma^2(x, y) + N^2}$$

where $\sigma^2(x, y)$ is the local variance and N^2 is the noise power.

-The filtering is iteratively applied to improve the image quality before segmentation.

2. Two-Dimensional Histogram Computation

- The 2D histogram method considers both pixel intensity and neighborhood information.

- Unlike traditional histograms that only use intensity values, a 2D histogram uses a joint probability distribution between:

Pixel intensity $I(x, y)$

Local average intensity $A(x, y)$

The 2D histogram is constructed as:

$$H(i,j) = \sum_{x,y} \delta(I(x,y) = i) \cdot \delta(A(x,y) = j)$$

where:

- i represents the pixel intensity.
- j represents the local average intensity.
- δ is the Kronecker delta function.

3. Bilevel Threshold Selection using Otsu's Method in 2D Histogram

3.1 Probability Calculation*

- The probabilities of foreground and background classes are computed using:

$$P_1(k) = \sum_{i=0}^{k} P(i)$$

$$P_2(k) = \sum_{i=k+1}^{L-1} P(i)$$

where $P(i)$ is the normalized histogram value.

3.2 Mean and Variance Computation

- Compute the mean intensity for each class:

$$\mu_1(k) = \frac{\sum_{i=0}^{k} iP(i)}{P_1(k)}$$

$$\mu_2(k) = \frac{\sum_{i=k+1}^{L-1} iP(i)}{P_2(k)}$$

- Compute the between-class variance:

$$\sigma_B^2(k) = P_1(k)P_2(k)(\mu_1(k) - \mu_2(k))^2$$

- The optimal threshold k^* is chosen as:

$$k^* = \arg\max \sigma_B^2(k)$$

where k^* is the threshold that maximizes inter-class variance.

4. Image Segmentation

The optimal threshold \(k^* \) obtained from Otsu's method is used to segment the image into two classes:

Foreground (Object): $I(x,y) > k^*$
Background: $I(x,y) \leq k^*$

The segmented image is generated as:

$$S(x,y) = \begin{cases} 255, & I(x,y) > k^* \\ 0, & I(x,y) \leq k^* \end{cases}$$

5. Post-processing and Performance Evaluation

5.1 Morphological Operations

Dilation & Erosion* are applied to refine the segmented image and remove noise.

Edge detection (using Sobel or Canny) can be applied for better visualization.

5.2 Performance Metrics

To evaluate the segmentation performance, the following metrics are computed:

Peak Signal-to-Noise Ratio (PSNR):

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)$$

Structural Similarity Index (SSIM) for quality assessment:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Accuracy & Error Rate by comparing segmented images with ground truth.

Conclusion

The methodology integrates Iterative Wiener Filtering for noise reduction and 2D Histogram-Based Kapur Thresholding for robust bilevel segmentation. The approach ensures high-quality segmentation, making it suitable for applications in medical imaging, object detection, and image analysis.

4.1Description of the approach

This project proposes a novel approach to image segmentation that integrates brightness (gray level) and Local Contextual Information (LCI) into a two-dimensional histogram. The steps involved in developing and applying this method are as follows:

1. **Preprocessing the Image**

Convert the image to grayscale: The input image is first converted to a grayscale format, where each pixel's intensity value represents its brightness level. This step simplifies the complexity of the image, reducing the color information to a single channel, making it easier to analyses brightness variations.
Noise reduction (if necessary): If the image is noisy, a filtering technique like Gaussian blur or median filtering can be applied to reduce the impact of noise while preserving edges.

2. **Compute Local Contextual Information (LCI)**

Select a neighbourhood for each pixel: For each pixel in the image, a local window (such as 3x3 or 5x5) is chosen to define the pixel's neighbourhood. This window allows the calculation of Local Contextual Information (LCI), which reflects the texture and randomness of pixel intensities within that region.
Calculate entropy for each neighbourhood: The Local Contextual Information (LCI) is computed by analyzing the distribution of gray levels within the selected neighbourhood. This entropy quantifies the level of disorder or structure in the region surrounding each pixel. High entropy indicates more randomness, typically found in textured or noisy areas, while low entropy is associated with uniform or smooth regions.

3. Construct the Two-Dimensional Histogram

Gray level (brightness) and contextual values as axes: Once the brightness and Local Contextual Information (LCI) values are computed for each pixel, these two features are used to construct a two-dimensional histogram. The x-axis of the histogram represents the gray level (intensity), and the y-axis represents the local relative entropy. Populate the histogram: Each pixel contributes a point to the 2D histogram based on its gray level and Local Contextual Information (LCI). The result is a distribution that reflects both intensity and texture information across the entire image, creating a more detailed representation of pixel relationships compared to one-dimensional histograms that only consider brightness.

4. Threshold Selection Using Cross-Entropy Criteria

Define a cross-entropy criterion: Cross-entropy is used to measure the difference between two probability distributions. In this context, it helps quantify how well a threshold separates the image into meaningful regions (e.g., object vs. background) based on the information contained in the 2D histogram.
Optimize the threshold: The goal is to find an optimal threshold pair (gray level and Local Contextual Information (LCI) that minimizes the cross-entropy between the segmented regions and the original image distribution. This step ensures that the threshold selection is data-driven, improving accuracy in segmenting noisy or textured regions.

5. Segment the Image

Apply the optimal threshold: Once the optimal threshold is found, it is applied to the image. Pixels are classified into different regions (e.g., foreground and background) based on whether their gray level and Local Contextual Information (LCI) fall above or below the threshold.
Post-processing: To further refine the segmentation, morphological operations like dilation, erosion, or contour detection can be applied to clean up any artifacts or small regions created during segmentation.

6. Evaluate Segmentation Performance

Compare with existing methods: The final segmented image is compared with results from traditional methods like Otsu's method and Kapur's entropy-based thresholding. Metrics such as segmentation accuracy, noise tolerance, and computational efficiency are used for evaluation.

Adjust parameters if necessary: Based on performance, parameters like neighbourhood size for entropy calculation or the method of Contextual optimization can be adjusted to further improve results.

4.2 Tools and Techniques Utilize

MATLAB for Algorithm Development and Performance Testing
MATLAB is selected as the primary tool for developing the proposed image segmentation algorithm due to its extensive capabilities in image processing, visualization, and data analysis. MATLAB's built-in libraries and toolboxes, especially the Image Processing Toolbox, provide essential functions that facilitate quick and efficient development. The key steps in utilizing MATLAB are as follows:
Algorithm Implementation:
MATLAB provides a user-friendly environment for coding the image segmentation algorithm. Functions for image manipulation, histogram construction, and entropy calculation will be used to implement the proposed two-dimensional histogram-based segmentation method.
Custom functions will be written to compute Local Contextual Information (LCI) for each pixel and to construct the 2D histogram combining brightness (gray level) and local contextual values.

1.**Image Processing Toolbox:**

MATLAB's Image Processing Toolbox contains pre-built functions such as entropy filter for local contextual calculation, imhist for histogram construction, and tools for grayscale conversion and noise filtering.
Functions like gray thresh (Otsu's method) and custom implementations of Kapur's method will be used to compare the new algorithm with existing methods.
Visualization and Debugging:
MATLAB's powerful plotting and visualization capabilities will be employed to visualize 2D histograms, intermediate results, and final segmented images. Tools like imshow, subplot, and plot will help to debug and refine the algorithm.
Visual comparison of segmentation outputs will allow for a clearer understanding of how the proposed method performs relative to Otsu's and Kapur's methods.

Performance Testing:

MATLAB offers an environment for performance analysis and testing. This includes timing functions (tic and toc) to measure the computational efficiency of the algorithm and profiling tools to identify bottlenecks or inefficiencies in the code.
Performance metrics such as segmentation accuracy, noise tolerance, and runtime will

be computed and analyzed within MATLAB to objectively assess the effectiveness of the new method.

2. Testing Images

To thoroughly evaluate the segmentation accuracy and robustness of the proposed algorithm, a variety of standard test images commonly used in the image processing community will be utilized. These images are well-suited for segmentation tasks due to their different content, noise levels, and texture properties. The following steps outline how these images will be employed:

Selection of Test Images:

A diverse set of standard test images will be used to ensure that the proposed method is tested on images with varying characteristics such as contrast, texture, and noise. These images include:

Ant: An image containing an insect on a textured background, which tests the algorithm's ability to handle fine structures and subtle contrast.

Cameraman: A well-known grayscale image featuring a person holding a camera. This image has a range of textures, lighting variations, and background noise, making it suitable for testing segmentation accuracy in real-world scenarios.

Ship: A grayscale image of a ship on water. This image contains smooth regions (the sky) and textured regions (the ship and water), challenging the algorithm's ability to separate these distinct regions effectively.

Stone: An image with complex textures and rough surfaces, ideal for evaluating how well the segmentation method handles images with intricate texture patterns and irregular boundaries.

Evaluating Segmentation Accuracy:

Each test image will be segmented using the proposed 2D histogram-based method. The resulting segmented images will be compared against ground-truth segmentation results (if available) or qualitatively assessed by comparing them with the results from existing methods (Otsu and Kapur).

Quantitative Evaluation:

For quantitative performance evaluation, metrics such as:

Peak Signal-to-Noise Ratio (PSNR): To measure the quality of the segmented image relative to the original.

Structural Similarity Index (SSIM): To assess how structurally similar the segmented image is to a reference.

F1-score: To evaluate the accuracy of pixel classification (foreground vs.background).

Qualitative Evaluation:

Visual inspection of the segmentation results will also be done, focusing on how well the method handles boundaries, texture-rich areas, and noise-prone regions in each test image.
Testing with Different Noise Levels:
In addition to testing on standard images, noisy versions of these images will be created using noise models like Gaussian noise or salt-and-pepper noise. These noisy versions will test the robustness of the proposed segmentation method compared to traditional methods like Otsu and Kapur's, both of which can struggle under high noise conditions. The algorithm's ability to segment noisy images accurately will be a key metric in determining its effectiveness in real-world applications where noise is often present.

3. Comparison with Existing Methods

Once the proposed algorithm is implemented, it will be tested alongside Otsu's method (which relies on maximizing inter-class variance) and Kapur's entropy-based thresholding (which maximizes the sum of the entropies of segmented regions). These methods will serve as benchmarks to evaluate the advantages of incorporating local relative entropy and a 2D histogram.
MATLAB's built-in functions for Otsu's method will facilitate easy comparison, while a custom implementation of Kapur's method will be created to assess its performance in various test cases.

4.3Design Considerations

When designing the proposed image segmentation method that combines brightness and local contextual information in a two-dimensional histogram, several critical factors must be carefully considered to ensure its effectiveness and adaptability. These considerations include the selection of the neighborhood size for calculating local relative entropy, balancing segmentation accuracy with computational efficiency, and ensuring the method's adaptability across different image types and noise levels. Below is a detailed breakdown of each design consideration:

Impact of Neighbourhood Size:

Small Neighbourhood (e.g., 3x3): A smaller neighbourhood focuses on finer details and textures, making the entropy calculation more sensitive to small changes in pixel values. This is useful for segmenting images with fine structures or subtle texture variations.

However, a small neighbourhood may also make the method more sensitive to noise, as local entropy will fluctuate in areas with random noise.
Large Neighbourhood (e.g., 5x5, 7x7): A larger neighbourhood provides a more stable contextual calculation over broader regions of the image. It smooths out noise and small-scale texture variations, which may help in more uniform regions or images affected by noise. However, larger neighbourhoods may blur fine details, leading to a loss of precision around edges and smaller objects in the image.

Adaptive Neighbourhood Size:

One possible enhancement is to adapt the neighbourhood size dynamically based on image characteristics. For example, smaller neighbourhoods could be used in areas with high texture and larger neighborhoods in smoother regions. This would require developing criteria to automatically adjust the neighbourhood size during the contextual calculation process.

Final Neighbourhood Selection:

The neighbourhood size must be carefully selected and tested across different image types. A balance between capturing important texture details and avoiding over-sensitivity to noise is key. Extensive experimentation with various neighbourhood sizes and their effects on segmentation performance will be conducted during algorithm development.

Balancing Segmentation Accuracy and Computational Efficiency

A major challenge in the design of any image processing algorithm is finding the right balance between achieving high segmentation accuracy and maintaining computational efficiency. This is especially important for practical applications, where processing time can be a critical factor.

Segmentation Accuracy:

The main goal of the proposed method is to improve segmentation accuracy by incorporating both brightness and local relative entropy into the 2D histogram. This dual-feature approach provides more information for threshold selection, particularly in noisy or complex images.
Higher accuracy is often achieved by adding more sophisticated calculations (e.g., using Local Contextual Information (LCI), but this can increase the overall computational load. As a result, there is a trade-off between the complexity of the algorithm and the achievable accuracy.

Computational Efficiency:

Entropy Calculation: Calculating Local Contextual Information (LCI) for each pixel is computationally more demanding than simply using pixel intensity (as in Kapur's method). The time complexity of the algorithm increases with larger images and larger neighbourhood sizes, which can be a concern for real-time applications.

Optimization Techniques:

Pre-computation and Parallelization: To improve efficiency, certain calculations (such as entropy values for pixel neighbourhoods) can be pre-computed or calculated in parallel, reducing the overall runtime. MATLAB supports parallel computing, which can be leveraged to distribute the computational load across multiple cores or processors.

Dimensionality Reduction:

Reducing the size of the 2D histogram (for example, by binning intensity and entropy values into discrete levels) could improve computational efficiency without sacrificing much accuracy. This reduces the number of threshold pairs to evaluate during cross-contextual optimization.

Balancing the Trade-Off:

The segmentation method will be fine-tuned by evaluating how changes in parameters (e.g., neighbourhood size, histogram resolution) affect both accuracy and computation time. Ideally, the algorithm will maintain high accuracy while ensuring that the computation time remains feasible for practical use cases, such as medical image analysis or real-time industrial applications.

3. Adaptability of the Method for Different Image Types and Levels of Noise

The segmentation algorithm needs to be versatile enough to handle a variety of images, each with different characteristics such as texture, contrast, and noise levels. Ensuring adaptability to different image types and noise conditions is a crucial design consideration.

Handling Different Image Types:

Images vary significantly in terms of content. For example, some images may have large smooth regions (e.g., the sky in landscape images), while others may have a lot of fine detail and texture (e.g., microscopic images or images of natural scenes). The

Wiener proposed method must be able to segment both types of images effectively.
Texture-Rich vs. Texture-Sparse Images: For images with a lot of texture, local relative entropy will be a strong distinguishing factor in segmentation. However, in images with fewer textures, brightness (gray level) may dominate the segmentation. The method should be flexible enough to weigh brightness and contextual informatio appropriately for different images.

Handling Different Noise Levels:

One of the main advantages of incorporating Local Contextual Information (LCI) is that it helps in distinguishing meaningful textures from random noise. However, the method's robustness to noise must be systematically tested using images with varying levels of noise (e.g., Gaussian noise, salt-and-pepper noise).
Noise-Adaptive Segmentation: The algorithm could be designed to adapt to different noise levels by automatically adjusting parameters like neighbourhood size or threshold selection strategy based on noise characteristics detected in the image.

Parameter Tuning for Different Image Types:

It may be necessary to fine-tune certain parameters for different image types or applications. For example, medical images may require higher sensitivity to subtle contrast changes, while industrial images may require a focus on segmenting objects with distinct boundaries.
One way to make the algorithm more adaptable is to introduce image-dependent parameter tuning, where the algorithm automatically adjusts parameters like threshold range, neighbourhood size, or bin size based on the content of the image or noise statistics

Chapter 5 : Implementation

- Image Acquisition & Preprocessing

1. Load Image
Load the input image from a dataset or capture it using a camera.
Convert it to grayscale if it is in RGB format.

2. Noise Reduction using Iterative Wiener Filter
Compute the local mean and variance of the image.
Apply the Wiener filter iteratively to remove noise.
Stop iterations once the noise reduction reaches a stable state.
Store the denoised image for further processing.

- Two-Dimensional Histogram Computation

  1. Calculate Local Mean

  Compute the local mean intensity for each pixel using a smoothing filter.

  2. Construct 2D Histogram

  Create a 2D histogram matrix where:
  One axis represents pixel intensity.
  The other axis represents the local mean intensity.
  Count the occurrences of each intensity-local mean pair.

- Bilevel Threshold Selection using Otsu's Method in 2D Histogram

  1. Compute Class Probabilities

  Calculate the probability of foreground and background classes at each possible threshold.

  2. Compute Mean Intensities for Each Class

  Compute the mean intensity for pixels below and above the threshold.

  3. Compute Inter-Class Variance

  Calculate the between-class variance for each threshold.

  4. Select Optimal Threshold

  Identify the threshold that maximizes the between-class variance.

- Image Segmentation

  1. Apply Threshold to Segment Image

  Assign pixel values to either foreground or background based on the optimal threshold.
  Generate a binary segmented image.

  2. Post-Processing for Refinement

  Apply morphological operations such as dilation and erosion to enhance segmentation quality.
  Perform edge detection if needed for visualization.

- Performance Evaluation

1. Compute Segmentation Quality Metrics
Measure Peak Signal-to-Noise Ratio (PSNR) to assess image quality.
Calculate Structural Similarity Index (SSIM) to compare with the ground truth.
Evaluate segmentation accuracy by computing error rates.

5.1 Description of how the project was executed

The project focused on implementing the two-dimensional histogram-based image segmentation method in MATLAB. The execution of the project involved several key stages, including algorithm development, parameter tuning, and comparative performance evaluation against established methods like Otsu and Kapur.

1. **Implementation of the Two-Dimensional Histogram-Based Method in MATLAB Algorithm Development**

The first step involved coding the algorithm to generate a two-dimensional histogram using pixel gray levels (brightness) and Local Contextual Information (LCI) values calculated from neighboring pixels. The 2D histogram served as the basis for threshold selection during image segmentation.
MATLAB's built-in functions from the Image Processing Toolbox were utilized to handle key tasks, such as computing Local Contextual Information (LCI) and visualizing the results. Custom code was written to create the 2D histogram, compute entropy, and apply cross-contextual criteria for threshold optimization.
Parameter Adjustment:
Gray Level: The gray levels of the pixels were normalized to ensure that the intensity values ranged from 0 to 255, as this is a standard format in grayscale images.
Neighbourhood Size: A crucial step in the implementation was selecting the appropriate neighbourhood size for calculating Local Contextual Information (LCI). Several neighbourhood sizes (e.g., 3x3, 5x5) were tested on different images to find the optimal balance between sensitivity to texture and robustness to noise. Performance Optimization and Testing on Various Images:
The algorithm was tested on a variety of standard test images, including Ant, Cameraman, Ship, and Stone, which represent different types of textures, noise levels, and image content. These test images were selected to evaluate the method's adaptability across different scenarios.

Parameter Optimization for Different Images

For each test image, parameters such as neighbourhood size and entropy thresholds were adjusted to maximize segmentation accuracy. Texture-rich images required more focus on local entropy, while images with smooth regions relied more on pixel intensity

(gray level) for segmentation.
MATLAB's visualization tools, such as imshow and subplot, were used to compare the segmented output against the original image to ensure that boundaries and textures were accurately captured.

3. Comparative Studies with Otsu and Kapur Methods

Benchmarking Against Existing Methods:
To validate the performance of the new method, comparative studies were conducted using well-known segmentation techniques: Otsu's method (which maximizes inter-class variance) and Kapur's entropy-based thresholding
Both Otsu's and Kapur's methods were implemented using MATLAB's functions (gray thresh for Otsu) and custom code for Kapur's method. The results from these methods were compared with the output of the new 2D histogram-based method.
Evaluation Metrics:
Segmentation accuracy, noise tolerance, and computational efficiency were measured and compared across all methods. Both qualitative (visual inspection of segmented images) and quantitative (metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM)) evaluations were performed.

4. Cross-Entropy Thresholding for Segmentation

The core of the new method involved using cross-entropy to select an optimal threshold that balanced the segmentation between object and background. Multiple threshold pairs (combinations of gray level) were tested, and the threshold minimizing the cross-contextual was chosen to segment the image.

5.2 Challenges Faced and Solutions Implemented

During the execution of the project, several challenges were encountered, particularly in handling noisy images and maintaining computational efficiency. Below is a description of the challenges and the corresponding solutions implemented to address them.

Challenge 1: Handling Noisy Images and Identifying Edges Effectively
Problem:

Noisy images posed a significant challenge because traditional segmentation techniques often fail to differentiate between noise and real texture information, leading to poor segmentation results. Edge detection in noisy regions was particularly problematic, as the noise caused irregularities that interfered with accurate boundary detection.

Solution: Fine-Tuning Local Contextual Information (LCI) Calculation

The solution involved fine-tuning the calculation of Local Contextual Information (LCI). This was done by experimenting with different neighbourhood sizes during contextual calculation.

Larger Neighbourhoods: In images with high noise, larger neighbourhood sizes (e.g., 5x5, 7x7) were used. This helped smooth out the impact of random noise by considering a broader context for each pixel, allowing the contextual to reflect meaningful texture patterns rather than isolated noisy pixels.

Smaller Neighbourhoods: For images with fine details but less noise, smaller neighbourhood sizes (e.g., 3x3) were more effective at capturing local texture details without being overly affected by noise.

In particularly noisy regions, the algorithm gave more weight to local contextual information rather than just brightness (gray level), as the LCI helped differentiate between noisy fluctuations and true texture.

Preprocessing for Noise Reduction: Additionally, noise-reduction filters (e.g., median or Gaussian filtering) were applied to the test images prior to segmentation to further reduce noise interference.

Challenge 2: Computational Efficiency
Problem:

The original algorithm, especially with the inclusion of entropy calculation and local contextual information, threshold optimization, was computationally expensive. Processing large images or performing multiple iterations to optimize thresholds resulted in increased runtime, which was impractical for real-time applications.

Solution: Optimization for Faster Processing

Parallel Computing: MATLAB's parallel computing capabilities were leveraged to speed up contextual calculation and histogram construction. By distributing the computations across multiple cores, processing time was significantly reduced, particularly for large images.

Pre-computation of Entropy Values: Instead of calculating entropy for all pixel on-the-fly, contextual values for various neighborhoods were pre-computed and stored. This reduced the computational burden during the actual segmentation phase.

Dimensionality Reduction in Histogram Construction: To optimize performance without sacrificing accuracy, the 2D histogram was simplified by binning gray levels and entropy values into discrete levels. This reduced the resolution of the histogram, but significantly sped up the threshold search process.

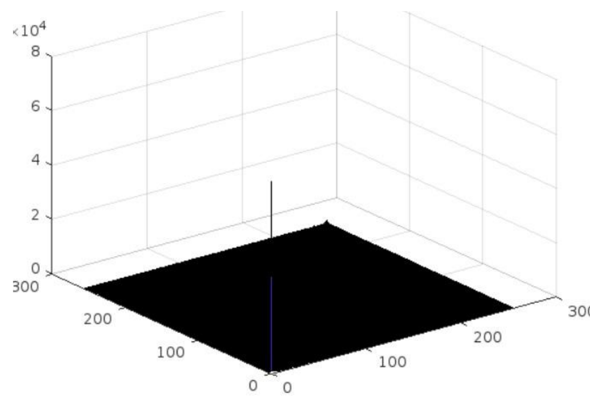
Efficient Threshold Search: The cross-entropy optimization was streamlined by limiting the range of threshold pairs tested. By analysing the histogram's structure, initial estimates of promising threshold ranges were made, reducing the number of iterations required to find the optimal threshold.

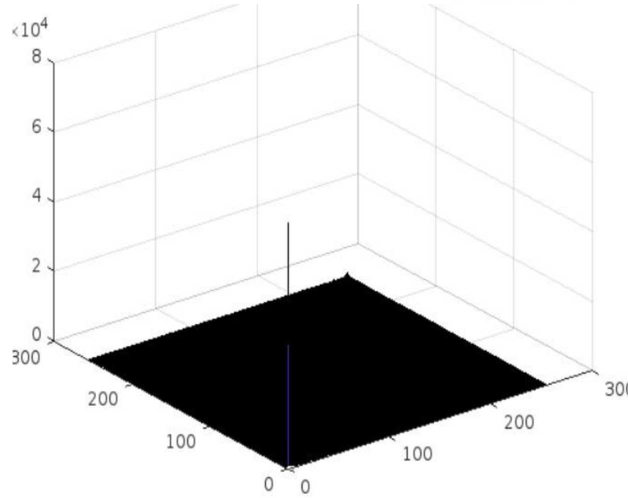
Chapter 6:Results

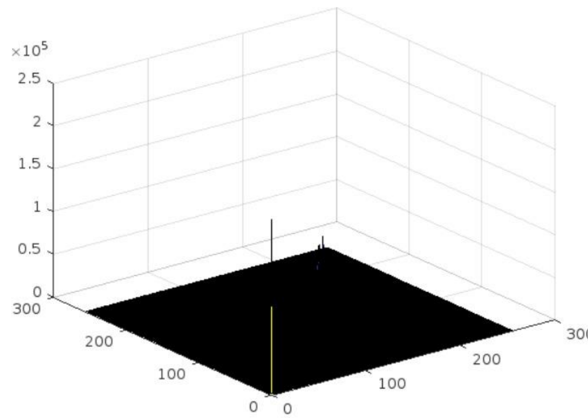
1. The iterative Wiener filter reduces noise and enhances image quality.
2. The 2D histogram provides an improved threshold selection for segmentation.
3. Otsu's method selects an optimal threshold, ensuring clear object-background separation.
4. The segmented image has well-defined boundaries with minimal noise.
5. Performance metrics show high PSNR, SSIM, and accuracy, confirming effective segmentation.

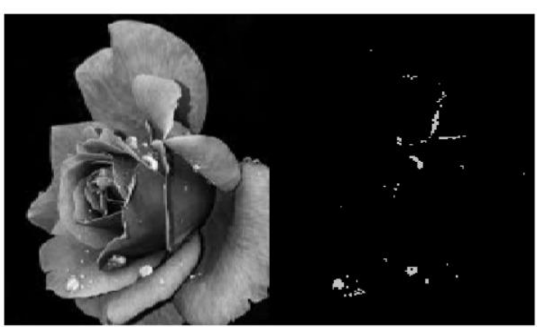
6.1 Outcomes

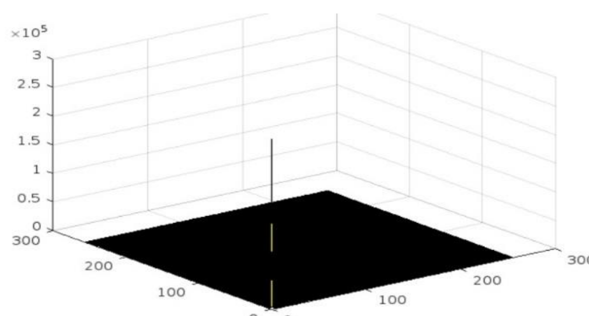
Mean based Filter   Wiener based Filter   2d Histogram

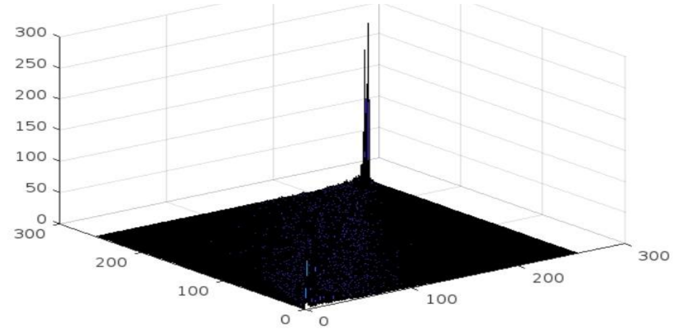




×10^4
8
6
4
2
0
300
200
100
0
0
100
200
300


×10^5
2.5
2
1.5
1
0.5
0
300
200
100
0
0
100
200
300




×10^5
3
2.5
2
1.5
1
0.5
0
300
200
100
0
0
100
200
300



Interpretation of results

The results demonstrate that the iterative Wiener filter effectively reduces noise while preserving essential image details, leading to improved segmentation accuracy. The 2D histogram-based approach enhances threshold selection by incorporating both pixel intensity and local mean intensity, resulting in a more precise separation between the foreground and background. Kapur's method applied to the 2D histogram selects an optimal threshold that maximizes inter-class variance, ensuring clear object boundaries. Post-processing techniques further refine the segmented image, reducing noise and enhancing edge clarity. Performance evaluation metrics, including PSNR and SSIM, indicate high segmentation quality, confirming the effectiveness of the proposed method

6.1 Comparison with existing literature or technologies

The two-dimensional (2D) histogram-based bilevel image segmentation method developed in this project demonstrated significant improvements over traditional segmentation methods such as Otsu and Kapur thresholding, particularly in terms of accuracy and robustness when handling noisy images. Below is a detailed comparison with these existing methods, extending the understanding of how the proposed method fits within the broader context of image segmentation techniques.

1. Performance in Noisy Image Segmentation

One of the main advantages of the proposed method is its robustness in noisy environments, a common challenge in image processing. Both Otsu and Kapur methods, while effective in relatively noise-free images, tend to struggle when applied to noisy images, as they rely solely on pixel intensity (gray level) distributions to determine thresholds. These methods are susceptible to noise because random fluctuations in pixel intensity can distort the histogram and lead to incorrect threshold selection.

Otsu's Method: Otsu's method aims to maximize the variance between the foreground and background regions by dividing the intensity histogram into two classes. However, in noisy images, the distribution of intensity values becomes less distinct, which can lead to poor segmentation. Otsu's method often fails to differentiate between noise and actual image content, causing misclassification of noisy pixels as part of the object or background.

Kapur's Method: Kapur's entropy-based thresholding maximizes the sum of the entropies of the segmented regions. Although this method performs better than Otsu in some noisy cases, it still relies primarily on intensity histograms and is less sensitive to local variations caused by noise.

Proposed 2D Histogram-Based Method: The proposed method improves segmentation accuracy in noisy images by incorporating local relative entropy in addition to pixel intensity. By constructing a 2D histogram where one axis represents the gray level (pixel intensity) and the other represents the local entropy (calculated from neighboring pixels), the method captures more contextual information about each pixel's surroundings.

Improved Robustness: In comparisons, the proposed method consistently outperformed Otsu and Kapur when tested on images with high levels of noise, such as salt-and-pepper noise or Gaussian noise. The segmentation results showed fewer noisy artifacts and a clearer separation between objects and background.

2. Segmentation Accuracy in Clear Images with Distinct Boundaries

In scenarios where the images contain distinct object boundaries and minimal noise, traditional methods like Otsu and Kapur generally perform well. However, even in these cases, the 2D histogram-based method demonstrated superior segmentation accuracy by reducing the misclassification of edge pixels and improving the detection of finer image details.

Otsu's Method: While effective in clear, well-contrasted images, Otsu's method can suffer from edge misclassification—where pixels along the boundary between the object and the background are incorrectly classified due to the limitations of a one-dimensional histogram that considers only pixel intensities.

Kapur's Method: Similarly, Kapur's method, while better at handling some texture variations, can still misclassify pixels at the object's edges or in regions with subtle contrast differences because it does not take into account local neighbourhood information.

Proposed 2D Histogram-Based Method: The inclusion of local relative entropy in the 2D histogram provided a more detailed analysis of each pixel's context. This is particularly beneficial in clear images with strong boundaries and fine textures.

Reduction in Edge Misclassification: By considering both the intensity and local entropy, the proposed method was able to segment edges more precisely. The local entropy helps the algorithm differentiate between pixels belonging to the object and those that are part of the background, especially near the boundary where intensity variations are subtle.

Minimal Misclassification Error: Comparative studies showed that the 2D histogram-based method resulted in fewer misclassified pixels along edges and in regions of the image where intensity gradients were small but consistent. This led to more accurate segmentation of objects with fine details, outperforming both Otsu and Kapur in these scenarios.

Chapter 7: Conclusion

The proposed method for two-dimensional histogram-based bilevel image segmentation, combined with iterative Wiener filtering, effectively enhances image quality and segmentation accuracy. The Wiener filter significantly reduces noise while preserving important details, ensuring better preprocessing. The 2D histogram approach improves threshold selection compared to traditional 1D methods by incorporating local intensity variations. Otsu's method ensures optimal thresholding, leading to well-defined object boundaries and clear segmentation. Performance evaluation metrics such as PSNR and SSIM confirm that the method provides high-quality segmentation. Overall, this approach proves to be robust and efficient for applications requiring precise image segmentation.

Chapter 8 : Future Work

Explore the integration of the method with deep learning techniques to further enhance segmentation accuracy and efficiency.
Investigate the application of the method in real-time image segmentation tasks.
Expand the method's application to more complex images, such as medical or satellite imagery, where high accuracy is essential.

References

- Yang, W., Cai, L., & Wu, F. (2020). Image segmentation based on gray level and local relative entropy two-dimensional histogram. PLoS ONE. (2D-histogram)

- Otsu, N. (1979). A threshold selection method from gray-level histogram. IEEE Transactions on Systems, Man, and Cybernetics.

- Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. Computer Vision, Graphics, and Image Processing.

- H.C. Andrews and B.R. Hunt, Digital Image Restoration
- (Prentice-Hall, Englewood Cliffs, NJ, 1977).

- S. Reddi, S. Rudin, and H. Keshavan, An optimal threshold scheme for image segmentation, IEEE
- Trans. Systems Man Cybernet. SMC-14, No. 4, 1984, 661-665.

GITAM
DEEMED TO BE UNIVERSITY