**Two Dimensional Histogram Based Bi-level Image Segmentation**

**Submitted by**

**SUCHITHRA C BU21EECE0100426**

**CHARSIMA S BU21EECE0100356**

**SIREESHA P BU21EECE0100210**

**Under the Guidance of**

**Dr. Pankaj Kandhway**

**Assistant Professor**

**[01/06/24 to 30/11/24]**

****

**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.**



**Department of Electrical, Electronics and Communication Engineering**

**GITAM School of Technology**

**DECLARATION**

**I/We declare that the project work contained in this report is original and that I did it under the guidance of my project guide.**

**Date:**

**Registration No. Student Name Signature**

**BU21EECE0100426 SUCHITHRA.C**

**BU21EECE0100356 CHARISMA.S**

**BU21EECE0100210 P. SIREESHA**

**Department of Electrical, Electronics and Communication Engineering**

**GITAM School of Technology, Bengaluru-561203**

****

**CERTIFICATE**

This is to certify that **SUCHITHRA C bearing BU21EECE0100426, CHARISMA S bearing BU21EECE0100356, SIREESHA P bearing BU21EECE0100210** has satisfactorily completed Mini Project Entitled in partial fulfillment of the requirements as prescribed by University for VIIth 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 1**](#_heading=h.gjdgxs)

[1.1 Overview of the Problem Statement 1](#_heading=h.30j0zll)

[1.2 Objectives and Goals 1](#_heading=h.1fob9te)

[**Chapter 2: Literature Review 2**](#_heading=h.3znysh7)

[**Chapter 3: Strategic Analysis and Problem Definition 3**](#_heading=h.2et92p0)

[3.1 SWOT Analysis 3](#_heading=h.tyjcwt)

[3.2 Refinement of Problem Statement 3](#_heading=h.2s8eyo1)

[**Chapter 4: Methodology 4**](#_heading=h.17dp8vu)

[4.1 Description of the approach 4](#_heading=h.3rdcrjn)

[4.2 Tools and Techniques Utilized 4](#_heading=h.26in1rg)

[4.3 Design Conside 4](#_heading=h.lnxbz9)

[**Chapter 5: Implementation 5**](#_heading=h.1ksv4uv)

[5.1 Description of how the project was executed 5](#_heading=h.44sinio)

[5.2 Challenges faced and solutions implemented 5](#_heading=h.2jxsxqh)

[**Chapter 6:Results 6**](#_heading=h.z337ya)

[6.1 outcomes 6](#_heading=h.3j2qqm3)

[6.2 Interpretation of Results 6](#_heading=h.1y810tw)

[6.3 Comparison with existing literature or technologies 6](#_heading=h.2xcytpi)

[**Chapter 7: Conclusion 7**](#_heading=h.1ci93xb)

[**Chapter 8: Future Work 8**](#_heading=h.2bn6wsx)

[Here write Suggestions for further research or development Potential improvements or extensions 8](#_heading=h.qsh70q)

[**References 9**](#_heading=h.1pxezwc)

# Chapter 1: Introduction

## **Overview 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 (LRE) of its neighboring pixels. By incorporating the LRE, 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 and edge-detection scenarios.

# **Chapter 2: Literature Review**

Traditional thresholding methods, such as Otsu and Kapur, rely solely on brightness (gray level) information from the image to segment foreground from background. While these methods are effective for simple and noise-free images, they often result in poor segmentation when images are noisy or contain intricate textures and edges. The primary limitation of these approaches is their reliance on a one-dimensional histogram, which summarizes the distribution of pixel intensities but ignores valuable spatial and contextual information from neighbouring pixels. This can lead to misclassification in noisy regions or where pixel intensity alone does not provide enough distinction between object and background.

Traditional Methods:

* Otsu’s Method: This technique seeks to maximize the inter-class variance between two classes (foreground and background). It performs well in simple, noise-free images but is prone to inaccuracies in noisy environments where the intensity histogram becomes distorted.
* Kapur’s Entropy-Based Thresholding: Kapur’s method uses an entropy maximization approach, where the goal is to find a threshold that maximizes the entropy of the two classes. While it performs slightly better in some noisy cases compared to Otsu, it still suffers from poor segmentation results when noise significantly alters the intensity distribution.

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 relative entropy.

Research Supporting 2D Histograms:

* Liao et al. (2001): In their work on 2D Otsu thresholding, Liao and colleagues introduced a method that incorporates spatial correlation between pixels into the histogram. They demonstrated that this two-dimensional approach leads to more accurate segmentation, particularly in images with texture and fine details. Their work showed that combining intensity with spatial or gradient information results in a more robust segmentation process compared to traditional 1D methods.
* Zhang and Hu (2008): This study focused on gradient-based 2D histograms for image segmentation, highlighting that including gradient information (edge magnitude) improves the identification of object boundaries and reduces misclassification in complex scenes.

Local Relative Entropy:

Building on this body of research, the concept of local relative entropy has emerged as a promising addition to 2D histograms for image segmentation. Entropy, in the context of image processing, measures the amount of information or randomness in a pixel's neighbourhood, helping differentiate between smooth regions and areas with texture or noise.

* Local Relative Entropy: Unlike traditional entropy, which measures global randomness in an image, local relative entropy focuses on the relationship between a pixel and its neighbouring pixels. It captures the brightness differences and structural variations within local regions of the image. This local entropy helps in distinguishing between object boundaries, textures, and noise more effectively than intensity-based methods.

The inclusion of local relative entropy in a two-dimensional histogram provides additional contextual information, making it possible to perform more accurate segmentation even in noisy images or those with complex textures and edges. This is particularly important in images where object boundaries are not clearly defined by intensity alone.

Research on Entropy-Based Segmentation:

* Jian Yu et al. (2011): They proposed a 2D histogram-based thresholding method that utilizes spatial information to improve segmentation. Their research demonstrated that combining intensity with local neighbourhood features leads to more robust and accurate results.
* Yin et al. (2016): This research introduced a method that uses local entropy to handle noise and preserve edges in medical images. They showed that entropy-based methods are more resilient to noise compared to traditional intensity-based approaches.

The Developed Method:

The two-dimensional histogram-based method developed in this project integrates brightness (gray level) and local relative entropy, building on the foundations laid by previous research on 2D histograms and entropy-based segmentation. By constructing a 2D histogram where one axis represents pixel intensity and the other represents local relative entropy, 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

**Existing Implementations – Products| Opensource| GitHub etc**

[**https://github.com/SuchithraChalapathii/Finalyear\_project/blob/main/2D%20Poster\_presentation%20.pptx**](https://github.com/SuchithraChalapathii/Finalyear_project/blob/main/2D%20Poster_presentation%20.pptx)

[**https://github.com/SuchithraChalapathii/Finalyear\_project/blob/main/2D%20project%20video.mp4**](https://github.com/SuchithraChalapathii/Finalyear_project/blob/main/2D%20project%20video.mp4)

[**https://github.com/SuchithraChalapathii/Finalyear\_project/blob/main/2D%20project\_ppt.pptx**](https://github.com/SuchithraChalapathii/Finalyear_project/blob/main/2D%20project_ppt.pptx)

**https://github.com/SuchithraChalapathii/Finalyear\_project/blob/main/2DREPORT\_FINAL.docx**

# **Chapter 3: Strategic Analysis and Problem Definition**

## **3.1 SWOT Analysis**

**Strengths:**

* Accounts for both brightness and contextual pixel information.
* Reduces errors in noisy or complex images.
* Efficient for edge detection.

**Weaknesses:**

* May be computationally intensive compared to simpler methods.
* Requires careful parameter tuning for optimal performance.

**Opportunities:**

* Application in medical imaging, plant disease detection, and crack detection.
* Could integrate with other advanced computer vision techniques.

**Threats:**

* Competition from deep learning-based segmentation methods.
* Complexity may limit usage in real-time applications.

### 

# 

# **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 relative entropy 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 relative entropy 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 compromised.

**Chapter 4: Methodology**

**4.1 Description of the Approach**

This project proposes a novel approach to image segmentation that integrates brightness (gray level) and local relative entropy 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 analyse 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 Relative Entropy

* 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 relative entropy, which reflects the texture and randomness of pixel intensities within that region.
* Calculate entropy for each neighbourhood: The local relative entropy is computed by analysing 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 entropy values as axes: Once the brightness and local relative entropy 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 relative entropy. 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 relative entropy) 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.
* Iterate through possible thresholds: Different threshold pairs are tested, and the cross-entropy for each is calculated. The pair that minimizes cross-entropy is selected as the optimal threshold for segmentation.

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 relative entropy 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 cross-entropy 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 relative entropy for each pixel and to construct the 2D histogram combining brightness (gray level) and local entropy values.
  + MATLAB’s optimization tools will be leveraged to apply the cross-entropy criteria and find the optimal segmentation threshold.
* Image Processing Toolbox:
  + MATLAB’s Image Processing Toolbox contains pre-built functions such as entropyfilt for local entropy calculation, imhist for histogram construction, and tools for grayscale conversion and noise filtering.
  + Functions like graythresh (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 analysed 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.

**4.3 Design Considerations**

When designing the proposed image segmentation method that combines brightness and local relative entropy 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 neighbourhood 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:

1. Selection of Neighbourhood Size for Calculating Local Relative Entropy

The choice of the neighbourhood size used to calculate local relative entropy for each pixel is an important factor that influences both the accuracy of the segmentation and the method's sensitivity to local textures.

* 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 entropy 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 (high entropy) and larger neighbourhoods in smoother regions. This would require developing criteria to automatically adjust the neighbourhood size during the entropy 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.

2. 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 entropy), 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 relative entropy for each pixel is computationally more demanding than simply using pixel intensity (as in Otsu’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 of Cross-Entropy Threshold: Cross-entropy-based threshold selection requires iterative testing of multiple threshold pairs to find the optimal segmentation point. While effective, this can be computationally intensive if not optimized.
* 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-entropy 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 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 entropy appropriately for different images.
* Handling Different Noise Levels:
  + One of the main advantages of incorporating local relative entropy 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. For instance, if high noise levels are detected, the algorithm may prioritize local entropy to reduce the impact of intensity fluctuations caused by noise.
* 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**

## **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 relative entropy 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 relative entropy and visualizing the results. Custom code was written to create the 2D histogram, compute entropy, and apply cross-entropy 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 relative entropy. 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.
  + Entropy Calculation: For each pixel, local relative entropy was computed using a selected neighbourhood size. MATLAB’s entropyfilt function helped in calculating the local entropy for each neighbourhood, which was then combined with the pixel’s gray level to form the 2D histogram.

2. 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 (which maximizes entropy of segmented regions).
  + Both Otsu’s and Kapur’s methods were implemented using MATLAB’s functions (graythresh 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 and entropy) were tested, and the threshold minimizing the cross-entropy 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 Relative Entropy Calculation
  + The solution involved fine-tuning the calculation of local relative entropy. This was done by experimenting with different neighbourhood sizes during entropy calculation. For instance:
    - 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 entropy 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 entropy rather than just brightness (gray level), as the entropy 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 cross-entropy 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 entropy 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 every pixel on-the-fly, entropy 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**

**6.1 Outcomes**

The proposed method achieved more accurate segmentation results compared to traditional methods, especially in noisy images and at object boundaries.

INPUT:



GRAYSCALE IMAGE:



GROUNDTRUTH IMAGE:



OUTPUT:



**6.2 Interpretation of Results**

By considering both brightness and local relative entropy, the segmentation accuracy improved significantly, particularly in edge detection and noise handling.

INPUT:



GRAYSCALE IMAGE:



GROUNDTRUTH IMAGE:



OUTPUT:



**6.3 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.
  + Local Entropy Helps to Identify Textures: In noisy regions, local entropy tends to highlight areas where pixel values fluctuate randomly. By factoring in this local entropy, the algorithm can better distinguish noise from meaningful textures or edges. As a result, noise in the intensity histogram is not misinterpreted as significant image content.
  + 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.

3. Handling of Texture-Rich Images

Texture is another aspect where the 2D histogram-based method excelled compared to traditional techniques. While Otsu and Kapur’s methods rely solely on intensity variations, they often struggle to segment images with complex textures, as intensity-based thresholds alone may not capture the intricacies of textured regions.

* Otsu’s Method: Otsu’s method, which works by maximizing variance between two classes, tends to perform poorly in texture-rich images. The method may fail to capture the fine details of textures, leading to over-segmentation (too many regions) or under-segmentation (missing texture details).
* Kapur’s Method: While Kapur’s entropy-based method performs slightly better in texture-rich images, it still lacks the capacity to account for local textural information, leading to misclassifications in complex textures.
* Proposed 2D Histogram-Based Method: By incorporating local relative entropy, the proposed method was more effective at identifying and segmenting texture-rich regions.
  + Texture Sensitivity: The local entropy measure is particularly sensitive to texture, as it captures the randomness or regularity in pixel neighbourhoods. This allows the algorithm to segment textured regions more accurately than methods that rely solely on pixel intensities.
  + Enhanced Texture Segmentation: The method was tested on texture-rich images like Stone, where the texture is an integral part of the object. The 2D histogram-based method was able to segment these textures with a higher degree of precision, accurately differentiating between textured and non-textured areas of the image.

4. Comparative Literature Review

In existing literature, bilevel image segmentation using 2D histograms has been explored, but typically with less emphasis on local entropy. For instance:

* Jian Yu et al. (2011) proposed a similar 2D histogram-based method for thresholding, but their approach focused on joint distributions of intensity and spatial information, without fully exploring the benefits of local entropy in handling noisy images.
* Liao et al. (2001) explored a 2D Otsu method using spatial relationships between pixels, but the algorithm’s performance in noisy environments was limited compared to the new method introduced in this project.

Compared to these earlier approaches, the integration of local relative entropy in the 2D histogram not only improves noise resilience but also enhances the algorithm’s adaptability to different types of images (e.g., texture-rich, low contrast, noisy) and provides a more reliable segmentation, even in challenging conditions.

**Chapter 7: Conclusion**

This project successfully developed a two-dimensional histogram-based image segmentation method that integrates both brightness (gray level) and local relative entropy, resulting in a more accurate and robust segmentation technique. The primary objective was to overcome limitations of traditional thresholding methods, particularly in handling noisy images and identifying complex object boundaries. By combining these two key features—pixel intensity and local entropy—the method creates a richer representation of the image, capturing not only the brightness but also the texture and structural patterns within neighbouring pixels.

The integration of local relative entropy provides a distinct advantage, especially for noisy images. Unlike traditional methods like Otsu and Kapur, which rely solely on intensity-based histograms, the proposed method uses a two-dimensional histogram where one axis represents gray levels and the other represents local entropy. This dual-dimensional approach helps distinguish noise from significant textures, enabling more accurate segmentation even in heavily degraded images. For instance, in cases of salt-and-pepper or Gaussian noise, the method reduces noise artifacts in the segmented output, outperforming Otsu and Kapur by maintaining clearer object boundaries and reducing misclassification errors.

Moreover, the proposed method demonstrated superior performance in images with strong edges and fine details, where traditional methods often struggled. By considering both pixel intensity and the entropy of its surrounding neighborhood, the algorithm accurately identifies object boundaries and subtle textures, resulting in fewer edge misclassifications. This advantage extends to texture-rich images, where the method effectively differentiates textured regions from the background, an area where Otsu and Kapur generally underperform.

In comparative tests using standard images, the 2D histogram-based method consistently achieved better segmentation results in terms of accuracy and noise resilience, making it a powerful alternative to traditional segmentation techniques like Otsu and Kapur for a wide range of image types and conditions.

**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.