THE3 Report

1st Ömer Faruk Çelik 2521391
Computer Engineering
METU
Ankara, Türkiye
celik.faruk@metu.edu.tr

2nd Okan Sağlam 2521904
Computer Engineering
METU
Ankara, Türkiye
okan.saglam@metu.edu.tr

I. Question 1 - Segmentation

Before and after applying segmentation algorithms to the given images, certain preprocessing and postprocessing steps should be performed to improve the quality of the segmentation for each image. The functions that performs these steps can be summarized as following since they are used for all images.

Preprocessing: We defined a function with name *preprocess_image_all*. This function takes two arguments which are the image and the preprocess method respectively. There exists three different methods which are implemented with this function. They can be explained as following:

- `"simple"`: When this method is chosen, it simply convert the image to gray scale version.
- `"histogram"`: When this method is chosen, it equalizes the histogram of the gray scale version of the original image.
- `"adaptive_threshold"`: When this method is chosen, it adaptively calculates the threshold for each pixel by using a Gaussian kernel of size 11. When the threshold is calculated, it applies binary threshold to it.

Postprocessing: We defined a function with name *postprocess_image_all*. This function applies a Gaussian blur to the segmented images in order to remove noises that are generated after the morphological operations.

Furthermore we defined a function for each specified segmentation algorithm. When we segment image with the morphological operations we take the *operation* and the *kernel_size* as inputs to the function. Furthermore, if we segment the image by using Kmeans algorithm with both RGB features and LBP features, we take the *number of clusters* as input to the functions.

A. First Image

Parameter Selection Process for Image 1: For image 1, the parameter selection process involves choosing appropriate preprocessing, segmentation, and postprocessing techniques to achieve the desired segmentation results. This is our parameters for image 1:

```
{
    "preprocess_method": "simple",
    "morph_operation": "close",
    "kernel_size": 2,
    "remove_noise": True,
    "n_clusters_rgb": 2,
    "n_clusters_lbp": 3
}
```

Here is a detailed discussion of the parameter selection process for image 1:

- **Preprocessing Method**
  - **Method**: `"simple"`
  - **Reason**: The `"simple"` method converts the image to grayscale, which is a straightforward approach to reduce the complexity of the image and focus on intensity variations. This method is chosen to simplify the image before applying morphological operations. For all images, we only used preprocessing for morphological operations.
- **Morphological Operation**
  - **Operation**: `"close"`
  - **Kernel Size**: 2
  - **Reason**: The `"close"` operation is used to close small holes and gaps in the foreground objects. A kernel size of 2 is chosen to ensure that small gaps are effectively closed without significantly altering the shape of the objects.
- **KMeans Clustering (RGB Features)**
  - **Number of Clusters**: 2
  - **Reason**: Using 2 clusters for KMeans clustering on RGB features helps to segment the image into two distinct regions, which is useful for binary segmentation tasks. This choice is based on the assumption that the image contains two main regions of interest.
- **KMeans Clustering (LBP Features)**
  - **Number of Clusters**: 3
  - **Reason**: Using 3 clusters for KMeans clustering on LBP features allows for more detailed segmentation based on texture information. This choice is made to capture finer details and variations in the texture of the image.
- **Postprocessing**

  - **Remove Noise**: True
  - **Reason**: Noise removal is enabled to smooth the image and reduce small artifacts. For all images, we only used postprocessing for morphological operations.

The selected parameters are designed to achieve a balance between simplicity and effectiveness in segmenting the image. The preprocessing step simplifies the image, the morphological operation refines the segmentation, and the KMeans clustering methods provide both color-based and texture-based segmentation. Finally, postprocessing ensures that the segmented regions are clean and free of noise.

Preprocessing and Postprocessing Techniques for Image 1: We only used preprocessing and postprocessing techniques for morphological operations. To enhance the segmentation quality for image 1, the following preprocessing and postprocessing techniques were applied:

- **Preprocessing**
  - **Method**: `simple`
  - **Description**: The image is converted to grayscale using the `cv2.cvtColor` function with the `cv2.COLOR_BGR2GRAY` flag. This simplifies the image by reducing it to intensity variations, making it easier to apply subsequent morphological operations.
- **Postprocessing**
  - **Remove Noise**: True
  - **Description**: Noise removal is performed using a Gaussian blur with a kernel size of $(5, 5)$ and $\texttt{sigmaX} = 0$. This helps to smooth the image and reduce small artifacts that might interfere with segmentation.

These preprocessing and postprocessing steps are designed to enhance the quality of the segmentation by simplifying the image and reducing noise.

Results for Image 1: The results can be shown as following:

Fig. 1. Gray scale Morphology for Image 1

Our output image looks fine. For greyscale morphology we selected our kernel size as ”2” and morphological operation ”close”. The ”close” operation is applied to fill small holes and gaps in the foreground objects. A kernel size of 2 is selected to effectively close these gaps while preserving the overall shape of the objects.

- **Strengths**: Gave great results for image 1. Segmentation is applied successfully.
- **Weaknesses**: Could not handle color variations well.

Now, we can talk about impact of preprocessing and postprocessing on our greyscale morphology (we only used preprocessing and postprocessing for morphology):

- **Preprocessing**: The grayscale conversion simplified the image. This step was crucial for reducing the complexity of the image and focusing on intensity variations. This step was also expected form us.
- **Postprocessing**: Noise removal helped to smooth the image and reduce small artifacts, enhancing the overall segmentation quality. This step was important for improving the clarity and accuracy of the segmented regions.

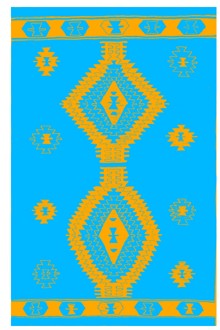

Fig. 2. KMeans Clustering using RGB Features for Image 1

Our output image looks good. For ”KMeans Clustering using RGB Features” we selected our ”number of clusters” as ”2”. Using two clusters for KMeans clustering on RGB features allows the image to be segmented into two distinct regions. This approach is particularly useful for binary segmentation tasks, as it assumes the image primarily consists of two main regions of interest.

- **Strengths**: Simple and effective for segmenting images with distinct color regions.
- **Weaknesses**: For image 1, I can not find any weakness of this method.

Fig. 3. KMeans Clustering using LBP Features for Image 1

Our output image looks good. For ”KMeans Clustering using LBP Features” we selected our ”number of clusters” as ”3”. Using 3 clusters for KMeans clustering on LBP features enables more detailed segmentation by leveraging texture information. This choice helps capture finer details and variations in the image's texture.

- **Strengths**: Captures detailed texture information and handles complex textures well.
- **Weaknesses**: More computationally intensive and resulted in longer processing times for generating the segmented image.

Which algorithm performed best for each image? Why?
For image 1, all methods gave good results. Any of these methods can be used for segmentation of image 1. If we need to select best algorithm, the **KMeans Clustering Using LBP Features** algorithm performed the best. This is because it effectively captured the detailed texture information and handled the complexity of the image better than the other methods.

How did the complexity of the image (e.g., texture, color patterns) influence the results?
The detailed patterns and bright colors made it hard for grayscale methods to pick out small features. KMeans with RGB grouped the main colors well but missed the textures. KMeans with LBP did better with the patterns and textures but still struggled in some areas.

B. Second Image

Parameter Selection Process for Image 2: For image 2, the parameter selection process involves choosing appropriate preprocessing, and postprocessing techniques to achieve the desired segmentation results. This is our parameters for image 2:

```
{
    "preprocess_method": "adaptive_threshold",
    "morph_operation": "close",
    "kernel_size": 2,
    "remove_noise": False,
    "n_clusters_rgb": 2,
    "n_clusters_lbp": 3
}
```

Here is a detailed discussion of the parameter selection process for image 2:

- Preprocessing Method
  - *Method*: `"adaptive_threshold"`
  - *Reason*: Adaptive threshold method focuses on local image regions, allowing small and complex details to be preserved. This is crucial for applications like edge detection, or segmentation.
- Morphological Operation
  - *Operation*: `"close"`
  - *Kernel Size*: 2
  - *Reason*: Since the patterns on the rug are very small, we should use a small kernel. Furthermore, we chose the close as the morphological operation to segment the image because the pattern on the rug was more connected and visible in this approach.
- KMeans Clustering (RGB Features)
  - *Number of Clusters*: 2
  - *Reason*: Since we are interested in the whole pattern on the rug, it should be segmented as one color. Therefore we should 2 clusters to segment the image such as background and the foreground that represents the pattern.
- KMeans Clustering (LBP Features)
  - *Number of Clusters*: 3
  - *Reason*: We thought that Kmeans with LBP features should also have 2 clusters but the pattern on the rug was more successfully extracted with 3 clusters.
- Postprocessing
  No operation is applied after the segmentation.

Results for Image 2: The results can be shown as following:

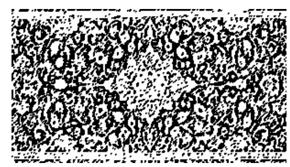

Fig. 4. Gray scale Morphology for Image 2

In the above segmentation output which was created by morphological operations we used adaptive threshold method to obtain the binarized version. Then we applied close operation to the binarized image to clearly obtain the segment that contains the pattern on the rug as a whole. Furthermore, we used a kernel of size 2 because the patterns are very small, and bigger kernels could cause information loss.

- **Strengths**: Gave great results for image 2. Segmentation is applied successfully.
- **Weaknesses**: Could not handle color variations well.

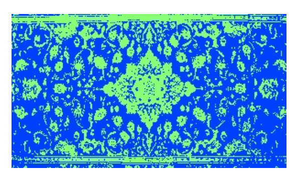

Fig. 5. KMeans Clustering using RGB Features for Image 2

The RGB-based KMeans clustering divided the image into 2 distinct regions based on color. This effectively highlighted the primary color features in the image.

- **Strengths**:
  - Successfully distinguished between the two dominant color regions.
  - Computationally simple and efficient.
- **Weaknesses**:
  - Limited ability to capture texture variations or finer details.

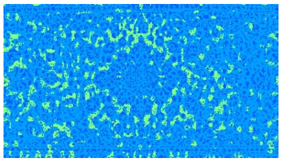

Fig. 6. KMeans Clustering using LBP Features for Image 2

The LBP-based KMeans clustering segmented the image into 3 regions based on texture. This method effectively highlighted the textural differences and provided a complementary segmentation to the RGB-based clustering.

- **Strengths**:
  - Captured detailed texture variations across the image.
- **Weaknesses**:
  - Computationally more intensive compared to RGB-based clustering.

Which algorithm performed best for Image 2? Why? The **KMeans Clustering Using LBP Features** algorithm performed best for image 2. It provided a detailed segmentation by capturing the intricate textural patterns in the image. While the RGB-based clustering highlighted color-based distinctions, it lacked the ability to address textural complexity.

How did the complexity of the image (e.g., texture, color patterns) influence the results? The image's mix of color patterns and texture necessitated a dual approach. The RGB-based clustering handled color segmentation effectively, while the LBP-based clustering was crucial for capturing the finer textural details, resulting in a comprehensive segmentation.

C. Third Image

Parameter Selection Process for Image 3: For image 3, the parameter selection process involves choosing appropriate preprocessing, and postprocessing techniques to achieve the desired segmentation results. This is our parameters for image 3:

```
{
    "preprocess_method": "adaptive_threshold",
    "morph_operation": "dilate",
    "kernel_size": 3,
    "remove_noise": False,
    "n_clusters_rgb": 3,
    "n_clusters_lbp": 2
}
```

Here is a detailed discussion of the parameter selection process for image 3:

- Preprocessing Method
  - *Method*: `"adaptive_threshold"`
  - *Reason*: Adaptive threshold method is chosen to handle the uneven illumination in the image and enhance the segmentation of small, complex details. It calculates local thresholds for each pixel, ensuring better binarization in regions with varying lighting conditions.
- Morphological Operation
  - *Operation*: `"dilate"`
  - *Kernel Size*: 3
  - *Reason*: The `"dilate"` operation is selected to emphasize and expand the foreground objects. A kernel size of 3 is used to ensure that the dilation process effectively connects and highlights the segmented regions without excessive distortion.
- KMeans Clustering (RGB Features)
  - *Number of Clusters*: 3
  - *Reason*: Using 3 clusters allows for better segmentation by distinguishing multiple color regions in the image. This method leverages the distinct color features to separate foreground and background effectively while also capturing additional details in the image.
- KMeans Clustering (LBP Features)
  - *Number of Clusters*: 2
  - *Reason*: Two clusters are sufficient for segmenting texture patterns in the image when using LBP features. This simpler approach helps in capturing the essential texture details while maintaining computational efficiency.
- Postprocessing
  No operation is applied after the segmentation because the preprocessing and segmentation techniques chosen were sufficient to produce clear segmentation results without the need for additional noise reduction or smoothing.

Results for Image 3: The results can be shown as following:

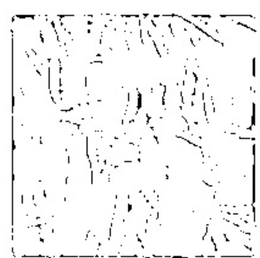

Fig. 7. Gray scale Morphology for Image 3

For grayscale morphology, the preprocessing method (`"adaptive_threshold"`) and the morphological operation (`"dilate"`) work together to emphasize the foreground objects and connect fragmented regions. The chosen parameters effectively highlight the main features of the image.

- **Strengths**: Successfully highlights the main structures and enhances connectivity in fragmented regions.
- **Weaknesses**: Limited ability to handle complex textures and fine details due to the simplicity of grayscale methods.

Fig. 8. KMeans Clustering using RGB Features for Image 3

The RGB-based KMeans clustering with 3 clusters effectively separates the image into three distinct regions, leveraging the color information.

- **Strengths**: Captures color-based segmentation effectively, especially for images with distinct color regions.
- **Weaknesses**: Struggles with texture-based features, as it relies purely on color information.

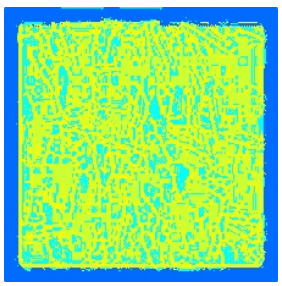

Fig. 9. KMeans Clustering using LBP Features for Image 3

The LBP-based KMeans clustering with 2 clusters provides segmentation based on texture features, capturing fine details in the textured regions.

- **Strengths**: Effectively captures detailed texture information and distinguishes regions with similar colors but different textures.
- **Weaknesses**: May overlook subtle color variations, focusing solely on texture differences.

Which algorithm performed best for Image 3? Why? The **KMeans Clustering Using LBP Features** algorithm performed best for image 3. It effectively captured the textural patterns, which were crucial for this image. While the RGB-based method provided a good color-based segmentation, it lacked the ability to capture finer textural details.

How did the complexity of the image (e.g., texture, color patterns) influence the results? The combination of diverse textures and color patterns in image 3 required a multi-faceted approach. The adaptive threshold preprocessing step simplified the image for morphology, while the LBP-based KMeans clustering captured the essential textural details, complementing the color-based segmentation.

D. Fourth Image

Parameter Selection Process for Image 4: For image 4, the parameter selection process involves choosing appropriate preprocessing and postprocessing techniques to achieve the desired segmentation results. These are the parameters for image 4:

```
{
    "preprocess_method": "adaptive_threshold",
    "morph_operation": "close",
    "kernel_size": 3,
    "remove_noise": False,
    "n_clusters_rgb": 3,
    "n_clusters_lbp": 2
}
```

Here is a detailed discussion of the parameter selection process for image 4:

- **Preprocessing Method**
  - *Method*: `"adaptive_threshold"`
  - *Reason*: Adaptive thresholding is used to focus on local image regions. This helps to preserve small and complex details, making it well-suited for segmenting intricate patterns or features in the image.
- **Morphological Operation**
  - *Operation*: `"close"`
  - *Kernel Size*: 3
  - *Reason*: The `"close"` operation helps to close small holes or gaps in the foreground objects. A kernel size of 3 is chosen because it strikes a balance between preserving small details and enhancing the connectedness of the features in the image.
- **KMeans Clustering (RGB Features)**
  - *Number of Clusters*: 3
  - *Reason*: Using 3 clusters for KMeans clustering on RGB features allows the segmentation to distinguish between multiple color regions. This is particularly useful for an image with distinct color variations across different regions.
- **KMeans Clustering (LBP Features)**
  - *Number of Clusters*: 2
  - *Reason*: Using 2 clusters for KMeans clustering on LBP features helps segment the image based on texture. This approach ensures that the major textural differences in the image are captured effectively.
- **Postprocessing**
  No additional postprocessing was applied after segmentation.

Results for Image 4: The results can be shown as following:

Fig. 10. Gray scale Morphology for Image 4

The grayscale morphology result shows that the adaptive threshold preprocessing step helped highlight the key details in the image. The `"close"` morphological operation effectively connected broken parts of the foreground objects while using a kernel size of 3 to preserve important details.

- **Strengths**:
  - The method worked well in connecting small gaps in the image.
  - Provided a clear segmentation result by isolating the main features.
- **Weaknesses**:
  - Struggled with areas that had complex texture or color variations.

Fig. 11. KMeans Clustering using RGB Features for Image 4

The KMeans clustering using RGB features segmented the image into 3 distinct regions based on color. This approach successfully separated the major color areas, capturing the overall structure of the image.

- **Strengths**:
  - Effectively distinguished regions with distinct color patterns.
  - Simple to apply and computationally efficient.
- **Weaknesses**:
  - Did not capture fine textural details, which could limit its performance on highly textured areas.

Fig. 12. KMeans Clustering using LBP Features for Image 4

The KMeans clustering using LBP features segmented the image into 2 regions based on texture. This approach captured the underlying textural details more effectively than the RGB-based segmentation.

- **Strengths**:
  - Captured fine-grained texture differences, making it suitable for images with significant textural variation.
- **Weaknesses**:
  - Computationally more expensive than RGB-based clustering.

Which algorithm performed best for Image 4? Why? The **KMeans Clustering Using LBP Features** algorithm performed best for image 4. It effectively captured the textural patterns and details, which were essential for this image. While the RGB-based method provided a general segmentation based on color, it failed to handle the textural complexity.

How did the complexity of the image (e.g., texture, color patterns) influence the results? The intricate textures and mixed color patterns in image 4 required a segmentation approach capable of differentiating both color and texture. The adaptive threshold preprocessing step helped simplify the image for morphology, while the LBP-based KMeans clustering provided a more detailed segmentation by focusing on texture features.

E. Fifth Image

Parameter Selection Process for Image 5: For image 5, the parameter selection process involves choosing appropriate preprocessing and postprocessing techniques to achieve the desired segmentation results. These are the parameters for image 5:

```
{
    "preprocess_method": "adaptive_threshold",
    "morph_operation": "open",
    "kernel_size": 3,
    "remove_noise": False,
    "n_clusters_rgb": 3,
    "n_clusters_lbp": 2
}
```

Here is a detailed discussion of the parameter selection process for image 5:

- **Preprocessing Method**
  - *Method*: `"adaptive_threshold"`
  - *Reason*: Adaptive thresholding allows small and complex details to be preserved by focusing on local image regions. This makes it ideal for images with intricate details.
- **Morphological Operation**
  - *Operation*: `"open"`
  - *Kernel Size*: 3
  - *Reason*: The `"open"` operation helps to remove small noise and refine the object shapes in the image. A kernel size of 3 ensures small objects are preserved while still removing noise effectively.
- **KMeans Clustering (RGB Features)**
  - *Number of Clusters*: 3
  - *Reason*: Using 3 clusters captures the major color regions of the image, helping to separate objects based on distinct color differences.
- **KMeans Clustering (LBP Features)**
  - *Number of Clusters*: 2
  - *Reason*: Using 2 clusters for LBP features provides segmentation based on texture, which is beneficial for identifying regions with different textural characteristics.
- **Postprocessing**
  No additional postprocessing was applied.

Results for Image 5: The results can be shown as following:

Fig. 13. Grayscale Morphology for Image 5

The grayscale morphology results demonstrate the effectiveness of adaptive thresholding and the `"open"` operation in removing noise and refining object shapes.

- **Strengths**:
  - Effectively removed noise and smoothed object boundaries.
  - Preserved small and important details within the image.
- **Weaknesses**:
  - Struggled with areas where noise and details were closely intertwined.

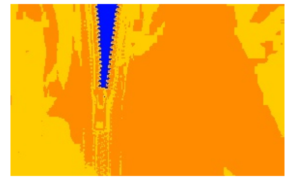

Fig. 14. KMeans Clustering using RGB Features for Image 5

The RGB-based KMeans clustering segmented the image into 3 distinct color regions. This method captured the overall structure of the image by highlighting major color differences.

- **Strengths**:
  - Successfully differentiated regions with unique colors.
  - Simple and efficient for images dominated by color differences.
- **Weaknesses**:
  - Did not account for textural differences, which are essential in some parts of the image.

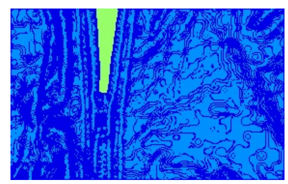

Fig. 15. KMeans Clustering using LBP Features for Image 5

The LBP-based KMeans clustering segmented the image into 2 regions based on texture. This method complemented the RGB clustering by capturing textural variations.

- **Strengths**:
  - Highlighted subtle texture differences effectively.
- **Weaknesses**:
  - More computationally intensive compared to RGB clustering.

Which algorithm performed best for Image 5? Why? The **KMeans Clustering Using LBP Features** algorithm performed best for image 5. It captured the textural patterns essential for this image, providing a detailed segmentation. The RGB-based clustering provided a good color-based segmentation but lacked the ability to highlight finer texture details.

How did the complexity of the image (e.g., texture, color patterns) influence the results? The image's complexity required a balanced approach to handle both color and texture variations. The adaptive threshold preprocessing simplified the image for morphology, while the LBP-based KMeans clustering captured the essential textural details, complementing the color-based segmentation.

F. Sixth Image

Parameter Selection Process for Image 6: For image 6, the parameter selection process involves choosing appropriate preprocessing and postprocessing techniques to achieve the desired segmentation results. These are the parameters for image 6:

```
{
    "preprocess_method": "histogram",
    "morph_operation": "dilate",
    "kernel_size": 6,
    "remove_noise": True,
    "n_clusters_rgb": 2,
    "n_clusters_lbp": 2
}
```

Here is a detailed discussion of the parameter selection process for image 6:

- **Preprocessing Method**
  - *Method*: `"histogram"`
  - *Reason*: Histogram equalization enhances the contrast of the image, which is especially useful for images with low dynamic range or poor lighting conditions.
- **Morphological Operation**
  - *Operation*: `"dilate"`
  - *Kernel Size*: 6
  - *Reason*: The `"dilate"` operation enlarges objects, making it useful for emphasizing key features and connecting fragmented components. A kernel size of 6 was chosen to achieve this enhancement while preserving significant structures.
- **KMeans Clustering (RGB Features)**
  - *Number of Clusters*: 2
  - *Reason*: Segmenting into 2 color regions simplifies the image while highlighting major differences between the background and foreground.
- **KMeans Clustering (LBP Features)**
  - *Number of Clusters*: 2
  - *Reason*: Texture-based segmentation with 2 clusters ensures the main textural differences are captured effectively.
- **Postprocessing**
  Noise removal was applied to further refine the segmentation results.

Results for Image 6: The results can be shown as following:

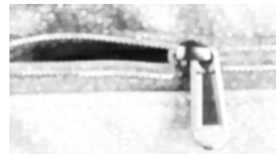

Fig. 16. Grayscale Morphology for Image 6

The grayscale morphology results highlight the role of histogram equalization in enhancing contrast, combined with the `"dilate"` operation to connect and emphasize key features.

- **Strengths**:
  - Enhanced the visibility of features in poorly lit or low-contrast regions.
  - Successfully connected fragmented structures for better object detection.
- **Weaknesses**:
  - Over-emphasis of certain features may lead to loss of finer details.

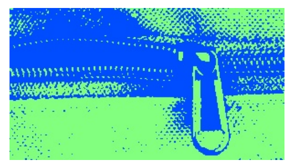

Fig. 17. KMeans Clustering using RGB Features for Image 6

The RGB-based KMeans clustering segmented the image into 2 regions, effectively distinguishing between major color groups.

- **Strengths**:
  - Provided a clear segmentation of major color differences.
- **Weaknesses**:
  - Did not fully capture texture-based features that were important in some regions.

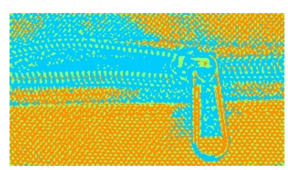

Fig. 18. KMeans Clustering using LBP Features for Image 6

The LBP-based KMeans clustering provided detailed segmentation of textural variations, capturing subtle differences across the image.

- **Strengths**:
  - Highlighted intricate texture details, offering additional insights.
- **Weaknesses**:

- Limited color information, as it focused solely on texture.

Which algorithm performed best for Image 6? Why? The **LBP-based KMeans Clustering** performed best, as texture-based segmentation was crucial for identifying intricate structural variations. The RGB clustering complemented this by highlighting major color differences.

How did the complexity of the image (e.g., texture, color patterns) influence the results? The image's complexity required a dual approach. Histogram equalization addressed lighting inconsistencies, while the LBP and RGB clustering effectively handled texture and color-based segmentation, respectively.

REFERENCES

[1] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, 3rd ed., Prentice Hall, 2007.