Tribhuvan University
Institute of Science and Technology

Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu
2023

Seminar Report on
"Image Segmentation using K-means Clustering Algorithm"

In partial fulfillment of the requirement for Master's degree in Computer Science and Information Technology (M.Sc. CSIT), 1st Semester

Submitted to:
Central Department of Computer Science and Information Technology, Tribhuvan University, Kirtipur, Kathmandu, Nepal

Submitted By:
Babita Rimal (7915008)

Tribhuvan University

Institute of Science and Technology

Supervisor Recommendation

This is to certify that Miss. Babita Rimal has submitted the seminar report on the topic "**Image Segmentation using K-means Clustering Algorithm"** for the partial fulfilment of Masters of Science in Computer Science and Information Technology, first semester. I hereby, declare that this seminar report has been approved.

____

Supervisor

Assoc. Prof. Mr. Jagdish Bhatta

Central Department of Computer Science and Information Technology

Certificate of Approval

This is to certify that the seminar report prepared by Miss. Babita Rimal "**Image Segmentation using K-means Clustering Algorithm”** in partial fulfilment of the requirements for the degree of Masters of Science in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Evaluation Committee

……………..…………………………..

Asst. Prof. Sarbin Sayami

(H.O.D)

Central Department of Computer Science and Information Technology

………..…………………………………

Assoc. Prof. Jagdish Bhatta

(Supervisor)

Central Department of Computer Science and Information Technology

……………………………

(External)

Acknowledgement

The success and final outcome of this report required a lot of guidance and assistance from many people and I am very fortunate to have got this all along the completion. I am very glad to express my deepest sense of gratitude and sincere thanks to my highly respected and esteemed supervisor **Assoc. Prof. Jagdish Bhatta,** Central Department of Computer Science and Information Technology for his valuable supervision, guidance, encouragement, and support for completing this paper.

I am also thankful to **Asst. Prof. Sarbin Sayami**, HOD of Central Department of Computer Science and Information Technology for his constant support throughout the period. Furthermore, with immense pleasure, I submit by deepest gratitude to the Central Department of Computer Science and Information Technology, Tribhuvan University, and all the faculty members of CDCSIT for providing the platform to explore the knowledge of interest. At the end I would like to express my sincere thanks to all my friends and others who helped me directly or indirectly.

Thanking you,

Babita Rimal (7915008)

Abstract

Image segmentation is a fundamental task in computer vision that involves dividing an image into meaningful regions or objects. It plays a crucial role in various applications such as object recognition, scene understanding, and medical image analysis. One popular and widely used technique for image segmentation is the K-means clustering algorithm which has been used for this project.

K-means clustering is an unsupervised learning algorithm that partitions data into K clusters based on similarity measures. K-means clustering is applied to image data i.e., breast cancer(malignant) where group of pixels with similar characteristics together and separate them from other regions, effectively achieving image segmentation. The algorithm iteratively assigns pixels to clusters based on their color or feature similarity, optimizing a clustering criterion. Varying the value of k the changes with the cluster is observed. The quality of the cluster is evaluated by using DBI value. Furthermore, the project will explore the strengths and limitations of K-means clustering for image segmentation.

Keywords: clustering, image segmentation, K-means clustering algorithm, medical image analysis, object recognition, unsupervised learning.

Table of Content

Supervisor Recommendation i

Certificate of Approval ii

Acknowledgement iii

Abstract iv

Table of Content v

List of Figures vii

List of Abbreviations viii

Chapter 1: Introduction 1

1.1 Introduction 1

1.2 Problem Statement 1

1.3 Objective 2

Chapter 2: Background Study and Literature Review 3

2.1 Background Study 3

2.1.1 Image Segmentation 3

2.2 Literature Review 4

Chapter 3: Methodology 5

3.1 Dataset Description 5

3.2 Data Processing 5

3.3 Image Segmentation Using K-means clustering 6

Chapter 4: Implementation and Result Analysis 7

4.1 Implementation 7

4.1.1 Implementation Tools 7

4.2.2 Implementation Details 7

4.2 Result Analysis 10

Chapter 5: Conclusion and Future Recommendations 14

5.1 Conclusion 14

5.2 Future Recommendations 14

References 15

List of Figures

Figure 1: Flow Chart 5
Figure 2: Code to Reduce Image Size 6
Figure 3:Code for load and Resize Image.. 7
Figure 4:Code to Convert Image to Numpy Array... 7
Figure 5: Code to Define number of Clusters 7
Figure 6: Code to Display Input Image... 8
Figure 7: Code for Image Segmentation using K-means Clustering 9
Figure 8: Code to Display Segment Image 9
Figure 9: Input Image.. 10
Figure 10: Output Image Segment.. 12

List of Abbreviations

BC	Breast Cancer
DBI	Davies-Bouldin Index
IS	Image Segmentation

Chapter 1: Introduction

1.1Introduction

Image Segmentation (IS) is a fundamental task in computer vision that involves dividing an image into meaningful and coherent regions or objects. The goal is to partition the image into distinct regions where pixels within each region share similar visual properties such as color, texture, intensity, or motion. The purpose of image segmentation is to extract semantically meaningful regions from an image, allowing for higher-level analysis and understanding of the visual content. By segmenting an image, we can identify and differentiate various objects or regions of interest, enabling a wide range of applications such as object recognition, scene understanding, image editing, medical image analysis, and more.

One fascinating application of K-means clustering is in image segmentation. For this report, K-means clustering algorithm is used to clearly observe the mammographic image of breast cancer (BC). K-means clustering is a type of unsupervised learning algorithm that aims partition an image into K clusters, where each cluster represents a distinct region or object. The algorithm achieves this by iteratively assigning pixels to the nearest cluster center and updating the centers based on the average values of the assigned pixels. This process continues until convergence, where the cluster centers no longer change significantly. It provides a straightforward approach to segmenting images into regions based on color or intensity similarity, without requiring prior knowledge about the image content.

1.2Problem Statement

The absence of image segmentation leads to significant problems such as confusion in object differentiation, inaccurate object recognition, limited semantic understanding, inefficient image editing, compromised medical diagnosis, inefficient compression and transmission, and challenges in autonomous systems. Breast cancer is the second most common malignancy among Nepalese women that places a substantial burden on the Nepalese healthcare system. Lack of proper analysis of mammographic image can leads to misinterpretation and wrong precaution. Addressing these issues requires effective segmentation techniques to overcome the challenges posed by the lack of proper segmentation. Overall, image segmentation is crucial in several fields to extract meaningful information and enable advanced applications based on visual content.

1.3Objective

The main objective includes;

- To investigate the effectiveness of the K-means algorithm for image segmentation.
- To analyze the impact of varying the number of clusters on segmentation results.
- To minimize the misinterpretation and detects the cancer effectively.

Chapter 2: Background Study and Literature Review

2.1 Background Study

2.1.1 Image Segmentation

Image segmentation is a vital task in computer vision, involving the division of an image into distinct regions based on visual characteristics. The K-means clustering algorithm is a widely used approach for image segmentation, iteratively assigning pixels to clusters based on their similarity to cluster centroids. Image segmentation techniques can be broadly categorized into two main types:

1. Boundary-based Segmentation:

Boundary-based segmentation focuses on identifying and delineating boundaries or edges between different objects or regions in an image. It aims to detect sharp transitions in intensity or color gradients to separate distinct regions. Common boundary-based techniques include edge detection algorithms, such as Canny edge detection, which highlight areas of rapid intensity changes.

2. Region-based Segmentation:

Region-based segmentation, on the other hand, groups pixels together based on their similarities, creating coherent regions within the image. This approach assigns pixels to regions by considering their color, texture, or other feature similarities. Region growing, thresholding, and clustering algorithms like K-means or mean-shift are commonly used in region-based segmentation.

Image segmentation is a challenging task due to variations in lighting conditions, object appearances, occlusions, and image complexity. Researchers continue to develop innovative algorithms and methodologies to tackle these challenges, leading to more accurate and efficient segmentation techniques. However, challenges such as high-dimensional data representation, initialization sensitivity, and post-processing requirements exist. Understanding the principles, strengths, limitations, and practical considerations of the K-means algorithm is essential for researchers and practitioners in computer vision, providing a foundation for exploring advanced segmentation approaches and facilitating image analysis and understanding. However, it's important to note that K-means clustering has limitations in handling complex image structures, overlapping regions, or cases where the desired segments have irregular shapes. In such scenarios, more advanced segmentation algorithms that incorporate spatial information or utilize deep learning techniques may be more appropriate.

2.2 Literature Review

Image segmentation is a prominent area of research in computer vision, aiming to partition an image into meaningful regions or objects. The K-means clustering algorithm has been widely utilized as a popular technique for image segmentation. This literature review provides an overview of key studies and advancements in the field of image segmentation using the K-means clustering algorithm.

Early works on image segmentation using K-means clustering focused on applying the algorithm directly to pixel values or color features. For instance, an image segmentation method based on K-means clustering using color and texture features has been proposed[1]. They demonstrated effective segmentation results by extracting color and texture information from each pixel and applying K-means clustering to group similar pixels.

To address the limitations of using K-means clustering solely on pixel values, researchers have explored incorporating additional features or information. Khan, Zubair; [2] proposed an improved K-means algorithm for image segmentation by combining color and texture features with spatial information. By considering the spatial relationship between pixels, their approach achieved more accurate segmentation results.

Furthermore, Pham[3] proposed an article provides an overview of various image segmentation methods, including clustering techniques. It discusses the application of clustering algorithms, such as K-means, in medical image segmentation. The authors highlight the challenges and advancements in medical image segmentation and discuss the potential of clustering algorithms for this domain.

Additionally, clustering algorithms inspired by K-means with improved watershed algorithm have been proposed for medical image segmentation[4]. Sulaiman [5] presented a variant called Fuzzy K-means clustering for image segmentation. By incorporating fuzzy membership values, their approach accommodated uncertainties in pixel assignment, leading to more flexible and accurate segmentation results.

In recent years, deep learning techniques have revolutionized image segmentation. However, K-means clustering still finds application in combination with deep learning approaches. Sharifrazi [6] proposed a hybrid segmentation method that combined K-means clustering with a deep convolutional neural network (CNN). Their approach first employed K-means clustering to generate initial region proposals, which were then refined using the CNN to achieve more precise segmentation results.

Chapter 3: Methodology

Image segmentation of a mammographic image (malignant) is done using k-means clustering algorithm. The detail flowchart of the system is presented below.

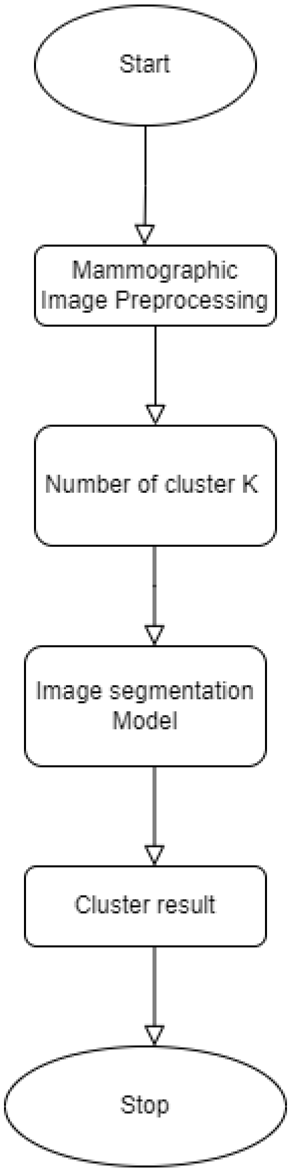


Figure 1: Flow Chart

3.1 Dataset Description

Input data for this project i.e., mammographic image is taken from Kaggle. Single mammographic image with malignant tumor is taken from the dataset which is fed as the input for image segmentation. Talking about the dataset, it contains 1575 mammographic image which was collected from 600 female patients of age 25 to 75 in 2018.

3.2 Data Processing

For faster processing the input data is reduced into smaller size.

```python
# Reduce image size for faster processing
resized = cv2.resize(img, (640, 480), interpolation=cv2.INTER_AREA)
```

Figure 2: Code to Reduce Image Size

3.3 Image Segmentation Using K-means clustering

K-means clustering is an unsupervised machine learning algorithm commonly used for clustering tasks, including image segmentation. It aims to divide a dataset into a predefined number of clusters, where each cluster represents a group of similar data points. Image segmentation using K-means clustering works as:

The algorithm for image segmentation using the K-means clustering algorithm is as follows:

1. Preprocessing: Read the image and convert it into a suitable format for analysis. This typically involves representing the image as a matrix of pixels, where each pixel contains color information (RGB). It also includes the process to reduce the image size for faster processing.

2. Reshape the Image: Reshape the image into a 2D array of pixels, where each row represents a pixel and its associated features (e.g., color values). This allows the image to be treated as a dataset for clustering.

3. Feature Extraction: Depending on the specific segmentation criteria, additional feature extraction or transformation is performed to reduce the dimensionality of the feature space.

4. K-means Clustering: Apply the K-means clustering algorithm to the dataset of pixels. Set the desired number of clusters, K, based on the desired level of segmentation. K-means seeks to minimize the within-cluster sum of squares by iteratively assigning pixels to the nearest cluster centroid and updating the centroids based on the assigned pixels. The algorithm repeats this process until convergence.

5. Label Assignment: After the K-means algorithm converges, each pixel is assigned a label corresponding to the cluster it belongs to. This label indicates which segment or region the pixel is associated with.

6. Reshape the Labels: Reshape the labels obtained from clustering to match the shape of the original image. Random color is assigned to each cluster which allows to visualize the segmented image where each pixel corresponds to its assigned cluster label.

7. Post-processing: Depending on the specific application, you might perform post-processing steps to refine the segmentation results. It includes operations like smoothing the boundaries between segments, removing small or noisy regions, or merging similar adjacent regions.

The resulting segmented image will have distinct regions or segments based on the similarities between pixels.

Chapter 4: Implementation and Result Analysis

4.1 Implementation

4.1.1 Implementation Tools

Python is used as the programming language to code the program. Vs Code is used as an IDE. Similarly, different libraries are also used as:

- OpenCV
- scikit-learn
- NumPy
- Matplotlib

4.2.2 Implementation Details

Implementation details includes;

1. **Load and Resize Image**: The code loads an image using `cv2.imread()` and resizes it using `cv2.resize()` to a smaller size (640x480) for faster processing.

```
# Load image
img = cv2.imread(r'D:\notes1st\seminar\malignant.png')

# Reduce image size for faster processing
resized = cv2.resize(img, (640, 480), interpolation=cv2.INTER_AREA)
```

Figure 3:Code for load and Resize Image

2. Convert Image to Numpy Array: The resized image is converted to a Numpy array using `np.array()` and reshaped into a 2D array of pixels using `reshape()`. This creates a dataset where each row represents a pixel and its associated features (color values).

```
# Convert image to numpy array
data = np.array(resized).reshape(-1, 3)
```

Figure 4:Code to Convert Image to Numpy Array

3. Define the Number of Clusters: The code specifies a list of `n_clusters_values` that contains the number of clusters to be tested during segmentation. Each value in this list represents the desired number of segments to divide the image into.

```
# List of n_clusters values to try
n_clusters_values = [2, 3, 5, 6, 8]
```

Figure 5: Code to Define number of Clusters

4. Display the Input Image: The original input image is displayed using `plt.imshow()` and `plt.show()` to show the image that will undergo segmentation.

```
# Display the input image
plt.figure(figsize=(5, 5))
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.axis('off')
plt.title('Input Image')
plt.show()
```

Figure 6: Code to Display Input Image

5. Segmentation using K-means Clustering: For each value of `n_clusters` in the list, the code performs the following steps:

- Create a K-means clustering model using `KMeans()` from scikit-learn, specifying the number of clusters (`n_clusters`).

- Fit the model to the image data using `model.fit()`. This applies the K-means clustering algorithm to group pixels into `n_clusters` segments based on their color values.

- Obtain the cluster labels for each pixel using `model.predict()`. Each pixel is assigned a label representing the cluster it belongs to.

- Create a new image array, `new_img`, with the same shape as the resized image to store the segmented image.

- Generate a random color palette using `np.random.randint()` to assign colors to each segment. The number of colors in the palette corresponds to the number of clusters (`model.n_clusters`).

- Iterate over the pixels and their corresponding labels. For each pixel, retrieve its position and assign the corresponding color from the palette based on its label to `new_img`.

```
for i, n_clusters in enumerate(n_clusters_values):
    # Use k-means clustering to cluster pixels into segments
    model = KMeans(n_clusters=n_clusters, n_init=10)
    model.fit(data)

    # Get cluster labels
    labels = model.predict(data)

    new_img = np.zeros((resized.shape[0], resized.shape[1], 3), dtype=np.uint8)

    # Assign random colors to each segment
    colors = np.random.randint(0, 255, (model.n_clusters, 3), dtype=np.uint8)

    for j, label in enumerate(labels):
        y = int(j // resized.shape[1])
        x = int(j % resized.shape[1])
        if y >= 0 and y < resized.shape[0] and x >= 0 and x < resized.shape[1]:
            color = colors[label]
            new_img[y, x] = color
```

Figure 7: Code for Image Segmentation using K-means Clustering

- Display the segmented image using `plt.imshow()` and `plt.show()`.

```
    # Display segmented image
    axes[i].imshow(new_img)
    axes[i].axis('off')
    axes[i].set_title(f'Segmented Image (n_clusters={n_clusters})')

plt.tight_layout()
plt.show()
```

Figure 8: Code to Display Segment Image

6. Repeat for Different Number of Clusters: The above steps are repeated for each value of `n_clusters` in the `n_clusters_values` list, resulting in a set of segmented images, each with a different number of clusters.

7. Calculation for DBI value: The generated cluster is now evaluating the quality of clustering using Davies-Bouldin Index (DBI).

```
# Display the Davies-Bouldin Index
axes[i, 1].text(0.5, 0.5, f'DBI: {dbi:.2f}', fontsize=12, ha='center')
axes[i, 1].axis('off')
```

Figure 9: Code for DBI Calculation

The segmented images visually represent different regions or objects in the original image, where each segment corresponds to a cluster. The number of clusters determines the level of detail in the segmentation, with more clusters providing finer segmentation.

By analyzing the segmented images for different numbers of clusters, you can observe how the image segmentation results vary based on the chosen number of clusters. This helps in determining the optimal number of clusters for a particular image or application.

4.2 Result Analysis

A mammographic image with malignant tumor is fed as an input. After applying k-means clustering algorithm with a list of value of number of clusters we get the output with corresponding numbers of segments as cluster.

The following image shows the input as well as output image:

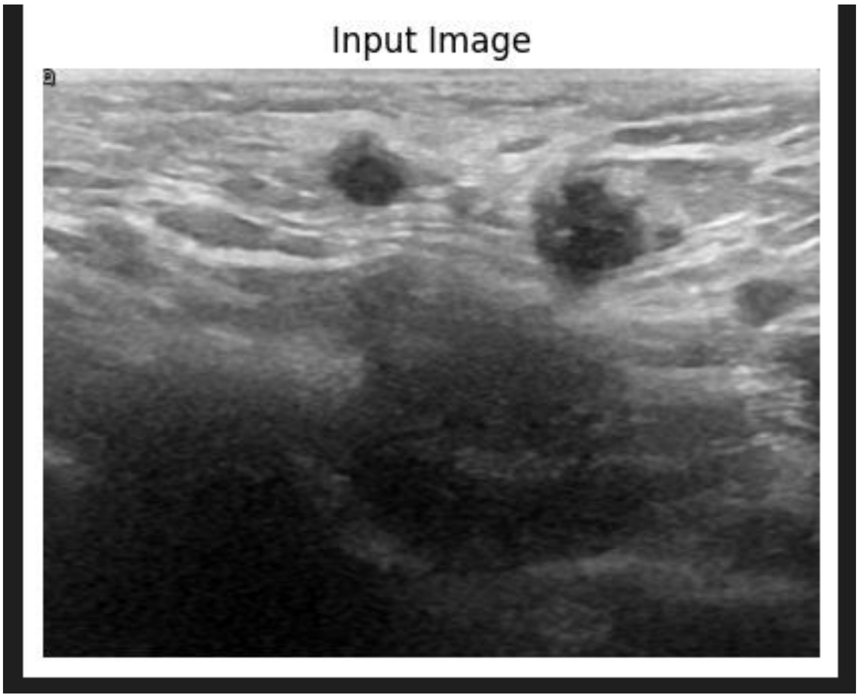


Figure 10: Input Image

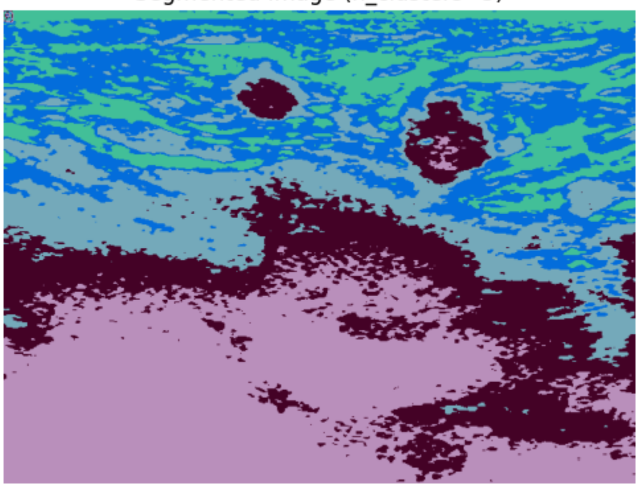
Segmented Image (n_clusters=2)

DBI: 0.43

Segmented Image (n_clusters=3)

DBI: 0.51

Segmented Image (n_clusters=5)



DBI: 0.54

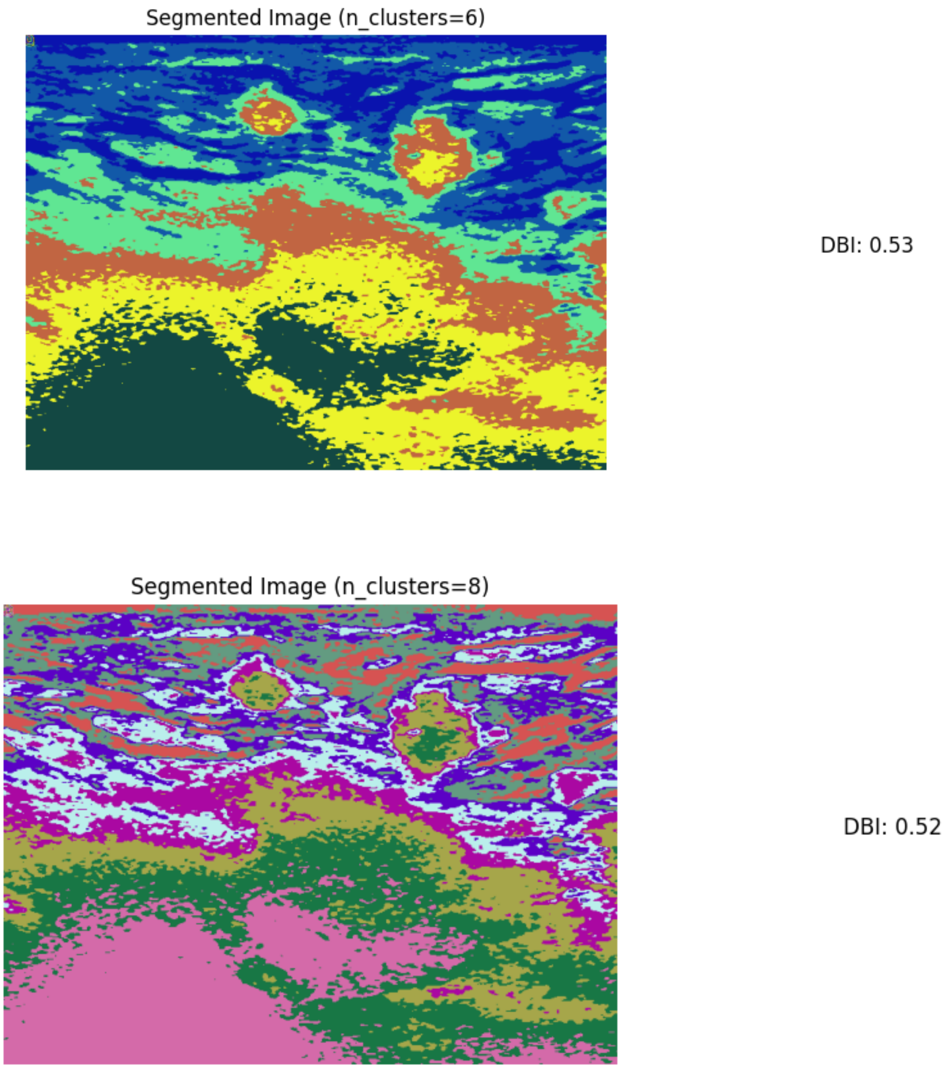


Figure 11: Output Image Segment

Here we can clearly observe that the number of segmentations is directly depended on the number value of cluster(k). More the number of clusters more segmentation and vice versa. The effectiveness of K-means clustering for image segmentation depends on various factors, including the nature of the image, the choice of features, and the selection of the number of clusters.

By varying the number of clusters, the level of detail and granularity in the segmentation results is controlled. It helps a medical person to accurately detects the cancer effectively with the proper segment of the mammographic image.

DBI[7] is a metric used to evaluate the quality of clustering in unsupervised machine learning tasks. It measures the compactness and separation of clusters, helping to determine how well-separated the clusters are and how tight the cluster members are within each cluster. The DBI score is computed as the average similarity between each cluster and its most similar neighboring cluster. A lower DBI score suggests better-defined and well-separated clusters, while a higher score indicates less effective clustering.

Chapter 5: Conclusion and Future Recommendations

5.1 Conclusion

In conclusion, the K-means clustering algorithm has proven to be a valuable and widely used technique for image segmentation in the field of computer vision. It offers a straightforward and intuitive approach to partitioning an image into meaningful regions based on pixel similarity. While the algorithm has its limitations, such as sensitivity to initialization and assumptions of equal-sized clusters, researchers have proposed various enhancements and adaptations to overcome these challenges. Incorporating additional features, spatial information, and hybridizing with deep learning approaches have further improved the segmentation accuracy and robustness. The versatility and effectiveness of the K-means clustering algorithm make it a valuable tool for a medical image analysis. As research in image segmentation continues to evolve, the K-means clustering algorithm serves as a foundational technique that provides insights and paves the way for further advancements in the field.

5.2 Future Recommendations

There are several recommendations to enhance the project on image segmentation using the K-means clustering algorithm. Firstly, integrating spatial information can improve segmentation results by considering neighboring pixel relationships. Secondly, exploring the combination of K-means clustering with deep learning techniques can lead to more accurate and advanced segmentation approaches. Additionally, efficient handling of high-dimensional data, developing robust initialization strategies, and defining comprehensive evaluation metrics can further refine the algorithm's performance. Furthermore, optimizing the algorithm for real-time implementation and improving its robustness to noise and outliers are crucial for practical applications. Addressing these recommendations will contribute to the advancement of image segmentation techniques using the K-means clustering algorithm.

References

[1] D. E. Ilea and P. F. Whelan, "CTex—An Adaptive Unsupervised Segmentation Algorithm Based on Color-Texture Coherence," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1926–1939, 2008, doi: 10.1109/TIP.2008.2001047.

[2] Z. Khan, J. Ni, X. Fan, and P. Shi, "An improved K-means clustering algorithm based on an adaptive initial parameter estimation procedure for image segmentation," *Int. J. Innov. Comput. Inf. Control*, vol. 13, pp. 1509–1526, Oct. 2017.

[3] D. L. Pham, C. Xu, and J. L. Prince, "Current Methods in Medical Image Segmentation," *Annu. Rev. Biomed. Eng.*, vol. 2, no. 1, pp. 315–337, Aug. 2000, doi: 10.1146/annurev.bioeng.2.1.315.

[4] H. P. Ng, S. H. Ong, K. W. C. Foong, P. S. Goh, and W. L. Nowinski, "Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm," in *2006 IEEE Southwest Symposium on Image Analysis and Interpretation*, 2006, pp. 61–65. doi: 10.1109/SSIAI.2006.1633722.

[5] S. N. Sulaiman and N. A. M. Isa, "Adaptive fuzzy-K-means clustering algorithm for image segmentation," *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2661–2668, 2010, doi: 10.1109/TCE.2010.5681154.

[6] D. Sharifrazi *et al.*, "CNN-KCL: Automatic Myocarditis Diagnosis using Convolutional Neural Network Combined with K-means Clustering." {Preprints.org}, 2020. doi: 10.20944/preprints202007.0650.v1.

[7] A. K. Singh, S. Mittal, P. Malhotra, and Y. V Srivastava, "Clustering Evaluation by Davies-Bouldin Index(DBI) in Cereal data using K-Means," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 2020, pp. 306–310. doi: 10.1109/ICCMC48092.2020.ICCMC-00057.