

# Task 4 Report

Computer Vision

Thresholding and Segmentation

## Team Members

Ahmed Abd Elraouf

Zeyad Amr

Mo'men Mohamed

Mazen Tarek

Michael Hany

## Table of content

❖ Introduction .....	3
❖ Technologies .....	4
❖ Thresholding	
• optimal thresholding .....	4
• Otsu .....	6
• spectral thresholding .....	8
• local thresholding .....	10
❖ Segmentation :	
• k-means .....	12
• Region growing .....	14
• Agglomerative .....	16
• Mean shift .....	18

## Introduction

It's a desktop application that aims to involve dividing an image into different regions or objects based on their visual characteristics. In this context, several methods are available to segment images into different regions, each with its own strengths and weaknesses.

In the case of grayscale images, thresholding is a popular method to segment images into different regions. This can be achieved using various thresholding methods such as optimal thresholding, Otsu's method, spectral thresholding, and local thresholding.

In the case of gray/color images, unsupervised segmentation methods are widely used to segment images into different regions. Some of the popular unsupervised segmentation methods include k-means clustering, region growing, agglomerative clustering, and mean shift segmentation. K-means clustering involves clustering pixels based on their similarity, while region growing involves grouping together adjacent pixels that have similar characteristics. Agglomerative clustering merges adjacent clusters based on their similarity, and mean shift segmentation involves shifting a window over the image to group together pixels with similar characteristics.

Overall, each of these methods has its own advantages and disadvantages, and the choice of segmentation method depends on the characteristics of the image and the specific application requirements.

## Technologies

- Desktop application using : qt, cpp
- Processing: cpp, opencv

## Thresholding :

### 1.optimal thresholding :

Optimal thresholding is a method used in image processing to determine the optimal threshold value for binarizing an image. Binarization involves converting a grayscale image into a binary image, where each pixel is either black or white based on its intensity value compared to a threshold value.

The optimal threshold value is the value that maximizes the separation between foreground and background pixels, making it the most appropriate value to use for binarization. There are several methods to find the optimal threshold value, including Otsu's method, the Triangle method, and the Kapur's entropy method.

The choice of which method to use depends on the characteristics of the image being processed, such as its contrast, noise, and distribution of pixel intensities. Optimal thresholding can improve the accuracy and effectiveness of subsequent image analysis tasks, such as segmentation and feature extraction

## Optimal thresholding output:



## 2.Otsu:

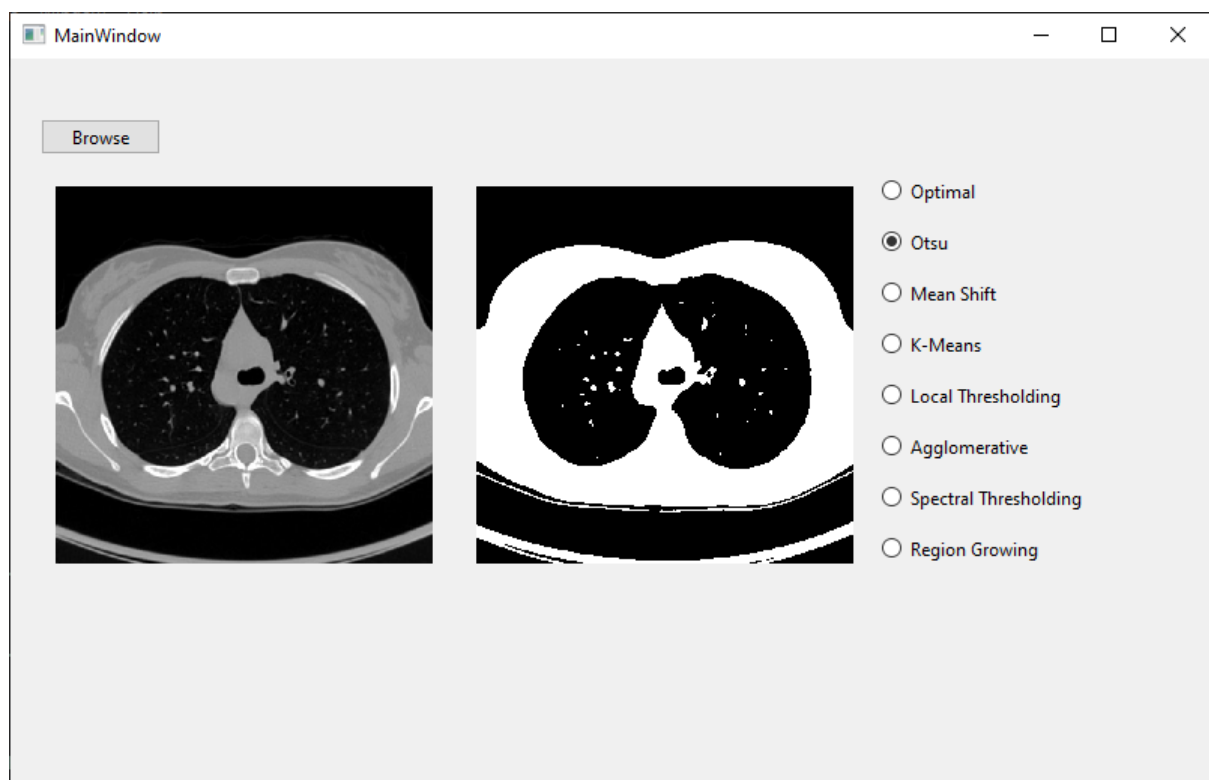
Otsu's method is a popular image thresholding technique that is used to automatically determine the optimal threshold value for binarization of grayscale images. It was developed by Nobuyuki Otsu in 1979 and is widely used in image processing applications.

Otsu's method works by analyzing the histogram of pixel intensities in an image and finding the threshold value that maximizes the separation between the foreground and background pixels. The separation between the two classes of pixels is quantified using the between-class variance, which is defined as the sum of the variances of the two classes weighted by their probabilities.

The algorithm starts by calculating the histogram of the image, which is a count of the number of pixels at each intensity level. It then computes the probabilities of each intensity level and the cumulative probabilities up to each intensity level. Using these values, it computes the between-class variance for each possible threshold value and selects the threshold that maximizes this value.

Otsu's method is effective for images with bimodal intensity distributions, where the foreground and background pixels have distinct peaks in the histogram. However, it may not perform well for images with more complex distributions. In such cases, other thresholding methods, such as the adaptive thresholding, may be more appropriate.

## Otsu output:



### **3.Spectral thresholding:**

Spectral thresholding is a method used in image processing and computer vision to segment an image into different regions based on the spectral characteristics of the image. In particular, it involves using the spectral properties of an image to identify areas of interest and separate them from the background.

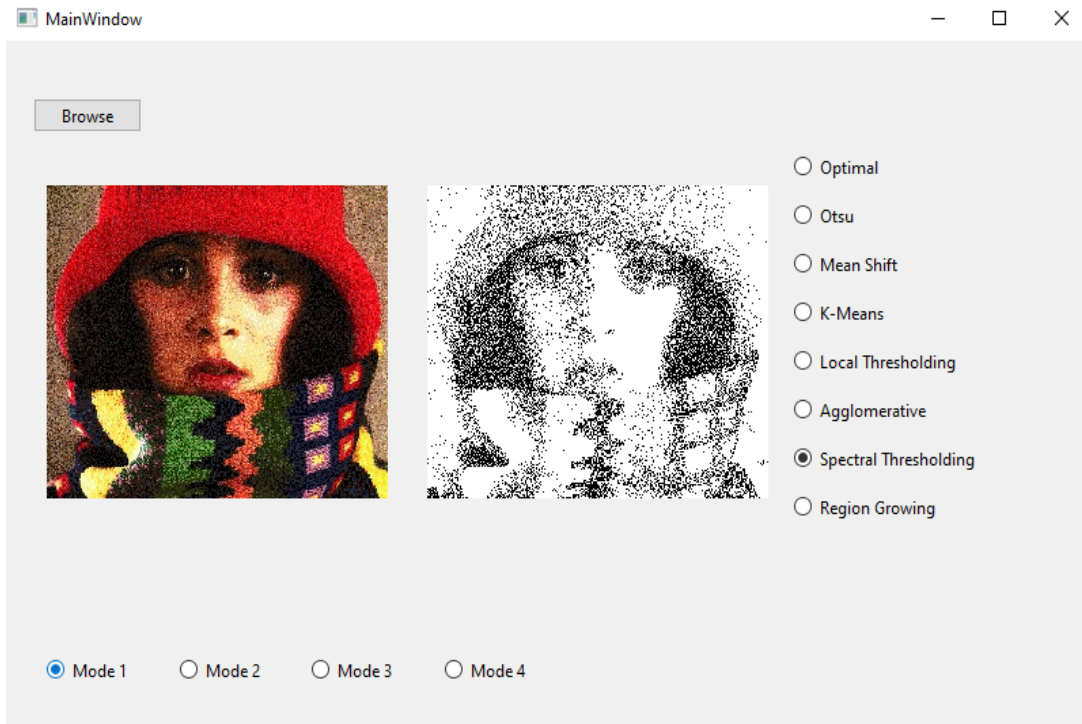
The term "spectral" refers to the different frequency components of an image, which can be represented by the Fourier transform or other frequency-based methods. In spectral thresholding, these frequency components are used to distinguish between different regions of the image.

The process of spectral thresholding involves computing the spectral features of an image, such as the Fourier transform or wavelet transform, and then selecting a threshold value to separate the regions of interest from the background. The threshold value is chosen based on the statistical properties of the spectral features, such as their mean and variance.

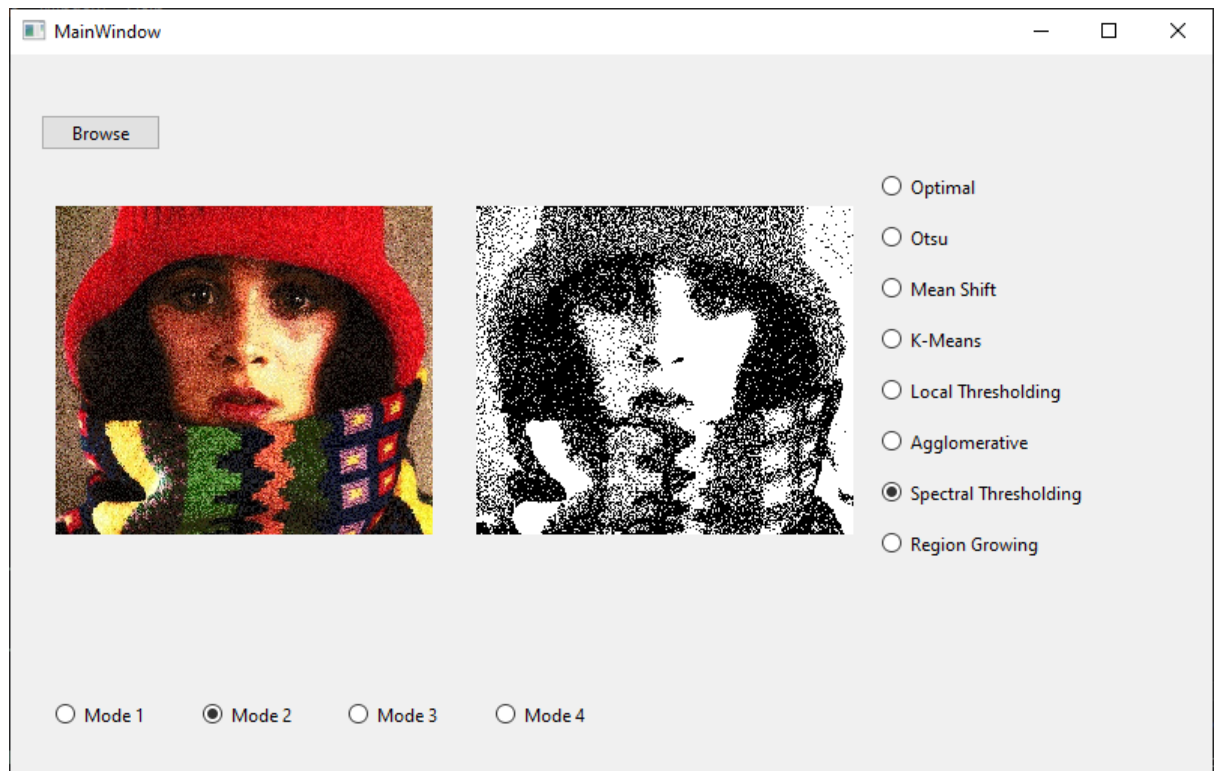
Spectral thresholding can be useful for segmenting images in applications such as medical imaging, remote sensing, and computer vision. However, it can be computationally intensive and may not be suitable for real-time applications or large datasets. Other thresholding methods, such as Otsu's method or adaptive thresholding, may be more appropriate in such cases.

### **Spectral thresholding output:**

## Mode 1:

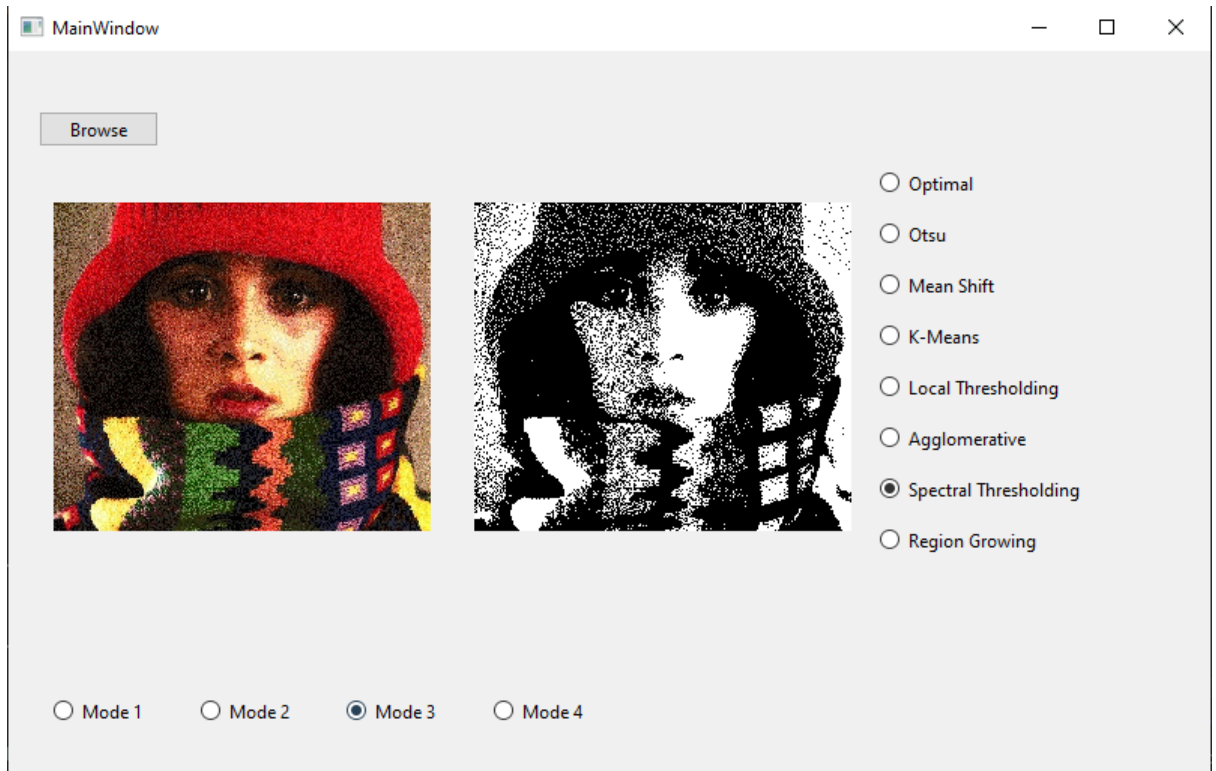


## Mode 2:

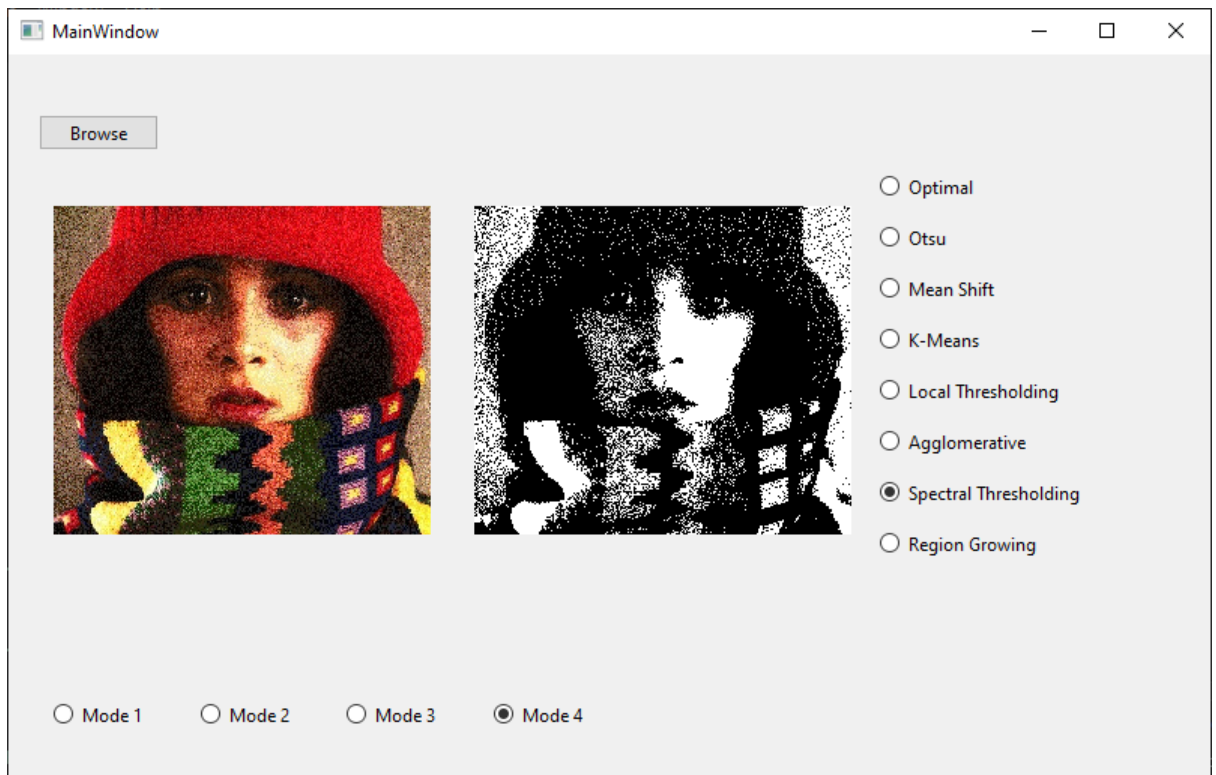




## Mode 3:



## Mode 4:



## 4. Local thresholding

Local thresholding is a method used in image processing and computer vision to segment an image into different regions based on the local properties of the image. In particular, it involves using a different threshold value for each pixel or local neighborhood of pixels based on their local intensity characteristics.

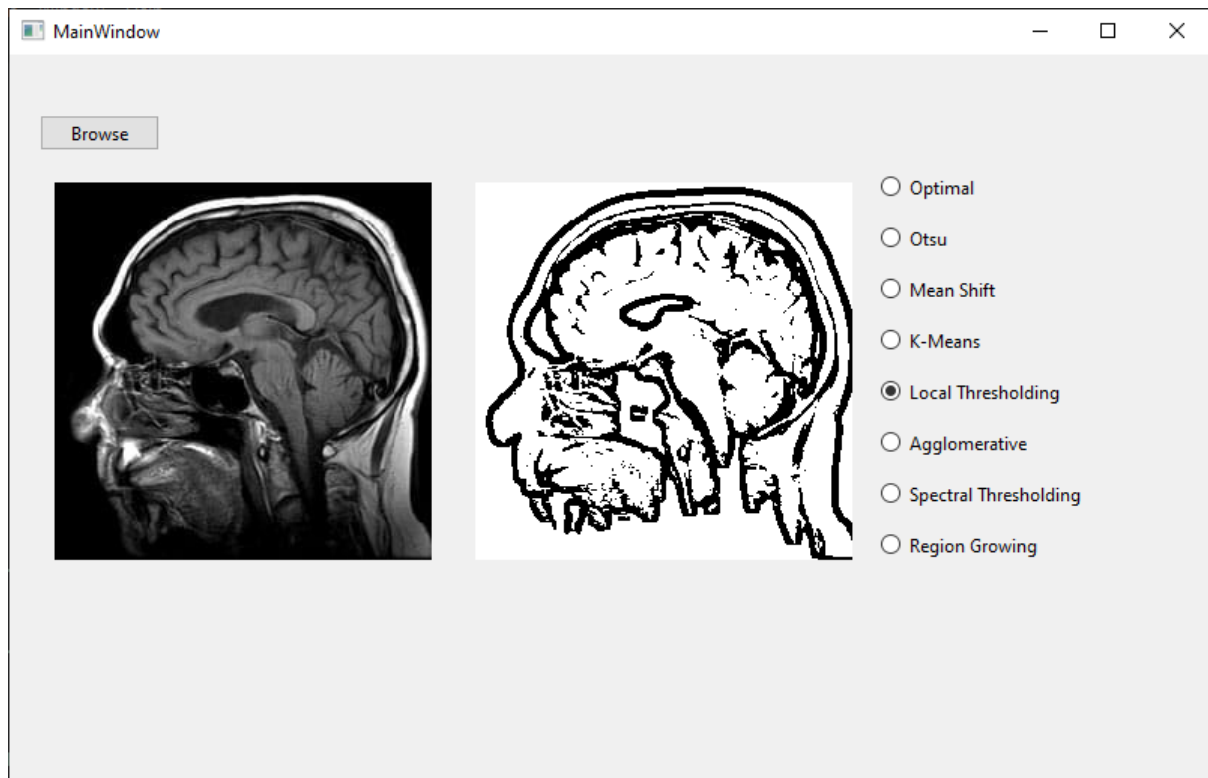
Local thresholding is often used when the global thresholding methods, such as Otsu's method or spectral thresholding, fail to provide satisfactory results. This is often the case when the image has non-uniform illumination or when the foreground and background regions have different contrast levels.

The process of local thresholding involves computing a threshold value for each pixel or a local neighborhood of pixels based on its local intensity characteristics, such as its mean or variance. The threshold value is then applied to the corresponding pixel or pixels to obtain a binary image.

There are several methods for local thresholding, including the Niblack method, the Sauvola method, and the Bradley method. These methods differ in their approach to computing the local threshold value, but they all aim to adapt the threshold to the local properties of the image.

Local thresholding can be useful for segmenting images in applications such as document processing, character recognition, and defect detection. However, it can also be computationally intensive and may require fine-tuning of the parameters to achieve satisfactory results.

## Local thresholding output:



## Segmentation :

### 1. k-means :

Segmentation using k-means is a method used in image processing and computer vision to segment an image into different regions based on the similarity of their color or intensity values. It involves clustering the pixels in an image into  $k$  clusters based on

their color or intensity values and then assigning each pixel to the cluster with the closest color or intensity value.

The k-means algorithm is an unsupervised machine learning technique that iteratively partitions the data points into k clusters based on their distance to the cluster centers. In image segmentation, each pixel is treated as a data point with its color or intensity values as the feature vector.

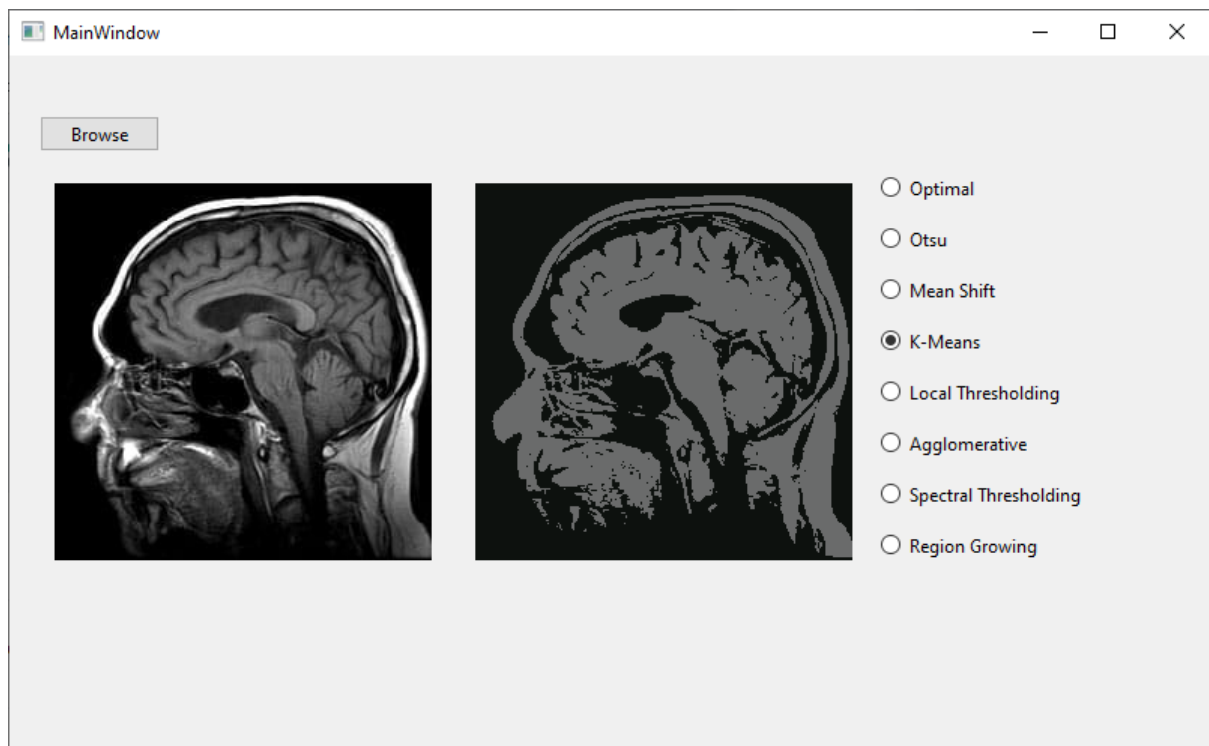
The process of segmentation using k-means involves the following steps:

1. Initialize k cluster centers randomly
2. Assign each pixel to the cluster with the closest color or intensity value
3. Recalculate the cluster centers based on the mean color or intensity value of the pixels in each cluster
4. Repeat steps 2 and 3 until convergence is reached or a maximum number of iterations is reached

The number of clusters k is typically chosen based on the characteristics of the image and the segmentation task. For example, k may be set to the number of regions or objects of interest in the image.

Segmentation using k-means can be effective for images with distinct regions or objects that have different color or intensity values. However, it may not perform well for images with complex textures or patterns or when the objects of interest have similar color or intensity values. In such cases, other segmentation methods, such as graph-based or region-based segmentation, may be more appropriate.

## k-means output:



## 2. region growing :

Segmentation using region growing is a method used in image processing and computer vision to segment an image into different regions based on the similarity of neighboring pixels. It involves

grouping together adjacent pixels that have similar characteristics, such as color or intensity, to form a region.

The region growing algorithm starts with an initial seed pixel or set of seed pixels that belong to a particular region. The algorithm then iteratively adds neighboring pixels to the region if they meet certain criteria, such as having similar color or intensity values. The process continues until no more pixels can be added to the region or a stopping criterion is reached.

The process of segmentation using region growing involves the following steps:

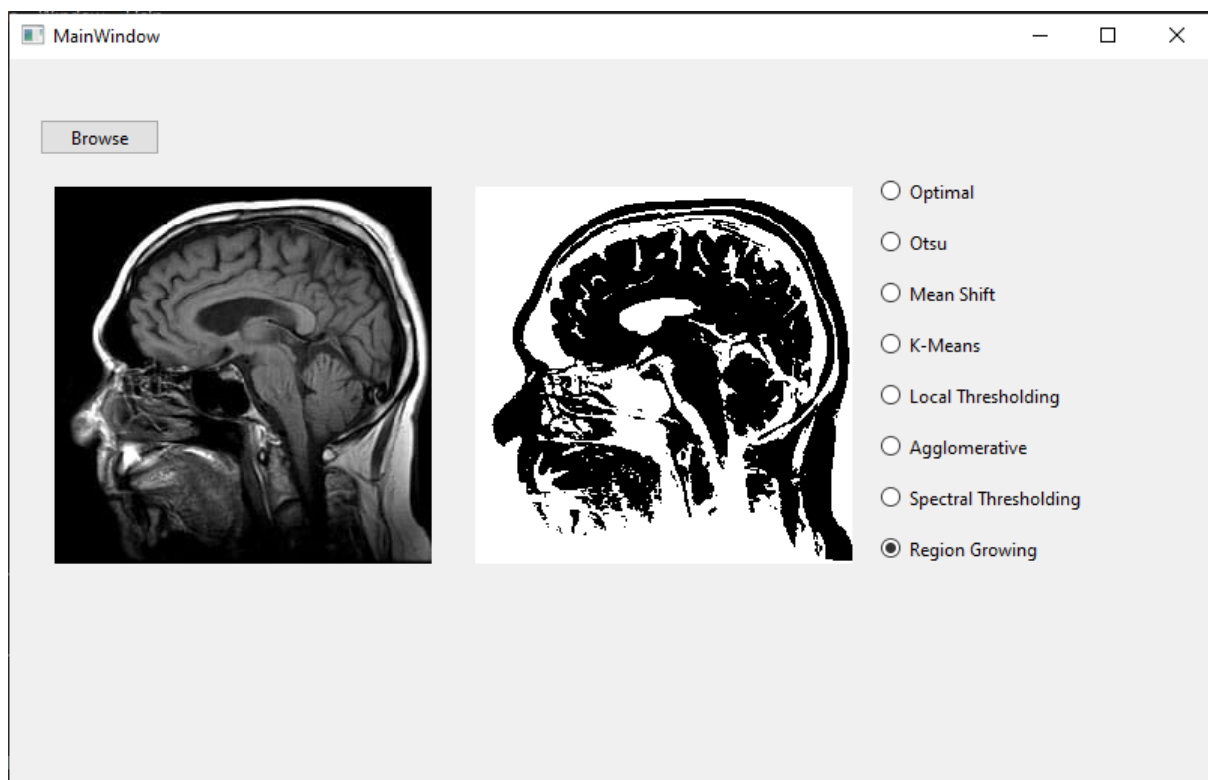
1. Select one or more seed pixels that belong to a region of interest.
2. Define a similarity criterion, such as the difference in color or intensity values between neighboring pixels.
3. For each seed pixel, examine its neighboring pixels and compare their similarity to the seed pixel based on the similarity criterion.
4. If a neighboring pixel satisfies the similarity criterion, add it to the region and consider its neighboring pixels in the next iteration.
5. Repeat steps 3 and 4 until no more pixels can be added to the region or a stopping criterion is reached.

The stopping criterion can be based on various factors, such as the size of the region, the number of iterations, or a threshold value for the similarity criterion.

Segmentation using region growing can be effective for images with distinct regions or objects that have similar color or intensity values. However, it may not perform well for images with complex

textures or patterns or when the objects of interest have different colors or intensities. In such cases, other segmentation methods, such as k-means clustering or graph-based segmentation, may be more appropriate.

## region growing output:



### 3. agglomerative :

Segmentation using agglomerative clustering is a method used in image processing and computer vision to segment an image into different regions based on the similarity of neighboring pixels. It involves grouping together adjacent pixels based on their

similarity, and progressively merging these groups until a specified number of regions is obtained.

The agglomerative clustering algorithm starts by treating each pixel in the image as a separate cluster. The algorithm then iteratively merges the two closest clusters based on a similarity measure, such as the Euclidean distance between their color or intensity values. The process continues until the desired number of clusters or regions is obtained.

The process of segmentation using agglomerative clustering involves the following steps:

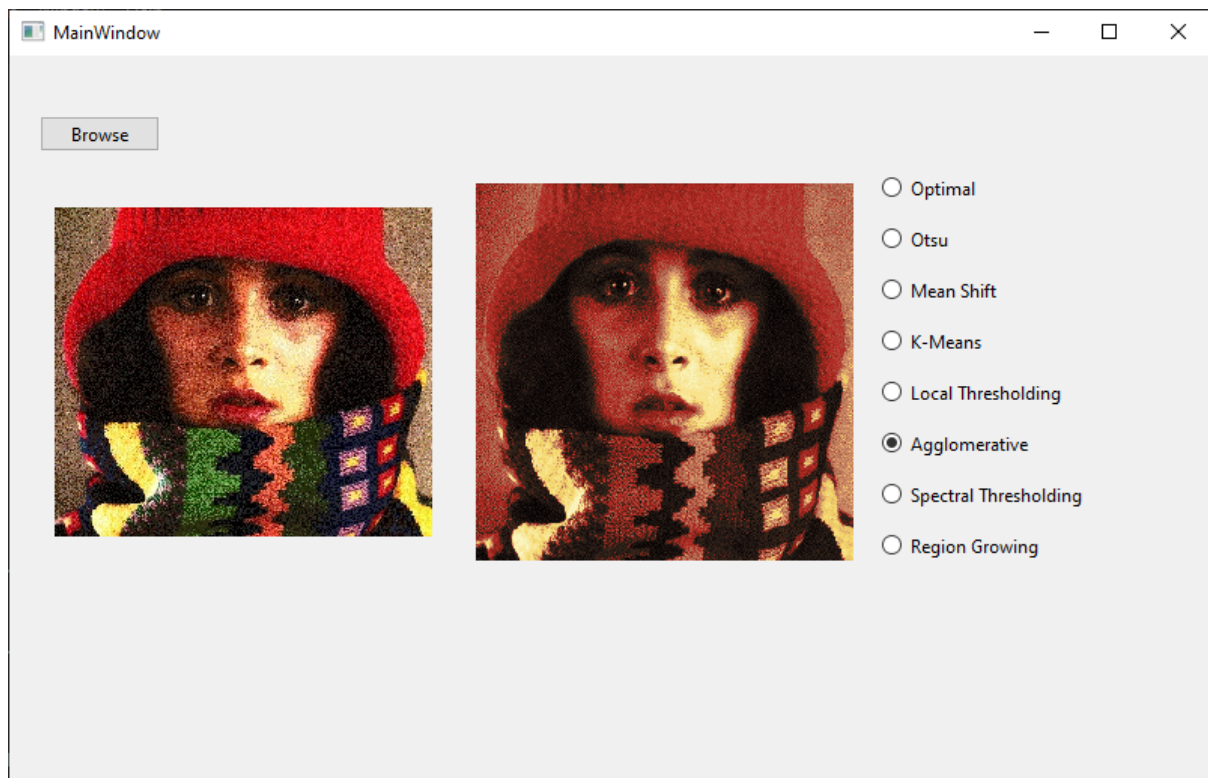
1. Treat each pixel in the image as a separate cluster.
2. Define a similarity measure between clusters, such as the Euclidean distance between their color or intensity values.
3. Merge the two closest clusters based on the similarity measure.
4. Recalculate the similarity measure between the new cluster and the remaining clusters.
5. Repeat steps 3 and 4 until the desired number of clusters or regions is obtained.

The desired number of clusters can be specified in advance or determined automatically using methods such as the elbow method or silhouette analysis.

Segmentation using agglomerative clustering can be effective for images with distinct regions or objects that have different color or intensity values. However, it may not perform well for images with complex textures or patterns or when the objects of interest have similar colors or intensities. In such cases, other segmentation methods, such as k-means clustering or region growing, may be more appropriate.



## Agglomerative output:



## 4. Mean shift :

Segmentation using mean shift is a method used in image processing and computer vision to segment an image into different regions based on the similarity of neighboring pixels. It involves

shifting a window over the image and iteratively updating the position of the window center to the mean of the pixels within the window until convergence is reached.

The mean shift algorithm starts with a window centered at a pixel of interest. The algorithm then computes the mean color or intensity values of the pixels within the window and shifts the window center to the location of the mean. This process is repeated until convergence is reached, which occurs when the window center no longer moves or moves by less than a specified threshold.

The process of segmentation using mean shift involves the following steps:

1. Choose a pixel of interest and a window size.
2. Compute the mean color or intensity values of the pixels within the window.
3. Shift the window center to the location of the mean.
4. Repeat steps 2 and 3 until convergence is reached.

Once the mean shift process is complete, each pixel in the image is assigned to the cluster associated with the window center it falls within. The result is a segmentation of the image into different regions based on the similarity of neighboring pixels.

Segmentation using mean shift can be effective for images with distinct regions or objects that have different color or intensity values. It can also handle images with complex textures or patterns. However, it may not perform well for images with very large or very small regions or when the objects of interest have similar colors or intensities. In such cases, other segmentation methods, such as

agglomerative clustering or region growing, may be more appropriate.

## Mean shift output:

