Faculty of Engineering
Cairo University

Computer Vision Task 3

NAME	SECTION	BN
Ahmed Hassan	1	4
Dina Khalid	1	29
Abdelrahman Yasser	1	52
Mohamed Nasser	2	21
Mahmoud Rabea	2	25

Table of contents

Segmentation

1.Region Growing:

The basic idea behind region growing is to start with a single pixel or a group of pixels, called seeds, and then grow the region by adding neighboring pixels that are similar to the seeds based on certain criteria. This process continues until all pixels in the image have been assigned to a region.

Advantages:

- Region growing is a simple and efficient algorithm that can produce accurate results in many cases.

- It is relatively easy to implement and can handle noise

Disadvantages:

- Sensitivity to the choice of seed points.

- may not work well on images with complex textures.

Steps:

1. Convert the input image to the LAB color space.

2. Create a blank mask of the same size as the input image to hold the segmented region.

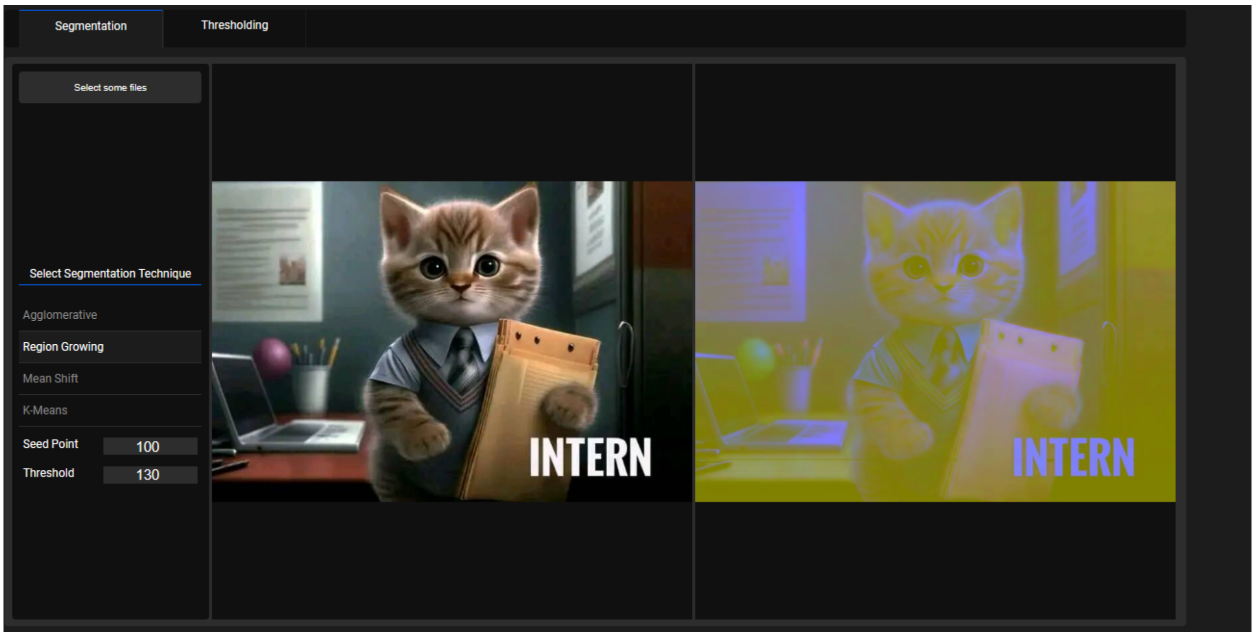
3. Set the seed point value as the initial region value by extracting the pixel values at the given seed point from the LAB color space image.

4. Define a neighborhood of 4-connected pixels to use for region growing. The neighborhood consists of the pixels to the top, bottom, left, and right of the current pixel.

5. Loop until the region stops growing:

6. Convert the mask to a 3-channel image and apply it to the input image to create the segmented output image.

7. Convert the output image to the LAB color space and save it.



2.Agglomerative:

It's a clustering technique that works by iteratively merging pairs of adjacent data points or clusters based on their distance until all the data points or clusters are grouped into a single cluster.

Advantages:

- Works well with a small number of clusters.

Disadvantages:

- It is computationally expensive for large datasets and may not scale well.

-Sensitive to the initial distance metric.

Steps:

1. Assign each pixel in the image to its own separate segment.
2. Compute a similarity measure between all pairs of adjacent segments.
3. Merge the two most similar segments into a new segment.
4. Recalculate the similarity between the new segment and its neighbors.
5. Repeat steps 3 and 4 until a stopping criterion is met.
6. Return segmented image in LAB color space.

3.K-means:

It’s an unsupervised clustering technique used for segmenting images into distinct regions based on pixel intensities. It works by grouping pixels into K clusters, where K is the user-defined number of clusters.

Advantages:

- K-means is a fast and efficient algorithm that is easy to implement.

- It can handle large datasets and is scalable.

Disadvantages:

- The quality depends on the number of clusters.

- K-means assumes that the data points are independent and identically distributed, which may not always be true.

Steps:

The function takes three arguments: the number of clusters to use, the input image, and the maximum number of iterations to run and returns the segmented image by following these steps:

1. Initialize centroids randomly.

2. Assign points to the nearest centroid.

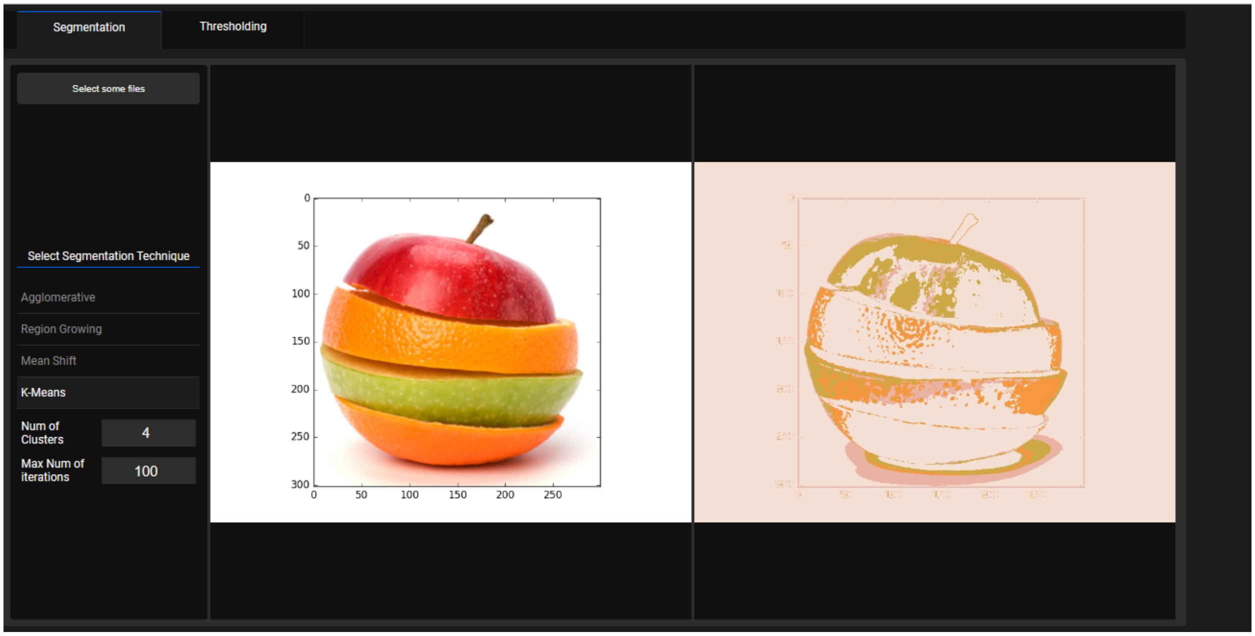
3. Update centroids by computing the mean of the data points in each cluster.

4. Repeat steps 2 and 3 until convergence or the maximum number of iterations is reached.

5. Reshape the input image into a list of pixels.

6. Cluster the pixels using the k-means algorithm and assign labels to each pixel.

7. Convert the label image to color using the centroid colors and save it.



4.Mean shift:

Mean shift is a non-parametric clustering algorithm. It works by identifying local maxima in the density of a data distribution and shifting each data point towards the nearest maximum until convergence.

Advantages:

- It is robust to noise and outliers.

- The number of clusters is automatically determined based on the data.

Disadvantages:

- Mean shift is computationally expensive, particularly for large datasets.

- The algorithm can converge to a suboptimal solution depending on your bandwidth choice.

Steps:

The function takes two arguments: the bandwidth and the input image and returns the segmented image by following these steps:

1. The input image in BGR format is loaded.

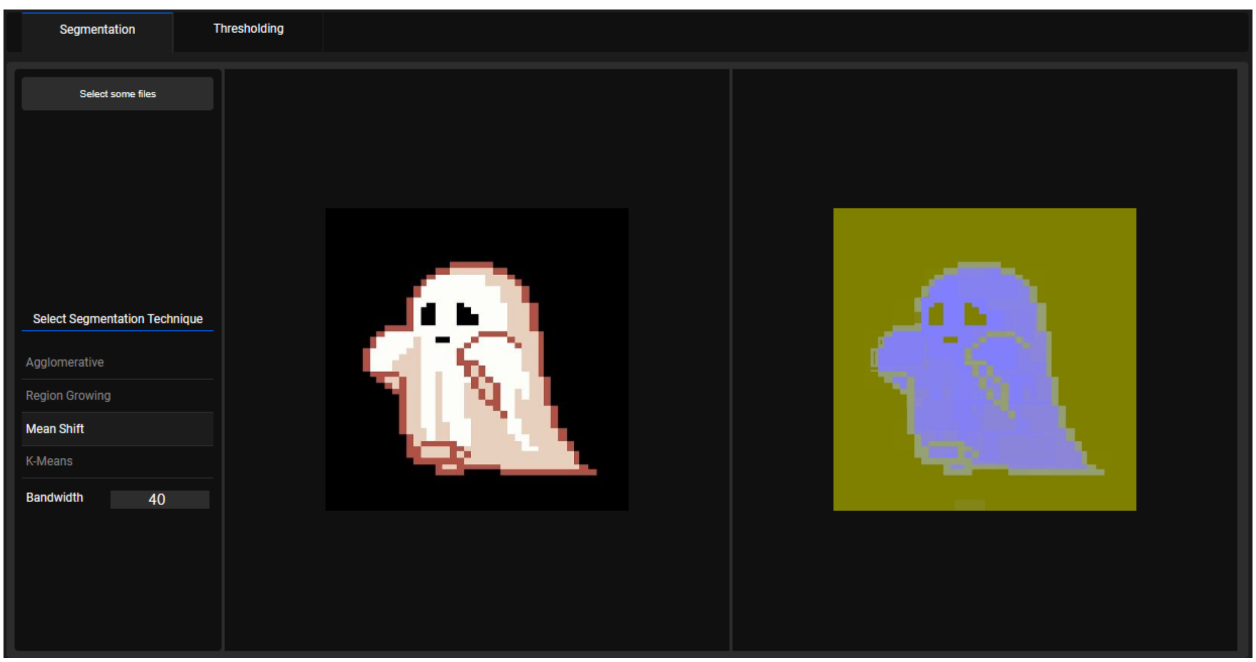
2. The input image is converted from BGR to LAB color space.

3. For each pixel, the mean shift vector is computed by finding the pixels within a certain bandwidth and taking the average of their LAB values.

4. Mean shift labels are assigned to each pixel based on their mean shift vector.

5. The mean shift labels are converted to a color image using the LAB color space and saved.

Note that Large bandwidth will take less time but will result in a smoother image while small bandwidth will take more time and give a more detailed image.



Thresholding

There are two types of thresholding methods: local thresholding and global thresholding.

Local thresholding calculates the threshold value for each pixel by analyzing a small neighborhood around it. The threshold value is then used to classify the pixel as foreground or background.

Global thresholding calculates a single threshold value for the entire image. Pixels with intensity values above the threshold are classified as foreground, and those below are classified as background

Advantages of Local thresholding:

1. Adaptive and can handle images with varying lighting conditions.

2. Effective in preserving the details and boundaries of the image

Advantages of Global thresholding:.

1. simple and computationally efficient..

2. Works well for images with uniform lighting and good contrast between the foreground and background.

Example of default local thresholding:

1.Optimal:

Thresholding technique that automatically determines the optimal threshold value by maximizing the between-class variance.

Advantages:

- Can be applied to images with bimodal histograms.
- Computationally efficient.

Disadvantages:

- May not work well with images with non-unimodal histograms.
- Sensitivity to noise and background variation.

Steps:

1. Compute the maximum number of rows and columns for the image

2. Calculate the mean value of background intensity using the four corner pixels of the image

3. Calculate the mean value of foreground intensity by iterating over all non-corner pixels of the image and returns the new image

LOCAL:

GLOBAL:

2.Spectral:

It is based on analyzing the frequency content of the image, and it is particularly useful for images with uneven illumination and noise. The algorithm works by applying a filter to the Fourier transform of the image, then applying an inverse Fourier transform to obtain the thresholded image.

Advantages:

- Effective for images with non-uniform illumination or low contrast.

- Can be used for both grayscale and color images.

Disadvantages:

- Computationally expensive.

- The number of clusters needs to be specified in advance, which may require some prior knowledge about the image.

Steps:

1. The method takes an image and a size parameter as input.

2. It iterates through the image in size intervals to process sub-images.

3. It extracts a sub-image of size ‘size’ at a time using the numpy indexing operation.

4. The sub-image is passed to the spectral_threshold() method to get a binary image.

5. The binary image replaces the corresponding portion of the original image.

6. Once all sub-images are processed, the method returns the thresholded image.

LOCAL:

GLOBAL:

3.Otsu:

It’s a thresholding algorithm used for image segmentation by determining the optimal threshold value by maximizing the variance between the two classes of pixels.

Advantages:

- It can work well for bimodal and multimodal images.

- It is computationally efficient and simple to implement.

Disadvantages:

- It may not work well for images with uneven lighting or uneven contrast.

- It assumes that the foreground and background pixels have distinct intensity levels, which may not always be the case.

Steps:

1. Initialize an empty list to store the variance between classes for each threshold value.

2. Loop over all possible threshold values from the minimum to maximum pixel intensity values in the image.

3. Compute the pixel intensities that are below and above the current threshold.

4. Compute the probabilities of the pixel intensities below and above the current threshold respectively.

5. Compute the variance of the pixel intensities below and above the current threshold respectively.

6. Compute the variance between classes and append it to the list.

7. Find the minimum variance value in the list.

8. Get the index of the minimum variance value in the list as the threshold value.

9. Return the thresholded image.

LOCAL:

GLOBAL: