A Survey on Image Segmentation

SIRG
Scientific Innovation Research Group

Lecture 3

Computer Vision: Algorithms and Applications

Supervised by

Dr. Ahmed Elngar
Faculty of Computers and Artificial Intelligence
Beni-Suef University

Outline

- Definition
- Mechanism
- Types and comparison
- Mathematical Formulation & Example
- Advantages & Disadvantages
- Conclusion
- References

What is Image Segmentation?

- **Image segmentation** is the process of dividing a digital image into multiple parts called segments (objects).
- **Segment** is sets of pixels.
- **The goal of segmentation** is to simplify the image into an image that more meaningful and easier to analyze.

What is Image Segmentation?

- The purpose of image segmentation is to partition an image into *meaningful* regions with respect to a particular application
- The segmentation is based on measurements taken from the image and might be *greylevel*, *colour*, *texture*, *depth* or *motion*

What is Image Segmentation?

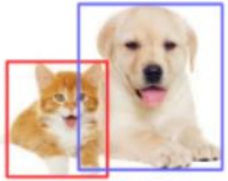
- Usually image segmentation is an initial and vital step in a series of processes aimed at overall image understanding
- Applications of image segmentation include
  - Identifying objects in a scene for object-based measurements such as size and shape
  - Identifying objects in a moving scene for *object-based video compression (MPEG4)*
  - Identifying objects which are at different distances from a sensor using depth measurements from a laser range finder enabling path planning for a mobile robots

What is Image Segmentation?

- There’s only one object here – a dog. We can build a simple **cat-dog classifier model** and predict that there’s a dog in the given image.

- In case we have multiple objects present, we then rely on the concept of **Object Detection (OD).**

Image Localization



Object Detection

How does image segmentation work?

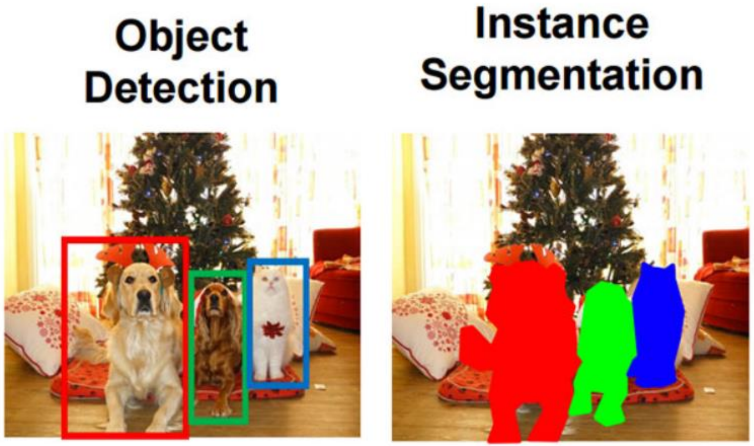
- We can divide the image into various parts called **segments**. It's not a great idea to process the entire image at the same time as there will be regions in the image which do not contain any information. By dividing the image into segments, we can use the important segments for processing the image.

- **An image** is a collection or set of different pixels. We group together the pixels that have similar values using image segmentation.

USAGE

- **Object detection** builds a bounding box corresponding to each object in the image. But it tells us nothing about the shape of the object.

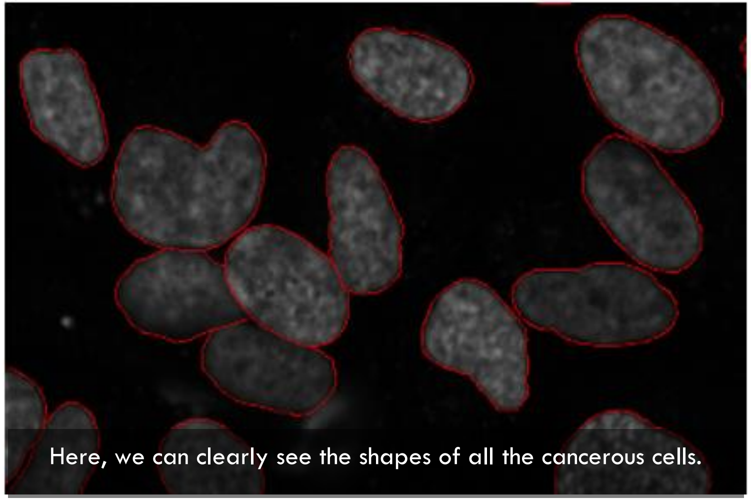
- **Image segmentation** creates a pixel-wise mask for each object in the image. This technique gives us more information about the image.



Why do we need Image Segmentation?

- Cancer has long been a deadly illness. Even in today's age of technological advancements, cancer can be fatal if we don't identify it at an early stage. Detecting cancerous cell(s) as quickly as possible can potentially save millions of lives.

- **Object detection** will not be very useful here. We will only generate bounding boxes which will not help us in identifying the shape of the cells.
- **Image Segmentation** make a MASSIVE impact here. They help us approach this problem in a more meaningful results.



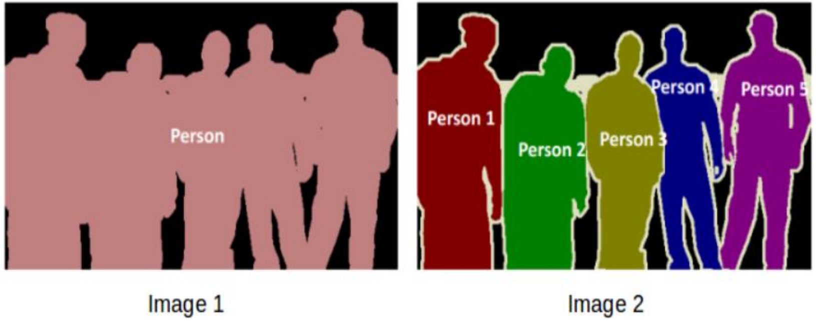
What are Types of image segmentation?

Types of image segmentation

- Semantic segmentation
- Instance segmentation

Types of image segmentation

- Both the images are using image segmentation to identify and locate the people present:
  - **In image 1**, every pixel belongs to a particular class (either background or person). Also, all the pixels belonging to a particular class are represented by the same color (background as black and person as pink). This is an example of semantic segmentation
  - **Image 2** has also assigned a particular class to each pixel of the image. However, different objects of the same class have different colors (Person 1 as red, Person 2 as green, background as black, etc.). This is an example of instance segmentation



Classification of image segmentation techniques:

A. Structural Segmentation Techniques :

- There are techniques of image segmentation that dependent on the information of the structure of required portion of the image. the required region which is to be segmented.

B. Stochastic Segmentation Techniques:

- There are techniques of the image segmentation which his work dependent on the discrete pixel values of the image instead of the structural information of region.

C. Hybrid Techniques:

- There are techniques of the image segmentation that uses the concepts of both techniques. these uses discrete pixel and structural information together.

What is image segmentation Methods?

What is image segmentation Methods?

- Threshold Method
- Edge Based Method
- Region Based Method
- Clustering Based Method
- Watershed Based Method

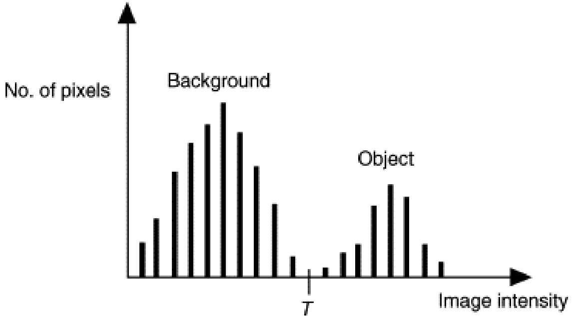
Thresholding Method

- **Thresholding methods are the simplest methods for image segmentation.** These methods divide the image pixels with respect to their intensity level. These methods are used over images having lighter objects than background. The selection of these methods can be manual or automatic i.e. can be based on prior knowledge or information of image features. There are basically three types of thresholding:

1) Global Thresholding: This is done by using any appropriate threshold value T. This value of T will be constant for whole image.

  - If pixel intensities vary between 0 and 255. Then the threshold T = 127 was selected as the minimum between two modes on a histogram.
  - On the basis of T the output image can be obtained from original image as:

$$q(x,y) = \begin{Bmatrix} 1, if\ p(x,y) > T \\ 0, if\ p(x,y) \le T \end{Bmatrix}$$



Thresholding Method (cont.)

2) Variable Thresholding: In this type of thresholding, the value of T can vary over the image. This can further be of two types:

- **Local Threshold:** chooses different threshold values for every pixel in the image based on an analysis of its neighboring pixels. This is to allow images with varying contrast levels where a global thresholding technique will not work satisfactorily.
- **Adaptive Threshold:** if an image has different lighting conditions in different areas. In that case, adaptive thresholding can help. determine the threshold for a pixel based on a small region around it. So we get different thresholds for different regions of the same image which gives better results for images with varying illumination . The threshold value is the mean of the neighbourhood area minus the constant C

3) Multiple Thresholding: In this type of thresholding, there are multiple threshold values like T0 and T1. By using these output image can be computed as:

$$q(x,y) = \begin{Bmatrix} m, if\ p(x,y) > T1 \\ n, if\ p(x,y) \le T1 \\ o, if\ p(x,y) \le T0 \end{Bmatrix}$$

The values of thresholds can be computed with the help of the peaks of the image histograms. Simple algorithms can also be generated to compute these.

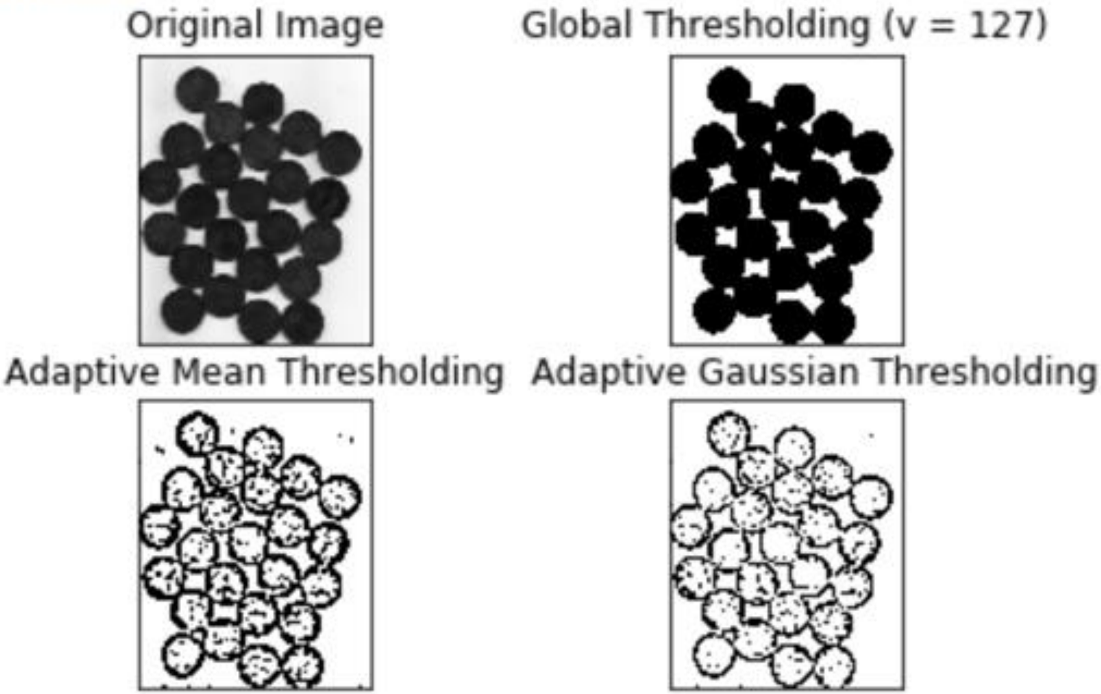
Thresholding Method (cont.)

Code

```python
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('beach.png',0)
img = cv2.medianBlur(img,5)

ret,th1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
th2 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_MEAN_C,\
            cv2.THRESH_BINARY,11,2)
th3 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,\
         cv2.THRESH_BINARY,11,2)

titles = ['Original Image', 'Global Thresholding (v = 127)',
            'Adaptive Mean Thresholding','     Adaptive Gaussian Thresholding']
images = [img, th1, th2, th3]

for i in range(4):
    plt.subplot(2,2,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([]),plt.yticks([])
plt.show()
```

Thresholding Method (cont.)

Implementation



Edge Based Segmentation Method

- The edge detection techniques are well developed techniques of image processing on their own.
- The edge based segmentation methods are based on the rapid change of intensity value in an image because a single intensity value does not provide good information about edges.
- Edge detection techniques locate the edges where either the first derivative of intensity is greater than a particular threshold.
- In edge based segmentation methods, first of all, the edges are detected and then are connected together to form the object boundaries to segment the required regions.
- The basic two edge based segmentation methods are: Gray histograms and Gradient based methods.
- To detect the edges one of the basic edge detection techniques like sobel operator, canny operator and Robert‟s operator etc can be used.
- Result of these methods is basically a binary image.
- These are the structural techniques based on discontinuity detection.

Edge Based Segmentation Method Cont..

Gradient Operators

a
b c
d e
f g

FIGURE 10.8 A 3 × 3 region of an image (the z's are gray-level values) and various masks used to compute the gradient at point labeled z_5.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

DA-IICT

Edge Based Segmentation Method Cont..

- Mathematical Formulation of Gradient Operators of an image f(x,y) is:

$$|\nabla f(x,y)| = \left\{[f(x,y) - f(x+1,y+1)]^2 + [f(x+1,y) - f(x,y+1)]^2\right\}^{\frac{1}{2}}$$
$$\approx |f(x,y) - f(x+1,y+1)| + |f(x+1,y) - f(x,y+1)|$$

- And For Laplacian operator we can use filters as

0	-1	0
-1	4	-1
0	-1	0

4-neighbourhoods

-1	-1	-1
-1	8	-1
-1	-1	-1

8-neighbourhoods

- With Mathematical Formulation of image f(x,y) for laplacian is:

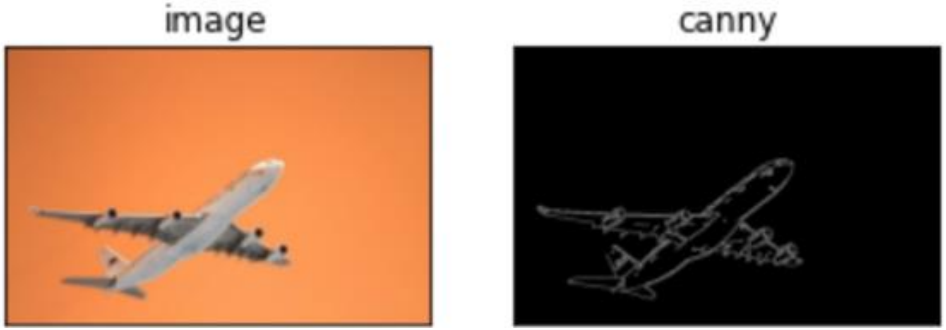
$$\nabla^2 f(x,y) = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

Edge Based Segmentation Method Cont..

Code

```
import numpy as np
from matplotlib import pyplot as plt
import cv2 as cv

img = cv.imread('C:\city.jpeg')
canny = cv.Canny(img, 100, 200)
titles = ['image', 'canny']
images = [img, canny]
for i in range(2):
    plt.subplot(1, 2, i + 1)
    plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([]), plt.yticks([])
plt.show
```

Implementation

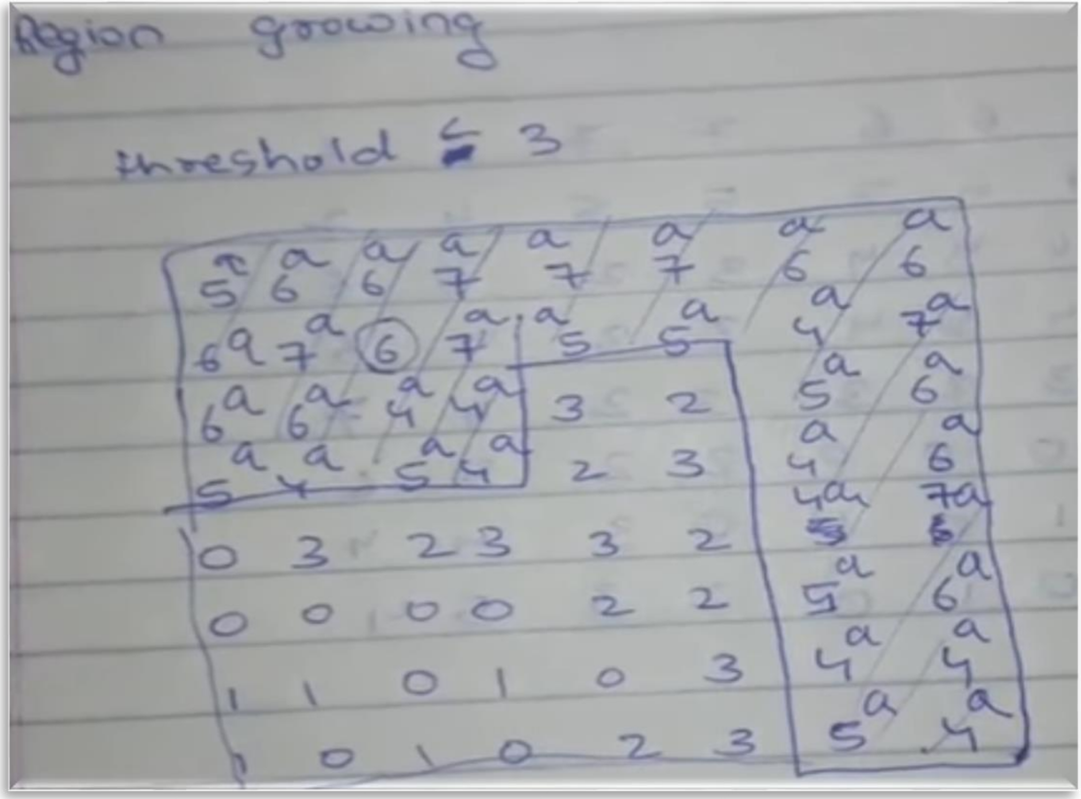


Region Based Segmentation Method

- The region-based segmentation methods are the methods that **segments the image into various regions having similar characteristics.**
- There are two basic techniques based on this method:
  1. **Region growing methods:** The region growing based segmentation methods are the methods that segments the image into various regions based on the growing of **seeds** (initial pixels). These **seeds** can be selected **manually** (based on prior knowledge) or **automatically** (based on particular application). we make these steps:
     - Calculate the mean of image pixels which integer value will be the **thresholds** of operation (T).
     - Select the nearly value of peak (bigger) value of pixels as P.
     - Subtract the value of P from neighbors and check if result < threshold(T) then the pixel belong to the region and put pixel value =1. Else this pixel is not belongs to region and put pixel value=0.
     - Repeat this operation on all pixels.

Region Based Segmentation Method Cont..

β)	1	2	3	4	5	6	7
1	β	β	α	α	α	α	α
2	β	β	β	α	α	α	β
3	γ	β	β	β	β	β	β
4	γ	γ	β	β	β	β	γ
5	γ	γ	γ	α	γ	γ	γ
6	γ	γ	α	γ	α	γ	γ
7	α	α	α	α	α	γ	γ



Region Based Segmentation Method Cont..

Code

```
import numpy as np
import cv2
from matplotlib import pyplot as plt


class Point(object):
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def getX(self):
        return self.x

    def getY(self):
        return self.y


def getGrayDiff(img, currentPoint, tmpPoint):
    return abs(int(img[currentPoint.x, currentPoint.y]) - int(img[tmpPoint.x,
```

Region Based Segmentation Method Cont..

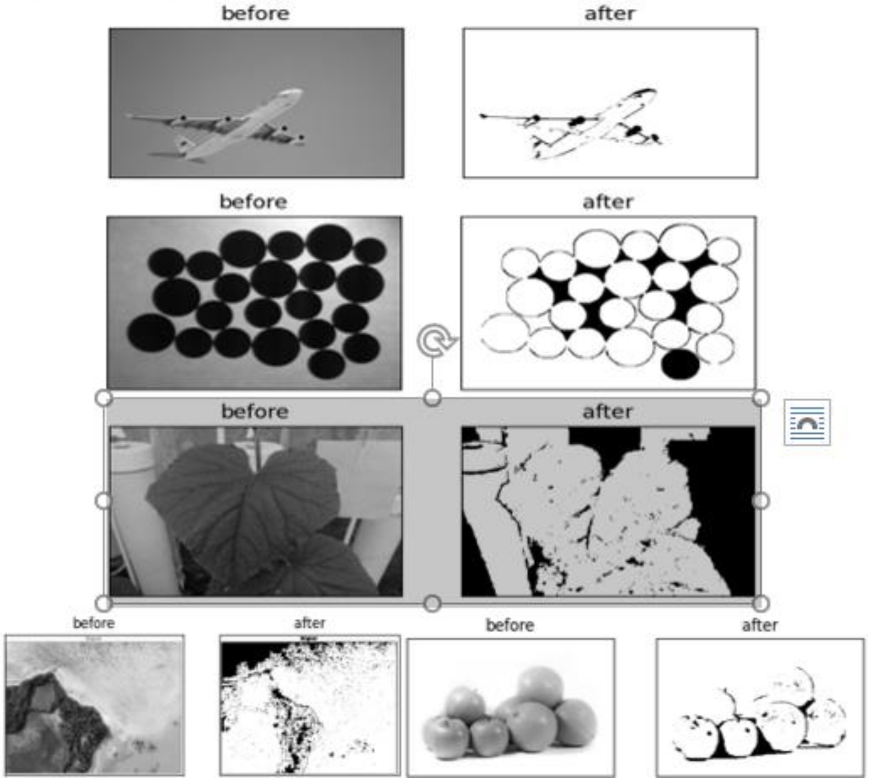

```
tmpPoint.y]))


def selectConnects(p):
    if p != 0:
        connects = [Point(-1, -1), Point(0, -1), Point(1, -1), Point(1, 0),
Point(1, 1), \
                    Point(0, 1), Point(-1, 1), Point(-1, 0)]
    else:
        connects = [Point(0, -1), Point(1, 0), Point(0, 1), Point(-1, 0)]
    return connects


def regionGrow(img, seeds, thresh, p=1):
    height, weight = img.shape
    seedMark = np.zeros(img.shape)
    seedList = []
    for seed in seeds:
        seedList.append(seed)
    label = 1
    connects = selectConnects(p)
    while len(seedList) > 0:
        currentPoint = seedList.pop(0)

        seedMark[currentPoint.x, currentPoint.y] = label
        for i in range(8):
            tmpX = currentPoint.x + connects[i].x
            tmpY = currentPoint.y + connects[i].y
            if tmpX < 0 or tmpY < 0 or tmpX >= height or tmpY >= weight:
                continue
            grayDiff = getGrayDiff(img, currentPoint, Point(tmpX, tmpY))
            if grayDiff < thresh and seedMark[tmpX, tmpY] == 0:
                seedMark[tmpX, tmpY] = label
                seedList.append(Point(tmpX, tmpY))
    return seedMark
```

Region Based Segmentation Method Cont..


```
            if tmpX < 0 or tmpY < 0 or tmpX >= height or tmpY >= weight:
                continue
            grayDiff = getGrayDiff(img, currentPoint, Point(tmpX, tmpY))
            if grayDiff < thresh and seedMark[tmpX, tmpY] == 0:
                seedMark[tmpX, tmpY] = label
                seedList.append(Point(tmpX, tmpY))
    return seedMark


img = cv2.imread('img.jpg', 0)
seeds = [Point(10, 100), Point(20, 30), Point(200, 300)]
binaryImg = regionGrow(img, seeds, 11)

titles = ['before', 'after']
images = [img, binaryImg]
for i in range(2):
    plt.subplot(1, 2, i + 1)
    plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([]), plt.yticks([])
plt.show()
```

Region Based Segmentation Method Cont..

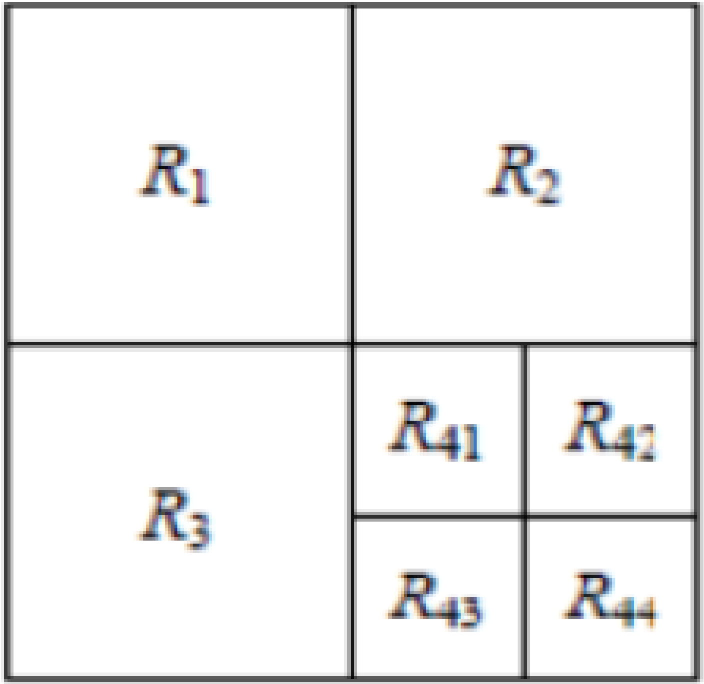
Implementation



Region Based Segmentation Method (cont.)

2. Region splitting and merging methods: The region splitting and merging based segmentation methods uses two basic techniques i.e. **splitting and merging** for segmenting an image into various regions.

- **Splitting stands** for iteratively dividing an image into regions having similar characteristics
- **Merging contributes** to combining the adjacent similar regions.



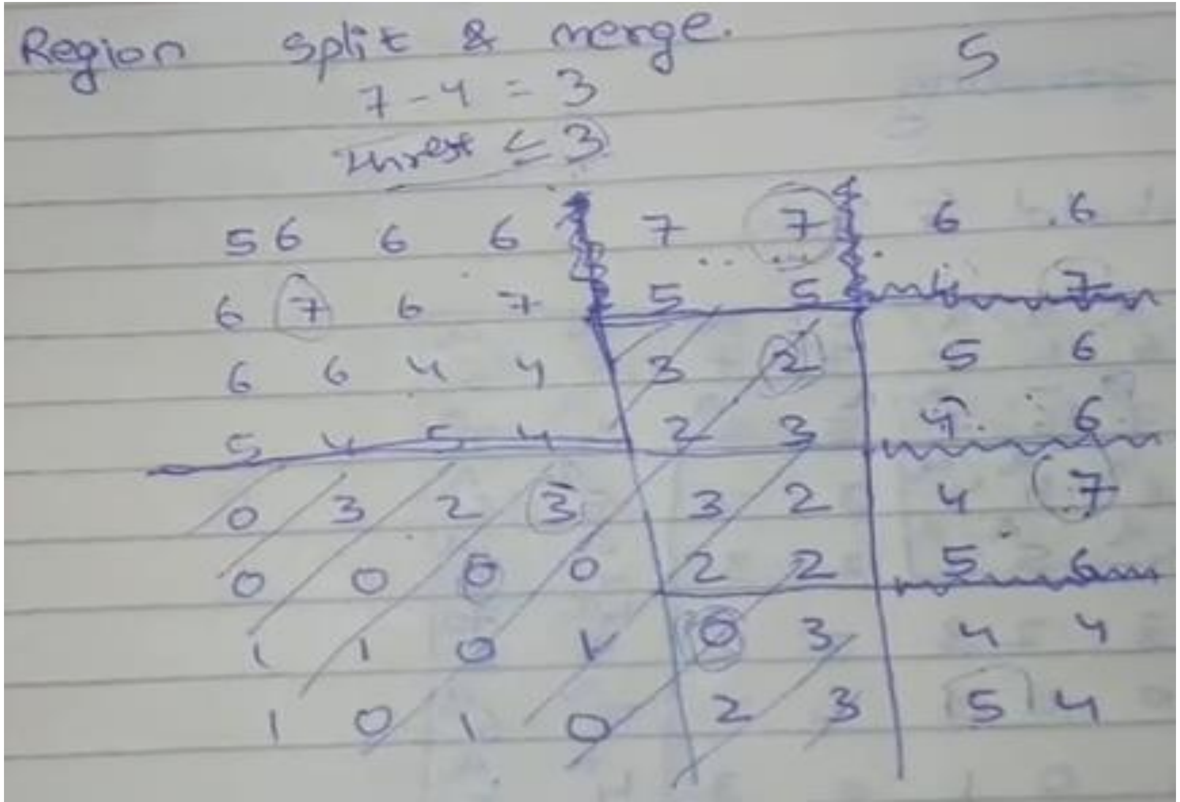
Region Based Segmentation Method (cont.)

we are make this steps:

- Divided image as four regions.
- Determine min pixel and max pixel of the first region and subtract it. Threshold (T) should be less than or equal result (<=result).
- Repeat previous step for all regions on if result more than T divide this region again Like this:

- Next step to collect related regions by subtract max pixel of region from min pixel from another region if result <= T. collect to regions as one region.
- Repeat previous step for all regions

Region Based Segmentation Method (cont.)



Clustering Based Segmentation Method

- Clustering is the task of dividing the data points into a number of groups (clusters), such that data points in the same groups are more similar to each other than those in other groups.
- One of the most commonly used clustering algorithms is k-means. Here, the k represents the number of clusters.

K-Means clustering example

The are 5 clusters and yellow dots represent the Centroid of each cluster.

Clustering Based Segmentation Method

K-means Equation

number of clusters

number of cases

centroid for cluster *j*

case *i*

$$\text{objective function} \leftarrow J = \sum_{j=1}^{k} \sum_{i=1}^{n} \underbrace{\left\| x_i^{(j)} - c_j \right\|^2}_{\text{Distance function}}$$

Clustering Based Segmentation Method

Clustering Algorithm

Initialize K from 2 to 10
Randomly initialize K cluster centroids $\mu_1, \mu_2, \cdots, \mu_k \in \mathbb{R}^n$
if $K \le 10$, repeat{
for each pixel $x^{(i)}$
$c^{(i)}$:= index (*from* 1 *to* *K*) of cluster centroid closest to $x^{(i)}$
for k = 1 to K
μ_k:= average (*mean*) *of points assigned to cluster k*
Compare the maximum connected domain results
if right, print results, break; else K=K+1;
}

Clustering Based Segmentation Method

Steps in K-Means algorithm:

1. Specify the desired number of clusters.
2. Randomly assign each data point to specific cluster.
3. Compute cluster centroids.
4. Re-assign each point to the closest cluster centroid.
5. Re-compute cluster centroids.
6. Repeat steps 4 and 5 until no improvements are possible.

Clustering Based Segmentation Method

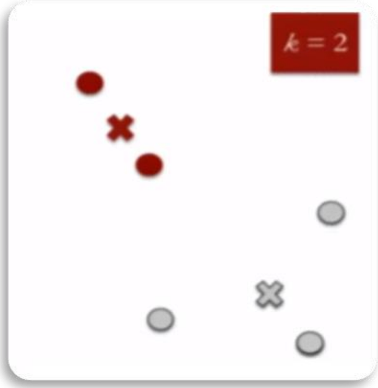
➢ **How it works:**

1. Specify the desired number of clusters K: Let’s choose k=2 clusters for these 5 data points in 2-D space.

2. Randomly assign each data point to specific cluster: Let’s assign three points in cluster 1 shown in red color and two points in cluster 2 shown in grey color.

Clustering Based Segmentation Method Cont..

5. Re-compute cluster centroids: Now, re-computing the centroids for the 2 clusters.



6. Repeat steps 4 and 5 until no improvements are possible: Similarly, we'll repeat the 4th and 5th steps until we'll reach global optima. When there will be no further switching of data points between two clusters for two successive repeats.

Clustering Based Segmentation Method Cont..

Example:

```
from skimage.io import imread
from skimage.color import rgb2gray
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy import ndimage


# Scaling the image pixels values within 0-1
img = imread('./apple-orange.jpg') / 255


plt.imshow(img)
plt.title('Original')
plt.show()
```

Clustering Based Segmentation Method Example

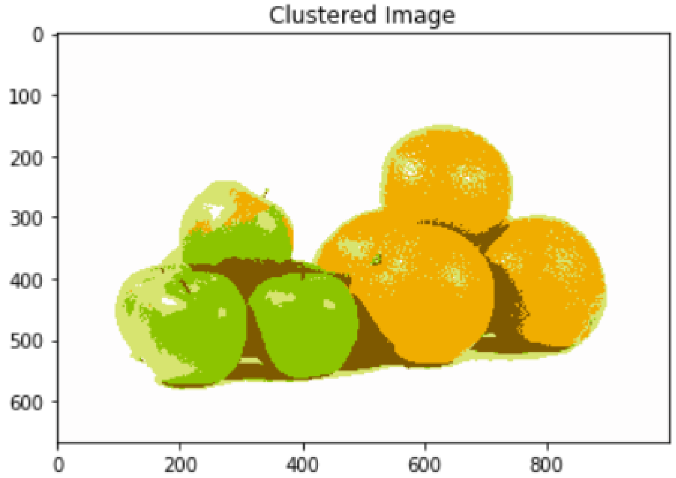
Most of the pixel points in apple are green, which is different from the pixel values of orange. If we can cluster these points we can distinguish each object from another. That's how the cluster segmentation works.

PC: Flickr

We can cluster them using our K-Means algorithm:

<u>There are five color segments in the Image:</u>

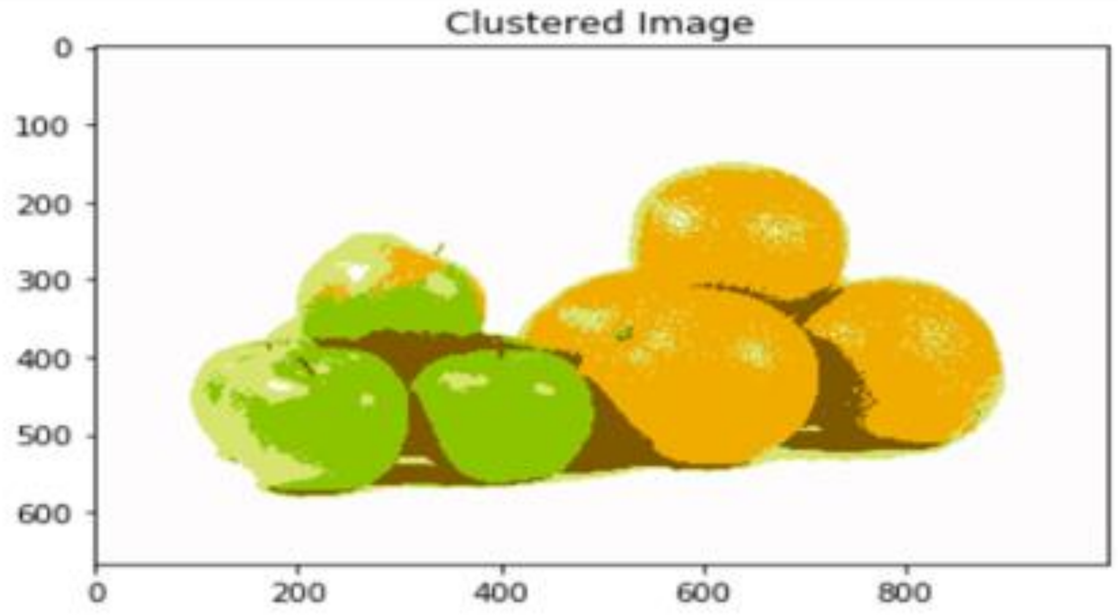
1. The green part of Apples.
2. The orange part of Oranges.
3. Gray Shadows at bottom of the Apples and oranges.
4. Bright Yellowish part of Apple's top and right parts.
5. White Background.



We can cluster them using our K-Means algorithm:


```
# For clustering the image using k-means, we first need to convert it
into a 2-dimensional array
image_2D = img.reshape(img.shape[0]*img.shape[1], img.shape[2])

# Use KMeans clustering algorithm from sklearn.cluster to cluster
pixels in image
from sklearn.cluster import KMeans

# tweak the cluster size and see what happens to the Output
kmeans = KMeans(n_clusters=5, random_state=0).fit(image_2D)
clustered = kmeans.cluster_centers_[kmeans.labels_]

# Reshape back the image from 2D to 3D image
clustered_3D = clustered.reshape(img.shape[0], img.shape[1],
img.shape[2])

plt.imshow(clustered_3D)
plt.title('Clustered Image')
plt.show()
```

We can cluster them using our K-Means algorithm:



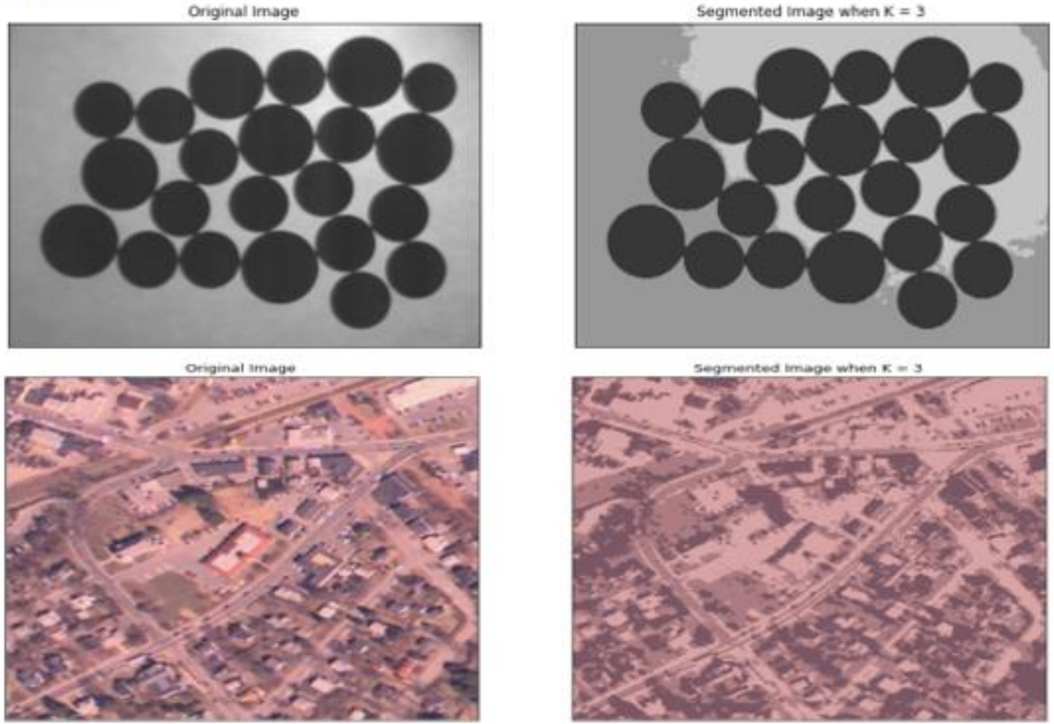
We can cluster them using our K-Means algorithm:

Code

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

original_image = cv2.imread('C:\Folder\orange.png')

img = cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB)
vectorized = img.reshape((-1, 3))
vectorized = np.float32(vectorized)
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
K = 9
attempts = 10
ret, label, center = cv2.kmeans(vectorized, K, None, criteria, attempts,
cv2.KMEANS_PP_CENTERS)
center = np.uint8(center)
res = center[label.flatten()]
result_image = res.reshape((img.shape))
figure_size = 15
plt.figure(figsize=(figure_size, figure_size))
plt.subplot(1, 2, 1), plt.imshow(img)
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(1, 2, 2), plt.imshow(result_image)
plt.title('Segmented Image when K = %i' % K), plt.xticks([]), plt.yticks([])
plt.show()
```

We can cluster them using our K-Means algorithm:

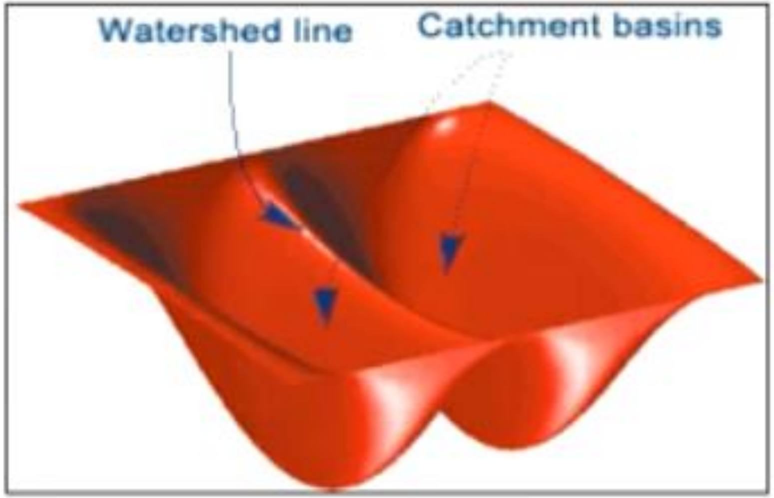
Implementation



We can cluster them using our K-Means algorithm:

Watershed Based Methods

- The watershed based methods uses topological interpretation that computes catchment basins and ridgelines (watershed lines), where catchment lines refers to image regions and ridgelines related to region boundaries.



Watershed Based Methods Example

- It is especially useful for segmenting objects that are touching one another.

Watershed Based Methods Example

Example

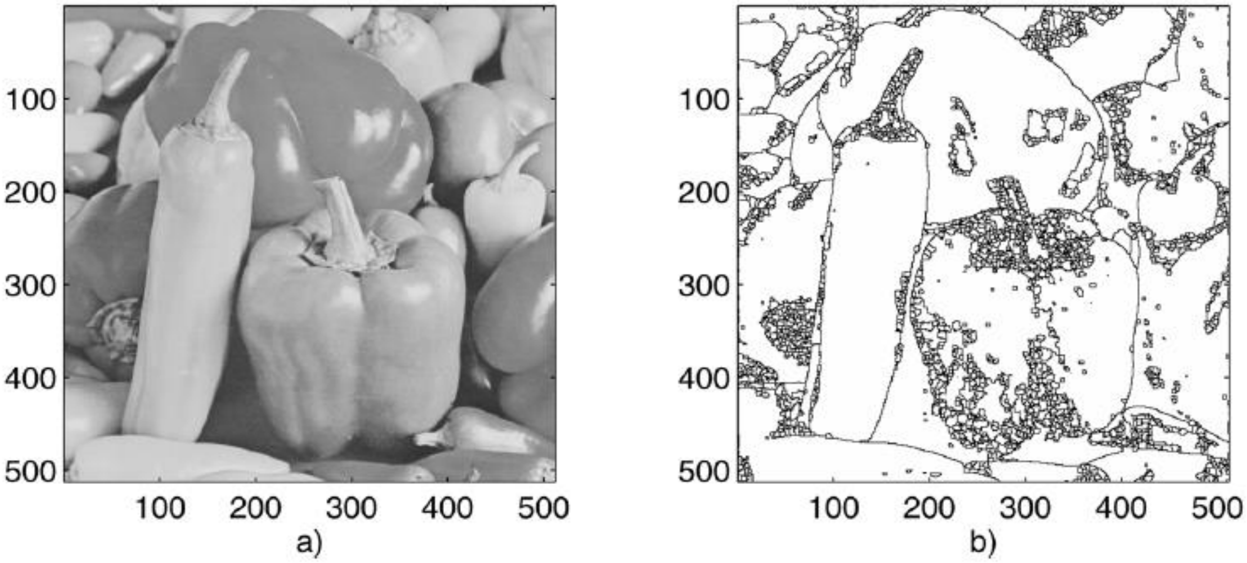


Fig. 3. *Peppers* (a) input image (b) output image.

Watershed Based Methods Example

Watershed pseudo-code

```
Algorithm 1 The pseudo-code of the algorithm
Input : f, Output : l
v[p] ← 0, l[p] ← 0, New_label ← 0, Scan_Step2 ← 1, Scan_Step3 ← 1 // Initialization
Scan from top left to bottom right : STEP1(p)
while Scan_Step2 = 1 do
    Scan image from top left to bottom right : STEP2(p)
    if v[p] is not changed then
        Scan_Step2 ← 0
    else
        Scan image from bottom right to top left : STEP2(P)
        if v[p] is not changed then
            Scan_Step2 ← 0
        end if
    end if
end while
```

end while

```
while Scan_Step3 = 1 do
    Scan image from top left to bottom right : STEP3(p)
    if l[p] is not changed then
        Scan_Step3 ← 0
    else
        Scan image from bottom right to top left : STEP3(p)
        if l[p] is not changed then
            Scan_Step3 ← 0
        end if
    end if
end while
function STEP1(p)
    if v[p] ≠ 1 then
        for each n of p // n is neighbor pixel of p
        if f[n] < f(p) then v[p] ← 1
        end if
    end if
end function
```

The basic steps in this algorithm:

- Add neighbors to priority queue, sorted by value.
- Choose local minima as region seeds.
- Take the highest priority level(pixel) from queue.
  - If neighbors contains only points with the same label, assign pixel to this label.
  - Add all non-marked neighbors into the hierarchical queue.
- Repeat step 3 until finished.

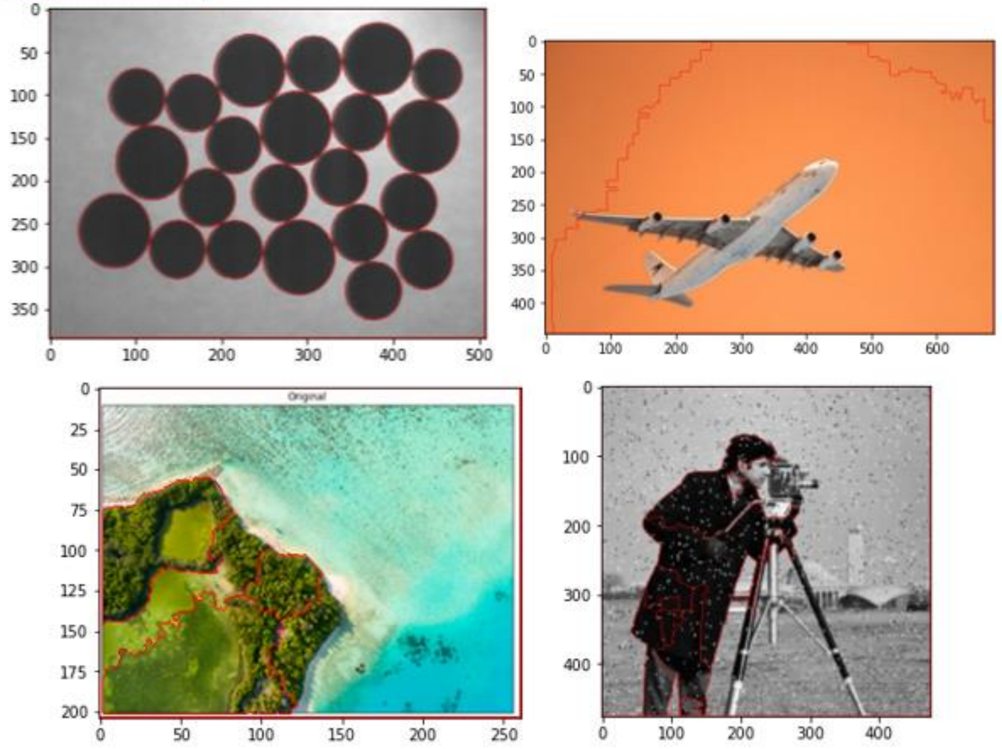
The basic steps in this algorithm:

Code

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

img = cv2.imread('C:\city.jpeg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
# noise removal
kernel = np.ones((3, 3), np.uint8)
opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel, iterations=2)
# sure background area
sure_bg = cv2.dilate(opening, kernel, iterations=3)
# Finding sure foreground area
dist_transform = cv2.distanceTransform(opening, cv2.DIST_L2, 5)
```

The basic steps in this algorithm:

Implementaion



Advantages and Disadvantages of Image Segmentation:

Advantages and Disadvantages of Image Segmentation:

Advantages

- The advantages of using these methods are that in the case of clustering algorithms, they are simple and efficient, theoretically derived (mathematically) in the case of other methods of segmentation, which is not the case with CNN or DL methods. We can easily see the hidden details in theoretically derived techniques and what characteristics lead to the result we get.

Disadvantages

- Applying DIP methods to a specific type of data set has been shown to not generalize well to another similar type of data set. For example, if we apply and create a pipeline of image segmentation to segment Indian clothes from a human, then the same pipeline can't work to segment the clothes of African or American people. This is due to the fact that the selection and implementation of the DIP methods according to the target data set is highly personalized and no parameter learning is carried out as in the case of ML and DL.

COMPARISON OF VARIOUS SEGMENTATION TECHNIQUES:

Segmentation technique	Description	Advantages	Disadvantages
Thresholding Method	based on the histogram peaks of the image to find particular threshold values	no need of previous information, simplest method	highly dependent on peaks, spatial details are not considered
Edge Based Method	based on discontinuity detection	good for images having better contrast between objects	not suitable for wrong detected or too many edges
Region Based Method	based on partitioning image into homogeneous regions	more immune to noise, useful when it is easy to define similarity criteria	expensive method in terms of time and memory
Clustering Method	based on division into homogeneous clusters	fuzzy uses partial membership therefore more useful for real problems	determining membership function is not easy
Watershed Method	based on topological interpretation	results are more stable, detected boundaries are continuous	complex calculation of gradients

Image Segmentation Applications:

1- Object Detection and Face Detection:

- Face detection: Algorithms detect and verify the presence of facial features.
- Medical imaging: extracts clinically relevant information from medical images.
- Machine vision: applications that capture and process images to provide operational guidance to devices.

2- Video Surveillance: video tracking and moving object tracking:

- Self-driving vehicles: autonomous cars must be able to perceive and understand their environment in order to drive safely.
- Iris recognition: It uses automated pattern recognition to analyze video images of a person’s eye.
- Face recognition: identifies an individual in a frame from a video source. This technology compares selected facial features from an input image with faces in a database.

Conclusions

- Image segmentation is a field in image processing .It works by dividing image into various parts.
- There are two types of image segmentation (Semantic Segmentation, Instance Segmentation)
- There are five methods of image segmentation (Threshold Method, Edge Based Method, Region Based Method, Clustering Based Method, and Watershed Method)
- There are basically three types of thresholding (Global Thresholding, Variable Thresholding, and Multiple Thresholding).
- Finally, after reviewing the results of the above application on all methods of image segmentation and comparing between each method and the other in order to find the best way to divide the images, we found that the results of the methods of reducing the best and simplest methods of segmentation of images, as they divide the objects inside the images by 95%, and the objects are clear and separate from the rest of the image.

Resources

- Dilpreet Kaur, Y. K. (2014). Various Image Segmentation Techniques. International Journal of Computer Science and Mobile Computing, ijcsmc.
- Piatetsky-Shapiro, G. (2018). Introduction to Image Segmentation with K-Means clustering. Retrieved from kdnuggets: https://www.kdnuggets.com/2019/08/introduction-image-segmentation-k-means-clustering.html
- Rajeshwar Dass, P. S. (2012). Image Segmentation Techniques.
- Team, D. C. (2020). Introduction to Image Segmentation in Deep Learning. Retrieved from debuggercafe: https://debuggercafe.com/introduction-to-image-segmentation-in-deep-learning/
- Team, i. k. (2018, December 2). Basic Introduction to computer vision. Retrieved from kapernikov: https://kapernikov.com/basic-introduction-to-computer-vision/
- Team, t. (n.d.). Image segmentation. Retrieved from tensorflow: https://www.tensorflow.org/tutorials/images/segmentation
- Dilpreet Kaur, Y. K. (2014). Various Image Segmentation Techniques. International Journal of Computer Science and Mobile Computing, ijcsmc.

Resources (Cont.)

- Bieniek, A., & A. Moga. (1999, 7 27). An effcient watershed algorithm based on connected. p. 10.
- Dubey, S. K., Vijay, D., & Pratibha. (2018). A Review of Image Segmentation using Clustering Methods. p. 6.
- Kaur, D., & Kau, Y. (2014, May 5). Various Image Segmentation. p. 6.
- KDnuggets. (2019, August). Retrieved from KDnuggets Web site: https://www.kdnuggets.com/2019/08/introduction-image-segmentation-k-means-clustering.html.
- Piatetsky-Shapiro, G. (2018). Introduction to Image Segmentation with K-Means clustering. Retrieved from kdnuggets: https://www.kdnuggets.com/2019/08/introduction-image-segmentation-k-means-clustering.html
- Rajeshwar Dass, P. S. (2012). Image Segmentation Techniques.
- SHARMA, P. (2019, April 1). Computer Vision Tutorial: A Step-by-Step Introduction to Image Segmentation Techniques (Part 1). Retrieved from https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/

Resources (Cont.)

- Spring Open. (2018, August 3). Retrieved from Spring Open Web site: https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-018-0309-3
- Team, D. C. (2020). Introduction to Image Segmentation in Deep Learning. Retrieved from debuggercafe: https://debuggercafe.com/introduction-to-image-segmentation-in-deep-learning/
- Team, i. k. (2018, December 2). Basic Introduction to computer vision. Retrieved from kapernikov: https://kapernikov.com/basic-introduction-to-computer-vision/
- Team, t. (n.d.). Image segmentation. Retrieved from tensorflow: https://www.tensorflow.org/tutorials/images/segmentation.
- Dilpreet Kaur, Y. K. (2014). Various Image Segmentation. International Journal of Computer Science and Mobile Computing.
- Jianjun Chen, H. S. (2017). Image Segmentation Based on Mathematical Morphological Operator. Retrieved from intechopen.
- professionals, g. o. (n.d.). introduction-image-segmentation-techniques-python. Retrieved from analyticsvidhya.

Resources (Cont.)

- Y.J. Zhang, (1996). A survey on evaluation methods for image segmentation.
- William Jackson, (2010). Image Segmentation by Using Threshold Techniques.
- IEEE, (1979). Image segmentation by clustering.
- Nassir Salman, (2004). Image Segmentation Based on Watershed and Edge Detection Techniques.
- Sharma, M. (2020, 3 4). towards data science. Retrieved from Cluster-based Image Segmentation -Python: https://towardsdatascience.com/cluster-based-image-segmentation-python-80a295f4f3a2
- S.A. Hojjatoleslami , J. Kittler, (1995). Region Growing: A New Approach.

Thanks and Acknowledgement

Thank you

Dr. Ahmed A. Elngar (Ph.D)
Assistant Professor
Faculty of Computers & Artificial Intelligence
Beni-Suef University, Beni Suef City, office box # (62511),, Egypt
Founder and Chair of the Scientific Innovation Research Group (SIRG)
Managing Editor in Journal of CyberSecurity and Information Management (JCIM)
Email: elngar_7@yahoo.co.uk
Email: ahmedelnagar@fcis.bsu.edu.eg
Mobile : (+2)01007400752
www.sirg.club

SIRG
Scientific Innovation Research Group