Computer Vision: Algorithms and Applications

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

Course Syllabus

- Introduction.
- Image Morphology.
- Image Segmentation & Color Spaces.
- Feature Detection.
- Feature Matching.
- Image Clustering.
- Image Classification.
- Face Detection & Recognition.
- Image Stitching.
- Scene Detection.
- Video Shot Detection.
- Object Tracking.

Lecture 03 (1): Image Segmentation

Introduction

- 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*

Introduction

- 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

Introduction

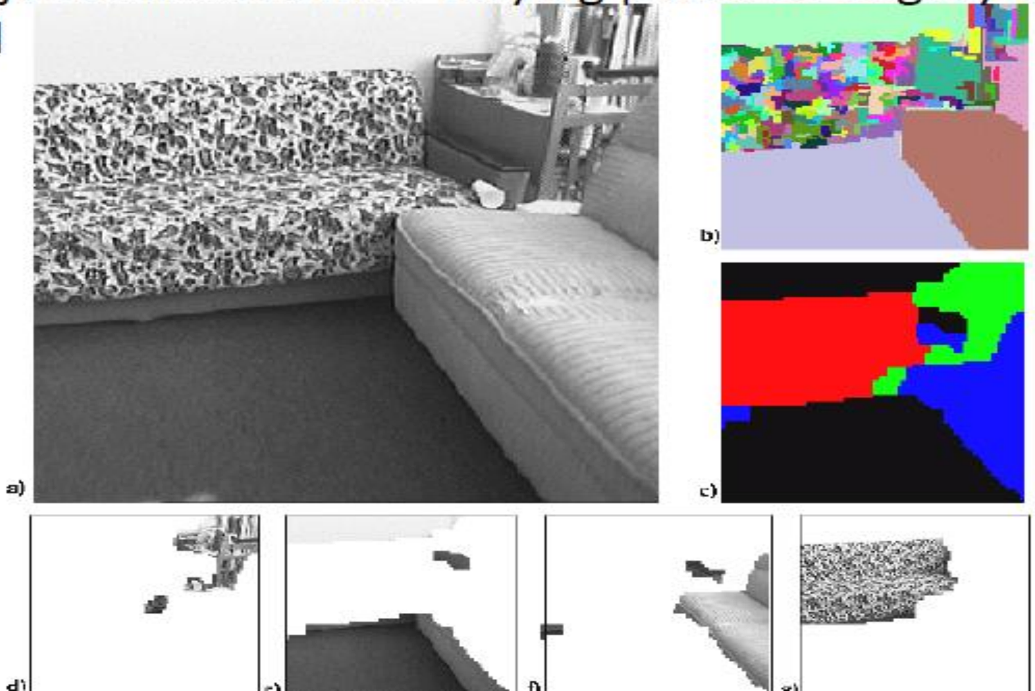
- Segmentation based on greyscale

Very simple 'model' of greyscale leads to inaccuracies in object labelling

Introduction

- Segmentation based on texture

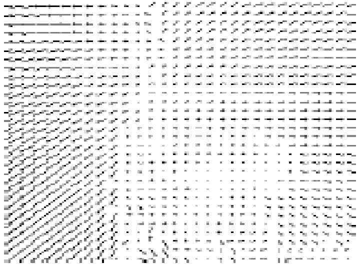
Enables object surfaces with varying patterns of grey to be segmented



Introduction

- **Segmentation based on motion**

The main difficulty of motion segmentation is that an intermediate step is required to (either implicitly or explicitly) estimate an *optical flow field*
The segmentation must be based on this estimate and not, in general, the true flow



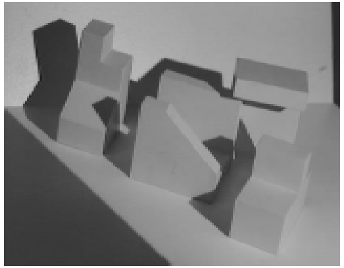
Introduction

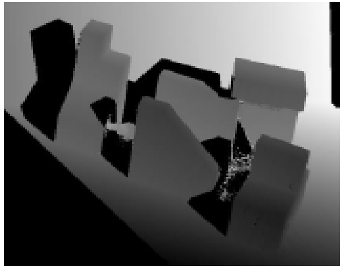
- **Segmentation based on depth**

This example shows a range image, obtained with a laser range finder

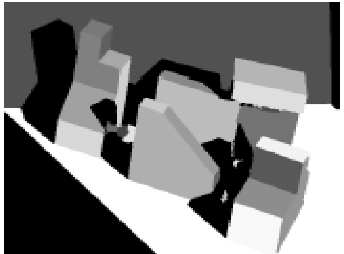
A segmentation based on the range (the object distance from the sensor) is useful in guiding mobile robots

Original image





Range image



Segmented image

Image Segmentation

Image segmentation is the operation of partitioning an image into a collection of connected sets of pixels.

1. into regions, which usually cover the image

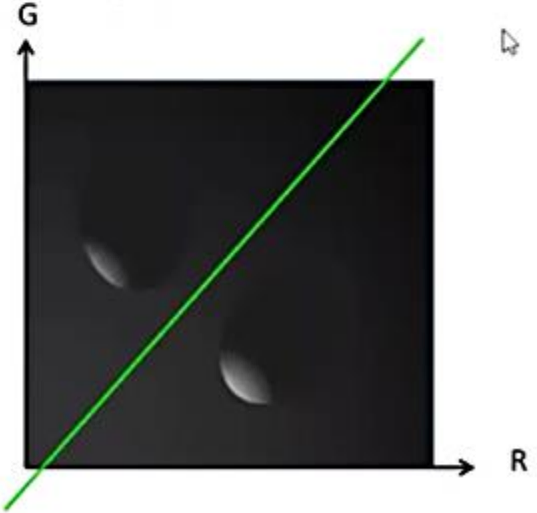
2. into linear structures, such as
   - line segments
   - curve segments

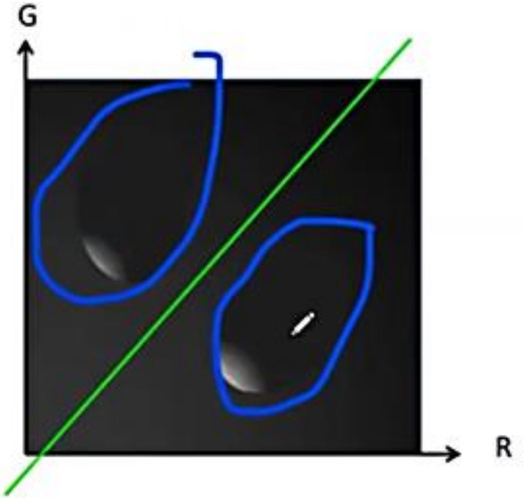
3. into 2D shapes, such as
   - circles
   - ellipses
   - ribbons (long, symmetric regions)

Colour images

Difficult to use thresholding for colour pixels

Can form a histogram in RGB space and choose straight lines that separate the modes





But sometimes this can't split the modes cleanly

Line may split another mode in half

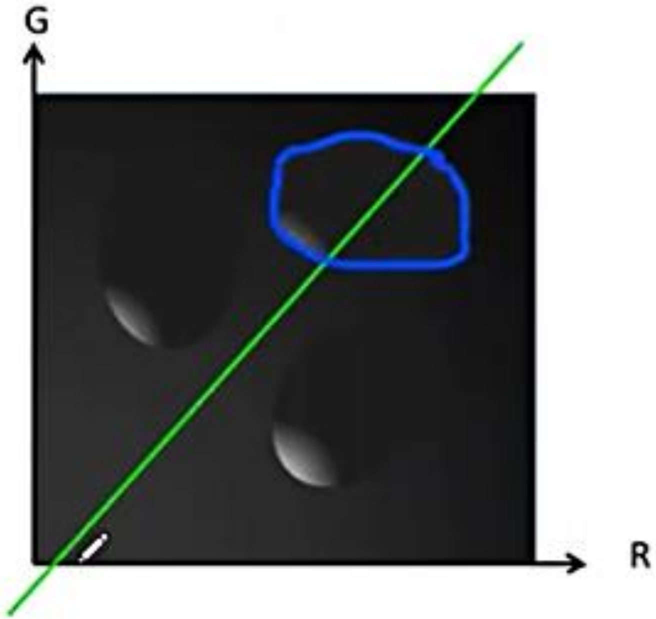


Image Segmentation

- 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.

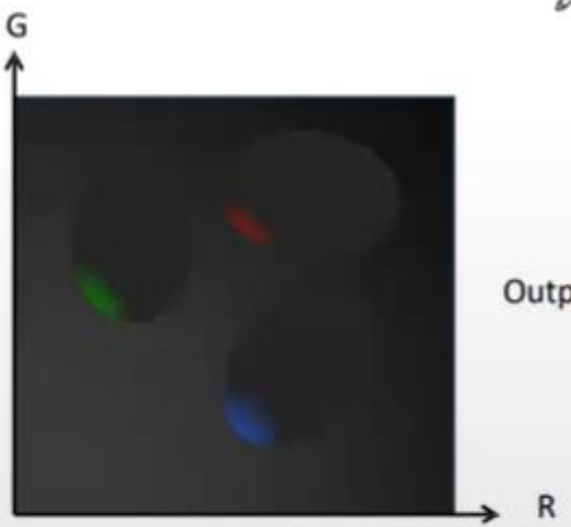
K-Means clustering example

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

K means clustering

Goal is to split a feature space into K regions/ clusters (K supplied by user)

Feature space is colour space in this example



Output of K-means clustering with K=3

SIRG
Scientific Innovation Research Group

K means clustering

R

K means clustering keeps K separate RGB values initialised to random colours in the image.

3 B G

Image is then scanned and each pixel is assigned to the nearest of the K values

When a pixel is assigned, that value is updated to be the mean of all the pixels that have been assigned to it

Repeat a few times to allow means to move to optimal positions

K-means Equation

number of clusters

number of cases

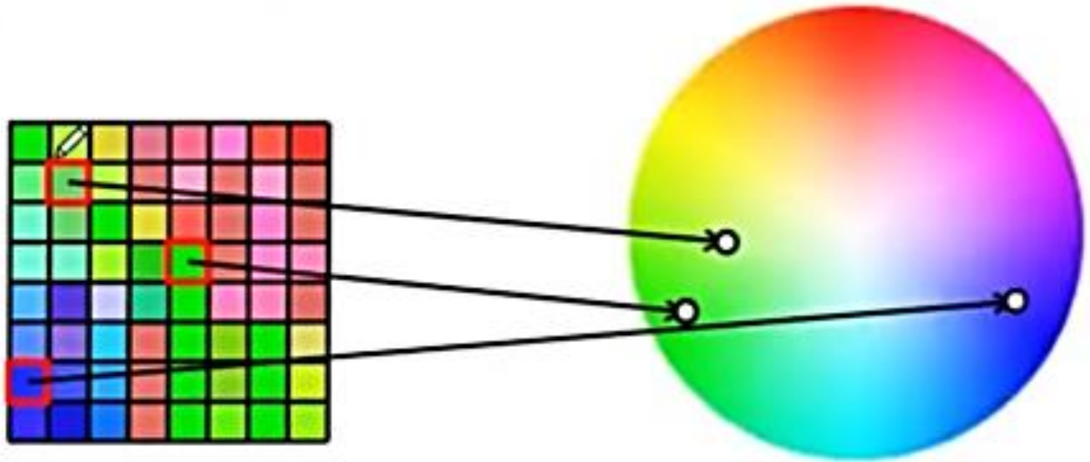
centroid for cluster *j*

case *i*

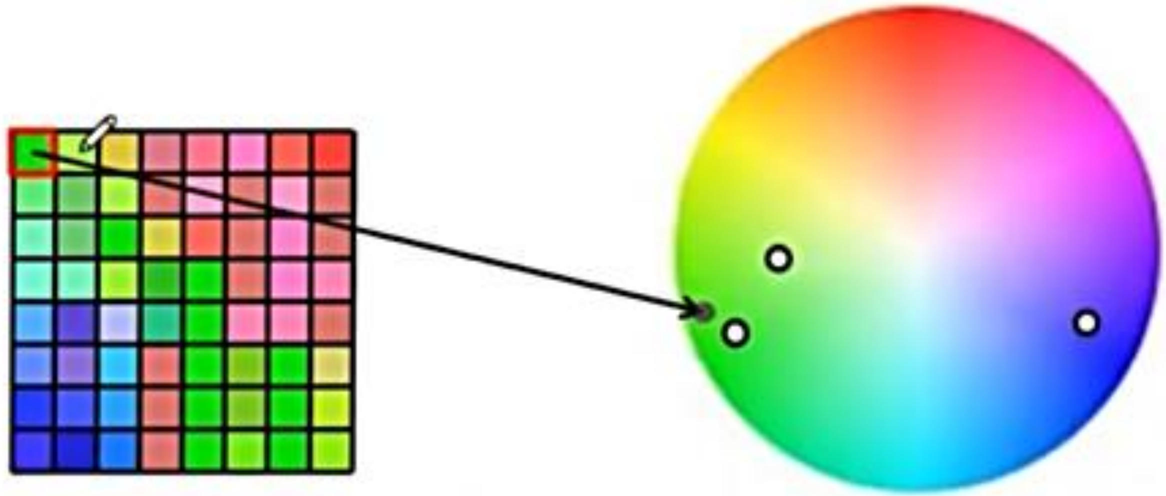
$$\text{objective function} \leftarrow J = \sum_{j=1}^{k} \sum_{i=1}^{n} \underbrace{\left\| x_i^{(j)} - c_j \right\|^2}_{\text{Distance function}}$$

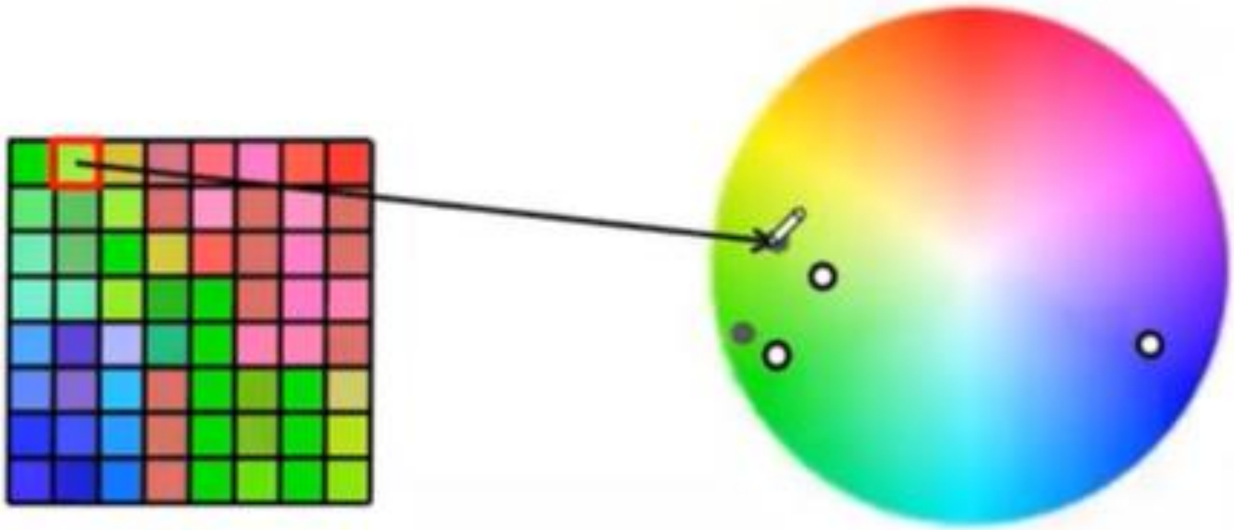
K means clustering

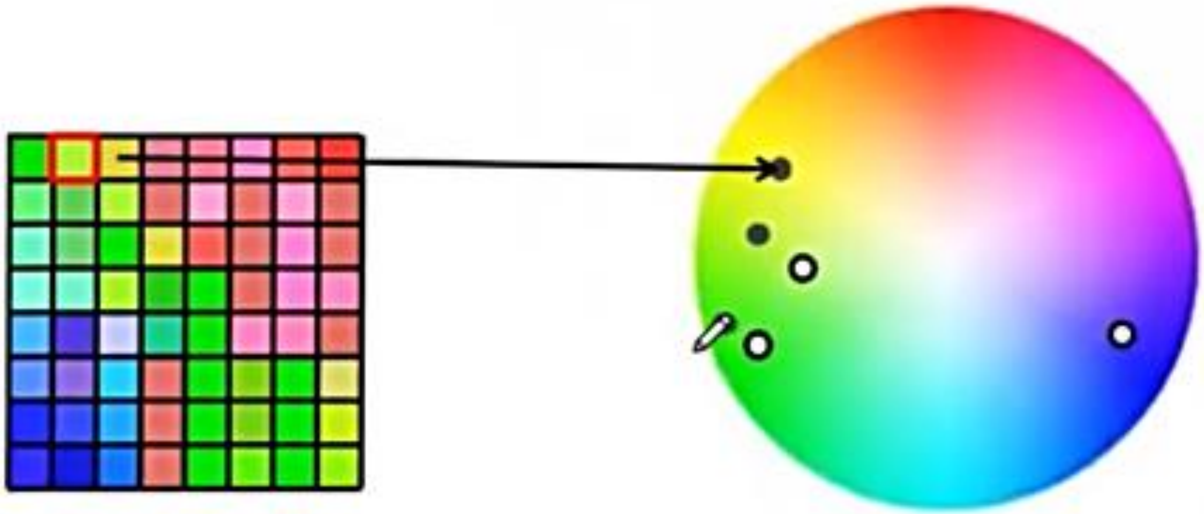
Pick 3 pixels at random and initialise means



Loop over pixels and project them into the colour space:







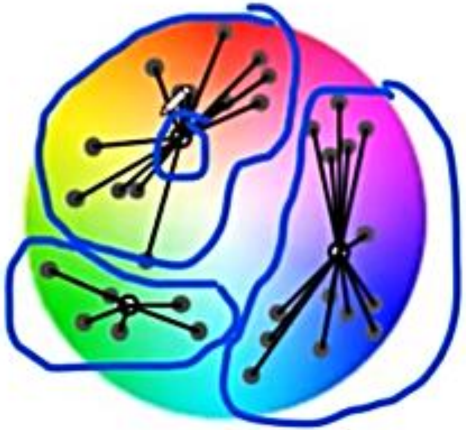
K means clustering

$K = 3$

Assign each pixel to its closest mean

K means clustering

Recompute the means



K means clustering

Recompute the assignments

K means clustering

Recompute the means again

K means clustering

Recompute the assignments

K means clustering

Recompute the means

K means clustering

Recompute the assignments

K means clustering

Recompute the means
And now the assignments are stable
So we can label each pixel by its nearest mean

K means clustering: Face Example

If k = 10, and we run 100 iterations to find the colour mean, this is what we will get:

K means clustering: Face Example

So what?

Well, you know there are 10 groups of colours since k=10.

In each of the 10 colour groups, you have different clusters formed.

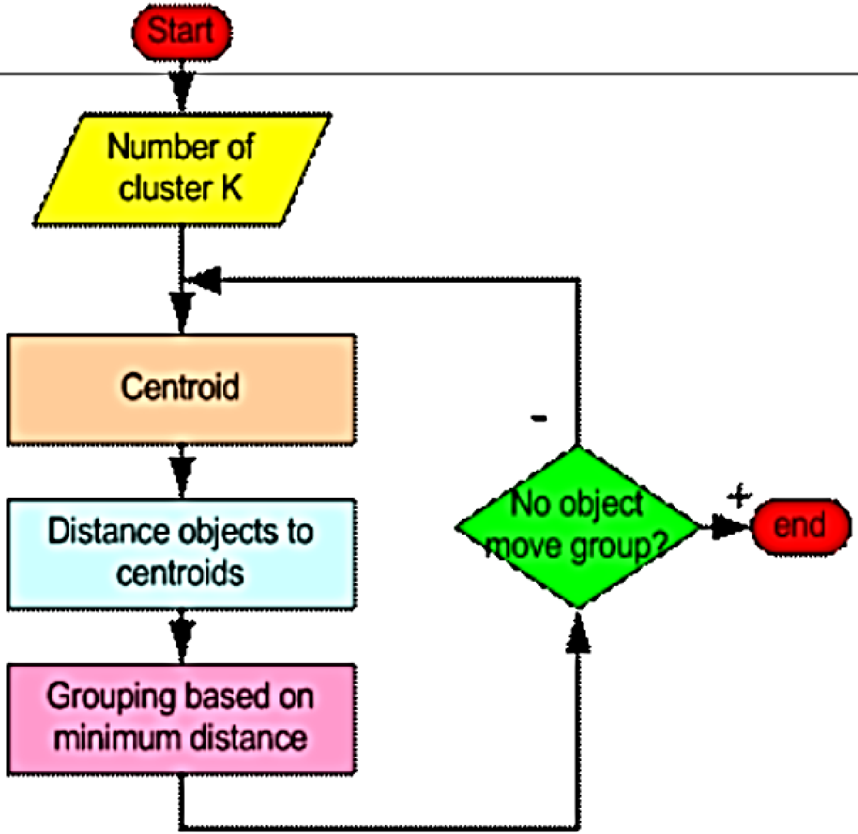
K means clustering: Harder Example

If k = 10, and we run 100 iterations to find the colour mean, this is what we will get:

Now if you threshold Cr and Cb

Then perform connected components and one of the clusters will be this

How the K-Mean Clustering algorithm works?



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 ≤ 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
$\mu_k :=$ average (*mean*) *of points assigned to cluster k*
Compare the maximum connected domain results
if right, print results, break; else K=K+1;
}

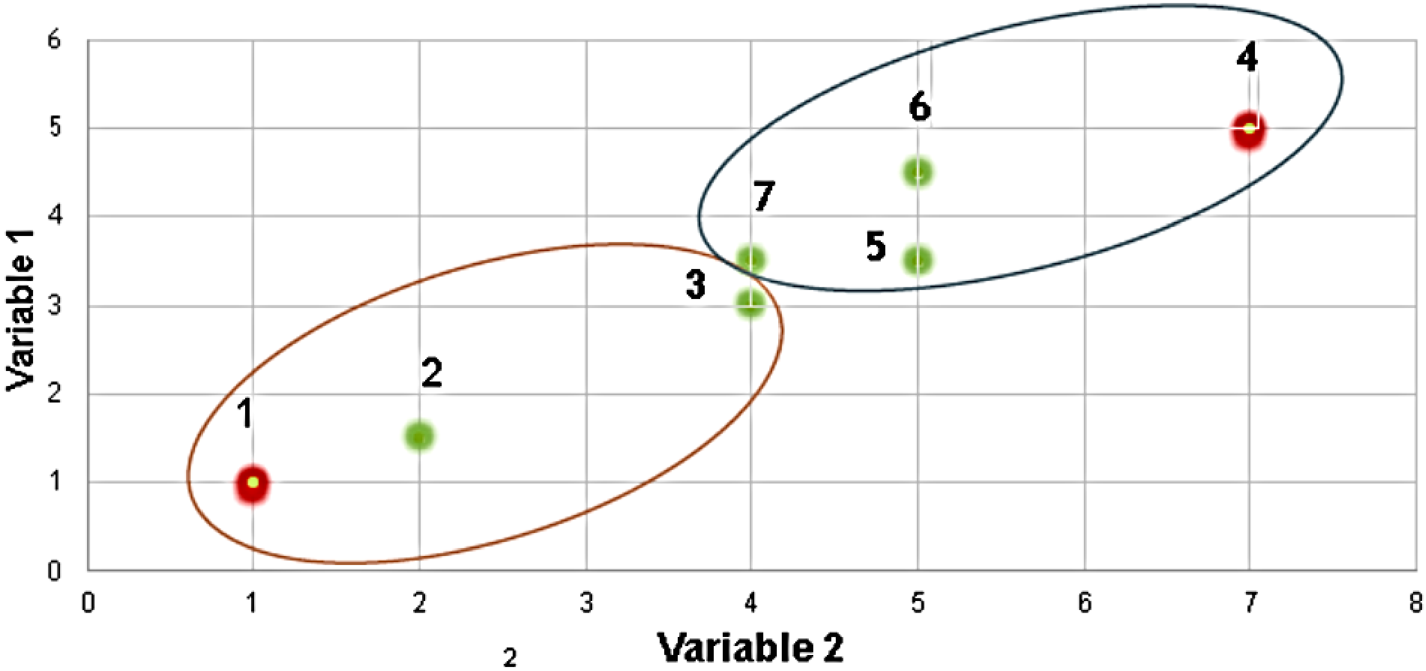
Steps in K-Means algorithm:

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.

A Simple example k-means (using K=2)

Individual	Variable 1	Variable 2
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5

K= 2



2

Step 1:

Initialization: Randomly we choose following two centroids (k=2) for two clusters. In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2:

	Centroid 1	Centroid 2
1	$\sqrt{(1-1)^2+(1-1)^2}$ = 0	$\sqrt{(5-1)^2+(7-1)^2}$ = 7.21
2	$\sqrt{(1-1.5)^2+(1-2)^2}$ = 1.12	$\sqrt{(5-1.5)^2+(7-2)^2}$ = 6.10
3	$\sqrt{(1-3)^2+(1-4)^2}$ = 3.61	$\sqrt{(5-3)^2+(7-4)^2}$ = 3.61
4	$\sqrt{(1-5)^2+(1-7)^2}$ = 7.21	$\sqrt{(5-5)^2+(7-7)^2}$ = 0
5	$\sqrt{(1-3.5)^2+(1-5)^2}$ = 4.72	$\sqrt{(5-3.5)^2+(7-5)^2}$ = 2.5
6	$\sqrt{(1-4.5)^2+(1-5)^2}$ = 5.31	$\sqrt{(5-4.5)^2+(7-5)^2}$ = 2.06
7	$\sqrt{(1-3.5)^2+(1-4.5)^2}$ = 4.30	$\sqrt{(5-3.5)^2+(7-4.5)^2}$ = 2.92

Step 2:

- Thus, we obtain two clusters containing: {1,2,3} and {4,5,6,7}.
- Their new centroids are:

$$\text{Group 1} = \left(\frac{1+1.5+3}{3}\right),\ \left(\frac{1+2+4}{3}\right) = (1.83\ ,\ 2.33)$$

$$\text{Group 2} = \left(\frac{5+3.5+4.5+3.5}{4}\right),\ \left(\frac{7+5+5+4.5}{4}\right) = (4.12\ ,\ 5.38)$$

Step 3:

	Centroid 1	Centroid 2
1	$\sqrt{(1.83-1)^2+(2.33-1)^2} = 1.57$	$\sqrt{(4.12-1)^2+(5.38-1)^2} = 5.38$
2	$\sqrt{(1.83-1.5)^2+(2.33-2)^2} = 0.47$	$\sqrt{(4.12-1.5)^2+(5.38-2)^2} = 4.29$
3	$\sqrt{(1.83-3)^2+(2.33-4)^2} = 2.04$	$\sqrt{(4.12-3)^2+(5.38-4)^2} = 1.78$
4	$\sqrt{(1.83-5)^2+(2.33-7)^2} = 5.64$	$\sqrt{(4.12-5)^2+(5.38-7)^2} = 1.84$
5	$\sqrt{(1.83-3.5)^2+(2.33-5)^2} = 3.15$	$\sqrt{(4.12-3.5)^2+(5.38-5)^2} = 0.73$
6	$\sqrt{(1.83-4.5)^2+(2.33-5)^2} = 3.78$	$\sqrt{(4.12-4.5)^2+(5.38-5)^2} = 0.54$
7	$\sqrt{(1.83-3.5)^2+(2.33-4.5)^2} = 2.74$	$\sqrt{(4.12-3.5)^2+(5.38-4.5)^2} = 1.08$

Therefore, the new clusters are:

{1,2} and {3,4,5,6,7}

Group 1 $= \left(\frac{1+1.5}{2}\right), \left(\frac{1+2}{2}\right) = (1.25 , 1.5)$

Group 2 $= \left(\frac{3+5+3.5+4.5+3.5}{5}\right), \left(\frac{4+7+5+5+4.5}{5}\right) = (3.9 , 5.1)$

Step 4:

	Centroid 1	Centroid 2
1	$\sqrt{(1.25 - 1)^2 + (1.5 - 1)^2} = 0.58$	$\sqrt{(3.9 - 1)^2 + (5.1 - 1)^2} = 5.02$
2	$\sqrt{(1.25 - 1.5)^2 + (1.5 - 2)^2} = 0.56$	$\sqrt{(3.9 - 1.5)^2 + (5.1 - 2)^2} = 3.92$
3	$\sqrt{(1.25 - 3)^2 + (1.5 - 4)^2} = 3.05$	$\sqrt{(3.9 - 3)^2 + (5.1 - 4)^2} = 1.42$
4	$\sqrt{(1.25 - 5)^2 + (1.5 - 7)^2} = 6.66$	$\sqrt{(3.9 - 5)^2 + (5.1 - 7)^2} = 2.20$
5	$\sqrt{(1.25 - 3.5)^2 + (1.5 - 5)^2} = 4.16$	$\sqrt{(3.9 - 3.5)^2 + (5.1 - 5)^2} = 0.41$
6	$\sqrt{(1.25 - 4.5)^2 + (1.5 - 5)^2} = 4.78$	$\sqrt{(3.9 - 4.5)^2 + (5.1 - 5)^2} = 0.61$
7	$\sqrt{(1.25 - 3.5)^2 + (1.5 - 4.5)^2} = 3.75$	$\sqrt{(3.9 - 3.5)^2 + (5.1 - 4.5)^2} = 0.72$

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and
  {3,4,5,6,7}.

Example:

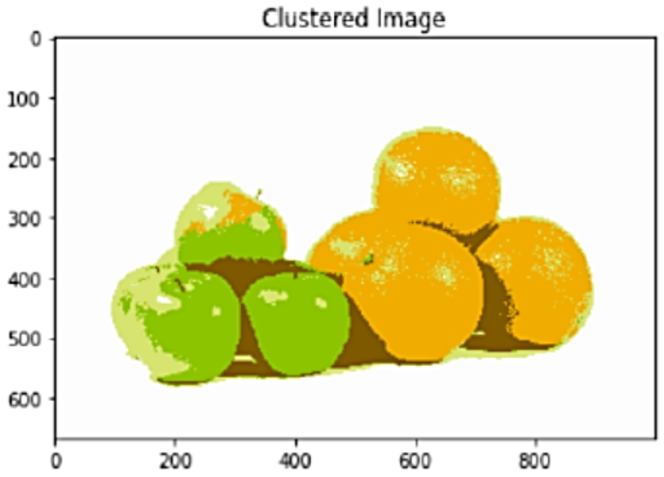
```
from skimage.io import imread
from skimage.color import rgb2gray
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy import ndimage


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


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

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.

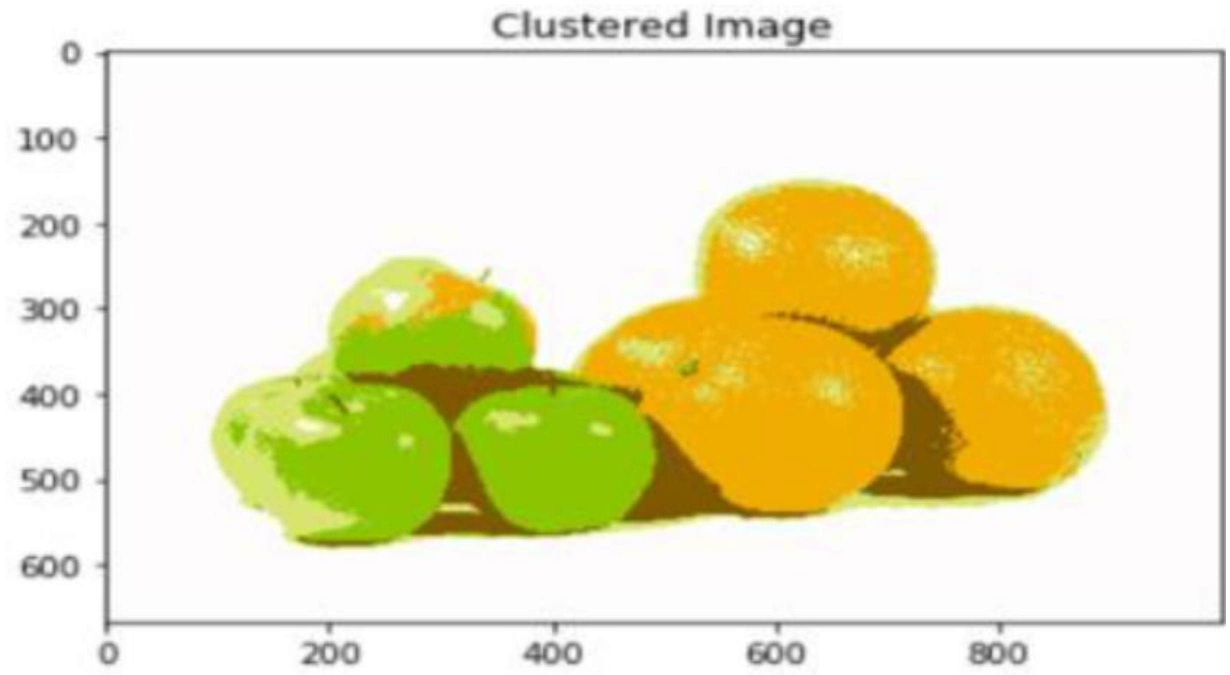
PC: Flickr

There are five color segments in the Image:

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.



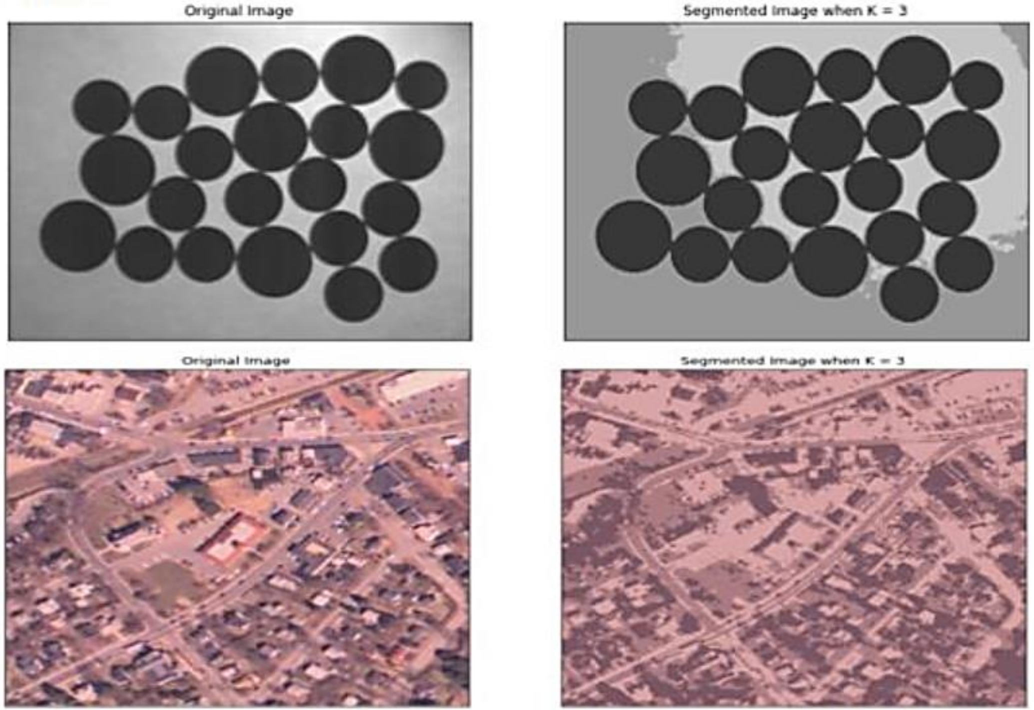
```
# For clustering the image using k-means, we first need to convert it
into a 2-dimensional array
image_2D = img.reshape(img.shape[0]*img.shape[1], img.shape[2])

# 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()
```

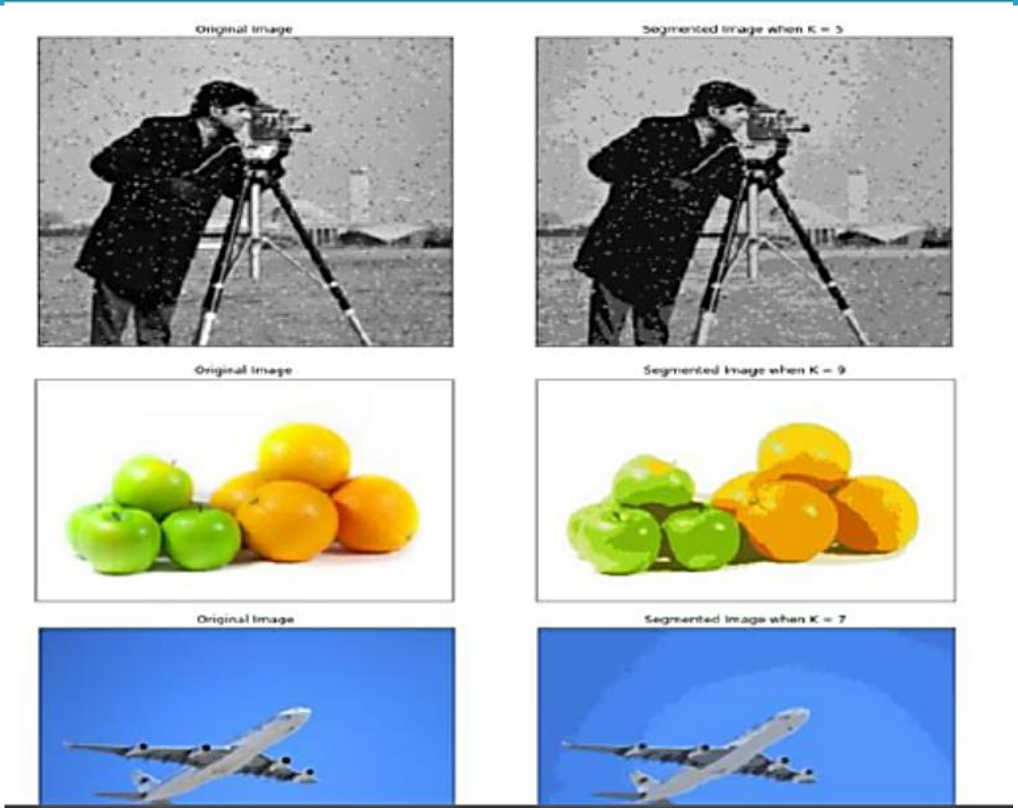

Clustered Image
0
100
200
300
400
500
600
0
200
400
600
800

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()
```

Implementation




Original Image
Segmented Image when K = 5
Original Image
Segmented Image when K = 9
Original Image
Segmented Image when K = 7

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.

Pros and cons

Advantages of k-means

1. Relatively simple to implement.
2. Scales to large data sets.
3. Guarantees convergence.
4. Easily adapts to new examples.

Disadvantages of k-means

1. Choosing \(k\) manually.
2. Being dependent on initial values.
3. Scaling with number of dimensions.

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.

Aim of our Research Group:

The aim of our Scientific Innovation research Group (SIRG) to evaluate the IOT performance by propose a secure architecture for the IoT security issues for Education.

Thanks and Acknowledgement

Thank you