CV Practice Class
6. Clustering and Segmentation

2017-07-12

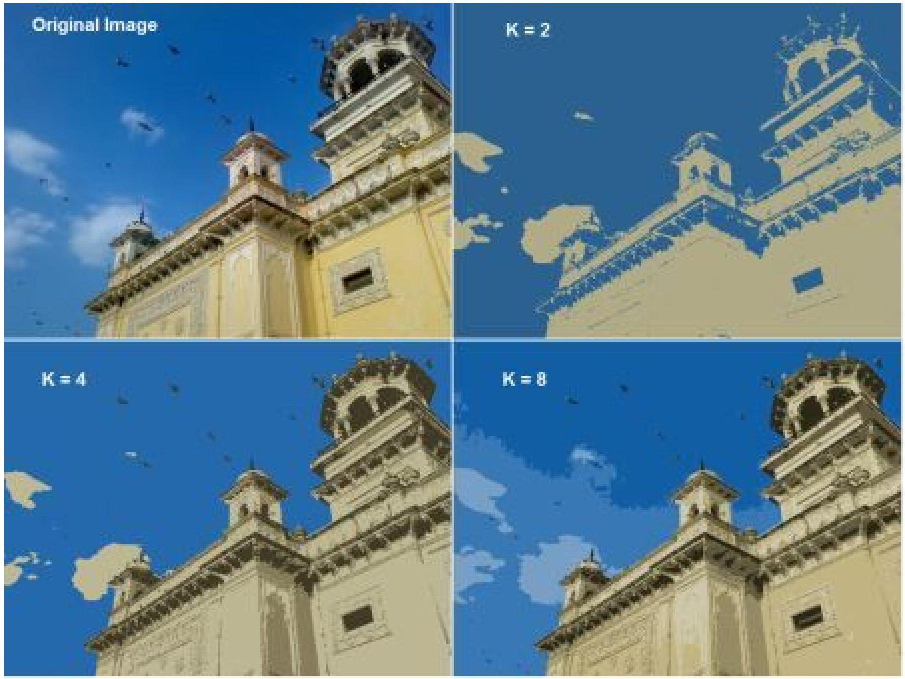
FIRA 인공지능 에이전트 과정
SNUVL Lab

Contents

1. Clustering
    a. K-means clustering
2. Segmentation
3. Face classification

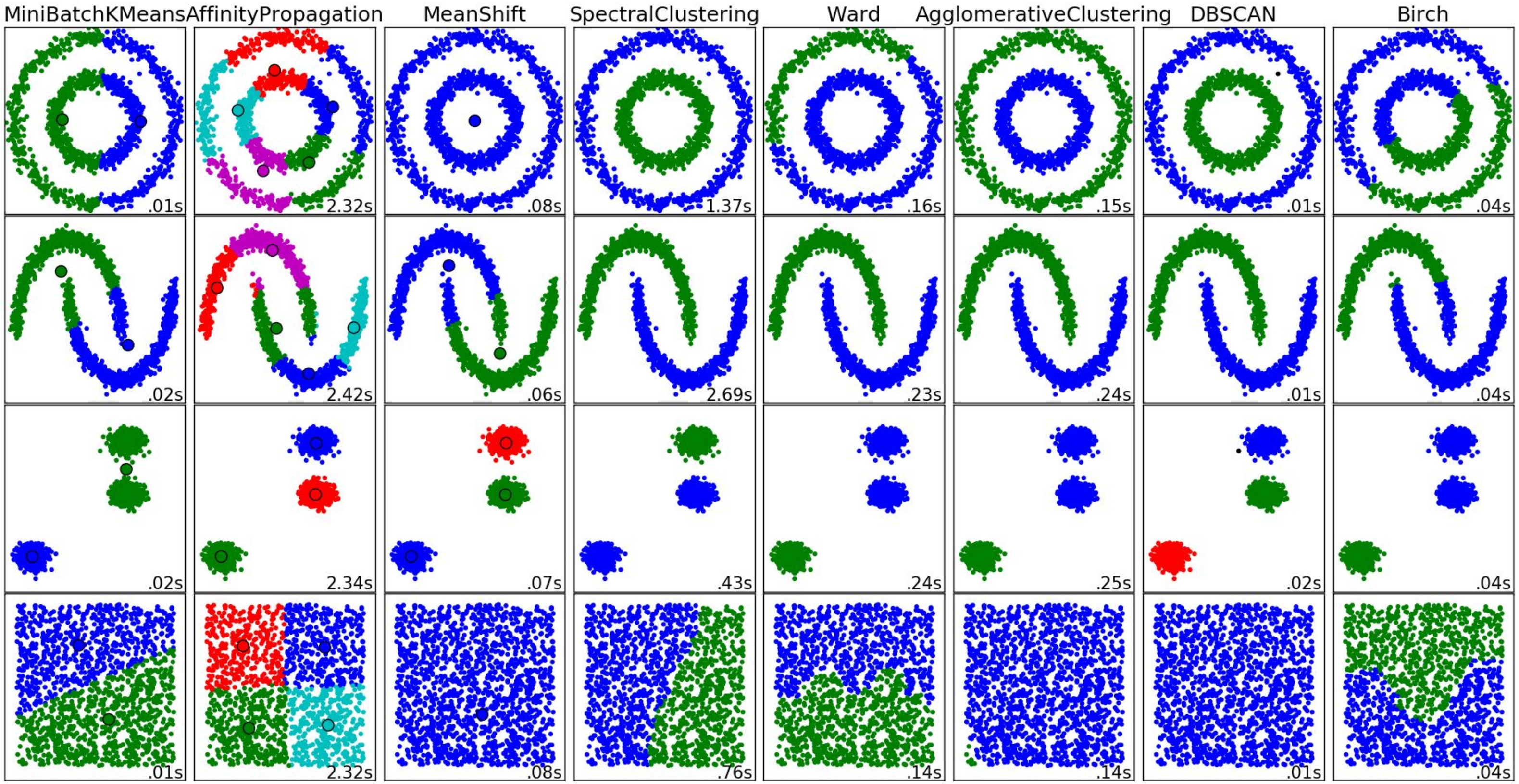
Clustering

K-means Clustering
Scikit-Learn



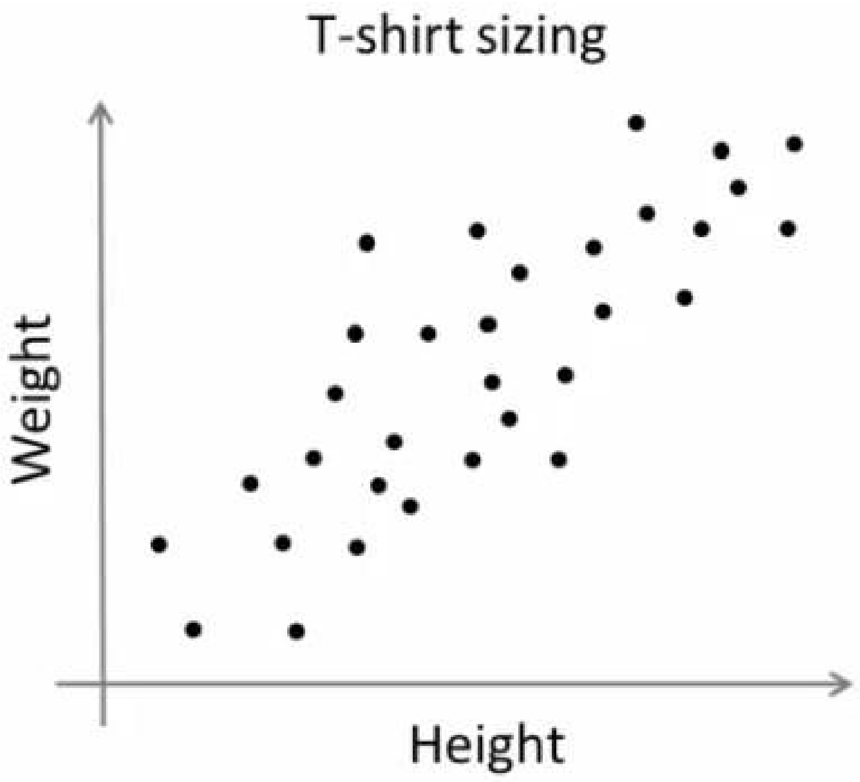
Clustering

Group together similar points and represent them with a single token



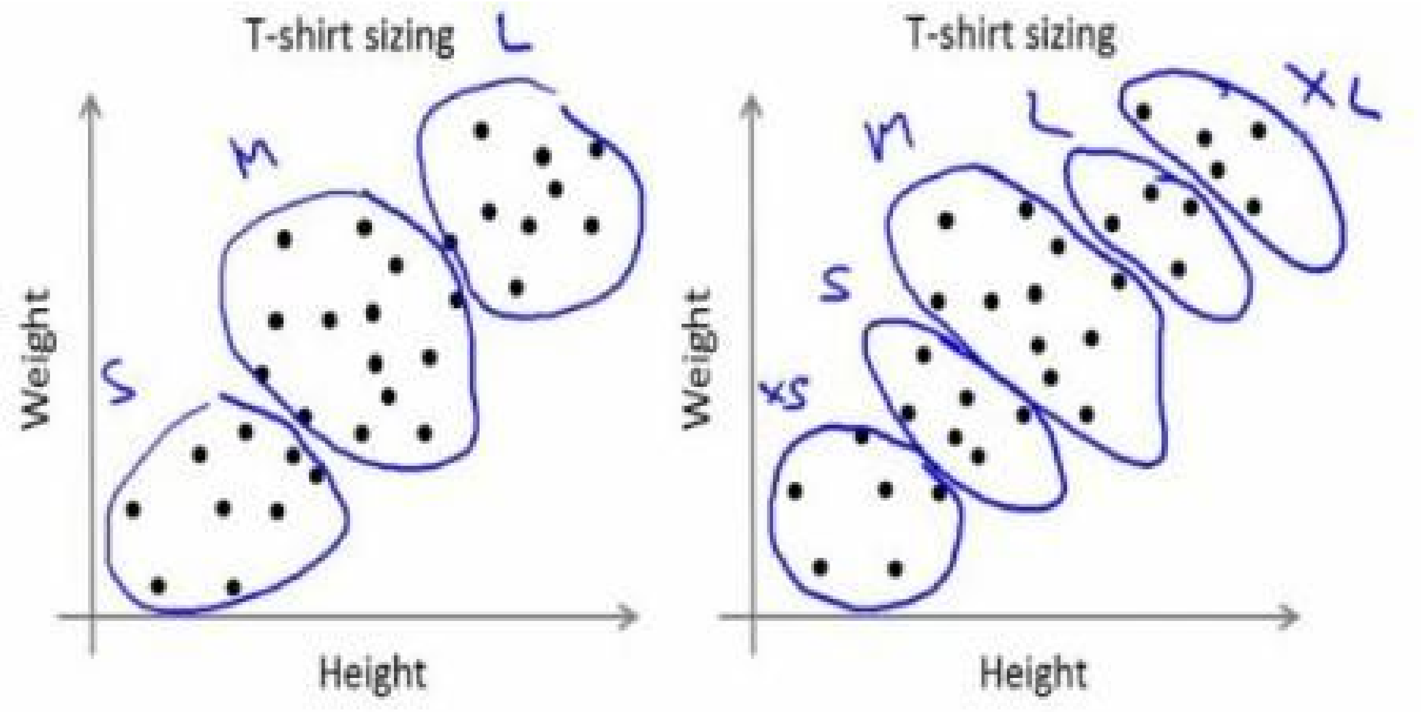
T-shirt Size Problem

How can we create t-shirts size?



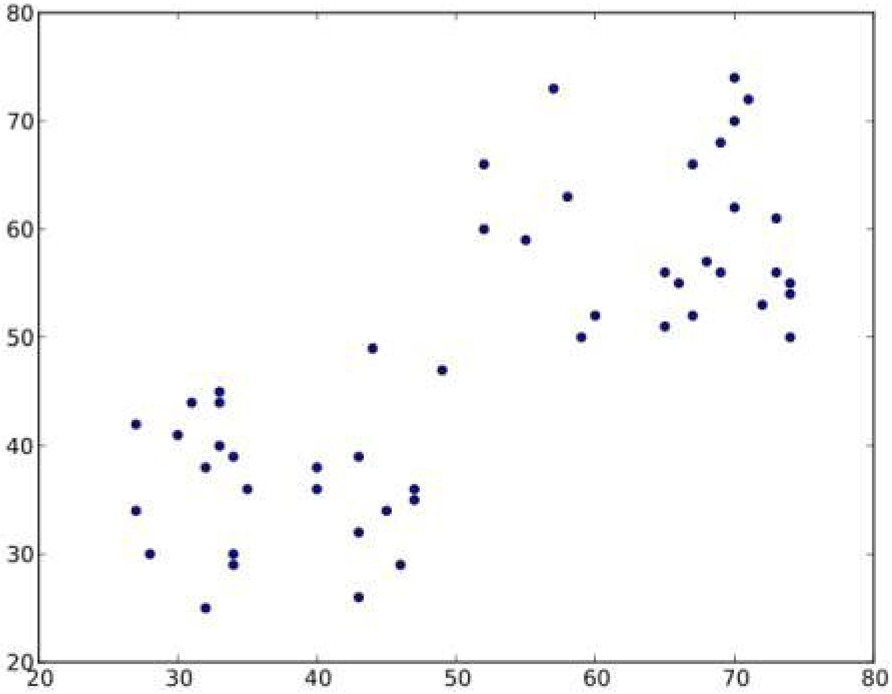
T-shirt Size Problem

Use K-means clustering!



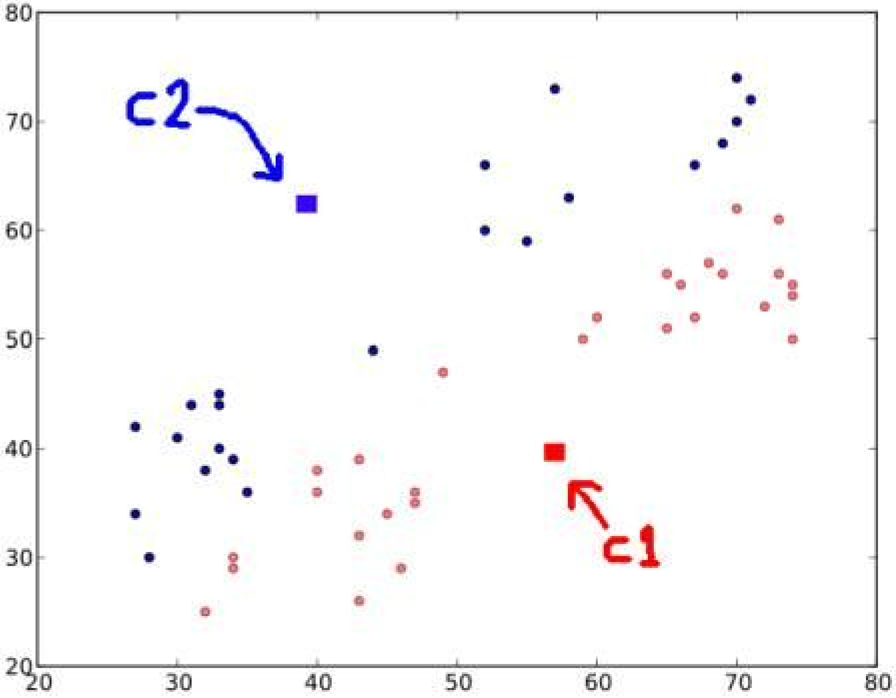
How Does It Work?

Let’s try K-means clustering step-by-step



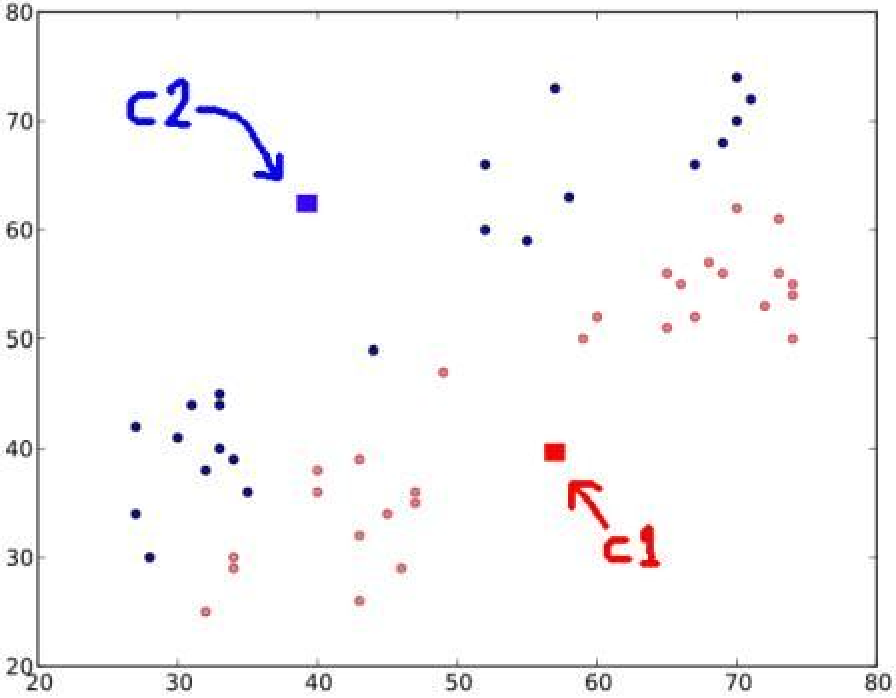
How Does It Work?

Step 1 - Randomly choose two centroids, C1 and C2



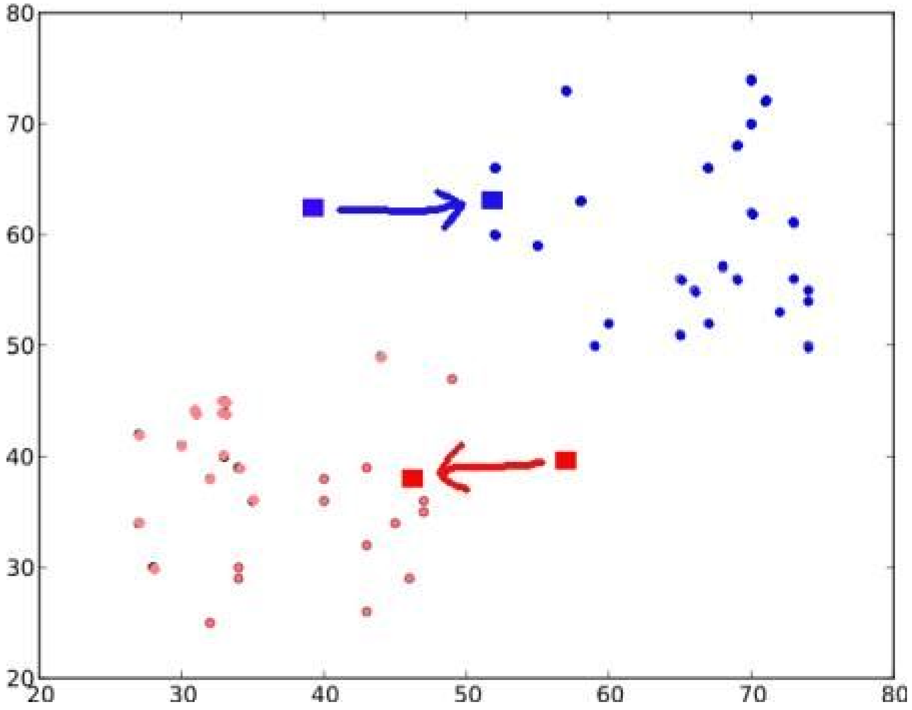
How Does It Work?

Step 2 - Calculates distance from each point to both centroids and label each datapoint to the nearest centroid's name (e.g. 'C1' or 'C2')



How Does It Work?

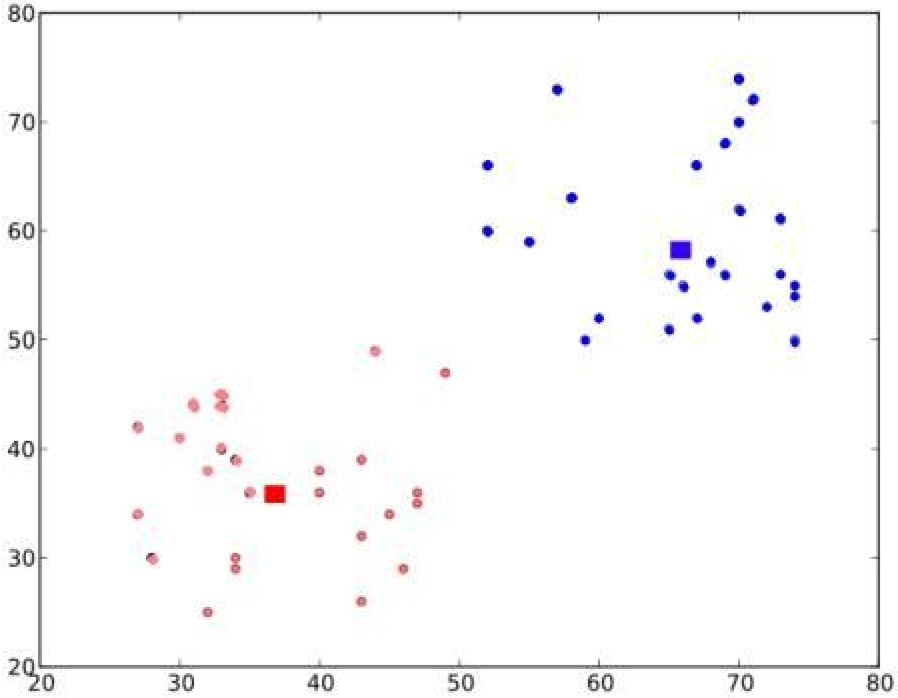
Step 3 - Calculate average of all blue points and red point separately and that will be our new centroids.



How Does It Work?

Iterate Step 2 and Step 3 until both centroids are converged to fixed points

$$\text{minimize}\left[J = \sum_{\text{All Red}_\text{points}} \text{distance}(C1, \text{Red\_Point}) + \sum_{\text{All Blue\_Points}} \text{distance}(C2, \text{Blue\_Point})\right]$$



K-Means Clustering

- K-Means in OpenCV-Python

```
compactness, labels, centers = cv2.kmeans(data, K, criteria, attempts, 
flags)
```

— Finds centers of clusters and groups input samples around the clusters

- ○ `data`: Data for clustering. An array of N-Dimensional points with float coordinates.
- ○ `K`: Number of clusters to split the set by.
- ○ `criteria`: Algorithm termination criteria
- ○ `attempts`: Number of times the algorithm is executed.
- ○ `flags`: Can take the following values
    - ■ `cv2.KMEANS_RANDOM_CENTERS`
    - ■ `cv2.KMEANS_PP_CENTERS`
    - ■ `cv2.KMEANS_USE_INITIAL_LABELS`
- ○ `compactness:` Distance between center and samples $\sum_{i} \|\texttt{samples}_i - \texttt{centers}_{\texttt{labels}_i}\|^2$
- ○ `labels:` Cluster indices for every samples
- ○ `centers:` Cluster centers, one row per each cluster center

Let’s Check the Code

`1_clustering.ipynb`

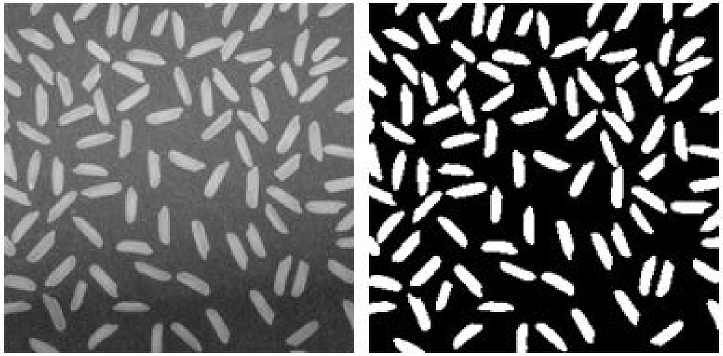
Segmentation

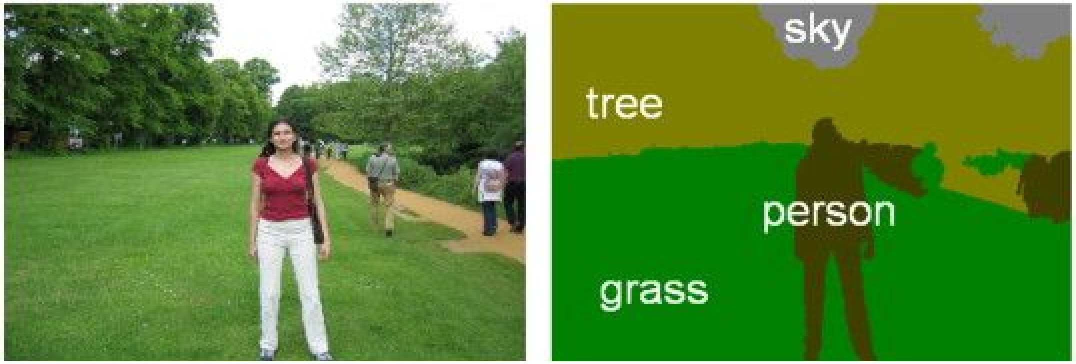
Watershed Algorithm

Content credit: Gunhee Kim’s CV slides, [OpenCV tutorials]

Image Segmentation

Break up the image into meaningful or perceptually similar regions



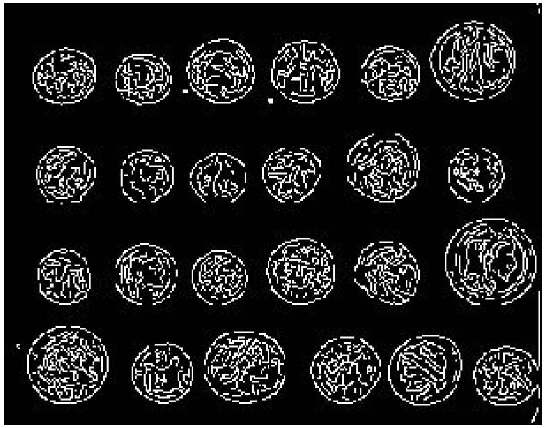


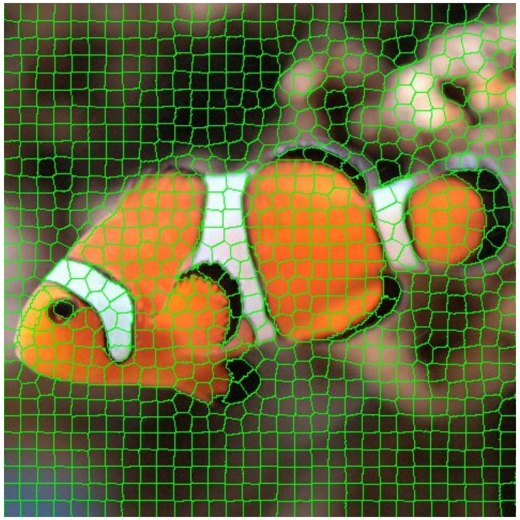
Clustering vs Segmentation

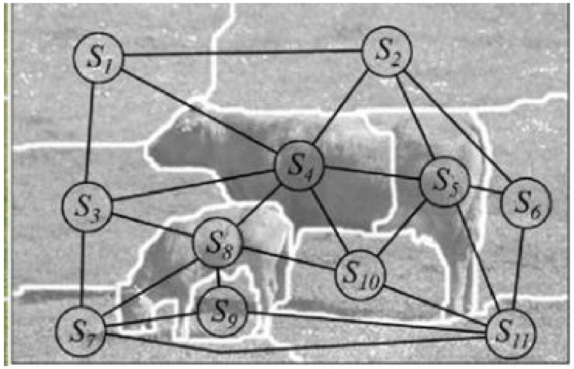
Clustering: grouping together similar points, images, feature vectors, etc.

Segmentation: dividing the image into meaningful regions

- Segmentation by clustering
- Graph based segmentation
- Segmentation from boundaries





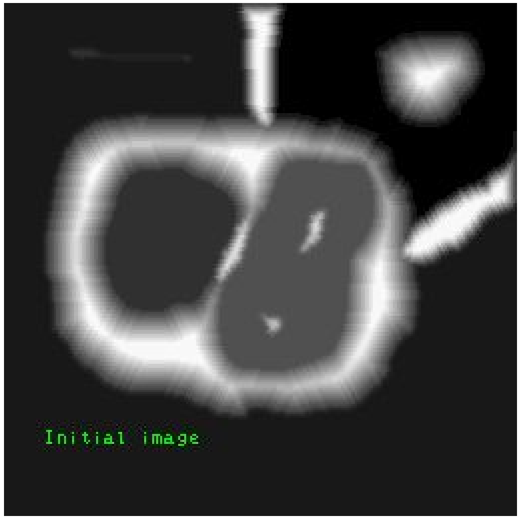


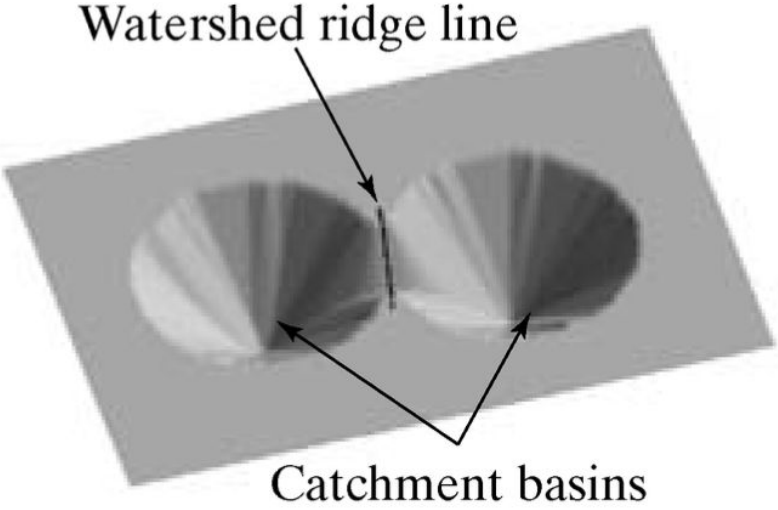
Watershed Algorithm

Instead of an image itself, working on its gradient image

Three types of points

- (a) Regional minima
- (b) Catchment basin / Watershed of a regional minimum
- (c) Divide lines / Watershed lines (Crest lines)





Watershed Algorithm

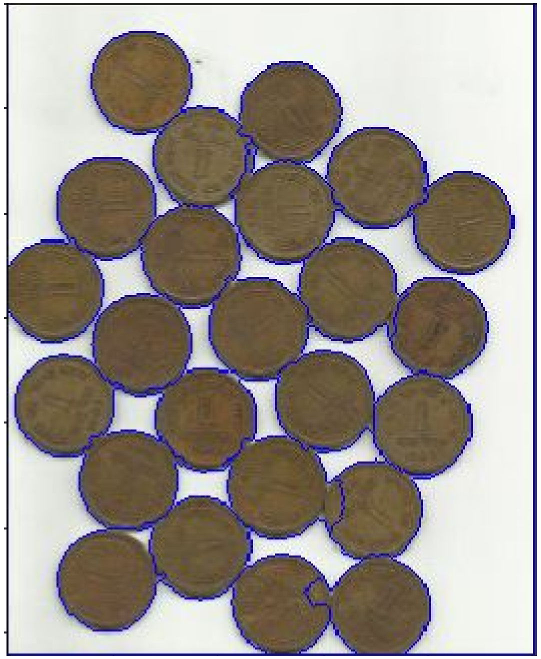
- Watershed algorithm in OpenCV-Python

```
markers = cv2.watershed(img, markers)
```

– Perform marker-based watershed algorithm

○ `img`: Input 8-bit 3-channel image

○ `markers`: Input/output 32-bit single-channel image (map) of markers.



Let’s Check the Code

`2_segmentation.ipynb`