Image Foreground/Background Segmentation using Max-Flow

Group Member：

Chengeng Liu

Shaowei Lin

Fei Gao

Siyuan Zhang

Runchuan Feng

2022/12/08

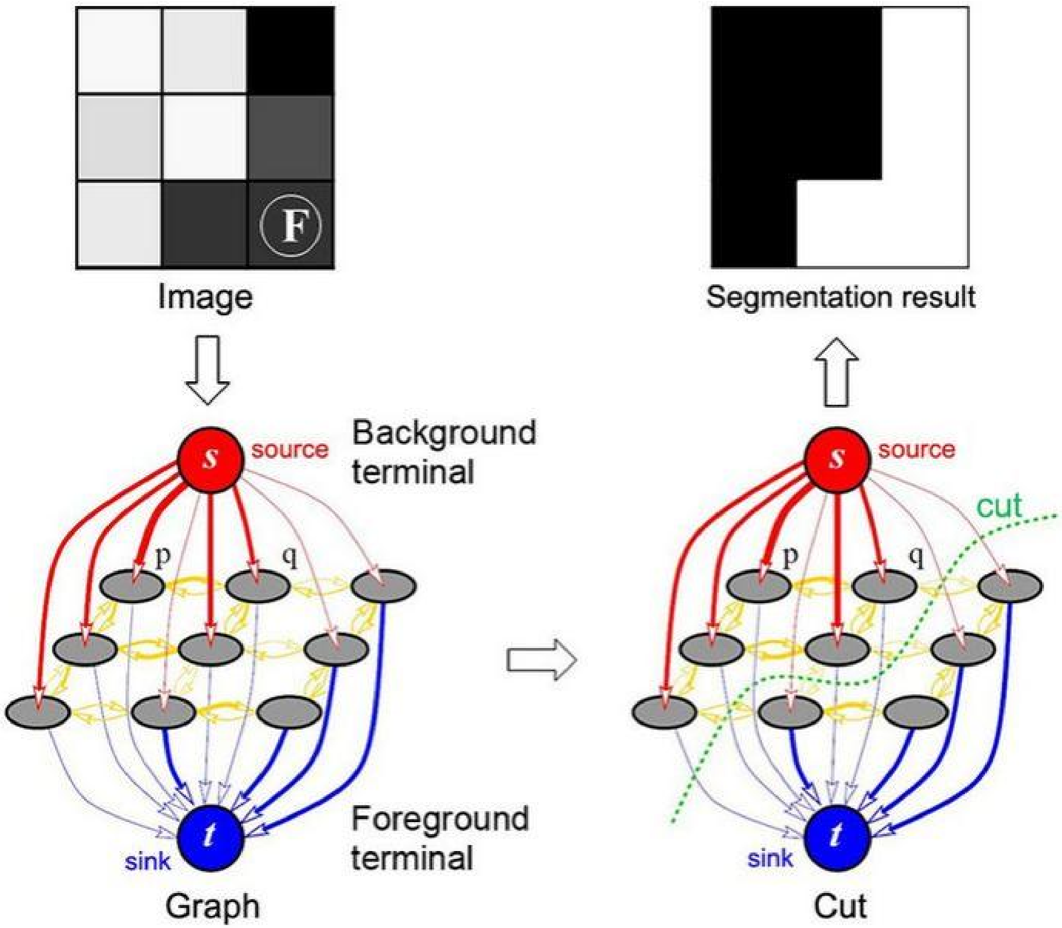
BOSTON UNIVERSITY

Problem: Segment Foreground and Background

- Image segmentation: Partitioning a digital image into multiple segments according to some certain features.
- Selecting a region of background pixels and another region of foreground pixels from the given gray scale images.

Based on Min-Cut Max-Flow Theorem

- Transform the input image into network flow graph
- Using max-flow algorithms to find the min-cut which leads to a optimal segmentation

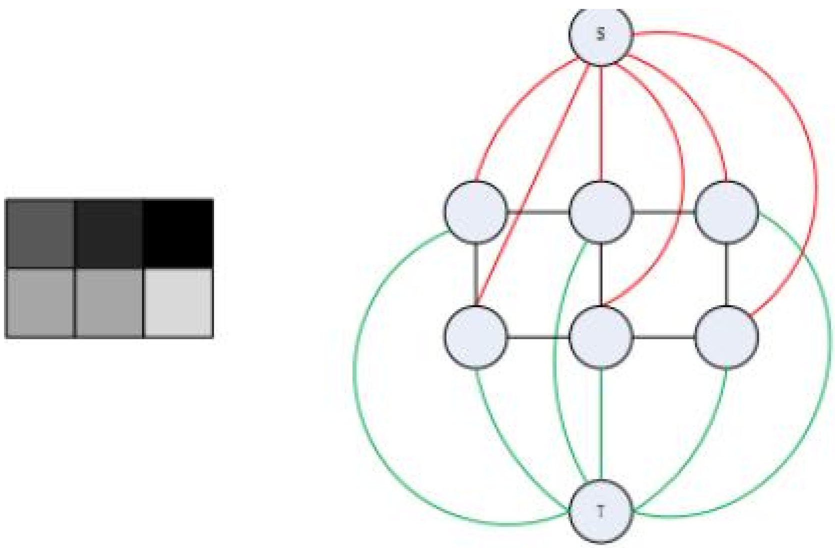


Transform Image into Graph

- 1.Each pixel in the image is considered as a node

  Two more specially designated terminal nodes: **S(source)** and **T(sink)**
- 2.Neighboring nodes are connected by edges in a regular grid-like fashion**.(n-link)**

  Another kinds of edges are used to connect nodes to terminals**(t-link)**
- 3.All graph edges are assigned some different nonnegative weight(cost)



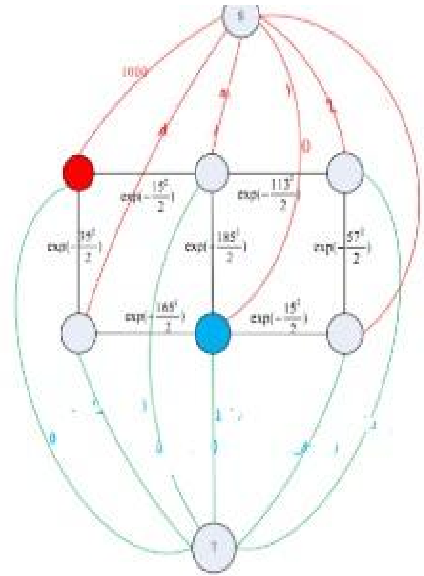
A image with 3*2 pixels and its graph

Calculation of Edge Weight

- T-links of terminals:
  - ①S-T & T-S:set to be a large value
  - ②S-S & T-T:set to be 0

- Other links: $B_{p,q}$

255	240	127
230	55	70



$$B_{p,q} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}.$$

Here, ***Ip*** and ***Iq*** are the intensities of pixels p and q respectively and σ from empirical results was chosen to be 30.

Max-Flow Algorithms

- **Edmonds-Karp （EK ）**

1. It’s an implementation of the Ford-Fulkerson method that uses **BFS** for finding augmenting paths

2. The algorithm runs in $O(VE^2)$ time.

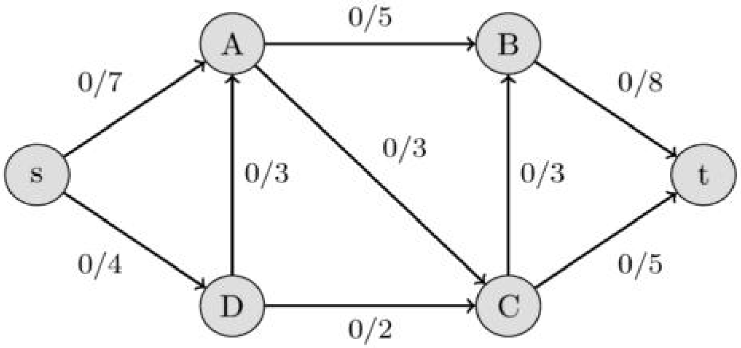
3. The flow of an edge cannot exceed the capacity. **Source s** only has an outgoing flow, **sink t** has only incoming flow.

    capacity[u][v] -= new_flow

    capacity[v][u] += new_flow;

4. When there’s no augmenting path can be found, **Current flow = max flow**.

Initial flow network



Ford-Fulkerson method

maximal flow in the flow network

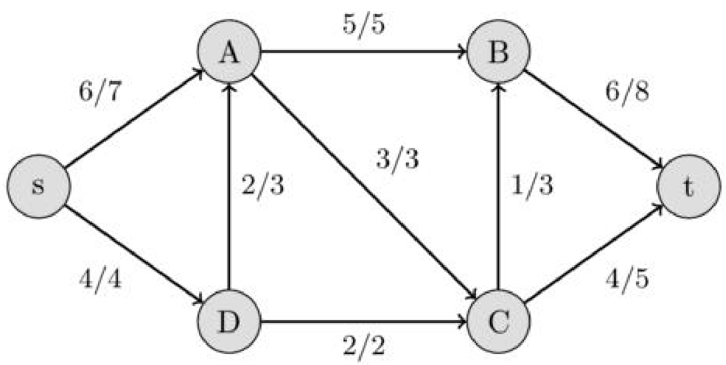
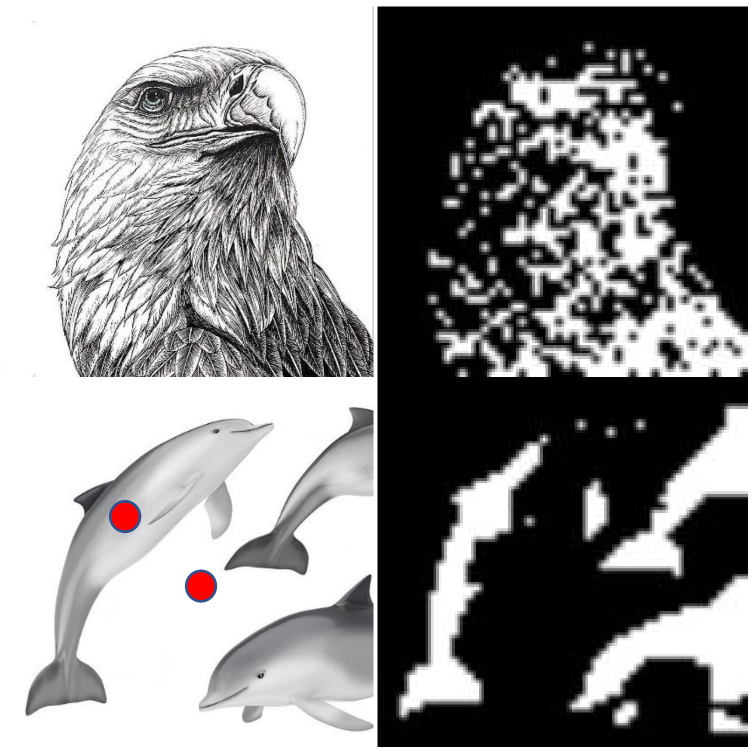
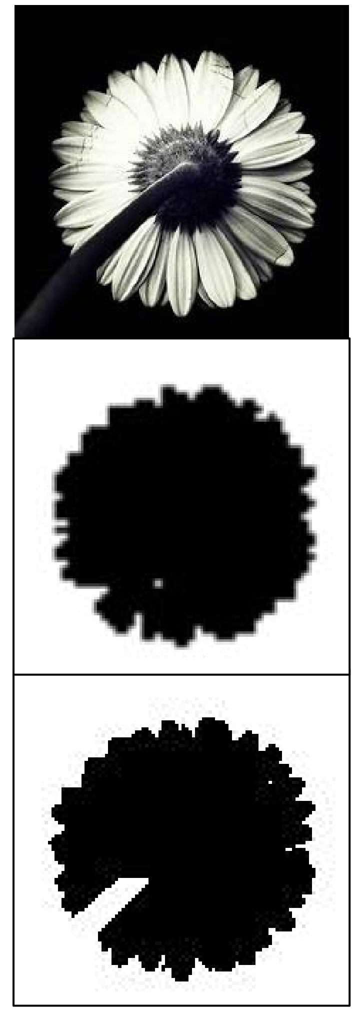


Image Segmentation





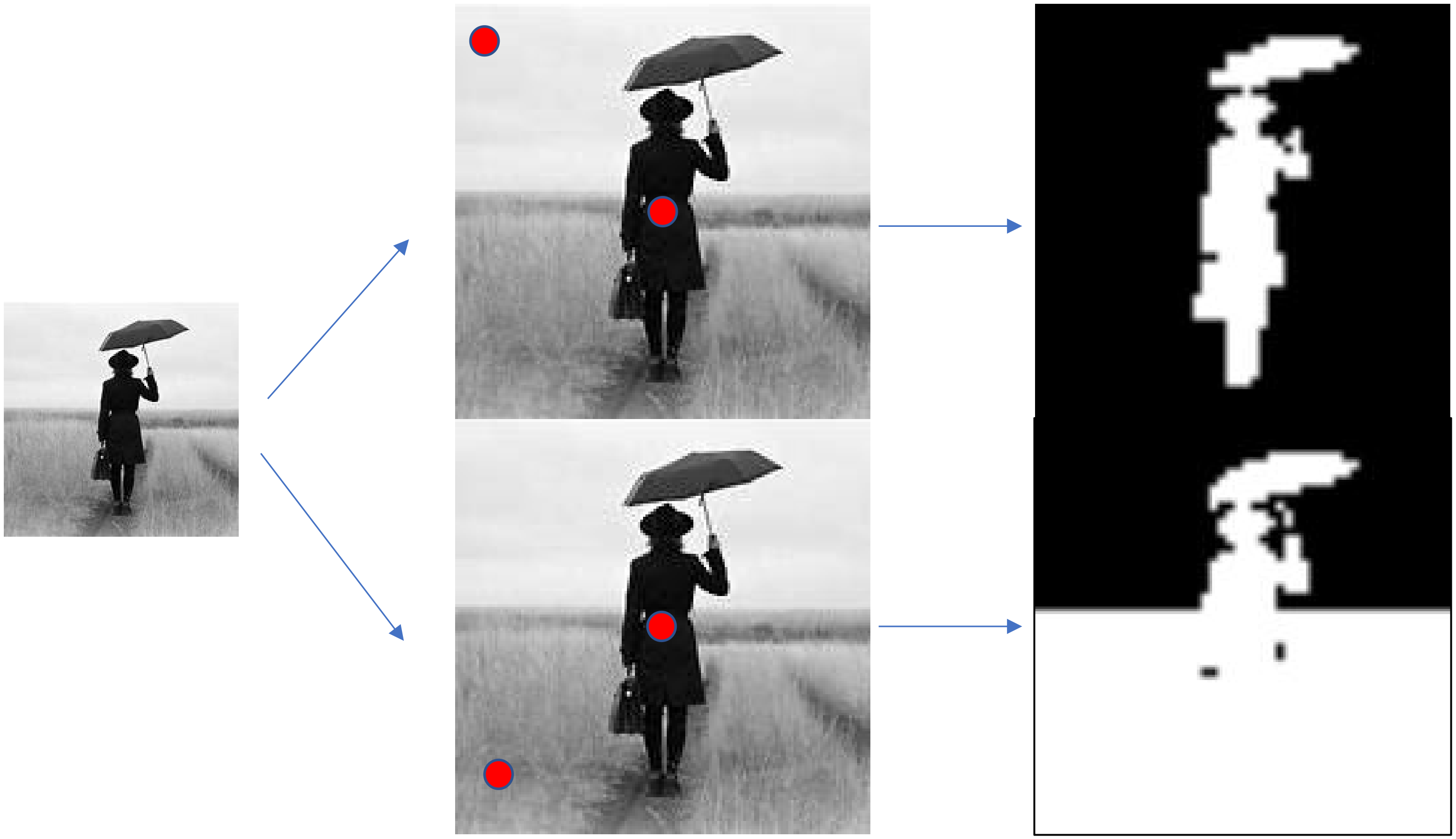
50×50

100×100

To shorten the calculation time of the algorithm, we choose to reduce the image to 50 * 50 pixels and then segment.

Image Segmentation

Different source/sink point choose



Compare with other algorithm

- **K-means algorithm**

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

Repeat {

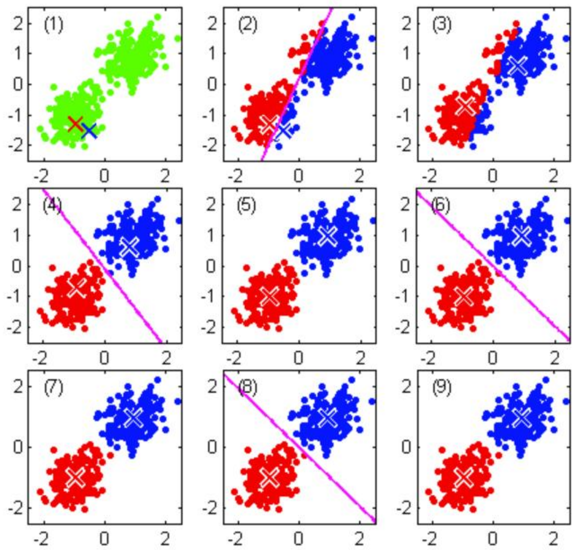
for i = 1 to m

$c^{(i)}$:= 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

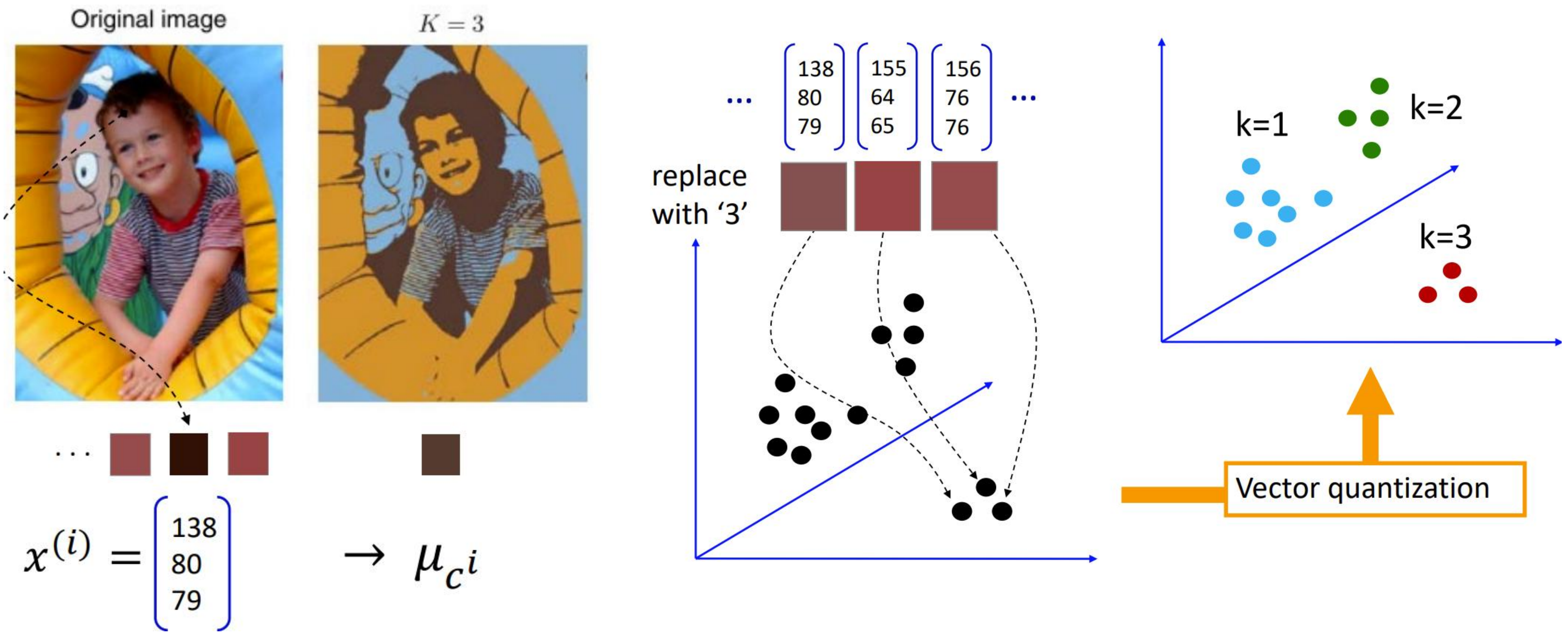
}



Input:

- K (number of clusters)
- Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

K-means For Image Segmentation



- Compress an image using clustering
- Each {R, G, B} pixel value is an input vector $x^{(i)}$
- Cluster into K clusters (using k-means)
- Replace each vector by its cluster's index $c^{(i)}$ (K possible values)
- For display, show the mean μ_{c^i}

K-means For Image Segmentation

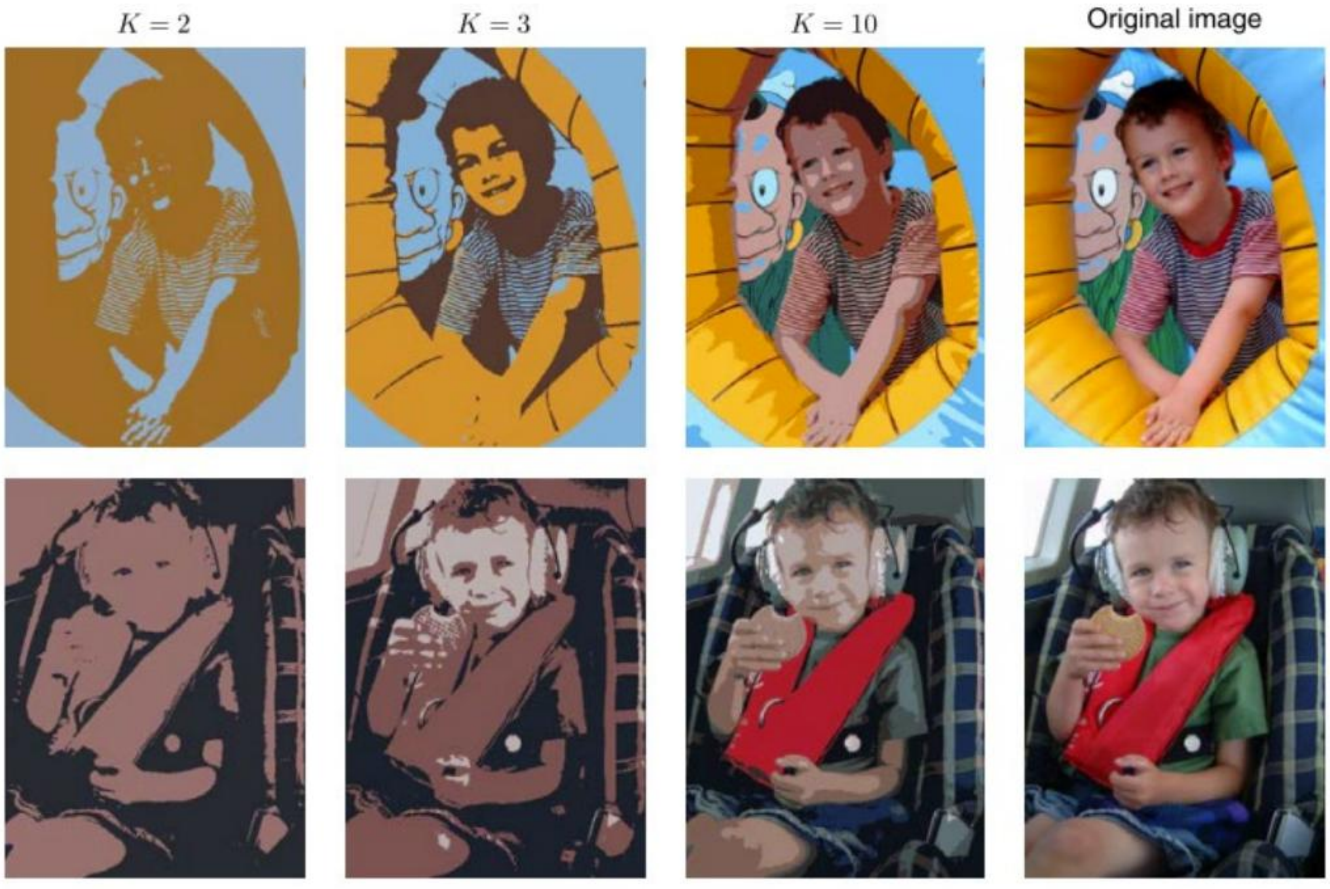
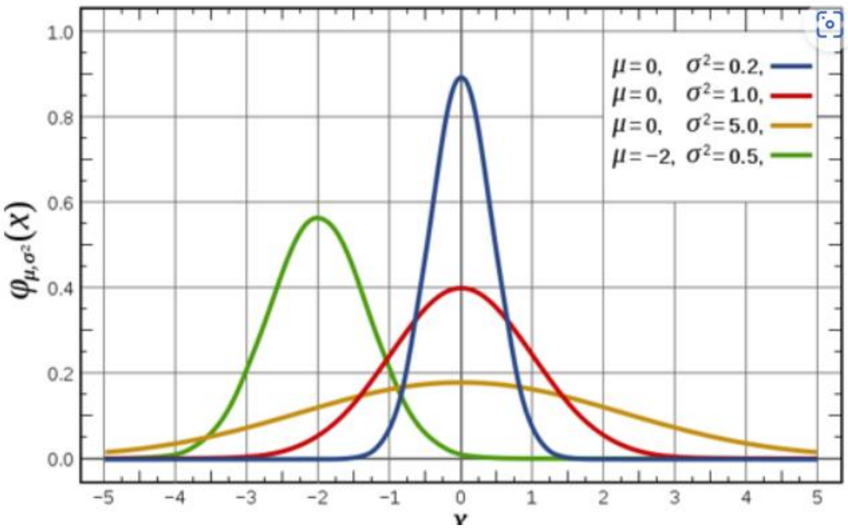


Figure 4: Two example of the application of the K-means clustering algorithm to image segmentation showing the intial images together with their K-means segmentations obtained using various values of K.

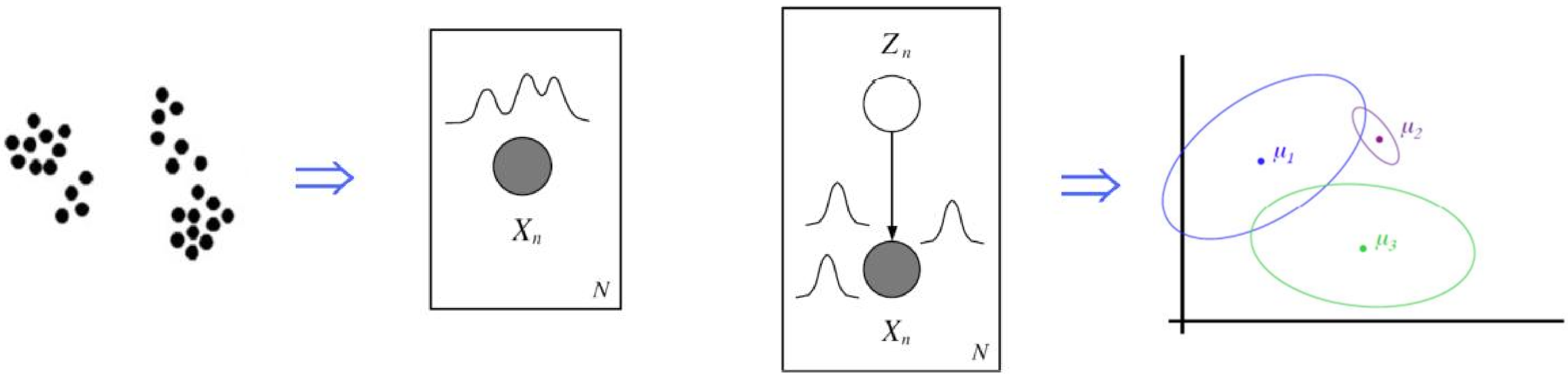
Compare with other algorithm

- Gaussian Mixture Model
- All data are generated by one of the gaussian distribution

$$g(x,\mu,\sigma)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/(2\sigma^2)}$$



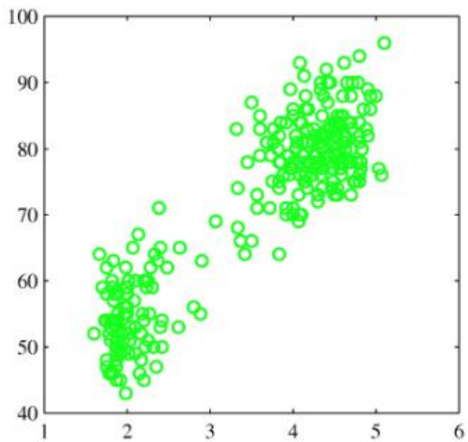
Gaussian Mixture Model

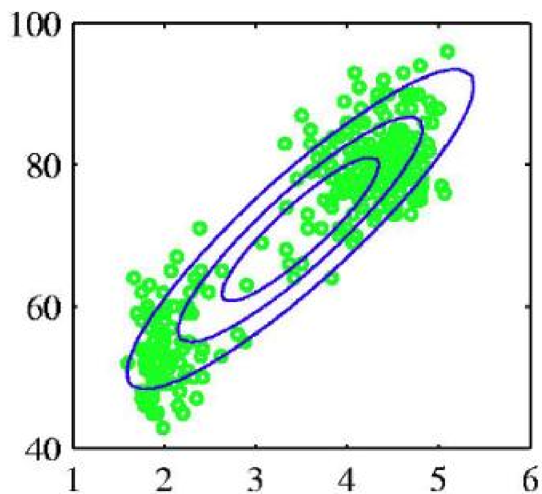


(a) (b) (c) (d)

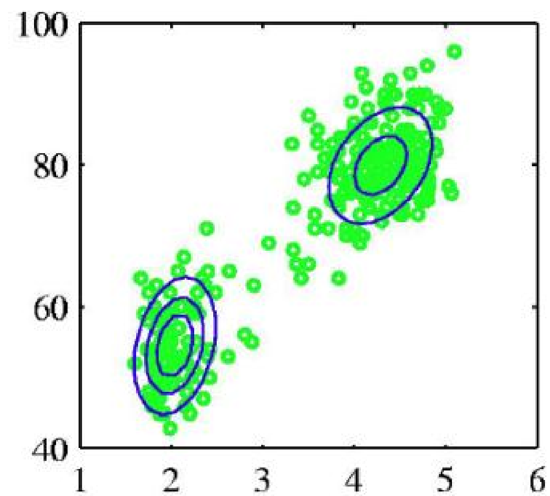
Gaussian Mixture Model

Use Expectation-Maximization method to train





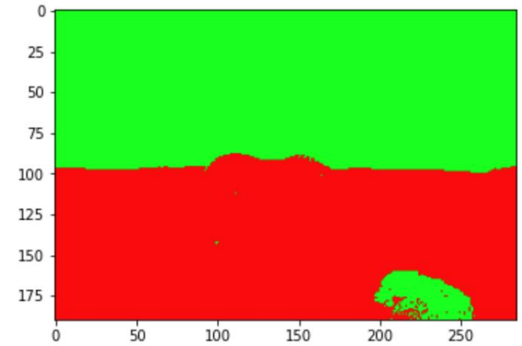
Single Gaussian



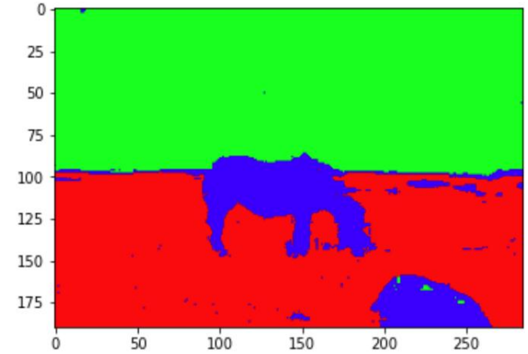
Mixture of two Gaussians

Gaussian Mixture Model

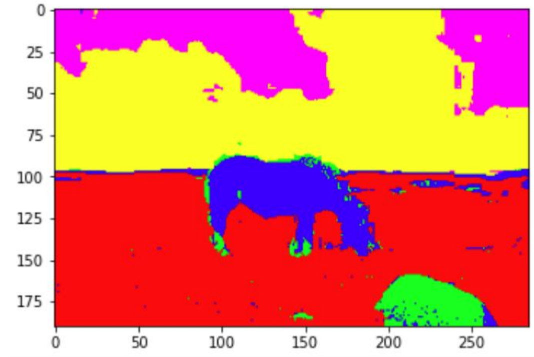
(original)



(K=2)



(K=3)



(K=5)

Thank You

BOSTON UNIVERSITY