Assignment – 4

2018701022

1. Implement the GrabCut algorithm

Description:

1. GrabCut is a image segmentation algorithm used for foreground extraction(Binary Image Segmentation). The basic idea behind this algorithm is foreground extraction from the background and remove the background.
2. Starting with a bounding box around the object which is to be segmented in the input image. Outside of the bounding box will be consider as background. Inside the bounding box is unknown. Sometimes it may be combination of both foreground and background.
3. With the help of user interaction strokes, the background will be removed inside the bounding box.
4. Basically what it does is, estimate the color distribution between the foreground object and the background using Gaussian Mixture Model(GMM). The GMM learns and creates labels for the unknown pixels and each pixel is clustered in terms of color statistics, create new pixel distribution.
5. A graph is built from this pixel distribution. Nodes in the graphs are pixels. Additional two nodes are added, **Source node** and **Sink node**. Every foreground pixel is connected to Source node and every background pixel is connected to Sink node. The weights of edges connecting pixels to the Source node and to the End node are defined by the probability of a pixel being in the foreground or in the background.
6. Based on the color similarity weight assigned to edges. The weight of the edge low, if the similarity less and the weight is high, if the dissimilarity low.
7. The min-cut method is applied on this graph. The algorithm segments the graph into two, separating the source node and the sink node with the help of a cost function which is the sum of all weights of the edges that are segmented.

8. After the segmentation, the pixels that are connected to the Source node is labeled as foreground and those pixels which are connected to the Sink node is labeled as background. This process is done for multiple iterations as specified by the user. This gives us the extracted foreground.

Algorithm:

1. Select the bounding box in input image
2. Initialize Mixture Models
3. Assign GMM components: Creating **K** components of multivariate Gaussian Mixture Models (GMM) for the two regions. **K** components for background and **K** components for foreground, total **2K** components.

Energy function

$$E(\mathbf{x}, \mathbf{k}, \boldsymbol{\theta}|\mathbf{I}) = \sum_i \varphi(x_i, k_i, \boldsymbol{\theta}|z_i) + \sum_{ij} \psi(x_i, x_j|z_i, z_j)$$

Variables

- $x_i \in \{0, 1\}$: Foreground/background label
- $k_i \in \{0, \dots, K\}$: Gaussian mixture component
- $\boldsymbol{\theta}$: Model parameters (GMM parameters)
- $\mathbf{I} = \{z_1, \dots, z_N\}$: RGB Image

Unary term $\varphi(x_i, k_i, \boldsymbol{\theta}|z_i)$

- Gaussian mixture model (log of a GMM)

Pairwise term

$$\psi(x_i, x_j|z_i, z_j) = [x_i \neq x_j] \exp(-\beta \|z_i - z_j\|^2)$$

4. Color Clustering: Calculate the GMMs the pixels need to be clustered somehow in order to determine the statistics. All pixels are placed in the same cluster in the beginning. The cluster is then split around it's mean values projection on the first principal component.
5. Calculate the mean, μ_1, and the covariance matrix Σ_1 of the cluster. Compute μ^*_n, Σ^*_n, μ_i and Σ_i

6. Graph Cut: For graph cut used the max flow min cut algorithm. The min-cut between two nodes are a way to separate the graph in two distinct parts with minimal effort by minimizing $\sum_{i \in I} w_i$ where **I** is the set of links between the nodes that were cutoff.
7. The graph consists of two parts. The first part describes how much each pixel, **m** and **n**, is connected to it's neighborhood, the N-links.

$$N(m,n) = \frac{\gamma}{dist(m,n)} e^{-\beta \|z_m - z_n\|^2}$$

8. The weight of links to the foreground and the background is the "labeling cost", L(m) for labeling a pixel m as foreground or background.

Pixel type	Background T-link	Foreground T-link
$m \in foreground$	0	$L(m)$
$m \in background$	$L(m)$	0
$m \in unknown$	$D_{fore}(m)$	$D_{back}(m)$

9. The values D_{fore} and D_{back} are function of the likelihood that the pixel **m** belongs to the foreground and background GMMs respectively.

$$D(m) = -\log \sum \pi_i \frac{1}{\sqrt{\det \Sigma_i}} e^{\frac{1}{2}[z_m - \mu_i]^T \Sigma_i^{-1} [z_m - \mu_u]}$$

Output Images:

Input Image

Output Image

Input Image

Output Image

Input Image

K = 5, Iterations = 5

K=10, Iterations =10

Input Image

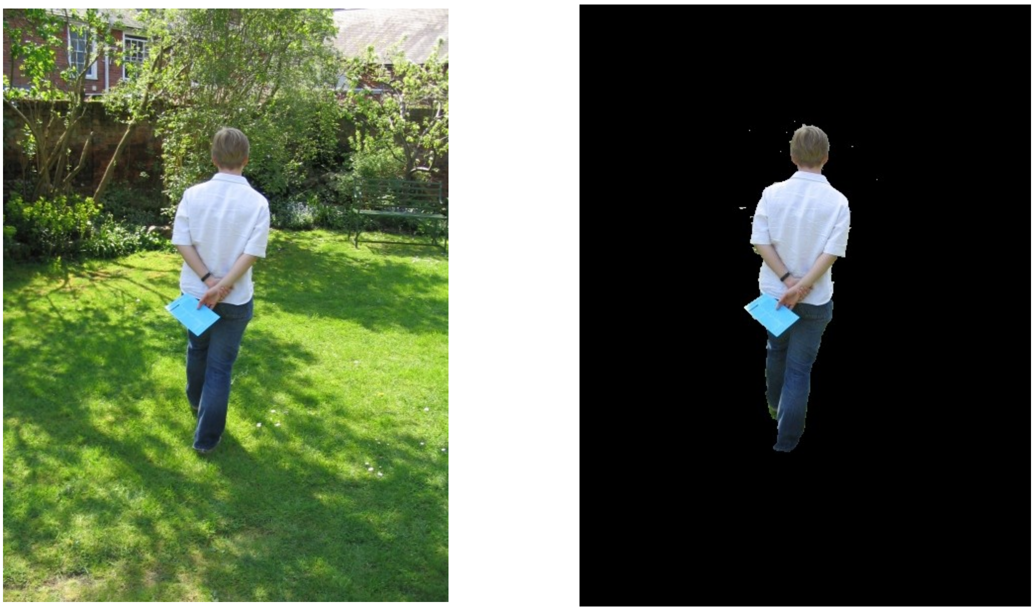
K = 5, Iterations = 10

K=10, Iterations = 5

Correspondence Points = 4

Didn't observed much difference by changing the GMM components and Iterations. But for no of iterations effecting the min cut algorithm.

If we increase the no of iterations the splitting of source node connected nodes from sink nodes more accurate.



K=5, Iterations = 10, Correspondence Points = 8

K=5, Iterations = 5, Correspondence Points = 4