Computer Vision Assignment 2 - GrabCut

Course: Computer Vision
Faculty: Avinash Sharma

Name: AadilMehdi Sanchawala

Roll Number: 20171043

The results can be obtained at the following One Drive link

https://iiitaphyd-my.sharepoint.com/:f:/g/personal/aadilmehdi_s_students_iiit_ac_in/EqonfZGN8vdElY5MchbmNE8BePoB1pn5t-vpEAk2LIo-zw?e=wmsOYo

To run the code run the following after installing the requirements,

Python main.py <path_to_image>

Introduction

In this report, we discuss an interactive implementation of the GrabCut, an image foreground background separation algorithm. We discuss the algorithm, implementation and results for the same. We vary the parameters and show the respective results as well.

The problem of efficient, interactive foreground/background segmentation in still images is of great practical importance in image editing. GrabCut is an iterative graph cut minimisation version of optimisation by modelling the problem as a conditional random field. We construct an st-graph and find the st-mincut of the graph to find the segmentation. We also learn the Gaussian Mixture Model parameters for the background and the foreground color parameters.

Graph Construction and Method

Modelling the Segmentation problem as an CRF

The paper models the segmentation problem as an energy minimisation problem wherein we try to find the segmentation between the foreground and background. We model the energy function as follows,

$$\mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z}),$$

We formulate the data term U as a GMM follows,

$$U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = \sum_n D(\alpha_n, k_n, \underline{\theta}, z_n),$$

$$D(\alpha_n, k_n, \underline{\theta}, z_n) = -\log \pi(\alpha_n, k_n) + \frac{1}{2}\log\det\Sigma(\alpha_n, k_n) + \frac{1}{2}[z_n - \mu(\alpha_n, k_n)]^\top \Sigma(\alpha_n, k_n)^{-1}[z_n - \mu(\alpha_n, k_n)].$$

We formulate the smoothness term as follows,

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n)\in\mathbf{C}} [\alpha_n \neq \alpha_m] \exp -\beta \|z_m - z_n\|^2 .$$

Finally the entire algorithm can be written as follows,

Initialisation

- User initialises trimap T by supplying only T_B. The foreground is set to $T_F = \emptyset$; $T_U = \overline{T}_B$, complement of the background.
- Initialise $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$.
- Background and foreground GMMs initialised from sets $\alpha_n = 0$ and $\alpha_n = 1$ respectively.

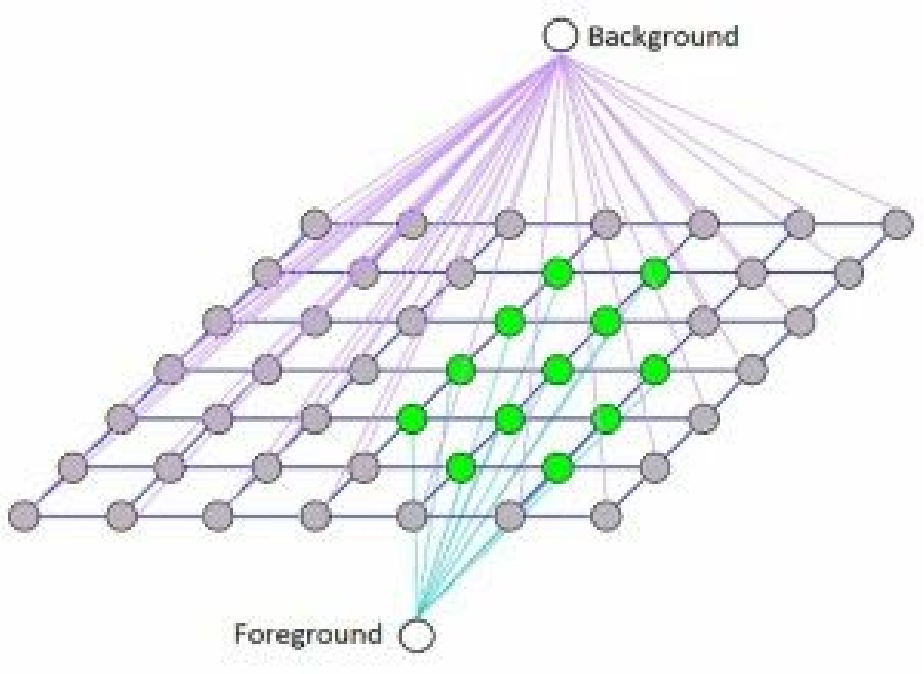
Iterative minimisation

1. *Assign GMM components to pixels:* for each n in T_U,
$$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \theta, z_n).$$
2. *Learn GMM parameters* from data $\mathbf{z}$:
$$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$
3. *Estimate segmentation:* use min cut to solve:
$$\min_{\{\alpha_n:\, n\in T_U\}} \min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}).$$
4. Repeat from step 1, until convergence.
5. Apply border matting (section 4).

User editing

- *Edit:* fix some pixels either to $\alpha_n = 0$ (background brush) or $\alpha_n = 1$ (foreground brush); update trimap T accordingly. Perform step 3 above, just once.
- *Refine operation:* [optional] perform entire iterative minimisation algorithm.

We finally construct the graph as follows,



Experiments by Varying Parameters

The number of iterations of GMM updating and energy minimisation

1 Iteration

2 Iteration

3 Iteration

4 Iteration

As the number of iterations increases, the energy minimisation better estimates the foreground background cut and separation. With every iteration, we estimate the foreground and background colors and assign them to the GMM. Then the GMM components learn the separation better in the next step of the iteration, hence, increasing the confidence and accuracy in the graph st-mincut and hence the segmentation.

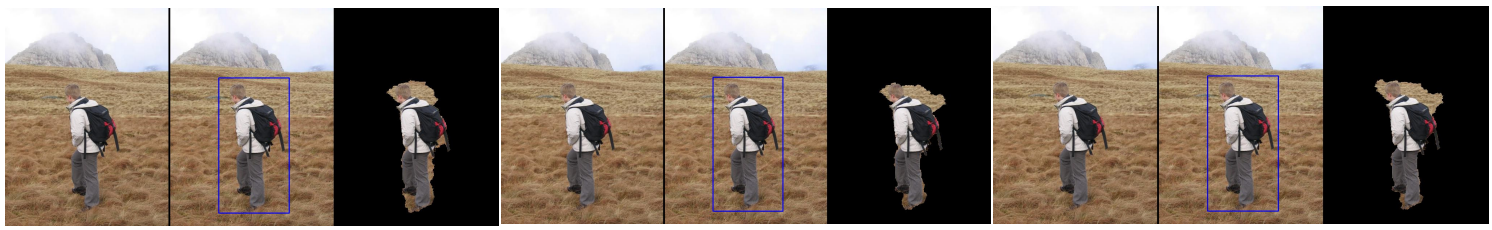
The number of mixture components in the GMM

K = 5

K = 10

K = 15

K = 20



As the number of components increase, it becomes harder for the GMM to learn the distinct separation between the foreground and the background colors. This is because there will be an overlap between the foreground and the background color components. However, when there are extremely small numbers of components, it becomes harder to represent all the colors in the foreground and the background space making the segmentation difficult. It is empirically seen that GMMs with K=5 components works the best for the various images.

4-neighborhood v/s 8-neighborhood in your pairwise term.

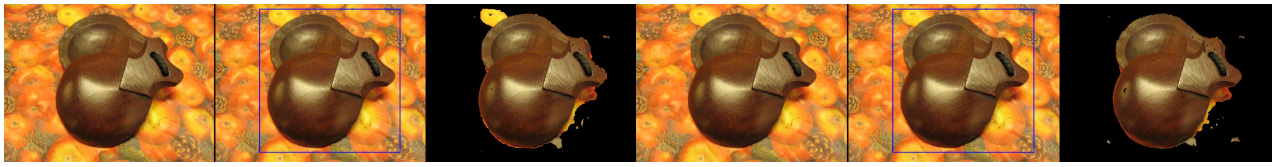
4-neighbourhood

8-neighbourhood

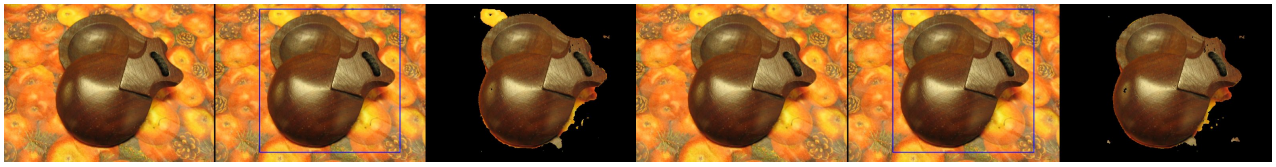
As can be seen from the results above, a 4-neighbour smoothness term takes longer time and higher number of iterations to converge, whereas the 8-neighbour smoothness takes a lesser number of iterations to converge. Moreover, the smoothness consistency is higher in the case of 8-neighbour formulation.

The choice of gamma.

Gamma = 5



Gamma = 50

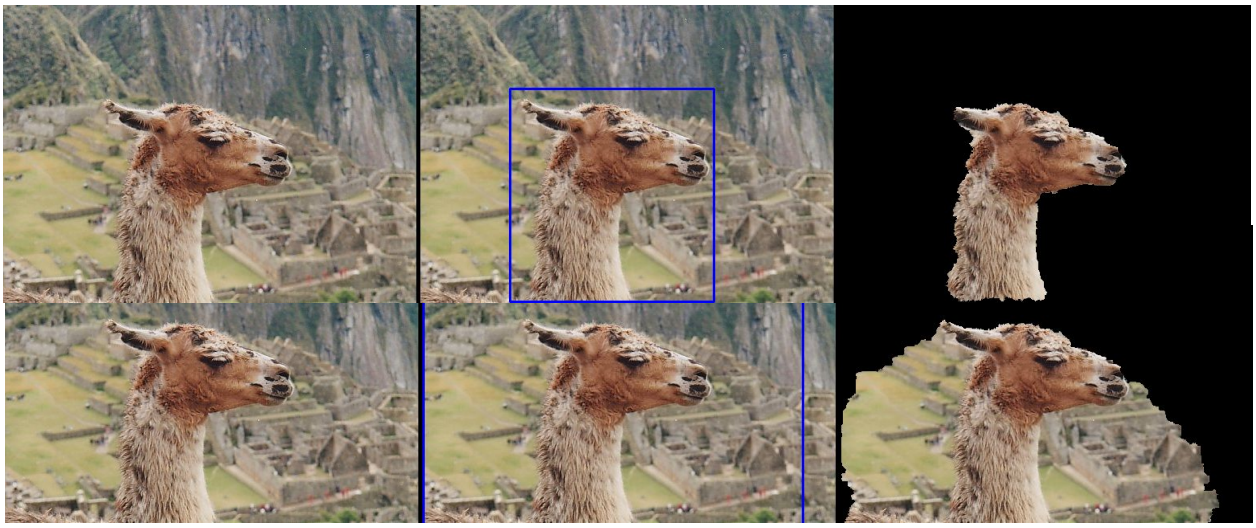


Gamma = 100

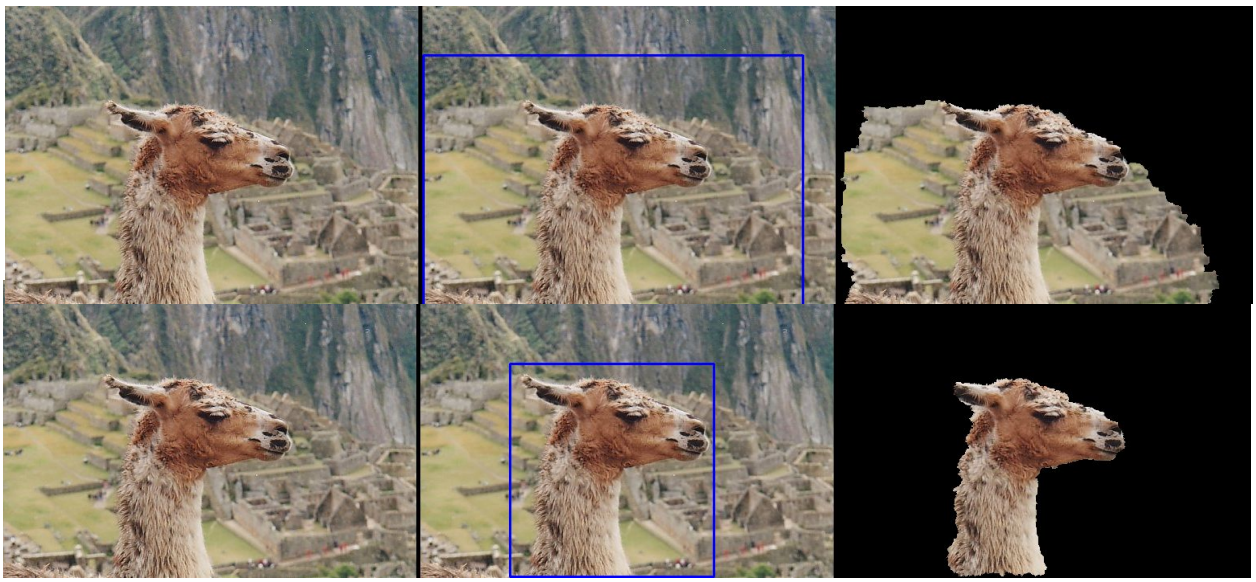
The choice of gamma affects the smoothness term of the energy equation. A low value of gamma can prevent the equation from penalising different neighbouring color intensities and a higher value of gamma can cause the penalisation to not flip correctly. As can be seen from the results, higher value of gamma penalises the difference in neighbouring pixels of different intensities much more than a smaller value of gamma. Moreover, the data term becomes more confident and corrective when we give a high value of gamma.

A tight initial bounding box or a loose bounding box.

Loose Bounding Box



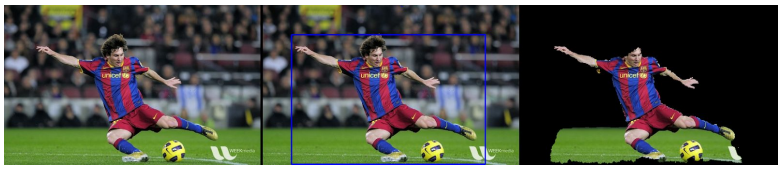
Tight Bounding Box



As can be seen from the results, a tighter bounding box provides a better segmentation than the loose bounding box. This is because we provide a better initial estimate of the foreground background GMM initialisation.

Different color spaces or ways of representing pixels.

BGR



RGB

YCbCr

HSV

LAB

More Results