# ECE 7866 Computer Vision
# Project 2

Lazy Snapping

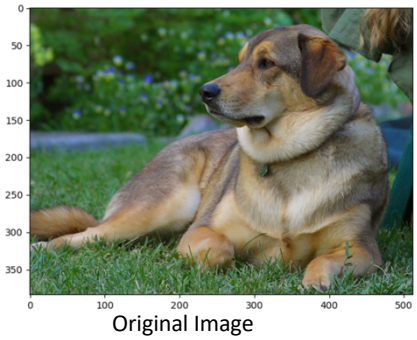
Basic Algorithm Graph Cut:

The algorithm for lazy snapping is implemented using (1) extraction of seed pixels, (2) K-means clustering and (3) Probability calculation. It takes the main image, stroke image and k value as input and displays foreground and background extraction of image by implementing the following functions: -

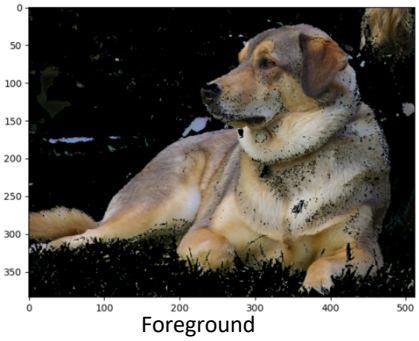
- Seed Pixel extraction: Each pixel of stroke is traversed to check if it belongs to red or blue pixels, and placed in separate arrays
- K means Clustering: In this clustering technique, centroids and indices are assigned randomly. They are updated by taking average of seed pixels for each index of centroids. The computation is done iteratively until old and new centroids become equal or total iteration exceeds 200.
- Calculation of probability: In this function, the likelihood of the pixels with respect to foreground and background region is calculated and a binary image is formed with 1 for higher probability of belonging to foreground pixel and 0 for background pixel

Results:

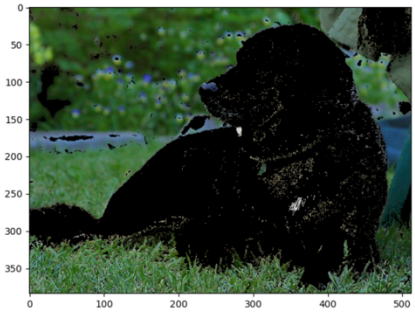
(1) Dog



Original Image



Foreground



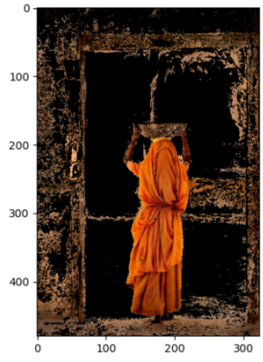
Background

(2) Lady

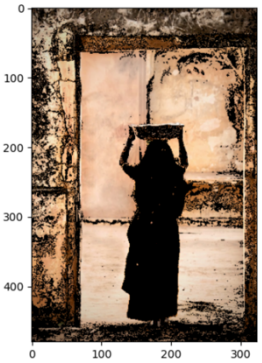
Using Lady Strokes 1 image

Using Lady Stroke 2 image

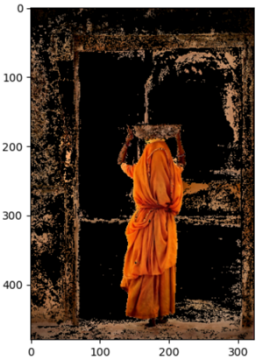
Original image



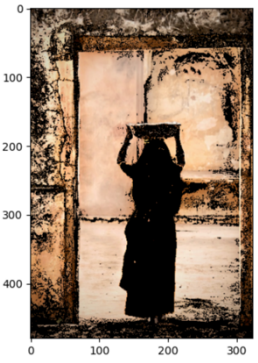
Foreground



Background



Foreground

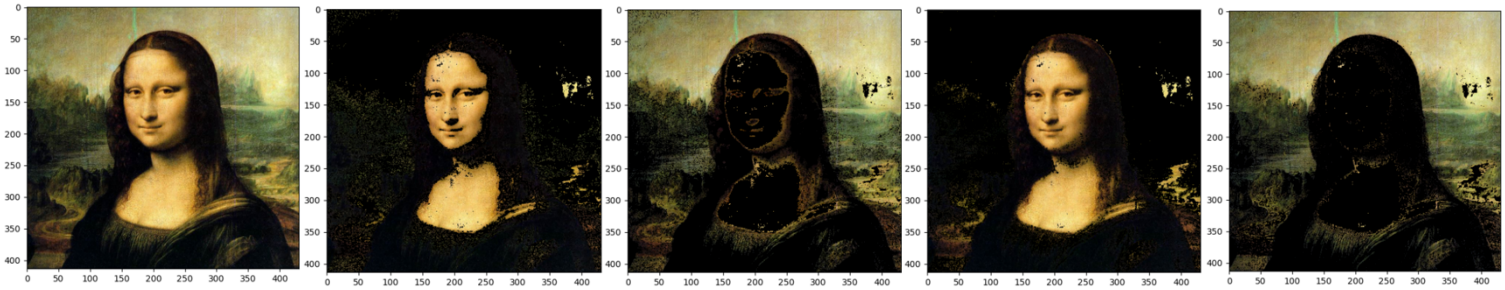


Background

(3) Mona-Lisa

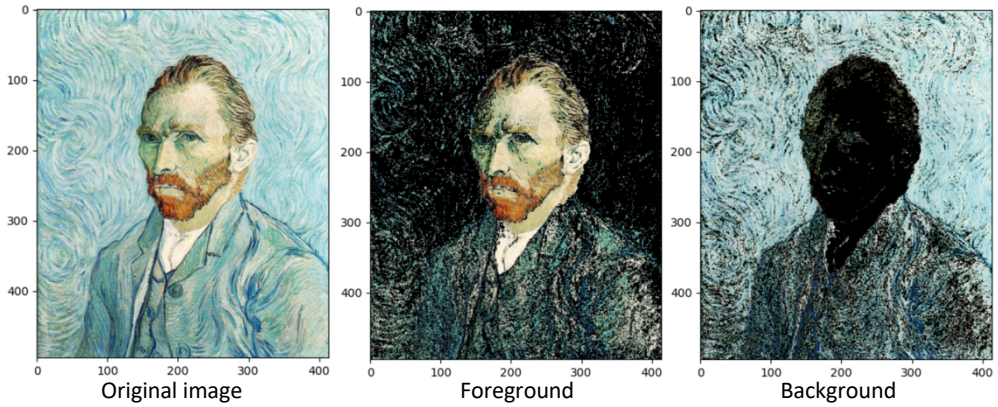
Using Stroke 1 image

Using Stroke 2 image



Original image | Foreground | Background | Foreground | Background

(4) Van Gogh

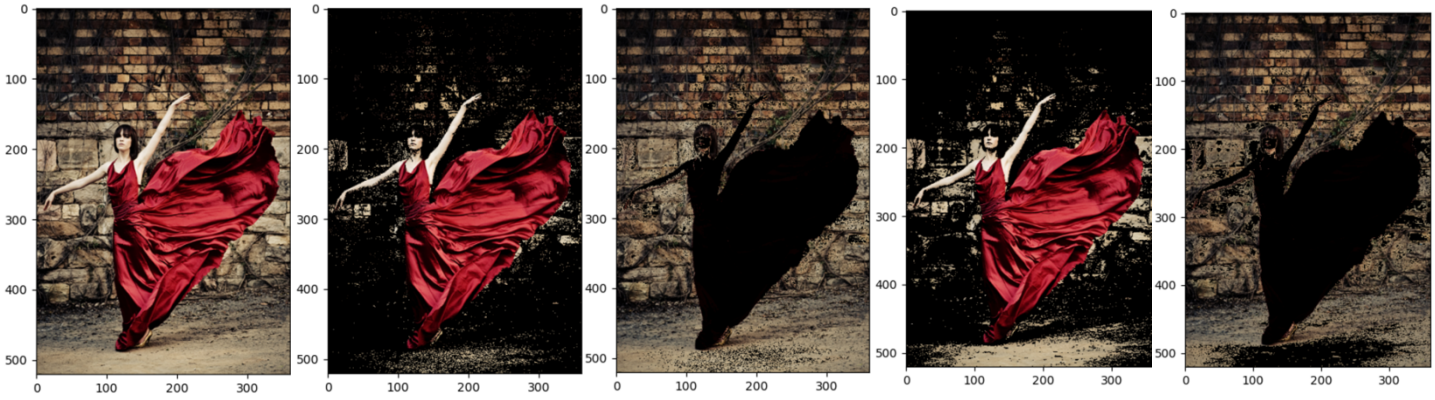


Original image | Foreground | Background

(5) Dancing Lady

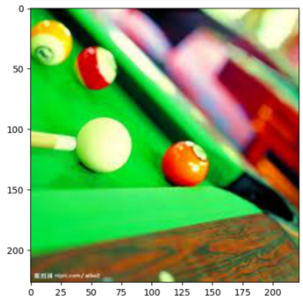
Using Stroke image 1

Using Stroke image 2

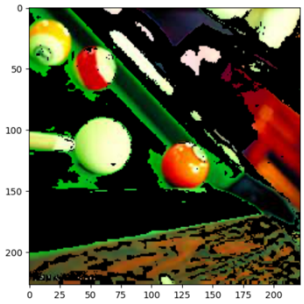


Original image | Foreground | Background | Foreground | Background

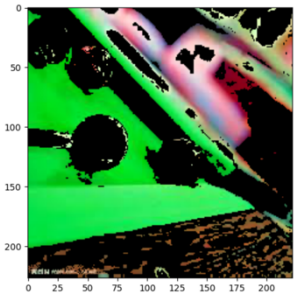
(6) Table ball



Original image



Foreground



Background

Observations:

Discussion:

From the results, it can be observed that as there are more strokes (red and blue strokes) in the stroke image, the lazy snapping technique works better and gives better results. The foreground and background pixels can be clearly distinguished from each other. Also, the algorithm does not work well when there are similar pixel values in foreground and background. This technique is not computationally intensive and gives accurate results.

Limitations:

- The algorithm does not work well on pictures with similar pixel values (colors) in foreground and background
- In case of larger images, the boundary is not smooth
- For smaller areas, the image details are lost, and accurate information cannot be retrieved

Improvements:

- As the algorithm for lazy snapping uses K-means for clustering foreground and background pixel values, in case if there are similar pixels in foreground and background, the clustering algorithm works poorly. To improve this, the number of similar pixels can be reduced for foreground and background
- For improving efficiency and speed, we can use watershed algorithm to graph cut formulation for dealing with segmented areas
- Different segmentation algorithms can be used instead of watershed segmentation algorithm to find optimal solution for pre-segmentation. This technique is called augmented lazy snapping

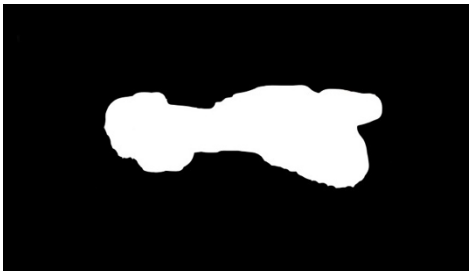
Poisson Image Blending

Poisson image blending is an efficient algorithm used for seamlessly blending two images together. It takes a source and target and a mask image as input. The mask image is used for extracting the foreground object from the source image, allowing some margin and the algorithm is used for blending and reconstructing the final image.

Results:

(1)

Source image



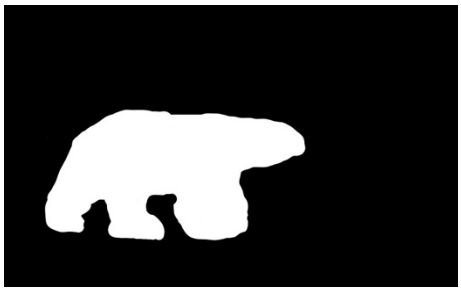
Mask Image

Target image

Final image

(2)

Source image



Mask Image

Target image

Final image

Observations:

Discussion:

In this method, the mask image is used for obtaining the foreground object from the source image. Laplacian matrix and Poisson equations are used in the algorithm for combining the source object onto the target background image. This resulted in the final image where the source object is placed onto the target image and its edges are blended to remove any discrepancies. This technique gives unnatural colors for the bear image as it does not maintain color consistency in the source image.

Limitations:

- Poisson image blending algorithm works best and produces accurate results when the source and target images have similar pixel values in background
- It does not affect the target image and includes intensity changes only in source image. Hence, color consistency is not always maintained while doing optimization of color in the source image. It tries to maintain similar color contrast for blended image as that of original image, and this gives rise to unnatural colors in the final blended image
- Although this algorithm solves discontinuity in color to overcome visible seams, it does not take texture discontinuity into consideration

Improvements:

- For Poisson image blending, as the source image is only edited, the color consistency is not always maintained. Therefore, if a term is added to the minimization problem that takes care of the consistency in color, the algorithm can be improved, and better results can be obtained
- Changes in the minimization objective can be used to consider the variations in source and target image regions, which in fact results in reduction in color distortions

Bonus

Interactive Digital Photomontage:

The process uses the Poisson image blending technique on multiple source images, so that a final target image can be reconstructed using foreground pixels from the different input source images and blending together. In this method, the Poisson image bending functions from the previous section is executed iteratively for a set of input images.

Result:

There are some issues with the coding and hence the blending is not as accurate as expected. The green pixels are taken as foreground from image 0176 and photomontage process is run to produce final image. The two foreground faces are distorted and hence the implemented functions are not working as expected.