Adva Helman
206087900

Noam Blau
207145293

Foreground Segmentation and Poisson Blending

GrabCut report

Results table: (5 components)

Img name	Accuracy	Jaccard	Time (sec)
banana1	0.6691	0.4359	63.44
banana2	0.9887	0.9529	24.22
book	0.978	0.9438	114.31
bush	0.8815	0.5837	24.45
cross	0.5074	0.3632	48.44
flower	0.9965	0.9825	14.84
fullmoon	0.9964	0.9443	6.79
grave	0.9889	0.9135	12.55
llama	0.9903	0.9451	16.74
memorial	0.9898	0.9454	19.32
sheep	0.9957	0.9225	10.36
stone2	0.9961	0.9842	28.79
teddy	0.9933	0.969	11.34
		Average time per image	30.43

Why we chose a relative delta for the threshold?

The threshold cannot be an absolute value because the absolute value of the segmentation error may vary widely between images and datasets. An absolute threshold is not suitable because it cannot account for the variability in the dataset.

Therefore, a relative delta criterion is used instead. A relative delta criterion is calculated as a percentage of the segmentation error in the previous iteration. Using a relative delta ensures that the iteration stops when the rate of improvement of the segmentation result slows down, rather than when the error itself reaches a fixed value.

The effect of blur

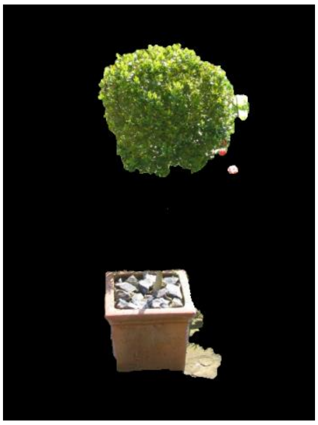
Bush (5 components)

Without:

Accuracy=0.8815

Jaccard=0.5837

Low:



Accuracy=0.9763

Jaccard=0.8733

High:

Accuracy=0.9698

Jaccard=0.8423

Cross (5 components)

Without:

Accuracy=0.5074

Jaccard=0.3632

Low:

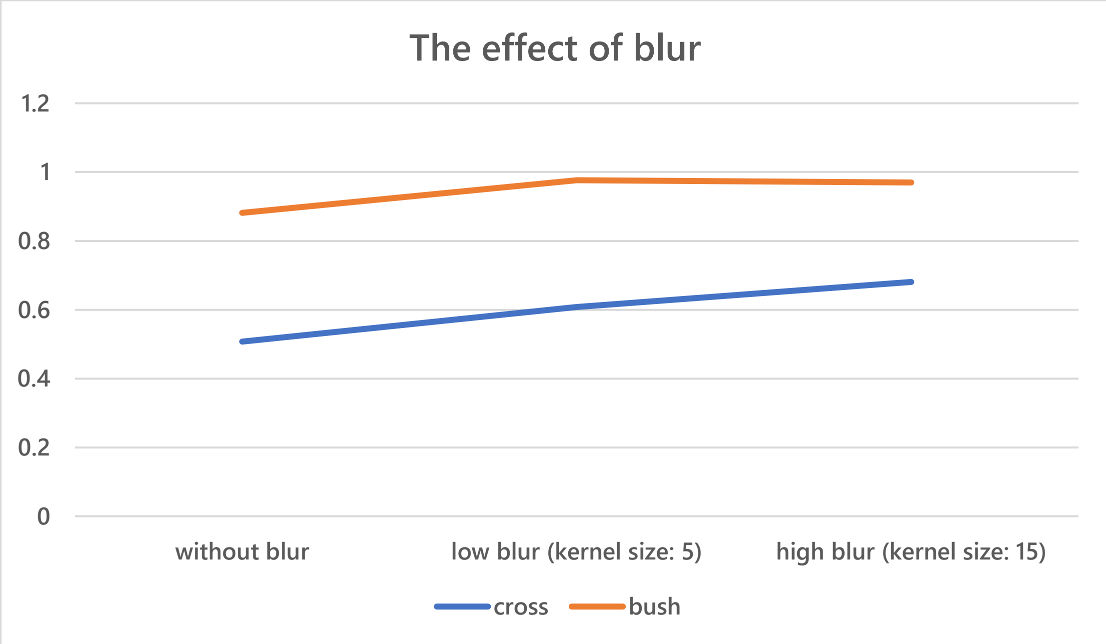
Accuracy=0.6078

Jaccard=0.417

High:

Accuracy=0.6807

Jaccard=0.535



In the GrabCut algorithm, blurring the image can have both positive and negative effects on the segmentation results.

Blurring the image can help to reduce noise and smooth out the image, which can make the segmentation more robust and less sensitive to small variations in the image. This can improve the overall segmentation accuracy, especially if the image is noisy or contains a lot of texture, as we can see the difference between the bush without blur and the low blur.

On the other hand, blurring the image can also cause loss of detail and reduce the sharpness of the edges in the image. This can lead to under-segmentation, where the algorithm fails to detect fine details and boundaries in the image, as we can see the difference between the bush with the low blur and the high blur around the bush top.

In our case, using a Gaussian blur with a kernel size of 5 is helping to reduce noise and smooth out both images, which is leading to a more accurate segmentation. However, using a larger kernel size of 15 can also cause loss of detail and reduce the sharpness of the edges in the image, which lead to under-segmentation in the bush image.

The number of GMMs

Banana1

5 components:

Accuracy=0.6691 Jaccard=0.43596

1 component:

Accuracy=0.9647 Jaccard=0.879

Cross

5 components:

Accuracy=0.5074 Jaccard=0.3632

1 component:

Accuracy=0.9874 Jaccard=0.9671

Adva Helman
206087900

Noam Blau
207145293

Banana2

5 components:

Accuracy=0.9887 Jaccard=0.9529

7 components:

Accuracy=0.9867 Jaccard=0.9463

The number of components in the Gaussian Mixture Model (GMM) can have a significant impact on the quality of the segmentation. Generally, using a larger number of components in the GMM can lead to a more accurate segmentation, but at the cost of increased computational complexity.

When using a small number of components in the GMM, the model may not be able to capture the complexity of the image color distribution, leading to under-segmentation, where some pixels that belong to the foreground or the background are misclassified. For example, using 5 components for banana2 considers the top of the banana as background, but using 7 components captures the complexity of the banana color distribution more precise.

On the other hand, when using many components in the GMM, the model may overfit the image color distribution, leading to over-segmentation, where the image is segmented into too many regions, including some regions that do not belong to the foreground or the background. For example, using 5 components for cross considers the garden as background, but using 1 component get the correct segmentation. Another example, using 5 components for banana1 considers part of the desk as foreground, due to a simple color distribution.

Therefore, the optimal number of components in the GMM depends on the specific image and the complexity of its color distribution. For images with more complex color distributions, we may need to use a larger value of components to capture the finer details of the image.

Different initialization of the rectangles

Banana1 (5 components)

rect=(16, 20, 620, 436):

Accuracy=0.6691 Jaccard=0.43596

rect=(30, 35, 605, 420):

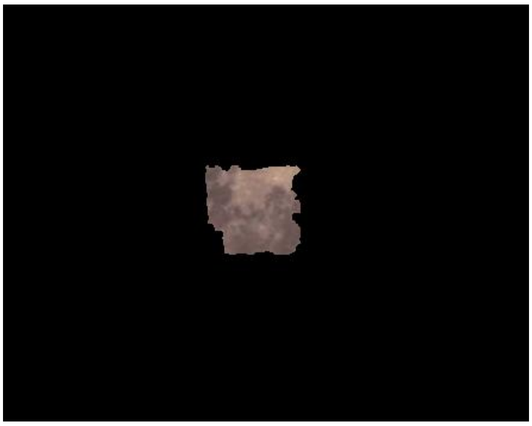
Accuracy=0.721 Jaccard=0.4766

Fullmoon (5 components)

rect=(150, 114, 271, 234):

Accuracy=0.9964 Jaccard=0.9443

rect=(170, 135, 250, 210):



Accuracy=0.973 Jaccard=0.5525

The initialization of the rectangles in the GrabCut algorithm can have a significant impact on the quality of the segmentation. The initial rectangle is used to define the foreground and background regions in the image, which are then used to initialize the GMM parameters for the foreground and background models.

If the initial rectangle is too small, some pixels that belong to the foreground or the background may be misclassified, leading to under-segmentation or over-segmentation. As we can see in the fullmoon image.

On the other hand, if the initial rectangle is too large, it may include pixels that do not belong to the foreground or the background, leading to incorrect classification and lower segmentation accuracy. As we can see in the banana1 image.

Therefore, it is important to carefully choose the initial rectangle. In general, a good initialization should include enough pixels from the foreground and the background to allow the GMM models to accurately capture the color distribution of each region. However, it should also avoid including pixels that do not belong to the foreground or the background, which can lead to incorrect classification and lower segmentation accuracy.

Adva Helman
206087900

Noam Blau
207145293

Poisson blending report

Results

Banana1 + Table

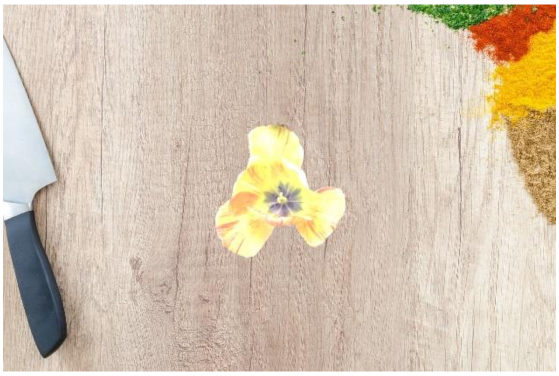
Banana2 + Table

Book + Table

Bush + Wall

Cross + Wall

Flower + Table



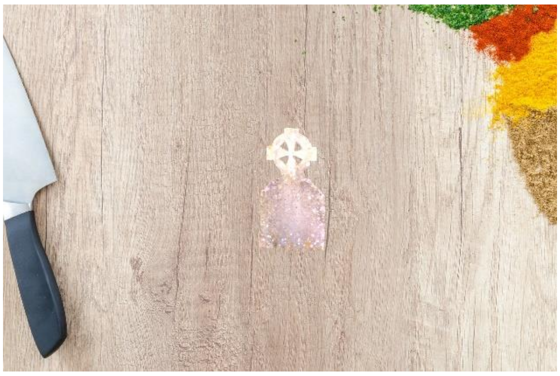
Adva Helman
206087900

Graphics - Assignment 1, April 2023

Noam Blau
207145293

Fullmoon + Wall

Grave + Table

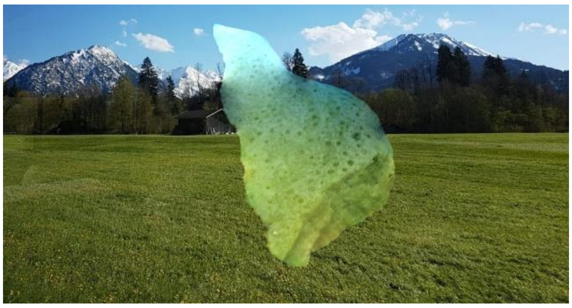


Llama + Grass Mountains

Memorial + Wall

Sheep + Wall

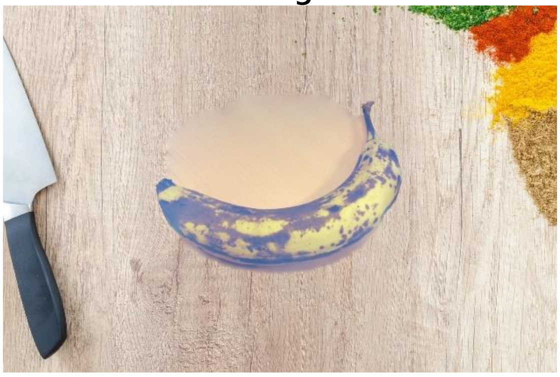
Stone2 + Grass Mountains



Teddy + Grass Mountains

What happens if the mark is not tight around the object

Banana1 + Table

not tight	tight
	

Cross + Wall

not tight	tight
	

In the Poisson blending algorithm, a tight mask is used to indicate the region of interest (foreground) in the source image. The mask specifies which pixels in the source image should be blended into the target image, and which pixels should be left unchanged (background).

When we do not use a tight mask, then the Poisson blending algorithm may produce artifacts or incorrect blending results. For example, the mask of banana1 includes table parts and cross includes sky parts. Those regions do not belong to the object of interest – the banana or the cross. Then those regions are blended into the target image and create a visible seam or boundary between the blended and non-blended regions.

Therefore, it is important to use a tight mask that accurately delineates the object of interest in the source and target images. This will ensure that the Poisson blending algorithm blends only the desired pixels and produces a seamless blending result.