Vidyavardhini's College of Engineering &Technology

Department of Computer Engineering

To study the Object segmentation
Date of Performance: 14/08/2023
Date of Submission: 21/08/2023

Aim: To study the Object segmentation

Objective: To study Object segmentation using the Watershed and GrabCut algorithms Example of foreground detection with GrabCut, Image segmentation with the Watershed algorithm

Theory: The process of object segmentation can be achieved through the application of the Watershed and GrabCut algorithms. While calculating a disparity map proves highly valuable in identifying the foreground elements within an image, it's essential to note that StereoSGBM isn't the sole algorithm for this task. In fact, StereoSGBM primarily focuses on extracting three-dimensional data from two-dimensional images. On the other hand, GrabCut emerges as an ideal tool specifically designed for the precise delineation of foreground objects.

GrabCut algorithm follows a precise sequence of steps:

1. A bounding rectangle is established to encompass the subject(s) within the image.
2. Automatically, the region outside this rectangle is designated as the background.
3. Information present in the background area serves as a reference for distinguishing between background and foreground regions within the user-defined rectangle.
4. The foreground and background are modeled using a Gaussian Mixture Model (GMM), categorizing undefined pixels as potential background or foreground.
5. Each pixel in the image is virtually linked to neighboring pixels through virtual edges, and these edges are assigned probabilities of being either foreground or background based on their color similarity to surrounding pixels.

6. Every pixel, or node as defined within the algorithm, is associated with either a foreground or a background node.
7. Once the nodes have been connected to their respective terminals (background or foreground, also referred to as source and sink), the algorithm proceeds to cut the edges between nodes belonging to different terminals. This pivotal step, known as the 'cut' operation, facilitates the effective partitioning of the image into distinct regions.

Image segmentation with the Watershed algorithm :- Finally, we take a quick look at the Watershed algorithm. The algorithm is called Watershed, because its conceptualization involves water. Imagine areas with low density (little to no change) in an image as valleys, and areas with high density (lots of change) as peaks. Start filling the valleys with water to the point where water from two different valleys is about to merge. To prevent the merging of water from different valleys, you build a barrier to keep them separated. The resulting barrier is the image segmentation.

Code:-

```
import cv2

import numpy as np

from IPython.display import Image, display

from matplotlib import pyplot as plt

# Show image

image = cv2.imread("team.jpeg")

image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.subplot(1,1,1),plt.imshow(image),plt.title('Original Image')

# Create subplots with 1 row and 2 columns
```

```
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(8, 8))

# sure background area

sure_bg = cv2.dilate(bin_img, kernel, iterations=3)

imshow(sure_bg, axes[0,0])

axes[0, 0].set_title('Sure Background')

# Distance transform

dist = cv2.distanceTransform(bin_img, cv2.DIST_L2, 5)

imshow(dist, axes[0,1])

axes[0, 1].set_title('Distance Transform')

#foreground area

ret, sure_fg = cv2.threshold(dist, 0.5 * dist.max(), 255,
cv2.THRESH_BINARY)

sure_fg = sure_fg.astype(np.uint8)

imshow(sure_fg, axes[1,0])

axes[1, 0].set_title('Sure Foreground')

# unknown area

unknown = cv2.subtract(sure_bg, sure_fg)

imshow(unknown, axes[1,1])

axes[1, 1].set_title('Unknown')

plt.show()
```

OutPut:-

Sure Background

Distance Transform

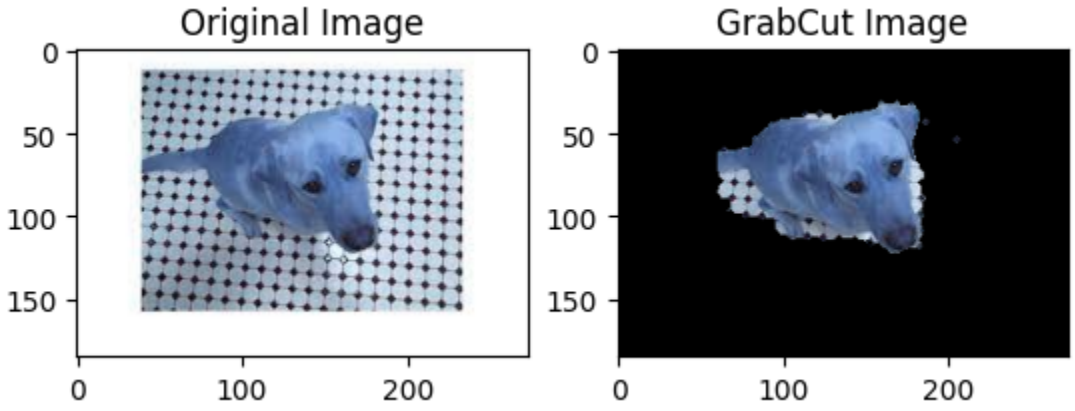
Sure Foreground

Unknown

Code For GrabCut:-

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
image = cv2.imread('doggy G.jpg')
mask = np.zeros(image.shape[:2], np.uint8)
backgroundModel = np.zeros((1, 65), np.float64)
foregroundModel = np.zeros((1, 65), np.float64)
rectangle = (20, 100, 150, 150)
cv2.grabCut(image, mask, rectangle,
      backgroundModel, foregroundModel,
      3, cv2.GC_INIT_WITH_RECT)
mask2 = np.where((mask == 2)|(mask == 0), 0, 1).astype('uint8')
image1 = image * mask2[:, :, np.newaxis]
plt.subplot(2,2,1),plt.imshow(image),plt.title('Original Image')
plt.subplot(2,2,2),plt.imshow(image1),plt.title('GrabCut Image')
```

OutPut:-



Conclusion: In conclusion, image segmentation employs two notable techniques in the field of computer vision, namely the Watershed and GrabCut algorithms. Watershed segmentation leverages a topographic analogy to partition regions based on gradient information, making it particularly adept at isolating objects with clearly defined boundaries. Conversely, GrabCut employs user interaction to iteratively refine object delineation, rendering it valuable in scenarios such as image manipulation and editing. Each approach possesses its unique advantages and limitations, and the selection between them hinges on the specific demands of the task under consideration