Experiment-7

Image Segmentation using Graph Cut

Name : N U Praneeth Reddy **Reg.No: 21BAI1500**

Aim: To Perform image segmentation using Graph-Cut algorithm on a sample image.

Resources Used: Anaconda Python Environment

Google Colab Jupyter Notebook

Theory :

OpenCV stands as an open-source library designed for computer vision and machine learning applications. Its primary goal is to offer a unified foundation for computer vision projects and to facilitate the integration of machine perception into various commercial products.

On the other hand, NumPy serves as a Python library, enabling support for large, multi-dimensional arrays and matrices, accompanied by an extensive array of high-level mathematical functions for manipulating these arrays.

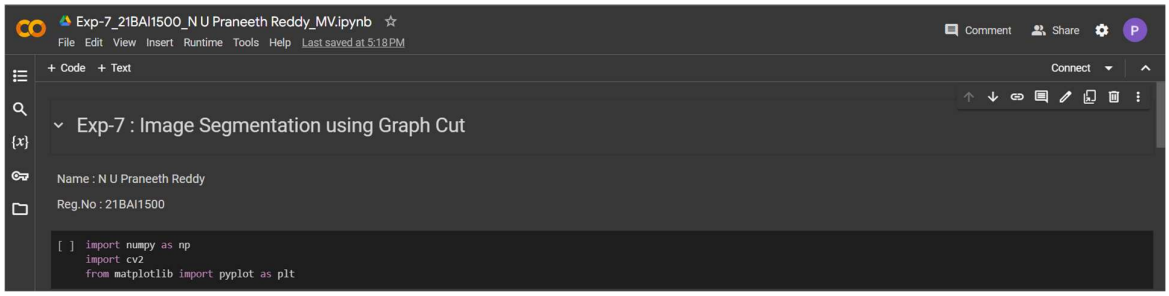
Additionally, Matplotlib functions as a Python plotting library, directly connected to the numerical mathematics capabilities of NumPy. It delivers an object-oriented API for seamlessly embedding plots within applications.

Tasks:

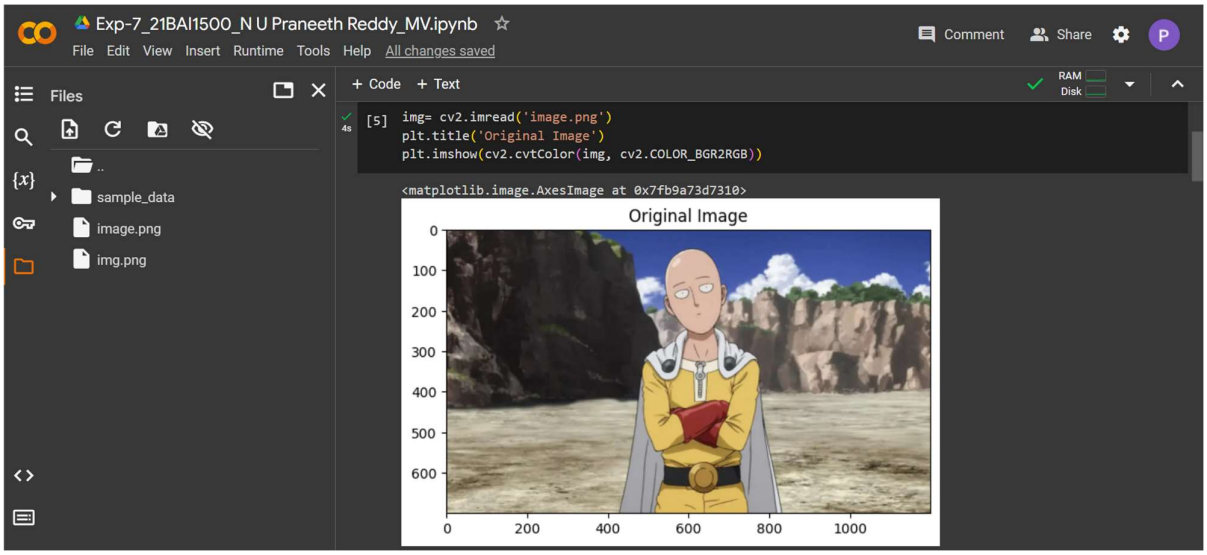
- Perform image segmentation using Graph-Cut algorithm on a sample image. Initialize a suitable seed point on the image (foreground) and perform the partition of foreground and background. Display the binary mask and also the extracted object.

Procedure :

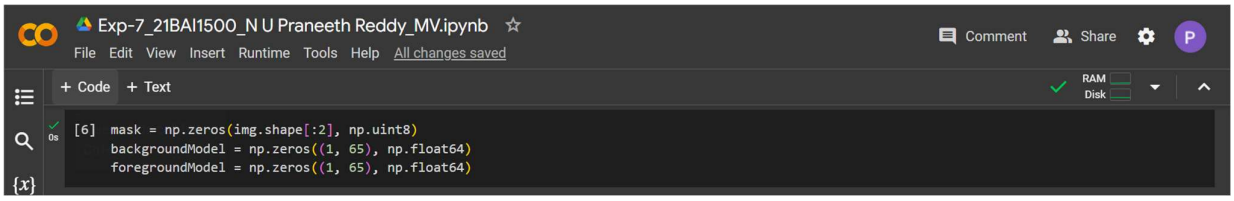
- Open Google Colab and create a new Jupyter Notebook.
- Import important libraries namely OpenCV, Numpy and Matplotlib.



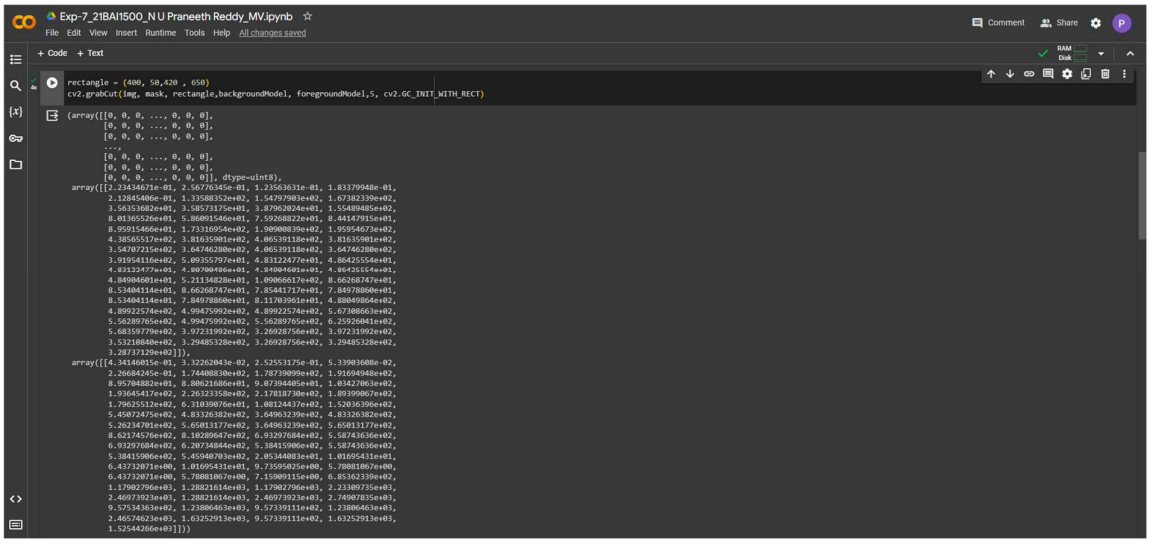
- Read the image using imread in the OpenCV library and display it in the RGB model.



- Create a mask of zeros with the same shape as the input image. This mask will be used to specify the regions of interest (background and foreground) for the Graph-Cut algorithm

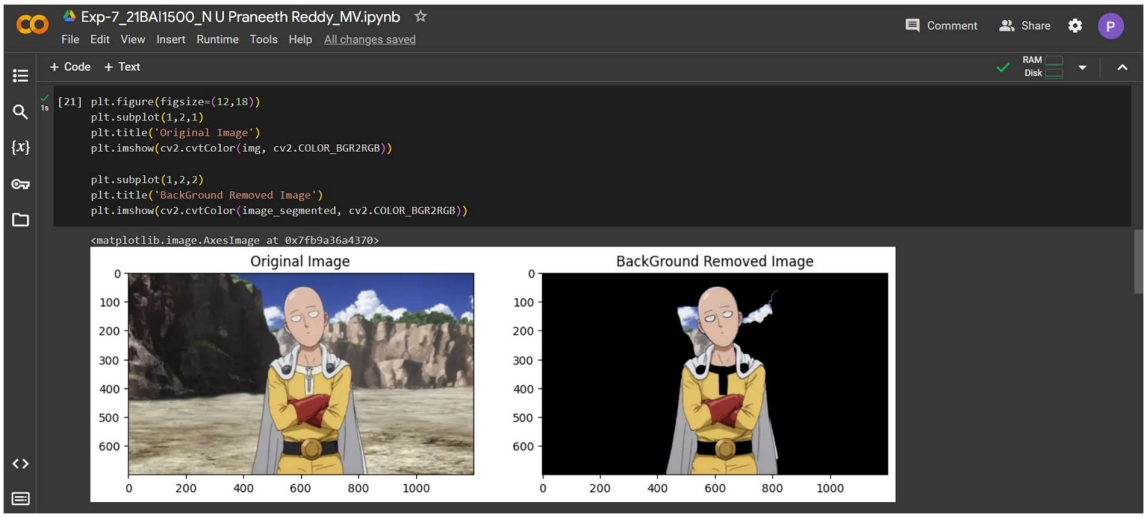


- Define a rectangle that roughly encloses the object of interest within the image. This rectangle serves as an initial approximation for the foreground object and Apply the GrabCut algorithm (cv2.grabCut) to the input image using the provided rectangle and masks



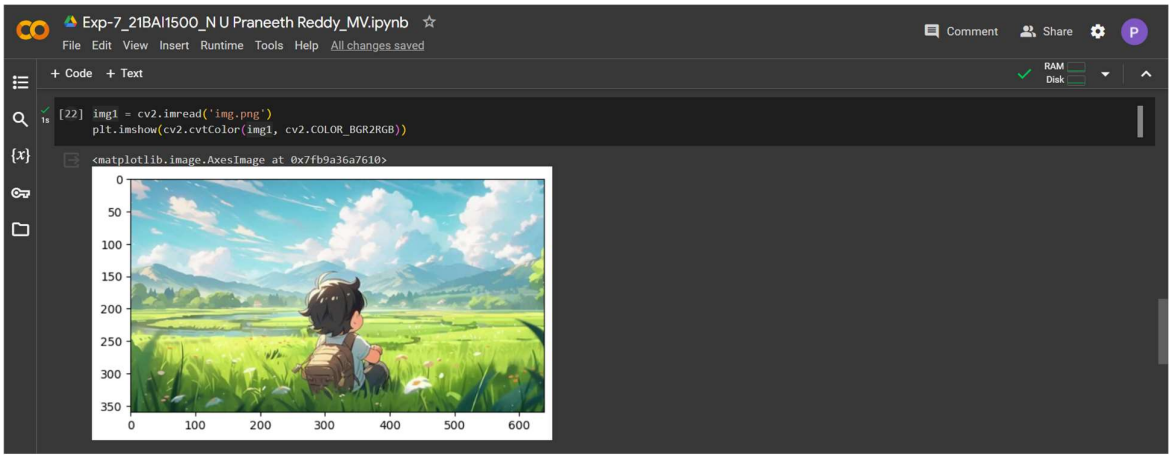
- Modify the mask to extract the foreground object by setting background and probable background pixels to 0 and all other pixels to 1, and also Multiply the original image with the modified mask to extract the segmented object.

- Display the original image and the extracted object side by side using plt.subplot and plt.imshow.

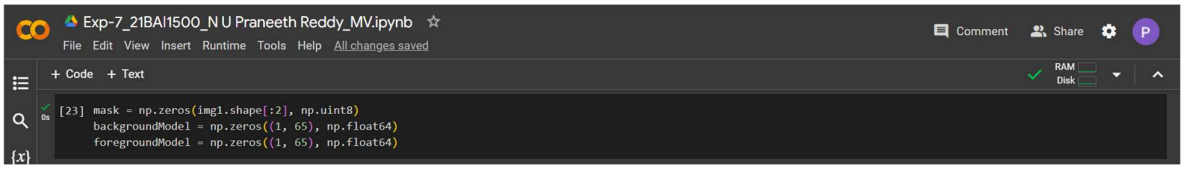


For Other Image :

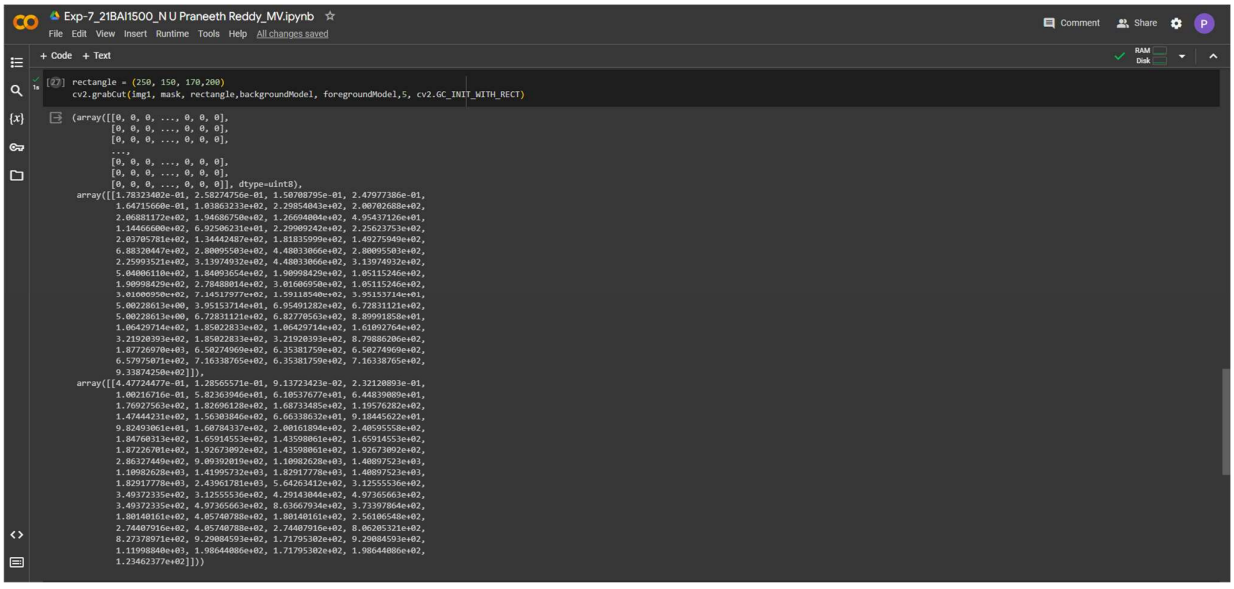
- Read the image using imread in the OpenCV library and display it in the RGB model.



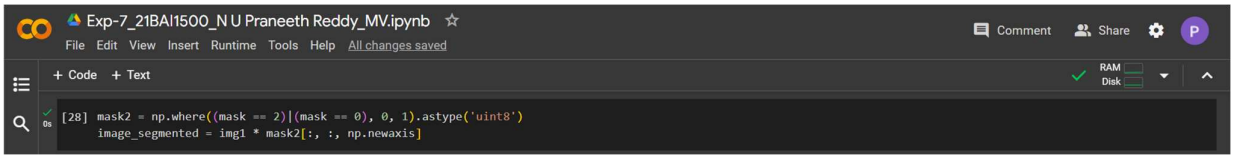
- Create a mask of zeros with the same shape as the input image. This mask will be used to specify the regions of interest (background and foreground) for the Graph-Cut algorithm



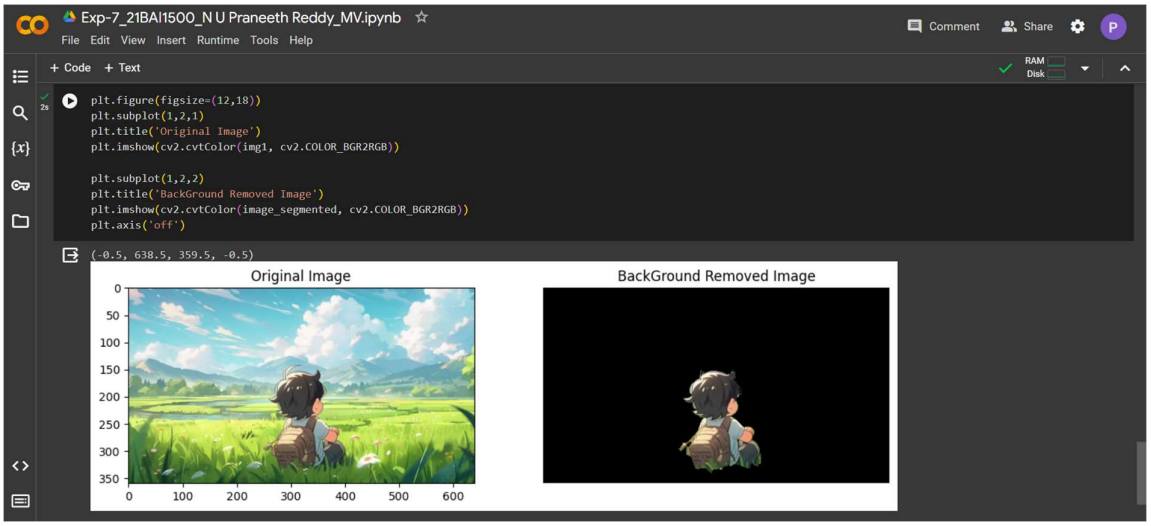
- Define a rectangle that roughly encloses the object of interest within the image. This rectangle serves as an initial approximation for the foreground object and Apply the GrabCut algorithm (cv2.grabCut) to the input image using the provided rectangle and masks



- Modify the mask to extract the foreground object by setting background and probable background pixels to 0 and all other pixels to 1, and also Multiply the original image with the modified mask to extract the segmented object.



- Display the original image and the extracted object side by side using plt.subplot and plt.imshow.



Results: The given task has been done using programs in Python using CV2, Matplotlib and numpy libraries and the Background has been removed successfully

Conclusion: The Python program successfully demonstrates the application of the GrabCut algorithm to segment and extract objects from an input image. By utilizing the initial rectangle approximation provided by the user, the algorithm accurately isolates the object of interest from the background. The segmented object is displayed alongside the original image, showcasing the effectiveness of the GrabCut algorithm in extracting objects from provided Images

Google Collab Link :
https://colab.research.google.com/drive/1bplywqz0gMj0aiVJIX6XlgY0MX7DX4IG?usp=sharing