

# **Image Segmentation**

**Prepared by**  
Sergio Torres

**Computer Vision**  
**Cap 4410, Section 01**

**3/6/20**

# Table of Contents

<b>Table of Contents</b> .....	<b>1</b>
<b>List of Figures</b> .....	<b>2</b>
<b>1. Report</b> .....	<b>3</b>
1.1 Otsu 2 Class .....	3
1.2 Otsu Multi Class .....	4
1.3 Mean Shift.....	5
1.4 Product Scope .....	5
<b>References</b> .....	<b>6</b>

## List of Figures

<b>Figure 1</b> Otsu Method 2 Class.....	6
<b>Figure 2</b> Otsu Method Multi Class.....	7
<b>Figure 3</b> Mean Shift.....	7

# 1. Report

## 1.1 Otsu 2 Class

I worked with the input images provided for the Otsu Method 2 classes. It took me some time to figure out how to do the otsu method and then doing semantic segmentation. I found some code that helped me figure out how to do the otsu method. I first load the input image and then change it to a gray image and then change the background so that any white pixels would be turned black. I then create a kernel to sharpen the image and apply it to the image. I then change the image to a binary one, do the distance transform algorithm and then normalize the image between 0 and 1. I then obtain the peaks for the foreground objects. I find the markers for the image for the watershed algorithm and then draw the foreground marker and show the image, which is what is being shown in the image below. I could not get the watershed algorithm working which did the semantic segmentation.

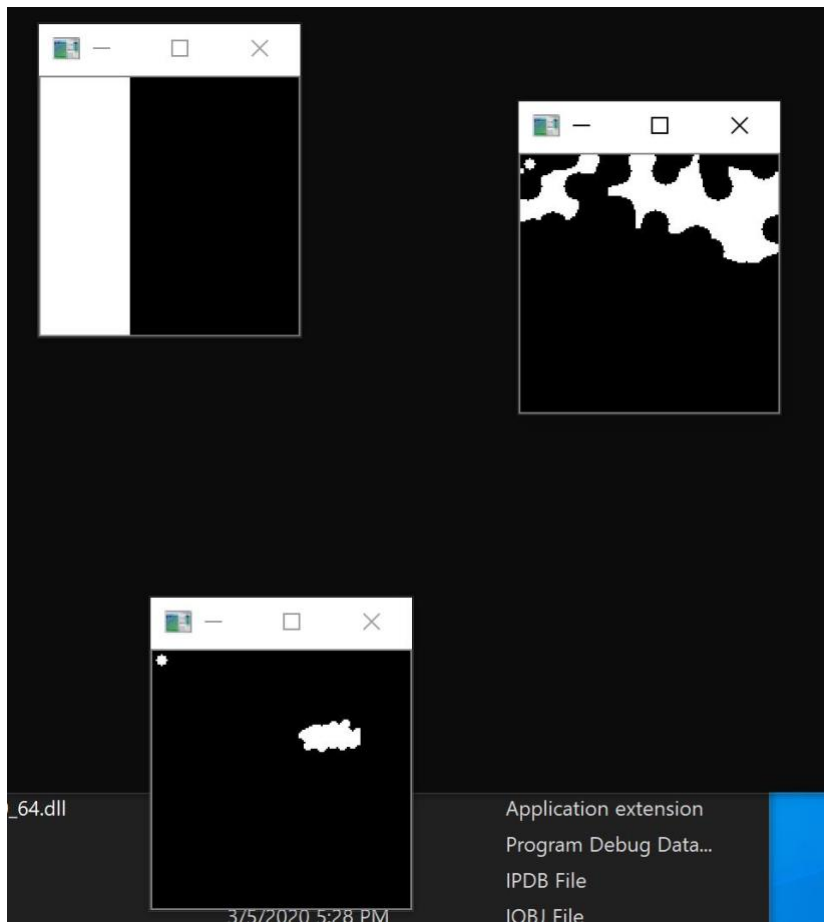


Figure 1. Otsu Method 2 Class

## 1.2 Otsu Multi Class

I worked with the input images provided for the Otsu Method Multiple classes. It took me some time to figure out how to do the otsu method and then doing semantic segmentation. I found some code that helped me figure out how to do the otsu method. I first load the input image and then change it to a gray image and then change the background so that any white pixels would be turned black. I then create a kernel to sharpen the image and apply it to the image. I then change the image to a binary one, do the distance transform algorithm and then normalize the image between 0 and 1. I then obtain the peaks for the foreground objects. I find the markers for the image for the watershed algorithm and then draw the foreground marker and show the image, which is what is being shown in the image below. I could not get the watershed algorithm working which did the semantic segmentation. The only difference between this Otsu Method and the one above is that I used multiple Otsu methods with this code.

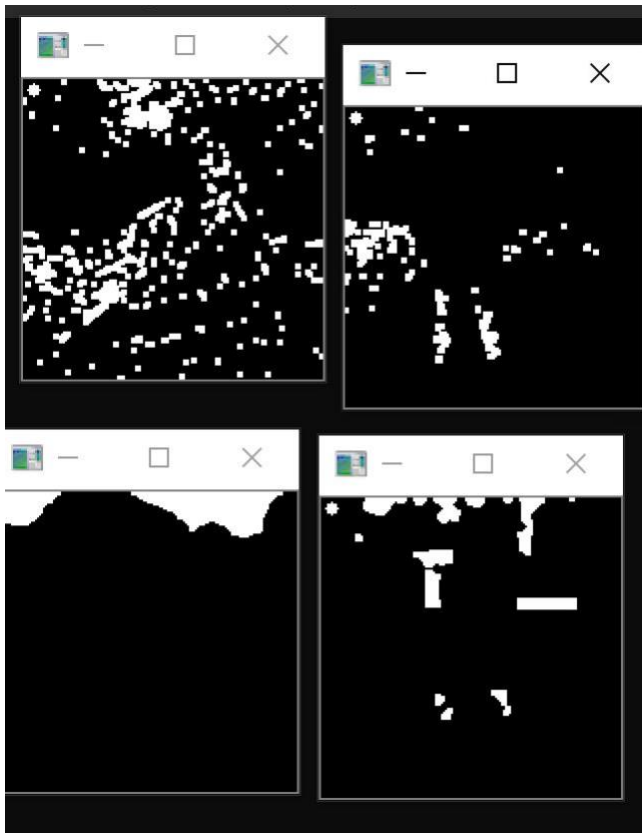


Figure 2. Otsu Method Multi

### 1.3 Mean Shift

I didn't have much time to do the mean shift method for this part and ended up doing the Mean Shift Filter to show something.

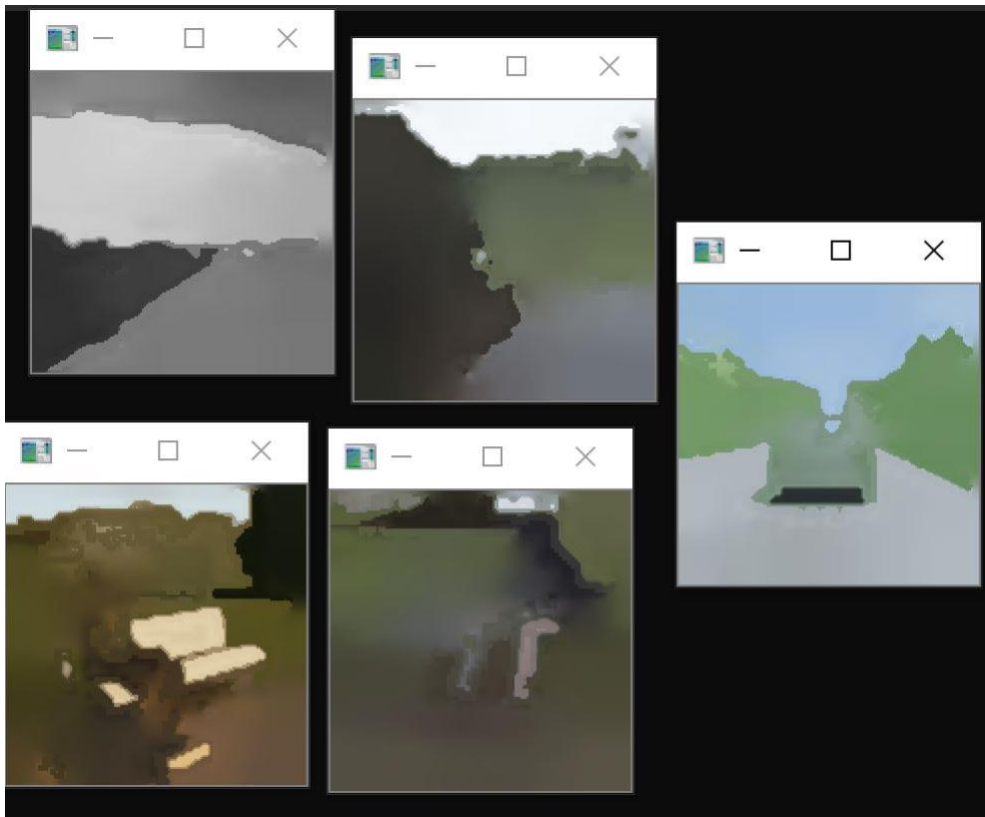


Figure 3. Mean Shift

### 1.4 Product Scope

In this project the objective is to work out a suggestion for semantic segmentation method for some simple input images. I am provided a set of experimental data.

# References

- [1] "User Interface — Opencv 2.4.13.7 Documentation". *Docs.Opencv.Org*, 2020, [https://docs.opencv.org/2.4/modules/highgui/doc/user\\_interface.html?highlight=namedwindow](https://docs.opencv.org/2.4/modules/highgui/doc/user_interface.html?highlight=namedwindow).
  
- [2] Image Segmentation with Watershed Algorithm. (n.d.). Retrieved from [https://docs.opencv.org/3.4/d7/d1c/tutorial\\_js\\_watershed.html](https://docs.opencv.org/3.4/d7/d1c/tutorial_js_watershed.html)
  
- [3] Image Segmentation with Watershed Algorithm. Retrieved from [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_watershed/py\\_watershed.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_watershed/py_watershed.html)
  
- [4] Image Segmentation with Distance Transform and Watershed Algorithm. Retrieved from [https://docs.opencv.org/master/d2/dbd/tutorial\\_distance\\_transform.html](https://docs.opencv.org/master/d2/dbd/tutorial_distance_transform.html)
  
- [5] Python OpenCV: Meanshift. Retrieved from <https://www.geeksforgeeks.org/python-opencv-meanshift/>