# **DPS920/CVI620 – Lab 10**

# **Segmentation**

| Total Mark: | 10 marks (3% of the total course grade)   * 6 out of 10: Learn@Seneca submission (Due: Wednesday November 22nd end of day) * 4 out of 10: Lab demo (Due: During Workshop of week 11) |
| --- | --- |
| Submission file(s): | * Lab10\_1.py / Lab10\_1.ipynb * Lab10\_2.py / Lab10\_2.ipynb * Lab10.docx |

Please work in **groups** to complete this lab. This lab is worth 3% of the total course grade and will be evaluated through your written submission, as well as the lab demo. During the lab demo, group members are *randomly* selected to explain the submitted solution. Group members who are not present during the lab demo will lose the demo mark.

Please submit the submission file(s) through Learn@Seneca. ALL team members must submit the final work.

***Please paste the resulting images and answers in this document.***

## **Part I: The Watershed Algorithm**

1. Use the following permalink to access this book:

<https://senecacollege.primo.exlibrisgroup.com/permalink/01SENC_INST/17thfn4/alma997310770703226>

Then in “Chapter 4- Depth Estimation and Segmentation”, find “Image segmentation with the Watershed algorithm”. Follow the instructions to segment the attached image 5\_of\_diamond.png.

Write your code in Lab10\_1.py.

Change the code to draw the green contours on a black image (instead of on the image itself). Paste the final result here.

A screenshot of a computer

Description automatically generated

1. Then explain what the effect of each step is. Paste results of each step (if applicable) and discuss.

To do so, you might need to remove that step and/or do some modifications to the code.

Now, I will explain the effect of each step:

1. **Grayscale Conversion**: This simplifies the image to a single channel, making it easier to perform thresholding and noise removal.

A card with diamonds and numbers

Description automatically generated

1. **Thresholding**: This step converts the grayscale image to a binary image. It distinguishes the foreground (the objects we want to segment) from the background.

A black and white playing card with diamonds

Description automatically generated

1. **Noise Removal**: By applying morphological opening, which is erosion followed by dilation, small noise is removed from the binary image.

A black and white playing card with diamonds

Description automatically generated

1. **Sure Background Area**: Dilation increases the object area. The dilated image serves as a sure background region—it is certain that all the regions in the dilated area are background.

A black and white card with diamonds

Description automatically generated

1. **Sure Foreground Area**: The distance transform provides a measure of the distance to the closest boundary from each pixel. A threshold is then applied to determine the sure foreground areas.

A black background with white diamonds

Description automatically generated

1. **Unknown Region**: This is the region that is neither sure foreground nor sure background. It is obtained by subtracting the sure foreground from the sure background.

A black and white card with white diamonds

Description automatically generated

1. **Marker Labelling**: Connected components of the sure foreground are identified and labeled.

A black background with white diamonds

Description automatically generated

1. **Markers Adjustment**: The labels for the background and foreground are distinguished, with unknown regions marked as zero.
2. **Watershed Algorithm**: This is applied to segment the objects. The algorithm finds the watershed lines, marking them as borders between objects.
3. **Drawing Contours on a Black Image**: Instead of drawing contours on the original image, they are drawn on a black image to highlight the segmentation boundaries.

A black background with green diamonds

Description automatically generated

If you remove a step, you can then see how its absence affects the outcome of the segmentation process. For instance, without noise removal, you might see more false edges in the final segmentation. Without the distance transform, the algorithm might not accurately identify the objects' centers and boundaries.

## **Part II: DL-Based Segmentation with Mask R-CNN**

1. Do some research about Mask R-CNN. What did you learn? Write one or two paragraphs. Include images and links to references.

Mask R-CNN is a segmentation algorithm which detects certain objects on the image and puts a mask on the object, highlighting all pixels detected as belonging to this object. Mask R-CNN excels at pixel-level segmentation of detected objects. It detects multiple classes and can handle instances with overlapping boundaries. Developers can use a pretrained Mask R-CNN network through the maskrcnn object, trained on the MS-COCO dataset to detect objects from 80 different classes.

Figure 1. Displaying instance masks and bounding boxes

A traffic lights on a street

Description automatically generated

*Note.* Image from MathWorks

The Mask R-CNN model operates in two stages: the first stage involves a region proposal network (RPN) predicting object proposal bounding boxes, while the second stage employs an R-CNN detector to refine and classify these proposals, ultimately computing pixel-level segmentation. Mask R-CNN builds upon the Faster R-CNN model, introducing improvements such as an roiAlignLayer for more accurate sub-pixel level ROI pooling and the addition of a mask branch dedicated to pixel-level object segmentation.

References:

(n.d.). *Getting Started with Mask R-CNN for Instance Segmentation*. MathWorks. Retrieved November 22, 2023, from https://www.mathworks.com/help/vision/ug/getting-started-with-mask-r-cnn-for-instance-segmentation.html

1. In Lab10\_2.py, write the code to segment images using Mask R-CNN.

Watch the following video to learn about using Mask R-CNN network for object detection and segmentation.

* [Instance Segmentation MASK R-CNN | with Python and Opencv - YouTube](https://www.youtube.com/watch?v=8m8m4oWsp8M)
* Accompanying sites:

[Instance Segmentation MASK R-CNN | with Python and Opencv - Pysource](https://pysource.com/2021/05/18/instance-segmentation-mask-r-cnn-with-python-and-opencv/)

[GitHub - matterport/Mask\_RCNN: Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow](https://github.com/matterport/Mask_RCNN)

* 1. First, detect objects (draw the bounding boxes) for the “Street” and the “Wildlife” images. Paste results here.

A person walking dogs on the street

Description automatically generated

A giraffe and zebras in a field

Description automatically generated

* 1. How many classes of objects could the network detect? What were these classes, and what were their corresponding object class numbers?

Street: 3 classes of objects

Classes detected: [17.0, 17.0, 0.0, 2.0, 0.0, 2.0, 2.0, 0.0, 2.0]

1. – person

2.0 - car

17.0 - dog

Wildlife: 4 classes of objects

Classes detected: [24.0, 21.0, 23.0, 23.0, 23.0, 21.0, 21.0, 20.0]

20.0 – cow

21.0 – elephant

23.0 – zebra

24.0 – giraffe

* 1. Now, segment objects (take out the mask) for the “Street” and the “Wildlife” using 3 different scores (0.1, 0.3, and 0.5). Paste results here.

Street: 0.1 score

A colorful silhouettes of people

Description automatically generated

Street: 0.3 score

A colorful silhouettes of a person and a dog

Description automatically generated

Street: 0.5 score

A colorful silhouettes of people

Description automatically generated

Wildlife: 0.1 score

A group of colorful shapes

Description automatically generated with medium confidence

Wildlife: 0.3 score

A group of colorful shapes

Description automatically generated with medium confidence

Wildlife: 0.5 score

A group of colorful shapes

Description automatically generated with medium confidence

## **Part III: Group Work**

1. Add this declaration to your file:

We, group 5, Davender Singh and Liliya Panfilova, declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. We have not copied any part of this assignment, manually or electronically, from any other source including web sites, unless specified as references. We have not distributed our work to other students.

1. Specify what each member has done towards the completion of this work:

|  | Name | Task(s) |
| --- | --- | --- |
| 1 | Davender Singh | Part I: The Watershed Algorithm |
| 2 | Liliya Panfilova | Part II: DL-Based Segmentation with Mask R-CNN |
| 3 |  |  |