Introduction to Computer Vision (ECSE 415)
Assignment 5: Segmentation

DEADLINE: April 2, 2024
Please submit your assignment solutions electronically via the **myCourses** assignment dropbox.

The submission should include a single Jupyter notebook. More details on the format of the submission can be found below. Submissions that do not follow the format will be penalized 10%.

The assignment will be graded out of a total of **100 points**. There are *50 points* for accurate analysis and description, *40 points* for bug-free and clean code, and *10 points* concerning the appropriate structure in writing your report with citations and references.

Each assignment will be graded according to defined rubrics that will be visible to students. Check out MyCourses, the "Rubrics" option on the navigation bar. You can use **OpenCV**, **sklearn**, **skimage**, **Numpy**, and **Pytorch** library functions for all parts of the assignment except those stated otherwise. Students are expected to write their own code. (Academic integrity guidelines can be found **here**).

Assignments received late will be penalized by 10% per day.

Submission Instructions

1. Submit a single Jupyter notebook consisting of the solution of the entire assignment.
2. Comment your code appropriately.
3. Give references for all codes which are not written by you. (Ex. the code is taken from an online source or from tutorials)
4. Do not forget to run **Markdown** ('Text') cells.
5. Do not submit input/output images. Output images should be displayed in the Jupyter notebook itself.
6. Make sure that the submitted code is running without error. Add a **README** file if required.
7. If external libraries were used in your code please specify their name and version in the **README** file.
8. We are expecting you to make a path variable at the beginning of your codebase. This should point to your working local (or google drive) folder.
   Ex. If you are reading an image in the following format:

   ```
   img = cv2.imread ( '/content/drive/MyDrive/Assignment1/images/shapes.png' )
   ```

   Then you should convert it into the following:

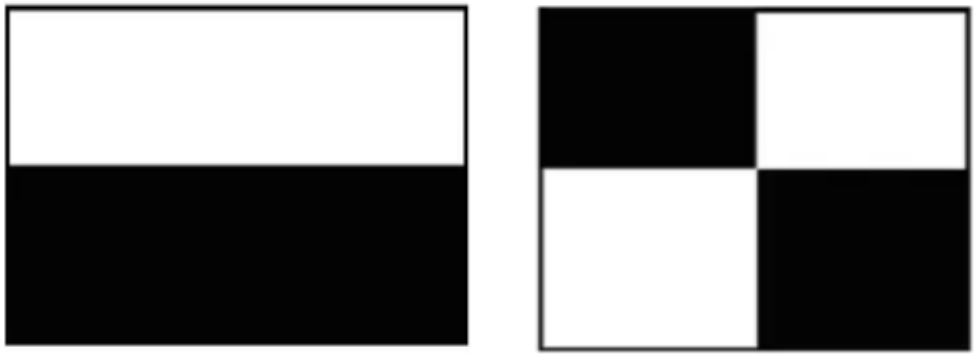
   ```
   path = '/content/drive/MyDrive/Assignment1/images/'
   img = cv2.imread(path + ' shapes.png' )
   ```

   Your path variable should be defined at the top of your Jupyter notebook. While grading, we are expecting that we just have to change the path variable once and it will allow us to run your solution smoothly. Specify your path variable in the **README** file.
9. Answers to reasoning questions should be comprehensive but concise.

1 K-Means and Mean-Shift Clustering for Segmentation (50 points)

In this section, you will be asked to compute image segmentations by using several basic clustering techniques. Clustering is used to determine the class of each pixel, and the result can be different depending on the feature space. The images for this part are placed under the same dictionary.

1. Compute the features of the Dog.jpg and Person.jpg images by convolving the images with the two Haar filter kernels shown below. The white areas of the Haar filter kernel all have a weight of +1, while the black areas have a weight of -1. For the purposes of obtaining the convolution values for pixels near the border of the image, assume that the intensity values of pixels outside the borders of the image are 0. You could use the integral image technique to implement the Haar filtering in a more computationally efficient (i.e. faster) manner.
   Display the filtered feature images.



(a) Rectangle with size12x6 pixels. (b) Square with size 12x12 pixels.

Figure 1: Haar Filters for computing image features.

2. Implement the K-means clustering to compute the segmentation of the Dog.jpg and the Person.jpg image with Haar features. Set K=3. Display the segmented images.
3. Implement the Mean-shift clustering to compute the segmentation of the Dog.jpg and Person.jpg images. Display the segmented images.
   You can use the *scikit* implementation of the mean shift method:
   sklearn meanshift
4. Discuss the benefits and limitations of these clustering methods for image segmentation.

(a) Dog image (b) Person image

Figure 2: Images for segmentation.

2 Neural Network Implementation for Image Segmentation (50 points)

There are several neural networks widely used for object detection and image segmentation. In this assignment, you will be asked to use a *well-trained* Mask R-CNN and YOLOv8n-seg to construct a neural network that can perform segmentation tasks. The output of Mask R-CNN and YOLOv8n-seg should include bounding boxes on detected objects with the object category. The network also provides the instance level segmentation of the object inside each bounding box. For more information, please refer to the GitHub repository 1: Mask R-CNN for Object Detection and Segmentation and GitHub repository 2: Ultralytics YOLOv8.

1. Implement the pre-trained Mask R-CNN and YOLOv8n-seg model and run them on the 552.jpg and 7726.jpg image included in the assignment folder.
2. Display the result that shows the bounding boxes, object classes, and segmentations inside each bounding box.
3. Repeat steps 1 and 2 for an image of a Montreal street scene that you took with your own camera. You can use the image that you acquired for Assignment 4.
4. Compare the performance of these two models and explain the steps that Mask R-CNN took to achieve the final result.

(a) 552.jpg

(b) 7726.jpg

Figure 3: Two examples in the MSCOCO 2017 test set.