Computer Vision 1: Assignment 5
(Due date: 12.02.2021)

Submission instructions:

- For each programming task, submit a `.py` source code file. Do not include any images or data files you used.
- For each pen & paper task, submit a `.pdf` file. Type your solution in LaTeX, Word, or another text editor of your choice and convert it to PDF – **do not submit photographs or scans of handwritten solutions!**
- In all files, include at the top names of all students in the group
- Choose exactly one person in your group that submits your solution via Moodle, it will count for the entire group.

Task 2: Undersegmentation error (pen & paper)

We have developed a segmentation algorithm that has partitioned an image into segments S_i, $i = 1, \ldots, n$ as described on the lecture slides. For the same image, we have a *ground truth segmentation* G_j, $j = 1, \ldots, m$, which specifies how the image ideally should be partitioned.

Intuitively, the *undersegmentation error* for an individual ground truth segment G_j measures the amount of "bleeding" that a segmentation exhibits beyond G_j. The undersegmentation error for a ground truth segment G_j is defined as

$$\mathrm{UE}(G_j) = \frac{\left[\sum_{\{S_i | S_i \cap G_j \neq \emptyset\}} \mathrm{Area}(S_i)\right] - \mathrm{Area}(G_j)}{\mathrm{Area}(G_j)} \tag{1}$$

where the summation is over all segments S_i which have any overlap (non-empty intersection) with G_j, and the function Area returns the area, i.e., the number of pixels, of the corresponding segment.

Our image has 12 pixels in it, and our segmentation algorithm produces the segments S_i shown in Figure 1. Calculate the undersegmentation errors $\mathrm{UE}(G_1)$ and $\mathrm{UE}(G_2)$ of the ground truth segments G_1 and G_2 shown in Figure 2.

1	4	4	4
1	2	4	4
1	2	3	3

Figure 1: A segmentation S_i, $i = 1, 2, 3, 4$, with i indicated on each pixel.

Figure 2: Two ground truth segments G_1 (left) and G_2 (right) highlighted in gray.