Home Assignment 1

About this homework

Submission

You are required to submit your solutions to all parts via Moodle. For this assignment you will need to submit:

- Python files (.py or .ipynb) for the coding parts;
- PDF file with a detailed explanation of your approach.

Submit files as a single archive with your name. E.g. HW1_Ivanov_Ivan.zip.

Plagiarism

Plagiarism will not be tolerated and a plagiarised code will be heavily penalised for all parties involved. Remember that you learn nothing when you copy someone elses work which defeats the purpose of the exercise!

Problem 1

Implement adaptive thresholding from scratch. Your function should take an image as input and output a thresholded binary image at the output. Please, use template shown below. Also, your solution should contain a convenient user interface to adjust the parameters of the function (sliders, buttons, checkboxes etc).

Code template 1: Adaptive thresholding

```
def adaptive_thresholding(image, thresholding_type = 'mean', block_size = 3):
"""
Function calculates adaptive thresholding for a given image
Input:
    image: grayscale or color image
    thresholding_type: type of thresholding;
                       possible values: 'mean' and 'gaussian'
    block_size:  kernel size in which the threshold value is
                 calculated for the current pixel
Output:
    image_bw:  binary thresholded image
"""

    if thresholding_type == 'mean':
        # Your code here
    elif thresholding_type == 'gaussian':
        # Your code here
    return image_bw
```

Home Assignment 1

Grading policy: 2 points (code - 60 % , documentation - 20%, interface - 20%)

Problem 2

Calculate an object size by photo and order objects according certain size. If two objects have the same size (two identical objects), then sort them from left to right. Show rank, and size values (area, perimeter, height, width) in selected units (mm, cm, m, pixels). Please, use template shown below. Also, your solution should contain a convenient user interface to adjust the parameters of the function (sliders, buttons, checkboxes etc).

Figure 1: Example photo for object size calculation

Code template 2: Object size calculation and ranking

```
def calculate_sizes_and_rank(image, units = 'm', rank_by = 'area'):
    """
    Function calculates objects sizes and rank them on image
    Input:
        image: grayscale or color image
        units: unit of measurement of object size;
               possible values: 'mm','cm', 'm' or 'pixels'
        rank_by: object ranking parameter;
                 possible values: 'height', 'width', 'perimeter' or 'area'
```

Home Assignment 1

```
    Output:
        image_result:  result image with ranking and measured objects sizes
        sizes: tuple of sizes (heights, widths, perimeters, areas),
               where heights, widths, perimeters, areas are lists

    """

    # Your code here
    sizes = (heights, widths, perimeters, areas)
    return image_result, sizes
```

Grading policy: 4 points (code - 60% [single object size calculation - 30%, multiple objects ranking by size - 30%], documentation - 20%, interface - 20%)

Problem 3

First eliminate all coins touching the boundary of the paper. Next, extract the remaining coins, and count total value of coins.

Figure 2: Example photo coins calculator

Code template 3: Coins total value calculation

Home Assignment 1

```
def coins_calculator(image):
"""
Function calculates adaptive thresholding for a given image
Input:
    image: grayscale or color image
Output:
    total_val:  total value of coins
"""

    # Your code here

    return total_val
```

Grading policy: 4 points (code - 60 % , documentation - 20%, interface - 20%)