Computer Vision
Workshop 5
Regions & Morphology: Counting Coins

Aims of the workshop

The aim of this session is to introduce Morphology operations on segmentation masks, to obtain a cleaner segmentation results.

These exercises will rely on techniques shown in Lectures for connected components and morphology.

No code will be provided in this workshop, please see “Useful Information” below on how to look up MATLAB functionality. The concept behind this workshop is about discovery, experimentation, and promoting critical thinking around the techniques covered so far in the module.

Feel free to discuss the work with peers, or with any member of the teaching staff.

Useful Information

MATLAB Documentation

MATLAB functionality mentioned during lecture or workshops is fully-documented and available at: https://uk.mathworks.com/help/matlab/

You may find the “Getting Started” section of this documentation to be useful: https://uk.mathworks.com/help/matlab/getting-started-with-matlab.html

In addition, MATLAB Answers may be useful in finding common solutions to any issues you may encounter: https://uk.mathworks.com/matlabcentral/answers/help?s_tid=gn_mlc_ans_hlp

Useful functions: imhist, graythresh, imbinarize, im2bw, imcomplement, strel, imerode, imdilate, imopen, imclose, imfill, label2rgb.

Reminder

We encourage you to discuss the contents of the workshop with the delivery team, and any findings you gather from the session.

Workshops are not isolated, if you have questions from previous weeks, or lecture content, please come and talk to us.

The contents of this workshop are not intended to be 100% complete within the 2 hours. This session involves substantial comparisons and critical thinking, and as such it’s expected that some of this work be completed outside of the session. Exercises herein represent an example of what to do; feel free to expand upon this and make your own comparisons, in addition, if you wish.

Functions

```
function [output1, output2] = my_func_name_is_my_filename(param1,
param2)
      output1 = param1 + param2;
      output2 = param1 * param2;
end
```

You can defined functions within MATLAB, similar to the above. Each new .m file, if defining a function, is named after that function.
E.g meanfilter.m would contain the function named meanfilter.

Any additional code you try to implement in meanfilter.m would result in the editor complaining. New Function -> New File.

Note: This behaviour may be different for those who are using .mlx scripts.

These functions can then be invoked either through the command window, or by other scripts in the same folder. You may find this useful for part of your image processing pipeline.

Note: In these workshops, and the ACW, you can work in either .m or a .mlx "live script". Feel free to experiment with using either, and see which you find most useful.

Exercise 1: Try the above function, invoking it through:

1. The command window
2. Within a MATLAB script.

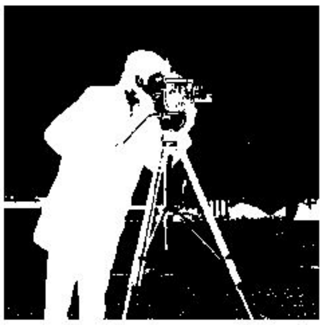
Remember, this function outputs two results. Make use of [ ] to obtain all the return values.

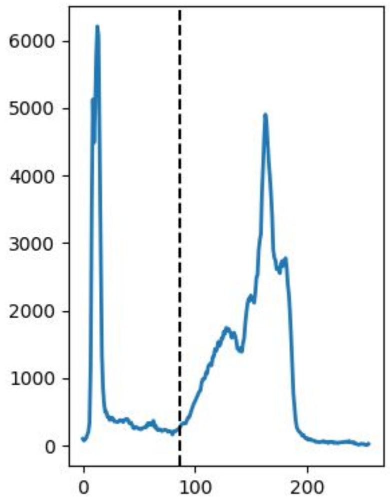
Exercise 2: Canvas contains some images of coins. Download these, and extract them to a folder of your choice (perhaps same directory as your script).
These images have been corrupted with noise. You must first attempt to de-noise these images by identifying the noise and, using appropriate methods, reduce the noise; maintaining image quality.

Using prior knowledge of thresholding, convert the 'corrupted coins' image into greyscale, and inspect the histogram.

1. Threshold this image such that coins are separated from the background.
2. If required, invert the image such that coins are the foreground object. Recall how complement works when dealing with sets.

Exercise 3: *Graythresh* is a function provided by MATLAB which implements a technique known as Otsu’s Thresholding. This will automatically segment a greyscale distribution, finding the global best threshold unique to that image. See Figure below.





From inspection of the coins histogram, you will see the distribution is bi-modal - it has two peaks (modes). Otsu’s Thresholding will find the best threshold between those two peaks. This is done to maximise the inter-class variance, and minimise the intra-class variance.

1. Replace your manual thresholding, with Graythresh. Compare the difference in output. Is your manual value better? What is the difference, did you select a higher or lower grey level to split on?
2. Find a similar coin-based image online. Usi the same manual threshold from Exercise 2 to convert that image to binary. Did this work well? If not, why not?
3. Now try Graythresh on this new image. How well does this perform on our new image?

Consider how more intelligent, automatic approaches may be beneficial when dealing with unseen data (such as that new coins image from the internet). Can you think of any negatives to this approach? Where might manual thresholding be more advantageous?

Exercise 4: Separate and count how many coins are in the provided ‘corrupted_coins’ image.

Note: It may require a combination of morphology operations, with varying arguments / elements. You can use multiple structuring elements for different morphology operations. I.e Box r =3 open, followed by cross r = 5 erode, etc. You can chain morphology functions. E.g Erode, Erode, Open, Dilate, Close, and so on...

1. Create an appropriate structuring element. Consider different shapes and how this, in combination with size, alters the resulting binary image. E.g Use of a box, vs cross structuring element. Radius = 3 vs Radius = 10.
    a. Create these structuring elements manually
    b. Create these structuring elements using built-in functions (strel).
2. Apply morphology operations on the binary output from Exercise 3 to isolate the coins. At the moment they are all touching. Recall the different types of operation (open, close, erode, dilate) and how they change the shape of our binary mask. Refer back to the lecture slides if needed.
3. Try the ‘holes_corrupted_coins’ image. What effect do the holes have on your morphology process? Does it still work, or do you need to modify it? Fill these holes, experimenting with which stage in the pipeline to perform this. I.e does it need to be done at the start, or can it come later on?

Exercise 5: Use connected components to count the number of coins in the final morphology output from Ex 4.2 and Ex 4.3. Display these by converting their labels to RGB colours; additionally, output the total number of coins within the image.

The Extended Exercises are **optional**, and are offered as an advanced supplement for those who have completed the existing work and wish to expand on their practical knowledge; challenging themselves further.

Extended Exercise 1: Create a function which implements Otsu's Thresholding from scratch. This function must take in an RGB Image, convert it to greyscale, and threshold according to that method. The output should consist of a singular logical image, and a found threshold number - the grey level you are splitting on.