Assignment Week 5:
Deconvolution and Image Transformations
Signal and Image Processing

March 5, 2023

You can work on this assignment and submit your solution (report and code) as a GROUP. This assignment counts towards your grade and have to be submitted in order to pass the course. You must follow the report guidelines found in `guidelines.pdf`. The page limit for this assignment is **15 pages** including everything, i.e. illustrations and code snippets.

1. Morphology

1.1. Apply, separately, morphological 'opening' and 'closing' with a disk SE (e.g. using `disk`, radius 1), to the input image `cells_binary_inv.png` (available on Absalon). Using function found in `skimage.morphology` is allowed.

1.1.1. (1 point) **Deliverables:** For each of the two operations, show the resultant image along with the original. Select a region and show a zoom-in of this region for the original and the two result images. Highlight a few key locations and describe the changes.

1.1.2. (1 point) **Deliverables:** Answer these questions:

A. Why are the two operations not producing the same result?

B. What are the challenges of using just these two operations for separating this particular image into individual cells?

1.1.3. (1 point) Use connected components to label the individual cells in both the results of performing opening and closing from question 1.1.1. Use 8-connectivity to define neighbors. We suggest that you use the implementation of connected components in `skimage.measure.label`, but notice that the documentation calls 8-connectivity for 2-connectivity.
Deliverables: Include a code snippet illustrating how you compute the connected components. Provide an illustration of the result of labelling the cells using the connected component algorithm for both the opening and closing results. Report how many different connected components (labels) you get. Reflect on how well you have succeeded in identifying the individual cells in the image using either opening and closing.

1.2. (1 point) Using mathematical morphology, can you count how much money there is in the image `money_bin.jpg` on Absalon. You are also given the additional information of the labeled image found in figure 1.
Deliverables: Please include both a description of what you did, and the final amount.

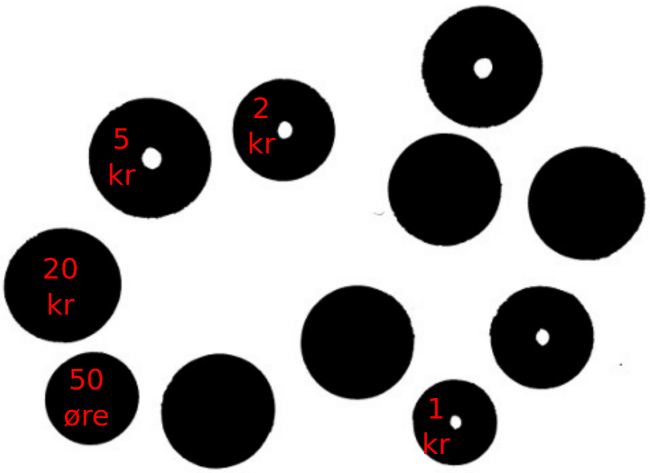

Figure 1: Given the underlying binary image `money_bin.jpg` and the labels seen in this figure, can you count the money without performing segmentation?

2. Inverse filtering

2.1. (1 point) Write a program that takes an image, a kernel and a realisation (a sample) of a noise source (a noise image), and which returns the result of applying the linear shift invariant (LSI) degradation model to the image. We will use the resulting image for testing in the next questions.
Deliverables: Provide a code snippet and explanation of your solution. Illustrate how the program works by applying it to the `trui.png` image for different noise levels and kernels.

2.2. (1 point) Implement direct inverse filtering and apply it to the LSI degraded image of `trui.png` from the previous question. You can only assume that you have the degraded image and that you know the PSF function. You are to implement this yourself - not use a function from e.g. `skimage` or other packages that implement direct inverse filtering.
Deliverables: Provide a code snippet and explanation of your solution. Experiment with different noise levels (remember to try extreme levels) and provide illustrations. Discuss this approach's ability to recover the original image.

2.3. (1 point) Repeat the above exercise by implementing the Wiener filter - eq. (5.8-6) in Gonzales and Woods (again you are to implement this yourself and not use an implementation from e.g. `skimage` or other packages).
Deliverables: Provide a code snippet and explanation of your solution. Experiment with different settings of the constant K approximating the fraction of noise and signal power spectra in eq. (5.8-2). Also experiment with different noise levels (remember to try extreme levels). Discuss this approach's ability to recover the original image.

3. Transformations on images - Translation

Let $I(x, y)$ be a 2D image. Consider the image $\tilde{I}$ obtained by translation of I by one pixel to the right.

3.1. (1 point) Express what $\tilde{I}(x, y)$ is with respect to the pixels in image I.
Deliverables: A mathematical expression.

3.2. (1 point) Write the matrix corresponding to this transformation expressed in homogeneous coordinates and write how this transformation can be used to produce $\tilde{I}$.
Deliverables: Two mathematical expressions.

3.3. (1 point) It is possible to formulate this translation using a linear filter. Write the linear filter kernel corresponding to this transformation.
Deliverables: Include the filter kernel in the report.

3.4. (1 point) Create a zero-valued image of odd size in x and y containing a centred white square.
Deliverables: A figure illustrating the image.

3.5. (1 point) Write a Python function to translate this image by any specified integer number of pixels in both directions using filter masks like in question 3.3. You can use any existing Python package to do the filtering.
Deliverables: Include your code for the function in the report and a figure illustrating the results of applying the function to the image from the previous question. Discuss the different boundary conditions that you can put and what they do.

3.6. (1 point) Make a new function to handle non-integer translations, using the homogeneous transform formulation from question 3.2. and nearest neighbor interpolation. You are not allowed to use `skimage.transform.warp` or similar – instead you should implement this simple image transformation yourself.
Deliverables: Include your code for the function in the report and an illustration of your white square image, as well as the image you get by translating by $\mathbf{t} = (0.6, 1.2)^T$.

3.7. (1 point) Recall the equation for the Fourier transform of a translated image with respect to the Fourier transform of the original image. Write a Python function that performs translation based on the Fourier transform and compare outputs with the previous approach.
Deliverables: Include your code for the function in the report and a figure illustrating the results of applying the function to your white square image. Include a few sentences of comparison with the previous approaches.

3.8. (1 point) Can you do non-integer translations with the Fourier method? Comment on what happens for the previously defined white square image. Apply it also on another image.
Deliverables: Answer to the question and comments about the white square image and examples of applying it to another image of your choice.