Take-Home Exam 2025
FYS 2010 Digital Image Analysis

Start date: **3rd of March 9:00**, deadline for submission: **14:00 the 24th of March**.

- You should provide a Zip folder containing your code, data and your report. Your report must be a pdf file. In addition to the Zip folder, you must submit your report (as a pdf file) separately in Wiseflow.
- Wiseflow checks automatically for plagiarism on the report, so make sure you do not have identical, copy-pasted, answers from others. You may work in groups but the report you give must be your own personal, individual report.
- Images that should be analyzed in this home exam will be provided on Canvas.
- It is recommended to insert small parts of the code inside your report to make your point and make it easy to follow. For example, copy the lines of code which define a particular function you have implemented to answer a question and explain what it does directly in the report. It is clearer and easier for us than checking the code file and trying to find which part is relevant.
- All figures that you produce while solving the exercises should be included in the report. These should be annotated by labeled axes, captions and legends explaining the graphs when it is appropriate.
- If you can't actually solve a problem, due to time or programming problems, you may want to write down some notes about how you would have done it. Your intentions may gain you some partial score for the question.

To have a fair evaluation, we have tried to make the exam so that it is not straightforward for a chatbot to answer the questions. Answering the questions requires to try different approaches and making choices. We focus the evaluation on your understanding of the problem and your scientific approach to it. So we expect you to address the following points:

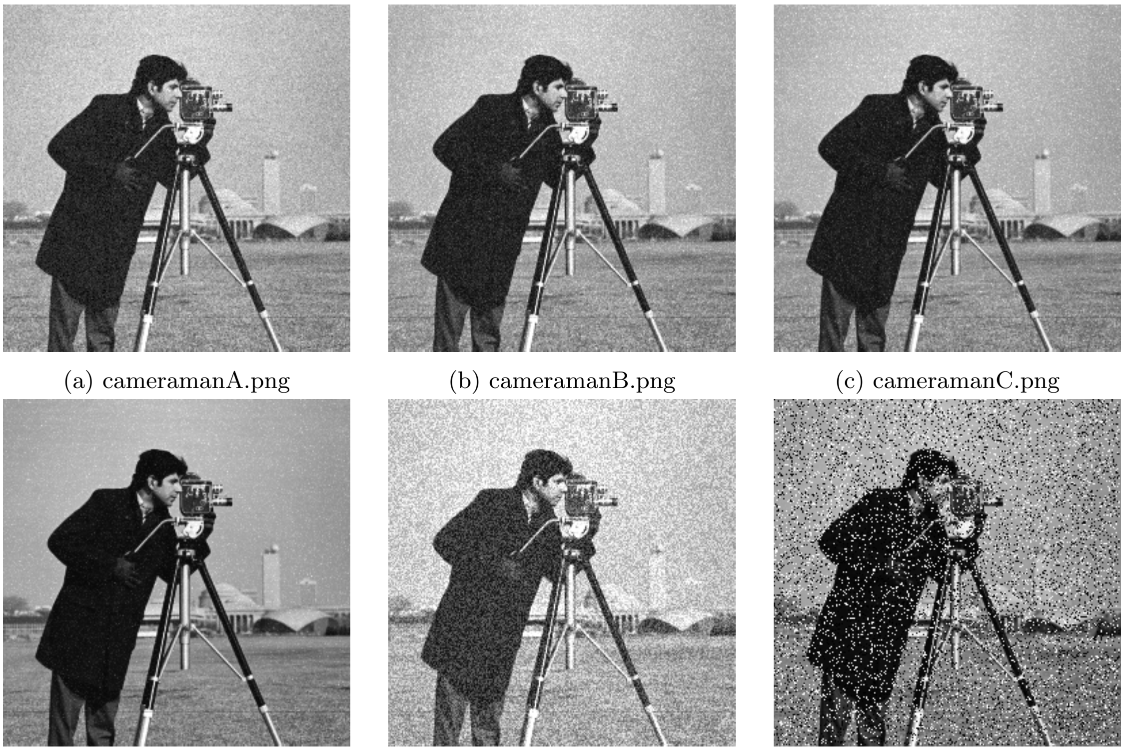
- Explain your approach to solving the problems, the assumptions you make, and how you reason. Lack of explanations will result in a lower score.
- Some of the exercises may require you to make your own implementational choices. That is, the implementation of the algorithms may not be completely specified. In such cases, state your choice and argue for it. It does not need to be a lengthy argument.

The exercises are weighted as follows:

- Task 1 : 20%
- Task 2 : 20%
- Task 3 : 20%
- Task 4 : 20%
- Task 5 : 20%

Task 1

In Fig. 1 you find 6 versions of the famous "cameraman" image, each with a different kind of noise. The aim of this task is to recognize how the type of noise can be identified. There are several approaches possible, so feel free to play around and explore the images in order to answer the following questions. Points are awarded for "justifications", not for mere claims. Hence, you may be rewarded for good explanations even if you end up stating the wrong type of noise.



(a) cameramanA.png (b) cameramanB.png (c) cameramanC.png

(d) cameramanD.png (e) cameramanE.png (f) cameramanF.png

Figure 1: Noisy cameraman

Question 1

Given any of the noisy cameraman images from Fig. 1a-1f, outline step-by-step how you would go about determining the type of noise present?

Question 2

Write a code to analyze the image in any way you deem appropriate (perhaps following your outline from Question 1?).

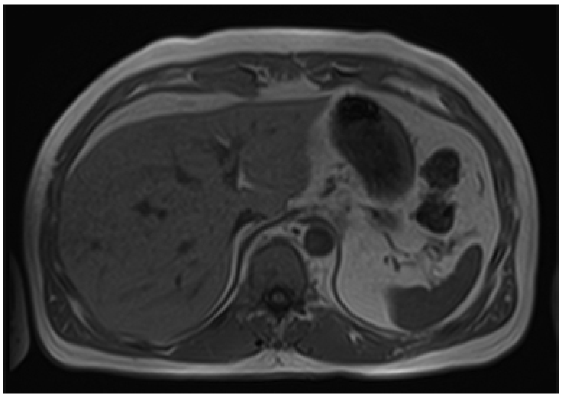
Question 3

Use the previous two questions to answer the following: What type of noise is present in each of the images in Fig. 1? Justify your answers with plots/arguments, etc.[1]

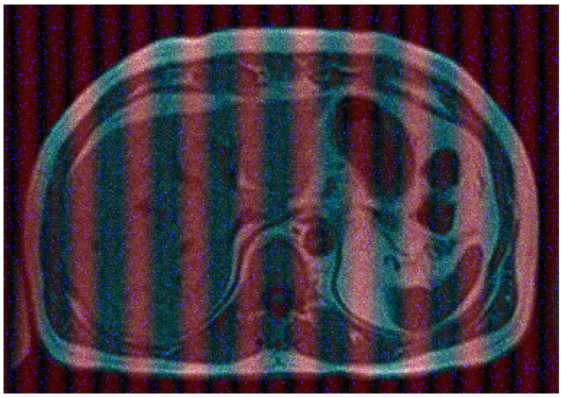
[1]HINT: Chapter 5 of the textbook may be used as an inspiration.

Task 2

In this task we will look at identifying noise, removing noise, and enhancing images post noise removal. You are provided the image



(a) Original image



(b) Image with noise named NoisyLiver.png

Figure 2: CT liver scan

Question 1

In general, what is image noise? How can it occur? Give a few examples for sources of noise.

Question 2

In the *NoisyLiver.png* image (Fig 2b) there are three noise types. Use what you've learned in this course and find evidence for three types of noise. Note that while visually analyzing the image is valid in a normal setting, we're looking for more in-depth analysis.

Question 3

Given the noises you have found, what are some great filters / techniques of dealing with them? Provide the kernels if applicable, and give a theoretical argument for why it works well against the noise you've found.

Question 4

Write the code to denoise the image. Try to get as close to the original as you can. Provide the final image and asses how well your restoration is. Did you get a perfect recreation? If not, why not?

Question 5

Finally we want all the features to "pop" a bit more. Choose one spatial and one Fourier edge enhancement technique[2]. Give a brief explanation on how they work, and apply them to your reconstructed image. Did one of your choices work better than the other? Discuss the results.

[2]This is to say, one method which is applied in the spatial domain, and one that is applied in the Fourier domain

Task 3

(a) Original image (b) Filtered Image A (c) Filtered Image B

Figure 3: Spatial Filtering of Cameraman.

The image in Fig. 3a has been filtered as seen in Fig. 3b and Fig. 3c.

Question 1

What type of filter would have an effect on the image in Fig. 3a as shown in Fig. 3b and Fig. 3c?

Question 2

What is the difference between the images in 3b and Fig. 3c and what was different in the filtering process to cause this?

The 2-D DFT can be expressed in polar form as

$$\begin{aligned} F(u,v) &= R(u,v) + iI(u,v) \\ &= |F(u,v)| \exp(i\phi(u,v)) \end{aligned}$$

where the magnitude is given by

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{\frac{1}{2}}$$

and the phase by

$$\phi(u,v) = \arctan\left[\frac{I(u,v)}{R(u,v)}\right].$$

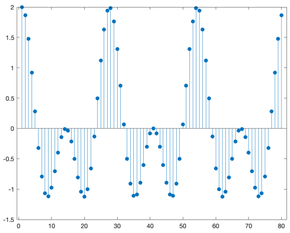
Load the images `robot.jpg` and `pig.jpg`. Turn both of those color images into grayscale and resize `robot.jpg` to be of the same size as `pig.jpg`. Find the magnitude and phase responses of both of those images and compute the frequency responses by swapping the phases of the two images, i.e. construct one image with the magnitude of `robot.jpg` and the phase of `pig.jpg`, and vice versa. Transform both of these frequency responses into the spatial domain and visualize the real part of the result.

Question 3

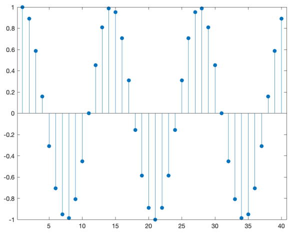
Include the two resulting images. Do not forget to note which one is which.

Question 4

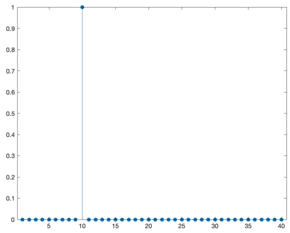
Based on your observation in this exercise, is it the phase or the magnitude that encodes the dominant structures of the image?



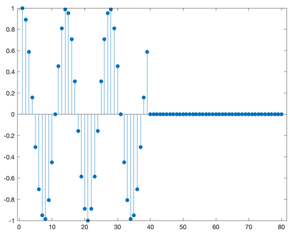
(a)



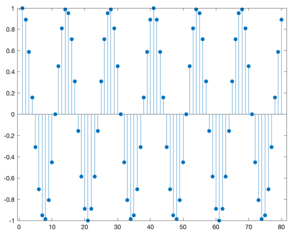
(b)



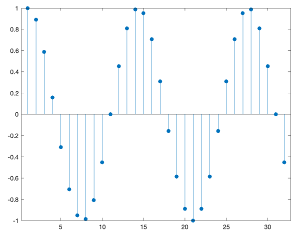
(c)



(d)

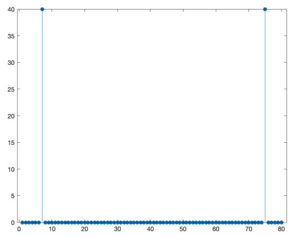


(e)

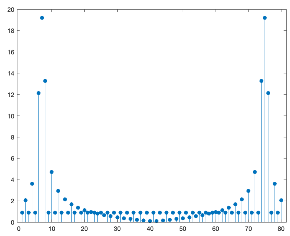


(f)

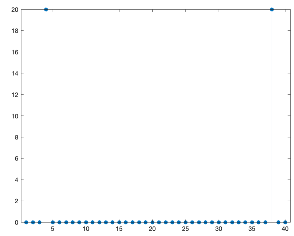
Figure 4: Signals



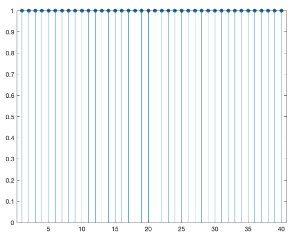
(i)



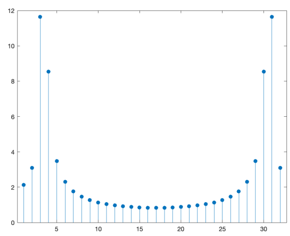
(ii)



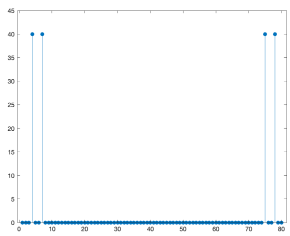
(iii)



(iv)



(v)



(vi)

Figure 5: DFTs of signals in Fig. 4

Question 5

Match the signals shown in Fig. 4 with the DFTs shown in Fig. 5.

Question 6

For all pairs, explain in words why the signal in Fig. 4 and the corresponding DFT in Fig. 5 are a match.

Task 4

Question 1

Upload the image (Fig. 6) and perform a 3-level discrete wavelet decomposition (DWT) (see the Multiresolutional Analysis - MRA - `pywt.wavedec2()`). Import necessary libraries: `numpy`, `matplotlib`, `skimage`, and `pywt`. Assess whether it is possible to detect noise in any of the decomposition coefficients. If noise is detectable, describe how it can be identified at the decomposed level of the image. Visualize the results of this analysis through the corresponding plots.

Figure 6: Image with a diagonal noise

Figure 7: Image for a discrete wavelet transform

Question 2

Calculate by coding the maximum number of decomposition levels for the image (Fig. 7). Explain by computation the process of estimating the maximum number of decomposition levels.

Question 3

Randomly select one of the detailed coefficients (horizontal, vertical, or diagonal) from an image and modify it by setting one of its values to zero. Then, apply a DWT at decomposition levels 3, 5, and 8. Finally, plot the recovered images after each of these modifications at the specified levels of decomposition. Explain the difference in the results of the recovered images.

Task 5 Computed Tomography

In this task, you need to help the hospital which has lost its engineer specialist of medical images in a snowstorm. The CT scan machine is working but the software has been corrupted. The output of the sensors can not be converted into an image any more. You get the sinogram, but the reconstruction of the image, based on the inverse Radon transform is not working. Fortunately, you have learnt this step in your studies and you decide to help the hospital. One of the doctors gives you a file containing a sinogram (`sinogram.npy`) so that you can experiment on it. To save time, you ask a chatbot to provide the code you need. It gives you the code in file `CT_code.py` but you need to double check it before you give it to the hospital.

Question 1

Load the sinogram and plot it as a 2d image. Don't forget to label the axes. (you may get some hints on how to do it in the code provided). Run the code provided by chatbot. It should work but do you get a good image? Show the reconstructed image and comment it.

Question 2

Inspect the python function and use you knowledge of computed tomography to check what is wrong. Make some changes to it to get a better reconstructed image. Show the resulting picture and explain what you changed in the function. Explain your choices. You may display different images if that helps making your point.

Question 3

At the hospital, the data recording can be modified to make the aquisition faster and reduce the Xray dose received by the patient. In that case, projections with fewer angles are recorded. Therefore, a sinogram with less lines (less angles) is outputted. To simulate this and check that the code is dealing correctly with this case, you need to downsample the sinogram and run the code again. Use the following function: `sinogram_fast = sinogram[::4,:].copy()` to downsample by 4 the number of projections.

Run the code on the downsampled data. Some artificial lines should appear on the reconstructed image now. Explain why. Find new parameters for the reconstruction function so that the lines are removed or reduced and you get the best image possible. Show the images you obtained while experimenting, with the parameters you used to get them. Motivate your choices and explain why your final result is good. Do not give only the final best result, show several images from your experiments and explain each time what happens. Show at least 4 cases. Showing only the final result will give you less points. No explanation will result in no point given.