Image Processing: Assignment #4

Submission date: 10\03\2024, 07:59.
Please submit the assignment via Moodle.
For questions regarding the assignment please contact:
Alon Papini (imageprocessinghaifau@gmail.com).

Assignment Instructions:

- Submissions in ***pairs***.
- The code must be written in Python 3.11 or higher.
- The code must be reasonably ***documented***.
- Please submit one single ***.zip*** whose name should be using this format: ID1_ID2_HW4.zip. It should contain ***only***:
  - 'q2' folder, and inside it will be:
    - The original image, 'zebra.jpg'
    - The plot result, 'zebra_scaled.png'
    - Your python script, 'zebra.py'.
  - 'q3' folder, and inside it will be:
    - 'main.py' and the original image folder you were given.
    - Your python script, 'BadImagesFixing.py'
  - The PDF file where your written answers will be (including all images you were asked to create or comment on).

Assignment Topics:

- Fourier transform
- Filtering in the frequency domain

Good Luck 😊

Problem 1 – Understanding Fourier (12 points):

Note: The following questions are relatively open questions (think HW1 question 1).

a) By now we've covered two methods for scaling an image:

   1. Using geometric operations to scale pixel coordinates with interpolation.
   2. Changing the Fourier transform of the image and then inverse transforming it back to get the scaled image.

   Which method is better in most cases, and why?

   Note: this question is a relatively open question (think HW1 question 1).

b) We've learned in the lectures that the Fourier transform is unique for each image (In other words, if we are given two Fourier transforms such that $G(u, v) = F(u, v)$, then $f(x, y) = g(x, y)$). Please explain in your words why that is – why is the Fourier transform unique for each function/image?

Problem 2 – Scaling the Zebra (18 points)

We are going to work with two different scaling methods in the frequency domain and understand the differences between them (as discussed in tutorial 6).

Recall the zebra image from Fourier demo in tutorial 6, with size (H,W):

In zebra.py you'll find several plt.imshow calls for 'image'. You'll need to display several different images, so please do the following:

a) Show the original image with its Fourier transform.

b) Use zero padding to scale the image into size (2*H,2*W), and then plot both the Fourier transform and the image itself.

c) Now, use the Fourier transform scale formula:

$$F[f(ax, by)] = \frac{1}{|ab|} F(\frac{u}{a}, \frac{v}{b})$$

and generate an image with size (2*H,2*W) containing 4 copies of the zebra, as seen in tutorial 6.

Hint: when resizing the image using the formula, what are the values of u,v that you should work with? Does u correspond to x, or to a*x?

Recall the first two questions from tutorial 6. Remember the differences? Please include the final plot of all images you're required to show and explain the differences between both methods scaling in the PDF file.

For this question you'll probably want to use the following functions:

- To perform the fft: 'fourier_transform = fft2(gray_array)'.
- Shift to the center: 'fftshift(fourier_transform)'.
- To display: 'img=np.log(1+np.absolute(fourier_transform))'.
- To restore: 'source=np.abs(ifft2(ifftshift(larger_fourier)))'.

Problem 3 – Fix me! (70 points):

You are given 7 bad images to enhance/clean/restore inside the 'q3' folder.

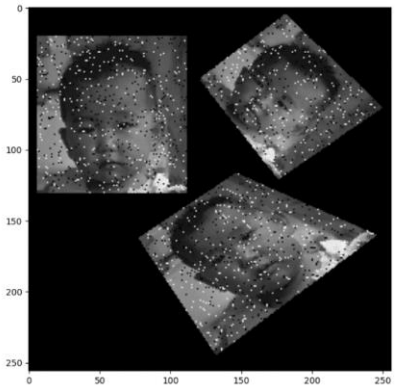
You can use anything we learned: Gamma corrections, Histogram equalization, filters convolutions (gradient, median, bilateral, sharpening… etc), Fourier transform and manipulations in frequency domain.

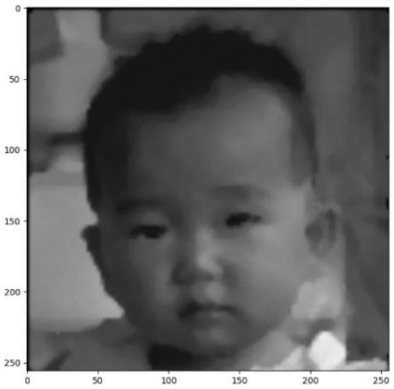
Modify ‘BadImagesFixing.py’ to clear the images. You can test it using ‘main.py’

The format of your submission for this question in the PDF should be something like this: The original bad image, a screenshot of your fixed version of it (use ‘main.py’ to get that image or write your own python code to save the image and then attach it) and an explanation of about 1-3 lines on how you fixed it.

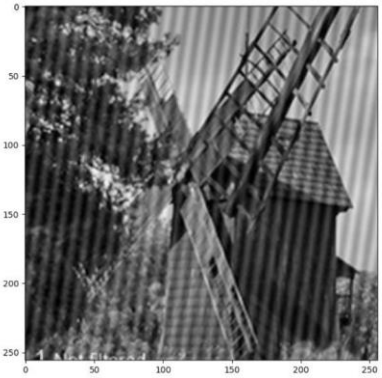
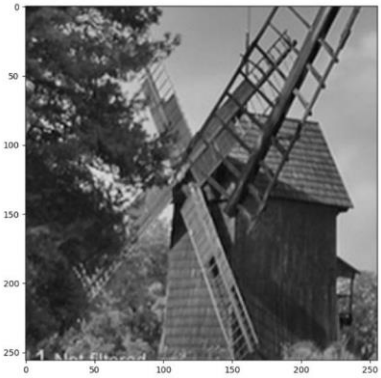
Grading will be given based on the optimality of your method/result. Here you are given the original images next to a (not entirely optimally) fixed version, with some hints:

<u>Baby - There are 3 images, utilize **all three**</u>:

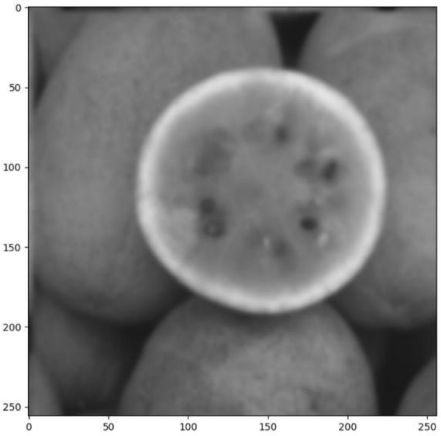
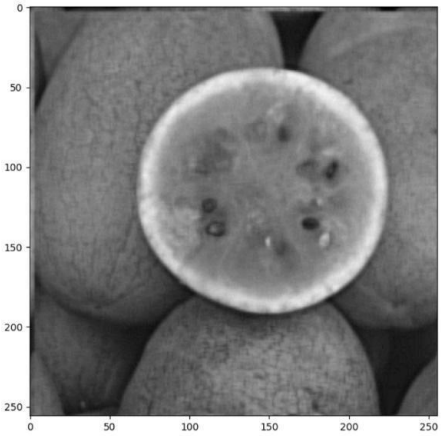




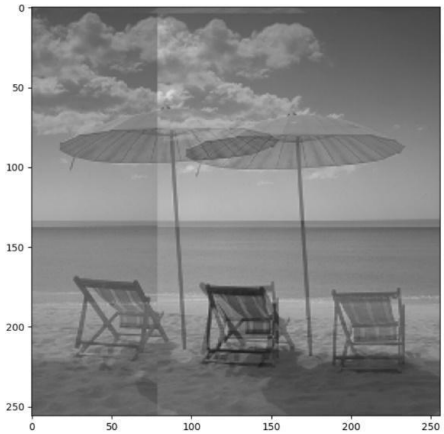
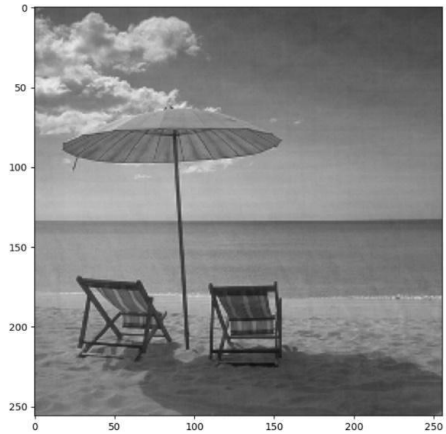
Windmill - This is an image with a noise of a very specific frequency:

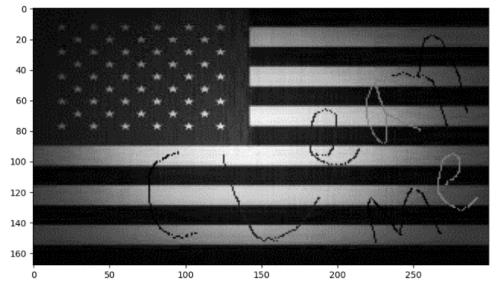
Watermelon - Which filter sharpens an image (Went over this in the tutorial):

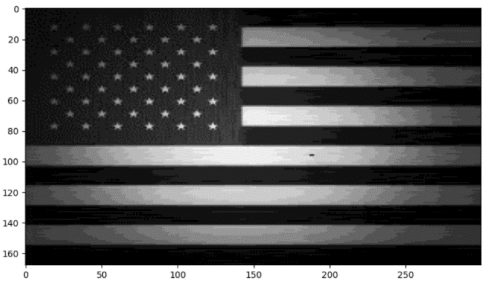



Umbrella - The noised image is a result of an average between original image and a shift of the original image:

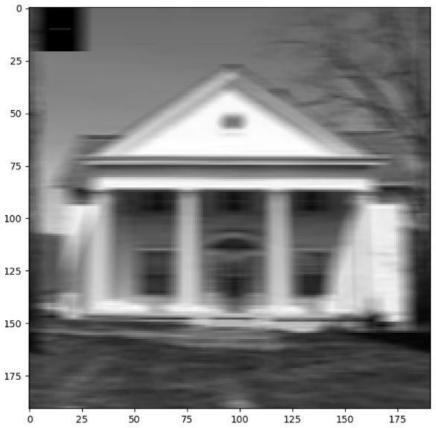



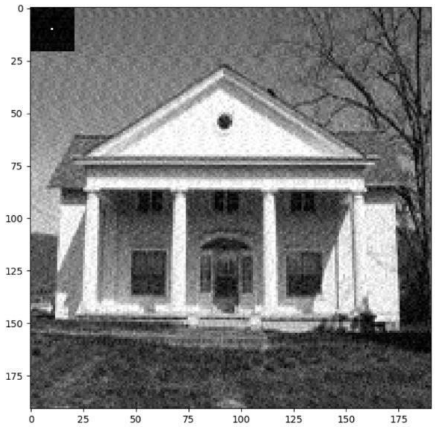
USAFlag - Clean me:





House - Someone took 10 images of a house while moving (motion blur), the result is 10 shifted images averaged in one:





Bears - Pretty dark in here, like all the gray values are low: