Middle East Technical University Department of Computer Engineering

CENG 466

Fundamentals of Image Processing

Fall '2021-2022

Take Home Exam 2

Due date: December 24 2021, Friday , 17:00

1 Objectives

The purpose of this assignment is to familiarize you with the fundamental frequency domain techniques. You are given several tasks. You are expected to carry out these tasks using algorithms in lecture notes and the text book.

2 Specifications

You are given two questions, which you should solve with your own algorithms. In addition to the solutions, you are required to prepare a report that explains your methodology and includes the analysis of the results and your comments on them. The report should be **maximum 10 pages** long and should be prepared in IEEE Conferance Proceedings Template (LaTeXis recommended) provided in the following link.

`https://www.ieee.org/conferences_events/conferences/publishing/templates.html`

- Grading will be based on the quality of the outputs, script contents and the report
- The report should clearly explain the methodology and rationale behind the algorithm design. It should also explain the difficulties encountered in the design, implementation and experimentation stages, and your solutions on them. Last but not least, the report should contain your comments on the results. Even if the results does not match your expectations you should discuss the encountered situation.
- In the following sections you will be asked to implement several functions. Implement them in a single file called ***the2.py***. This file can contain any additional functions. Submit this file with your report. Do not submit input or output images.

3 Frequency Domain Image Filtering (50 Points)

In this part you will be implementing frequency domain filtering.

3.1 Edge Detection (25 Points)

In this part you will implement an edge detection in frequency domain. In order to complete this task follow the given steps:

Step 1: Define a function whose input will be the path of the original gray-scale image, and output path. You will save the edge map o the image to output path, also return this edge map

```
def part1(input_img_path, output_path):
    '''
    Reads the image I from input image path
    Applies edge detection in frequency domain
    Saves the edge map to output path
    Returns the edge map
    '''

    return processed img
```

Step 2: Read the input image from the given path. (You can use any python libraries such as cv2 or PIL). Note that the images will be provided in gray-scale in this question. Create the output directory (output_path) if it does not exist. (You can use os for this purpose).

Step 3: Apply any edge detection algorithm in frequency domain.

Step 4: Extract the edge map of the given image. Save the output as an image to output_path with the name ***'edges.png'***.

Step 5: **TESTING:** Use provided images 1.png and 2. png (see Figure 1a and 1b) to test your function.

Step 6: **REPORT:** Report and discuss your findings. Compare your results with the ones you have obtained in THE1.

3.2 Noise Reduction (25 Points)

In this question you will work on specific images. You will not write generic functions.
You are given two noisy images named as 3.png and 4.png (see Figures 1c and 1d). Your task is to enhance these images in Fourier domain. For the sake of easy evaluation please follow the steps below;

Step 1: Define two functions whose inputs will be the path of the images(3.png and 4.png), and the output path

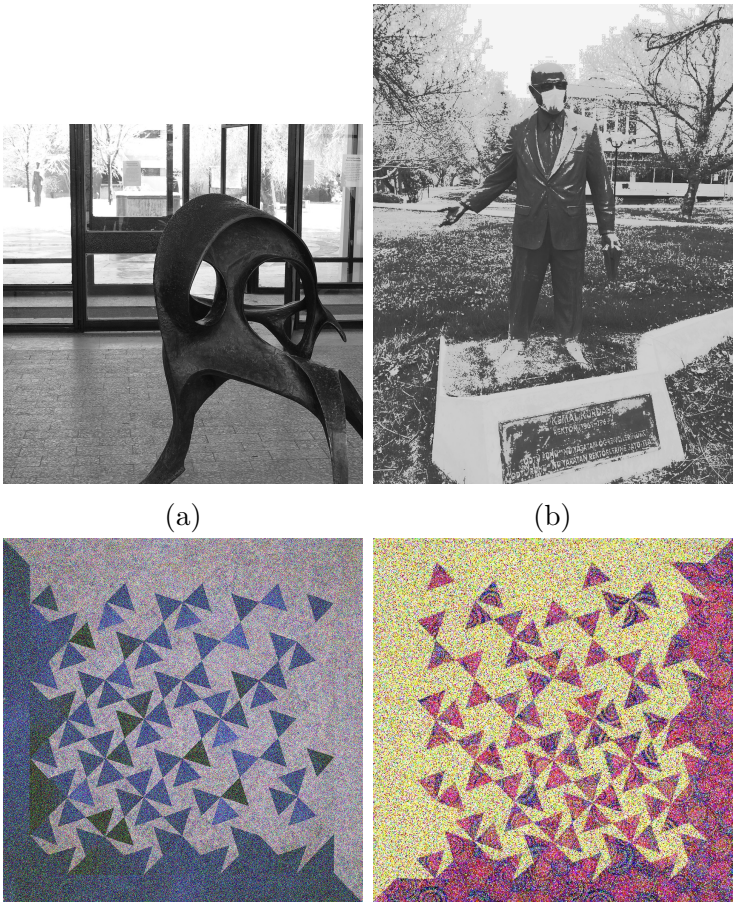
```
def enhance_3(path_to_3, output_path):
    '''
    Read image 3.png from input image path
    It will be given such as './THE1_images/3.png')
    Apply image enhancement
    Return the enhanced image
    '''
def enhance_4(path_to_4, output_path):
    '''
    Read image 4.png from input image path
    It will be given such as './THE1_images/4.png')
```

```
    Apply image enhancement
    Return the enhanced image
    '''
```

Step 2: Read the input image from the given path. (You can use any python libraries such as cv2 or PIL). Create the output directory (output_path) if it does not exist. (You can use os for this purpose).

Step 3: Analyze the type of the noise in the images and denoise them in Fourier domain. Note that you should examine each channel of the images. You an use any combination of filters.
Hint: Recall that spatial domain convolution is equal to the Fourier domain multiplication. You can use the same filters you have used in THE1. However you can use other techniques if you find suitable.

Step 5: **REPORT:** Report and discuss your findings. Compare your results with the ones you have obtained in THE1.



(a) (b)

(c) (d)

Figure 1: Given images for THE2. a and b are provided in gray-scale, whereas c and d are RGB images

4 JPEG Image compression (50 Points)

In this question you will implement jpeg image compression
For the sake of easy evaluation please follow the steps below;

Step 1: Define a write function, which will read an RGB png image, compress and saves in the new format. Create the output directory (output_path) if it does not exist. (You can use os for this purpose).

```
def the2_write(input_img_path, output_path):
    '''
    Reads the image I from input image path
    Applies jpeg image compression
    Saves the compressed image with any name (You can choose the ↩
        file extension)


    examples:
    input_image_path = './THE2_images/5.png'
    output_path = './TH2_images/compression_outputs/'

    img_name = Full path of the compressed image (Including the file↩
        name)
    '''

    return img_name
```

Step 2: Write function should compress the image in jpeg format. In order to achieve this you will,

- Convert the color space
- Apply discrete cosine transformation
- Apply Huffman Coding

Note that; You can alter this algorithm as long as you discuss your reasoning in your report. If you do so, please compare your results with original jpeg compression.

Step 3: Read function should print the amount of compression in the following format

```
Original size, compressed size, compression ratio
```

Step 4: Implement a read function, which will read the compressed image and show it to the user.

```
def the2_read(input_img_path):
    '''
    Reads the compressed image I from input image path
    Applies decompression
    Shows the read image

    '''
```

Step 5 **Testing:** Use the provided image '5.png' (See Figure 2) to evaluate your method.

Step 6 **Report:** Report and discuss your findings. As you know, jpeg compression is a lossy method. Please be specific on the amount of loss and compression ratio you have encountered on the sample image (5.png)

Figure 2: Image to be compressed

5 Regulations

1. **Group:** You are required to do your assignment in a group of two students. If there is an unclear part in your code, we may ask any of the group member to describe that code segment. Also group members may get **different** grades. We reserve the right to evaluate some or all of the groups to determine the contribution of each group member to the assignment.

2. **Programming Language:** You must code your program in Python. Your submission will be tested on department lab machines. You are expected make sure your code runs successfully on department lab machines.

3. **Late Submission:** Late Submission is **not** allowed!

4. **Newsgroup:** You must follow the odtuclass for discussions and possible updates on a daily basis.

6 Submission

Submission will be done via Odtuclass. Submit 'the2.py' (which includes all your functions), and your report **Do not send the input and output images.** Only one member should submit the homework. Hence, do not forget to **write your names and student id's at the beginning of the scripts.**

7 Cheating

We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations.

Cheating Policy: Students/Groups may discuss the concepts among themselves or with the instructor or the asistants. However, when it comes to doing the actual work, it must be done by the student/group alone. As soon as you start to write your solution or type it, you should work alone. In other words, if you are copying text directly from someone else - whether copying files or typing from someone else's nots or typing while they dictate - then you are cheating (committing plagiarism, to be more exact). This is true regardless of whether the source is a classmate, a former student, a website, a program listing found in the thrash, or whatever. Furthermore, plagiarism even on a small part of the program is cheating. Also, starting out with code that you did not write, and modifying it to look like your own is cheating. Aiding someone else's cheating also constitutes cheating. Leaving your program in

plain sight or leaving your computer without logging out, thereby leaving your programs open to copying, may constitute cheating depending upon the circumstances. Consequently, you should always take care to prevent others from copying your programs, as it certainly leaves you open to accusations of cheating. We have automated tools to determine cheating. Both parties involved in cheating will be subject to disciplinary action. [Adapted from http://www.seas.upenn.edu/cis330/main.html]