INT3404E - Image Processing: Homework 2

Le Cong Thuong

April 2024

1 Homework Objectives

Here are the detailed objectives of this homework:

1. To achieve a comprehensive understanding of how basic image filters operate.

2. To gain a solid understanding of the Fourier Transform (FT) algorithm.

2 Setup

It is assumed that you have set up a virtual environment. You can install the necessary packages using the command:

```
pip install -r requirements.txt
```

Additionally, you may choose to upload the notebook to Google Colaboratory for execution if you prefer not to run it on your local machine.

3 Details

3.1 Image Filtering

The exercise will help you understand basic image filters by manipulating box/mean and median filters. You will implement the first two problems in the provided Python script file (ex1.py), Specifically, you are required to:

- Implement one Replicate padding

- Implement the box/mean and median filters for removing noise.

- Implement the evaluation metric

(a) Implement the following functions in the supplied code file: $padding\_img, mean\_filter, median\_filter$. Figure 1 illustrates replicate image padding used in the exercise.

Replicate padding

Figure 1. Illustration of replicate padding

(b) Implement the Peak Signal-to-Noise Ratio (PSNR) metric

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right)$$

where MAX is the maximum possible pixel value (typically 255 for 8-bit images), and MSE is the Mean Square Error between the two images.

(c) Considering the PSNR metrics, which filter should we choose between the mean and median filters for provided images?

3.2 Fourier Transform

In this exercise, you will implement the Discrete Fourier Transform (DFT) algorithm from scratch. The goal is to familiarize yourself with the fundamental concepts and procedural steps involved in applying this algorithm. You will implement the first two problems in the provided Python script file (ex212.py), and the remaining two problems will be completed in the provided Jupyter notebook (ex223.ipynb).

3.2.1 1D Fourier Transform

The Discrete Fourier Transform:

$$F(s) = \frac{1}{N}\sum_{n=0}^{N-1} f[n]e^{-i2\pi sn/N}$$

and its inverse by:

$$f[n] = \sum_{s=0}^{N-1} F(s)e^{i2\pi sn/N}$$

where s represents the frequency, and n denotes the sampling order.

$$\begin{bmatrix} F_0 \\ F_1 \\ ... \\ F_N \end{bmatrix} = \begin{bmatrix} e^{-i2\pi 0*0/N} & e^{-i2\pi 0*1/N} & e^{-i2\pi 0*2/N} & ... & e^{-i2\pi 0*N/N} \\ e^{-i2\pi 1*0/N} & e^{-i2\pi 1*1/N} & e^{-i2\pi 1*2/N} & ... & e^{-i2\pi 1*N/N} \\ ... & ... & ... & ... & ... \\ e^{-i2\pi N*0/N} & e^{-i2\pi 1*N/N} & e^{-i2\pi 2*N/N} & ... & e^{-i2\pi N*N/N} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ ... \\ f_N \end{bmatrix}$$

Referencing the represenstation provided above, implement a function named DFT_slow to perform the Discrete Fourier Transform (DFT) on a one-dimensional signal.
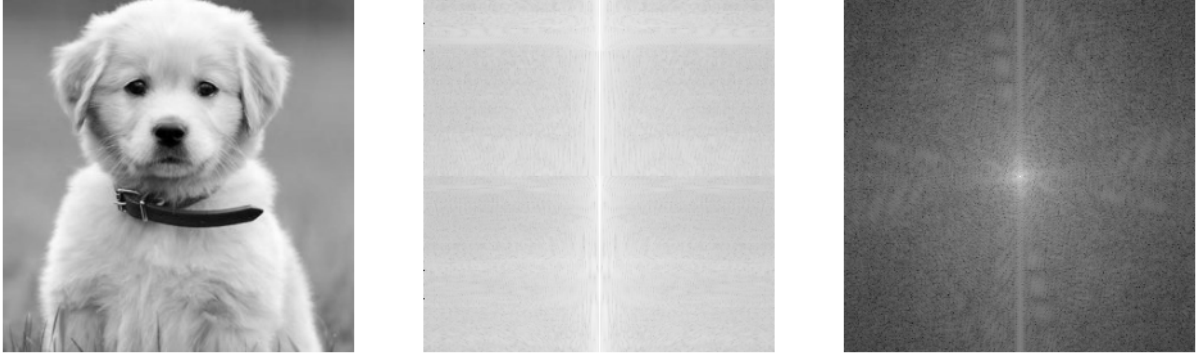
Figure 2. Expected Output for 2D Fourier Transform Exercise

3.2.2 2D Fourier Transform

Along with the scipy.fft, Numpy.fft provides functions related to the Fourier transform.

1. fft: Compute the one-dimensional discrete Fourier Transform.

2. ifft: Compute the one-dimensional inverse discrete Fourier Transform.

3. fft2: Compute the 2-dimensional discrete Fourier Transform.

4. ifft2: Compute the 2-dimensional inverse discrete Fourier Transform.

5. fftn: Compute the N-dimensional discrete Fourier Transform.

6. ifftn: Compute the N-dimensional inverse discrete Fourier Transform.

In this exercise, we aim to manually simulate the operation of the np.fft.fft2 function, which performs a two-dimensional Fourier Transform on a 2D signal. To achieve this, we will solely utilize the np.fft.fft function, designed for one-dimensional Fourier Transforms. Your task is to implement the function DFT_2D within the provided code. The procedure to simulate a 2D Fourier Transform is as follows:

1. Conducting a Fourier Transform on each row of the input 2D signal. This step transforms the signal along the horizontal axis.

2. Perform a Fourier Transform on each column of the previously obtained result.

   The output should look similar to Figure 2.

3.2.3 Frequency Removal Procedure

Follow these steps to manipulate frequencies in the image:

1. Transform using fft2

2. Shift frequency coefs to center using fftshift

3. Filter in frequency domain using the given mask

4. Shift frequency coefs back using ifftshift

5. Invert transform using ifft2

Implement the $filter\_frequency$ function in the notebook as described. The expected output should look similar to what is depicted in Figure 3.
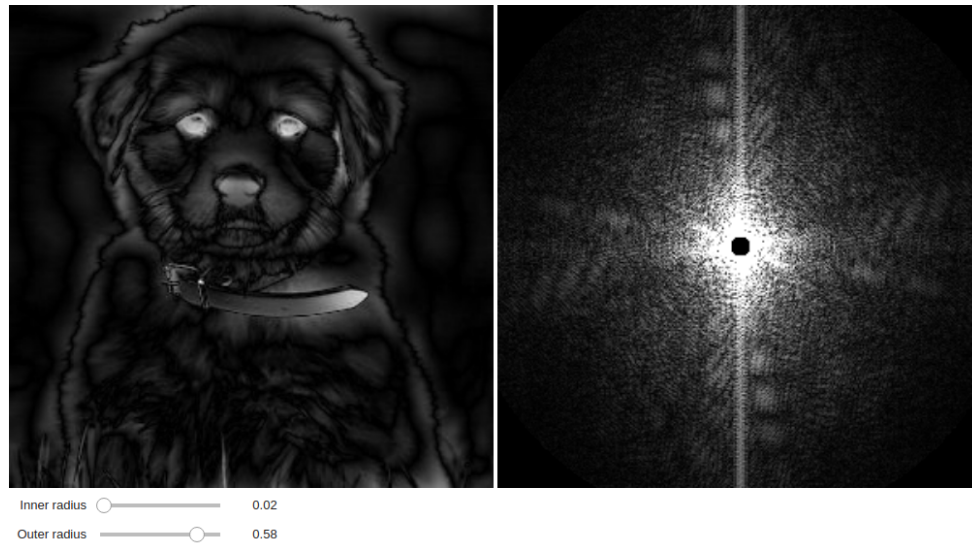

Figure 3. Expected Output for 2D Frequency Removal Exercise

3.2.4 Creating a Hybrid Image

To create a hybrid image, follow these steps:

1. Transform using fft2

2. Shift frequency coefs to center using fftshift

3. Create a mask based on the given radius (r) parameter, as described in Figure 5

4. Combine frequency of 2 images using the mask

5. Shift frequency coefs back using ifftshift

6. Invert transform using ifft2

Implement the function $create\_hybrid\_img$ in the notebook as instructed. The expected output should look similar to what is depicted in Figure 4. You are not required to use the provided images; feel free to use any images of your choice.

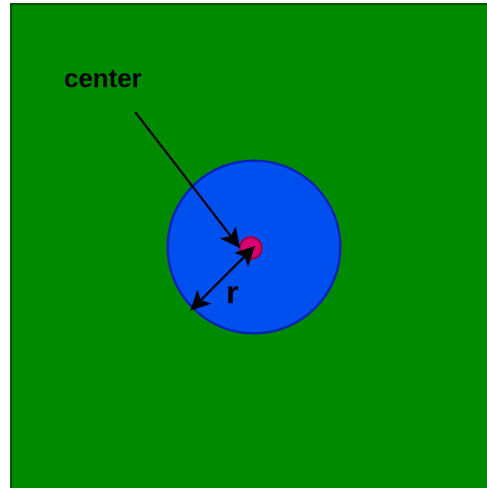Figure 4. Expected output of exercise Creating a Hybrid Image

Figure 5. Mask Creation: The blue area represents the frequencies from the first image, while the remaining area pertains to the frequencies of the second image.

Please note the following requirements:

1. Implement all specified functions.

2. Document the results and code for all functions using the provided LaTeX template, and upload them to the UET course.

3. Create a new directory named "HW2" within the same GitHub repository as "HW1". Upload all Python script files (.py), Jupyter notebooks (.ipynb), and your report to this directory.