**2110431 Introduction to Digital Imaging**

**2147329 Digital Image Processing and Vision Systems**

**Homework #2**

**Deadline : October 5th, 2024 @23:59**

**Submissions: (1) PDF version of this file**

**(2) .ipynb file; template in the link below**

COLAB TEMPLATE: <https://colab.research.google.com/drive/1vXUwb4AcX3vDvUTgzqP_1ag3aSHq7OvT?usp=sharing>

**Use these commands in colab to download the images.**

|  |
| --- |
| !wget https://drive.google.com/uc?id=1o0UMPTyUFzX9CaQp-BwYXgkCho1Zo6yL -O kitty55.png  !wget https://drive.google.com/uc?id=1\_-\_yN30miNhzR9ZC5DHTiljH6LVq4hZz -O clean\_cat.png  !wget https://drive.google.com/uc?id=1LEwFRI2vjSqQEd68lYwyuJ4JyJYPQX2m -O blurry\_noisy\_cat.png  !wget https://drive.google.com/uc?id=1lnO\_PK81O54bLBUprBlo8-2x9smWYDcP -O moodeng.jpg |

1. (2 points) Apply Gaussian low pass filter in frequency domain on “Kitty55.png” image which has pixels. Find the minimum cutoff frequency (C) that still maintain the total image power more than 99%. Where the total image power, is calculated by summing the components of spectrum power at each point , for and

is the spectrum power provided in the lecture slides

percent of the image power can be calculated from 100 x / , where is the total image power of the original image and of the filtered image

Put your results in the blank box below

Cutoff frequency (C) =

|  |
| --- |
| **27** |

=

|  |
| --- |
| **99.01590767196541** |

|  |  |
| --- | --- |
| Original Image (“kitty55.png”) | Fourier Spectrum of the original image |
|  |  |
| Fourier Spectrum of the filtered image | Filtered images ( > 99%) |
|  |  |

**2. Problem: Restoring blurry and noisy image**

You are provided with a blurred and noisy of cat image blurry\_noisy\_cat.png and a clean reference image clean\_cat.png. You have to restore the image using the **Wiener Filter** and compute the **Structural Similarity Index (SSIM)** between the restored image and the clean reference image.

A math equations on a white background

Description automatically generated

#### **SSIM Overview:**

The **Structural Similarity Index (SSIM)** is a perceptual metric that quantifies the similarity between two images. It considers changes in luminance, contrast, and structure to measure how close the restored image is to the original clean image. SSIM values range from -1 to 1, where:

* 1 indicates perfect similarity.
* 0 indicates no similarity.
* Negative values indicate dissimilarity.

(Hint: this metric is available inside skimage.metrics )

A group of cats in a crowd

Description automatically generated A group of cats in a black and white photo

Description automatically generated

Blurry Noisy Cat Clean Cat Image

# **Show how to restore the image using Wiener Filter from noise and blur effects and display the result (don’t worry if the output is not perfect, just select the best one in your thought),**

|  |
| --- |
| **Blurred\_img:** **Reconstructed\_img:**  **Methods:** 1. Convert the blurred image to the frequency domain using Fourier Transform.  2. Create a degradation filter (which might have caused the blur). Since it's likely a smoothing filter, I use a low-pass Gaussian filter in the frequency domain with a certain cutoff frequency.  3. Apply the Wiener filter (I use K = 0.01) using the blurred image and the Gaussian filter to calculate the restored image in the frequency domain.  4. Convert the restored image back to the spatial domain (original form).  5. Calculate the SSIM (Structural Similarity Index) between the original and restored image to see how well the restoration worked.  6. Loop over different cutoff frequencies to find the best SSIM value, which indicates the best restoration ("best" SSIM index may not be the clearest one. But this one looks good enough for me). |

# **Structural Similarity Index (SSIM) of the restored image**

# 

|  |
| --- |
| **SSIM org vs restored = 0.8498180213734453** |

# 

3. **WAVELETS AND MULTIRESOLUTION PROCESSING**

3.1 You are provided with Nong Moo Deng image moodeng.jpg. Your task is to perform **multi-level wavelet decomposition** using the **Discrete Wavelet Transform (DWT)** and analyze the different layers of decomposition. The DWT breaks the image into four sub-bands: Approximation, Horizontal details, Vertical details, and Diagonal details. You will progressively decompose the image into three levels, visualizing the components at each level.

A hippo with its mouth open

Description automatically generated

# **Your Wavelet Decomposition Level & Thresholding :**

|  |  |  |  |
| --- | --- | --- | --- |
| Original Image  **Level J** | Horizontal Detail Image | Vertical Detail  Image | Diagonal Detail  Image |
| **Level J-1** |  |  |  |
| **Level J-2** |  |  |  |
| **Level J-3** |  |  |  |

3.2 For level J-1, remove the right half of the three components - horizontal, vertical and diagonal details - of the ‘moodeng’ image and then apply inverse Discrete Wavelet Transform to reconstruct the original-sized image with a blur on the right half. You can blur the right side of the image using a grayscale image, but you will receive a 1-point bonus for producing an RGB output.

|  |
| --- |
| **METHOD:**  1. Given an RGB image of Moo Deng, we first separate the 3 color channels into R, G, and B.  2. We perform Haar wavelet decomposition on each channel to get the A, H, V, and D components of level J-1.  3. Find the midpoint of the level J-1 image (width // 2) and change the values of H, V, and D components [:, w//2:] to 0 (removing high-level details or edges to blur them). We do this for each channel.  4. Apply inverse wavelet decomposition with the modified A, H, V, and D from step 3 on each channel.  5. Merge the 3 reconstructed channels back together and plot the result. |
| **RESULTS:**  **ORIGINAL IMG:**    **RIGHT SIDE BLURRED IMG:** |