

DIGITAL IMAGE AND VIDEO PROCESSING LAB - EXPERIMENT NO. 4

FREQUENCY DOMAIN FILTERING

Submitted by Unnati Singh, 21EC39027

Overview:

This experiment focuses on the implementation of low-pass and high-pass filters in the frequency domain using the Fast Fourier Transform (FFT). By applying these filters, we can manipulate the frequency components of grayscale images to emphasize or suppress certain features. Additionally, the experiment demonstrates the creation of a hybrid image and the denoising of an image using FFT-based techniques.

Files Included:

1. `Exp4_21EC39027.py`: Python script containing all the functions required to perform low-pass and high-pass filtering, hybrid image creation, and image denoising.
2. `Exp4_21EC39027.ipynb`: Jupyter Notebook with the same code, allowing interactive execution and visualization.
3. `Exp4_21EC39027_report.pdf`: The experiment report detailing the objective, methodology, results, and conclusion.
4. `4_2_EXP.pdf`: Lab manual detailing the questions and expectations of outcome of this experiment.
5. `input/`: Directory containing the input images.
6. `output/`: Directory containing the output images obtained from the code.
7. `README.pdf`: This README file.

Functions Overview:

1. `lpf(lp_type, image_path, cutoff, order=1):`
 - a. **Purpose:** Applies a low-pass filter to an image, allowing only low-frequency components to pass through.
 - b. **Output:** The filtered image, highlighting the smooth areas by removing high-frequency noise.
2. `hpf(hpf_type, image_path, cutoff, order=1):`
 - a. **Purpose:** Applies a high-pass filter to an image, allowing only high-frequency components to pass through.
 - b. **Output:** The filtered image, emphasizing edges and other high-frequency details.

3. `hybrid(image1='input/einstein.png', image2='input/marylyn.png'):`
 - a. Purpose: Creates a hybrid image by combining the high-frequency components of one image with the low-frequency components of another.
 - b. Output: The hybrid image, displaying different features based on the viewing distance.
4. `denoising(img_path):`
 - a. **Purpose:** Denoises an image by filtering out specific frequency components associated with noise.
 - b. **Output:** A clearer version of the original image with reduced noise.