Sashi:

**Report on Image Processing Techniques and Filters**

**Introduction**

This report presents the implementation of various image processing techniques, including the application of filters and transformations, as outlined in the five questions. The goal is to explore different filtering techniques using both low-pass and high-pass filters, dithering algorithms, Kuwahara filtering, Fourier transform-based filters, and quantization. I have used Python, along with libraries like OpenCV, NumPy, and Matplotlib, to implement the required tasks. This report will discuss the methods applied for each question and provide answers to the theoretical parts where necessary.

**Question 1: Binary Mask and Filters**

**Task:**

* Create a binary mask for the region of interest in the image.
* Apply low-pass filters (Gaussian and Average) and high-pass filters (Laplacian and Prewitt) to the image.

**Approach:**

The image was converted to grayscale and a binary mask was created using a threshold value. The mask helped to identify areas of interest. Next, both low-pass and high-pass filters were applied.

* **Gaussian Filter**: A smoothing filter that reduces noise and detail.
* **Average Filter**: A basic smoothing filter that averages pixel values.
* **Laplacian Filter**: A high-pass filter that detects edges by calculating the second derivative of the image intensity.
* **Prewitt Filter**: Another high-pass filter used for edge detection, which calculates intensity gradients in the horizontal and vertical directions and combines them to highlight edges.

**Result:**

The binary mask and the filtered images were displayed, and the output images showed the expected results, with Gaussian and Average filters smoothing the image, while Laplacian and Prewitt filters highlighted the edges.

**Text Answer:**

* Low-pass filters like **Gaussian** and **Average** help in blurring and reducing noise, while high-pass filters like **Laplacian** and **Prewitt** enhance edges by highlighting intensity changes.

**Question 2: Dithering Algorithms**

**Task:**

* Implement and compare the Floyd-Steinberg and Jarvis-Judice-Ninke dithering algorithms.

**Approach:**

The original grayscale image was first loaded, and two dithering algorithms were applied:

* **Floyd-Steinberg Dithering**: An error diffusion algorithm that distributes quantization errors to neighboring pixels to achieve a more visually pleasant result.
* **Jarvis-Judice-Ninke Dithering**: A more complex error diffusion algorithm that spreads the error across a larger neighborhood, resulting in a smoother dithered image.

**Result:**

Both dithering algorithms were successfully implemented, and the results showed how these techniques convert a grayscale image into a binary-like image while maintaining a semblance of detail.

**Text Answer:**

* **Floyd-Steinberg** produces a sharper, more detailed result compared to **Jarvis-Judice-Ninke**, which provides a smoother output by spreading error across a wider area.

**Question 3: Kuwahara Filter**

**Task:**

* Explain what a Kuwahara filter is and apply it to the image.

**Text Answer:**

The **Kuwahara filter** is a non-linear smoothing filter used to preserve edges while reducing noise. It works by dividing the filter window into four overlapping regions, calculating the mean and variance for each region, and then choosing the mean value from the region with the smallest variance. This approach helps smooth homogeneous regions of the image while preserving edges.

**Approach:**

A custom implementation of the Kuwahara filter was applied to the grayscale image using a 5x5 window. The filter window was divided into four subregions, and the mean from the region with the smallest variance was used for each pixel.

**Result:**

The Kuwahara-filtered image showed effective smoothing while preserving important edges, as seen in the transition zones of the image.

**Question 4: Fourier Transform, Butterworth, and Gaussian Filters**

**Task:**

* Take an image, apply the Fourier Transform, and implement both Butterworth and Gaussian low-pass filters in the frequency domain.

**Approach:**

The image was first transformed into the frequency domain using the **Fourier Transform**. The zero-frequency component was shifted to the center using np.fft.fftshift(). Two filters were created:

* **Butterworth Low-Pass Filter**: A smoother transition filter that attenuates high frequencies based on a specified cutoff and order.
* **Gaussian Low-Pass Filter**: A low-pass filter based on a Gaussian function, which smoothly attenuates high-frequency components without abrupt changes.

After applying the filters, the inverse Fourier transform was computed to obtain the filtered images.

**Result:**

Both the Butterworth and Gaussian filters successfully blurred the image, with the Gaussian filter providing a smoother transition compared to the Butterworth filter, which exhibited a sharper cutoff in the frequency domain.

**Text Answer:**

* **Butterworth** filters have a smoother cutoff with more control over sharpness based on the filter's order, while **Gaussian** filters provide smoother blurring by reducing high-frequency components gradually.

**Question 5: Image Quantization**

**Task:**

* Take an image, quantize it to 32 grayscale levels, and describe the steps taken in the process.

**Approach:**

The image was quantized by reducing the number of grayscale levels from 256 to 32. This was achieved by dividing the pixel values by 8, truncating the values, and then multiplying them back by 8. This approach reduces the number of unique intensity levels while maintaining the overall structure of the image.

**Result:**

The quantized image retained most of the key details of the original image, but with fewer shades of gray. The visual difference between the original and quantized images was subtle, but noticeable in smoother regions of the image.

**Text Answer:**

The quantization process involves dividing the pixel intensities into fewer levels, in this case, 32. This was achieved by scaling the original intensity values down by a factor of 8, reducing the overall range of values.

**Conclusion:**

Through the completion of these five tasks, various image processing techniques have been successfully implemented and analyzed. Low-pass filters like Gaussian and Butterworth were found to effectively smooth images, while high-pass filters like Prewitt and Laplacian highlighted edges. Dithering algorithms helped to simulate shades of gray with binary values, and the Kuwahara filter provided an edge-preserving smoothing technique. Lastly, quantization demonstrated how reducing the number of intensity levels impacts image quality. The methods applied provide insight into the practical applications of these image processing techniques.