CS675: Computer Vision Assignment 1

Due date: March 6th at 11:59pm.
Topic: Image Filtering
Assignment Title: Image Filtering Techniques and Applications

Instructions:

1. **Download an Image:** Choose any color image. Ensure the image is in .jpg or .png format.
2. **Apply Filters:** You will apply the following filters to the image:
    - **Smoothing Filters:**
        - Mean Filter (using a 3x3 kernel)
        - Gaussian Filter (sigma = 1.0)
        - Median Filter
    - **Sharpening Filters:**
        - Laplacian Filter
        - Unsharp Masking
    - **Edge Detection Filters:**
        - Sobel Filter (X and Y directions)
3. **Add and remove high-frequency noise to the image:**
    - Generate a noisy color image (random number generation functions such as Numpy's *randint()*. However, noise generated by these random number generators span all frequencies of the spectrum).
    - Design a filtering approach to generate high-frequency noise from the noise generated above.
    - Add both noises to the image and compare them.
    - Design a filter that completely removes the high-frequency noise.
    - Plot the results.

Requirements:

Write a Python script using OpenCV, NumPy, and Matplotlib to apply different filters to an image. For each filter, display the original image and the filtered image side by side using Matplotlib. If the original image is colored, optionally plot the results in grayscale (encouraged). Compare the performance of each filter visually and in terms of pixel statistics such as mean and variance.

- Code Implementation: Use the following Python libraries for the implementation: **OpenCV** (cv2)**, Numpy, Matplotlib.**
- Example template for applying a filter:

```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Load the image
img = cv2.imread('image.jpg', 0)  # 0 for grayscale, 1 for color

# Apply a filter (e.g., Gaussian)
filtered_img = cv2.GaussianBlur(img, (3, 3), 1)

# Display original and filtered images
plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(img, cmap='gray')

plt.subplot(1, 2, 2)
plt.title('Gaussian Blurred Image')
plt.imshow(filtered_img, cmap='gray')

plt.show()
```

Submission Instructions:

- **Format:** Submit your work as a **Jupyter Notebook (.ipynb)** file, ensuring all code cells are executable. Additionally, export the executed notebook as a **PDF or HTML report**, where all outputs and results are visible.
- **Structure of Submission:** Each task (**2 & 3**) in your assignment should follow this structured approach:

    a. Introduction and objectives
    b. Implementation with justification
    c. Results and analysis
    d. Discussion of findings

    At the end of the report, provide a conclusion summarizing key findings and reflecting on the effectiveness of the approaches used, highlighting their strengths and limitations.

    e. Conclusions

Evaluation Criteria:

Your assignment will be evaluated based on the following criteria:

1. **Introduction and objectives (5%)**
    - Clearly state the purpose of the assignment.
    - Explain what you are doing and why it is important. Provide context or background information if necessary.
2. **Implementation with justification (25%)**
    - Present a well-structured and efficient implementation. Ensure proper documentation and readability of the code.
    - Justify every critical decision made in the code. Explain the logic behind your approach, choice of algorithms, tools, or frameworks.
3. **Analysis of the results (30%)**
    - Explain the results in relation to the objectives. Present them clearly, using plots, graphs, or explanations as needed.
    - Highlight any patterns, trends, or key insights from the findings.
4. **Discussion of findings (40%)**
    - Critically evaluate the results and their implications.
    - Compare findings with expected outcomes (if applicable).
    - Discuss any limitations or challenges faced during the process. Provide suggestions for improvements or future work.

Additional Resources:

- OpenCV Documentation: https://docs.opencv.org/
- NumPy Documentation: https://numpy.org/doc/
- Matplotlib Documentation: https://matplotlib.org/stable/contents.html

---

Good luck and have fun experimenting with image filtering techniques!