UYG332 IMAGE PROCESSING FINAL PROJECT

General Overview

This **individual project** aims to evaluate your understanding of core image processing concepts and your ability to apply them using Python and the OpenCV library. You will be provided with a set of images and a list of image processing problems to solve. Your solutions should be presented in a **Jupyter Notebook** format.

Requirements & Guidelines

Programming Language & Libraries

- All tasks must be completed using **Python**.
- You are expected to use **OpenCV** as your primary image processing library.
- You are allowed to use other helper libraries like NumPy, Matplotlib, etc., where needed. Make sure all required libraries are listed in the README file.

Jupyter Notebook Format

- Submit your project as a single Jupyter Notebook (.ipynb).
- Organize your notebook clearly using Markdown cells to explain each step, your logic, and results.
- Include image visualizations before and after applying your methods.

Image Dataset

- https://drive.google.com/drive/folders/1AGBIoMgdp9IGnwJRXgEeGbT5AIijwg2x?usp=sharing
- You must use only the provided images in the dataset. Usage of external images or datasets is not allowed.

Use of OpenCV Functions

- You are **not required** to implement all algorithms from scratch.
- If OpenCV provides a function for a certain task, you **can use it** efficiently.

Academic Integrity

- This is an individual assignment. Sharing code or collaborating with others is not allowed.
- All submissions will be checked for plagiarism and AI-generated content.
- Any violation will result in disciplinary action in accordance with the university's academic honesty policy.

Submission Checklist

- Jupyter Notebook file (.ipynb)
- All code cells run without error
- Results are explained with comments and Markdown cells
- All output images and/or graphs are shown in the notebook
- A dedicated GitHub repository for your project which includes a single Jupyter Notebook and a README.MD file includes your name, surname, student id that explains how to use Jupyter Notebook (used libraries, requirements etc.)
- A simple .txt file named as **<StudentID>_<Name_Surname>.txt** (example: 2102021_Leo_DaVinci.txt) that includes link to your GitHub repository must be uploaded to UBIS via dedicated project activity.

Problems

Problem-1 (15 Points)

For this problem, use the image named as **tf2_engineer.jpg**.

1. Read and display the original image in colour.
2. Find the centre of the image (y_c, x_c) and print the intensity value at the centre.
3. Place a rectangular colour patch around the centre of the image. Height of the patch is 30 pixels; width of the patch is 40 pixels. Patch Colour (in Hex): #329ea8
4. Print the intensity value at the centre of the colour patch.
5. Display the new image with colour patch.

Problem-2 (15 Points)

For this problem, use the image named as **einstein.tiff**.

1. Read and display the original image in grayscale.
2. Find the negative of the input image.
3. Display the negative image.
4. From the original image get the intensity value of 5 random pixels, and from the negative image get the intensity value of same pixels. Print pixel values to show pixel values before and after.

Problem-3 (15 Points)

For this problem, use the image named as **pout.tiff**.

1. Read and display the original image in grayscale.
2. Apply log transform to original image and display the result.
3. Apply inverse log transform to original image and display the result.
4. Apply inverse log transform to image you get from step 2 and display the result.
5. Comment on the images you get from step 2 and step 4.

Problem-4 (15 Points)

For this problem, use the image named as **moon.tiff**.

1. Read and display the original image in grayscale.
2. Apply unsharp masking in spatial domain using below formulation:
$$g(x,y) = f(x,y) + k * mask(x,y)$$
You need to use three different k values. You can select the value of k arbitrarily.
3. Apply unsharp masking in frequency domain using below formulation:
$$g(x,y) = \mathcal{F}^{-1}[1 + k * H_{HP}(u,v)]F(u,v)$$
where $\mathcal{F}^{-1}$is the inverse Fourier transform, H_{HP} is the high-pass filter function in frequency domain, and $F(.)$ is the input image in frequency domain. You need to use three different k values. You can select the value of k arbitrarily (**BUT** use same k values from the step 2). Remember that when designing your high-pass filter, you need to select the D_0 (cut-off frequency). You can select the proper D_0 by trial or analysing the original input image in frequency domain. There is no ideal D_0 and k so while finding results, keep that in mind.
4. Display the result from the step 2 side by side with the result from the step 3 for each value of k. Example: Fig-1 (k = 0.2), Fig-2 (k=0.5), Fig-3 (k=1)

TIP: Slides from Week-4 and Week-5 would be a good starting point.

Problem-5 (15 Points)

For this problem, use the image named as **pcb.tiff**.

1. Read and display the original image in grayscale.
2. Identify and describe the type of noise (e.g., salt-and-pepper, Gaussian) using visual inspection and/or statistical tools (e.g., histogram).
3. Remove the noise using proper filtering sequence.

TIP: Refer to Week-4 slides for guidance on filtering techniques.

Problem-6 (25 Points)

For this problem, use the image named as **pollen.tiff**.

1. Read and display the original image in grayscale.
2. Comment on the problem presented at that image. While commenting on the problem, provide proofs (numerical, visual, statistical etc.).
3. Use two independent approaches to solve the problem.
4. For each solution, display the results.
5. For each solution, justify your answer by providing proofs (numerical, visual, statistical etc.).

TIP: For this question, there is not a single correct answer but only use the approaches we discussed in class. Avoid using AI-based denoising or enhancement techniques not discussed in class.

For any questions regarding the project, you can e-mail me directly using the batuhanhangun@aydin.edu.tr or asking during the lectures.

Lect. Batuhan Hangün