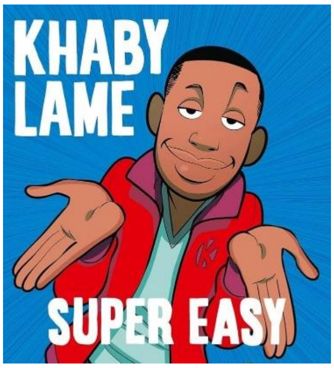
Assignment 2
Working with images in Spatial Domain

Please note:

1. What you must hand in includes the assignment report (.pdf) and – if necessary – source codes (.py or .m). Please zip them all together into an archive file named according to the following template: XXXXXXXX.zip
   Where XXXXXXXX must be replaced with your student ID.
2. Some problems are required to be solved *by hand* (shown by the icon), some need to be implemented (shown by the icon), and some need to be dockerized (shown by the icon).
3. As for the first type of the problems, you are free to solve them on a paper and include the picture of it in your report. Here, cleanness and readability are of high importance.
4. Your work will be evaluated mostly by the quality of your report. Don't forget to explain what you have done, and provide enough discussions when it's needed.
5. 5 points of each homework belongs to compactness, expressiveness and neatness of your report and codes.
6. By default, we assume you implement your codes in Python. If you're using Matlab, you have to use equivalent functions when it is asked to use specific Python functions.
7. Your codes must be separated for each question, and for each part. For example, you have to create a separate .py file for part b. of question 3. Please name it like p3b.py.
8. Problems with bonus points are marked by the icon.
9. If there is ***any*** question, please don't hesitate to contact us through the following email addresses:
   - Atghaei@aut.ac.ir
   - fardin.aiar@gmail.com
   - minoo.dolatabadi75@gmail.com
   - aidin.khalili@ymail.com
   - b.roshanfekr@aut.ac.com
10. Unfortunately, it is quite easy to detect copy-pasted or even structurally similar works, no matter being copied from another student or internet sources. Try to send us your own work, without being worried about the grade! ;)

1. Steganography Using Bit Plane Manipulation (15 Pts.)

Multimedia steganography refers to the techniques by which information is embedded within a digital cover signal in a way that prevents third-party observers from detecting differences between the original and the modified media. One of the simplest and most widely used steganographic techniques is Least Significant Bit (LSB) manipulation, where the least significant bits of an image are altered to hide secret data. In this assignment, you will explore bit-plane steganography, manipulate bit planes, and evaluate the detectability of the modifications.



a. Load the given image "q1.jpg" as an input image and convert it to grayscale. Then, extract and visualize all bit planes of the image.
b. Convert your full name (in English) into a binary message. Then, apply an XOR operation between your binary name and each bit plane separately. After modifying the bit planes, reconstruct the image by merging all the altered bit planes. Determine which modified images remain visually undetectable when compared to the original image.
c. Select one of your reconstructed images and extract the hidden message (your name) from it. Verify the correctness of the extracted message.
d. Modify your steganographic approach so that the secret message can be extracted without prior knowledge of its length. Implement a method to detect the hidden message's termination without additional metadata.
e. Discuss the trade-offs between imperceptibility, robustness, and capacity in bit-plane steganography. How does modifying different bit planes affect the balance between these three factors? Provide examples to support your discussion.

2. Advanced Histogram-Based Image Enhancement Techniques (15 Pts.)

Histogram Equalization (HE) is a widely used technique for contrast enhancement in digital images. However, it has several limitations, particularly in applications where maintaining mean brightness is crucial. This is evident in two real-world scenarios:

- **Historical image restoration:** Old photographs suffer from extreme contrast variations due to aging and scanning artifacts.
- **Medical imaging (X-ray enhancement):** Uncontrolled brightness shifts may obscure important anatomical structures.

Our goal is to explore and implement advanced histogram-based methods that enhance contrast while preserving brightness. The following techniques will be investigated:

- **Classical Histogram Equalization (HE):**

Given an input image $f(x,y)$ with intensity levels $L = \{0, 1, 2, \dots, 255\}$, HE defines a transformation function:

$$T(k) = round\left((L-1)\frac{\sum_{j=0}^{k} P(j)}{N}\right)$$

Where $P(j)$ is the probability density function of intensity j, and N is the total number of pixels.

- **Bi-Histogram Equalization (BBHE):**
  1. Compute the mean brightness m of the image:
  $$m = \frac{1}{N}\sum_{i=0}^{255} iP(i)$$
  2. Divide the histogram into two parts:
     a. H_1 for pixels where $f(x,y) \leq m$
     b. H_2 for pixels where $f(x,y) > m$
     c. Apply HE separately to H_1 and H_2, ensuring that the mapping function preserves the mean brightness.

- **Dualistic Sub-Image Histogram Equalization (DSIHE):**

Instead of using the mean brightness, **DSIHE finds a threshold** l_m that divides the cumulative histogram into equal pixel populations:

$$l_m = \arg\min \left|\sum_{i=0}^{l} P(i) - 0.5\right|$$

This ensures an equal distribution of pixels between two sub-histograms, reducing brightness distortion.

- **Brightness Preserving Dynamic Histogram Equalization (BPDHE)**
  1. Histogram Smoothing: Apply a Gaussian filter to smooth the histogram:
  $$G(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$$
  Where x is the coordinate and σ is the standard deviation.
  2. Local Maximum Detection: Identify peaks in the smoothed histogram for adaptive partitioning.
  3. Dynamic Histogram Partitioning: Divide the histogram into $n+1$ sub-histograms based on the detected peaks.
  4. Adaptive Equalization: Each sub-histogram is mapped to a dynamic range:
  $$range_i = L \times \frac{(span_i)}{M}$$
  ensuring controlled contrast stretching.
  5. Brightness Normalization: Adjust the output brightness to match the original mean:

$$g(x, y) = f(x, y) \times \frac{M_i}{M_o}$$

Where M_i and M_o are the mean brightness values before and after equalization.

Task Requirements

Implementation

a. Load and analyze both historical and medical images (q2_historical.png, q2_medical.png, q2_beach.jpg).

b. Compute the image histogram and cumulative histogram.

c. Implement HE, BBHE, DSIHE, and BPDHE, ensuring:

- Mean brightness preservation (for BBHE, DSIHE, and BPDHE).
- Adaptive contrast enhancement.
- Avoiding over-enhancement or saturation.

Comparative Analysis

- Visual Comparison: Display the results of all four methods and analyze the improvements.
- Statistical Evaluation:
    - Compute Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM).
    - Measure Mean Brightness Shift (MBS) to evaluate brightness preservation.
- Application-Specific Discussion:
    - Which method is best suited for historical restoration?
    - Which method performs better for medical X-ray images?

Deliverables

- Code Implementation: Develop a Python script to process images using all four methods.
- Visualization & Report:
    - Display input and output images.
    - Include histograms before and after enhancement.
    - Summarize key observations.
- Comparison Table: Evaluate methods based on contrast enhancement, brightness preservation, and computational efficiency.

Expected Outcome

- A well-structured comparison of classical HE vs. advanced histogram equalization methods.
- Insight into the trade-offs between brightness preservation and contrast improvement.
- Identification of the most effective technique for different real-world scenarios (historical images vs. medical imaging).

3. Create Your Own Pointillism Art (15 Pts.)

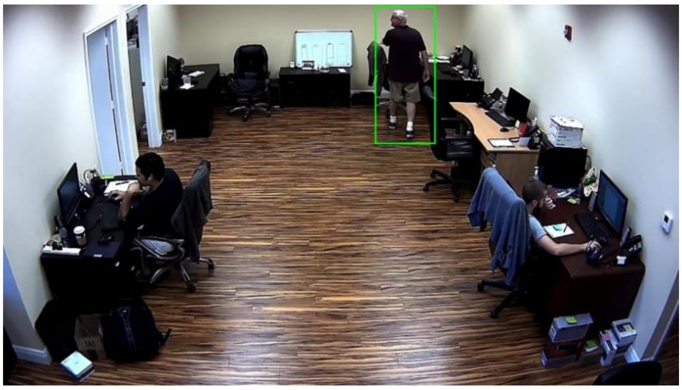
Pointillism is a technique of painting in which small, distinct dots of color are applied in patterns to form an image. In this assignment, you will create a unique piece of pointillism-style art using computer vision techniques. You will process an image, extract its edges using various spatial filters, and then render the image using randomly placed dots to mimic the pointillism effect.

Implement a function that takes an image, its corresponding edge-detected version, and the number of random points, then returns a pointillism-stylized image. To that end do the following steps:

a. Load and Convert Image to Grayscale and display it.
b. Implement and apply the following edge detection filters and display the edge-detected images.
   - Sobel (combined): Compute Sobel in both X and Y directions and combine them.
   - Laplacian: Apply the Laplacian filter to detect sharp edges.
   - Prewitt (combined): Compute Prewitt in both X and Y directions and combine them.
c. Implement the Pointillism Effect.
   i. Start with an empty white image of the same size as the original.
   ii. Generate random points on the image.
   iii. Draw circles at the generated points, using them as center coordinates, with a predefined radius.
   iv. Assign smaller radius for points on edges and larger radius elsewhere.
   v. Fill the circles with the color corresponding to the center coordinate pixel in the original image.
   vi. Return the final pointillism-stylized image.
d. Display the final pointillism effect image for the images “q3_1.jpg”, “q3_2.jpg”.
e. Experiment with different numbers of points and dot sizes to refine your effect.

4. Automated video surveillance (25 Pts.)

In an era where automated video surveillance is becoming increasingly vital, the capability to accurately detect and highlight motion is essential for both security and monitoring applications. This assignment challenges you to develop a motion detection system that leverages frame differencing—a straightforward yet powerful method—to identify significant movements in a video sequence. The goal is to detect substantial motion while filtering out minor, inconsequential changes, ensuring that only meaningful activities are captured.



The task requires the implementation of a simple motion detection algorithm applied to a provided surveillance video (q4.mp4). The method involves computing the difference between consecutive frames, which isolates areas of change by highlighting pixels that have altered in value. Due to the inherent noise present in video frames, it is necessary to apply an image smoothing technique, such as Gaussian blur, prior to the differencing process to enhance the reliability of detection. The system must be configured to ignore trivial movements, like slight head movements of seated individuals, and instead focus on detecting significant motion. After identifying the regions of interest, you should utilize contour detection techniques—using functions like cv2.findContours or their equivalents—to draw bounding boxes around these regions. The final deliverable is a video that displays the moving objects marked with bounding boxes, accompanied by a detailed report explaining the output results and the assumptions or design choices made during the implementation.

5. Image Denoising Using Spatial Filters (15 Pts.)

The goal of this assignment is to apply spatial filters to denoise images corrupted by different types of noise and analyze the effect of different kernel sizes using PSNR. To this end consider a sample image as the base image and do the followings ("q5.jpg"):

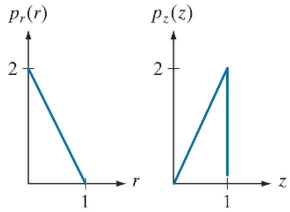
a. Add the following types of noise to the image:
   - Gaussian noise
   - Salt-and-pepper noise
b. Use OpenCV to apply the following spatial filters:
   - Mean filter (cv2.blur)
   - Gaussian filter (cv2.GaussianBlur)
   - Median filter (cv2.medianBlur)
   - Bilateral filter (cv2.bilateralFilter)
c. Experiment with different kernel sizes (e.g., 3x3, 5x5, 7x7) and observe the impact on denoising performance.
d. Compute the Peak Signal-to-Noise Ratio (PSNR) between the denoised image and the original clean image.

e. Compare the performance of different filters for each type of noise.
f. Discuss the effectiveness of each filter based on PSNR and visual results.
g. Explain the trade-offs between noise reduction and edge preservation for each filter.

6. Hands-On Image Processing: Exploring Visual Computation Without a (15 Pts.)

The goal of this assignment is to obtain a deeper understanding of fundamental image processing concepts by manually performing key operations without relying on a computer.

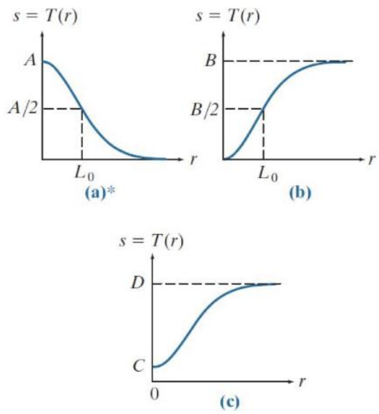
a. Explain why the discrete histogram equalization technique does not yield a flat histogram in general.
b. An image with intensities in the range $[0,1]$ has the PDF, $p_r(r)$, shown in the following figure. It is desired to transform the intensity levels of this image so that they will have the specified $p_z(z)$ shown in the figure. Assume continuous quantities, and find the transformation (expressed in terms of r and z) that will accomplish this.



c. The following kernel is separable. Find w_1 and w_2 such that $w = w_1 \star w_2$.

$$w = \begin{bmatrix} 2 & 1 & 1 & 3 \\ 4 & 2 & 2 & 6 \\ 2 & 1 & 1 & 3 \end{bmatrix}$$

d. Exponentials of the form $e^{-\alpha r^2}$ with α a positive constant, are useful for constructing smooth intensity transformation functions. Start with this basic function and construct transformation functions having the general shapes shown in the following three figures. The constants shown are input parameters, and must be included in your answer.



Good Luck
Happy New Year