E9 241 Digital Image Processing

Assignment 02

Due Date: September 25, 2025 - 11:59 pm **Total Marks:** 50

Instructions:

For all the questions, write your own functions. Use library functions for comparison only.

- Your function should take the specified parameters as inputs and output the specified results.
- Also provide the wrapper/demo code to run all your functions and obtain results. Your code should be self-contained i.e., one should be able to run your code as is without any modifications.
- For Python, if you use any libraries other than numpy, scipy, scikit-image, opencv, pillow, matplotlib, pandas, and default modules, please specify the library that needs to be installed to run your code.
- Along with your code, also submit a PDF with all the **results** (images or numbers) and **inferences** (very important: you may not be explicitly asked to give inferences in each question. You should always include your inferences from what you have observed). Include answers to subjective questions, if any.
- Put all your files (code files and a report PDF) into a **single zip file** and submit the zip file. Name the zip file with your name.

---

1. **Spatial Filtering and Binarisation:** Apply box blurring on the image '`moon_noisy.png`'. Generate a spatial box filter of size $m \times m$ by generating a filter kernel as:

$$B_f(x, y) = \frac{1}{K}, \tag{1}$$

where $x, y \in \{-(m-1)/2, \cdots, (m-1)/2\}$ and K normalizes the filter such that $\sum_x \sum_y B_f(x, y) = 1$. You can use a library function to convolve the input image with the kernel you created to obtain the box blurred image.

Apply Otsu's Binarization algorithm on the blurred image and note the optimal within-class variance $\sigma_w^{2\,*}$ for a given blur parameter.

Plot the histogram and the binarized image for the blurred images and find their corresponding optimal within-class variances $\sigma_w^{2\,*}$ for each filter size $m \in \{5, 29, 129\}$. Find the optimal filter size that minimizes $\sigma_w^{2\,*}$. Comment on your observations.

(15 Marks)

2. **Scaling and Rotation with Interpolation:**

   (a) Upsample the image '`flowers.png`' by a factor of 2 using bilinear interpolation and then rotate the result by 45° (clockwise or counterclockwise).

   (b) Rotate the image by 45° (clockwise or counterclockwise) and then upsample the result by a factor of 2 using bilinear interpolation.

Compute the difference between the two resulting images:

$$\text{diff} = \text{Result1} - \text{Result2}$$

Plot the difference and analyze:

- Observe the range of values of `diff` (minimum and maximum).
- What do you see when you plot using
  - `plt.imshow(diff, vmin=0, vmax=255)` in Python
  - The equivalent MATLAB plotting commands for `imshow` e.g.:
    `imshow(diff, [min_val, max_val]);`

Discuss the differences observed between the two operations and provide your comments whether the results are identical or not.

(15 Marks)

3. **Image Sharpening Concept:** Design a function `sharpenAdjust(img,p)` that takes an image and a parameter `p` as input, where `p` controls the sharpness of the image.

   In particular, the function should behave as follows:

   - `p=0` $\rightarrow$ Output is identical to the input image (no sharpening).
   - `p=1` $\rightarrow$ Strong sharpening is applied, such that halos (bands of white/black pixels around edges) may appear.
   - $0 < p < 1$ $\rightarrow$ Output varies gradually between no sharpening and strong sharpening.

   Test your implementation on '`study.png`' and comment on your observations for different values of `p`. Also try on few additional images from internet and report the observations.

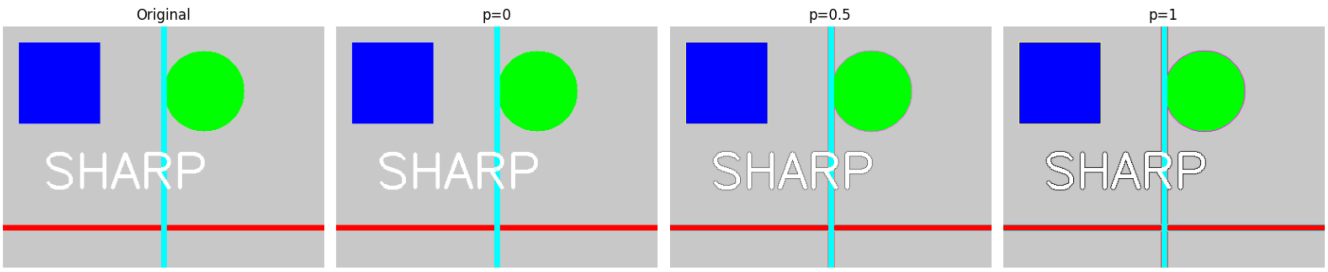

Figure 1: sharpened images with different p values

(20 Marks)