Amirkabir University of Technology
(Tehran Polytechnic)

Assignment 2
Pixel vs Kernel: Mastering in spatial domain!

Homeworks Guidelines and Policies

- ***What you must hand in.*** It is expected that the students submit an assignment report (HW2_[student_id].pdf) as well as required source codes (.m or .py) into an archive file (HW2_[student_id].zip). Please combine all your reports just into a single .pdf file.
- ***Pay attention to problem types.*** Some problems are required to be solved *by hand* (shown by the icon), and some need to be implemented (shown by the icon). Please do not use implementation tools when it is asked to solve the problem by hand, otherwise you will be penalized and lose some points.
- ***Don't bother typing!*** You are free to solve by-hand problems on a paper and include their pictures in your report. Here, cleanness and readability are of high importance. Images should also have appropriate quality.
- ***Reports are critical.*** Your work will be evaluated mostly by the quality of your report. Do not forget to explain your answers clearly, and provide enough discussions when needed.
- ***Appearance matters!*** In each homework, 5 points (out of a possible 100) belong to compactness, expressiveness, and neatness of your report and codes.
- ***MATLAB is also allowable.*** By default, we assume you implement your codes in Python. If you are using MATLAB, you have to use the equivalent functions when it is asked to use specific Python functions.
- ***Be neat and tidy!*** 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, which must be named 'p3b.py'. (or .ipynb)
- ***Use bonus points to improve your score.*** Problems with bonus points are marked by the icon. These problems usually include uncovered related topics, or those that are only mentioned briefly in the class.
- ***Moodle access is essential.*** Make sure you have access to Moodle, because that is where all assignments as well as course announcements are posted. Homework submissions are only made through Moodle.

- ***Assignment Deadline.*** Please submit your work **before the end of April 27th**.
- ***Delay policy.*** During the semester, students are given only 10 free late days which they can use them in their own ways. Afterwards, there will be a 15% penalty for every late day, and no more than four late days will be accepted.
- ***Collaboration policy.*** We encourage students to work together, share their findings, and utilize all the resources available. However you are not allowed to share codes/answers or use works from the past semesters. Violators will receive a zero for that particular problem.
- ***Any questions?*** If there is any question, please do not hesitate to contact us through the Telegram group chat or following email addresses: **m.ebadpour@aut.ac.ir**, **Peymanhashemi@aut.ac.ir**, and **atiyeh.moghadam@aut.ac.ir**.

2. Mathematics of Image Point Processing Operations (21 Pts.)

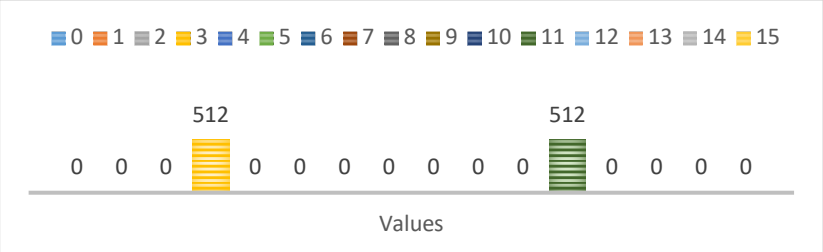
Keywords: *Image Point Processing, Image Histogram, Histogram Equalization, Histogram Matching (Specification), Image Negative, Bit-plane Slicing, Image Thresholding, Contrast Stretching*

The first task contains several problems in the topic of Point Processing. Point processing operations are an important group of Image Enhancement techniques in Spatial Domain, which focus on enhancing images solely with manipulating pixel intensity values. The problems here are aimed to make sure you've theoretically mastered those operations.

10	12	10	12	10	4	2	2
8	8	10	8	10	6	4	4
10	10	6	10	12	6	4	4
7	10	6	8	6	10	6	6
8	8	5	6	12	12	10	6
10	8	4	12	12	12	13	6
10	10	8	8	10	8	8	6
4	4	6	6	4	1	4	3

Consider the following 4-bit image of the size 8×8:

a. Find the digital negative of the image.
b. Perform bit-plane slicing and write down all bit-planes representations of the image.
c. Compute the image histogram.
d. Determine a "valley" in the histogram, and threshold the image at this value. Write down the resultant image matrix.
e. Apply linear scaling for stretching the gray levels of the image to the range of 0-255. Write down the result matrix.
f. Apply histogram equalization to the image. Write the resultant matrix as well as the equalized histogram.
g. g. Suppose you want to shape the histogram of the above image to match that of a second image, given below:



Use histogram specification to devise a gray-level transformation that shapes the histogram of the original image to look like the second. Note that you have to adopt a strategy to handle certain cases, including when two input values map to the same output value, and no input values map onto a particular output value.

Now, assume an image with the following gray-level histogram:

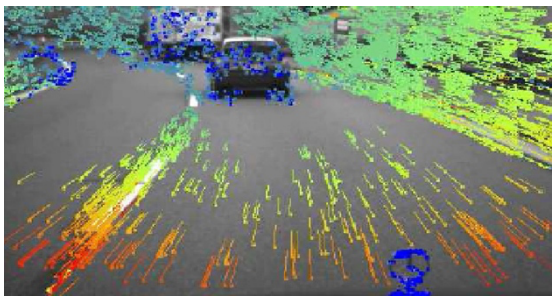
$$p_r = \frac{e^{\frac{r}{2}+1} - 3}{e - 1},\ r \in [-2, 0]$$

h. Find a gray-level transformation $s = T(r)$ to make the transformed histogram P_s uniform, i.e.: $P_s = 1,\ s \in [0,1]$

Note: Make sure to show all intermediate steps in the calculations.

2. Pixel Prowess: Tracking Traffic with Optical Finesse (26 Pts.)

Keywords: *Optical flow, gradient operator, Lucas-Kanade method, Horn-Schunck method, variational methods, Shi-Tomasi algorithm, sparse optical flow, dense optical flow, motion estimation, feature tracking, intensity gradients, object tracking, second-order derivatives..*



Optical flow plays a crucial role in self-driving cars by enabling them to understand and navigate the surrounding environment in real-time. In this context, optical flow refers to the perception of motion of objects relative to the car's position and movement, which is essential for tasks such as obstacle detection, collision avoidance, and path planning.

Optical flow is a fundamental concept in computer vision that refers to the pattern of apparent motion of objects in a scene caused by the relative motion between the observer and the scene. It is widely used in various applications such as motion estimation, object tracking, video compression, and autonomous navigation. In tracking applications, optical flow plays a crucial role in estimating the motion of objects by analyzing the changes in intensity patterns between consecutive frames of a video sequence. One simple yet effective operator used in optical flow computation is the gradient, which calculates the spatial intensity variations in an image. By computing the gradient, the direction and magnitude of motion can be estimated, forming the basis for optical flow algorithms.

Optical flow estimation involves determining the velocity field of the apparent motion of objects in a scene. Mathematically, optical flow can be represented by a vector field, where each vector corresponds to the motion of a point in the image plane. Let $I(x,y,t)$ denote the intensity of the image at pixel coordinates (x,y) and time (t). The optical flow field $V = (u,v)$ represents the horizontal and vertical components of motion, respectively. The optical flow equation can be formulated as:

$$\boldsymbol{I_x u + I_y v + I_t = 0}$$

Where I_x, I_y, and I_t are the spatial and temporal intensity gradients, respectively. This equation represents the constraint that the intensity remains constant along the trajectory of moving pixels over time (more justification detail in this link).

To solve the optical flow equation, various techniques such as Lucas-Kanade method, Horn-Schunck method, and variational methods are employed. These methods involve minimizing an objective function that measures the discrepancy between observed image intensities and predicted motion.

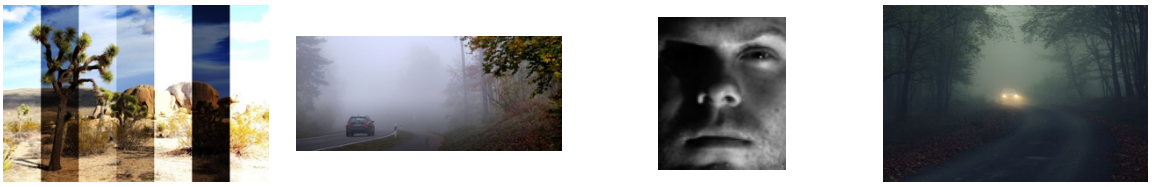
a. To track a scene, optical flow requires identifying features to track, and one naive approach is to assume the corners of objects as features. Research and report on the Shi-Tomasi algorithm and its fundamentals. Discuss its success and failure cases. Implement it as a function `*shi_tomasi(.)*` and discuss your implementation's parameters and returns.

b. Explain the difference between sparse optical flow and dense optical flow.
c. Load random 10-second frames from the given video captured by a traffic camera. Apply optical flow in both dense and sparse ways. Track each car with a random color line. Discuss the challenges faced and your efforts to handle them, such as appearing/disappearing new cars, considering dynamic reference images, selecting hyperparameters, equation solving method, and etc.
d. Assume you are asked to enhance optical flow by adding second-order derivatives as well as gradients. Present your strategy and discuss its pros and cons using mathematical notation and simplifications. Implement your strategy with simple test cases to indicate and compare results.

3. Pixels Perfected: A Journey through Image Enhancement (19 Pts.)

Keywords: *spatial domain, image enhancement, histogram equalization, adaptive filtering, image sharpening, image smoothing, gray-level transformation, linear, power/log, piecewise, median filtering, denoising techniques, Jupyter notebook, Jupyter widgets*

You encountered important spatial domain image enhancement approaches such as histogram equalization, adaptive filtering, image un/sharpening, image smoothing, gray-level transformations (linear, power/log, piecewise, etc.), median filtering, and other denoising techniques. In this project, you are tasked with utilizing your entire knowledge to enhance the attached images.



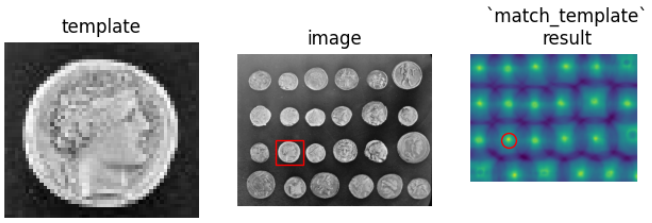
Examples of given images.

For each image, document the applied techniques in the order in which they were used, along with their hyperparameters and their effects. Generate a series of images for each image, depicting the progression towards the final enhanced image. To achieve this objective, implement a Jupyter notebook file with **Jupyter widgets**.

4. From Pixels to Patterns: Extracting Graphs in Image Analysis (29 Pts.)

Keywords: *template matching, convolution operation, edge detection, Hough Transform, graph analysis, node detection, image interpretation, pattern recognition, NetworkX library*

In certain security scenarios, processing and understanding images can be of paramount importance and even critical. For instance, consider a situation where a spy captures a screenshot of confidential documents containing graphical representations like schemas or graphs and sends them for analysis. In this project, the task is to develop algorithms to interpret such images, specifically focusing on graphs composed of nodes and edges.



Template matching is a valuable technique applied in various domains, including classic coin counting. In this application, a template image representing a coin of interest is matched against regions of an input image containing multiple coins.

a. Convolution operations on images are often utilized for template matching purposes. Start by researching and reporting on this technique. Subsequently, implement a function named `get_match_map(.)` that takes two input images - a reference image and a template image - and returns a heatmap indicating the probability of a match and the shape of the result. Given a reference image of shape n*m and a template image of shape p*q, determine the regular shape of the output. Plot the result for a given reference image and template, ensuring that built-in convolution functions are not used. **Hint**: Adjustments to the size of the template image may be necessary before applying it in template matching algorithms.
b. How can nodes be detected in the results obtained from template matching? Propose your approach and explain it. Then, implement the detection algorithm and assign unique identifiers from 1 to n to the detected nodes in image coordinates.
c. The Hough Transform is a powerful technique in image processing and computer vision, particularly useful for detecting shapes like lines or curves. In edge detection tasks, the Hough Transform can identify straight lines. Discuss the mathematical principles behind the Hough Transformation and implement it from scratch to detect edges between nodes. Utilize built-in gradient calculation functions, plot the Hough matrix, and define edges between nodes. Explain your strategy for assigning lines as edges.
d. Utilizing the NetworkX library, visualize the obtained graph.
e. Detecting the direction of edges involves additional considerations.

Note: Your code may be evaluated using different images.

Good Luck!

Mohsen Ebadpour, Atiyeh Moghadam, Peyman Hashemi