Amirkabir University of Technology
(Tehran Polytechnic)

Assignment 2
When Cameras Fail: Enhancing Images 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 .ipynb (or .py) file for part b. of question 3, which must be named 'p3b.ipynb' (or 'p3b.py').
- ***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 also made through Moodle.

- ***Assignment Deadline.*** Please submit your work **before the end of May 13th**.
- ***Delay policy.*** During the semester, students are given only 7 free late days which they can use them in their own ways. Afterwards, there will be a 20% 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 following email addresses: **ebp.mohsen@gmail.com** and **ali.the.special@gmail.com**.

1. Bringing Damaged Photos of Pre-Revolutionary Iran Back to Life (22 Pts.)

Keywords: *Image Enhancement, Bit-plane Slicing, Gray Coding*

The Islamic Revolution had a profound impact on Iranian society, and the stark contrast between the country before and after the revolution is evident in historical photographs. In this problem, we aim to examine some of these photos and utilize a technique based on **Bit-plane Slicing** to enhance their quality.

The main purpose of the algorithm, which is first introduced by E. Ardizzone *et al.* [1], is to make use of bit-plane slices to recover damaged areas of an image. Given a mask containing 1 in damaged positions and 0 elsewhere, this technique restores input images in three steps:

I. Bit-plane decomposition and gray coding. In the initial phase, the image is decomposed into its bit planes, before being gray-coded [2] to reduce the correlation between planes.

Figure 1 Several damaged photos from the streets of Tehran before the Islamic Revolution. Top: Damaged images. Bottom: Masks of the damaged regions

II. Information storage. Next, we aim to create a dictionary which contains information about the clean regions. To accomplish this, a square window W_n with window size n is defined surrounding each non-damaged pixel $b(x,y)$ in each bit plane. Then, An index is generated by arranging the bit sequence in the window in scan order. Likewise, a second index is constructed for the preceding significant bit plane, using a square window of size m, and added as a header to the aforementioned index:

$$k(x,y)=\left[\sum_{(x_j,y_j)\in W_m^{i+1}} b^{i+1}(x_j,y_j)\cdot 2^j\right]\cdot 2^{n\cdot n}+\sum_{(x_l,y_l)\in W_n^i} b^i(x_l,y_l)\cdot 2^l$$

The statistics of these sequences, specifically the conditional probability of the bit sequences in the i-plane given the corresponding sequence in the previous (i+1)-plane, are stored in the dictionary:

$$H(i,k)=P(W_n^i \mid W_m^{i+1})$$

III. Restoration. In the final step, the goal is to determine the likelihood of the central bit in the window being either 1 or 0, based on the neighboring bits in the current plane and the bits in the preceding plane. To calculate the statistics of each submask within a window, one can aggregate the statistics of all possible windows that share the same submask:

$$P(\hat{W}_n^i \mid W_m^{i+1}, b_c = 0) = \sum P(W_j^i \mid W_m^{i+1}, b_c = 0)$$

$$P(\hat{W}_n^i \mid W_m^{i+1}, b_c = 1) = \sum P(W_j^i \mid W_m^{i+1}, b_c = 1)$$

$$W_{j_1}^i \bigcap W_{j_2}^i = \hat{W}_n^i$$

We are seeking two statistics, which are:

$$S_0^i(x,y) = \sum_{p=0}^{2^{n_d}-1} H\left[i, B_p^i(x,y)\right] = P(\hat{W}_n^i \mid W_m^{i+1}, b_c = 0)$$

$$S_1^i(x,y) = \sum_{p=0}^{2^{n_d}-1} H\left[i, W_p^i(x,y)\right] = P(\hat{W}_n^i \mid W_m^{i+1}, b_c = 1)$$

Where $H[i,k]$ denotes the dictionary, n_d is the number of damaged bits in the window, B_p^i and W_p^i represent the sequences with a 'black' (0) and 'white' (1) central bit in the window, respectively, and both of them contain the bits from the $\hat{W}_n$ submask. To determine which information (0 or 1) to place in the center of the window, the algorithm compares a randomly generated number with the two statistics. These statistics are weighted by a user-defined parameter α:

$$r^i(x,y) = rand\left\lfloor 0, S_0^i(x,y) + S_1^i(x,y) \right\rfloor$$

$$S_1^i(x,y) > S_0^i(x,y) \rightarrow r^i(x,y) < \alpha \cdot S_1^i(x,y) \quad \begin{cases} b_c = 1 \\ b_c = 0 \end{cases}$$

$$S_0^i(x,y) > S_1^i(x,y) \rightarrow r^i(x,y) < \alpha \cdot S_0^i(x,y) \quad \begin{cases} b_c = 0 \\ b_c = 1 \end{cases}$$

If there are no statistics in the dictionary that match the actual sequence, the process of generating randomness operates on the basis of the following probabilities:

$$P_0 = P\left(b^i(x,y) = 0 \mid b^{i+1}(x,y)\right)$$

$$P_1 = P\left(b^i(x,y) = 1 \mid b^{i+1}(x,y)\right)$$

Once all of the planes have been restored, the bit planes are combined in order to reconstruct the complete image.

a. Utilize the algorithm to process the input images with the provided masks of damaged regions. Display the output results to visually represent the effectiveness of the algorithm.

Note: Use trial and error to determine the optimal values for the algorithm parameters.

b. Analyze the influence of the parameter α and the window size n and m by experimenting with different values for each. Evaluate how changes in these parameters affect the performance of the algorithm

2. Brightness Also Matters (20 Pts.)

Keywords: *Image Point Processing, Image Histogram, Histogram Equalization, Brightness Preserving Dynamic Histogram Equalization (BPDHE)*

Figure 2 Two images of the students of The University of Tehran in the 1970's, one grayscale and the other colored, with extreme contrast levels.

Prior to the Islamic Revolution in Iran, university students experienced a vastly different set of circumstances than they do today. The most notable difference was the degree of personal freedom that they enjoyed. This freedom was reflected in the way they dressed, expressed themselves, and interacted with one another. We are given two images of the students of The University of Tehran in the 1970's, and we are to enhance their contrast using histogram-based methods.

The conventional method of **Histogram Equalization** may not be a suitable choice for implementation in consumer electronics like televisions due to its tendency to produce unwanted visual degradation, such as the saturation effect. To address this issue, one potential solution is to maintain the mean brightness of the input image within the output image. Brightness preserving dynamic histogram equalization (BPDHE) [3] is an algorithm which attempts to accomplish this through the following steps:

I. Smooth the histogram with Gaussian filter. Detecting local maximums of a digital image's histogram can be difficult due to normal fluctuations in the histogram and missing probability for certain brightness levels. Therefore, smoothing the histogram is necessary. In this step, the missing brightness levels are first restored using linear interpolation. Next, a one-dimensional Gaussian filter, defined by the following equation, is used to smooth the histogram:

$$G(x) = \exp\left(-x^2/2\sigma^2\right)$$

where x and σ are the coordinate with respect to the center of the kernel and the standard deviation, respectively. The former is typically assumed to be a Gaussian filter of size 1×9, and the latter is set to 1.0762.

II. Detection of the location of local maximums from the smoothed histogram. Next, the signs of the first derivative of the smoothed histogram are computed. Since there may still be fluctuations in the signs, a method of eliminating stray signs is implemented. This involves inspecting three consecutive signs and changing +-+ to +++ and -+- to ---. Subsequently, the local maximums are

identified as points where four successive negative signs are followed by eight successive positive signs.

III. Map each partition into a new dynamic range. Assuming that m_0, m_1, ..., m_n are the $n+1$ gray levels representing the local maximums detected in the previous step, and that the original histogram prior to smoothing falls within the range of $[I_{min}, I_{max}]$, the algorithm divides the histogram into $n+1$ sub-histograms. The first sub-histogram ranges from I_{min} to m_0, the second sub-histogram ranges from m_0+1 to m_1, and so on, until the last sub-histogram ranges from m_n+1 to I_{max}. Before equalization occurs, the algorithm spans each sub-histogram using a function that depends on the total number of pixels in the sub-histogram:

$$span_i = high_i - low_i$$
$$factor_i = span_i \times \log_{10} M$$
$$range_i = (L-1) \times \frac{factor}{\sum_{k=1}^{n+1} factor_k}$$

where $high_i$ and low_i denote the highest and the lowest intensity values contained in the sub-histogram i, respectively, whereas M indicates the total number of pixels in that section, and $span_i$ and $range_i$ are the dynamic ranges used by sub-histogram i in the input and the output images, respectively. Assuming that the first sub-histogram in the output image is within the range of $[0, range_1]$, we can calculate the values of $start_i$ and end_i for $i > 1$ as follows:

$$start_i = \sum_{k=1}^{i-1} range_k + 1$$
$$end_i = \sum_{k=1}^{i} range_k$$

IV. Equalize each partition independently. In this part, the equalization of the sub-histogram i with the range $[start_i, end_i]$ is obtained through the following transformation:

$$y(x) = start_i + (end_i - start_i) \sum_{k=start_i}^{x} \frac{n_k}{M}$$

where n_k is the number of pixels with intensity value k.

V. Normalize the image brightness. Finally, the mean brightness of the input image, denoted by M_i, and the mean brightness of the output image after the equalization process, denoted by M_o, are computed. To bring back the mean brightness to the level of the input image, brightness normalization is performed using:

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

Here, $f(x,y)$ represents the output image after equalization and $g(x,y)$ represents the final output image.

a. Implement the previously mentioned algorithm and utilize it to enhance the input images by adjusting their contrast. Display the resulting images to showcase the effectiveness of the algorithm in improving the contrast of the input images.
b. Conduct a comparative analysis between the results obtained from the algorithm used for enhancing the contrast of input images and the traditional histogram equalization method. Evaluate and contrast the performance of both methods and provide insights on the strengths and limitations of each approach.

3. Iterative Version of the Mean Filter (20 Pts.)

Keywords: *: Image Spatial Filtering, Noise Reduction, Mean Filter, Iterative Mean Filter (IMF)*

Comparing photographs taken in Tehran 50 years ago to present-day images, it is striking to see how much the city has transformed. The urban landscape has undergone a significant change, with the addition of more modern buildings and high-rises, leading to a drastic increase in population density. Despite this increase, the streets in Tehran used to be cleaner, less crowded, and easier to navigate than they are now. In this problem, given a few of these photos, we seek to make use of **Spatial Filtering** techniques to remove their noise.

More precisely, we tend to implement a modified version of the mean filter, i.e., the iterative mean filter, in which a fixed-size window is utilized to calculate the average of gray values from noise-free pixels. The IMF algorithm [4] is as follows:

Input: Noisy image $B=\left[b_{ij}\right]_{m\times x}$

Output: Restored image $A=\left[a_{ij}\right]_{m\times x}$

1: $r\leftarrow 1,\ t\leftarrow 0,\ A^0\leftarrow B,\ \varepsilon$

2: $\delta_{\max}\leftarrow \max\limits_{\substack{1\le i\le m\\ 1\le j\le n}}\left\{b_{ij}\right\}, \delta_{\min}\leftarrow \min\limits_{\substack{1\le i\le m\\ 1\le j\le n}}\left\{b_{ij}\right\}$

3: **while** $\left|A^{t+1}-A^t\right|_1\le\varepsilon$

4: **for** each pixel $(i,\ j)$ of an image A^t at step t

5: **if** $\left(a_{ij}^t\ge\delta_{\max}\ \|\ a_{ij}^t\le\delta_{\min}\right)$

6: Define $W_{ij}^*\left(A^t,r\right)$

7: Update $a_{ij}^{t+1}\leftarrow\bar{W}_{ij}^{mean}\left(A^t,r\right)$

8: **else**

9: $a_{ij}^{t+1}\leftarrow a_{ij}^t$

10: **end if**

11: $t\leftarrow t+1$

12: **end for**

13: **end while**

The following definitions should be taken into consideration:

I. The indices set of a window. Assuming a window of size $(2r + 1) \times (2r + 1)$ centred at the pixel (i, j), the indices set of the pixels inside the window, shown by $W_{ij}(A, r)$, can be written as:

$$\left\{\left(i^*, j^*\right) \in I : \left|i^* - i\right| \le r, \left|j^* - j\right| \le r\right\}$$

where I denotes all pixel locations.

II. The strict indices set of a window. Assuming a window of size $(2r + 1) \times (2r + 1)$ centred at the pixel (i, j), the strict indices set of the pixels inside the window, shown by $W^*_{ij}(A, r)$, can be written as:

$$\left\{\left(i^*, j^*\right) \in W_{ij}(A,r) : a_{i^*j^*} \neq \delta_{\min}, a_{i^*j^*} \neq \delta_{\max}\right\}$$

III. Constrained mean of a window. Assuming $A = [a_{ij}]$ be an $m \times n$ image, the constrained mean of $W_{ij}(A, r)$, shown by $\bar{W}_{ij}^{mean}\left(A, r\right)$, can be written as:

$$\begin{cases} a_{ij} & W_{ij}^*(A,r) = \varnothing \\ \dfrac{1}{card\left(W_{ij}^*(A,r)\right)} \displaystyle\sum_{(i^*, j^*) \in W_{ij}^*(A,r)} & W_{ij}^*(A,r) \neq \varnothing \end{cases}$$

where *card*() returns the set of cardinality, which is the number of pixels.

a. Utilize the provided algorithm to eliminate the noise from the input images. Present the results through visual representation and calculate the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) by comparing them with the corresponding ground truth images.
b. Implement the classic mean and median filters, and compare their noise removal performance with the ones obtained by the IMF algorithm both quantitatively and qualitatively.

Figure 3 Four noisy images taken from different zones of the city of Tehran in the 1960's

4. Delving Into the Image Point Processing Operations (10 Pts.)

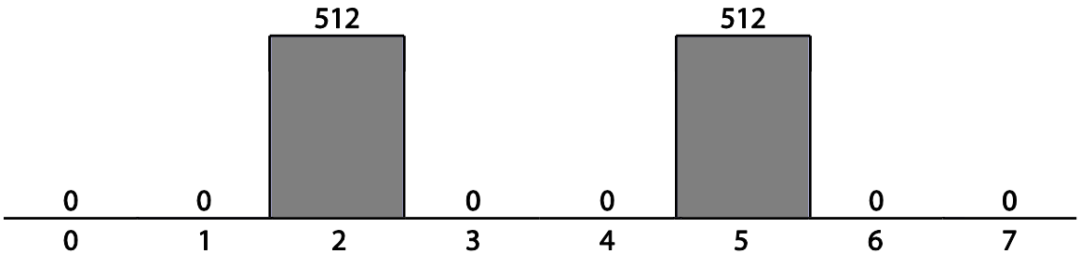
Keywords: *Image Point Processing, Image Histogram, Histogram Equalization, Histogram Matching (Specification), Image Negative, Bit-plane Slicing, Image Thresholding, Contrast Stretching*

The objective of this assignment is to test your theoretical understanding of **Point Processing** operations, which are a crucial set of **Image Enhancement** techniques in **Spatial Domain**. These techniques primarily involve the manipulation of pixel intensity values to enhance images. The assignment comprises multiple problems related to this topic.

First, consider the following 3-bit image of the size 8×8:

1	2	3	3	3	2	2	3
2	2	2	4	3	1	1	2
3	2	3	4	3	0	0	2
4	5	3	3	2	1	0	2
5	6	5	6	5	4	4	3
6	7	5	5	4	4	4	4
6	7	7	6	5	2	3	5
5	4	4	3	3	2	3	4

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. 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:

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

h. Find a gray-level transformation $s = T(r)$ to make the transformed histogram p_s uniform, i.e.:

$$p_s = 1 \quad s \in [0,1]$$

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

5. Practicing Image (De)Convolution Calculation (15 Pts.)

Keywords: *Image Spatial Filtering, Kernel (Mask), Image Convolution, Image Smoothing, Image Sharpening, Image Gradient*

Image Filtering is another way to enhance images in spatial domain. It is done through convolving a predefined **Kernel** over an image. So before jumping into the world of pixels and colors, it's good to practice some **2D Convolution** problems first.

0	7	7	0	0	3	3	3
0	7	0	0	0	3	7	3
0	7	3	3	3	3	3	3
0	7	0	0	0	7	0	0
0	0	7	0	3	3	3	0
3	0	7	0	3	3	3	0
7	0	7	0	3	3	3	3
3	3	0	0	0	7	0	7

First, a 3-bit image of the size 8×8 is given. Convolve each one of the given kernels (a) to (h) with this image and write the output image matrix. Also describe how each mask affects the input image.

-1	-1	-1
-1	8	-1
-1	-1	-1

(a)

-1	0	1
-1	0	1
-1	0	1

(b)

-1	-1	0
-1	0	1
0	1	1

(c)

0	-1	0
-1	4	-1
0	-1	0

(d)

1	1	1
1	1	1
1	1	1

(e)

-1	-1	-1
2	2	2
-1	-1	-1

(f)

0	-1	-1
1	0	-1
1	1	0

(g)

-1	2	-1
-1	2	-1
-1	2	-1

(h)

Now, let's do something more exciting. We're going to do the reverse process, i.e. **Image Deconvolution**. It's a technique widely used to restore the original image by estimating the convolution kernel. Here, you are asked to specify each one of the 3×3 kernel which was convolved with the original image to yield the outputs (i) to (p). Please clearly explain the strategy you used for each part.

Note: Rounded intensities are shown by blue, and out-of-bound values are shown by red.

2	2	2	1	1	2	2	2
2	3	4	2	2	3	3	2
2	3	3	1	2	3	3	2
2	3	3	2	2	3	2	1
1	3	2	2	2	3	2	1
1	3	2	3	2	3	2	1
2	3	2	2	2	3	3	2
1	2	1	1	1	2	3	1

(i)

7	7	7	7	3	6	7	6
7	7	7	0	3	7	7	7
7	7	7	7	7	7	7	6
7	7	7	0	7	7	7	0
0	7	7	7	6	7	6	3
3	7	7	7	6	7	6	3
7	7	7	7	6	7	7	6
6	6	3	0	7	7	7	7

(j)

0	5	2	0	0	2	3	2
0	7	3	1	1	3	4	3
0	7	1	1	1	4	3	2
0	5	3	1	2	4	2	1
1	2	5	0	2	4	2	0
3	0	7	0	3	3	3	1
4	1	5	0	2	4	2	3
3	1	2	0	1	3	1	3

(k)

0	7	7	0	0	7	2	7
0	7	0	0	0	2	7	2
0	7	5	7	7	0	2	7
0	7	0	0	0	7	0	0
0	0	7	0	7	0	7	0
7	0	7	0	6	3	6	0
7	0	7	0	7	0	6	5
5	7	0	0	0	7	0	7

(l)

7	0	0	7	6	0	0	0
7	0	7	7	7	0	0	0
7	0	0	0	0	0	2	0
7	0	7	7	7	0	7	7
7	7	0	7	0	0	0	6
0	7	0	7	0	0	0	7
0	7	0	7	0	0	2	0
0	0	7	7	7	0	7	0

(m)

1	4	3	1	1	2	3	2
2	4	3	1	1	3	4	3
2	4	3	2	2	3	3	2
1	3	3	1	2	3	2	1
1	2	3	2	2	3	2	1
2	3	4	3	2	3	2	1
3	3	3	2	2	3	3	2
2	2	1	1	1	3	3	2

(n)

0	0	0	0	0	0	0	0
0	4	5	0	0	0	0	0
0	0	0	0	0	0	7	7
7	7	2	2	3	0	3	6
1	4	0	0	0	2	0	0
0	0	0	0	0	0	0	0
0	1	7	7	2	0	0	0
7	7	7	7	7	7	7	7

(o)

2	0	0	0	1	0	0	0
4	0	4	3	2	0	0	0
4	0	1	0	0	0	0	0
2	0	3	4	2	0	2	2
2	3	0	3	0	0	0	1
0	5	0	5	0	0	0	2
0	3	0	3	0	0	1	0
0	1	0	3	1	0	2	0

(p)

6. Some Explanatory Questions (8 Pts.)

Please answer the following questions as clear as possible:

a. How will the result of applying a 3×3 mean filter to an arbitrary image change when the top left pixel of the kernel is taken into consideration instead of the central one?
b. Assess the feasibility and potential limitations of using image filtering to perform template matching on both binary and grayscale images.
c. When applying a 3x3 smoothing filter followed by a 3x3 sharpening filter to an image, are there any conditions under which the result would be equivalent to applying the sharpening filter first, followed by the smoothing filter?
d. How does setting the Least Significant Bit (LSB) plane or the Most Significant Bit (MSB) plane to zero affect the histogram of an image?
e. Explain why a discrete histogram equalization method does not always produce a perfectly flat histogram.

Good Luck!
Mohsen Ebadpour, Ali Abbasi
Peyman Hashemi, Alireza Sheikholeslam