Assignment 5

Practicing Basics of Image Compression and Image Morphology Techniques

Please remember:

1. What you must hand in includes the assignment report (.pdf), source codes (.m) and output files (.png). Please insert each part in a different folder, and zip them all together into an archive file named according to the following template: HW5_XXXXXXXX.zip
   Where XXXXXXXX must be replaced with your student ID.
2. 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 which proves you have realized the subject.
3. 5 points of each homework belongs to compactness, expressiveness and neatness of your codes and report.
4. By default, we assume you implement your codes in MATLAB. If you're using Python, you have to use equivalent functions when it is asked to use specific MATLAB functions.
5. Your codes must be separated for each question, and for each part. For example, you have to create a separate .m file for part b. of question 3. Please name it like p3b.m.
6. "Keywords" will help you find useful information about the problem. They may also include some ideas for solving that problem.
7. Using built-in functions is not allowed, except for simple operations like reading, displaying, converting and saving images, or in cases it is clearly mentioned in "Allowed MATLAB Functions" section of each problem.
8. **Please upload your work in Moodle, before the end of July 13th.**
9. If there is ***any*** question, please don't hesitate to contact me through the following email address: **ali.the.special@gmail.com**
   I'd be glad to help.
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 me your own work, without being worried about the grade! ;)

1. Getting to Know Some of the Simplified JPEG Compression Steps (9 Pts.)

Keywords: *JPEG Compression, Color Spaces, YCbCr Color Space, Discrete Cosine Transform, DCT Compression, Zig-Zag Ordering, Run-Length Coding, Entropy Coding, Peak Signal-to-Noise Ratio*

Joint Photographic Experts Group (JPEG) is a common method used for compression of digital images. The JPEG compression algorithm is at its best on photographs of realistic scenes containing smooth variations of color and tone. It is also very popular for web usage, where reducing the amount of data used for an image is vital for responsive representation.

In every variation of it, JPEG compression algorithm consists of several steps, which you will get familiar with some of them in this problem. You will also apply JPEG compression by yourself using MATLAB built-in functions..

a. **Color Space Transformation** is the first step of JPEG compression, where the image is converted from RGB space to $Y'C_BC_R$ space. Using the following equation, convert the image "trump_g7.jpg" into $Y'C_BC_R$ space, and display the results alongside each of three channels (Y', C_B and C_R) separately.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.403 \\ 1.000 & -0.344 & -0.714 \\ 1.000 & 1.773 & 0.000 \end{bmatrix} \cdot \begin{bmatrix} Y \\ C_b - 128 \\ C_r - 128 \end{bmatrix}$$

Figure 1 Image of Trump and other G7 leaders in RGB space, which must be converted to $Y'C_BC_R$ space

b. **Discrete Cosine Transform (DCT)** of the input image is an intermediate step in JPEG compression. DCT method for image compression is the most popular technique for image compression over the past several years. Its selection as the standard for JPEG compression method is one of the major reasons for its popularity. Using the instructions given in this page, apply image compression using discrete cosine transform on the image "trump_un.bmp". Set block size as 4, and discard 75% of the DCT coefficients in each block. Display the result, and compute the PSNR related to the original image. Also comment on the visual differences among the input image and the corresponding compressed image.

Figure 2 Input image of Trump and Un with swapped hair in .bmp format. The image must be compressed using DCT method

c. **Zig-Zag Ordering** is employed on DCT coefficients obtained from the previous step, so that similar frequencies group together. Read the image "ali.png" and display the corresponding zig-zag ordered vector. Include the resultant numeric vector in you report as well.

Figure 3 A Binary image of the size 8×8, used in part c.

d. **Run-length Encoding** is a simple method for lossless data compression, which is used in many applications including JPEG compression. In this method, the goal is to use runs of repeated bits and store them as a single data value and count. Implement a function which takes a bitstream and represent the equivalent run-length code in 4-bit counts. Apply it on the following bitstream, and display the coded bit sequence.

111111000011110000000011100000111111111111111111101111111110101110111100

e. **Entropy Coding** is also an important process in JPEG compression, where the **Huffman Coding** is applied on the intermediate result. Load the image "trump_bearded.jpg", and after building a probability model, compute the entropy of it.
Hint: Probability model is an alphabet with associated probabilities. This alphabet consists of symbols $\{0,1,\ldots,255\}$.

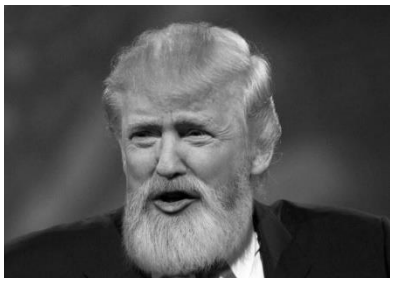

Figure 4 Bearded Trump, which must be used for entropy computation

f. Finally, use MATLAB function `imwrite()` to perform JPEG compression on the image "donald_kim_meeting.png". Set the quality level to 10, 25, 50, 75 and 100, and display the results. Also compare the obtained results using their PSNR values and comment on their visual appearances.

Allowed MATLAB functions: `im2double()`, `dctmtx()`, `blockproc()`, `imwrite()`

Figure 5 Image of Donald Trump and Kim Jong Un meeting in .png format, which must be converted to .jpg with different levels of quality

2. A Glance at Various Image Compression Techniques (10 Pts.)

Keywords: *Image Compression, Embedded Zerotree Wavelet, Set Partitioning In Hierarchical Tress, Spatial-Orientation Tree Wavelet, Wavelet Difference Reduction, Adaptively Scanned Wavelet Difference Reduction, Video Compression, Peak Signal-to-Noise Ratio*

The main goal in **Image Compression** techniques is to reduce image file size while preserving as much quality as possible. Due to the increasing requirements for transmission of images in computer and mobile environments, image compression plays a critical role in the field of digital image processing.

In this problem, you will get hands-on experience in several image compression methods.

a. Please give a brief description about how each of the following image compression methods work.
a1. Embedded Zerotree Wavelet (EZW)
a2. Set Partitioning In Hierarchical Tress (SPIHT)
a3. Spatial-Orientation Tree Wavelet (STW)
a4. Wavelet Difference Reduction (WDR)
a5. Adaptively Scanned Wavelet Difference Reduction (ASWDR)
Useful Link: [1]

b. Using MATLAB function `wcompress()` with appropriate settings, apply each of the above methods on the images in the folder "Trump Ties" locating in the folder P2 ("trump_ties_01.jpg" to "trump_ties_30.jpg"), and fill the following table.

Method	Average PSNR	Folder Size
None	∞	4.05 MB
EZW		
SPHIT		
STW		
WDR		
ASWDR		

Figure 6 A collection of images in "Trump Ties" folder, which must be compressed with various compression techniques

Note: Save the compressed images obtained from each method in a separate folder. You don't have to display or discuss the results.

c. Apply each of the above methods on the video "all_for_team_melli.mp4", and fill the following table.

Method	Video Size
None	18.4 MB
EZW	
SPHIT	
STW	
WDR	
ASWDR	

Figure 7 A frame extracted from the video "all_for_team_melli.mp4", consisting a total of 1788 frames. These frames must be extracted, compressed and finally assembled to create a compressed video file

Hint: Video files consist of frames, which means that all you have to do in this part is to extract them as image files, compress them using the mentioned methods, and finally attach the compressed frames together to create a new .mp4 video file.
Note 1: You must download the input video from a separate file in Moodle.
Note 2: Save each of the compressed video files in a separate folder. You don't need to include compressed frames.
Useful Links: [1], [2], [3]

Allowed MATLAB functions: `wcompress()`, `VideoReader()`, `VideoWriter()`, `audioread()`, `readFrame()`, `writeVideo()`, `read()`, `dir()`, `psnr()`

3. Basic Morphological Operations in Binary and Grayscale Images (12 Pts.)

Keywords: *Image Morphology, Image Logical Operations, Structuring Element, Image Dilation, Image Erosion, Image Opening, Image Closing*

Image Morphological Operations are a group of non-linear image processing operations related to the morphology or shape of features in an image. More precisely, these operations probe an image with a small template or shape called **Structuring Element**, which is positioned at all possible locations in the image and compared with the corresponding neighborhood of pixels.

The goal of this problem is to get you familiar with basic morphological operations, both in binary and grayscale images.

a. Implement a code to apply binary image dilation to the image "barcode_binary.jpg". Test it with an arbitrary structuring element, and display the result.
b. Implement a code to apply binary image erosion to the image "barcode_binary.jpg". Test it with an arbitrary structuring element, and display the result.
c. Implement a code to apply binary image opening to the image "barcode_binary.jpg". Test it with an arbitrary structuring element, and display the result.
d. Implement a code to apply binary image closing to the image "barcode_binary.jpg". Test it with an arbitrary structuring element, and display the result.

Figure 8 Binary image of a barcode, used as the input image in part a. to part d.

e. Implement a code to apply grayscale image dilation to the image "barcode_grayscale.jpg". Test it with an arbitrary structuring element, and display the result.
f. Implement a code to apply grayscale image erosion to the image "barcode_ grayscale.jpg". Test it with an arbitrary structuring element, and display the result.
g. Implement a code to apply grayscale image opening to the image "barcode_ grayscale.jpg". Test it with an arbitrary structuring element, and display the result.
h. Implement a code to apply grayscale image closing to the image "barcode_ grayscale.jpg". Test it with an arbitrary structuring element, and display the result.

Figure 9 Grayscale image of a barcode, used as the input image in part e. to part h.

Note 1: In all eight cases, you have to select a proper structuring element which clearly demonstrates the effect of that morphological operation.

Note 2: Please note that the usage of MATLAB built-in functions (like `imdilate()`, `imerode()`, etc.) is not allowed in this problem.

Allowed MATLAB functions: `im2double()`, `im2bw()`

4. Specifying Morphological Operation and Structural Element From the Result (9 Pts.)

Keywords: *Binary Image Morphology, Image Logical Operations, Structuring Element, Image Dilation, Image Erosion, Image Opening, Image Closing*

Selecting proper structuring element in **Image Morphology** is the key to obtain desired results. Needless to say, choosing correct **Morphological Operation** is also high of importance.

In this problem, you are given some images obtained from applying a specific morphological operation on an image, and your goal is to find out which morphological operation and structuring element have caused the result. For simplicity, consider only **Erosion**, **Dilation**, **Opening** and **Closing**, and assume a structuring element of the size 3 × 3.

The groundtruth image is "viva_team_melli.png".

a. Apply an appropriate morphological operation with a proper structuring element on the groundtruth image so that the image "viva_team_melli_a.png" is obtained.
b. Repeat part a. for the image "viva_team_melli_b.png".
c. Repeat part a. for the image "viva_team_melli_c.png".
d. Repeat part a. for the image "viva_team_melli_d.png".
e. Repeat part a. for the image "viva_team_melli_e.png".
f. Repeat part a. for the image "viva_team_melli_f.png".

Figure 10 Groundtruth binary image of this problem

Figure 11 From left to right, expected outputs of part a. to part f.

Allowed MATLAB functions: `imdilate()`, `imerode()`, `imopen()`, `imclose()`, `strel()`, `isequal()`

5. Practicing Some Basic Applications of Binary Image Morphology (16 Pts.)

Keywords: *Binary Image Morphology, Image Binarization, Image Logical Operations, Image Thresholding, Hit and Miss Transform, Opening Top-Hat, Boundary Detection, Region Filling, Removing Undesirable Details, Gap Filling, Character Counting, Text Enhancement, Defect Detection, Touching Objects Separation*

Although **Image Morphological Operations** in binary images may look futile at first glance, they are used in many simple applications of image processing and computer vision, especially as a preprocessing step. In this problem, you will get familiar with some of them.

a. **Boundary Detection** is the process of finding boundary pixels of the objects inside an image. Read the image "squares_and_circles.png", and count the boundary pixels inside it using morphological operations.

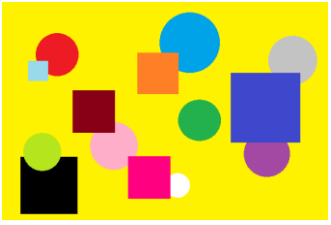

Figure 12 Input image for boundary detection

b. **Region Filling** is another morphological application, which deals with filling the regions in the image with some colors. When it comes to binary image, the goal is to fill areas in the image with opposite color. Read the image "russia_2018.png", and fill the empty area inside the cup.
Note: The cup must be monochrome (completely black) in your final result.

Figure 13 Binary image of FIFA World Cup 2018 logo. The goal is to fill the white areas inside the cup

c. **Removing Undesirable Details** using image morphology is applied when the goal is to remove undesired effects – usually appearing as dots or circles – around the main object in the image. Load the image "leo.jpg", and remove the unwanted details (areas around the head) in the image.

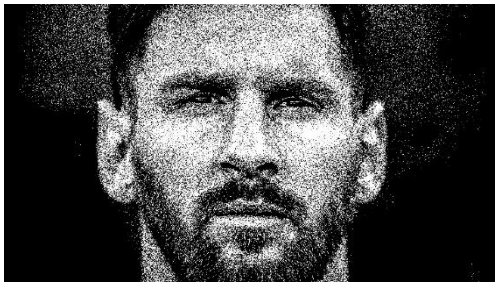

Figure 14 Binary image of Lionel Messi, with unwanted details appeared around his head

d. **Gap Filling** is a similar application to region filling, with only difference is filling 'all' gaps in an image (not a specific region). Read the image "iran_map.png". As you can see, the image consists of small dots with gaps between them. Fill the gaps using appropriate image morphological operations, and display the result.
Note: The map must be monochrome (completely black) in your final result.

Figure 15 Dotted map of Iran. The gaps inside the dots must be filled to obtain a seamless result

e. **Character Counting** is also possible using image morphology, by applying appropriate morphological operation as well as selecting a proper structuring element. Read the image "sleep_to_dream.png", and use the structuring element "letter_e.png" to find the number of repetitions of the letter 'e' inside the given text.

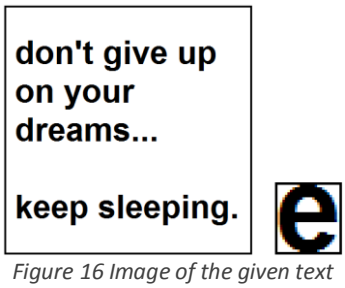

Figure 16 Image of the given text alongside the structuring element, which must be used to find the number of repetition of the letter 'e' inside the text

f. **Text Enhancement** techniques are applied to the texts which are not easily readable, mostly due to bad scan quality. Image morphology provides strong tools for handling this problem. Read the image "shahnameh.jpg", and apply appropriate morphological techniques to make it more readable and clear. Comment on the result and discuss which parts of the text have been more improved, and why.

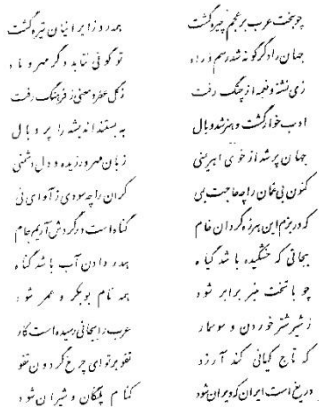

Figure 17 A piece of a scanned poem from Shahnameh. As can be seen, some words are hardly readable due to bad scan quality

g. **Defect Detection** is used when the goal is to find unusual effects in an image. By applying proper structuring element, morphological operations can also come in hand to do so. Read the image "apples.png", and find the logo of Apple co. inside the image (a bitten apple).

Figure 18 A bunch of forged Apple co. logos alongside a real one, which the goal is to find it by applying image morphological operations

h. **Separate Touching Objects** is useful as a preprocessing step in many image processing and computer vision applications, especially those which deals with detecting, counting or tracking objects. It is one of those areas where image morphology becomes useful. Read the image "pills.jpg", and use morphological operations to count the number of visible pills after separating them.

Figure 19 Touching circular pills, which must be separated using image morphology techniques

Note: In some cases, you first need to binarize the input image.

Hint: First specify the appropriate morphological operation(s), then design proper structuring element(s).

Allowed MATLAB functions: `rgb2gray()`, `im2double()`, `im2bw()`, `imdilate()`, `imerode()`, `imopen()`, `imclose()`, `bwhitmiss()`, `imtophat()`, `strel()`

6. Signature Recognition Feature Extraction using Morphological Operations (10 Pts.)

Keywords: *Signature Recognition, Binary Image Morphology, Image Dilation, Image Erosion, Image Opening, Image Closing, Morphological Skeleton, Morphological Thinning*

Offline Signature Recognition is the process of analysing the shape and form of a signature to identify its owner. **Offline Signature Verification** is also a similar area, where the goal is to verify whether a given signature is genuine or forged. Like every other classification problem, the phase of feature extraction plays a crucial role in the accuracy of the final result.

The goal of this problem is to apply morphological operations to extract useful features from signature images. You will work with two images "trump_signature.jpg" and "rouhani_signature.jpg" as the input images. Extract the following features for the given signatures, and examine how stable these features are among the same person's signature.

a. Skeleton
b. End points
c. Junction point
d. Upper and lower envelopes

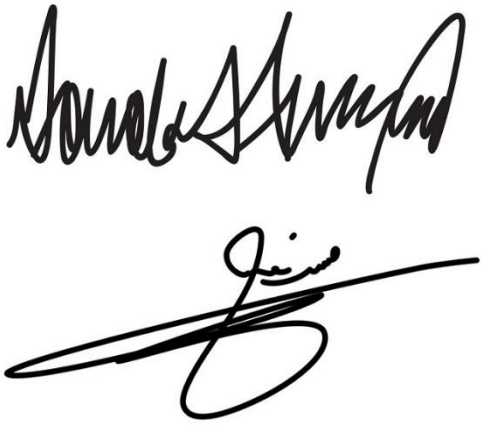

Figure 20 Donald Trump (top) and Hassan Rouhani (bottom) signatures, as the two input images of the problem

Hint: Here, stable features are those you can always find approximately in the same locations.

Allowed MATLAB functions: `im2double()`, `im2bw()`, `imdilate()`, `imerode()`, `imopen()`, `imclose()`, `strel()`

7. Applying Image Morphological Operations for Filtering Images (12 Pts.)

Keywords: *Grayscale Image Morphology, Image Morphological Filtering, Morphological Noise Reduction, Morphological Edge Detection*

A more complicated application of **Image Morphology** in grayscale images is **Image Filtering**, where the goal is to apply different types of filtering, e.g. smoothing filter, using a combination of **Morphological Operations**.

In this problem, your task is to examine some of these operations, and compare their results.

a. Load the image "queiroz.jpg" and apply morphological smoothing using dilation and erosion.
b. Repeat part a. with opening and closing. Compare the results obtained from part a. and b., and comment on their performance in **Noise Reduction**.

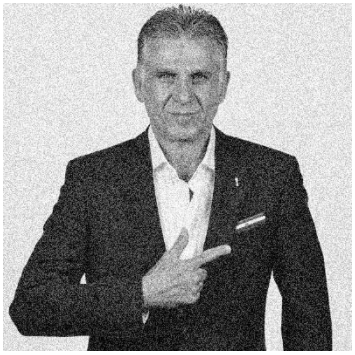

Figure 21 Noisy portrait of Carlos Queiroz, which is used for morphological smoothing

c. Load the image "jahanbakhsh.jpg" and apply morphological 2nd derivatives using dilation and erosion.
d. Repeat part c. with opening and closing.
e. Repeat part c. with dilation, erosion, opening and closing. Compare the results obtained from part c., d., and e., and comment on their performance in **Edge Detection**.

Figure 22 Image of celebrating Alireza Jahanbakhsh, which is used for applying morphological derivatives

Figure 23 Saman Ghoddos portrait, which is used for applying morphological gradient

f. Load the image "ghoddos.jpg" and apply morphological gradient using dilation and erosion.
g. Repeat part f. with opening and closing.
h. Repeat part f. with dilation, erosion, opening and closing. Compare the results obtained from part f., g., and h., and comment on their performance in **Edge Detection**.

Allowed MATLAB functions: `imdilate()`, `imerode()`, `imopen()`, `imclose()`, `strel()`

8. Playing with FIFA World Cup Match Balls (12 Pts.)

Keywords: *Image Morphological Operations, Image Binarization, Image Logical Operations, Image Thresholding, Hit and Miss Transform, Opening Top-Hat, Boundary Detection, Region Filling*

Object Detection and **Object Counting** are two other applications of **Image Morphology**, where the objects in an image are detected (and then counted) by applying appropriate morphological operations and structure elements.

In this problem, you are to practice object detection and counting using morphological operations. More specifically, you are provided with an image which consists of multiple circular objects from different categories, each with specific sizes. The information regarding each category of objects is also given in Figure 25.

Read the image "worldcup_balls.png" and considering the details provided in Figure 25, answer the following problems using what you have learnt from morphological operations.

a. Detect the biggest and smallest sized balls.
b. Count the total number of balls.
c. Count the number of Adidas-branded match balls.
d. How many World Cups took place with a match ball which was not produced by Adidas?
e. How many World Cups have there been in a European country?

Figure 24 A collection of World Cup balls with different sizes and colors. Note that the balls of the same category are also of the same size

Figure 25 Image of FIFA World Cup match balls over the history alongside their details, which will be used to answer some parts of this problem

Note: Please display the intermediate results, and include them in your report as well.

Allowed MATLAB functions: `rgb2gray()`, `im2double()`, `im2bw()`, `imdilate()`, `imerode()`, `imopen()`, `imclose()`, `bwhitmiss()`, `imtophat()`, `strel()`

9. Some Explanatory Questions (5 Pts.)

Please answer the following questions as clear as possible:

a. Why is it preferred to convert the image color space (RGB to $Y'C_BC_R$) in JPEG standard?
b. What difference does it make if we assume 0 instead of 1 for the origin of a structuring element?
c. Is it possible to apply morphological operations in color images? If the answer is yes, how? And if no, why?
d. Is there a way to shift pixels of an image using image morphology? If the answer is yes, how? And if no, why?
e. Is it possible to propose a method for applying morphological thinning and thickening in grayscale images? If the answer is yes, how? And if no, why?

Good Luck!
Ali Abbasi