Amirkabir University of Technology
(Tehran Polytechnic)

Assignment 1
Let's Play with Pixels!

Homeworks Guidelines and Policies

- ***What you must hand in.*** It is expected that the students submit an assignment report (HW1_[student_id].pdf) as well as required source codes (.m or .py) into an archive file (HW1_[student_id].zip).
- ***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 don't use implementation tools when it is asked to solve the problem by hand, otherwise you'll be penalized and lose some points.
- ***Don't bother typing!*** You are free to solve by-hand problems on a paper and include picture of them 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. Don't forget to explain what you have done, and provide enough discussions when it's needed.
- ***Appearance matters!*** In each homework, 5 points (out of a possible 100) belongs to compactness, expressiveness and neatness of your report and codes.
- ***Python is also allowable.*** 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.
- ***Be neat and tidy!*** 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.
- ***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's where all assignments as well as course announcements are posted on. Homework submissions are also done through Moodle.

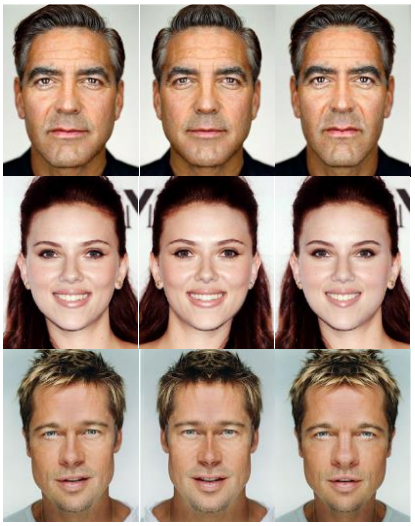
- ***Assignment Deadline.*** Please submit your work **before the end of April 16th**.
- ***Delay policy.*** During the semester, students are given 7 free late days which they can use them in their own ways. Afterwards there will be a 25% penalty for every late day, and no more than three late days 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 don't hesitate to contact me through **ali.the.special@gmail.com**.

1. Is Beauty Measurable? (15 Pts.)

Keywords: *Pixel Operations, Image Mirroring*

What makes a face pretty? Generally, there is no simple answer. Researchers have long been interested in discovering the features that make an individual attractive to others. One decisive facial attribute is said to be *facial symmetry*. It has been shown that, in mate selection, people are more willing to choose partners with more symmetrical faces.

But should good-looking people necessarily possess symmetrical faces? Does making someone's face symmetrical cause him/her to look better? And above all, is a person's beauty quantifiable? In this problem, we aim to answer these questions using basic image processing operations.



(a) (b) (c)

Figure 1 Faces of George Clooney (top), Scarlett Johansson (middle), and Brad Pitt (bottom). A well-established theory states that people with symmetrical faces generally appear more attractive than those with asymmetrical ones. (a) Original. (b) Left symmetry. (c) Right symmetry.

a. Write a function `img_mir = immirror(img)` which takes an input image `img` and returns its mirror version `img_mir`.
   Note: Your function must only use `for` loops, and using built-in functions as well as matrix operations is not allowed.
b. Divide each of the given face images into two 'L' (left half of the face) and 'R' (right half of the face) images from an appropriate point. For each of the images, obtain 'L ⅃', 'Я R', and 'Я ⅃' images using your `immirror()` function.
c. We now want to measure the similarity between the face of each individual with their corresponding symmetric faces obtained in the previous part. To do so, we consider two similarity metrics; *peak signal-to-noise ratio* (PSNR) and *structural similarity index measure* (SSIM). Based on the definitions described here and here, write down two functions `psnr_val = impsnr(img, ref)` and `ssim_val = imssim(img, ref)` which take a symmetric face image `img` and the original face image `ref`, and return the similarity indexes `psnr_val` and `ssim_val`. For each of the resultant face images obtained in part (b), calculate and report PSNR and SSIM. Based on these metrics, determine the individual who has the most symmetric face.

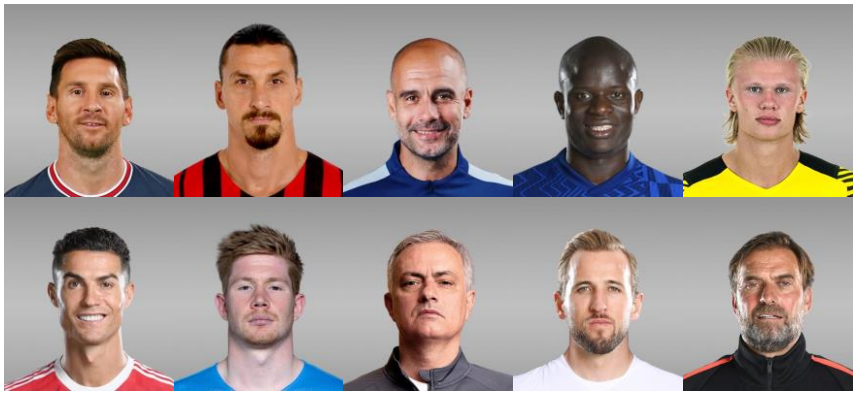


Figure 2 To investigate the effect of symmetry, portraits of some well-known football players and coaches are provided.

2. Steganography: The Art of Hiding Messages in Pixels (16 Pts.)

Keywords: *Pixel Operations, Image Steganography, Image Difference*

Have you ever wanted to send a secret message to someone without other people being able to discover it? Then you should have turned to **Steganography**, i.e., the art and science of concealing hidden messages. In **Image Steganography**, a message is embedded into an image by modifying the values of some pixels, which are chosen by a certain encryption strategy. While the message is hidden in the image, the pixels are changed so carefully that one can barely detect any change. Here, we aim to practice simple techniques of discovering and hiding secret messages in two distinct parts.

First, you are given two seemingly random grayscale images which contain a secret message. The idea is to move one image on another and find the hidden message using basic image operations.

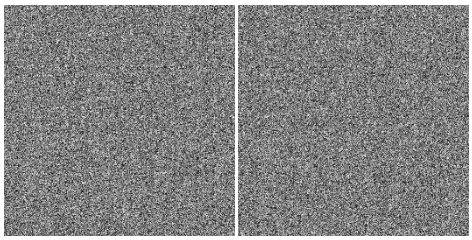


Figure 3 A secret message is concealed within these two apparently random images.

a. Slide one image onto another pixel by pixel from their horizontal boundaries, and calculate their difference. Continue until a meaningful image appears. Display the message.

In the next part, you are asked to hide the image of a secret message `msg` into a color image `img` given the following decoding pseudo-code:

```
1. img_r = img(:,:,1), img_b = img(:,:,3);
2. img_mid = abs(img_r - img_b);
3. for each pixel p in img_mid
       if img_mid(p) is even
          msg(p) = 1;
       else
          msg(p) = 0;
       end
   end
```

b. Devise a strategy to hide the given secret image into the color image, so that it could be decoded according to the above procedure.

   Note: You must first binarize and resize the secret message `msg`.

   Hint: You might need to make unnoticeable modifications to the intensity values of `img`.

Figure 4 The main color image (left) and the message (right). The message should be binarized before being embedded into the color image.

c. Decode the color image with the secret message obtained in the previous part using the given algorithm.

3. There Is No Planet B: Analyzing Some Global Warming Impacts (24 Pts.)

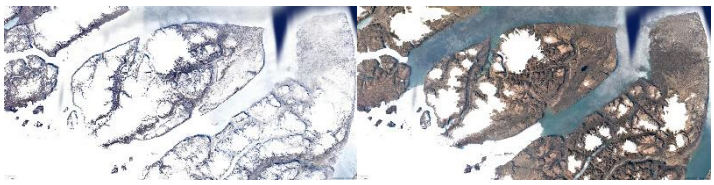
Keywords: *Basic Operations on Images, Color Space, RGB Space, Pixel Operations*

It is fairly unlikely that you haven't experienced the influences of the climate change in your area. Food and water scarcity, increased flooding, hotter temperatures, severe storms, more disease, and increased drought are only a few environmental changes associated with this phenomenon. In this problem, our goal is to make use of satellite images taken from four different regions in different dates in order to investigate the impacts of global warming in more detail.

(a) (b) (c) (d)

Figure 5 Four different locations which have been severely affected by the climate change. (a) Melting ice sheets in Greenland. (b) Dried-up Lake Urmia in Iran. (c) Shrinking Furtwangler Glacier in Tanzania. (d) Rising Siling Lake in China.

I. Mylius-Erichsen Land (Greenland). The greater part of Greenland's ice sheets is located on land and melts into the ocean. Over the past few decades, warming temperatures have accelerated the process of melting of these ice sheets.



2000 2020

Figure 6 The satellite images clearly demonstrate the speedy changes in the mass of Greenland's ice sheets.

a. Estimate the percentage of icy surface melted in the area during the given period.

II. Lake Urmia (Iran). Once the second largest saltwater lake in the middle east, Lake Urmia in Iran is now reduced to one-tenth of its size in the 1970s. Consecutive droughts caused by rising temperature in the region along with illegal wells and dams are partly to blame.

2000 2018 2020

Figure 7 Satellite images taken from Lake Urmia over a two-decade time span. The effects of the restoration program as well as mass flooding is visible in the image taken in 2020.

b. Assuming an average depth of 2.8 and 0.6 meters in 2000 and 2018, respectively, find the approximate volume of water evaporated between 2000 and 2018.
c. In 2020, it was reported that following the lake's restoration program and the increased rainfall, the volume of the lake had risen. What percentage of the lake was restored, considering an average depth of 0.9 meters in 2020?

III. Furtwangler Glacier (Tanzania). Close to the summit of Mount Kilimanjaro lies a rapidly receding glacier that was once part of an ice cap at the summit of Kilimanjaro. Approximately, over 85% of the glacier has disappeared over the past century.

d. Assuming an average height of 6.2 and 4.8 meters in 2003 and 2017, respectively, what proportion of the glacier has been melted over the given time interval?

2003 **2017**

Figure 8 Two satellite images captured from the summit of Mount Kilimanjaro, with Furtwangler Glacier located in the bottom-left corner of the images.

IV. Siling Lake (China). Runoff from melting glaciers in the Himalayas has contributed to the formation of new lakes and the expansion of others. Located in the Tibet Autonomous Region, Siling Lake has reportedly expanded by 40% due to glacial melt in the past half-century.

e. Assuming an average depth of 6.4 meters in 1985 and a gradual increase of 30 centimeters per every five years, plot a bar graph representing the approximate volume of the lake in each five years separately.

1985 **1990** **1995** **2000** **2005** **2010** **2015** **2020**

Figure 9 The expansion of Siling Lake throughout the past 35 years.

Hint: Scales are given in the bottom-right corner of the satellite images.

4. Creating A Pac-Man Demo: Let's Have Some Nostalgia! (30 Pts.)

Keywords: *Basic Operations on Images, Color Space, Pixel Operations, Frame, Video Sequence*

Today, it would be a shock to see a gamer who hasn't heard of Pac-Man. Released in 1980, the game involves a fictional character named Pac-Man who must be controlled by the player to eat all the dots inside an enclosed maze while avoiding four colored ghosts named Blinky (red), Pinky (pink), Inky (cyan), and Clyde (orange), which attempt to hunt him. While the original game file size was only 24kB, it went on to achieve several Guinness World Records and is currently ranked 12th in the list of best-selling video games of all time, ahead of some well-known titles such as *Call of Duty: Modern Warfare*, *Grand Theft Auto: San Andreas*, and *Red Dead Redemption 2*. To celebrate the 30th anniversary of the game's release, a playable version of the game was presented by Google on its homepage during the weekend of May 21-23, 2010, which was later said to be played by over one billion people worldwide (play it here).

Figure 10 Japanese flyer for the game. The game was originally named 'Puck-Man' in Japan.

In this problem, the goal is to create a demo version of Pac-Man, which, in spite of being non-playable, follows the rules of the game. You are given an image of the maze at its initial point as well as different elements of the game in various states.

a. Write down a function `game_img = init_game(maze_img,item_imgs)` which takes the image of the maze `maze_img` and a list `item_imgs` which contains the images of all elements, and generates an image `game_img` which is a state of the game in which characters are placed in random positions with random states. The position of Pac-Man must be on one of the dots in the maze with its mouth closed, whereas the directions of ghosts as well as their movement types should be determined randomly. Use your function once to create an initial state of the game.

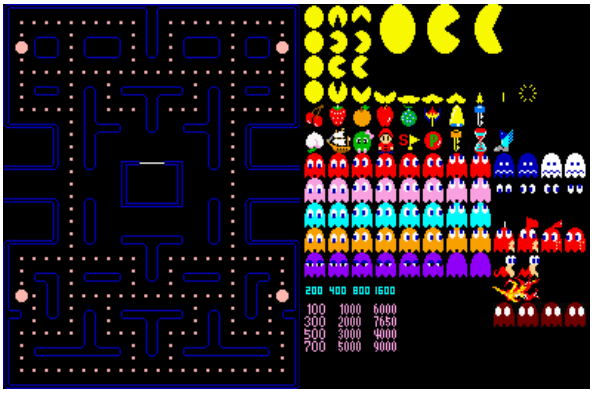


Figure 11 The maze (left) and the elements of the game (right) in all their possible states. Different characters take different gestures based on the events of the game.

b. Write down another function `next_game_img = gen_demo(game_img, item_imgs)` which takes a `game_img` together with `item_imgs` and produces an image `next_game_img` which depicts the next state of the game. The function must first detect the states of the characters (their directions and gestures) and then update the game state according to the following rules:

	Rules of the game
1	In each state, characters should move 3 pixels according to their current direction.
2	If the characters encounter walls, their direction should be altered. If there are two options, one should randomly be selected.
3	Pac-Man mouth opens when eating dots, and closes afterwards. It takes 12 pixels (four frames) for Pac-Man to reach a dot from the previous one, therefore, in each 3 pixels, its mouth should change from “closed” to “half-open” to “open” to “half-open” to “closed”.
4	Ghosts’ legs states change each 6 pixels.
5	Ghosts’ eyes change according to their direction.
6	Dots should disappear after Pac-Man reaches them.

Use `gen_demo()` to generate screenshots from the game. Continue until 25 dots are eaten. Include the initial state (the output of `init_game()`), the final state, and 10 random middle states in your report.

Note: In states in which Pac-Man mouth is closed (and therefore his previous direction is not discernible), you can either select a random direction (not efficient) or store his previous direction in a separate variable.

c. Use the frames obtained above and create a video sequence of all the states.

Hint: You can use this video as a guideline for your implementation.

5. Some Explanatory Questions (10 Pts.)

Please answer the following questions as clear as possible:

a. Describe some of the fundamental image processing steps which take place in our retina.
b. Samsung Galaxy S22 has a 50MP (f/1.8) rear camera, whereas iPhone 13 is equipped with a 12MP (f/1.6) one. Compare the images captured by these two phones in terms of resolution, storage, and quality.
c. As you know, the conversion of a color image into a grayscale one is possible using a simple mapping formula. Explain mathematically why a straightforward equation cannot be given for the reverse process.
d. Explain the concepts of re-sampling and sub-sampling in the context of image processing.
e. Imagine you wish to compute the similarity between two color images. Do different color spaces make any difference in the result? If no, explain the reason. If yes, specify which one of the following color spaces are more suitable, and why: Grayscale, RGB, CMYK, YUV, HSV, HSL.

Good Luck!
Ali Abbasi