Assignment 1

Pixel Magic: Mastering Some Basic Image Processing Techniques!

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). 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 6th**.
- ***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.

1. Change Detection as the result of spectral imaging (27 Pts.)

Keywords: *Change detection, AVIRIS sensor, RGB composite, Spectral analysis, Remote sensing, Environmental shifts, Hyperspectral imaging*

Change detection in spectral images is a crucial task with numerous applications across various fields including agriculture, environmental monitoring, urban planning, disaster management, and defense. It involves comparing images captured at different times to identify and analyze changes that have occurred in the scene. This process enables us to gain valuable insights into dynamic phenomena and understand temporal trends, which is essential for decision-making and resource management. The provided dataset comprising multi-temporal hyperspectral images offers a rich resource for conducting change detection research and applications.

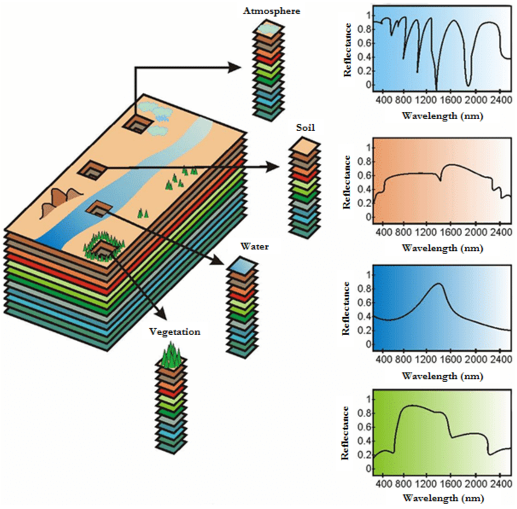

Figure 1. The concept of hyperspectral imaging is illustrated

This dataset facilitates the application of change detection techniques in multi-temporal hyperspectral images, offering insight into environmental shifts over time. It comprises two distinct hyperspectral scenes captured by the AVIRIS sensor: the Santa Barbara scene, acquired in 2013 and 2014, covering the Santa Barbara region in California, with spatial dimensions of 984 x 740 pixels and encompassing 224 spectral bands; and the Bay Area scene, captured in 2013 and 2015, around the city of Patterson, California, featuring spatial dimensions of 600 x 500 pixels and also containing 224 spectral bands.

(a) Research on the mentioned sensor, its properties, spectral bands, and imaging details.

(b) Drawing from course knowledge, explore methods for converting hyperspectral images to normal RGB images based on band selection and combination. One method commonly encountered involves utilizing specific spectral bands to simulate the visible spectrum. Implement a method based on mentioned key where involves selecting relevant bands and mapping them to the red, green, and blue channels to generate an RGB composite image.

(c) Leveraging understanding of imaging mechanisms from coursework, suggest a method to detect changes in water and vegetation (green) within each scene is fundamental. Propose a method to extract two single-channel images for each scene, each depicting the levels of green and water, facilitates change detection analysis. This method involves identifying spectral bands associated with vegetation and water absorption features and computing indices or thresholds to delineate these features. Implement this approach enables the calculation of change ratios and the visualization of change areas.

Note: Detailed explanation of the methodology and its implementation, including band selection rationale and index computation, is crucial for understanding the approach's efficacy for the AVIRIS sensor.

2. Automated RGB Channel Identification and Channel registration (17 Pts.)

Keywords: *gray-scale images, statistical analysis, cropping, channel identification, RGB channels, scene registration, error metric, channel registration, complexity reduction.*

Can you produce colored photography with only a black-and-white camera? As impossible as it sounds, Sergei Mikhailovich Prokudin-Gorskii (1863-1944), a man of genius, was able to produced thousands of "colored" pictures of the Russian empire by recording his photos in three different exposures: blue, green and red. With the technologies today, we can digitally combine the three exposures to gain a view of the old Russian empire. You have been provided with two images, each containing three grayscale single-channel images.

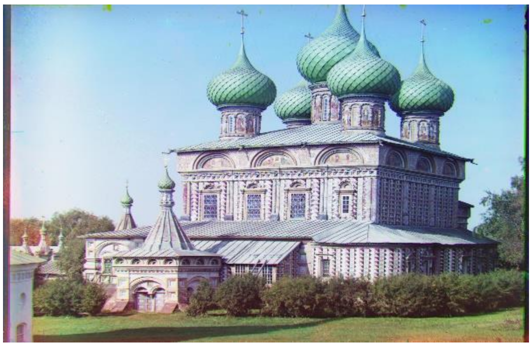

Figure 2. One RGB sample captured by Mikhailovich method.

(a) Propose a method based on statistical and other techniques to crop and extract each grayscale image. Explain your method, implement it, and report the results.

(b) Suggest a technique for identifying which channel corresponds to green, blue, and red. Can this process be automated? Implement your own technique and report the RGB channels.

(c) It's evident that the RGB channels exhibit noise, and scene objects are not accurately registered. Propose a method to address this challenge. You may select and define an error metric, and aim to reduce it for accurate channel registration.

(d) With mathematical and statistical justification, calculate the complexity of your registration algorithm, and attempt to reduce it using innovative methods.

3. Unlocking Color from Black and White: The Fascinating Illusion of Color (26 Pts.)

Keywords: *Color Assimilation, Grayscale Conversion,Visual Illusion, Color Perception, Optical Illusion, Visual Harmony, Illusion Creation.*

The utilization of a 'Color Assimilation Grid Illusion' to manipulate black and white images is a remarkable feat in visual perception. This ingenious technique has the power to deceive the mind and eyes into perceiving a grayscale image as vibrant and colorful. By strategically overlaying colorful lines onto the original black and white picture, the illusion is created, captivating viewers with its mesmerizing effect. This phenomenon not only showcases the intricacies of human perception but also demonstrates the creative potential of optical illusions in art and design.

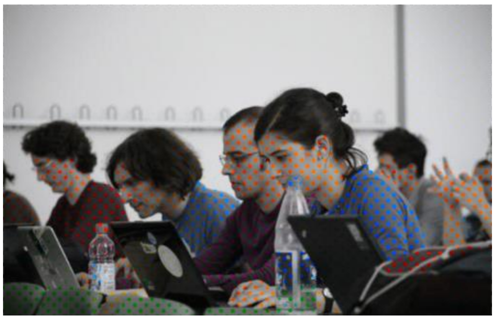

Figure 3. Color Assimilation Grid Illusion sample

(a) Explore a few eye illusions and associated risks in real life.
   i. Peripheral Drift Illusion
   ii. Motion Aftereffect
   iii. The Troxler Effect
   iv. The Hermann Grid Illusion
   v. The Pinna-Brelstaff Illusion
   vi. The Moiré Effect

(b) As you observed in Figure 3, adding certain structures or elements to a grayscale image can create the illusion of a colorful image. Now, you have been provided with two colorful images. Convert them to grayscale and attempt to replicate the Color Assimilation effect by adding circle patterns. Propose your own implementation method, apply it to the images, and then report the outcomes. Discuss the results and challenges such as circle distribution and harmony, shape, and color selection within the image.

(c) Repeat previous part for Rhombus shape. Compare the results.

(d) Now, attempt to convert the grayscale image, which employs color assimilation with circle elements, into a true RGB image. To accomplish this, feel free to utilize any technique you believe will be effective. One approach is to employ circle elements to propagate color to neighboring areas, known as Neighbor Color Propagation. Additionally, refinements may be necessary to preserve the shapes of objects and scenes. Introduce the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) metrics to quantify the error in the constructed RGB image.

4. Tetris game is coming! (25+15 Pts.)

Keywords: *Directional Movement, Image Generation, Scene Understanding, Object Transition, Video Generation, User Interface.*

Tetris is a classic and iconic puzzle video game that was created by Russian software engineer Alexey Pajitnov in 1984. It has become one of the most recognizable and widely played video games of all time, captivating audiences across generations and platforms.

The gameplay of Tetris is simple yet highly addictive. Players are presented with a vertical game board or "well" where different-shaped blocks, known as tetrominoes, descend from the top. The objective is to manipulate and rotate these falling tetrominoes to create horizontal lines without any gaps. When a line is completed, it disappears from the board, and any blocks above it will fall down to fill the space. The game continues with increasing speed as players strive to clear as many lines as possible and prevent the stack of blocks from reaching the top of the screen. Once the stack reaches the top, the game ends.

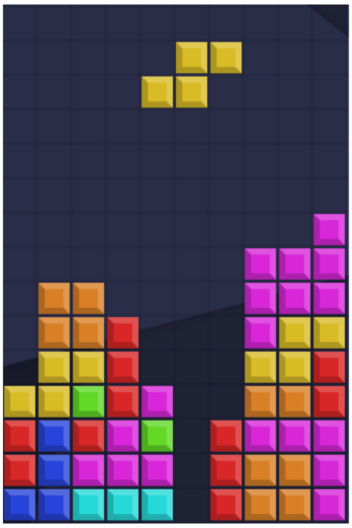

Figure 4. A frame of Tetris video game.

Tetris is a game featuring seven tetrominoes, each capable of rotating 90, 180, and 270 degrees, as depicted in Figure 5.

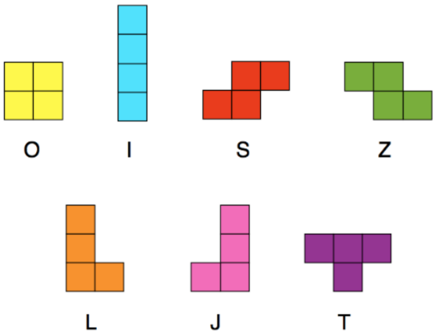

Figure 5. Seven tetrominoes in Tetris video game.

(a) Utilizing NumPy and OpenCV, endeavor to generate tetrominoes against a transparent background. These tetrominoes will be plotted in random colors and saved as PNG files.

(b) Develop a vertical grid board measuring 24 by 12 cells, with a background color of #b7d7f7 and bordered in #e6eaed.

(c) Generate a Tetris game in a randomized manner. At each step, a tetromino with a random color descends from a random position. Frames of the game are generated until the tetromino sticks to the board. Then, another tetromino appears. Implement the two key rules of the game:

i. A tetromino can hang suspended over a column's base, depending on its position.

ii. If a row is filled, it is removed, and all tetrominoes above fall to stick again.

Generate frames of the game, create a GIF, and document the gameplay. Additionally, incorporate two boxes into one of the possible positions, indicating the game level and score. The level box signifies the number of tetrominoes currently on the board (which may fluctuate), while the score box indicates the number of rows removed thus far (always increasing). Repeat the randomized game until at least one level increases. Only report on this implementation.

⭐ (d) Employing simple Matplotlib and/or PyQT libraries, you can develop a Tetris game interface where players can interactively control the movement and rotation of tetrominoes. Create a user-friendly graphical user interface (GUI) that allows players to manipulate tetrominoes by clicking on directional images. The GUI should display only three directional images corresponding to bottom, left, and right movements, as these are sufficient for gameplay. Implement this GUI to enable players to enjoy the Tetris experience with intuitive controls. (Hint: Link 1 and Link 2)

Important Note: Saving game status is not permitted. In each step, the previous image should be read to detect the game status, whether a tetromino is falling and should continue to fall for one frame, or a tetromino has stuck and another should appear, or a tetromino has stuck and filled a row requiring removal, etc.

Good Luck!

Mohsen Ebadpour, Atiyeh Moghadam, Peyman Hashemi