Middle East Technical University Department of Computer Engineering

CENG 466

Fundamentals of Image Processing

Fall '2024-2025

Take Home Exam 2

Due date: Dec 6, 2024, Friday, 17:00

1 Objectives

The purpose of this assignment is to familiarize you with the fundamental image enhancement techniques. You are required to develop your own algorithm based on the techniques you learned in the lectures.

2 Specifications

You should solve the given task with your own algorithms. In addition to the solutions, you are required to prepare a report that explains your methodology and includes the analysis of the results and your comments on them. The report should be **3-5 pages** long and should be prepared in IEEE Conferance Proceedings Template (LaTeXis recommended) provided in the following link.

`https://www.ieee.org/conferences_events/conferences/publishing/templates.html`

- Grading will be based on the quality of the outputs, script contents and the report
- The report should clearly explain the methodology and rationale behind the algorithm design. It should also explain the difficulties encountered in the design, implementation and experimentation stages, and your solutions on them. Last but not least, the report should contain your comments on the results. Even if the results does not match your expectations you should discuss the encountered situation.
- Note that with the permission of the students, the codes and outputs of the best paper will be published in our book, "Fundamentals of Image Processing with Python"

2.1 Question 1 - Pattern Extraction

In this part you will be extracting patterns from the given rug images. In order to complete this task follow the given steps:

1. You are given 2 images. (See Figure 1). Convert these images to grayscale
2. Apply edge detection filters on the images, use Sobel, Roberts and Prewitt filters.
3. Blur the images using three different kernel sizes.

(a) 1 (b) 2

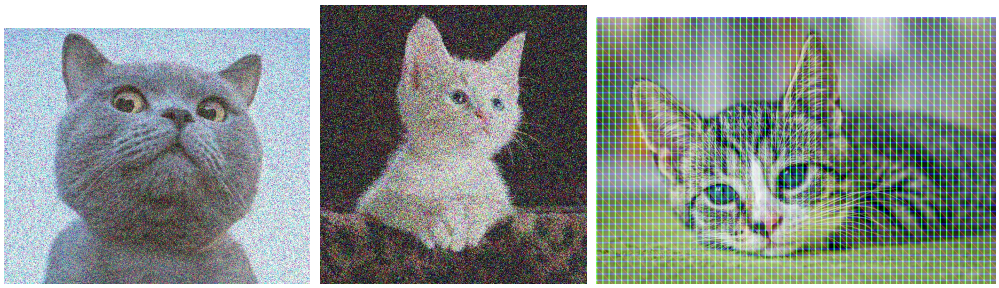
Figure 1

4. Apply Step 2 for each blurred image.
5. Binarize the gray scale images by extracting the most significant bit.
6. Apply Step 2 for MSB images.
7. Blur the MSB images using three different kernel sizes.
8. Apply Step 2 for blurred MSB images.
9. Save the output of each step.
10. Discuss your findings. Analyze the effects of the different edge detection filters, blurring, and binarization.

2.2 Question 2 - Image Enhancement

In this part you will be enhancing images in space and transform domains. In order to complete this task follow the given steps:

1. You are given 3 images. (See Figure 2. We have manually added different kinds of noises to Images. Examine these images in both spatial and Fourier domain. Can you identify the type of the noises? Note that RGB channels may have different degrees of noises or different types of noises. Therefore examine each channel independently.
2. Apply the following filters to images. Remember that your goal is to enhance the images. Therefore, carefully select the parameters such as kernel sizes or cut-off frequencies. These parameters can be image specific. You can use available libraries to complete this task. You may need different parameters per channel.
   - In spatial domain, apply
     - Gaussian Filter (Save output images as x_gaussian.png)
     - Median Filter (Save output images as x_median.png)
   - In Fourier domain, apply
     - Ideal Low pass Filter (Save output images as x_ilp.png)
     - Band-pass filter (Save output images as x_bp.png)
     - Band-reject filter (Save output images as x_br.png)
3. Discuss your findings from the previous step for each image:



(a) 1 (b) 2 (c) 3

Figure 2

- How did you select the parameters?
- What is the effect of each filter, does it depend on the image or the parameters?
- Can you remove the noises? If not, suggest a new filter to remove noises.

2.3 Question 3 - Image Compression

In this part you will apply compression techniques to three images in transform domains. In order to complete this task follow the given steps:

1. You are given 3 grayscale images. (See Figure 3. Apply Haar Wavelet Transform for a single level and Discrete Cosine Transform on all of the images.
2. For both transformations, retain the highest $N\%$ values and discard the remaining coefficients. Save the resulting compressed files for each transformation. Perform the compression using three different values of $N = \{1, 10, 50\}$ for comparison. You may use any appropriate format to store the compressed images.
3. Load the compressed files, reconstruct the images by setting the discarded coefficients to zero, and apply the inverse transformation to regenerate the images.
4. Compute the Mean Squared Error (MSE) between the compressed images and the original images to evaluate the information loss due to compression.
5. Analyze and discuss your findings for each image:
   - How much compression was achieved. You can compare the saved file sizes for each value of N and for both Wavelet and Cosine transforms.
   - What is the effect of using the Haar wavelet transform versus the Discrete Cosine Transform (DCT) on the compression performance?
   - Discuss the loss of information based on the MSE and the visual quality of the reconstructed images.

3 Regulations

1. **Group:** You are required to do your assignment in a group of two students. If there is an unclear part in your code, we may ask any of the group member to describe that code segment. Also group members may get **different** grades. We reserve the right to evaluate some or all of the groups to determine the contribution of each group member to the assignment.

(a) 1 (b) 2 (c) 3

Figure 3

2. **Programming Language:** You must code your program in Python. Your submission will be tested on department lab machines. You are expected make sure your code runs successfully on department lab machines.

3. **Late Submission:** Late Submission is **not** allowed!

4. **Newsgroup:** You must follow the odtuclass for discussions and possible updates on a daily basis.

4 Submission

Submission will be done via odtuclass. Create a tar.gz file named THEX.tar.gz that contains all your source code files and the report as a PDF file. Do not send the input and output images. Only one member should submit the homework. Hence, do not forget to **write your names and student id's at the beginning of the scripts.**

5 Cheating

We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations.

Cheating Policy: Students/Groups may discuss the concepts among themselves or with the instructor or the asistants. However, when it comes to doing the actual work, it must be done by the student/group alone. As soon as you start to write your solution or type it, you should work alone. In other words, if you are copying text directly from someone else - whether copying files or typing from someone else's nots or typing while they dictate - then you are cheating (committing plagiarism, to be more exact). This is true regardless of whether the source is a classmate, a former student, a website, a program listing found in the thrash, or whatever. Furthermore, plagiarism even on a small part of the program is cheating. Also, starting out with code that you did not write, and modifying it to look like your own is cheating. Aiding someone else's cheating also constitutes cheating. Leaving your program in plain sight or leaving your computer without logging out, thereby leaving your programs open to copying, may constitute cheating depending upon the circumstances. Consequently, you should always take care to prevent others from copying your programs, as it certainly leaves you open to accusations of cheating. We have automated tools to determine cheating. Both parties involved in cheating will be subject to disciplinary action. [Adapted from http://www.seas.upenn.edu/cis330/main.html]