ENGN6528 Computer Vision – 2021

Computer Lab-1 (CLab1)

Objectives:

The goal of this lab is to help you become familiar with, and practice Python-based or Matlab-based basic image processing operations, binary image analysis, image filtering, template tracking and mathematical morphology.

This is CLab-1 for ENGN6528. The objective is to help you become familiar with basic image I/O functions in either Matlab or Python. Note, in this lab, you are free to choose either of those two languages; Lab task descriptions are provided for both.

If you have not used Matlab or Python before, this lab is an opportunity to get you to quickly familiar with basic language usages and relevant libraries for image processing and computer vision. Please note that Python is now increasingly used in computer vision, and we encourage you to practise it in this course.

Special Notes:

1. Each Computer Lab task lasts for three weeks and has two lab sessions: session-A and session-B in the first weeks. Tutors/Lab instructor will provide basic supervision to both sessions. The third week has no lab, which is for you to complete and submit the lab report.
2. Before you start to work on writing lab report, we strongly recommend that you watch the video '**how to write a good lab report**' in Computer Labs section on Wattle. Keep in mind, Lab markers would not be interested in a pure collection of your experiment results. (Imagine you are demonstrating your lab works to others, you need appropriate interpretations for experimental results.)
3. The requirement of Lab submission is attached in the last pages. Please ensure your submission meets the requirement.
4. Your Lab will be marked based on the overall quality of your lab report. The report is to be uploaded to Wattle site before the due time, which is usually on the Sunday evening of the third week after the announcing of computer lab tasks. (e.g. Sunday of Week5 for Clab1). The penalty for late submission is 10% per day. Note, all students should consider problems of their own personal Internet access in advance.
5. It is normal if you cannot finish all the tasks within the two 2-hour sessions — these tasks are designed so that you will have to spend about 9 hours to finish all the tasks including finishing your lab report. This suggests that, before attending the second lab session (e.g. the Lab in Week4 for Clab1), you must make sure that you have almost completed 80%.
6. Please do note that we assume familiarity with Matlab or Python as background. If you need to revise these that is likely to take additional time.

Academic Integrity

You are expected to comply with the University Policy on Academic Integrity and Plagiarism. You are allowed to talk with / work with other students on lab and project assignments. You can share ideas but not code, you should submit your own work. Your course instructors reserve the right to determine an appropriate penalty based on the violation of academic integrity that occurs. Violations of the university policy can result in severe penalties.

C-Lab-1 Tasks

Task-1: Matlab (Python) Warm-up. (2 marks):

Describe (in words where appropriate) the result/function of each of the following commands of your preferred language in report. Please utilize the inbuilt *help()* command if you are unfamiliar with the these functions.

Note: Different from Matlab, Python users need to import external libraries by themselves. And we assume you already know some common package abbreviations (e.g. *numpy = np*). [You only need to complete one set of questions either in matlab or Python.] (0.2 marks each)

Matlab	**Python**
(1) a = [2, 3, 4; 5, 2, 200] ;	(1) a = np.array([[2, 3, 4],[5, 2, 200]])
(2) b = a(: ,2);	(2) b = a[: , 1]
(3) f = randn(400, 1) + 3;	(3) f = np.random.randn(400,1)+3
(4) g = f(find(f > 0)) *3;	(4) g = f[f > 0]*3
(5) x = zeros (1,100) + 0.45 ;	(5) x = np.zeros (100) + 0.45
(6) y = 0.5 .* ones(1,length(x));	(6) y = 0.5 * np.ones([1, len(x)])
(7) z = x + y;	(7) z = x + y
(8) a = [1:2: 500];	(8) a = np.linspace(1,499, 250, dtype=int)
(9) b = a([end: -2:1]);	(9) b = a[: :-2]
(10) b(b > 50)=0;	(10) b[b >50]=0

Hint:
Do the necessary typecasting (uint8 and double) when processing and displaying the image data in the following tasks. For Python, please be aware of the default datatype of different libraries (e.g. *Image*, *matplotlib*, *cv2*). An improper datatype of image will cause many troubles when you want to display the image.

Task-2: Basic Coding Practice (1 marks)

Write functions to process an input grayscale image with following requirements, where you need to write a script to load the given image in the Lab package, apply each transformation to the input, and display the results in a figure. For Matlab, you can use *subplot()* function; for Python, we suggest to use *matplotlib.pyplot.subplot()*. Please notice that each subplot needs to be labelled with an appropriate title. (0.2 marks each)

1. Load a grayscale image (such as Atowergray.jpg from wattle), and map the image to its negative image, in which the lightest values appear dark and vice versa. Display it side by side with its original version.
2. Flip the image horizontally (i.e, map pixels from right to left changed from left to right).
3. Load a colour image, swap the red and green colour channels of the input.
4. Average the input image with its horizontally flipped image (use typecasting).
5. Add a random value between [0,127] to every pixel in the grayscale image, then clip the new image to have a minimum value of 0 and a maximum value of 255.

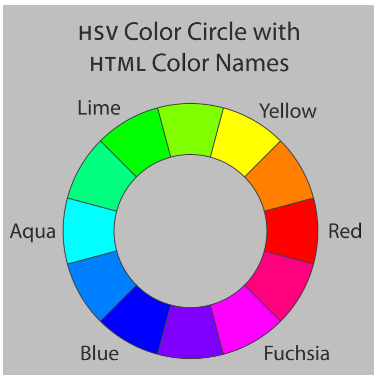
(Note: the intensity values of the original grayscale image range from 0 to 255.)

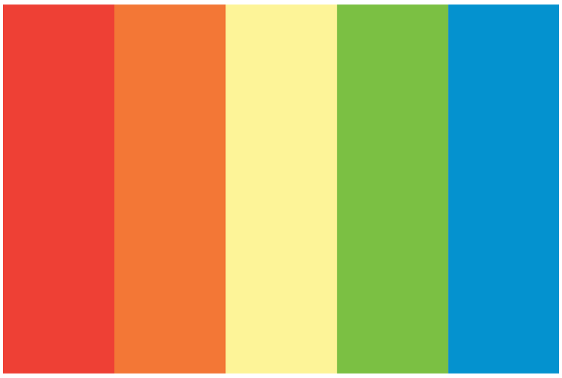
Task-3: Basic Image I/O (2 marks)

Note: You need to download the image1.jpg, image2.jpg, image3.jpg from wattle.

In this task, you are asked to:

1. Using image1.jpg, develop short computer code that does the following tasks:
   a. Read this image from its JPG file, and resize the image to 384 x 256 in columns x rows (0.2 marks).
   b. Convert the colour image into three grayscale channels, i.e., R,G,B images, and display each of the three channel grayscale images separately (0.2 marks for each channel, 0.6 marks in total).
   c. Compute the histograms for each of the grayscale images, and display the 3 histograms (0.2 marks for each histogram, 0.6 marks in total).
   d. Apply histogram equalisation to the resized image and its three grayscale channels, and then display the 4 histogram equalization images (0.15 marks for each histogram, 0.6 marks in total). (**Hint:** you can use inbuilt functions for implementing histogram equalisation. e.g. *histeq()* in Matlab or *cv2.equalizeHist()* in Python).





Fig, 2 (a) Colour Wheel (b) Palettes

Task-4: Colour space conversion (3 marks)

Use the two images in Fig.2 to study colour space conversion from RGB to YUV (you can download them from the Wattle site):

1. Based on the formulation of RGB-to-YUV conversion, write your own function cvRGB2YUV()that converts the RGB image to YUV colour space (0.7 marks). Read in Fig.2(a) and convert it with your function, and then display the Y, U, V channels in your report (0.1 marks for each channel, 0.3 marks in total).
2. Compute the average Y values of five colour regions in Fig. 2(b) with your function and the Matlab’s inbuilt function rgb2yuv(). Print both of them under the corresponding regions (0.1 marks for each value, 0.5 marks in total). You also need to explain how to distinguish and divide the five regions, and how to calculate the average Y value (1.5 marks, higher marks only for a smarter

solution).

(Note: The elements of both colormaps are in the range 0 to 1.).

Task-5: Image Denoising via a Gaussian and Bilateral Filter (9 marks)

In this task, you are asked to:

1. Read in image2.jpg. Crop a square image region corresponding to the central part of the image, resize it to 512×512, and save this square region to a new ***grayscale image***. Please display the two images. Make sure the pixel value range of this new image is within [0, 255]. Add Gaussian noise to this new 512x512 image (Review how you generate random number in Task-1). Use Gaussian noise with zero mean, and standard deviation of 15.

Hint: Make sure your input image range is within [0, 255]. Kindly, you may need *np.random.randn()* in Python. While Matlab provides a convenient function *imnoise()*. Please check the default setting of these inbuilt function.

Display the two histograms side by side, one before adding the noise and one after adding the noise (0.5 marks).

2. Implement your own Matlab/python function that performs a 5x5 Gaussian filtering (1.5 marks). Your function interface is:

   my_Gauss_filter()

   input: *noisy_image, my 5x5 gausskernel*

   output: *output_image*

3. Apply your Gaussian filter to the above noisy image, and display the smoothed images and visually check their noise-removal effects, investigating the effect of modifying the standard deviation of the Gaussian filter. You may need to test and compare different Gaussian kernels with different standard deviations (0.5 marks).

   Note: In doing this task, and the bilateral filter below you MUST NOT use any Matlab's (or Python's) inbuilt image filtering functions (e.g. *imfilter()*, *filter2()* in Matlab, or *cv2.filter2D()* in Python). In other words, you are required to code your own 2D filtering code, based on the original mathematical definition for 2D convolution. However, you are allowed to generate a 5x5 sized Gaussian kernel with inbuilt functions.

4. Compare your result with that by Matlab's inbuilt 5x5 Gaussian filter (e.g. *filter2(), imfilter()* in Matlab, or conveniently *cv2.GaussianBlur()* in Python,). Please show that the two results are nearly identical (0.5 marks).

Further reading material: http://setosa.io/ev/image-kernels/

Image Denoising via a Bilateral (Challenge task: this task will be difficult for most students to complete.) (5 marks)

1. Using your Gaussian filter as a base, implement your own Matlab/Python function that performs a 5x5 Bilateral filtering to gray-scale image. (1.5 marks)

   Your function interface must be:

my_Bilateral_filter()

input: *noisy_image, my_5x5_gausskernel, colour_sigma*

output: *output_image*

where colour_sigma is the sigma applied to the intensity/gray scale part of the filter.

2. Apply your Bilateral filter to the above noisy image (greyscale version) from the last task, and display the smoothed images and visually check their noise-removal and bilateral edge preserving effects (0.5 marks in total).

   In addition to the Gaussian filter, the range filter also has a standard deviation. You may need to test and compare different standard deviations for range (1.0 marks).

   Note: Do not use in-built functions, the same as for task 4.

3. Extend the Bilateral filter to colour images (eg. The color version of the previous grayscale image.). For this you may need to consider the CIE-Lab colour space as described in the paper. You will need to explore this for yourself. Namely, You need to generate the noisy color image and implement the bilateral filter to the color image (CIE-Lab). (2.0 marks) **(Note this task is more difficult again).**

   Tomasi, C; Manduchi, R (1998). Bilateral filtering for gray and color images (PDF). Sixth International Conference on Computer Vision. Bombay. pp. 839–846.

4. In up to half a page, discuss the impact of smoothing on colour. What are the difficulties of smoothing in RGB colour space? Why is CIE-Lab space a good idea for this smoothing? (Compare images smoothed in CIE-Lab space vs RGB) Does the bilateral filter itself help? You may want to compare Gaussian smoothed colour images and bilateral filtered ones, and investigate filtering in different colour spaces. For this you can use your processed images as examples, possibly cropping and enhancing detail in your report to illustrate your discussion. (1 marks)

Task-6: Image Translation (3 marks)

Choose an image among (image1.jpg, image2.jpg, image3.jpg) and resize it to 512 x 512,

1. Implement your own function *my_translation*() for image translation by any given number of pixels between [-100, +100], in both x and y. Note that this can be a real number (partial pixels). Display images translated by (2.0,4.0),(-4.0,-6.0),(2.5, 4.5), (-0.9,1.7), (92.0,-91.0) (0.20 for each image, 1.0 mark in total).

   Note: positive for away from upwards and right from the bottom left hand corner.

2. Compare forward and backward mapping and analyze their difference (1.0 mark).

3. Compare different interpolation methods and analyze their difference (1.0 mark).

   Hint: When analyzing the difference, you can focus on: (1) visual results; (2) the principles in terms of formulation or others relevant; (3) advantages and drawbacks; (4) the computational complexity. You can also think about it from other aspects.

Submission Quality

Note: Completing the experiment does not mean getting full marks, please leave enough time for documenting your Lab. Please note that deductions will apply for poor quality submissions.

- Successful submission, correctly submitting the whole lab package. (1 mark)
- Report quality, including clear interpretation, presentation and correct report format (e.g. table captions and reference). (2 marks)
- Code quality, including clear demonstration, comments. (1 mark)

==================== END of C-Lab-1 ====================

Clab-* Report

ENGN6528

name
UID Master

dd/mm/yyyy

Lab Report Requirement

1 Files

Upload a single ZIP file by the due date. You must use the following file name: CLab-1/2/3-Report-Uxxxxxx.zip, replacing Uxxxxx with your uni-ID.

Your ZIP file must contain the following contents:

(a) Three JPG photos of yourself (Only for CLab 1, requirements are in CLab1 Task 3).

(b) A PDF of your Lab Report. The report generally contains sample results from all the Lab Tasks, along with necessary comments and descriptions, questions and answers. For more detail. Please refer to the following Template and General Instructions for Lab Report on the next page.

(c) A file named "code" to include all your *.m or *.py files.

2 Lab Report

Kindly document different question under respective headings provided with the assignment. For example:

Task-1: The Question
1. Your first question under this theme

Documentation, observations, results, analysis etc.

2. Your second question under this theme

Documentation, observations, results, analysis etc.

3. Your third question under this theme

Documentation, observations, results, analysis etc.

2.1 General Formatting Instruction

Kindly use the same font single-spaced type for the entire document as much as possible, you may use the bold and italic version of the same font to highlight the important points. In the report, you need to use Times New Roman, which is quite widely used font to document projects and research papers.

- Kindly, use appropriate font size for sections heading and its contents accordingly. For example, 15 points Times, boldface type for heading and 12 points single-spaced type for the content is one of the widely used font sizes for documenting research papers.

- Please number all your sections and subsections of the tasks as provided in the assignment.

- Please show the images mentioned in each task to make your answer clearer.

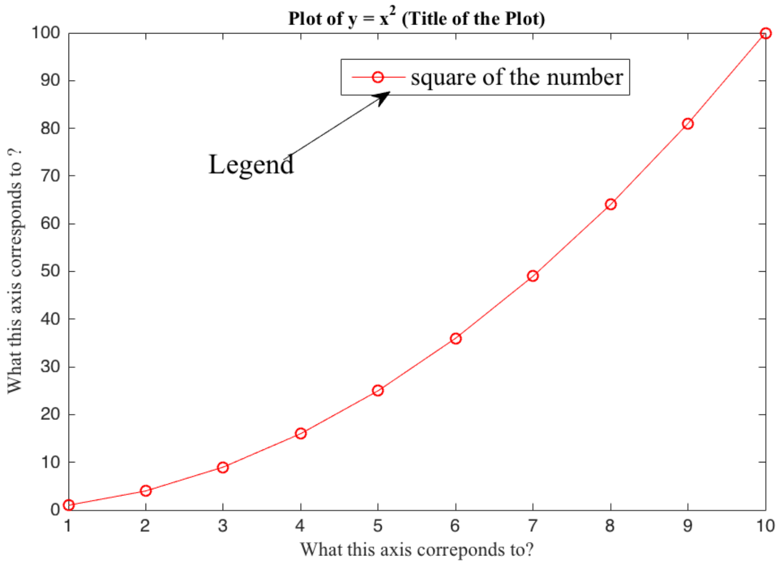
Brief explanations on how you solve the problem are expected.

- Please give your own answer following the question guidelines. ***Handwriting draft is not permitted***.

- Handwriting draft is not permitted for your submission.

2.2 Table, Figures and Plots

This is one of the important aspects of evaluating your report. Figures and the caption of the tables must be appropriately addressed. The figure should have an appropriate title if required. All the legends in the figure should be properly highlighted. The caption of the figure should explain your observation and understanding which may comprise of quantitative or qualitative evaluation to endorse your observation. Some of the widely used font to caption your figure, table and callouts are 10-11 points Roman type, 10-11 points Helvetica non-boldface type. Kindly, adjust the size of the figure in the document appropriately such that its clearly visible and perfectly eligible to illustrate your observation. We encourage you to look into the below example for reference. Note: You **cannot insist** we should zoom in or out to see tiny details on the graphs, plots, photographs, illustration, etc. Also, make sure the figures you include in your document is ***not a copyright image***.



Caption: Variation in the y-axis corresponding to the values in the x-axis and What does this mean, your observations?

For tables, graphs and others as well, kindly document the purpose of the statistical illustration which should include titles and proper labelling of the data and statistics.

Please follow the requirements to write your own Lab report.

================== END of C-Lab-1 report ==================