

DIGITAL IMAGE PROCESSING COURSE - 505060

PRACTICE LABS

LAB 01. IMAGE FORMATION & COLOR SPACES

Requirements

- (1) Follow the instructions with the help from your instructor.
- (2) Finish all the exercises in class and do the homework at home. You can update your solutions after class and re-submit all your work together with the homework.

- (3) Grading

Progress Evaluation 2 score = 50% * Attendance + 50% * Exercises

Rules:

- If the number of finished exercises is less than **80% total number of excercises**, you will get **zero** for the lab.
- Add the text of your Student ID to each of the output image.
- Submit the source code and output image files directly to Elearning assignment, **donot compress the files**.

- (4) Plagiarism check

If any 2 of the students have the same output images, then all will get zero for the corresponding exercises.

INTRODUCTION

In this Lab, you will learn how to

- Basic image manipulation
- Draw simple shapes
- Color manipulation

INSTRUCTIONS

Digital image

A digital image is an image composed of picture elements, also known as **pixels**. Typically, pixels are organized in an ordered rectangular array. The image width is the number of columns, and the image height is the number of rows in the array. Thus the pixel array is a matrix of M columns x N rows.

One more parameter, intensity, is needed to truly define an image. Each pixel has its own **intensity** value, or brightness. If all the pixels have the same value, the image will be a uniform shade; all black, white, gray, or some other shade. It is in the type of intensity used for each pixel that image types vary. Black and white images only have intensity from the darkest gray (black) to lightest gray (white). Color images, on the other hand, have intensity from the darkest and lightest of three different colors, Red, Green, and Blue.

Digital Image Processing

Digital Image Processing means processing digital image by means of a digital computer [1]. We can also say that it is a use of computer algorithms, in order to get enhanced image either to extract some useful information. Image processing mainly include the following steps:

1. Importing the image via image acquisition tools;
2. Analysing and manipulating the image;
3. Output in which result can be altered image or a report which is based on analysing that image.

OpenCV for Python

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. The library is used extensively in companies, research groups and by governmental bodies.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.

Install OpenCV for Python

- Install Python (version 3.6 or higher)
- Upgrade PIP (a package management system used to install and manage software packages/libraries written in Python) by typing the following command in the Command-Line Interface:
 - o `python -m pip install --upgrade pip`
- Install OpenCV by typing the following command in the Command-Line Interface:
 - o `python -m pip install opencv-python`

To check if OpenCV is correctly installed, just run the following commands to perform a version check (4.6.0 at this time):

```
python
>>>import cv2
>>>print(cv2.__version__)
```

Some reference websites for setup OpenCV for Python:

- o <https://www.geeksforgeeks.org/how-to-install-opencv-for-python-in-windows/>
- o <https://www.geeksforgeeks.org/how-to-install-opencv-for-python-in-linux/>
- o <https://www.geeksforgeeks.org/how-to-install-opencv-4-on-macos/>

Here is a simple program:

```
# Python code to read image
import cv2

# To read image from disk, we use
# cv2.imread function, in below method,
img = cv2.imread("geeksforgeeks.png", cv2.IMREAD_COLOR)

# Creating GUI window to display an image on screen
# first Parameter is windows title (should be in string format)
# Second Parameter is image array
cv2.imshow("Cute Kittens", img)

# To hold the window on screen, we use cv2.waitKey method
# Once it detected the close input, it will release the control
# To the next line
# First Parameter is for holding screen for specified milliseconds
# It should be positive integer. If 0 pass an parameter, then :
# hold the screen until user close it.
cv2.waitKey(0)

# It is for removing/deleting created GUI window from screen
# and memory
cv2.destroyAllWindows()
```

Note: search Google for OpenCV functions documentation

Example keywords: *opencv python imread (docs.opencv.org)*

EXERCISES

Getting Started

- 1) [Reading and display an image in OpenCV using Python](#)
- 2) [Writing an image in OpenCV using Python](#)
- 3) [Draw a text string](#)
- 4) [Draw a line](#)
- 5) [Draw arrow segment](#)
- 6) [Draw an ellipse](#)
- 7) [Draw a circle](#)
- 8) [Draw a rectangle](#)
- 9) [Color Spaces](#)

Code and run all the examples in the above instructions.

Additional exercises

Do exercises in pages 17, 18, 19, 20 of the “Theory1” file in Elearning website.

HOMEWORK

Finish all the exercises and re-submit to get higher scores.