**DIGITAL IMAGE PROCESSING COURSE - 2021.FALL   
PRACTICE LABS**

**LAB 01. IMAGE FORMATION & COLOR SPACES**

**Requirements**

1. Follow the instructions with the help from your instructor.
2. Finish all the exercises (given at the end of this document) in class and do the homework at home. You can update your solutions after class and re-submit all your work in the total submission together with the homework.
3. Grading  
   Total score = 50%\* In-class submission + 50% \* (Total submission) + Homework
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**

**OpenCV for Python**

OpenCV [OpenCV] is an open source (see http://opensource.org) computer vision

library available from http://opencv.org. In 1999 Gary Bradski [Bradski], working at

Intel Corporation, launched OpenCV with the hopes of accelerating computer vision

and artificial intelligence by providing a solid infrastructure for everyone working in

the field. The library is written in C and C++ and runs under Linux, Windows, and

Mac OS X. There is active development on interfaces for Python, Java, MATLAB, and

other languages, including porting the library to Android and iOS for mobile applica‐

tions.

One of OpenCV’s goals is to provide a simple-to-use computer vision infrastructure

that helps people build fairly sophisticated vision applications quickly. The OpenCV

library contains over 500 functions that span many areas in vision, including factory

product inspection, medical imaging, security, user interface, camera calibration,

stereo vision, and robotics. Because computer vision and machine learning often go

hand-in-hand, OpenCV also contains a full, general-purpose Machine Learning

library (ML module).

OpenCV was designed for computational efficiency and with a strong focus on real-time applications. It is written in optimized C++ and can take advantage of multicore

processors. If you desire further automatic optimization on Intel architectures [Intel],

you can buy Intel’s Integrated Performance Primitives (IPP) libraries [IPP], which

consist of low-level optimized routines in many different algorithmic areas. OpenCV

automatically uses the appropriate IPP library at runtime if that library is installed.

Starting with OpenCV 3.0, Intel granted the OpenCV team and OpenCV community

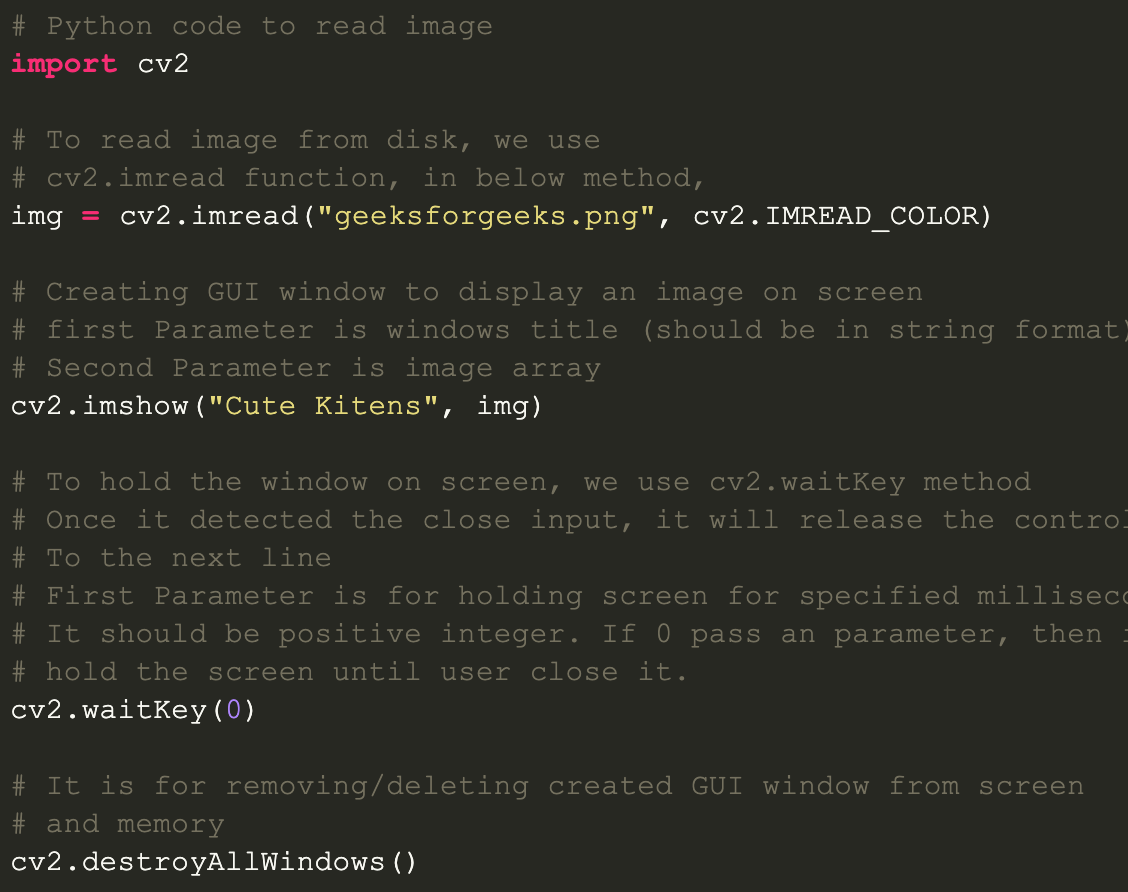
a free-of-charge subset of IPP (nicknamed IPPICV), which is built into and acceler‐

ates OpenCV by default.

**Install OpenCV**

* Windows 10: <https://www.youtube.com/watch?v=d3AT9EGp4iw>
* Linux: <https://www.youtube.com/watch?v=cGmGOi2kkJ4>
* MacOS: <https://www.youtube.com/watch?v=iluST-V757A>

Test your installation:  
CLI: >>> import cv2  
IDE: A simple program



Note: Google to search for OpenCV functions reference

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

## 

## **Working with Images**

**Getting Started**

1. [Reading an image in OpenCV using Python](https://www.geeksforgeeks.org/reading-image-opencv-using-python/)
2. [Display an image in OpenCV using Python](https://www.geeksforgeeks.org/python-opencv-cv2-imshow-method/)
3. [Writing an image in OpenCV using Python](https://www.geeksforgeeks.org/python-opencv-cv2-imwrite-method/)
4. [OpenCV | Saving an Image](https://www.geeksforgeeks.org/opencv-saving-an-image/)
5. [Color Spaces](https://www.geeksforgeeks.org/color-spaces-in-opencv-python/)
6. [Arithmetic operations on Images](https://www.geeksforgeeks.org/arithmetic-operations-on-images-using-opencv-set-1-addition-and-subtraction/)
7. [Bitwise Operations on Binary Images](https://www.geeksforgeeks.org/arithmetic-operations-on-images-using-opencv-set-2-bitwise-operations-on-binary-images/)

**Image Processing**

1. [Image Resizing](https://www.geeksforgeeks.org/image-resizing-using-opencv-python/)
2. [Create Border around Images](https://www.geeksforgeeks.org/python-opencv-cv2-copymakeborder-method/)
3. [Grayscaling of Images](https://www.geeksforgeeks.org/python-grayscaling-of-images-using-opencv/)
4. [Scaling, Rotating, Shifting and Edge Detection](https://www.geeksforgeeks.org/image-processing-in-python-scaling-rotating-shifting-and-edge-detection/)
5. [Convert an image from one color space to another](https://www.geeksforgeeks.org/python-opencv-cv2-cvtcolor-method/)
6. [Filter Color with OpenCV](https://www.geeksforgeeks.org/filter-color-with-opencv/)
7. [Visualizing image in different color spaces](https://www.geeksforgeeks.org/python-visualizing-image-in-different-color-spaces/)

**Drawing Functions**

1. [Draw a line](https://www.geeksforgeeks.org/python-opencv-cv2-line-method/)
2. [Draw arrow segment](https://www.geeksforgeeks.org/python-opencv-cv2-arrowedline-method/)
3. [Draw an ellipse](https://www.geeksforgeeks.org/python-opencv-cv2-ellipse-method/)
4. [Draw a circle](https://www.geeksforgeeks.org/python-opencv-cv2-circle-method/)
5. [Draw a rectangle](https://www.geeksforgeeks.org/python-opencv-cv2-rectangle-method/)
6. [Draw a text string](https://www.geeksforgeeks.org/python-opencv-cv2-puttext-method/)

**EXERCISE**

Code and run all the examples in the instructions.

Your task is to

* add the text of the exercise requirement to each of the output image and then,
* save the output image which is named in a format of “image\_XX.png” where XX is the exercise number.

Each correct output image will score 0.5 points.

**HOMEWORK**

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