**Department of Electrical Engineering**

**Faculty Member:** LE Munadi Sial **Date:** 27-Sep-2023

**Semester:** 7th **Group:**

# CS471 Machine Learning

**Lab 3: Introduction to OpenCV and PIL**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO4 - CLO4** | **PLO4 -CLO4** | **PLO5 -CLO5** | **PLO8 -CLO6** | **PLO9 -CLO7** |
| **Name** | **Reg. No** | **Viva /Quiz / Lab Performance** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics** | **Individual and Team Work** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Introduction

This laboratory exercise will introduce OpenCV which is a popular and widely used library for image processing and computer vision applications. The field of computer vision overlaps extensively with the field of Machine Learning particularly deep learning. A number of image-based applications such as face detection, image classification, object tracking and pose estimation rely on machine learning techniques. In such cases, images will be the dataset and so, image processing becomes a prerequisite for applying machine learning to images. It is important to familiarize with the basics of image processing which is the subject of this lab. You will also learn PIL which is another imaging library native to python.

## Objectives

* Load, save and display image data using Python
* Access and modify pixels as well as ROIs in images
* Place lines, rectangles, circles and text in images
* Resize and rotate images at various scales/angles

## Lab Conduct

* Respect faculty and peers through speech and actions
* The lab faculty will be available to assist the students. In case some aspect of the lab experiment is not understood, the students are advised to seek help from the faculty.
* In the tasks, there are commented lines such as #YOUR CODE STARTS HERE# where you have to provide the code. You must put the code between the #START and #END parts of these commented lines. Do NOT remove the commented lines.
* Use the tab key to provide the indentation in python.

**Theory**

**OpenCV** is a library that focuses on image processing and computer vision. An image is an array of colored square called pixels. Each pixel has a certain location in the array and color values in BGR format. By referring to the array indices, the individual pixels or a range of pixels can be accessed and modified. OpenCV provides many functions for resizing, rotating, and placing objects in images. Rotation involves computing a 2-D rotation matrix which is applied for the transformation of the image. **PIL** (Python Imaging Library) is another imaging library native to python. The extension of PIL for Python 3 is called “Pillow”. Unlike OpenCV which uses a BGR format for image data, PIL uses the RGB format.

A brief summary of the relevant keywords and functions in python is provided below:

**print()** output text on console

**input()** get input from user on console

**range()**  create a sequence of numbers

**len()** gives the number of characters in a string

**if** contains code that executes depending on a logical condition

**else** connects with **if** and **elif**, executes when conditions are not met

**elif** equivalent to **else if**

**while** loops code as long as a condition is true

**for** loops code through a sequence of items in an iterable object

**break** exit loop immediately

**continue** jump to the next iteration of the loop

**def** used to define a function

In this lab, you will be provided with a number of image files which you must use in the following given tasks. You may also have to download some images.

**Lab Task 1 – Load and Display Images \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a python script in which you will load 3 different images from disk. Then, display the images in different windows at the same time. You will need to provide the code and a single screenshot which shows all 3 windows at the same time.

***### TASK 1 CODE STARTS HERE ###***

*### TASK 1 CODE ENDS HERE ###*

***### TASK 1 OUTPUT SCREENSHOT STARTS HERE ###***

*### TASK 1 OUTPUT SCREENSHOT ENDS HERE ###*

**Lab Task 2 – Cropping Images \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write code to load the walle.jpg file. Using the slice operation, crop out the four quadrants of the image and display them in separate windows. The code must be generic enough to take the image size into account. For submission, provide the code and a single screenshot showing all 4 windows.

***### TASK 2 CODE STARTS HERE ###***

*### TASK 2 CODE ENDS HERE ###*

***### TASK 2 OUTPUT SCREENSHOT STARTS HERE ###***

*### TASK 2 OUTPUT SCREENSHOT ENDS HERE ###*

**Lab Task 3 – Modifying Pixel Data \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write code to load the walle.jpg file and place alternating green and white horizontal lines in the image. Do **NOT** use the line function (cv2.line). You need to this by changing the pixel colors. Each line must be 1-pixel thick. The lines are also spaced apart by 1-pixel wide gap. Thus, the image will have one green line, then one line of image pixels, then one white line, then another line of image pixels and so on. Provide the code and screenshot for the submission.

***### TASK 3 CODE STARTS HERE ###***

*### TASK 3 CODE ENDS HERE ###*

***### TASK 3 OUTPUT SCREENSHOT STARTS HERE ###***

*### TASK 3 OUTPUT SCREENSHOT ENDS HERE ###*

**Lab Task 4 – Placing Shapes and Text \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Load any one of the provided images and place a line, rectangle, circle and text using the inbuilt functions in OpenCV. Each of the placed object must have a different color. The text must contain the name of at least one member of your group. Provide the code and screenshot of the image.

***### TASK 4 CODE STARTS HERE ###***

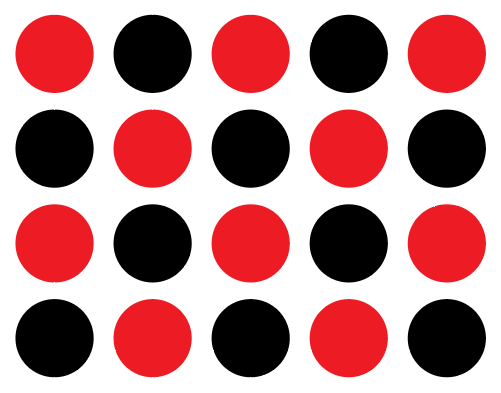
*### TASK 4 CODE ENDS HERE ###*

***### TASK 4 OUTPUT SCREENSHOT STARTS HERE ###***

*### TASK 4 OUTPUT SCREENSHOT ENDS HERE ###*

**Lab Task 5 – Placing Shapes II \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

In this task, you will place solid circles of alternating colors similar to those shown in the figure.



To make the circle solid, the thickness argument should be set to -1 in the cv2.circle function. The above shown pattern must be placed on one of the provided images. It is up to you to choose the radius of the circles as well as their colors. At least, 2 colors must be used. Provide the code and screenshot of the final result.

***### TASK 5 CODE STARTS HERE ###***

*### TASK 5 CODE ENDS HERE ###*

***### TASK 5 OUTPUT SCREENSHOT STARTS HERE ###***

*### TASK 5 OUTPUT SCREENSHOT ENDS HERE ###*

**Lab Task 6 – Bounding Boxes \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

For detection of faces, people, objects etc. in images, the result of the detection is depicted as a rectangular box around the detection. Load the robots.jpg image and use the cv2.rectangle function to place bounding boxes around the robots. Each bounding box must be of a different color. Provide the code and screenshot of the final result.

***### TASK 6 CODE STARTS HERE ###***

*### TASK 6 CODE ENDS HERE ###*

***### TASK 6 OUTPUT SCREENSHOT STARTS HERE ###***

*### TASK 6 OUTPUT SCREENSHOT ENDS HERE ###*

**Lab Task 6 – Resizing Images \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Load any one of the provided images and use the resize function to make copies of the image at different sizes. Display at least 3 different sizes in separate windows and take the screenshot. Provide the code and screenshot for the submission.

***### TASK 7 CODE STARTS HERE ###***

*### TASK 7 CODE ENDS HERE ###*

***### TASK 7 OUTPUT SCREENSHOT STARTS HERE ###***

*### TASK 7 OUTPUT SCREENSHOT ENDS HERE ###*

**Lab Task 8 – Rotating Images \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Load any one of the provided images and use the rotate function to rotate the image at angles of 30, 60 and 90 degrees. For each rotated image, you need to manually adjust the scale factor (in get2DRotationMatrix function) so that the entire image is shown in the window. The rotated image’s border/corner must touch the window’s border. Show all 3 windows in the screenshot. Provide the code and screenshot for the submission.

***### TASK 8 CODE STARTS HERE ###***

*### TASK 8 CODE ENDS HERE ###*

***### TASK 8 OUTPUT SCREENSHOT STARTS HERE ###***

*### TASK 8 OUTPUT SCREENSHOT ENDS HERE ###*

**Lab Task 9 – PIL \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Import the Python Imaging Library and use it to load any one of the provided image files. Stretch the image by changing the scale of the image and place different shapes and text in the image. Provide the code and screenshot of the final output. You will need to learn to use PIL on your own for this task.

***### TASK 9 CODE STARTS HERE ###***

*### TASK 9 CODE ENDS HERE ###*

***### TASK 9 OUTPUT SCREENSHOT STARTS HERE ###***

*### TASK 9 OUTPUT SCREENSHOT ENDS HERE ###*