# Laboratory Exercise 1
Introduction to OpenCV

Submitted by:

## Group # 7

| SO | PI | Category | Exceptional 4 | Acceptable 3 | Marginal 2 | Unacceptable 1 | Score |
|---|---|---|---|---|---|---|---|
| b | 1 | **Compliance** 30% | All procedures were followed, the output is as expected, additional related functionalities were augmented. | All procedures were followed and the output is as expected. | All procedures were followed but the output is not as expected. | Did not follow the set procedures | |
| b | 1 | **Analysis** 20% | Data interpretation is professionally written with appropriate and clear illustrations. | Data is clearly and correctly explained with proper illustrations. | Data is not clearly explained, has minor flaws, or no illustrations | There is no data explanation about the data or results. | |
| b | 1 | **Validity** 20% | The implementation uses the concepts and principles of the experiment as well as advanced topics. | The implementation uses the concepts and principles of the theory discussed for the experiment. | Implementation did not clearly express the use of theory discussed for the experiment. | There is no implementation. | |
| b | 1 | **Interpretation** 20% | The conclusion is professionally written and points the theories in the experiment and its implications in engineering. | The conclusion points to the main ideas and applications of the theory in the experiment. | The conclusion does not point out the main ideas and applications of the theory in the experiment. | There is no conclusion. | |
| | | **Format and Clarity** 10% | Follows the prescribed format, observes proper and technical grammar, and observes proper citation and referencing according to IEEE journal standards. | Follows the prescribed format, observes proper and technical grammar, and observes proper IEEE citation and referencing. | Did not follow the prescribed format, has poor grammar, or incorrect citations and references scheme. | Did not follow the prescribed format, has poor grammar, and has no citations and references. | |
| | | **TOTAL SCORE** | | | | | |

| Group Members | | | |
|---|---|---|---|
| **STUDENT NUMBER** | **NAME** | **CONTRIBUTION** | **SCORE** |
| 202113843 | Betchayda, Ezekiel | Programming/Coding, Results and Discussion | |
| 201911991 | Marquita, Bhea Marrianne | Programming/Coding, Results and Discussion | |
| 202111103 | Mejia, Zaldy Enrico Louis | Programming/Coding, Results and Discussion, Conclusion | |
| 202110017 | Sangco, Jerrold Cornelius | Programming/Coding, Results and Discussion | |

Submitted to:
Engr. Dexter James L. Cuaresma

Date:
02/12/2024

## OBJECTIVES

- To be familiarized with python and anaconda using PyCharm environment.
- To create a simple python program to interface cameras in PyCharm using OpenCV.

## DISCUSSION

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being an Apache 2 licensed product, OpenCV makes it easy for businesses to utilize and modify the code. (OpenCV, 2020)

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers. (OpenCV, 2020)

## MATERIALS

**Software:**

- PyCharm
- OpenCv
- Anaconda
- Python

## PROCEDURES

1. Install the following Software Applications (Latest Version)
    a. Pycharm
    b. Python
    c. Anaconda
2. Use python and Open Library to capture images and videos.

Python provides various libraries for image and video processing. One of them is OpenCV. OpenCV is a vast library that helps in providing various functions for image and video operations. With OpenCV, we can capture a video from the camera. It lets you create a video capture object which is helpful to capture videos through a webcam and then you may perform desired operations on that video.

OpenCV (Open Source Computer Vision) library. Following types of files are supported in OpenCV library:

- Windows bitmaps – *.bmp, *.dib
- JPEG files – *.jpeg, *.jpg
- Portable Network Graphics – *.png
- WebP – *.webp

- Sun rasters – *.sr, *.ras
- TIFF files – *.tiff, *.tif
- Raster and Vector geospatial data supported by GDAL

The steps to read and display an image in OpenCV are:

1. Read an image using imread() function.

2. Create a GUI window and display an image using imshow() function.

3. Use function waitkey(0) to hold the image window on the screen by the specified number of seconds, o means till the user closes it, it will hold GUI window on the screen.

4. Delete image window from the memory after displaying using destroyAllWindows() function.

Let's start reading an image. using cv2.

To read the images cv2.imread() method is used. This method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.

Example 1

```
# Python code to read image
import cv2
img = cv2.imread("image file name", cv2.IMREAD_COLOR)

# Creating GUI window to display an image on screen
cv2.imshow("image", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Example 2

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread("image file name ")
# Displaying image using plt.imshow() method
plt.imshow(img)

# hold the window
plt.waitforbuttonpress()
plt.close('all')
```

Example 3

```python
import cv2
# path
path = r'image filename'

# Using cv2.imread() method
# Using 0 to read image in grayscale mode
img = cv2.imread(path, cv2.IMREAD_GRAYSCALE)

# Displaying the image
cv2.imshow('image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Example 4

```python
import cv2

# define a video capture object
vid = cv2.VideoCapture(0)
while (True):

    ret, frame = vid.read()
    cv2.imshow('frame', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

vid.release()
cv2.destroyAllWindows()
```

3. Take a screenshot of your design and **describe each image**.
4. Save your works as Lab1_GroupNo. as a PDF file. And upload it to the submission file.

## RESULTS AND DISCUSSION

❖ OpenCV:

a. Example 1


*Figure 1. Python Reading Image Files using imread() function*

Observed in Figure 1, using OpenCV and assigning “img” as a variable to call the path file of the image. By using the “imread”, the program’s output can open the image as a new window.

b. Example 2

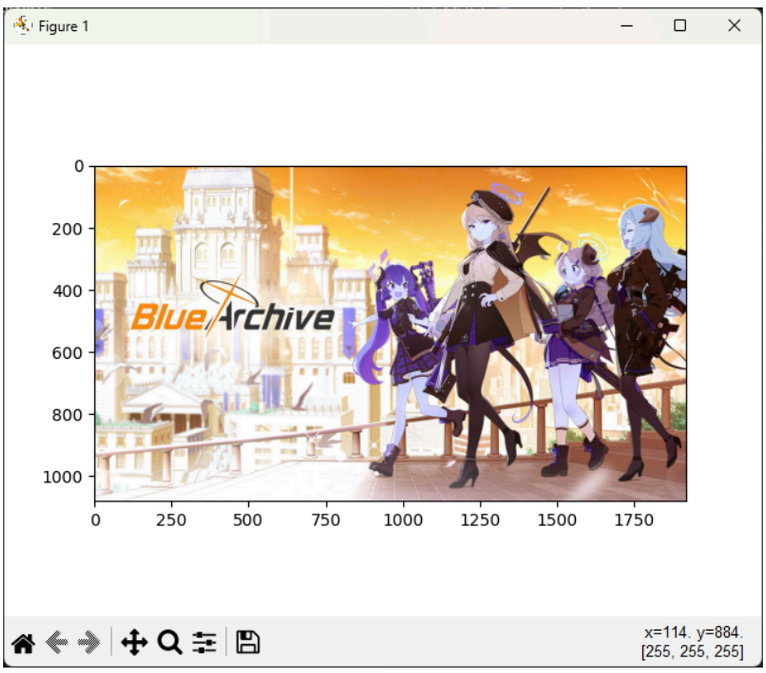


*Figure 2. Python Plotting the image using matplotlib*

Observed in Figure 2, the “imshow()” function from the matplotlib was used to display the image. matplotlib is a collection of functions that are commonly used in plotting data (examples: histograms, scatter plots, bar charts, etc.). To hold the

window, “waitforbuttonpress” was used. Its function is to wait for the user to press a key from the keyboard or mouse and then close the window afterwards.

c. Example 3


*Figure 3. Python Grayscaling an image using imread_grayscale*

Observed in Figure 3, same as program of Figure 1, but in this program instead of using the code “IMREAD_COLOR” to turn the picture black and white or grayscale the code that will be used in the code line of “img” will be IMREAD_GRAYSCALE”.

d. Example 4

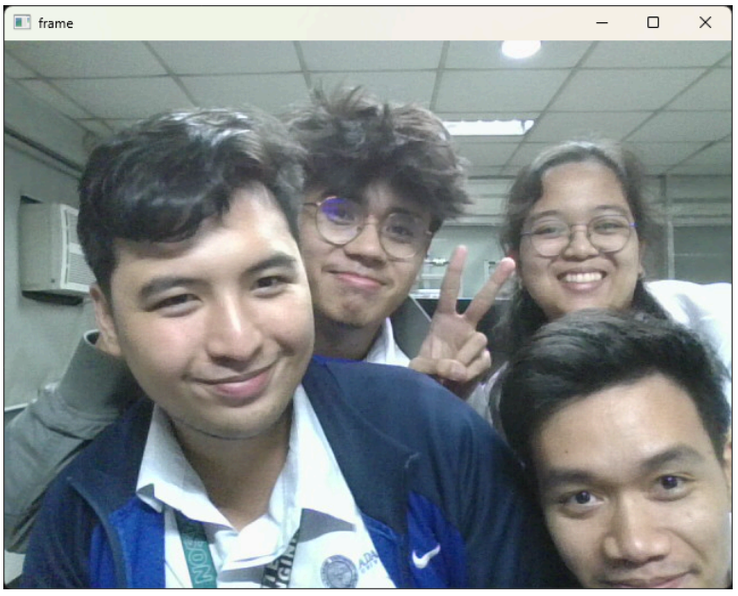


*Figure 4. Python VideoCapture*()

For example number 4, the given code example demonstrated how to open the camera using the videocapture() function. The 3rd line is initializing the variable x as a class from the opencv library which is the 'cv2.videocapture'. What this does is that it can either open a video file or image file sequence or a capturing device. A

while loop is then created that is set to true and inside that loop, two variables called ret and frame will be used to fetch frames from the video camera. To display the frame, OpenCV's class "Imshow()" is used.

Now in order to terminate the program, an if statement must be used. using the waitKey() function, it waits for a key to be pressed for a specified duration. Now '& 0xFF' is a bitwise AND operation that is for the ASCII value returned by the waitkey() function. It is used to extract the least significant 8 bits from the integer returned by the waitkey() function. Now " == ord('q') " is for comparing the ASCII value returned by the waitkey() function. To turn 'q' into an ASCII value, the ord() function is used. Combining everything together, the line of code basically means that it waits for an input of 'q' and if the user presses 'q', the break statement is executed and the program exits the loop, ending the video capture and closing the program through "x.release()" and "cv.destroyAllWindows()".

Github link of the repository for the experiment:
https://github.com/bheanne/MLPGrp7/tree/ab9b36260ae8ac5b4ee721798028117ba531c1db/LabAct1

## CONCLUSION

In conclusion, the learners were introduced to new libraries of python which include the OpenCV and Matplot package/library and the Anaconda environment. Through a series of code examples and demonstrations, the learners gained a comprehensive understanding of how OpenCV works for image processing tasks. Throughout the lab session, the basics of OpenCV was explored such as loading and opening an image, loading and opening an image alternatively using the matplot library, loading and opening an image as a grayscale output, and lastly, loading and opening the camera, all with the use of python scripts and the said libraries. Furthermore, the integration of the conda interpreter provided the learners an efficient environment for python as it supports the required libraries without having to install libraries relating to Data Science and Analytics. Overall, this lab provided a solid foundation for learners to explore image processing in python.