

## **WorkShop2**

### *Contents*

Instructions: .....	2
Part One: Description.....	3
Numpy help:.....	3
Part Two: Workshop Structure .....	3
Part Three: Read an Image and Extract Information .....	4
Part Four: Reducing the Number of Intensity Levels in an Image .....	5
Deliverables and Group Work.....	6

## WorkShop2

### Instructions:

- The workshop can be completed **in groups (maximum of four members per group (recommended))**.
- All group members should work together, and they will receive the same mark.
- This workshop is worth 2.5% of the total course grade, and it will be graded out of 25 marks and evaluated through your written submission, as well as the lab demo as follows:
  - 25 marks (2.5% of the total course grade)
    - 15 out of 25 (60%): Blackboard submission
    - 10 out of 25 (40%): Lab demo during the lab session
- Please submit the submission file(s) through Blackboard. **Only one person must submit for the group; only the last submission will be graded.**
- During the lab demo, group members are **randomly** selected to explain the submitted solution.
- **Group members who do not present during the lab demo will lose the demo mark.**

## WorkShop2

# Part One: Description

This problem set just ensures you can load an image, manipulate the values, produce some output, and submit the code along with the report.

Numpy Library (Necessary because OpenCV uses it in the background)

## Numpy help:

- a) Python Numpy Tutorial (with Jupyter and Colab): <https://cs231n.github.io/python-numpy-tutorial/>
- b) Learn NUMPY in 5 minutes - BEST Python Library! : <https://youtu.be/xECXZ3tyONo>
- c) UCSB Numpy Tutorial: <https://sites.engineering.ucsb.edu/~shell/che210d/numpy.pdf>
- d) Numpy Tutorial: A Simple Example-based Guide: <https://stackabuse.com/numpy-tutorial-a-simple-example-based-guide/>

# Part Two: Workshop Structure

Download the starter code “**ws2.zip**” for workshop2 from Blackboard.

The folder structure of **ws2** is as follows:

- **code/**: directory contains files named as **question<x>.py**. For example, **question1.py**, .... You must write your code in these python files.
- **data/**: directory contains the input images, videos or other data supplied with this workshop.
- **output/**: directory must be used to store your output from this workshop. Your output can be images, videos or other file type.
- **zip\_submission.py**: this python file will be used by you to create a zip file containing your code and output. To generate the submission once you are finished, use the following command: “**python zip\_submission.py**”.
  - Note: **run this command inside the ws2 directory**
  - You must submit this Zip file with your pdf report through Blackboard.
- **\*.py**: add any other supporting files you may need to complete your workshop to the **code** directory.

## WorkShop2

# Part Three: Read an Image and Extract Information

**Question 1.** Getting familiar with image manipulation in Python – OpenCV. Read the image "bicycle.bmp" from the **data** directory using the OpenCV flag **cv2.IMREAD\_UNCHANGED** and then find and print the following information about this image

- a) find image height (number of rows)
- b) find image width (number of columns)
- c) find the image number of channels.
- d) find image datatype.
- e) find the image number of pixels.
- f) convert the image to gray level and then save it in the **output** directory with name "bicyclegray.jpg"
- g) find the maximum value of the pixel values.
- h) Calculate the mean/average of the pixel values.
- i) Change the image's pixel values in the following way: all pixels' values less than the average value calculated at point (h) will be equal to 0 and all the other pixels will be equal to 1. Then, save it in the **output** directory with name "bicycleoutA.jpg"
- j) What type of image is generated at (i)?

## WorkShop2

# Part Four: Reducing the Number of Intensity Levels in an Image

### Question 2. Reducing the Number of Intensity Levels in an Image

Write a computer program capable of reducing the intensity levels in a gray image from 256 to 2 levels, in integer powers of 2. That is intensity levels: 256 defaults, 128, 64, 32, 16, 8, 4, 2. Use the image "[lena.tif](#)" from the [data](#) directory.

Note: Your code should generate the image with the required intensity and then write/save the image to the [output](#) directory. The name of generated images as following [lena256.jpg](#), [lena128.jpg](#), [lena64.jpg](#), [lena32.jpg](#), [lena16.png](#), [lena8.png](#), [lena4.png](#), and [lena2.png](#).

## WorkShop2

# Deliverables and Group Work

Create workshop report with the following name format

**group\_<number>\_ws\_<workshop number>\_report.pdf**

For example, if the **group16** created a report for **workshop20**, then the report name should be

**group\_16\_ws\_20\_report.pdf**

The workshop report should include:

(a) Complete this declaration by adding your names:

We, ----- (mention your names), declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. We have not copied any part of this assignment, manually or electronically, from any other source including web sites, unless specified as references. We have not distributed our work to other students.

(b) Specify what each member has done towards the completion of this work:

	Name	Task(s)
1		
2		
3		
4		

(c) The report shows all your answers and outputs for all the workshop questions. So, you include the output images under the question and write a response to answer some of the workshop questions.

(d) Submit two files.

a. **submission.zip**

b. **group\_<number>\_ws\_<workshop number>\_report.pdf**