WorkShop3

Contents

WorkShop3

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.**

WorkShop3

Part One: Description

This problem set is just to ensure you can create an image, draw shapes, and manipulate the pixel values. You will also produce some output images and submit them along with your code and the written report.

In many computer vision applications, ***we often wish to select only a portion of an image to analyze and ignore the rest of the image***. Thus, we need to create or perform an image cropping (create a rectangular sub-image) with **NumPy** slicing. Moreover, in some situations, we would like to create another special image of the same size as the original input image, with white pixels indicating the region to save (or to keep) and black pixels everywhere else. Such an image is called a **mask**. In preparing such a **mask**, we need to be able to draw some shapes (i.e., circles, rectangles, lines, and geometric shapes) on a black image.

One of the basic operations of OpenCV is the ability to draw over the image. The ability to add lines, circles, and geometric shapes to an image is an operation that will prove very useful in many computer vision applications. For example, in digital image analysis, you want to highlight an image's portion (or a region) by ***adding one or more rectangle bounding boxes surrounding the detected objects***. Furthermore, we would like to add an arrow to indicate or highlight something interesting in that image. Also, you would like to add a grid or draw a contour on your image in some situations.

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/

WorkShop3

Part Two: Workshop Structure

Download the starter code "ws3.zip" for workshop3 from Blackboard

The workshop folder/directory structure 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 types.
- **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 ws3 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.

WorkShop3

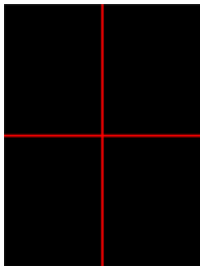
Part Three: Create an Image and Accessing Pixels

Question 1. Write a code in **question1.py** file to do the following tasks:

a) Create color image (full black image) with the size 480 x 640, and then display this image
   a. Note: color image means three channels

b) Save the black image in the **output** file with the name "**black.jpg**"

c) Draw two red lines on the black image with thickness 4.
   a. Use OpenCV **`cv2.line`** to draw the two lines.
   b. One line divides the width into two halves.
   c. One line divides the height into two halves.
   d. After drawing these two lines, the black image will be divided into four quarters.



d) Display the result image from **part (c)** and then save it in the **output** folder with the name "**twolines.jpg**"

e) **Randomly** pick one of the empty four quarters and draw a **white rectangle** with a height of **50** and a width of **75**.
   a. Use OpenCV **`cv2.rectangle`** to draw the rectangle.
   b. Make sure that the drawn rectangle is located within the picked quarter.
   c. Note: the upper left corner point **(x,y)** of the drawn rectangle can be selected **randomly** as long as it is located in the picked quarter and guarantees that the drawn rectangle will be completely drawn inside the picked quarter
   d. Hint: You can use the NumPy to find a random integer number (**np.random.randint**). You can read more about this at numpy.random.RandomState.randint
   e. Copy the output image here.

>>>>>Copy your output image here <<<<<

WorkShop3

f) Pick **one** of the remaining empty quarters **randomly**, and then draw a blue circle with a radius of 40.

a. Use OpenCV **`cv2.circle`** to draw the circle.
b. Make sure that the drawn circle is located within the picked quarter.
c. Note: The drawn circle's center point **(x,y)** can be selected **randomly** as long as it is located in the picked quarter, which guarantees that the drawn circle will be completely drawn inside the picked quarter. The NumPy package can generate random numbers (see the above link).
d. Copy the output image here.

>>>>>Copy your output image here <<<<<

g) Pick one of the empty remaining quarters randomly, and then draw a yellow **pentagon** with any dimensions as long as it has **five vertices** and a size length greater than **20**.

a. Use OpenCV **`cv2.fillPoly`** to draw the **pentagon.**
b. Make sure that the drawn pentagon is located within the picked quarter.
c. Note: the vertices points of the drawn pentagon can be selected randomly as long as they are located in the picked quarter and guarantee that the drawn pentagon will be completely drawn inside the picked quarter. The NumPy package can generate random numbers (see the above link).
d. Copy the output image here.

>>>>>Copy your output image here <<<<<

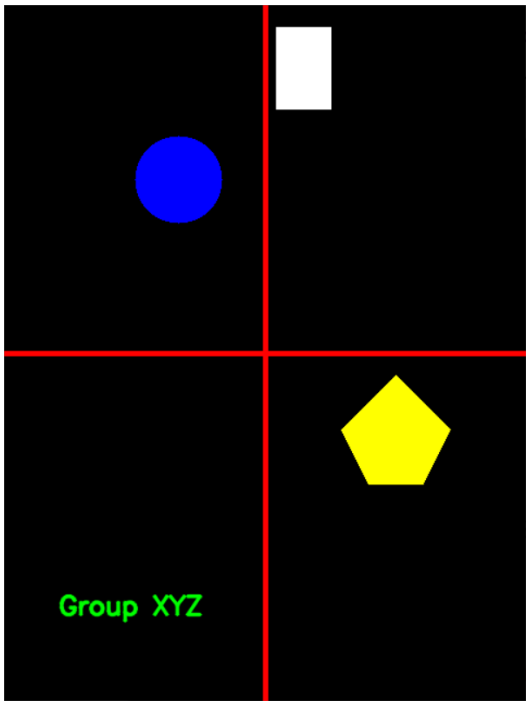
WorkShop3

h) Pick the last remaining quarter and then write your group name in green color on the image produced by the previous part**.**

a. For example, write "Group one" if you are working in group 1.
b. Use OpenCV **`cv2.putText`** to write text on an image.
c. The written text should be completed and located inside the selected quarter.
d. OpenCV supports different FONTs; pick one of these fonts. Here is a short list of these fonts
FONT_HERSHEY_SIMPLEX = 0, FONT_HERSHEY_PLAIN = 1, FONT_HERSHEY_DUPLEX = 2, FONT_HERSHEY_COMPLEX = 3,FONT_HERSHEY_TRIPLEX = 4,FONT_HERSHEY_COMPLEX_SMALL = 5, FONT_HERSHEY_SCRIPT_SIMPLEX = 6, FONT_HERSHEY_SCRIPT_COMPLEX = 7.

i) Save the image into the **output** directory named "**ws3q1.jpg**".

<<One of possible output>>



WorkShop3

Deliverables and Group Work

Create workshop report with the following name format

group_<number>_ws_<workshop number>_report.pdf

For example, if **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 (or part of 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**