# **DPS920/CVI620 - Lab 3**

# **Image Arithmetic and Drawing Tools**

| Total Mark: | 10 marks (3% of the total course grade)   * 6 out of 10: Learn@Seneca submission (Due: Wednesday February 1st end of day) * 4 out of 10: Lab demo (During Workshop of week 4) |
| --- | --- |
| Submission file(s): | * Lab03\_1.py / Lab03\_1.ipynb * Lab03\_2.py / Lab03\_2.ipynb * Lab03.docx |

Please work in **groups** to complete this lab. This lab is worth 3% of the total course grade and will be evaluated through your written submission, as well as the lab demo. During the lab demo, group members are *randomly* selected to explain the submitted solution. Group members not present during the lab demo will lose the demo mark.

Please submit the submission file(s) through Learn@Seneca. ALL team members must submit the final work.

***Please paste the resulting images and answers in this document.***

## **Part I: Image Arithmetic & Histograms**

1. Create a program (save as Lab03\_1). Include code to:
   1. Brightness & Contrast:
      1. Open a color image and display. Paste a sample here.
      2. Increase the brightness by adding a constant (e.g. 100) to the all color channels of the image. Display in a separate window. Paste a sample here.
      3. Increase the contrast by multiplying the image by a constant (e.g. 1.5). Display in a separate window. Paste a sample here.
   2. Linear blend:
      1. Open a second images and display. Resize the second image to match the first, if needed. Paste the second image here.
      2. Ask the user for a number (alpha) between 0 and 1.
      3. Implement a linear blend of the two images:

blend = (1 - alpha) \* img1 + alpha \* img2;

Display the result. Paste a sample here.

* 1. After converting nail\_polish.jpg to HSV color space, apply histogram equalization on the ‘saturation’ channel. Show the equalized image and the changes to the histograms. Paste a screenshot here.

Hint: [Image Histograms in OpenCV Python | Image Processing | coseries](https://www.coseries.com/image-histograms-in-opencv-python/#:~:text=Image%20Histograms%20in%20OpenCV%20Python%20In%20image%20processing%2C,representation%20of%20the%20intensity%20distribution%20of%20an%20image.)

## **Part II: A Drawing Application**

1. Create a program (save as Lab03\_2) and include code from [OpenCV: Mouse as a Paint-Brush](https://docs.opencv.org/4.5.3/db/d5b/tutorial_py_mouse_handling.html) (Advanced demo).
2. Run the program. You should be able to draw green rectangles or small red circles using the mouse.
3. Lower the refresh rate, remove the (-1) flag from the shape functions and change thickness. What do you notice?

Display the result. Paste a sample here.

1. Change the code to do the following:
   1. Allow the user to specify the color for drawing by entering one letter, for example:

* r: red (0,0,255)
* w: white (255,255,255)
* g: green (0,255,0)
* y: yellow (0, 255, 255)
  1. Load a new image (or use the black image) and draw some polylines and polygons with that color.

Hint: <https://mlhive.com/2022/04/draw-on-images-using-mouse-in-opencv-python>

* 1. Save the image when ‘x’ is pressed.

Display the result. Paste a sample here.

## **Part II: Group work**

1. Add this declaration to your file:

We, ------------ (mention assigned group number and 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.

1. Specify what each member has done towards the completion of this work:

|  | Name | Task(s) |
| --- | --- | --- |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |