# **Assignment 2**

# **Image Arithmetic, Histogram Equalization, Drawing Application, Geometric Transformation, Noise & Filtering**

| Total Mark: | 12 marks (6% of the total course grade)   * 9 out of 12: Learn@Seneca Submission (Due: Monday June 9 at 8:00am) * 3 out of 12: Assignment Demo (During the Lab of Week 6) |
| --- | --- |
| Submission file(s): | * Assignment2\_1.py / Assignment2\_1.ipynb * Assignment2\_2.py / Assignment2\_2.ipynb * Assignment2\_3.py / Assignment2\_3.ipynb * Assignment2\_4.py / Assignment2\_4.ipynb * Assignment2.docx (this document with your answers) |

Please work **within your group** to complete this assignment.

This assignment is worth 6% of the total course grade and will be evaluated through your written submission, as well as the assignment demo.

During the assignment demo, group members are *randomly* selected to explain the submitted solution. Group members who are not present during the assignment demo will lose the demo mark.

Please submit the submission file(s) through Learn@Seneca.

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

## **Part I: Image Arithmetic & Histogram Equalization**

Create a program (save as Assignment2\_1). Include code to:

1. Brightness & Contrast:
2. Open a color image and display. Paste a sample here.
3. Change the brightness by a constant (e.g., 100). Display in a separate window. Paste the result here.
4. Change the contrast by a constant (e.g., 0.5, 2). Display in a separate window. Paste the result here.
5. Linear blend:
6. Open a second images and display. Resize the second image to match the first, if needed. Paste a sample here.
7. Implement a linear blend of the two images: create a for loop to linearly blend these images, progressively varying the blending ratio (alpha) throughout the loop.
8. Paste the blended images at different stages of blending (at alpha = 0.2, alpha = 0.5, and alpha = 0.8).
9. (Optional) Save the blended images to a file, create a GIF animation, and submit it with the assignment.
10. Histogram Equalization
11. Open ‘nail\_polish’ (located in the ‘Assignment2\_Files’ folder) and apply histogram equalization to it. To do so, convert the image to HSV color space, apply histogram equalization once on the ‘saturation’ channel and once on the ‘value’ channel. See: [Image Histograms in OpenCV Python | Image Processing | coseries](https://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.).
12. Show (1) the equalized color images and (2) the changes to the color histograms.

## **Part II: Drawing Application**

Create a program (save as Assignment2\_2).

1. Include code from [OpenCV: Mouse as a Paint-Brush](https://docs.opencv.org/4.11.0/db/d5b/tutorial_py_mouse_handling.html) ‘More Advanced Demo’. Modify the program to draw rectangles or small circles with your desired colors. Paste a sample here.
2. Lower the refresh rate, remove the (-1) flag from the shape functions and change thickness. What do you notice? Explain. Paste a sample here.
3. Change the code to do the followings:
4. Load a new image (or use a black image). Once draw **polylines** and once draw **polygons** with your desired colors. See: [Draw on Images using Mouse in OpenCV Python - ML Hive](https://mlhive.com/2022/04/draw-on-images-using-mouse-in-opencv-python).
5. Save images when ‘x’ is pressed. Paste a sample here.

## **Part III: Geometric Transformation**

Create a program (save as Assignment02\_3). Include code to:

1. Open a color image and display. Paste a sample here.
2. In a loop, ask the user whether they want to rotate, resize, apply perspective transformation, or exit.
3. If **rotation** is selected, ask for an angle (in degrees). Then rotate the image around the center of the image with the given angle and displays the rotated image. Run the code with:
4. Rotate for 25 degrees clockwise. Paste the sample here.
5. Rotate for 50 degrees counterclockwise. Paste the sample here.
6. If **resizing** is selected, ask for the resizing factors along each axis and resize the image. Run the code resizing the image to double the width and half the height. Paste the sample here.
7. If **perspective transformation** is selected, apply the transform with the following homography matrix and use same image size for output image. Paste a sample here.

## **Part IV: Noise and Filtering**

Create a program (save as Assignment02\_4). Include code to:

1. Open a color image and display. Paste a sample here.
2. Ask the user for a probability value between 0 and 1 and implement the ‘Salt & Pepper Noise’ with the given probability.

To create this noise, note that:

* The noisy image is the same as image except at affected pixels.
* The number of pixels affected by noise is of image.
* The location of each affected pixel (row and column) is random within possible row and column values.
* The color of the noise is either black or white.

1. Create two noisy images with p = 0.10 and p = 0.20. Paste the results here.

For the following steps, use separable filters where applicable to denoise the noisy images.

1. Apply a Box Filter. Paste the result here.
2. Apply a Median Filter. Paste the result here.
3. Apply a Bilinear Filter. Paste the result here.
4. Compare the results of the above filters (parts 4, 5, 6). Which filter works better for de-noising? Why is that?
5. Try different kernel sizes for Box and Median Filters, for example, and . Explain how the kernel size affects the results. If helpful, include sample outputs with your explanation.

## **Part V: Calculations**

1. In Part IV, select a pixel and output its B, G, and R values, along with the values of its neighbors, before applying denoising using a kernel.
2. Calculate the expected de-noised pixel value. For the B channel, use the Box Filter; for the G channel, use the Median Filter; and for the R channel, use the Bilinear Filter.
3. Output the value of the pixel after the image has been de-noised.
4. Do the output value and the result of your calculation match? Explain.

## **Part VI: Group Work**

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.

Specify what each member has done towards the completion of this assignment:

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