Lab 2 – Basic Image Processing Techniques and Application of OpenCV to Color Processing

Goals:

- *Introduction to basic image processing operations.*
- *Getting familiar with OpenCV library. Basic processing of color images.*
- *Getting familiar with different color spaces.*

Introduction

In this lab we will study basic image-processing operations: image resizing and rotation.

We will analyze the effects of resizing and rescaling on image quality and the methods to improve the quality and avoid artifacts.

We will get familiar with different color spaces while using Application of OpenCV to Color Processing.

OpenCV library

OpenCV is an open source computer vision library, written in C and C++. One of the main goals of OpenCV is to provide a simple-to-use computer vision infrastructure that helps people to build sophisticated vision applications quickly. Among the tools provided by OpenCV are image transformations, histograms, video tracking, 3D vision algorithms and machine learning.

The OpenCV library is available for download from [1] *. In the downloaded package you'll find a tutorial, comprehensive documentation and many useful examples. Also, the OpenCV documentation can be found online at OpenCV homepage [2]. OpenCV programming in Python can be studied using Ref [3].

** OpenCV already installed in Google Colab, use 'import cv2'.*

Preliminary report

Part 1 – Basic Image Processing Techniques

- Study the python functions **resize, rescale** and **rotate** from skimage.transform library.

  Give a short explanation about each of the input argument for these functions.

Give a short explanation about each of the interpolation methods (*order* input) that can be used as an input argument for these functions (nearest-neighbor, bilinear, biquadratic, etc.).

- What problem may occur when down-scaling an image? How is it possible to overcome this problem? Refer to the help of **resize** function and explain how the solution to the above problem is used in **skimage.transform.resize** function.

<u>Part 2: Color Spaces</u>

1. Give the definition of histogram. What is histogram used for?
2. Give a short description of RBG, CMYK,HSV and LAB color spaces. Explain the meaning of each letter in these abbreviations. Give examples for common uses for each of these color spaces. (Hint: the information about color spaces can be found in many digital image processing books, e.g., in [9].)
3. What are the RGB and HSV values for the colors black, white, gray, red, cyan and yellow?
4. Study the OpenCV function **cvtColor** from the cv2 library. Write a command that converts an image, given in BGR color space, to HSV color space. Write a command that performs the backwards conversion (i.e., from HSV to BGR).
5. Study the OpenCV functions **split** and **merge** from cv2 library. Write a command splitting an HSV image into its 3 components (H, S and V) and merging them again to a single HSV image.

Find a grayscale image and a color image to be used in the lab and send it to your email.

Description of the experiment

Open the Jupyter notebook supplied for Lab2 and follow the instructions in the 2 sections:
1. Basic Image Processing Techniques - Resizing (Scaling) and Rotation.
2. Color Space Conversion

Remainder: *for opening Jupyter notebook in the lab:*

1. Open cmd in lab2 folder.

2. Type: conda activate iplab_env.

3. Type: jupyter notebook.

Final report

Part 1 – Basic Image Processing Techniques

Submit the results of the demonstrations from Jupyter Notebook with the image of your choice. Answer all questions and give the relevant examples as instructed in this part.

Part 2: Color Spaces

Submit the results of the demonstrations from Jupyter Notebook with the image of your choice. Answer all questions and give the relevant examples as instructed in this part.

Pay attention to all submission guidelines in the lab1 manual file.
Submit a single PDF file with all your answers.
Verify that the PDF contains all your content.

References

1. Install: https://pypi.org/project/opencv-python/
2. Documentation: https://docs.opencv.org/master/index.html
3. Tutorial: https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
4. A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., 1989 (Library Dewey number 621.368 JAI).
5. http://www.scipy-lectures.org/advanced/image_processing/
6. http://scikit-image.org/docs/dev/index.html
7. https://opencv-python-tutroals.readthedocs.io/en/latest/index.html
8. https://pythontic.com/image-processing/
9. R. C. Gonzalez, R. E. Woods, *Digital Image Processing.* Pearson Education, 2008, Third Edition. Chapter 6 (Color Image Processing).

10. RMS calculation:

$$RMS = \left[\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[\hat{f}(i,j) - f(i,j) \right]^2 \right]^{\frac{1}{2}}$$

where f is the original image and $\hat{f}$ is the output of shrinking and then expanding the image by the same factor.