FCIM.M.IA Spring 2023

Lab 4: Processing Images with OpenCV

Handed out: Tuesday, March 21, 2023

Luna-City changes passports

Luna-City needs your help again. The administration decided to change the passports of the citizens and to automatize a couple of things in the process. They want to keep up with the newest technologies and they know that Artificial Intelligence can help solve all of their problems (at least that's what they think).

Currently, the most important problem in the whole passport issuing process is when a citizen applies for a passport and their photo does not correspond to the requirements for a passport photo. Many of those who want to become citizens bring photos from their vacations, photos taken with friends and even photos which are black and white.

Manually reviewing photos is a laborious task so you are asked to build an automated system to filter photos which can be used for a passport of a future Luna-City resident or not.

Reporting

At the end of this lab, you will need to submit a *report* describing what you have implemented. The *report* must be uploaded on Else, in the according assignment activity. You should use the provided template. Suggested development environments are Google Colab or JupyterLab.

Grading policy

Task 1 Write the following functions using OpenCV. Adjust the parameters and explain your approach. Plot the initial image and the blurred image in the same plot by using Matplotlib subplots. **(1p.)**

- A blurring function;
- A sharpening function.

Task 2 Implement a face detection system using OpenCV. The function should take as input one image and output the result as the coordinates of the face, in case the image contains a face, or `None` if the image does not contain any faces. Assume that the image contains no more than one face. **(2p.)**

Task 3 Implement a system that detects if a photo is accepted for passport or not, by using OpenCV. You can be creative in determining the optimal strategy, but the system should at least follow the listed requirements. **(5p.)**

- The photo should be colored. You can check that by comparing the RGB values of all the pixels. If the image is gray scale image then the values for each pixel should be equal;
- The photo should be in portrait orientation or square. Assume that the image given as input is not rotated;
- The eyes of a subject should be at the same level (with a max error of 5 pixels);
- The photo should contain only one person;
- The head of a person should represent 20% to 50% of the area of the photo.

Task 4 Test how good your system performs on a test dataset. You are required to apply your system to all the images in the test set, then compute the accuracy for the solution. **(1p.)**

- Download the provided image dataset and unzip it in the same folder as this notebook;
- Load the dataset from the unzipped folder, in the subfolder `test_images`. Then read the labels from the file `test.csv`;
- Calculate the accuracy of your system on the test dataset by using the formula.

$$accuracy = \frac{\text{nr of images correctly detected}}{\text{total nr of images}}$$

Report & Presentation Clear explanations, report formatting, code quality, visualisations if relevant etc. **(1p.)**

Plagiarism will not be tolerated.
Good Luck!