FCIM.FIA Autumn 2024
Lab: Processing Images with OpenCV and CNNs

Handed out: Tuesday, November 5, 2024

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. They know that Artificial Intelligence can help solve all 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 the future Luna-City resident.

General Guidelines

- Submit your solution as a .zip archive, containing .ipynb files. Your explanations and reporting MUST be written as markdown cells in your notebook.
- Do NOT host your solution in public repositories (e.g. Github etc). You can use private repositories if you need to.
- **Plagiarism is NOT tolerated!**

Grading Policy

Task 1

- Using OpenCV, write a function to blur an image. Adjust the parameters and explain your approach. Plot the initial image and the blurred one in the same plot using Matplotlib subplots.
- Using OpenCV, write a function to sharpen an image. Adjust the parameters and explain your approach. Plot the initial image and the sharpened one in the same plot using Matplotlib subplots.

(1p.)

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. If the image does not contain any faces, return None. Assume that the image contains no more than one face. **(1p.)**

Task 3 Implement a system that detects whether a photo is accepted for a passport or not, by using OpenCV. You can be creative in determining the optimal strategy, but the system should follow the provided requirements:

- the photo should be colour;
  Hint: you can check that by comparing RGB values of all the pixels; if the image is a greyscale image then (R=G=B) for each pixel
- the photo should be in portrait orientation or square (assume that the image given as input is not rotated);
  Hint: you can use image height and width ratio
- the eyes of the subject should be at the same level (with a max. error of 5 pixels);
- the photo should contain only one person;
- the head should represent 20% to 50% of the area of the photo;
- there are no requirements regarding the background of the photo.

(3p.)

Task 4 Download the provided image dataset and unzip it in the same folder as this notebook. You can use the images folder as data and the labels.csv file for the labels. Split your data into 3 parts the train, the validation and test sets. **(0.5p.)**
Note: while it is not a strict rule, you are encouraged to use the split of 65% for the train, 20% for the validation, and 15% for the test; you can adjust it later for better performance

Task 5 Using Tensorflow or PyTorch, develop a CNN model that will learn how to classify the images as accepted or not for passport photos (binary classifier). Train your model on the train set. It is encouraged to use the validation set for hyperparameter optimization.
Hint: look and research what activation functions are used for binary classification (for the last layer) **(2p.)**

Task 6 Test how well your OpenCV and CNN systems perform on a test dataset. You are required to apply your systems to all the images in the test set, and then compute the accuracy for both solutions. Calculate the accuracy of your system on the test dataset by using the formula:

$$accuracy = \frac{nr\_correct\_detected\_images}{total\_nr\_of\_test\_images}$$

(0.5p.)

Task 7 Elaborate on conclusions. Make sure to cover the following questions:

- Which approach performs better on this task and why?
- Is it useful to use a CNN for this task? Why?
- How can you improve the results obtained with the CNN?
- What can you say about the dataset?
- Do you think such systems would work in a real-life scenario?
- What approaches can be used to obtain more data?

(1p.)

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

Good Luck!