Fundamentals of Artificial Intelligence
Lab 5: Processing Images with OpenCV and CNNs

Luna-City changes passports

The Luna-City needs your help again. The administration of the city decided change the passports of the citizens and to automatize many things. They want to keep up with the newest technologies and they know that Artificial Intelligence can help them solve their problem.

The most important problem when a citizen applies for the passport is that the photos do not correspond to the requirements for a passport photo. That is, many tourists who want to become citizens bring photos from their vacations, photos taken with friends and even photos which are black and white. It's hard for the people from administration to review each photo, therefore you are asked to build an automated system to filter photos which are ok for the passports of the future Luna-City residents or not.

General guidelines

- For this laboratory you are allowed to use only the Python libraries **opencv (cv2), matplotlib, seaborn, numpy**, **pandas**, **Tensorflow** and **PyTorch**.
- Submit the solution as a **.ipynb notebook** uploaded on ELSE, or as a **.zip** file containing your code and a **PDF report**.
- Please don't host your solution in public repositories (e.g Github etc). You can use private repositories if you need.
- **Please refrain from plagiarism**

Grading policy

Task 1 - Simple image processing [1 point]

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

Task 2 - Face detection with OpenCV [1 point]

In this task, 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. For the simplicity, assume that the image contains no more than 1 face for this task.

Task 3 - Passport photo detection with OpenCV [2 points]

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 follow the requirements for the photo below:

- ***the photo should be color***

  hint: You can check that by Comparing RGB Values of All the pixels. If the image is a Gray Scale 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***

Task 4 - Train, validation, test split [0.5 points]

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 set (65%), the validation set (20%) and the test set (15%),

Task 5 - Passport photo detection with Convolutional Neural Networks (CNNs) [2 points]

Using ***Tensorflow*** or ***PyTorch***, develop a CNN model that will learn how to classify the images as accepted or not for passport photo. It will be a binary classifier.

Train your model only on the train set from Task 4. You can use the validation set for hyperparameter optimization.

Hint: look at the softmax activation function for the last layer of the network.

Task 6 - Computing the accuracy on the test set [0.5 points]

In this task, you will test how good your both OpenCV and CNN systems perform on a test dataset. You are required to apply your system to all the images in the test set obtained in Task 3, then compute the accuracy for both solutions.

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 the accuracies for both OpenCV-based system (from Task 3) and CNN sysem (from Task 5)

Task 7 - Formulate conclusions [2 points]

- 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 a system would work in a real life scenario?
- What approaches can be used in order to obtain more data?

Task 8 - Additional approaches to get a better score (e.g regularization, hyperparameter optimization techniques, generative AI, data augmentation, etc.) [0.5 points]

Report & Presentation of the solution [0.5 points]

Clear explanations, report formatting, code quality, visualisations if relevant etc.

Good Luck!