Programming challenge for Perception team member in Kopernikus

At Kopernikus we work with a lot of image and video data. For example, we collect a lot of images during our work from cameras and evaluate objects recognition on them. To organise this dataset, we perform different optimisation techniques. One of them is to remove similar-looking images during data collection.

Your task is to write a program that will find and remove all similar-looking images in a folder. The input of the program should be a path to a folder with images, in the result program should remove all non-essential for data collection images - duplicated or almost duplicated images that have only minor differences from the original that may be considered as non-essential.

In the attachment, you will find `imaging_interview.py` script that contains functions for image comparison. You should use them to implement your solution. Use only provided functions for image comparison, there is no need to develop your own comparison algorithm. We will evaluate how you write the program to clean the data, not the algorithm you will use.

To compare images, you need to run `preprocess_image_change_detection` function on comparing images and then run `compare_frames_change_detection` with a pair of them. The result will be a score that will show how much images differ from each other. You can experiment with different images to get a sense of how the value depends on result.

Use the following subsample of our dataset for your experiments: dataset-candidates-ml.zip

In the dataset, filenames use the following formatting: `c%camera_id%-%timestamp%.png`. Timestamps could be in two formats.

Put your solution on GitHub in an open repo with no view restrictions and send us the link to it.

Please answer following questions:

- What did you learn after looking on our dataset?
- How does you program work?
- What values did you decide to use for input parameters and how did you find these values?

- What you would suggest to implement to improve data collection of unique cases in future?
- Any other comments about your solution?

We wish you good luck with the challenge!