Lab 2 - Filters cont and Introduction to augmentations

Computer Vision (10224)

Goals

- Removing noise from images.
- Implementing image augmentations.
- Implementing template matching.

Preparatory report

- Write a function which addes gaussian noise to an image, the function template is as follows:
  `def add_guassian_noise(img: np.ndarray, mean: float, sigma: float) -> np.ndarray: return noised_image`
  - Pass the parameters below to the function and observe and explain the results.:
    - mean = 0.1
    - sigma = 0.001
  - Do the same as the above with the following parameters:
    - mean = 0
    - sigma = 5

- Write a function which removes noises, and use it to clean the noisy images from above.
  You may choose any filter you deem right for the task. the function template is as follows: `def remove_noise(img: np.ndarray, filter: np.ndarray) -> np.ndarray: return clean_img`
  - Show the original image, noised_image and clean image using **matplotlib subplot**.
  - Show the absoulute differnce between the images and compare their histograms (using numpy).
  - Measure the MSE of the following:
    - `mse(source_img, noisy_image)`
    - `mse(source_img, denoised_image)`
  - Do this on the two noisy images you got previously.
  - Explain the obtained scores.

- Write a function which rotates an image, the function template is as follows: `def rotate_image(img: np.ndarray, angle: float) -> np.ndarray: return rotated_img`
  - The rotation function shouldn't alter the image dimensions.
  - You aren't allowed to use cv2 image rotate function for this task but you may compare your results.
  - Review and write a short summary (up to 5 lines) of albumentations.

Lab Session

1. Write a class which executes the following augmentations:
   1. Flip Vertically or Horizontally (randomly each call)
   2. Rotate
   3. Gaussian-Blur
   4. Median-Blur
   5. Zoom
   6. Crop
   7. Sharpening

Final Report

1. Complete unfinished lab session tasks.
2. Load an image of your choice and create 10 variations of the image using the augmentations you implemented.
3. Show the 10 random images you got in an image (2X5) image grid, e.g. 2 rows and 5 images per row.
4. Optimisation - Revisit your image rotate function which you implemented in the begining of the lab, and check it's execution time. Re-implement the rotation function so the execution time is reduced by a factor of atleast 5 of your original implementation. To check the execution time of a function simply add %time before the execution line for example: `%time res = conv2d(im1, im2)`
   Hint: np.meshgrid
5. Write a short TL;DR (too long didn't read) summary to describe your solution and what you have learned from this lab.

Guidelines

1. Code written in the assignments MUST follow the PEP-8 guidelines, we will deduct points of code not following this guideline, points will be accumulated.
2. You may use either Google Colab or Local Jupyter notebook session.
3. Use Markdown cells in-between your code cells to explain what you've done.
4. Assignments must be submitted as .html with all the required plots. Some exercises will require online reading, you may use Google and stackoverflow for inspiration, although if you use it make sure to attach the reference link.
5. Add as much comments you need to explain yourself, you wouldn't want to assume we fully understand your intuition.
6. Code should be well ordered, formatted and readable.