Exercise 02 for MA-INF 2201 Computer Vision WS24/25
20.10.2024
Submission on 27.10.2024

1. **Convolution Theorem** Given f and g two functions in the spatial domain, prove that their convolution is equivalent to a multiplication in the frequency domain (*i.e.* $f(x) * g(x) \equiv F(\omega)G(\omega)$). Provide an exhaustive explanation for each step of your reasoning.

   (2 Points)

2. **Fourier Transform** In this task, we will put into practice the convolutional theorem. One of the advantages of filtering in the frequency domain is the speed of the process. The objective is to filter the `oldtown.jpg` image with a 7×7 Sobel filter. You are required to:

   - Filter the image in the spatial domain (you **cannot** use `cv2.filter2D`); plot the image.
   - Filter the image in the spatial domain **with** `cv2.filter2D`; plot the image.
   - Filter the image in the frequency domain (you can use `numpy.fft`); plot the image.
   - Compute the mean absolute difference of the three filtered images (in pairs) and the time taken for the processes.

   Provide explanation for each step of your reasoning.

   (3 Points)

3. **Template Matching** In this task, we will implement template matching using two different similarity measures: Sum Square Difference (SSD) and Normalized Cross-Correlation (NCC). Read the image `einstein.jpeg` and the template `eye.jpeg` and convert them to `float` in the range $[0, 1]$.

   - Implement template matching using your implementation of SSD and NCC.
   - In the image, draw the rectangles around the pixels where $similarity \leq 0.1$ for SSD and where $similarity \geq 0.7$ for NCC. You can use `np.where`.

   Now, try to subtract 0.5 to the image (make sure that the values do not become negative) and repeat the template matching. Are there any differences between using SSD and NCC? If so, why in your opinion?

   (3 Points)

4. **Gaussian Pyramid** The professor, during the lecture, told you that using the Gaussian pyramid for template matching will make it faster. Let's see if this is true. Read image `traffic.jpg` and the template `traffic-template.png`

   - Build a 4 level Gaussian pyramid.
   - Build a 4 level Gaussian pyramid using `cv2.pyrDown`. Compare it with your implementation by printing the mean absolute difference at each level.
   - Do the template matching by using your implementation of normalized cross-correlation, print the time taken by this routine.

- Use the pyramid technique to make template matching faster. Follow the procedure described in the lecture slides. Print the time taken by this routine.
- Show the template matching results using the pyramid technique.

You **cannot** use `cv2.matchTemplate`.

(8 Points)

5. **Pyramid Blending** When it comes to deciding who is the strongest modern soccer player, it is never easy to answer. This is why we will create an hybrid between `messi.jpg` and `ronaldo.jpeg` using the pyramid blending technique. You are asked to:

   - Implement `build_gaussian_pyramid` and `build_laplacian_pyramid` functions, using `cv2.pyrDown` and `cv2.pyrUp`
   - Combine the two Laplacian Pyramids
   - Collapse the combined pyramid so as to have the left part of the image with Messi's face and right part of the image with Ronaldo's face.
   - Plot every level of the combined pyramid and the final blended image.

   (4 Points)