Exercise 1 for MA-INF 2201 Computer Vision WS19/20 Submission Date:19.10.2019

October 14, 2019

Please write your code using python 3.7, opencv 4.* and numpy. You are not alllowed to use library functions unless specified otherwise. In case of any question, please write me an email **iqbalm@iai.uni-bonn.de**.

1. **Rectangles and Integral Images**
   Read the image *bonn.png* as a gray image using *cv.imread*
   (a) Compute and display the integral image without using the function *cv.integral*.
   (b) Compute the mean grey value of the image by:
      i. summing up each pixel value in the image, i.e., $\frac{1}{R}\sum_{p \in R} I(p)$
      ii. computing an integral image using the function *cv.integral*,
      iii. computing an integral image with your own function,
   (c) Select 10 random squares of size 100x100 within the image and compute the mean gray value using the three versions. Output the runtime of this task for the three versions in seconds.

   (3 Points)

2. **Histogram Equalization**
   Read the image *bonn.png* as a gray image using *cv.imread*. Perform histogram equalization:
   (a) using function *cv.equalizeHist*.
   (b) using your own implementation of the function equalizeHist.

   and display both results. Compute the absolute pixel wise difference between the results and print the maximum pixel error. (2 Points)

3. **Convolution Theorem**
   Prove that convolutions are in the continuous case associative. (2 Points)

4. **2D Filtering**
   Read the image *bonn.png* as a gray image using *cv.imread*, display it. Filter the image with a Gaussian kernel with $\sigma = 2\sqrt{2}$

   (a) using *cv.GaussianBlur*

   (b) using *cv.filter2D* without using *cv.getGaussianKernel*

   (c) using *cv.sepFilter2D* without using *cv.getGaussianKernel*

   and display the three results. Compute the absolute pixel-wise difference between all pairs (there are three pairs) and print the maximum pixel error for each pair. (2 Points)

5. **Multiple Gaussian Filters**
   Read the image *bonn.png* as a gray image using *cv.imread*, display it. Filter the image

   (a) twice with a Gaussian kernel with $\sigma = 2$

   (b) once with a Gaussian kernel with $\sigma = 2\sqrt{2}$

   and display both results, compute the absolute pixel-wise difference between the results, and print the maximum pixel error. (1 Point)

6. **More on Convolution**
   Prove that convolution two times with a Gaussian kernel with standard deviation σ is the same as convolution once with a Gaussian kernel with standard deviation $\sigma\sqrt{2}$. (2 Points)

7. **Denoising**
   Read the image *bonn.png* as a gray image using *cv.imread*, add 30% salt and pepper noise (the chance that a pixel is converted into a black or white pixel is 30%), and display it. Filter the image (you are allowed to use opencv functions) by

   (a) a Gaussian kernel

   (b) Median filter medianBlur

   (c) Bilateral filter bilateralFilter

   and display the three results. Select the filter size from the range [1, 3, 5, 7, 9] that minimizes the mean gray value distance to the original image. (3 Points)

8. **Separability of Filters**
   Read the image *bonn.png* as a gray image using *cv.imread*.

   (a) Filter the image using the two 2D filter kernels given below.

   (b) Use the class SVD of OpenCV to separate each kernel. If a kernel is not separable, use an approximation by taking only the highest singular value. Filter the images with the obtained 1D kernels and display the results.

(c) Compute the absolute pixel-wise difference between the results of (a) and (b), and print the maximum pixel error.

(5 Points)

$$kernel_1 = \begin{bmatrix} 0.0113 & 0.0838 & 0.0113 \\ 0.0838 & 0.6193 & 0.0838 \\ 0.0113 & 0.0838 & 0.0113 \end{bmatrix} \quad \text{kernel}_2 = \begin{bmatrix} -0.8984 & 0.1472 & 1.1410 \\ -1.9075 & 0.1566 & 2.1359 \\ -0.8659 & 0.0573 & 1.0337 \end{bmatrix}$$