University of Fribourg

Document Image Analysis

Homework 2

Student: Irakli Kelbakiani

Email: Irakli.Kelbakiani@students.unibe.ch

Assignment 2:

Task: Convolution

GitHub:https://github.com/KelbakianiIrakli/DIA_HW1

Branch: DIA-Task2

(a) Add a convolution function to your Python program. Write a function that can calculate the discrete convolution of an image I by a given kernel k without using a package (your own implementation!). You can use NumPy arrays for I and k.

```
def convolve(img, kernel):
    k = kernel.shape[0]
    convolved_img = np.zeros(shape=(img.shape[0]-k+1 , img.shape[1]-k+1)
    for i in range(img.shape[0]-k+1):
        for j in range(img.shape[1]-k+1):
            mat = img[i:i+k, j:j+k]
            convolved_img[i, j] = np.sum(np.multiply(mat, kernel))
    return convolved_img
```

The convolution function creates numpy array of all zeros. The size of the array is image size – kernel_size + 1 (As after convolution is applied image size is reduced). In order to have the final image size equal to the size of the original picture, I added 2 rows and 2 columns filled with all zeros to the original image. I multiply kernel with appropriate matrix and writing result to convolved images corresponding member.

(b) Use your convolution function to apply blurring to the example images provided on ILIAS. Specify what kernels you have used and why.

I have used Gaussian blur 3x3 kernel, as it reduces image noise and gives a strong visual smoothing effect.

(c) Use your convolution function to apply edge detection to the example images provided on ILIAS. Specify what kernels you have used and why.

I have used **Sobel–Feldman operator** as it performs a 2-D spatial gradient measurement image and emphasizes regions of high spatial frequencies.

```
np.array([[-1, -2, -1],
          [0, 0, 0],
          [1, 2, 1]])
```

The first column shows original, the second column shows blurred image, the third column shows edge detection.