1st homework: Digital Image Processing

Tiago Moreira Trocoli da Cunha
RA: 226078
t226078@dac.unicamp.br

I. Homework Specifications

The homework is divided into two parts. First, it is needed to apply four filters in the space domain and generate an image through the combination of other two, after that, it is required to comment about filters and results. Second, it is needed to apply gaussian filter to blur an image.

II. Requiements and Data

The code is written using Python 2.7 in Ubuntu 18.04. I used the following libraries:

- **scipy**: I used *ndimage* to apply convolution and *imsave* to save image.
- **opencv**: just used it to open an image.
- **matplotlib**: used to plot images.
- **numpy**: used for matrix operations and fast fourier transformation.
- **imageio**: used to save images.

The zip file contains *tarefa1.ipynb*, *tarefa1.py* and the image to be filtered *face.png*. The program receives an image face.png and generates two sets of outputs corresponding to the two parts of homework. The first part generates images *image_h1*, *image_h2*, *image_h3*, *image_h4* and *image_h5*. The second part generates images *guass1*, *guass2*, *guass3* and *guass4*. All of them as .png. The code from *tarefa1.ipynb* is equal to *tarefa1.py*.

Python 2.7 warned that *misc.imsave* is deprecated, so I had to use *imageio.imwrite* instead. However, when I apply it to images generated by guassian filter, Python warned about converting float to uint8, thus I prefered to use the deprecated function, since I didn't have time to find out solution for this small issue.

III. Implementation and Results

When filters are applied, they could change the pixel intensities of images out of grey scale range. So, I developed a small algorithm to normalize images, that is, for every pixel intensity x in the image A do:

$$x_{new} = (\frac{x - x_{min}}{x_{max} - x_{min}}) * 255,$$

which x_{max} and x_{min} are the maximum and minimum pixel intensity of A, respectively.

A. Filter for Spacial Domain

Filter mask h_1 has negative weights, except from the central one, so homogeneous regions tend to become black since the linear convolution equation closes to zero when pixel intensities are similar to each other. Moreover, as a high-pass filter it highlights the edges.

$$h_1 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

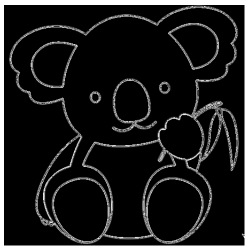

Fig. 1. Image after applying filter h_1.

Since the filter mask h_2 changes the pixel intensity by the weighted average values of its neighborhood, it is a low-pass filter.

$$h_2 = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

So the original image turns into figure 2. Zooming it, we can see that the filter decreases sharpness, becomes more blurry, as expected for a low-pass filter.

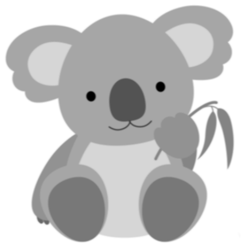
Fig. 2. Image after applying filter h_2.

The filter h_3 turns black homogeneous areas since pixel intensities in these areas are similar to each other resulting low pixel intensity after applying the mask. Filter h_3 also emphasizes vertical or quite vertical edges areas.

$$h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

We can see the image, figure 3, after the filter h_3.

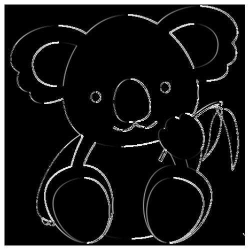
Fig. 3. Image after applying filter h_3.

The filter h_4 do the same as h_3 but with horizontal edges.

$$h_4 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

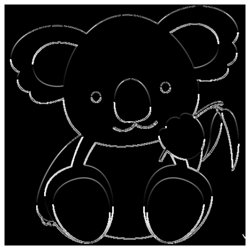
Fig. 4. Image after applying filter h_4.

The last image is a combination of images 3 and 4. Since both emphasizes horizontal and vertical edges we can expect that all of them are highlighted as we can see in figure 5.

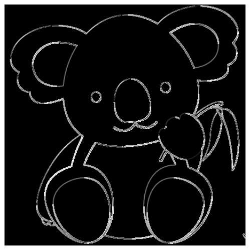
Fig. 5. Resulting image of $\sqrt{h_3^2 + h_4^2}$.

B. Filter for Frequency Domain

I applied four gaussian filters using sigma 2, 4, 6 and 8. Clearly, the greater the sigma the blurrier the image as shown below.

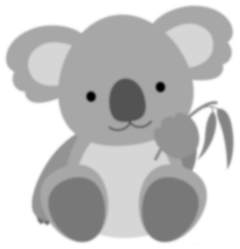
Fig. 6. Guassian filter with sigma 2.

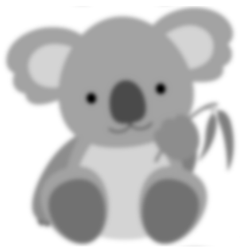
Fig. 7. Guassian filter with sigma 4.

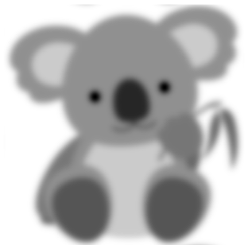

Fig. 8. Guassian filter with sigma 6.

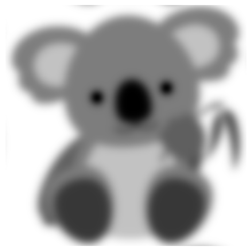

Fig. 9. Guassian filter with sigma 8.

IV. Final Comments

In my perspective, the results were reasonable since they shows what we expected for low-pass and high-pass filters. Nevertheless, I think that the last work form the first part, that is, $\sqrt{h_3^2 + h_4^2}$ had a misleading description. I don't know if h_3^2 is the square elements of the image or is the matrix multiplication of the image by itself. Besides, we had to combine two images after applying the filter h_3 abd h_4, however as described in the aforementioned equation it leads us to think that we had to combine the filters themselves.