Introduction to Image and Video Processing

Coronaproject 3: compression, morphological image processing

Selim Gilon* - i6192074

*Maastricht University - Department of Data Science and Knowledge Engineering

May 16^{th}, 2020

1 Compression

- Make your own Huffman code for your last name in Matlab or Python. Demonstrate the results and discuss them.

  My last name is Gilon. So each letter in this name has probability 0.2 and there are 5 letters in total. See the code for a detailed explanation of the algorithm. The result of my own Huffman algorithm for 'Gilon' is this set of bits: 011011000001.

  Where G = 01, I = 10; L = 11, O = 000 and N = 001.

  Lavg = (0.2)(2)+(0.2)(2)+(0.2)(2)+(0.2)(3)+(0.2)(3) = 2.4 bits/symbol.

  Here is the source reduction tableau.

0.2000	0.4000	0.4000	0.6000
0.2000	0.2000	0.4000	0.4000
0.2000	0.2000	0.2000	0
0.2000	0.2000	0	0
0.2000	0	0	0

Figure 1: Source reduction tableau

Using the same layout as the one studied in class:

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Figure 2: Source reduction tableau from the lecture

Here are the code of the corresponding probabilities.

```
"01"     "00"         "1"          "0"
"10"     "01"         "00"         "1"
"11"     "10"         "01"         <missing>
"000"    "11"         <missing>    <missing>
"001"    <missing>    <missing>    <missing>
```

Figure 3: Code tableau

Again, using this layout:

Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Figure 4: Code tableau example from the lecture

2 Morphological image processing

Here is the image I chose for the next task. It is a picture of some of my buddies and myself hanging out in the Slovenian's nature. My goal will be to binarize it, remove the noise and get the skeleton of our faces.

Figure 5: original image

- Binarize it using a threshold that is based on the image statistics. Hint: e.g. by choosing a certain amount of standard deviations away from the intensity mean. This is just an option, you are encouraged to look up optimal thresholding methods.

  In order to binarize it, I first created a matrix full of zeros (named z) which is as big as the matrix from the gray scale image matrix.

  I then chose to use the standard deviation of the gray-scale values, the mean and a coefficient.

When a value of the grayscale matrix is larger than: $mean + (c \times std)$, the value in the binarized image (in z) is set to 1.

By doing this, I get a black and white image for which "0" values represent black and "1" values represent white. I determined the value of the coefficient by experimenting and analyzing the results. Here is the result with a coefficient of 0.8 (so the treshold = 93.75):

Figure 6: Binarized image

We can see that there is some noose above our head (from the trees), there are quite a lot of details on our faces and on the bridge as well.

- Apply morphological operations that you judge necessary to "clean up" the result of binarizing, e.g. to remove small protrusions, bridges, noise, to smooth out its contour etc. Explain your motivation for the methods used and discuss your results.

  I would like to remove the noise above our faces. I would also like to have less details on our faces. My goal is to distinguish the faces and the bridge behind us.

  Let's try to apply an erosion operation on the image in order to remove unnecessary details.

Figure 7: Image with erosion operator

As expected, the operator removed a lot of details that I wanted to get rid of. Indeed, there are less details on our faces and way less white spots above us.

However, I believe that too many details have been removed by changing their from "1" to "0". For example the bridge's handle is not a nice line anymore but just some white points next to each others. I would like to fill in those gaps, smooth those lines. Therefore, a dilation operator will be chosen for this task. As studied in class, an erosion followed by a dilation is an opening. The functions of the opening are noise removal and boundaries smoothing out: this is exactly the desired effects on my original image. The details wanted should get amplified. Let's try the opening, here is the result:

Figure 8: Image with opening operator

We can see that almost all of the noise from the original binarized image got removed and the image still contains the most important details: on the faces and on the bridge. We can clearly distinguish the bridge and the faces' shapes and that was the desired goal.

All of the previous operations have been done with a disk-shaped structuring element of radius 7. I have done some experiments with some other radius but the results weren't that good when the radius was bigger than 7. Here is the opening with a radius of size 12 for example:

Figure 9: Image with opening operator (r=12)

However, the result with a smaller radius is still pretty good as you wan see here (radius = 5):

Figure 10: Image with opening operator (r=5)

As expected, compared to the opening with a bigger radius (7), there are a little bit more details and noise but the difference is not that obvious. Let's inspect one of the faces to see the difference:

Figure 11: L = original, C = radius 7, R = radius 5

Here, we can see the small difference of details on the 2 openings (center and right).

- Bonus: Find an image where granulometry would be appropriate, implement it, and explain your results.

  For this task, I took a picture of some mint leafs from my garden on a dark background. This image seems interesting for this task because it contains similar patterns of different sizes.

Figure 12: Original image

I applied the granulometry to this image (converted to gray scale first) to find the frequency distribution of disks of different sizes. It starts with a disk-shaped structural element of radius of size 3, increment it by 5 and the max (last) radius size is 50. At each iteration, an opening with disk structural element is applied to the original image.

We can see on the results that the more the radius of the structural element increases, the more the details of the image disappear. With a small radius, we can still see some details on the leafs while with a very big radius (e.g 50) we can't even distinguish the leafs.

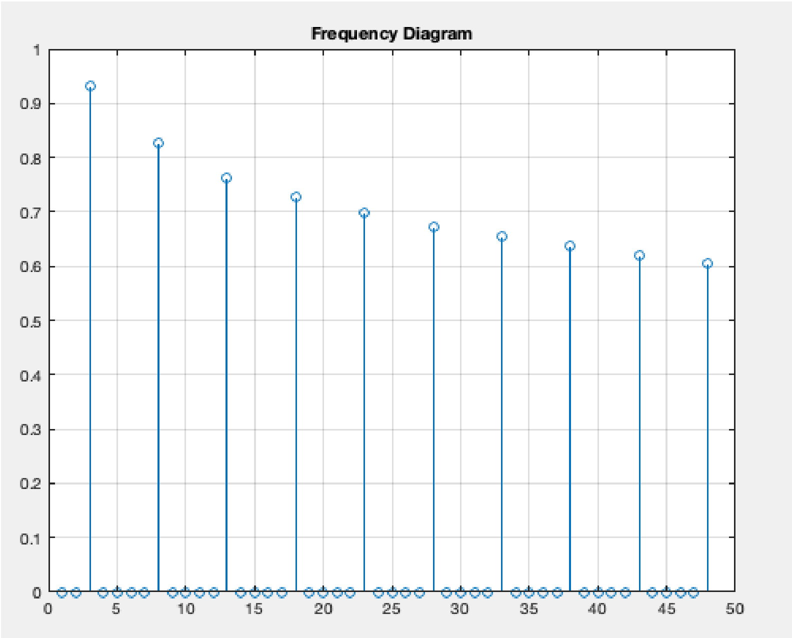

Figure 13: Frequencies plot

We can see on this diagram that the frequency decreases as the radius increases. However, the difference of frequencies between iterations is less and less as the radius increases. Indeed, difference between the ratio at the 1st iteration (r=3) and the 2nd one (r=8) is: $0.930731324864262 - 0.826564637784782 = 0.104166687079$ while the difference between the second to last and the last one is only: $0.620025566659026 - 0.604777401540953 = 0.0152481651181$. Therefore, we can conclude that the effect of the size of the structural element for the opening decreases when the radius size increases.

Figure 14: Radius = 3

Figure 15: Radius = 8

Figure 16: Radius = 43

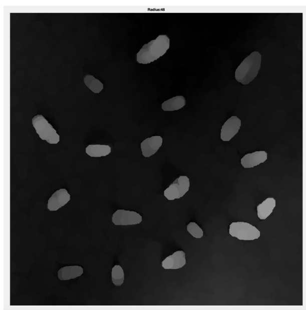

Figure 17: Radius = 48