Introduction to Image and Video Processing
coronaproject 3: compression, morphological image processing
due May 21

Alexia Briassouli (alexia.briassouli@maastrichtuniversity.nl)

Spring 2020

Logistics:

Each of these mini coronaprojects will count for 25% of your final grade. They will be checked for plagiarism (software + text). There might be small bonuses in some places (it will say "**bonus**"). You are requested to please hand in:

1. A report with your answers to the assignment questions and figures with your results. It should be 5-6 pages, but this is just a general guideline. You can use any doc editor you like.
2. The code for producing these results *with clear comments in the code!*. You can use any programming language you are comfortable with (preferably Python or Matlab for this class, but others are welcome). You should explain what you think are important parts of the code in the report (e.g. if you use a special trick that you are proud of).

Your grade will depend on how clearly you present and explain your results in the report and code. You are allowed some freedom to explore, so there is no one correct answer. However, you should demonstrate you have understood the class material and how it applies to these projects.

Assignment:

1 Compression

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

2 Morphological image processing

Choose an image of your liking, that would be interesting in its black and white version.

1. 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.*
2. 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.

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

Alternative Bonus: Make your own hit-and-miss transform example, implement it, and explain your results.