

EE551 Assignment 2

Industrial Visual Inspection in the Presence of Noise



Background

This Assignment extends Assignment 1 by carrying out the same operation, i.e. visual inspection of Coca-Cola bottles on a production line, **but in the presence of noise**. You will re-use whatever code you developed in Assignment 1 for the purposes of this Assignment.

The Assignment consists of two parts:

1. Evaluate the performance of the algorithms you developed in Assignment 1 in the presence of noise at different levels. This is largely “experimental characterisation” so that you can assess how robust the system you have already developed is.
2. Carry out **filtering** of the noisy images before you process them, and again evaluate the performance of the algorithms you developed in Assignment 1.

Note: Recall that Assignment 1 included sample images covering a set of 7 “faults” as follows:

1. bottle **under-filled** or not filled at all
2. bottle **over-filled**
3. bottle has **label missing**
4. bottle has label but **label printing has failed** (i.e. label is white)
5. bottle **label is not straight**
6. bottle **cap is missing**
7. bottle **is deformed** (i.e. squashed) in some way

There were also “combinations” of faults present in some of the images.

To simplify the problem in Assignment 2, it is not necessary to do a full evaluation of your visual inspection system over all 7 “faults” (or combinations) from Assignment 1. You can restrict the investigation to any **two** faults of your choice (e.g. you may choose only “label missing”, “printing failed”, “cap missing”, or any other subset you like). Then, you simply need to use the sub-set of sample images already placed on Blackboard for Assignment 1, for each of your chosen faults.

The objective of this Assignment is not to assess the inherent merit of your visual inspection system with clean images (that was the objective of Assignment 1); rather, the objective of this Assignment is to examine the effects of noise on your visual inspection system (however good or bad it is to start with), and then see if you can improve performance by filtering the noisy images.

EE551 – Assignment 2 – Industrial Inspection in the Presence of Noise, V2, March 2021

The individual components of the Assignment will now be discussed in more detail.

1. Evaluation of Visual Inspection System with Noisy Images

The first task is to evaluate your “baseline” visual inspection system from Assignment 1 with noisy images, for each of your **two chosen faults**. For each of the sample images, you will need to add noise. You can use the “imnoise” function for this task, and you can choose Gaussian noise with a mean of 0.

The first requirement is to “characterise” the system in the presence of increasing levels of noise variance, for each of the **two** faults (for example, you could do this using “for” loops in Matlab to automate the process). In essence, this requires determining the “average system performance” as a function of noise variance. To quantify *system performance*, use whatever way you quantified performance of your system in Assignment 1. For example, you may have used the percentage of the sample images for each fault in which the fault was correctly detected. If you start with low levels of noise variance, chances are your inspection system will work just fine when tested with the noisy images (or at least as well as it worked with clean images); however, at higher levels of noise, you should start to see your system fail.

Display the overall performance of the system as a graph of performance against noise variance.

Comment on the relative performance of your system with the clean images (which you’ll have done in Assignment 1) and with the noisy images. Are there any discernible trends e.g. does the detection of any particular fault perform better or worse in the presence of noise?

2. Investigation of Noise Removal using Filtering

The second part of the assignment involves adding filtering as a form of “pre-processing” of the noisy images, before processing them using your visual inspection system. While the addition of noise causes an increase in the error rate of the system, the objective here is to reduce the number of errors by cleaning up the signals. To simplify the problem, you can restrict the investigation to **ONE** of the two faults you looked at in Part 1. Furthermore, you can restrict the evaluation to a single noise variance value that is enough to cause some amount of error in the visual inspection – then, the objective is to “recover” some performance by using noise removal.

First, select one of the two faults from Part 1. It doesn’t really matter which one, but obviously it should be one that doesn’t perform all that well with noise so you actually have some scope for improvement. ☺ Secondly, from the plot of system performance vs. noise variance for that fault, pick a noise level that causes some level of “system failure”. In a sense, this is your chosen “operating point” for the test. Choose a value of noise variance that causes enough errors to be significant, but not too high – otherwise there’s a chance that the filtering to remove noise won’t work at all!

Once you’ve selected your operating point, then you are required to investigate the effect of filtering to remove the noise. This is basically an “extension” of the investigation in Part 1, with the addition of filtering before the visual inspection. The objective here is to obtain an improvement (if you can) over the performance of the system with no filtering.

EE551 – Assignment 2 – Industrial Inspection in the Presence of Noise, V2, March 2021

You have two approaches that you can use here (i) spatial domain filtering (as covered in Section 3 and Chapter 4 of the module textbook); and (ii) frequency-domain filtering (as covered in Section 4 and Chapter 5 of the module textbook). Investigate **two** spatial domain filter approaches (choose from Averaging, Gaussian, Laplacian) of different neighbourhood sizes to see which, if any, provide an improvement in performance of the visual inspection system. Investigating different neighbourhood sizes is something that you can “automate” using *for* loops in Matlab.

Also investigate if frequency domain filtering gives you any improvement in visual inspection performance. You can use the examples from Section 4 as an indication of how to design a simple ideal “low pass” filter in the frequency domain. Experiment with different values of “spatial cut-off frequency” (as expressed in the “size of the filter” in pixels in the “spatial frequency domain” – refer to the case we looked at in the example “basic_FD_filtering.m” on Blackboard). Choose a number of different values of cut-off frequency. Again, this is something that you can readily automate with the use of *for* loops in Matlab.

Comment on whether any of the approaches you have examined results in an improvement in performance (again, as expressed in e.g. percentage of images in which the fault was correctly detected). Which approach gives you the best overall performance improvement in the presence of noise?

At this point, you will have determined which filtering approach works the best for whatever noise variance value at which you did your investigations, and for whichever fault you chose for your investigations. Now, extend this to examine the full range of noise variance values that you looked at for that fault. Re-run the experiment for all values of noise variance and note the average system performance for each noise value. This will give you a second plot of system performance vs. noise variance, with the filtering applied. You can directly compare this with the plot from Part 1 that had no filtering applied to the noisy images (for whichever fault you chose for Part 2). Comment on the relative performance of the two cases.

Marks

The marks for this Assignment will be awarded as follows:

Part 1

- Characterisation of your chosen two faults in the presence of noise: **20% in total**
- Commentary on your experimental results. What useful insights can you make on the results? **10%**

Part 2

- Investigation of the effect of **two spatial domain filtering** approaches, as a function of spatial extent of the filter: **10% for each spatial filter examined = 20% in total.**
- Investigation of frequency domain filtering – test an ideal low-pass filter with different spatial cut-off frequencies as described above. **20%**
- Characterisation of the system with the “best” filter, over the full range of noise variances. **10%.**
- Analysis and observations on the results: **10%.**
- General presentation quality of your report, quality/clarity of code, use of comments etc.: **10%**

Total 100%

Submission

You must submit the following:

- Working Matlab .m script for implementing the system.
- Report detailing your work, including a comprehensive set of results, graphs of performance, and your analysis and critique of the results. Provide any illustrative images (as many as you feel necessary) to illustrate what's happening (e.g. a few cases of clean, noisy and "filtered" images). Reports should be as long as they need to be and no longer (i.e. no "padding"!). Submit your report as a PDF or a Microsoft Word document. Further details on submission (incl. deadline) will be communicated to you via e-mail.

Details on submission, deadlines etc. will be communicated to you via e-mail.

Notes

1. Marks are awarded for the quality and performance of your algorithm as well as for "working code", for each fault case. Where appropriate, your report should include a description of your algorithm that is, in a sense, "software independent", so use flow charts, pseudo-code, equations etc. as you need to (given the relative simplicity of the noise reduction methods, it is not expected that the algorithmic description will be excessively long – cover the main parameters of your noise removal approaches, e.g. range of neighbourhood sizes, etc.). There is no need to include a detailed description of the fault detection system itself, as you'll have already presented this in your report for Assignment 1. Concentrate only on the added noise reduction functionality. Marks will also be awarded for the "critique" of your system and its performance.
2. Your program must run on a PC with MATLAB and a copy of the image processing toolbox – no third party tools or scripts may be installed.
3. You should structure your source code so that all that needs to be done is to "point" the software at a directory that contains a set of test images to be processed – these could be directories that contain only one type of fault, or it could be a directory that contains all of the randomised images. It **must** be possible to run your software with minimal editing i.e. only directory names should need to be changed to reflect differences in the file structure on your own laptop and those who will be assessing your submissions. For submission, you can have all of your code ("main" program and any functions) in one source M-file; however, if you do this, remember that the functions will have limited scope and will not be visible outside this source file.
4. For submission, you can have all of your code ("main" program and any functions) in one source program; however, if you do this, remember that the functions will not be visible outside this source file. Also, for submission, if TurnItIn has difficulties with the .m file format (it also can have difficulties with Zip files) you can improvise and save your Matlab script as a .TXT document or MS Word document and upload that – as long as it's uploaded in some "text" format, that's fine. You should also e-mail your report and source code (.m file) to edward.jones@nuigalway.ie as well.

Plagiarism

You will no doubt be aware that the University treats plagiarism very seriously. Students are expected to work independently and to submit their own original report, without copying material

EE551 – Assignment 2 – Industrial Inspection in the Presence of Noise, V2, March 2021

from another person, or from another source. Submissions will be processed by Turnitin to detect plagiarism. Further information relating to plagiarism may be found at: <http://www.nuigalway.ie/plagiarism/> .

Also, while you may discuss and debate elements of the assignment with your classmates, **each student must write their own code, and complete and submit their report independently of all other students** (TurnItIn will be used to check for plagiarism) – in other words, you can “collaborate, but don’t copy”.