**Practical Exam and HW10 – Visual Inspection**

# *Production Line Visual Inspection : Coca-cola Bottling Plant*

**Deadline:**
I want you to pretend that you were doing this homework, and were limited to one hour of coding on a computer and one hour of typing up what you did – at maximum. I don't want you to write a huge decision tree. I don't want you to spend many hours trying to get it perfect. It is only worth one homework assignment.

Expect that your program will get several cases wrong. You will have false alarms, and you will have misses.
Do not spend excessive precious time before finals week on this. Limit yourself to a maximum of two hours.

The deadline for this is Monday night, but I am hoping that you will get it done sooner. In any case, plan your time judiciously. Do not procrastinate. I know some of you have other assignments due Monday night.

Best Regards,

- Professor Thomas Kinsman

**A Degree of Confidence Metric:**
In Computer Vision, once you detect a license plate, a stop sign, a face, or any object, you should then develop a separate degree of confidence metric (DOC) to determine how good your match is. Ideally the DOC should be developed using a different algorithm, different from the original algorithm that detected the object in the first place. The goal is to have an independent algorithm "cross-check" the first algorithm.

Here you are presented with the problem of identifying problematic Coca-Cola bottles. The problems include: bottles that are crushed, bottles that are under-filled, bottles that are over-filled, bottles that are missing their labels, and bottles that are missing their caps. The authors of our textbook wanted you to develop separate classifiers for all of these separate cases. However, you can achieve most of the job by creating one single DOC, and measuring the DOC for the center bottle of the assembly line. You are not told how to do this. You must find the technique yourself.

**Background**
*The most common use of image processing in an industrial setting is for the automated visual inspection of products leaving a production facility. Automated inspection is used to inspect everything from pharmaceutical drugs to textile production. It is estimated that the majority of products bought on supermarket shelves are inspected using automated "machine vision" based systems prior to dispatch to avoid the cost of shipping a faulty item to a supermarket shelf that no-one wants to buy!*

In this practical exercise we are dealing with a bottling production line in a facility bottling Coca-Cola for the UK domestic market. We have a set of images, taken under near constant factory lighting conditions, of the bottles as they leave the bottling line. The bottling company requires a vision system to automatically identify a number of different faults that may occur during filling, labeling and capping stages of production so that these bottles can be intercepted prior to shipping.

Your task is to design and prototype a vision system to detect the set of fault conditions that may occur together with identifying the type of fault that has occurred. You will develop this system using the software and the techniques discussed in the course.
**Task Specification – Automated Visual Inspection**

**Practical Exam and HW10 – Visual Inspection**

You are required to develop a visual inspection system that correctly identifies when any of the following conditions occur in the bottling plant:

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*

In each image we are ***only interested in classifying the central bottle in the image***. One image is taken for each bottle leaving the production line so faults occurring in bottles at the sides will be detected separately when those particular bottles are themselves photographed at the center of the image. **Additionally, some images may have no bottle in the center of the image – this is not a fault, just a gap in the production flow stemming from a machine operating further up the line.** *Faults with side bottles and missing bottles must be ignored by your system – only the seven faults above must be reported.* You can tackle each of the cases in any order you wish.

Some examples will contain more than one fault with the central bottle but these cases are rare – identify as many as you can.

*Design is important - think about the problem, think about the solution, and then write the code. Then test.*

As this is only a prototype – efficiency of your approach is less important than performance. Your code will be tested with a separate set of images that you will not see. (A validation set is not seen by the clients.) The validation images were collected under the same conditions using exactly the same camera setup (type, position, etc).

**Sample Data**
The sample data provided is a set of 141 images of bottles leaving the bottling facility. The images have been captured by a high-speed image camera as they leave the facility – although no motion blur occurs in the images, the positions of the bottles in the image varies slightly from image to image.

- The full set of (randomly ordered) images is available on myCourses in the file ALL_Bottles_for_Class.zip.

**REQUIREMENTS – WHAT TO TURN IN TO DROPBOX HW10:**

1. Create one directory named HW10_Firstname_Lastname. Make the obvious substitutions. (Don't get silly use your own name.) Put everything in this one directory. When ready, zip up the entire directory and submit it.

2. Write a function called: `HW10_BottleChecker.m`.

3. The function takes in the filename of one image and returns:
   a. A value of `true` if the center bottle is missing, or if the center bottle if fine.
   b. A value of `false` if the center bottle is defective in any way.

4. What to submit:
   a. Your function, names as in #2 above. This is a Matlab file that is <u>well-documented</u>.
   b. Any supporting functions you wrote in separate files.
   c. A one page write-up, in PDF format:
      i. The write-up will be a file named HW10_Firstname_Lastname.pdf.
      ii. A description of your imaging chain, in numeric order, numbering all of the steps you used.
      iii. Any supporting text needed to clarify how your algorithm works.

5. The grader might run your program, might read only your Matlab file, or might only read your write-up.
   So your program should work, and both the code and the write-up should both describe how you solved the problem.

Plagiarism: You must not plagiarize other's work. You must submit your own work. Do not work together. This isn't that complicated.