Carnegie Mellon University Africa

04-800-G Applied Computer Vision

Assignment 1
Counting Sweets

Deadline: 18:00 Thursday 12th October 2017

Problem Definition

The objective of this assignment is to develop a computer vision application that can count the number of disk-shaped coloured sweets in a scene. Two types of scene are possible, one where all the sweets are separated from each other (see Fig. 1) and one where the sweets are touching each other (see Fig. 2). In all cases, the scene background will be white and only complete non-defective sweets will be present. Equal marks will be awarded for the ability of the application to count the number of sweets in the first (separated) and second (touching) scenarios.

Figure 1.

Figure 2.

The application should read image input from image files, each file comprising an image of a scene comprising coloured sweets. The image file names should be provided in an input file named input.txt. The application should write the required data, i.e., the number of sweets in each image, to an output file named output.txt.

Input

The filenames of the images of the scenes containing coloured sweets.

Output

The output should begin with your Andrew Id on a separate line; you Id comprises the first letter of your first name and your last name in full (e.g. dvernon).

This should be followed by the input image filenames and the number of sweets in the image, one filename and number per line.

Sample Input

```
../data/assignment1A.jpg
../data/assignment1B.jpg
```

Sample Output

```
dvernon
../data/assignment1A.jpg: 20 sweets
../data/assignment1B.jpg: 30 sweets
```

Instructions
Submit the following in a zip file named with your Andrew Id to vernon@cmu.edu by the deadline shown above.

1. The source code: *.c, *.cpp, and *.h file(s)
2. The cmake file: CmakeLists.txt
3. The test input file: input.txt
4. The test output file: output.txt
5. The image files you used to test your program.

Do not include the Visual C++ solution files or the executable. Submit only the source code files, the Cmake file, and the input, output, and image files; do so in a single compressed (zipped) folder. Do not include subdirectories in the zipped folder.

The source code should contain adequate internal documentation in the form of comments. Internal documentation should include the following.

- A summary of the algorithm
- A summary of the manner in which the code was tested.

Do not forget to include your name in the internal documentation. Place this documentation in the .h include file.