

Assignment A7: Image Features and Segmentation

CS 4640
Fall 2021

Assigned: 9 November 2021

Due: Part 1: 18 November 2021

Due: Part 2: 2 December 2021

1. Develop a Matlab function, *CS4640_inspect*, that does the following order of inspection:

```
for every test image
  if no middle bottle, then set bit 8
  else
    if underfilled, then set bit 1 end
    if overfilled, then set bit 2 end
    if label missing, then set bit 3 end
    if label printing failed, then set bit 4 end
    if label not straight, then set bit 5 end
    if cap missing, then set bit 6 end
    if bottle deformed, then set bit 7 end
  end
end
report results
```

Consider having one function return the answer for filled errors and one for label errors. The returned report is a struct with one field to provide the name of each image, and the second to provide a detected defects array.

For 18 November, this function should run, but only implement defect detection for: no bottle, and no cap. The output variable should be an $n \times 8$ array, where n is the number of files in the test directory.

```

function [report,defects] = CS4640_inspect(d_name)
% CS4640_inspect - inspect all images in a directory
% On input:
%     d_name (string): name of directory path
% On output:
%     report (vector struct): info on defects
%
%     (k).name (string): image name (with directory path)
%
%     (k).defects (string): defects of kth image
defects (nx8 array): all defects info
%     col 1: overfilled
%
%     col 2: underfilled
%
%     col 3: label missing
%
%     col 4: label white
%
%     col 5: label not straight
%
%     col 6: cap missing
%
%     col 7: bottle deformed
%
%     col 8: no bottle
%
% Call:
%     [report,defects] = CS4640_inspect(d_name);
% Author:
%     <Your name>
%     UU
%     Fall 2021
%

```

2. Develop a Matlab function, *CS4640_performance*, which takes the results of an inspec-

tion and produces measures of performance in terms of *precision* and *recall*:

$$precision = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{retrieved\}|}$$

$$recall = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{relevant\}|}$$

Produce a table with 3 columns: *defect*, *precision*, and *recall*, and with the defects column going from 1 to 8 followed by all. E.g.:

<i>Defect</i>	<i>Precision</i>	<i>Recall</i>
1		
2		
3		
4		
5		
6		
7		
8		
all		

For example, to find precision for row 1, let O be the set of images with overfilled bottles, let D be the images detected as overfilled, let $I = O \cap D$, then precision is equal to $\frac{|I|}{|D|}$; recall is equal to $\frac{|I|}{|O|}$. For the row with 'all,' the precision and recall are to be determined for relevant meaning has an error of some type, and retrieved means determined to have an error by inspect.

For 18 November, this function should run on output from the inspect function.

```
function P = CS4640_performance(report, defects)
% CS4640_performance - calculates precision and recall for inspect on DB
% On input:
%   report (nx8 array): defects as determined by inspect
%   defects (nx8 array): ground truth for dataset
% On output:
%   P (9x3 array): performance result; rows 1 to 8 characterize
%   performance on individual errors, row 9 characterize overall
%   error detection
```

```
%      col 1: defect index
%      col 2: precision
%      col 3: recall
% Call:
%      P = CS4640_performance(report,defects);
% Author:
%      <Your name>
%      UU
%      Fall 2021
%
```

3. Develop the Matlab function, *CS4640_var*, which computes the gray level variance in every 5x5 window (may exclude border). Use this feature to detect cap missing defect. Discuss issues and show results.

```
function T = CS4640_var(im)
% CS4640_var - compute 5x5 neighborhood variance
% On input:
%     im (MxNx3 array): RGB image
% On output:
%     T (MxN array): variance of 5x5 window at each pixel
% Call:
%     T = CS4640_var(bot1);
% Author:
%     <Your name>
%     UU
%     Fall 2021
%
```

4. Use the Matlab function from 3, *CS4640_var*, along with PCA analysis of connected components, to detect label defects (i.e., missing label, label not straight, and white label). Discuss issues and show results.

5. Use the Matlab function from 3, *CS4640_var*, to detect the deformed bottle defect. Discuss issues and show results.

For 2 December, put together your best detector functions for the defects, and report performance (from *CS4640_inspect* and *CS4640_performance*). This should be your best shot at producing the best complete inspection system. Describe performance on 141 test image dataset.