UNIVERSITE DE BOURGOGNE

IMAGE PROCESSING

Semester Project

Automated Visual Inspection in Soft Drink Bottling Plant

Authors:

Abdullah Thabit
Tewodros W. Arega
Zohaib Salahuddin

Supervisor:
Desire SIDIBE

Contents

1 Introduction

1.1 Overview

In the last decades, new computer vision technologies and image processing techniques have been very important in the improvement and automation of manual inspections in industrial and manufacturing applications.

Machine vision (MV) is the technology and method used to provide imaging-based automatic inspection and analysis for applications such as automatic inspection, process control, and robot guidance, usually in industry. These intelligent inspection systems come equipped with one or multiple cameras, and lighting. They are capable of measuring parts, verifying parts whether they are in the correct position, and recognizing the shape of parts.[2]

In this project, we are dealing with an inspection system for bottling production line. More specifically, to correctly identifies the following seven fault conditions that may occur in the bottling plant: bottle under-filled or not filled at all, bottle over-filled, bottle has label missing, bottle has label but label printing has failed (i.e. label is white), bottle label is not straight, bottle cap is missing, bottle is deformed (i.e. squashed) in some way.

1.2 Objective

The main objective of the project is to design and prototype an image processing system that detects the set of fault conditions that occurred during the production, to identify the type of fault that occurred and to indicate the locations of the fault in the image. The following faults are to be detected and classified:

- Classify Middle Bottle Missing as Normal Case
- Bottle Under-Filled or Not Filled at all
- Bottle Over-Filled
- Bottle has Label Missing
- Bottle has Label but Label Printing has Failed (i.e. Label is White)
- Bottle Label is not Straight
- Bottle Cap is Missing
- Bottle is Deformed in Some Way
- Detecting Multiple Faults on the Bottles

2 Methodology

2.1 Middle Bottle Missing - Normal Case

The first step in the implementation of this project was to detect the presence of the middle bottle. If the bottle was present, we can proceed with the detection of faults. However in the absence of the presence of the middle bottle, this was classified just as a normal case. This is because it is just a mere gap in between the bottles but not a fault.

We Proceeded to detect the presence of the middle bottle by taking a window in the middle of the image as the middle bottle is present in the center of the image.

After cropping the image by placing the window at the center of the image, we applied thresholding. In case of the presence of the bottle, there is a variation in intensity. However , in case of absence of the bottle, the background is just plain white. The threshold was carefully chose. This can be illustrated with the help of the following pictures:

Figure 1: Original Image Showing Middle Bottle Missing

Figure 2: Region Cropping For Detecting if the Bottle is Present

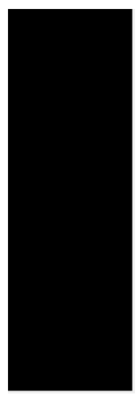

Figure 3: Applying a Threshold

The Threshold was chosen after the careful study of the normal bottles and the bottles that were missing. For Thresholding, the values of intensities were chosen to be between **150 and 230**. **The intensities between this region were then inverted**. This is because these are the intensities that are present in case of occurrence of the white background which shows up when the middle bottle is missing. The white pixels were then summed up and their percentage was calculated. **The percentage threshold for detecting the middle bottle was set at 5. If the**

percentage was less than 5, it was classified as a bottle missing case.

The following figures illustrate the Normal Case:

Figure 4: Original Image Showing Middle Bottle Present

Figure 5: Region Cropping For Detecting if the Bottle is Present

Figure 6: Applying a Threshold

2.2 Detecting the Center of the Middle Bottle

After the detection of the middle bottle, we proceed on to find the center of the middle bottle. This is necessary for the detection of faults like deformation and checking if the label is straight. After careful consideration, it was observed that the cap region of the bottle could be used to localized the center of the middle bottle. The center of the middle bottle varies in many cases.

A window was placed on the top of the bottle around the cap region. There were two cases that need to be countered in order to detect the center of the middle bottle using the cap region.

The first case was detecting the center in case the cap was present. The second case was detecting the center in case the cap was absent. The threshold was carefully tuned to adjust both the case.

The following steps were performed in order to compute the center of the middle bottle using the cap:

- Put the window in the cap region
- Threshold the cap region with the value set at 180.
- Applying image closing to fill in the holes
- Find the center only for values greater than one.

The following series of images illustrate the case of detecting the center in case the cap is present:

Figure 7: Original Image with Cap

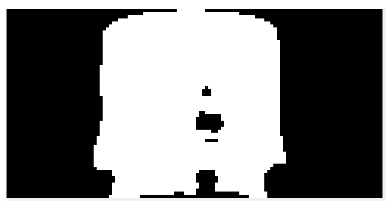

Figure 8: Thresholding the Cap Region

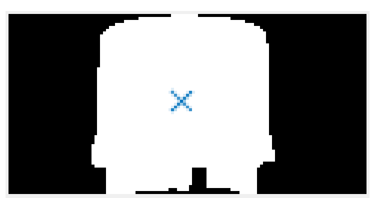

Figure 9: Applying subsequent Image Closing and Detecting the Center

The following series of images illustrate the case of detecting the center in case the cap is absent:

Figure 10: Original Image without Cap

Figure 11: Thresholding the Cap Region

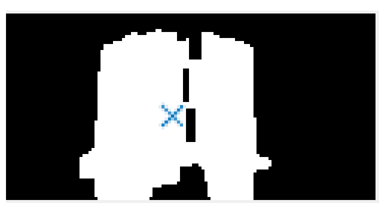

Figure 12: Applying subsequent Image Closing and Detecting the Center

2.3 Faults Detection

After the detection of the center of the middle bottle and also confirming that the middle bottle is indeed present, we proceed towards the detection of the various faults that could occur. It should be noted that special care was taken during the detection of these faults that they do not overlap so that multiple faults can be detected at the same time.

2.3.1 Over-filled Bottles Detection

In order to detect if the bottle is overfilled or not, the image was cropped for the middle bottle right above the normal level. The following steps were performed in order to get a parameter concerning the detection for over filled bottles:

- Crop just above the normal level of liquid
- Threshold the cropped section to detect the presence of liquid.
- The Threshold is set at 150.
- Invert the image obtain to detect the presence of black liquid.
- Take the Percentage of the white pixels in the cropped region.
- Apply a careful threshold to determine the faults.

The following figures illustrate the detection of overfilled bottles:

Figure 13: Overfilled Bottle

Figure 14: Cropping in the region above the normal level

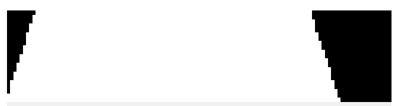

Figure 15: Thresholding and Inverting in the region above the normal level

The following figures illustrate the detection of overfilled bottles in normal case:

Figure 16: Normal Bottle

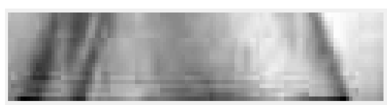

Figure 17: Cropping in the region above the normal level

Figure 18: Thresholding and Inverting in the region above the normal level

2.3.2 Under-Filled Bottles Detection

In order to detect if the bottle is underfilled or not, the image was cropped for the middle bottle right below the normal level. The following steps were performed in order to get a parameter concerning the detection for over filled bottles:

- Crop just below the normal level of liquid
- Threshold the cropped section to detect the presence of liquid.
- The Threshold is set at 150.
- Invert the image obtain to detect the presence of black liquid.
- Take the Percentage of the white pixels in the cropped region.
- Apply a careful threshold to determine the faults.

The following figures illustrate the detection of underfilled bottles:

Figure 19: Underfilled Bottle

Figure 20: Cropping in the region below the normal level

Figure 21: Thresholding and Inverting in the region below the normal level

The following figures illustrate the detection of underfilled bottles in normal case:

Figure 22: Normal Bottle

Figure 23: Cropping in the region below the normal level

Figure 24: Thresholding and Inverting in the region below the normal level

2.3.3 No Label Found

In order to detect the presence of the label, we cropped a region around the middle of the label in the middle bottle. This was an easy case to detect as in case of the absence of the bottle, the liquid in the bottle has very low intensities. So a threshold with high intesity value of 70 was applied. The intensity profile was then inverted. This low threshold was chosen so that the label print missing case and the label absent missing case could be detected simultaneously. The percentage of white pixels in the threshold regions was then determined.Subsequently, a careful threshold was applied to get the detection of this fault.

The following images illustrate the detection of 'no label found' fault in case of faulty image:

Figure 25: Image with a fault

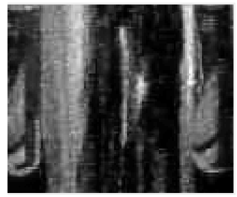

Figure 26: Cropping the label region

Figure 27: Thresholding and inverting the label region

The following images illustrate the detection of 'no label found' fault in case of normal image:

Figure 28: Image without a fault

Figure 29: Cropping the label region

Figure 30: Thresholding and inverting the label region

2.3.4 Label Printing has Failed

In order to detect if printing has been done on the label, we proceed with the following method:

- Crop a square Region of the middle bottle around the Center.
- Threshold on this cropped region with a value of 150. Note that the value was 70 in case of no label fault detection case.
- After applying the threshold, invert and find the percentage of white pixels.
- Apply a threshold to the percentage to detect the fault between normal bottle and faulty bottle.

The following images illustrate the detection of 'no label print' fault in case of faulty image:

Figure 31: Image with a fault

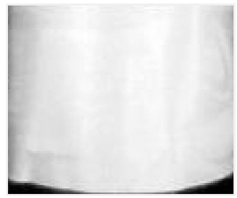

Figure 32: Cropping the label region

Figure 33: Thresholding and inverting the label region

The following images illustrate the detection of 'no label print' fault in case of normal image:

Figure 34: Image without a fault

Figure 35: Cropping the label region

Figure 36: Thresholding and inverting the label region

2.3.5 Bottle Cap is Missing

In order to detect if the middle bottle has a cap or it is missing, we need to perform the following steps on each image:

- Crop a region around the cap.
- Apply the threshold cropped region. A threshold of 150 was set in order to differentiate between a presence and absence of the cap.
- Invert the threshold region and find the percentage of the white pixels. An appropriate threshold was set in order to differentiate between the two cases.

It can be noted that the absence of the cap introduces more white intensities. Hence, this region in case of absence would have higher intensity values. This observation was used to detect this fault.

The following images illustrate the detection of 'no cap found' fault in case of faulty image:

Figure 37: Image with a fault

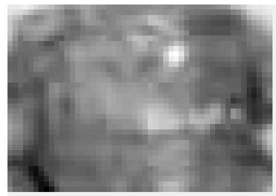

Figure 38: Cropping the cap region

Figure 39: Thresholding and Inverting the cap region

The following images illustrate the detection of 'no cap found' fault in case of normal image:

Figure 40: Image without a cap fault

Figure 41: Cropping the cap region

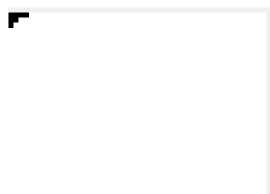

Figure 42: Thresholding and Inverting the cap region

2.3.6 Label is not Straight

In order to detect if the label is straight or not, we used the following to properties combined:

- Hough Transform.
- Correlation.

Detection with Hough Transform: The following steps are performed in order to detect if the label is straight with hough transform:

- The image is cropped around the upper part of the label.
- Horizontal Edge detection is applied in order to detect the horizontal edges of the label.It should be noted that there are strong edges near the start of the label.
- After the edge detection, we try to find the longest straight line that can fit the edge just determined.
- We take the length of that line. We threshold the maximum length of the line that is determined. This threshold is used to differentiate whether the label is straight or not.

The following images illustrate the detection of 'label not straight' fault in case of faulty image with hough transform:

Figure 43: Image with a fault

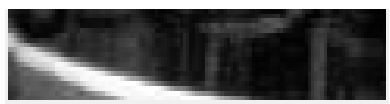

Figure 44: Cropping for Edge Detection

Figure 45: Edge Detection on Cropped Region

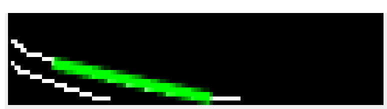

Figure 46: Hough Transform and Longest Line Detected Displayed

The following images illustrate the detection of 'label not straight' fault in case of normal image with hough transform:

Figure 47: Image without a fault

Figure 48: Cropping for Edge Detection

Figure 49: Edge Detection on Cropped Region

Figure 50: Hough Transform and Longest Line Detected Displayed

Detection with Correlation: An additional feature is combined with the hough transform to detect if the label is straight.The following steps are involved:

- Crop the upper part of the label for the middle bottle.
- Apply horizontal edge detection.
- Divide the result in two equal images at the center.
- Flip one of the parts.
- Perform Correlation.
- Take maximum of the result of Correlation.
- Threshold this value to detect if the label is straight.

The following images illustrate the detection of 'label not straight' fault in case of faulty image with correlation:

Figure 51: Image with a fault

Figure 52: Cropping for Edge Detection

Figure 53: Edge Detection on Cropped Region

Figure 54: First Half for correlation

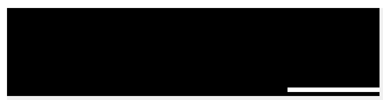

Figure 55: Flipped second half for correlation

The following images illustrate the detection of 'label not straight' fault in case of normal image with correlation:

Figure 56: Image without a fault

Figure 57: Cropping for Edge Detection

Figure 58: Edge Detection on Cropped Region

Figure 59: First Half for correlation

Figure 60: Flipped second half for correlation

2.3.7 Bottle is Deformed

In order to detect deformed bottles, we approached the problem in 3 ways:

- Detecting deformity on the left
- Detecting deformity on the right
- Detecting deformity on the front

Deformity Detection on Left and Right In order detect the deformity on left and right, same steps were repeated. However, the positions of the windows were different. The following steps were performed:

- It was noted that in case of a normal bottle,both the left and the right ends were 60 pixels were the center. So the windows were placed at the end of length 20 pixels. The windows were positioned 20 pixels before the end on either side.

- Only the red channel of the image was used. Since the label is of the red channel.If the bottle is deformed, there will be a dark area.

- The image is thresholded and inverted in order to detect that black area which becomes white after inversion. This area is used to detect deformity on either side.

- In order to get rid of the black areas in the label, image opening was applied.

- After image opening, the percentage of white pixels which correspond to the black area was calculated.

- A threshold was applied in order to classify the faults correctly on either sides.

The following images illustrate the deformity detection on left and right on a faulty image:

Figure 61: Image with a fault

Figure 62: Cropping on Right side

Figure 63: Cropping on Left side

Figure 64: Thresholding and Inversion on Right side

Figure 65: Image Opening on Right side

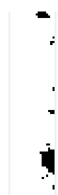

Figure 66: Thresholding and Inversion on Left side

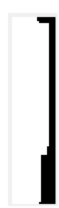

Figure 67: mage Opening on Left side

The following images illustrate the deformity detection on left and right on a normal image:

Figure 68: Image with a normal image

Figure 69: Cropping on Right side

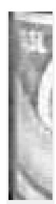

Figure 70: Cropping on Left side

Figure 71: Thresholding and Inversion on Right side

Figure 72: Image Opening on Right side

Figure 73: Thresholding and Inversion on Left side

Figure 74: Image Opening on Left side

Deformity Detection on the Front In order to detect the deformity on the front, we search for edges in the images in the transparent area just below the cap. If the bottle is deformed , there will be a lot of lines in the transparent area. We perform the following steps to achieve this:

- Crop the transparent area just below the cap
- Apply canny edge detection on that area.
- Apply Image opening with a vertical rectangle to get rid of horizontal edges.

- Only keep long connected lines and get rid of small connected components which are present there due to reflection.
- Detect the percentage of these white lines detected.
- Apply a threshold.

The following images illustrate the deformity detection on the front on a faulty image:

Figure 75: Image with a fault

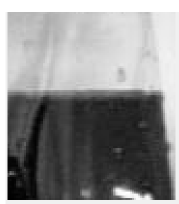

Figure 76: Cropping the region below the cap

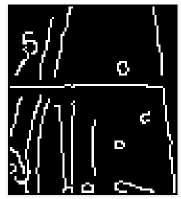

Figure 77: Canny Edge Detection

Figure 78: Image Opening to get rid of horizontal edges

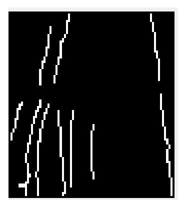

Figure 79: Final Result after keeping long connected components for Thresholding

The following images illustrate the deformity detection on the front on a normal image:

Figure 80: Image without a fault

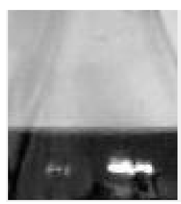

Figure 81: Cropping the region below the cap

Figure 82: Canny Edge Detection

Figure 83: Image Opening to get rid of horizontal edges

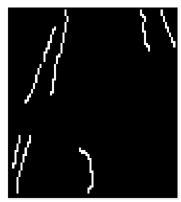

Figure 84: Final Result after keeping long connected components for Thresholding

3 Algorithm for Fault Detection

After all the faults have been individually been detected, we merged all the cases together in order to facilitate the detection of multiple faults at once. The following algorithm was implemented in order to detect all the faults at once:

1. Check if the Middle Bottle is present, if it is present, proceed otherwise display normal status and **abort**.
2. Check if the Bottle is Over Filled.
3. Check if the Bottle is Under Filled.
4. Check if the Bottle's Cap is Missing.
5. Check if the Label is Present. If the label is present, proceed. If the label is not present, display all the previous faults that have occurred and **abort**.
6. Check if the Label is printed. If the label is not printed, display all the previous faults that have occurred and **abort**.
7. Check if the Label is Straight.
8. Check if the Bottle is Deformed.
9. If no fault occured,display Normal. Otherwise, display all the previous faults that occurred.

4 Graphical User Interface

The GUI is created using App Designer. The GUI looks like this:

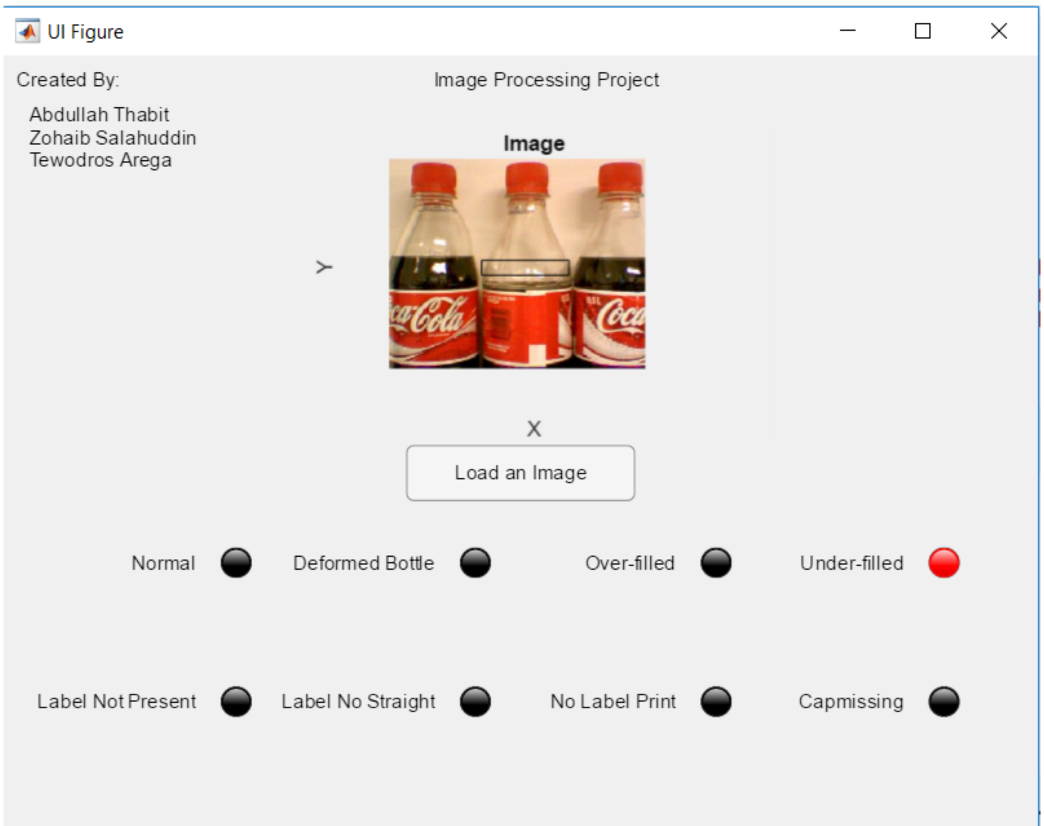

Figure 85: Graphical User Interface for underfilled

The GUI has three components: the first part is the image viewer, the second part is "Load Image" button and the last one is light lamps. To use the GUI, first click on the "Load image" button and select the image that you want to test. The image will be displayed on the image viewer with its fault indicator that indicates the position of the fault.Then the it will show the result using the light lamps. If fault is detected then the corresponding light lamp is will be on(red). Otherwise all light lamps will be off(black). In some cases when a bottle has more than one faults, the corresponding light lamps will be on.

Here we will display the results for each fault:

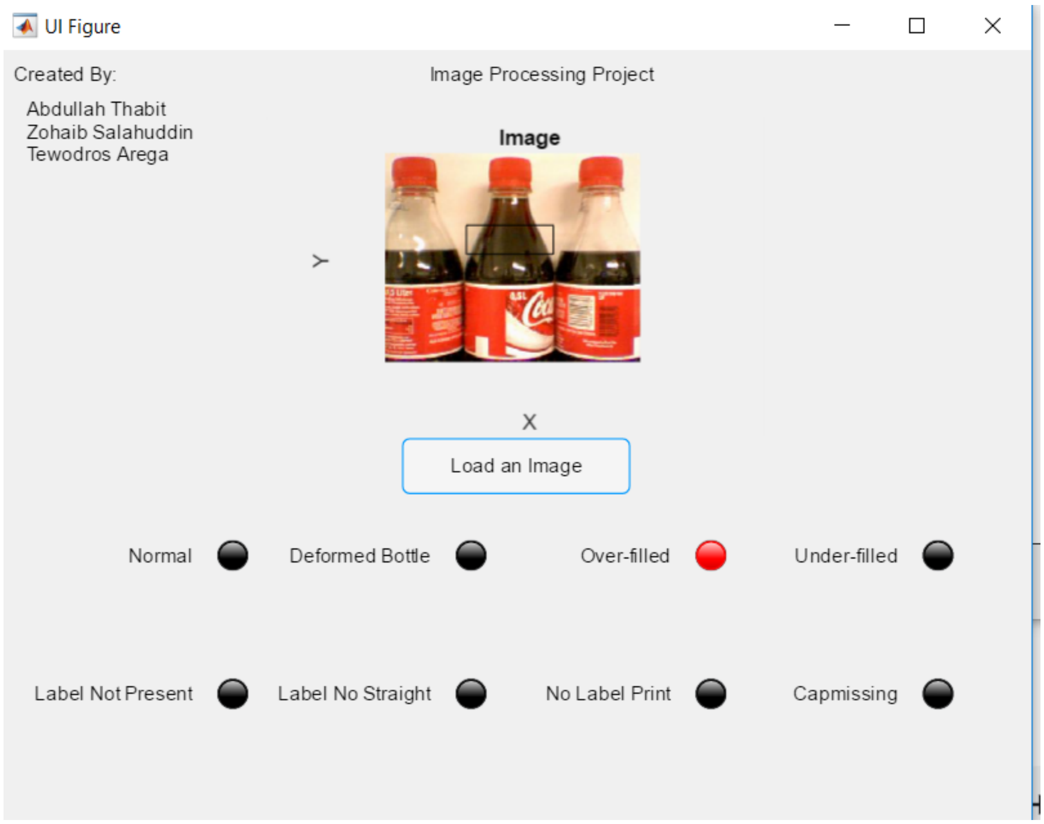

Figure 86: Result for Overfilled

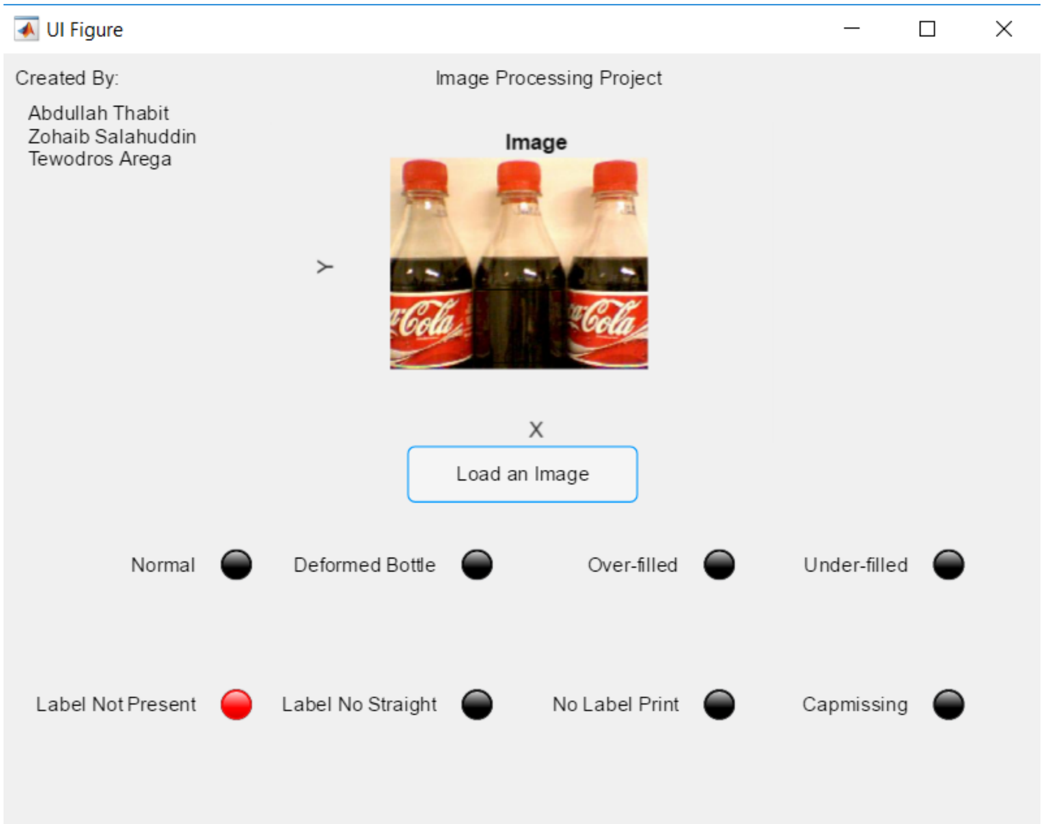

Figure 87: Result for No label

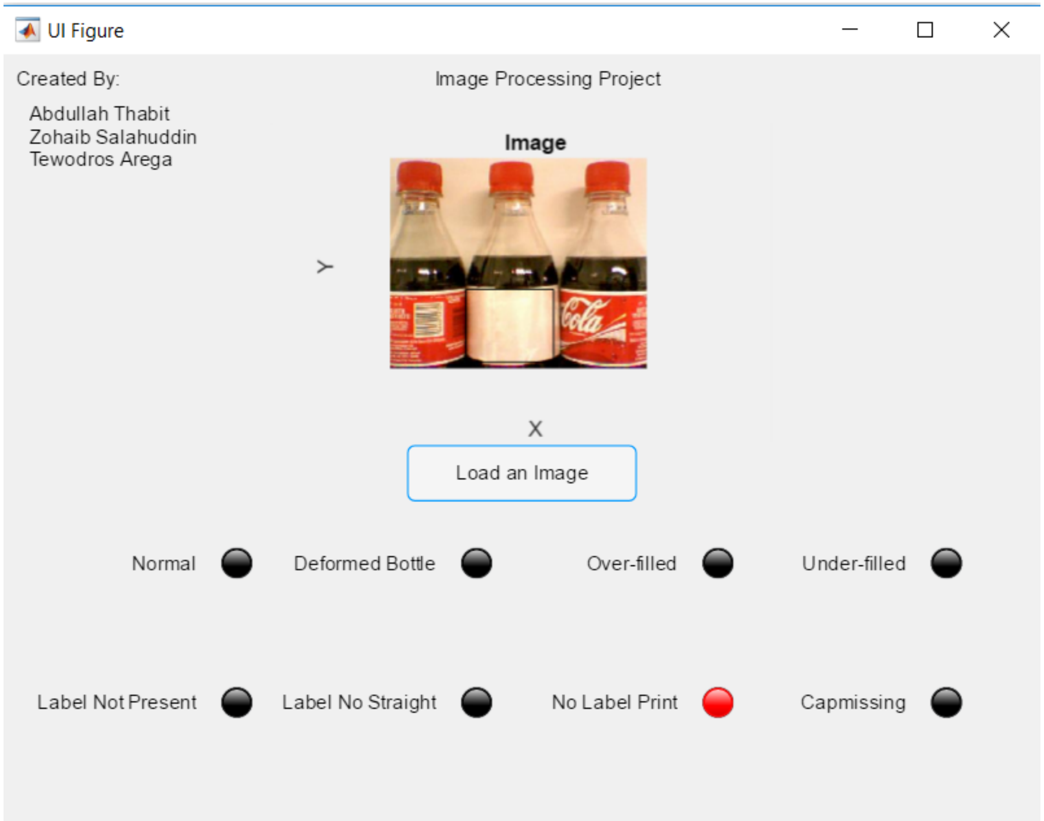

Figure 88: Result for No label printed

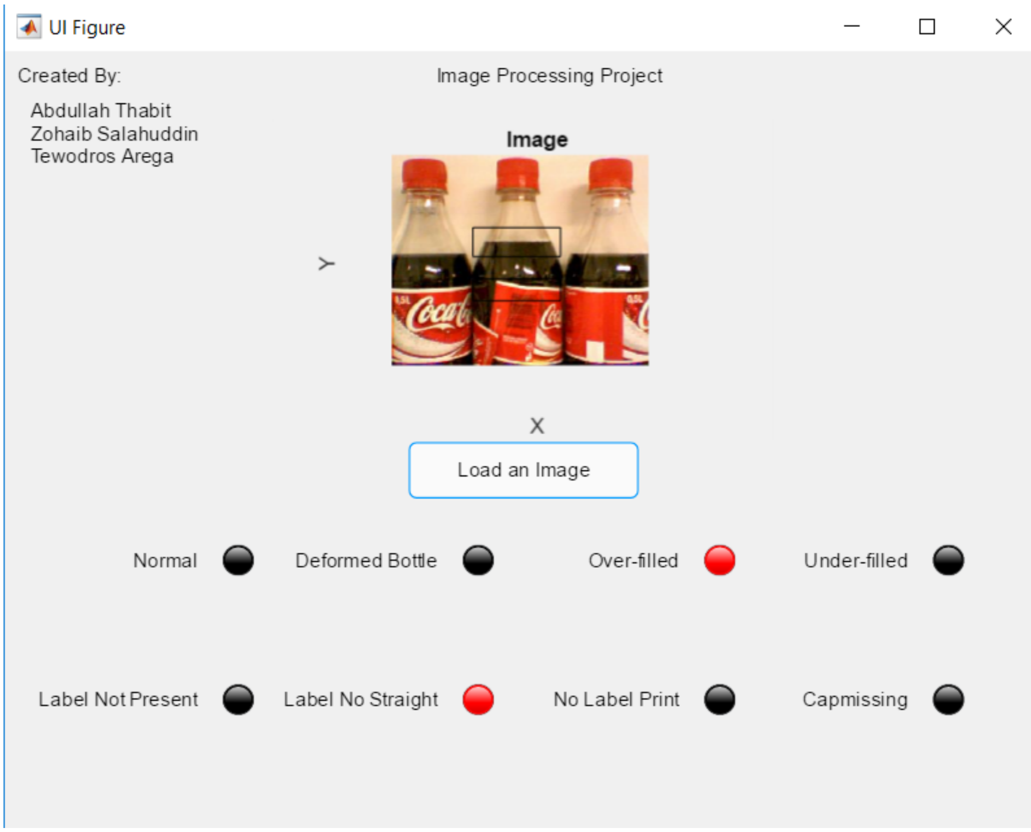

Figure 89: Result for No Straight Label

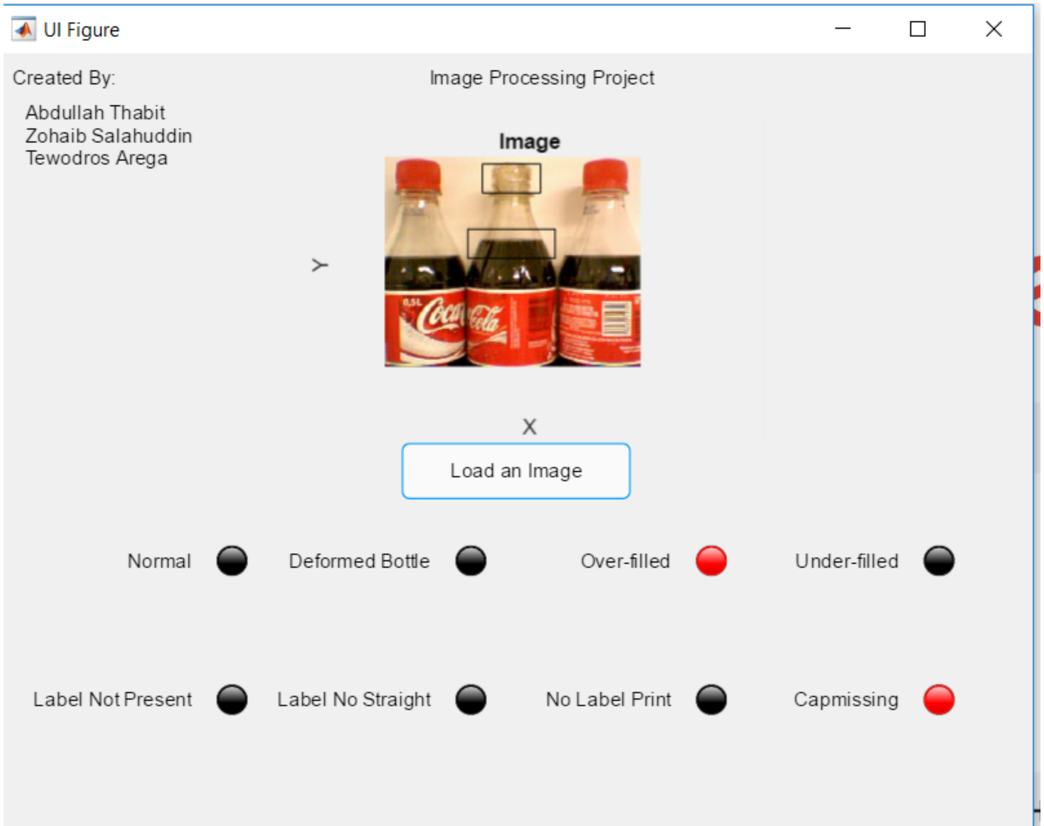

Figure 90: Result for Missing Cap

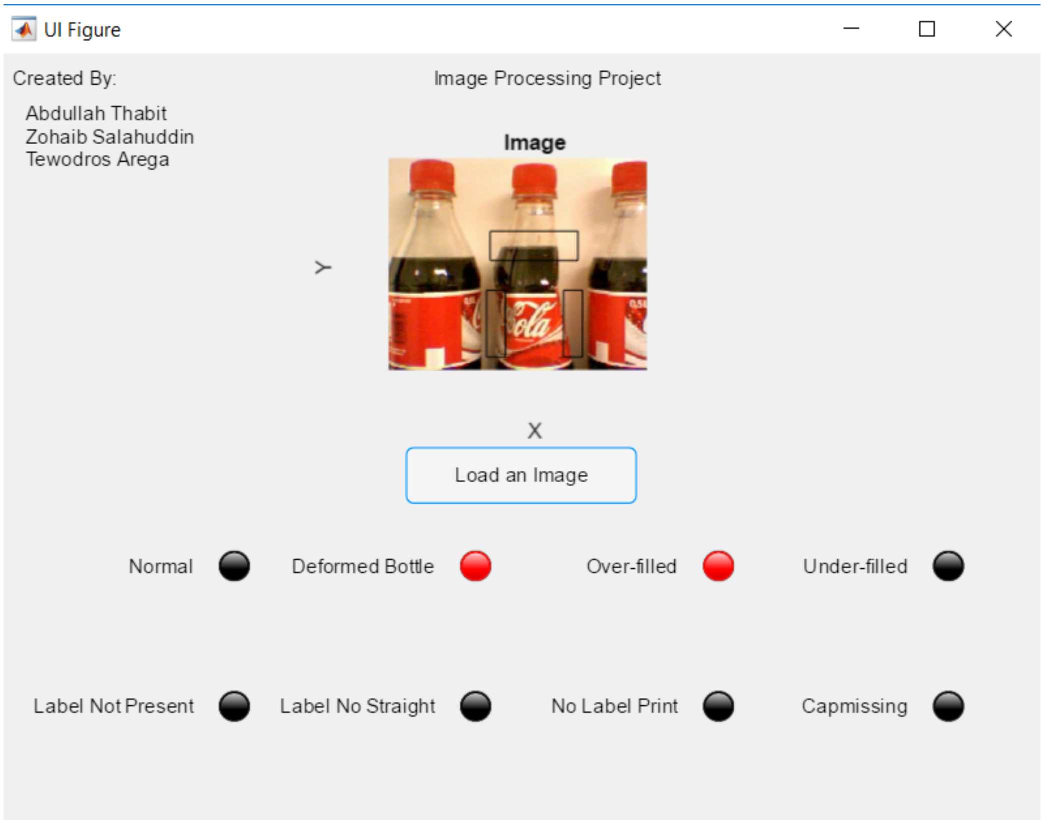

Figure 91: Result for Deformed Bottle

As you can see from the above figures, most of the figures have one fault but some of them have two faults. In figure 89, 90 and 91, the bottles have two faults. In three of them, they have overfilled fault in addition to their corresponding fault.

5 Results

After Detecting all the cases and establishing the algorithm, we tested the algorithm on the training data to test its accuracy. The following results were observed:

Fault type	**Classification Rate**
Middle Bottle Classified as Normal	100 Percent
Cap is Missing	100 Percent
Label is Missing	100 Percent
Label Print is Missing	100 Percent
Label is not Straight	100 Percent
Bottle is Deformed	100 Percent
Bottle is Overfilled	100 Percent
Bottle is Underfilled	100 Percent

Table 1: Results on of the Model on the Training Data

We are able to achieve 100 percent accuracy on the training data.During the training, we tried to keep the algorithm general so that in an attempt to achieve more accuracy, we do not overfit. Keeping that in view, we have still been able to achieve 100 percent accuracy.

6 Key Learning Points

After the implementation of the project, we were able to implement the following concepts that were learnt in class:

- Binarizing through threshold
- Convolution and Correlation
- Histogram Inspection
- Hough Transform
- Image Opening and Closing
- Edge Detection
- Windowing
- Low Pass Filtering

After the detection of the faults in the middle bottle, the faults on the side bottles can also be detected by detecting their center using the same method that was used to detect the center for the middle bottle. Once the center for those bottles has been detected, we can proceed to detect all the parameters in the same way. Since the side bottles do not come in the image as a whole, we can re-construct them by taking there mirror image.

References

[1] C. Solomon, T. Breckon (2011), `Fundamental of Digital Image Processing - A practical Approach with Examples in Matlab. Wiley-Blackwell.`

[2] `https://en.wikipedia.org/wiki/Machine_vision`