Proceedings of the ME 465 2025
Machine Vision and Robotics
MVR Fall 2025
December 13, 2025, University of Hartford

MVR 2025

AUTOMATED MACHINE VISION FOR LIQUID BOTTLE DEFECT DETECTION

Biraj Kakadiya

[1]Robotics Engineering, University of Hartford
West Hartford, CT

ABSTRACT

This paper presents an automated machine vision system designed for the quality control of liquid bottle packaging. The system utilizes digital image processing techniques to detect critical defects, specifically liquid level irregularities. The methodology employs a pipeline consisting of Gaussian smoothing for noise reduction, Canny edge detection for structural definition, and the Hough Transform for geometric feature extraction. The approach addresses the limitations of manual inspection by providing a robust, non-invasive, and efficient solution for high-speed production lines.

NOMENCLATURE

Symbol	Description
I(x,y)	Input image intensity at coordinates x, y
G(x,y)	Gaussian kernel function
σ	Standard deviation of the Gaussian distribution
ρ	Perpendicular distance from the origin to the line in Hough space
θ	Angle of the vector from the origin to the closest point on the line

1. INTRODUCTION

Quality control in the packaging industry is critical, particularly for liquid products where leakage, spoilage, or aesthetic defects can lead to significant financial loss and brand damage. Traditionally, bottle inspection has been performed manually. However, manual inspection is labor-intensive, time-consuming, and prone to errors due to operator fatigue and the subjectivity of human vision. As production speeds increase, often exceeding one meter per second on conveyor belts, human operators cannot maintain consistent inspection quality.

PROBLEM STATEMENT:

a) What is the state of the world

Ismail, Ahmad Najmuddeen, Nur Aidawaty Rafan, and Nor Affendy Norizan. 2025. "Automated Defect Detection in Perfume Bottle Packaging Using Machine Vision Approach for Improved Quality Control." *Multidisciplinary Science Journal* 7 (8). https://doi.org/10.31893/multiscience.2025400.

Kazmi, Majida, Basra Hafeez, Hashim Raza Khan, and Saad Ahmed Qazi. 2022. "Machine-Vision-Based Plastic Bottle Inspection for Quality Assurance †." *Engineering Proceedings* 20 (1). https://doi.org/10.3390/engproc2022020009.

Omid Farhangi, Ehsan Sheidaee, and Asma kisalaei. 2024. "Machine Vision for Detecting Defects in Liquid Bottles: An Industrial Application for Food and Packaging Sector." *Cloud Computing and Data Science*, June, 242–54. https://doi.org/10.37256/ccds.5220244756.

b) YOUR APPLICATION AREA

This research focuses on the Food and Beverage Packaging Sector, specifically for transparent liquid bottles (e.g., water, soft drinks). This area was chosen because liquid bottles present unique challenges—such as refraction, liquid sloshing, and surface reflections—that test the robustness of computer vision algorithms.

c) WHY YOU CHOSE IT

Liquid bottle inspection was selected because it represents a high-impact application where "seal integrity" and "fill level" are safety compliance issues, not just cosmetic ones. The transition from manual to automated inspection in this domain offers measurable improvements in accuracy and consistency.

Ismail et al. (2025) developed a system for perfume bottle inspection using a rotary platform and thresholding techniques. They utilized a black-and-white detection method, determining that a threshold of 0.2 was most suitable for accuracy testing. While they achieved an accuracy of more than 95%, the reliance on simple thresholding can be sensitive to lighting variations.

Kazmi et al. (2022) proposed a framework for plastic bottle inspection using distinct algorithms for different defects, such as Harris Corner detection for caps and Canny edge detection for labels. They argued that deep learning is data-hogging and computationally expensive, making simple image processing more feasible for fast inspection.

Farhangi et al. (2024) focused on liquid level and label placement using edge detection. Their system demonstrated an average accuracy of 95.6%, with specific accuracy for surface level detection at 100%. Their work highlights that determining the distance between the cap and liquid level is a reliable metric for quality control.

d) What is the problem with it?
Despite the advancements presented in the literature, current methodologies face significant challenges when applied to transparent liquid bottles featuring high-contrast labeling. Standard edge detection algorithms, such as the Canny detector utilized by Kazmi et al., depend heavily on intensity gradients; in this specific application, the bottle's silicone label creates a contrast boundary far stronger than the transparent liquid meniscus, causing the algorithm to misidentify the label as the fill level . Furthermore, simple thresholding techniques, like those proposed by Ismail et al., are fragile in industrial environments where lighting shifts or specular reflections on the glass neck can interrupt the edge of the water, leading to discontinuous data that simpler algorithms fail to process . While deep learning offers a potential remedy, its requirement for massive datasets and computational power contradicts the objective of developing a cost-efficient, high-speed inspection system.

e) How do you solve it?
To solve these challenges efficiently, this research employs a geometric approach constrained by a strict Region of Interest (ROI). By programmatically isolating the bottle neck, the system physically excludes high-contrast labels that otherwise confuse standard edge detectors. To handle broken edges caused by lighting glare, the algorithm utilizes the Hough Transform to switch from the spatial domain to the parameter domain, reconstructing a continuous fill line from fragmented edge data. This pipeline, preceded by Gaussian smoothing ($\sigma=1.5$) to reduce surface noise, ensures robust defect detection without the heavy computational requirements of deep learning.

2. METHODS/ALGORITYMS/MATHEMATICAL BACKGROUND/STATE OF ART

The mathematics of the algorithms.

The algorithms employed rely on mathematical convolutions and geometric transformations. The Gaussian distribution in 2D is used for smoothing:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

Following smoothing, the Canny Edge Detector calculates intensity gradients. The gradient magnitude and direction are critical for identifying the "strong" edges of the liquid meniscus

$$|\nabla I| = \sqrt{G_x^2 + G_y^2} \tag{2}$$

2.1 DOMAIN CHANGE

This section discusses how the image domain is transformed to parameter space. The Hough Transform converts the problem of finding collinear points in the image space (x,y) to finding intersecting curves in a parameter space (ρ,θ). A line in the image space is represented by the equation:

$$\rho = x\cos\theta + y\sin\theta \tag{3}$$

In this domain change, every edge point (x,y)votes for sinusoidal curves in the (ρ,θ)accumulator space. Peaks in this accumulator correspond to strong lines in the original image.

2.2 Utilizing the equations as object descriptors

By transforming the edge data into Hough space, we can describe the liquid level not as a collection of pixels, but as a geometric entity defined by ρ (height) and θ (angle). This makes the detection robust against broken edges caused by glare or label interference, as the "voting" process fills in the gaps.

2.3 Algorithm Steps

A. Image Acquisition & Preprocessingrgb2gray:

rgb2gray: Converts the color image to grayscale, as edge detection relies on light intensity changes, not color.

imgaussfilt(..., sigma): Applies Gaussian Smoothing with σ =1.5. This blurs the image slightly to remove "noise" (like small bubbles, glass texture, or dust) so the edge detector focuses only on strong lines.

```
% 1. Image Acquisition
rawImg = imread(filename);

% 2. Preprocessing
if size(rawImg, 3) == 3
   grayImg = rgb2gray(rawImg); % Convert RGB to Grayscale
else
   grayImg = rawImg;
end

% Gaussian smoothing to remove noise (Eq. 1)
sigma = 1.5;
smoothImg = imgaussfilt(grayImg, sigma);
```

B. Canny Edge Detection

edge(..., 'Canny', [0.03 0.10]): Finds the edges in the image.

It looks for gradients (sharp changes in brightness).

The numbers [0.03 0.10] are thresholds. Only edges stronger than these values are kept. This helps ignore faint edges while keeping the distinct liquid surface line.

```
% 3. Canny Edge Detection (Eq. 2)
% Thresholds [0.03 0.10] for hysteresis
edgeImg = edge(smoothImg, 'Canny', [0.03 0.10]);
```

C. Hough Transform (Finding Lines)
hough / houghpeaks: This math converts the pixel edges into a "voting space" to find mathematical lines.

houghlines: Reconstructs the strongest line segments from that voting space.

```
% 4. Hough Transform (Standard)
[H, T, R] = hough(edgeImg);

% Find peaks in Hough accumulator space
P = houghpeaks(H, 10, 'threshold', ceil(0.3*max(H(:))));

% Link peaks into lines
lines = houghlines(edgeImg, T, R, P, 'FillGap', 30, 'MinLength', 60);
```

D. Feature Analysis (Isolating the Liquid)

if abs(lines(k).theta) > 85 ...: This filter is crucial.The Hough transform finds all lines (vertical sides of the bottle, label edges, etc.).This if statement ignores everything except horizontal lines (angles near 90°)

detectedLevelY = mean(xy(:,2)): Once the horizontal liquid line is found, this calculates its exact Y-position (height) in the image.

```
% Loop through lines
for k = 1:length(lines)
  xy = [lines(k).point1; lines(k).point2];

  % Filter for horizontal lines (theta approx 90 degrees)
  if abs(lines(k).theta) > 85 && abs(lines(k).theta) < 95

    % Calculate the average Y-coordinate (height)
    currentY = mean(xy(:,2));

    detectedLevelY = currentY;
    levelFound = true;

    % Draw the line in green
    plot(xy(:,1), xy(:,2), 'LineWidth', 3, 'Color', 'green');
      break; % Stop after finding the first strong horizontal
line
  end
end
```

E. Decision Logic

(The"Brain")deviation=abs(detectedLevelYCALIBRATED_LEVEL_Y): Calculates the error—how far off is the actual liquid level from the ideal level?

Pass/Fail Check:

If the error is small (<15 px), the code marks it PASS (Green Box).

If the error is large (> 15 px), the code marks it FAIL (Red Box).

```
% 6. Decision Logic
if levelFound
  % Calculate how far off the liquid is
  deviation = abs(detectedLevelY - CALIBRATED_LEVEL_Y);

  if deviation <= TOLERANCE_PX
    message = 'RESULT: PASS (OK)';
    boxColor = 'green';
  else
    message = sprintf('RESULT: FAIL (Deviation: %.0f px)', deviation);
    boxColor = 'red';
  end
end
```

3. RESULTS AND DISCUSSION

As shown in Figure 1 below, the raw input captures the "good" bottle. This reference image displays an intact structure with the liquid level at the correct calibration height. Figure 2 represents the "defective" bottle, which is characterized by significant physical deformation and a peeling label.

Figure 1: Good Bottle

Figure 2: Defective Bottle

Figure 3: Result

Figure 3 presents the computer vision processing pipeline, specifically the Canny Edge Detection and Hough Liquid Detection results. The output of the code provides a visual and textual status for each bottle.

Test Case 1 (Good Bottle):The algorithm successfully processed the reference image. The system detected a horizontal line at a Y-coordinate close to the calibration value, with a measured deviation of $\le$ 15px. Because the liquid level fell within the acceptable tolerance range, the output result was "PASS”.

Test Case 2 (Defective Bottle):For the damaged sample, the edge detection revealed irregular contours. The system either detected a line significantly offset from the calibration value (Deviation >15px) or failed to find a coherent line entirely due to the bottle damage (as seen in smashed samples). Consequently, the system correctly flagged the item, and the output result was "FAIL".

4. CONCLUSION

I learned that preprocessing (Gaussian smoothing) is essential for reducing glass glare, and that the Hough Transform ("Domain Change") allows for robust detection of geometric features even when edges are partially broken.

SUGGESTION FOR NEXT SEMESTER:

If I were to spend another semester on this project, I would implement Pattern Matching for cap inspection as suggested by Farhangi et al..

WHAT WOULD YOU DO DIFFERENTLY:

If I started again, I would use Adaptive Thresholding instead of fixed Canny thresholds to handle variable lighting conditions in the factory environment.

ACKNOWLEDGEMENTS

I would like to acknowledge the research provided by Ismail et al., Kazmi et al., and Farhangi et al. for their valuable insights into defect detection methodologies.

REFERENCES

Ismail, Ahmad Najmuddeen, Nur Aidawaty Rafan, and Nor Affendy Norizan. 2025. “Automated Defect Detection in Perfume Bottle Packaging Using Machine Vision Approach for Improved Quality Control.” Multidisciplinary Science Journal 7 (8). https://doi.org/10.31893/multiscience.2025400.

Kazmi, Majida, Basra Hafeez, Hashim Raza Khan, and Saad Ahmed Qazi. 2022. “Machine-Vision-Based Plastic Bottle Inspection for Quality Assurance †.” Engineering Proceedings 20 (1). https://doi.org/10.3390/engproc2022020009.

Omid Farhangi, Ehsan Sheidaee, and Asma kisalaei. 2024. “Machine Vision for Detecting Defects in Liquid Bottles: An Industrial Application for Food and Packaging Sector.” Cloud Computing and Data Science, June, 242–54. https://doi.org/10.37256/ccds.5220244756.