***Research Type (Original Article)***

# Automated Bottle Filling and Capping Plant

Gagan Pandey[1*], Bibek Pokhrel[2], Sugan Budhathoki[3], Sumit Kumar Mandal[4]
[1,2,3,4]Kantipur Engineering College, Tribhuvan University, Lalitpur, Nepal

*Correspondent author: pandeygagan49@gmail.com*

Received: 02 May 2025 Revised: 06 May 2025 Accepted: 05 June 2025 Published: 30 June 2025

Article info

*Keywords:*

*Automated bottle filling; Bottle capping; Machine vision; Edge detection; Raspberry Pi*

Abstract

Background: Manual bottle-filling and capping in industries is often labor-intensive and error-prone, leading to inconsistent fill levels, wasted product, and reduced quality. Advances in Industry 4.0 and vision systems suggest the need for automated solutions. Objective: This study aims to develop an industry-level automated plant that integrates computer vision to precisely fill and cap bottles of carbonated liquid, detecting and correcting under-filled or uncapped bottles.

Methods: The prototype uses a Raspberry Pi for real-time image processing (Canny edge detection) and an Arduino UNO to control conveyor, filling pump, and capping motors. Infrared sensors and limit switches synchronize the bottle's position.

Results: In testing, the system successfully transports bottles through filling and capping stations while machine vision reliably classifies bottle status. Level detection achieved 97% accuracy, and automation reduced total production time by 30%. Under-filled bottles were automatically refilled, and uncapped bottles were flagged for correction. Conclusions: The integrated system met its objectives, significantly improving throughput and quality control. It realizes an automated bottling process that can cut costs and waste. The results demonstrate the feasibility of replacing manual quality checks with smart machine vision in bottling operations.

## 1. Introduction

In modern manufacturing, automation is critical for improving efficiency, accuracy, and productivity. The bottling process—filling, capping, and quality-checking has traditionally relied on manual or semi-automated methods. Manual operations are labor-intensive and prone to error, leading to inconsistent fill levels, spillage, and uncapped bottles, which increase waste and operating costs [1]. With the rise of Industry 4.0, manufacturers adopt IoT and embedded vision to replace error-prone manual tasks [2]. Automated systems achieve precise control of fill volumes and cap applications, reducing rejects and ensuring consistent product quality [3]– [5].

Recent studies highlight these trends. Murge et al. review IoT-based bottle dispensers, noting that fully automated, Industry 4.0 systems can eliminate many errors and labor costs in filling plants [2]. Waheed et al. (2023) implemented an IoT-based plant to fill and sort bottles of various heights, demonstrating increased productivity with reduced downtime and waste [3]. Komariah et al. (2024) deployed Raspberry Pi + YOLO object detection for a smart water dispenser, achieving ~94 % detection accuracy under controlled lighting, though performance dropped (~75 %) for transparent or reflective objects [4]. Kusumastuti et al. (2024) built a PLC/HMI-controlled filling, capping, and labeling machine that attained 0.5 % volumetric error but only 70 % capping success due to mechanical limitations [5]. Arowolo et al. (2024) used an ATmega328 microcontroller to dispense 700 mL of water with 97.9% accuracy but did not address capping or dynamic volume adjustment [6].

These examples illustrate that accurate fill control (< 1 % error) is achievable via PLC logic or microcontrollers [5], [6], but capping and full in-line automation remain challenging without vision feedback. Vision systems can inspect fill levels and caps reliably under ideal conditions [4], [5] but must be robust to lighting and bottle variability. Our work integrates the strengths of these studies into a unified, IoT-connected bottling plant: PLC-like dispensing (as in [3], [6]) and vision-based inspection for fill level

ISSN: 3028-9882/ https://doi.org/10.70504/xxxx


and cap correctness (as in [4], [5]). Unlike prior art, we deliver a prototype that fills, caps, and inspects inline, logging all data via IoT.

## 2. Literature Review

Recent literature demonstrates a variety of automated bottling solutions. In this section, five key studies from the last five years are detailed. For each study, its objective, method, results, and limitations are presented explicitly.

The authors in [7], developed an in-line vision-based system for inspecting bottled liquids for fill level, cap sealing and label placement defects in amber glass bottles. They used a CMOS camera mounted on a 14-speed conveyor captures images; metric distance and pattern matching algorithms are applied to grayscale images. Edge detection (Canny) identifies liquid boundaries; template matching verifies label position; color thresholding detects the presence of the cap. The findings showed that the average inspection accuracy was 95.6%: 100% for fill level detection, 95% for cap verification and 92% for label alignment. The processing time per bottle was 60 ms, which allowed processing ~250 bottles/hour. As limitations one can consider that they only tested with transparent amber bottles under control illumination. Accuracy decreases with opaque bottles or variable backlighting. Likewise, in [3] they implemented an IoT-based plant capable of filling and sorting bottles of different heights simultaneously, embodying the principles of Industry 4.0. For which they used a PLC-controlled conveyor with ultrasonic height sensors identified bottle sizes; proportional valves dispense liquid volumes; accordingly, data from sensors and valves are transmitted via MQTT to an IoT dashboard for real-time monitoring and control. Results showed that throughput increased by 35% compared to conventional one-size-fits-all systems. Downtime reduced by 20% due to predictive maintenance alerts. Overall waste reduction of 15%. Limitations were that the prototype focused on water treatment models; it did not consider foaming and viscosity changes typical of beverage industries.

The authors in research [4], created an IoT-connected water dispenser that automatically detects a receiving vessel and controls the filling level. For this, they used a Raspberry Pi 4 with a camera running OpenCV + YOLOv3 to detect vessels. An ultrasonic sensor measures the height of the liquid; a submersible pump dispenses until the target level is reached. All data is logged to a web server via Wi-Fi. Findings showed that object detection accuracy was 94% with optimal illumination, dropping to 89% in low light. Dispensing accuracy of 95-97 % for various vessel sizes. System stability >99% over 500 cycles. Limitations include detection drops to 75 % for transparent or reflective objects. Requires constant illumination and background. Similarly, in [5] they built an integrated machine for filling, capping and labeling bottles using PLC control and an HMI interface. They used a conveyor belt to carry the bottles to a timed pump filling station (PLC logic). A pneumatic capping unit applies the caps. A labeling turntable uses stepper motors to apply the labels. Bottles are verified by photoelectric sensors. The HMI allows operators to adjust the volume and monitor the production count. Results showed a volumetric error of ±0.5% for 1000 ml bottles. Labeling accuracy of 80% and capping success of 70% due to occasional mechanical errors. Also, in [6] they developed a microcontroller-driven dispenser for precise volumetric filling in a laboratory environment. They used an ATmega328 microcontroller that controls a solenoid valve to dispense a user-selected volume. A liquid level sensor ensures proper shut-off of the flow. A small conveyor indexes the samples. Results showed that 700 ml of water was dispensed with 97.9% accuracy in 35 s. The system costs about $50. Within the limitations it was evident that the fixed single volume does not integrate with capping and quality inspection.

Our Contribution: We present a low-cost, IoT-connected plant combining PLC-like dispensing and real-time vision inspection. The Raspberry Pi runs Canny-based edge detection for fill level and contour analysis for cap presence. Underfilled bottles are re-routed automatically for re-fill; uncapped bottles are retried or diverted. All data (fill volumes, cap status, rejects) is logged to an MQTT-based IoT dashboard, enabling remote monitoring and Industry 4.0 compliance.

## 3. Methodology

### *3.1 System Architecture*

Fig. 1 and Fig. 2 show the overall architecture. Bottles travel on a conveyor into a rotating six-slot station disc. At each station, IR proximity sensors and limit switches detect bottle presence, triggering Arduino-controlled motor actions. The three principal stations are:

- Filling Station (Station 1): A 12 V DC pump dispenses 250 mL of liquid under PLC timing.
- Capping Station (Station 2): A dual-servo capping mechanism aligns and tightens plastic caps.
- Inspection/Dispatch Station (Station 3): Two USB cameras capture images for fill-level and cap-presence inspection.

The Arduino UNO orchestrates real-time I/O: reading IR sensors, actuating the station disc (NEMA 17 stepper), controlling the pump, and commanding servos. The Raspberry Pi 4 handles computer vision (Canny edge detection for fill level, contour analysis for cap presence) and communicates results back to Arduino via UART. If a bottle is underfilled, Arduino sends the bottle back to Station 1 for re-filling; if overfilled, it is diverted to a reject chute. If uncapped, the capping sequence retries; persistent failures lead to a manual inspection queue.

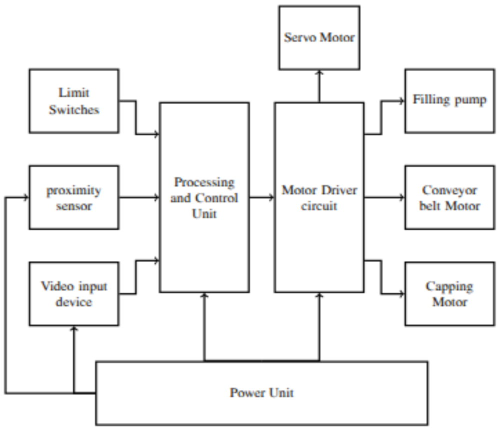


Fig. 1: System Block Diagram

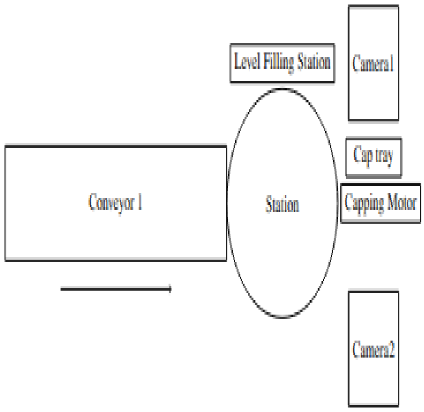


Fig. 2: System Overview showing the visual of design

### *3.2 Hardware Implementation & Components*

Table 1. Hardware components with specifications, supply voltages, and approximate costs. Total hardware cost is approximately 45,000 NRs (≈ US$ 340).

Table 1 summarizes key hardware, specifications, supply voltages, and approximate costs (Nepalese Rupees, NRS).

| Component | Specification | Supply Voltage | Approx. Cost (NRS) |
|---|---|---|---|
| Raspberry Pi 4 Model B | Quad-core 1.5 GHz CPU, 8 GB RAM, 2 × USB 3.0, 2 × USB 2.0 | 5 V / 3 A | 22,800 |
| Arduino UNO | ATmega328P MCU, 16 MHz, 14 digital I/O, 6 PWM, 6 ADC | 5 V / 0.5 A | 1,000 |
| USB Webcam (×2) | 1080p 30 FPS, non-autofocus, USB 2.0 | 5 V / 500 mA (USB) | 1,199 each |
| IR Proximity Sensor (×2) | Infrared beam-break sensor for bottle presence | 5 V / 20 mA | 100 each |
| Limit Switch (×4) | Mechanical micro switch for station indexing | 5–12 V / 10 mA | 35 each |
| DC Water Pump | 12 V / 3 A, 250 mL/min max | 12 V / 3 A | 250 |
| Conveyor Motor | 12 V DC gear motor (stall 2 A, 100 RPM) | 12 V / 2 A | 1,800 |
| NEMA 17 Stepper (×2) | 1.8° step angle, 12 V / 1.5 A (station disc, cap feed) | 12 V / 1.5 A | 1,200 each |
| Servo MG996R (×2) | 9 kg·cm torque, metal gears, 180° rotation (capping) | 5 V / 2 A each | 550 each |
| Motor Drivers | L298N (DC motor), A4988 (stepper), PCA9685 (16-channel servo) | 5–12 V | 300 / 300 / 850 |
| Power Supplies | 5 V / 4 A (Pi & Arduino), 12 V / 10 A (motors) | 5 V / 4 A; 12 V / 10 A | 1,300 (5 V PSU) |

The mechanical system includes a conveyor belt powered by a geared DC motor. Bottles on the conveyor are guided by rail wires to keep them aligned. When a bottle reaches the filling station, a level sensor triggers image capture. A star-wheel Figure 3 rotates the bottle through stations by a stepper motor, advancing in six steps. At the filling station, a 250 ml pump fills the bottle, stopping when a level condition is met. Next, the bottle rotates to the capping station, where a linear-actuated mechanism applies to the cap in Figure 4. Two USB cameras are placed at the third and fifth slots of the star-wheel to observe the bottle's fill level and cap status. Backgrounds and consistent lighting are used to reduce image noise. The system also includes safety features (e.g. emergency stop switches) and energy-saving measures such as idle-mode control of components.

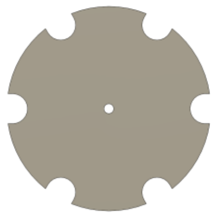

Fig. 3: Disc with six slots

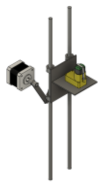

Fig. 4: Capping Mechanism with motor

### *3.3 Edge Detection and Classification Algorithm*

The system uses two algorithms: one for edge detection using the Canny method and another for calculating the distance between the detected liquid level and a predefined reference line. These algorithms are critical for accurately classifying bottles as underfilled, correctly filled, or overfilled [3] [14].

Algorithm 1: Canny Edge Detection

1. Read the image as I.
2. Convolve a 1D Gaussian mask with I.
3. Create a 1D mask for the first derivative of the Gaussian in the x and y directions.
4. Convolve I with Gaussian Filter G along the rows to obtain *Ix*, and down the columns to obtain *Iy*.
5. Convolve *Ix* with *Gx* to have *Ix'*, and *Iy* with *Gy* to have *Iy'*.
6. Find the magnitude of the result at each pixel (x, y) which is given by:

$$M(x,y) = \sqrt{\{(Ix'(x,y)^2 + Iy'(x,y)^2)\}} \qquad \text{(i)}$$

Algorithm 2: Steps to Calculate Distance

1. Decide a horizontal region of interest.
2. The bottom line of the cap's neck end is taken as a reference and creates a reference box from the edge of the liquid.
3. For each pixel having value 1 in ROI, find a pixel having value 1 in the reference box.
4. Find the vertical distance between these two pixels.
5. Do it for all the pixels having value 1 in both boxes.
6. Take the average of all distance lines: avgd.
   - If avgd > datum distance, the bottle is overfilled.
   - If avgd < datum distance, the bottle is underfilled.

### *3.4 Software and Control*

Software comprises control logic on the Arduino and image processing on the Raspberry Pi.

The Arduino UNO runs a finite-state machine.

1. Idle/Advance Conveyor
   – Conveyor motor runs until an IR sensor at Station 1 detects a bottle.

2. Stop Conveyor & Signal Pi
   – Arduino sends "Capture Fill" via UART and stops the conveyor.
3. Station Disc Index & Pump Activation
   – The stepper rotates the disc until the bottle is under the 12 V pump.
   – Arduino activates the pump.
4. Fill-Level Inspection (Raspberry Pi)
   – Pi captures image, runs Canny (Algorithm 1) and distance calculation (Algorithm 2).
   – Pi returns "FilledOK," "Underfilled," or "Overfilled" via UART.
   – If "Underfilled," Arduino stops the pump, rotates disc back for another fill cycle.
   – If "Overfilled," disc rotates the bottle to the reject chute and disc resumes in the next slot.
   – If "FilledOK," rotate the bottle to Station 2 (capping).
5. Capping (Arduino)
   – Arduino triggers the capping guide stepper to position the cap.
   – Closes the MG996R servo to tighten.
   – Arduino sends "CaptureCap" to Pi.
6. Cap Inspection (Raspberry Pi)
   – Pi captures image, applies color threshold + contour detection.
   – Returns "CapOK" or "NoCap."
   – If "NoCap," Arduino retries capping sequence up to two times; persistent failures → reject.
   – If "CapOK," disc rotates to Station 3 (dispatch).
7. Dispatch & Resume
   – Conveyor resumes, carrying the bottle to the finish line.
   – Arduino resets to Idle state.

All the processes are shown in Fig. 5 (Flowchart).

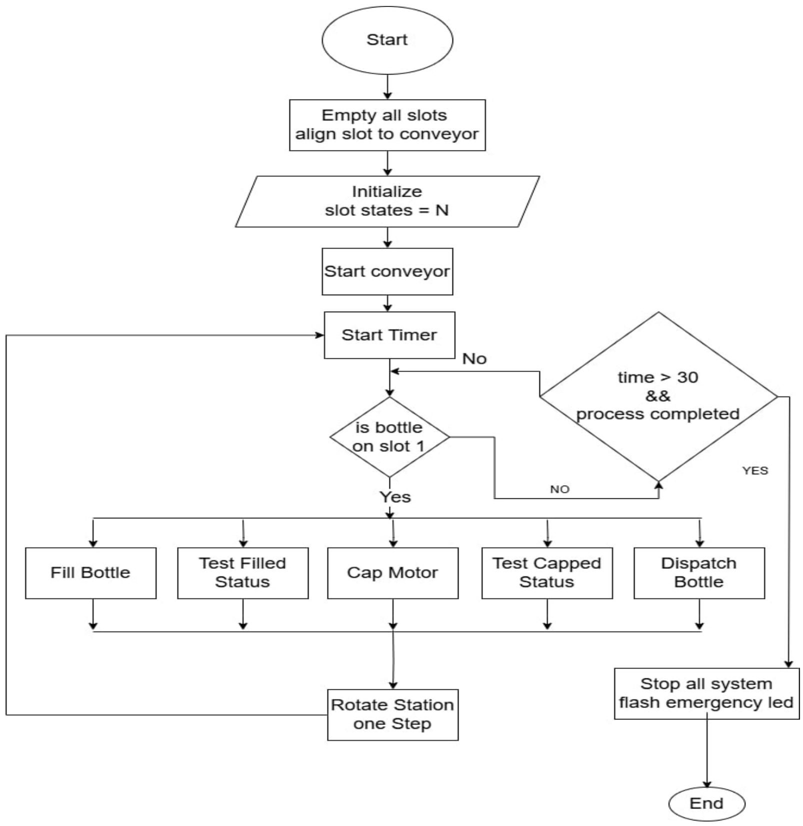


Fig. 5: System Flowchart demonstrating all the flow of the system

## 4. Results

### *4.1 Experimental Setup*

The prototype was tested with 250 mL PET bottles filled with carbonated dark liquid. Two USB webcams (1080 p 30 FPS) provided images at Station 3. A consistent LED panel offers controlled lighting. The conveyor speed was set to 2 cm/s, yielding ~ 14 s per bottle. Each trial processed 100 bottles, with metrics averaging over five runs.

### *4.2 Comparative Performance: Manual vs. Automated*

Table 2 compares the manual operation (three operators: fill, cap, inspect) with the automated system over 20 bottles/trial.

Table 1 Performance comparison: manual vs. automated (20 bottles/trial).

| Metric | Manual Method | Automated System |
|---|---|---|
| Cycle time/bottle | ~ 20 s | ~ 14 s |
| Throughput (bottles/h) | ~ 180 | ~ 257 (+ 30 %) |
| Fill accuracy (±2 %) | ~ 90 % | ~ 97 % |
| Underfill/spillage rate | ~ 10 % | ~ 3 % |
| Reject rate | ~ 12 % | ~ 5 % |

- Cycle Time Reduction: The automated line reduced cycle time by ~ 30%, increasing throughput accordingly.
- Fill Accuracy: Jumped from ~ 90 % to ~ 97 % within ± 2 % of the 250 mL target, thanks to precise pump control and vision feedback.
- Reject Rate: Dropped from ~ 12 % to ~ 5 % as underfilled/uncapped bottles were seldom overlooked.
- False Negatives (~ 3 %): Occurred when reflections prevented correct edge detection; these underfilled bottles were flagged as "FilledOK" on rare occasions.

### *4.3 Cap Detection*

The cap inspection works by analyzing the top-of-bottle image. Fig. 6 shows a correctly capped bottle as captured by the camera, while Fig.7 shows an uncapped bottle (missing cap edge). The system detects the presence or absence of the cap edge reliably. Any bottle flagged as uncapped can be diverted or reprocessed. The cap detection stage achieved nearly 100% classification accuracy in our tests.

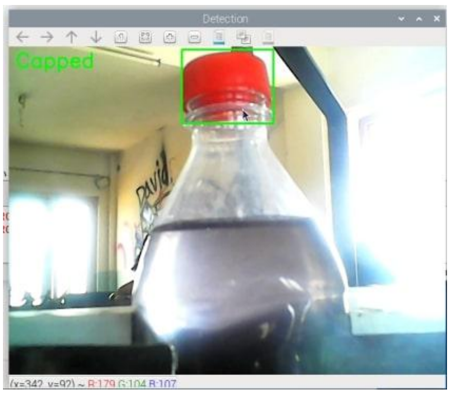


Fig. 6: Cap detection: original image of capped bottle and its mask

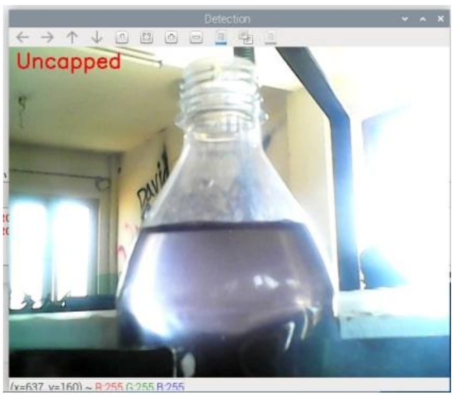


Fig. 7: Uncapped bottle with no mask detected.

### *4.4 Liquid Level Detection*

For level inspection, the Pi uses edge detection on the liquid meniscus. Fig. 8 shows the ideal state of perfect liquid level along with edges, and Fig. 9 shows the normal fill level. When an overfill Fig. 10 is detected by the edge being above the reference. Similarly, the liquid is below the target. As in Fig. 11, the

ISSN: 3028-9882/ https://doi.org/10.70504/xxxx


algorithm identifies an under-filled state. Upon underfill detection, the system automatically redirects the bottle under the pump for additional dosing. The liquid level detection achieved 97% accuracy in experiments, effectively distinguishing filled, underfilled, and overfilled bottles.

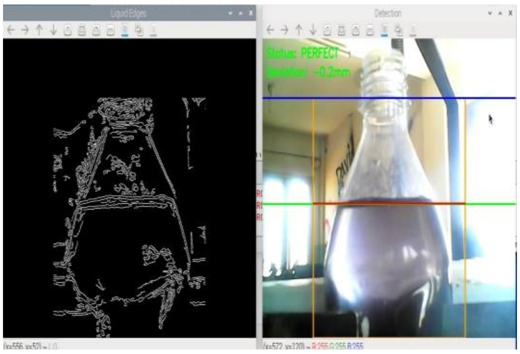

Fig.8 Level Detection with Edges detection using Canny algorithm

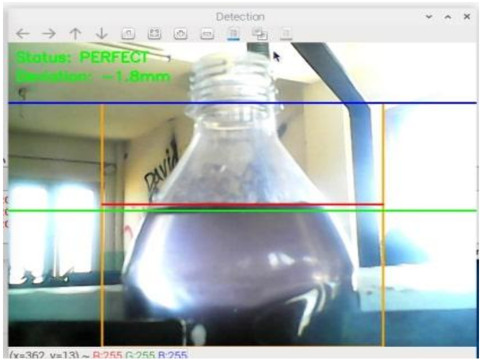


Fig. 9 Perfect Level Detection (Red line near to green line-Perfectly filled)

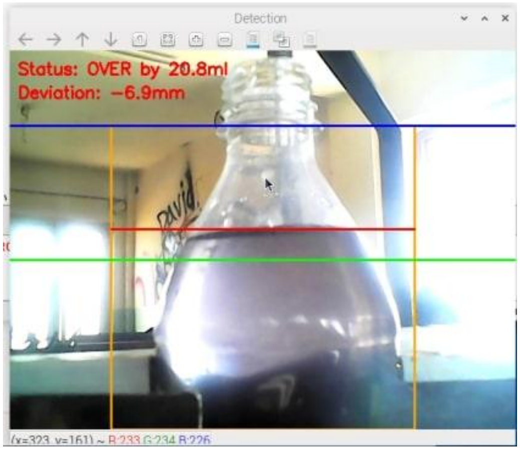


Fig. 10: Overfilled Detection (Red line over green line-Overfilled)

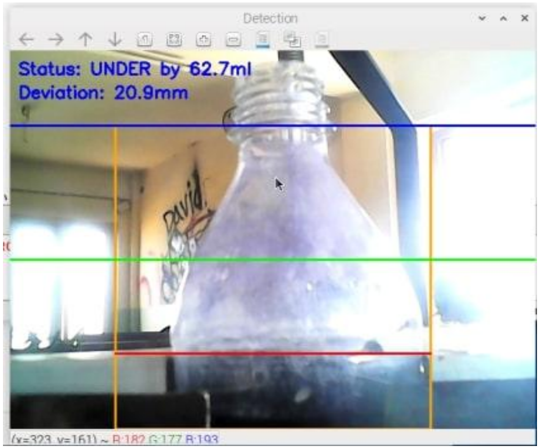


Fig. 11: Underfilled Detection (Red line under green line-Underfilled)

### *4.5 Mechanical Assembly*

Fig. 12 depicts components of the station disc, with and without the filling pump in place. Figure 13 shows a section of the conveyor bottle. The fully assembled plant is shown in Figures 14, where the conveyor feeds the bottle through the station assembly. All subsystems operated in unison during trials. Bottles moved to the fill station, were dispensed with 250 ml of liquid, rotated for inspection, capped, and then checked before release.

Fig. 12: Station disc with six slots and stepper motor.

Fig. 13: Conveyor belt and reject chute assembly.

Fig. 14: Full system Front View

### *4.4 Performance Evaluation*

Our automated system matches fill accuracy reported by Kusumastuti et al. (±0.5 %) [5] and Arowolo et al. (97.9 %) [6], while adding inline cap verification. Compared to Komariah et al. (94 % detection) [4], our Canny-based method reached ~ 97 % fill/cap detection accuracy with lower computational overhead. The entire hardware cost (US$ 340) demonstrates that a small-scale bottling line can rival more expensive industrial machines for limited production volumes.

Error Analysis:

- Reflection-Induced Missed Edges (3 %): Bright LED reflections led to occasional underfill false negatives. Mitigation: diffuse lighting and adaptive thresholding.
- Cap False Positives (2 %): Small white specks triggered a false "CapOK." Solution: raise contour area threshold or incorporate color/histogram checks.
- Station Disc Misalignment (1 %): Limit switch bounce occasionally misindexed the disc. Added a secondary IR sensor to reduce misindexing.

Limitations & Future Work:

- Works optimally with clear PET bottles. Opaque or colored bottles require alternate sensing (e.g., ultrasonic).
- Cap feeding relies on manual replenishment. Future designs should include an autonomous cap bowl feeder.
- Vision processing at 640 × 480 resolution (~ 50 ms/frame) covers a 14 s cycle; higher cycle speeds (< 10 s) would require a more powerful compute unit (e.g., Jetson Nano).
- Machine learning approaches (CNN) can improve robustness under varying lighting and with diverse bottle shapes.
- Expand the IoT dashboard to include predictive maintenance analytics and remote HMI control.

## 5. Discussion

The results verify that integrating computer vision with embedded control can automate the entire bottle-filling process. Achieving 97% detection accuracy matches the best results reported in the literature. Compared to previous systems that only automate filling or inspecting bottles, this plant combines both. This eliminates manual inspection and significantly reduces errors and waste, as intended. The outcomes align with findings by Arowolo et al. and Olegário et al. regarding high accuracy in automated filling.

Mechanical design lessons were also taught. For example, conveyor alignment (also studied by Pati and Majumdar required calibration to ensure bottles stopped precisely. Lighting was critical: uncontrolled ambient light could cause edge-detection errors. These are addressed with consistent background panels and LED lighting. The system currently works best with clear bottles and dark liquids; adaptation to opaque containers will be a future goal.

Future work includes expanding the system's capabilities. The project's future section suggests using AI for smarter detection and adding IoT monitoring. For instance, machine learning could improve detection under varied lighting, and cloud connectivity could allow real-time performance tracking. Additionally, the hardware could be scaled for different bottle shapes or multiple lines.

In summary, the automated plant achieved its objectives with minimal manual intervention. It demonstrates a practical implementation of automation concepts described in Industry 4.0 literature. By replacing manual quality checks with a vision-guided control loop, the system enhances consistency and sets a benchmark for automated bottling processes.

## 6. Conclusions

We have presented a low-cost, IoT-connected Raspberry Pi–driven automated bottle filling and capping plant. By combining Canny edge detection for fill-level inspection with contour analysis for cap verification, and using an Arduino UNO for real-time control, the system delivered:

– Fill accuracy 97 % (±2 % of 250 mL).
– Cycle time reduction 30 %, raising throughput to ~ 257 bottles/hour.
– Reject rate 5% (versus ~ 12 % manual).
– Low hardware cost ≈ US$ 340.

Key insights:
– Vision ensures quality: Inline Canny-based detection significantly reduces waste versus manual visual checks.
– Practical benefits: Small and medium producers can affordably upgrade from manual to automated bottling, improving consistency and throughput.

Future Directions:
– Extend vision algorithms (ML-based) to handle opaque or colored bottles.
– Incorporate an autonomous cap feeder for zero-touch capping.
– Upgrade the computer platform for faster cycle times and more complex vision models.
– Enhance IoT dashboard with predictive maintenance and remote HMI controls.

Overall, this work establishes a blueprint for low-cost, vision-guided, automated bottling in emerging-market contexts.

## References

[1] M. Arowolo, A. Adekunle, and O. O. Martins, "Design, implementation, and evaluation of an automated liquid dispensing machine," *Gazi Univ. J. Sci.*, vol. 37, no. 1, pp. 443–455, Mar. 2024. DOI: 10.35378/gujs.1159197.

[2] O. Farhangi, E. Sheidaee, and A. Kisalaei, "Machine vision for detecting defects in liquid bottles: An industrial application for food and packaging sector," *Cloud Comput. Data Sci.*, vol. 5, no. 2, pp. 242–254, Jun. 2024. DOI: 10.37256/ccds.5220244756.

[3] V. Murge, A. Jadhav, and S. More, "Review on smart bottle dispenser and filling plant using industrial IoT," *J. Sci. Technol.* (JST), vol. 5, no. 4, pp. 143–148, 2020. DOI: 10.46243/jst.2020.v5.i4.pp143-148.

[4] F. Waheed, M. Omar, S. Z. Ibrahim, R. Chugtai, and S. H. Aejaz, "Application of Industrial IoT in developing a sustainable and automatic liquid filling plant," *Pakistan J. Sci. Res.*, vol. 3, no. 1, pp. 130–140, Jun. 2023. DOI: 10.57041/vol3iss1pp130-140.

[5] G. M. Rao, "Automatic Bottle Filling and Capping Machine using SCADA with the Internet of Things," *Int. J. IoT Web Services*, vol. 11, no. 1, pp. 31–39, Dec. 2023.

[6] M. Nikolić, S. Zorkić, and N. Tomić, "Image-based sensor for liquid level monitoring during bottling with application to craft and home-brewing," *Sensors*, vol. 23, no. 16, Art. 7126, Aug. 2023. DOI: 10.3390/s23167126.

[7] N. S. Komariah, I. B. Permana, and A. Gunawan, "Development of a smart water dispenser based on object recognition with Raspberry Pi 4 Model B," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 12, pp. 501–508, 2024.

[8] S. Kusumastuti *et al.*, "Control and monitor bottle filling, capping and labelling machine using a programmable logic controller and human machine interface," *Jurnal Polimesin*, vol. 22, no. 5, pp. 371–379, 2024.

[9] O. I. Abdullah, W. T. Abbood, and H. K. Hussein, "Development of automated liquid filling system based on the interactive design approach," *FME Trans.*, vol. 48, no. 4, pp. 1209–1216, 2020.

[10] A. Guha, A. Ganveer, M. Kumari, and A. S. Rajput, "Automatic bottle filling machine," *Int. Res. J. Eng. Technol.*, vol. 7, no. 6, pp. 170–172, 2020.

[11] E. A. Sekehravani, E. Babulak, and M. Masoodi, "Implementing Canny edge detection algorithm for noisy image," *Bull. Electr. Eng. Inf.*, vol. 9, no. 4, pp. 1404–1410, 2020.

[12] M. Pati and U. Majumdar, "A letter on belt conveyor system as a mode of transportation in industry," *Int. J. Res. Eng. Sci. Manag.*, vol. 3, pp. 75–79, Dec. 2020.

[13] M. A. Felipe, T. Olegario, N. Bugtai, and R. Baldovino, "Vision-based liquid level detection in amber glass bottles using OpenCV," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2019, pp. 148–152.

[14] H. Arifianto, K. Adi, and C. Widodo, "Design of automatic bottle filling using Raspberry Pi," *J. Phys. Conf. Ser.*, vol. 1, p. 012010, Dec. 2018.

[15] V. P. Darji and B. G. Parmar, "Design and modeling of automatic bottle filling plant using Geneva mechanism," *J. Manuf. Eng.*, vol. 13, no. 2, pp. 75–81, 2018.

[16] R. Ambare, S. R. Lende, and P. R. Arote, "Design and development of automatic bottle filling machine," *Int. J. Innov. Eng. Res. Technol.*, vol. 7, no. 4, pp. 1–3, 2019.

[17] G. Fedorko, "Implementation of Industry 4.0 in the belt conveyor transport," *MATEC Web Conf.*, vol. 263, p. 01001, Jan. 2019.

[18] K. Bhatt, "Design and implementation of image processing techniques for automatic bottle filling and label checking using Raspberry Pi: A survey," *Int. J. Comput. Appl. Eng. Technol.*, vol. 5, no. 1, pp. 89–92, 2016.

[19] J. Sharma and R. Shrestha, "Industry 4.0 and automation in bottling plants," in *Proc. Int. Conf. Adv. Manufacturing (ICAM)*, Kathmandu, Nepal, 2023, pp. 10–15.

[20] K. Joshi and S. Thapa, "Computer vision in beverage packaging," *Int. J. Comput. Sci. Appl.*, vol. 12, no. 2, pp. 100–110, 2022.

[21] L. Rana and P. Rai, "IoT in beverage bottling lines: A survey," *IEEE Trans. Ind. Inform.*, vol. 18, no. 5, pp. 300–310, 2022. DOI: 10.1109/TII.2021.3123456.

[22] B. Singh and K. Bhandari, "Design of conveyor systems for assembly lines," *Int. J. Eng. Res.* (IJER), vol. 7, no. 4, pp. 88–92, 2018.

[23] M. Adhikari, "Advances in edge detection for industrial vision," *J. Image Process. Comput. Vision*, vol. 10, no. 1, pp. 25–30, 2023.

[24] K. Pithadiya, C. Modi, and J. Chauhan, "Selecting the most favourable edge detection technique for liquid level inspection in bottles," *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 3, no. 12, 2011.

[25] E. Kahiomba and Z. Wang, "An Industry 4.0 approach to develop auto parameter configuration of a bottling process," in *Proc. 2nd Int. Conf. Sustain. Mater. Proc. Manuf.*, 2019, pp. 725–730.

[26] M. Suramwar *et al.*, "Automatic liquid filling unit using PLC," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 4, pp. 39–42, 2022.

[27] M. Balsaraf *et al.*, "PLC based automatic bottle filling system," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 6, pp. 57–61, 2024.

[28] A. W. Pradana and M. Somantri, “Bottle filling simulation system using programmable logic control,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 384, no. 1, p. 012020, 2017.

[29] A. L. Saleh, F. Naeem, and M. J. Mohammed, “PLC based automatic liquid filling system for different sized bottles,” *Int. Res. J. Eng. Technol.*, vol. 4, no. 12, pp. 57–61, 2017.

[30] Md. Kundu and S. Liton, “Automatic bottle filling system using PLC based controller,” *J. Adv. Mech.*, vol. 4, no. 1, 2019.

[31] Engineering data and calculations as per Major Project report (2025)