Applied Intelligence (2025) 55:524
https://doi.org/10.1007/s10489-025-06334-3

# Real-time AI-driven quality control for laboratory automation: a novel computer vision solution for the opentrons OT-2 liquid handling robot

Sana Ullah Khan[2] · Vilhelm Krarup Møller[1] · Rasmus John Normand Frandsen[1] · Marjan Mansourvar[1]

Accepted: 1 February 2025 / Published online: 10 March 2025

**Abstract**
The adoption of robotics and automated solutions in life sciences R&D has accelerated in recent years, driven by the need to process increasing sample volumes, protect laboratory staff from hazardous substances, and manage financial pressures. Various automation systems, each with distinct levels of sample processing, transportation tasks, and data management, are available to meet specific application requirements, with liquid handling robots taking pivotal positions in these systems. However, current liquid handling robots, such as the Opentrons OT-2, lack integrated vision-based quality control, which limits their accuracy and reliability. This study presents an AI-driven computer vision model designed to enhance quality control in laboratory automation. By integrating the YOLOv8 object detection model with the OT-2, our model enables precise detection of pipette tips and liquid volumes, providing real-time feedback on errors, such as missing tips and incorrect liquid levels. Our results demonstrate the model's effectiveness and accessibility, presenting an affordable solution for improving automation in academic and research laboratories. This closed-loop system transforms the OT-2 into a robust tool for automated laboratory tasks, making it an accessible and cost-effective approach for enhancing quality control in laboratory automation and addressing a critical gap in available tools for resource-limited settings.

**Keywords** Life science Automation · Machine Learning · Robotics · Computer vision · Image processing

## 1 Introduction

The rapid advancements in robotics and artificial intelligence (AI) are increasingly transforming life sciences, particularly in laboratory automation [1]. The demand for automated laboratory solutions in life science R&D is driven by the need to process large sample volumes efficiently, protect laboratory staff from hazardous substances, and mitigate financial constraints. Automation enhances reproducibility, efficiency, and safety in laboratory processes. However, significant barriers, including high costs, protocol variability, and limited expertise, restrict its widespread adoption [2].

Despite the success of automation in fields like manufacturing and food production, laboratory automation in life sciences remains limited. Current systems are often restricted to partially automated tasks, with end-to-end automation being rare outside pharmaceutical high-throughput screening. University laboratories, unlike industrial and clinical labs, continue to rely heavily on manual processes, constrained by short-term funding, high protocol variability, and a focus on personnel development over equipment investments [3]. These constraints highlight the urgent need for flexible, modular, and cost-effective automation systems that can adapt to dynamic laboratory environments [4].

Most existing systems lack integrated real-time quality control mechanisms, which are critical for ensuring experimental accuracy. This paper addresses this gap by introducing an AI-driven computer vision solution that integrates YOLOv8 with the Opentrons OT-2 liquid handling robot. The system transforms

✉ Marjan Mansourvar
marjma@dtu.dk

Sana Ullah Khan
s203133@student.dtu.dk

Vilhelm Krarup Møller
vikmol@dtu.dk

Rasmus John Normand Frandsen
rasf@dtu.dk

[1] DTU Bioengineering, Department of Biotechnology and Biomedicine, Søltofts Plads, Building 221, DK-2800 Kongens Lyngby, Denmark

[2] Department of Technology, Management and Economics (DTU Management), Technical University of Denmark (DTU), Akademivej, Building 358, DK-2800 Kongens Lyngby, Denmark

the OT-2 into a closed-loop robotic platform capable of real-time error detection and correction. By focusing on accessibility and affordability, our approach enables wider adoption of high-quality laboratory automation in academic and resource-limited settings, bridging a significant gap in current technologies.

Liquid handling robots are central to laboratory automation, particularly for high-precision tasks like drug screening, biochemical assays, molecular genetics and cell biology, where precise sample transfer is essential [5]. Manual pipetting, a tedious and error-prone process, underscores the need for automated solutions to improve productivity and reproducibility. While high-end systems like the Tecan Fluent offer active monitoring, cost-effective alternatives such as the Opentrons OT-2 lack real-time feedback, which can lead to errors like missing tips or incorrect liquid transfers, compromising experimental validity [6].

The integration of AI, particularly computer vision, into robotics marks a significant advancement in laboratory automation [7]. Computer vision enables robots to "see" and interpret their environment [8], facilitating intelligent interactions with objects, laboratory instruments, and even humans [9]. This convergence of AI and robotics, often termed robotic vision [10], has unlocked new capabilities in sectors such as autonomous navigation, industrial automation, and human–computer interaction [11]. Using advanced object detection techniques, robots equipped with computer vision can perform essential tasks like object recognition, spatial positioning, and motion tracking, ensuring high precision in laboratory processes [12]. Additionally, computer vision allows robots to adapt to dynamic and unstructured environments by continuously analyzing visual data, enabling real-time decision-making and enhanced autonomy in challenging scenarios.

Traditional computer vision applications [13], relied on feature-based algorithms like Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) for object recognition. However, recent advances in deep learning [14] have shifted the field toward convolutional neural networks (CNNs), which excel at handling large datasets and complex image analysis tasks [15]. Among these, the You Only Look Once (YOLO) models [16], have demonstrated exceptional capabilities in real-time object detection and accuracy. The latest iteration, YOLOv8, combines fast processing speeds with high detection accuracy and features advanced components like the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) for reliable results [17]. Its user-friendly Python package [18] and optimized annotation tools make it particularly suitable for real-time applications in laboratory environments [19].

Beyond laboratory automation, YOLO models have shown versatility in fields such as agriculture, where they have been applied for crop detection, maturity classification, and disease identification. For example, Moya et al. demonstrated YOLOv5's adaptability to diverse environmental conditions for crop maturity classification [20]. Similarly, recent advancements, such as YOLO-TLA, incorporate lightweight modifications and global attention mechanisms to enhance small object detection in resource-constrained environments [21] reflecting ongoing efforts to make object detection more efficient and broadly applicable.

This paper contributes to the field by presenting: (i) a real-time feedback loop enabling closed-loop control for the OT-2 robot, (ii) the novel application of YOLOv8 optimized for high-throughput laboratory workflows, and (iii) the practical implementation of the system in real-world scenarios, demonstrating its scalability and robustness. These advancements mark a significant step toward modular, affordable, and highly adaptable laboratory automation systems, fostering broader adoption of robotics in life sciences.

## 2 Methodology

The experiments were conducted in February 2024, following a structured workflow to ensure precise data acquisition and processing in real-world laboratory settings. The methodology consisted of four main steps: (1) Data Collection – Capturing images of pipette tips and liquid volumes using a camera integrated with the OT-2 robot under varied lab conditions; (2) Image Annotation – Using the Computer Vision Annotation Tool (CVAT) to label pipette tips and liquid volumes for object detection; (3) Model Training – Training the YOLOv8 object detection model on the annotated dataset; and (4) Real-Time Integration and Error Detection – Deploying the trained model in a server-client setup for real-time pipetting error detection and feedback to the OT-2. This workflow supports efficient, high-quality control for automated liquid handling in laboratory environments.

### 2.1 Data acquisition

To collect the necessary data for our study, we equipped the OT-2 robot with a Logitech C920 HD PRO camera, strategically positioned to focus on the area of interest. This setup ensured timely capture of high-quality images under realistic laboratory conditions, including complex backgrounds and variable lighting, to enable the model's effective performance in real-world scenarios.

During operation, the robotic 8-channel pipetting arm triggered the server to capture an image at each predefined position. A total of 456 images were selected for training the model, encompassing a range of conditions typically encountered in laboratory workflows. The data collection focused on three key scenarios, summarized in Table 1:

1. **P300 Pipette Tips**: We captured 208 images of P300 pipette tips filled with transparent, red, blue, and green liquids, with volumes ranging from 50 to 300 µl in 10-µl increments.

**Table 1** Dataset Details

| Image Category | Quantity | Description |
|---|---|---|
| P300 Pipette Tips | 208 | Transparent, red, blue, green liquids (50–300 µL in 10 µL increments) |
| P20 Pipette Tips | 48 | Transparent, red, blue, green liquids (empty, 10 µL, 20 µL) |
| Complex Scenarios | 200 | Missing tips, varying liquid volumes and colors |
| **Total** | **456** | |

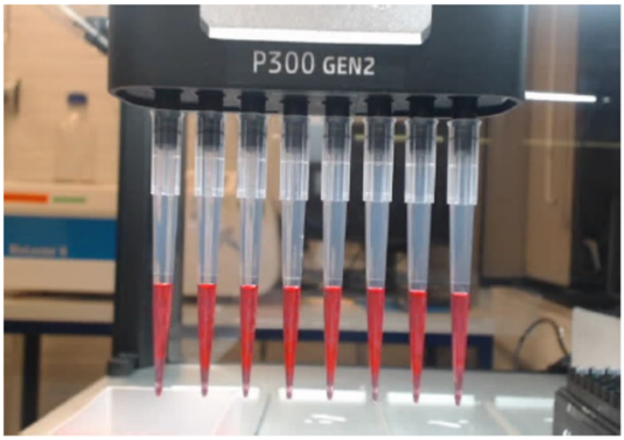


**Fig. 1** Representative sample of the captured images of filled pipette tips found in the training data set. To emulate as close to a realistic operational scenario as possible, images were captured with a complex background

2. **P20 Pipette Tips**: An additional 48 images were taken of smaller P20 tips, featuring the same color range but with liquid volumes at empty, 10 µl, and 20 µl.
3. **Complex Scenarios**: To simulate more complex scenarios, we included 200 images depicting missing tips and varying liquid volumes and colors.

These scenarios were designed to simulate typical experimental situations, ensuring the dataset's diversity and robustness for machine learning model development. Figure 1**.** presents a representative sample of the collected data.

## 2.2 Computer vision and AI model development using YOLOv8

To prepare the training dataset for deep learning, each image was pre-processed and aligned to fit the YOLOv8 format, optimizing it for transfer learning. During the image annotation process, we prioritize precise placement of bounding boxes and accurate delineation of the area within each box. This meticulous approach aimed to provide the deep learning network with detailed and precise training inputs, thereby bolstering the model's overall accuracy and performance. The images were annotated using the online 'Computer Vision Annotation Tool' (CVAT) [22], recoding whether each of the eight tips was attached to the robotic arm and further the specific amount of liquid in each tip. Precise annotation was conducted to label both the pipette tips and the liquid volumes within them. This process involved manually drawing bounding boxes around each pipette tip and the corresponding liquid area, which enables the model to distinguish between correctly and incorrectly attached tips, as well as assess liquid volumes accurately. Figure 2**.** shows an example of an annotated image from our dataset.

The YOLOv8 model was trained using Python libraries, specifically Ultralytics' super-gradients and PyTorch, on an NVIDIA RTX GPU with an intel core i9 processor. To ensure robustness and generalization, the dataset was divided into an 80:20 split for training and validation, respectively. This setup minimized overfitting and enabled the model to perform reliably on new data. The source code for this model is available on GitHub at:

https://github.com/BDD-G/OT2-Computer-Vision.

## 2.3 Design of workflow and robotics control

The operation of the Opentrons OT-2 liquid handler robot is controlled by a built-in Raspberry Pi 3 + circuit board with a Quad Core 1.2GHz Broadcom BCM2837 64bit CPU and 1GB RAM, into which workflow programs are loaded and executed from. The limited processing power of the Raspberry Pi makes it unfeasible to simultaneously support control of the robot and perform real-time image analysis.

To address this, we implemented a client–server setup where the Raspberry Pi manages robotic control tasks (as in the factory setup) while offloading the computationally demanding image processing tasks to an external PC. This system setup is illustrated in Fig. 3**.** showing the integration of the OT-2 robot, external PC, and camera, enabling distributed processing and real-time feedback for the robot's performance. This distributed approach allows for efficient task management without overburdening the robot's onboard processor.

### 2.3.1 Implementation of distributed processing

Distributed processing enables the system to overcome the inherent limitations of the OT-2's low-powered Raspberry Pi. In this setup (Fig. 4**.**), an external PC with a powerful processor acts as a server, managing computationally

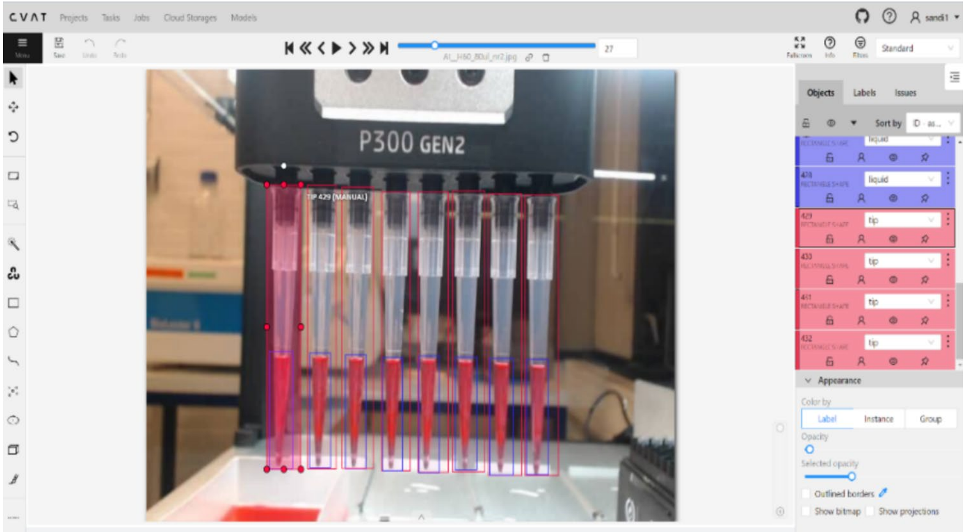


**Fig. 2** Example of an annotated image using the CVAT annotation environment. Two annotation classes were manually added; one outlining the pipette tips (red boxes) and the second outlining the liquid found within individual pipette tips (blue boxes)

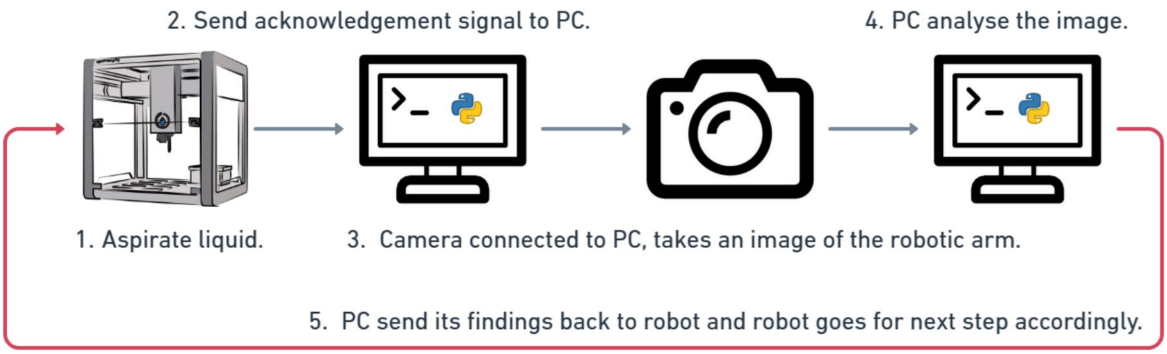

2. Send acknowledgement signal to PC.
4. PC analyse the image.
1. Aspirate liquid.
3. Camera connected to PC, takes an image of the robotic arm.
5. PC send its findings back to robot and robot goes for next step accordingly.

**Fig. 3** Integration of the Opentrons OT-2 with the external PC and camera for an AI control vision system

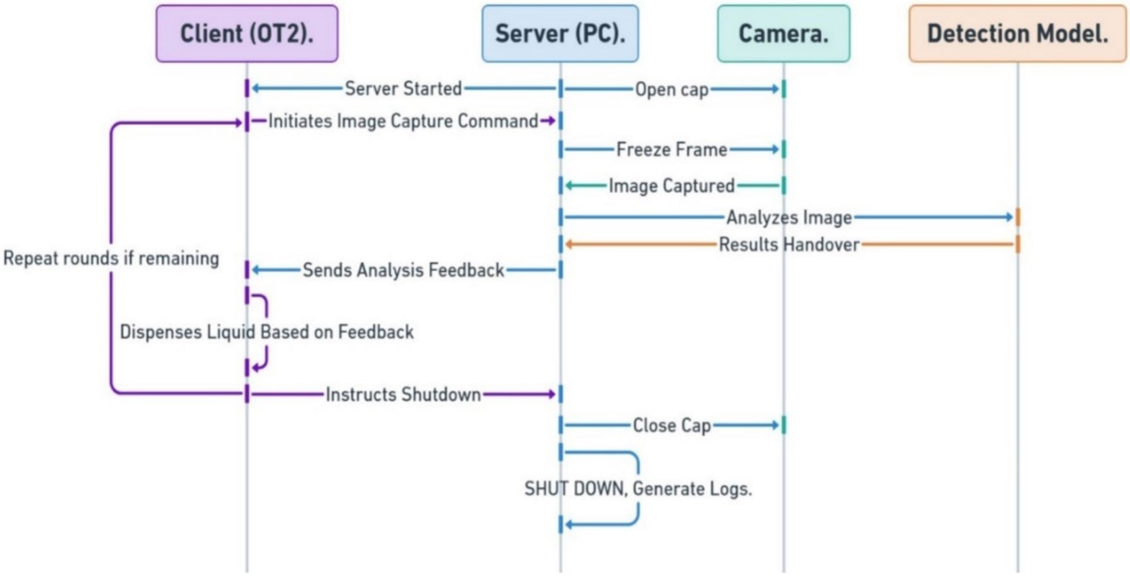


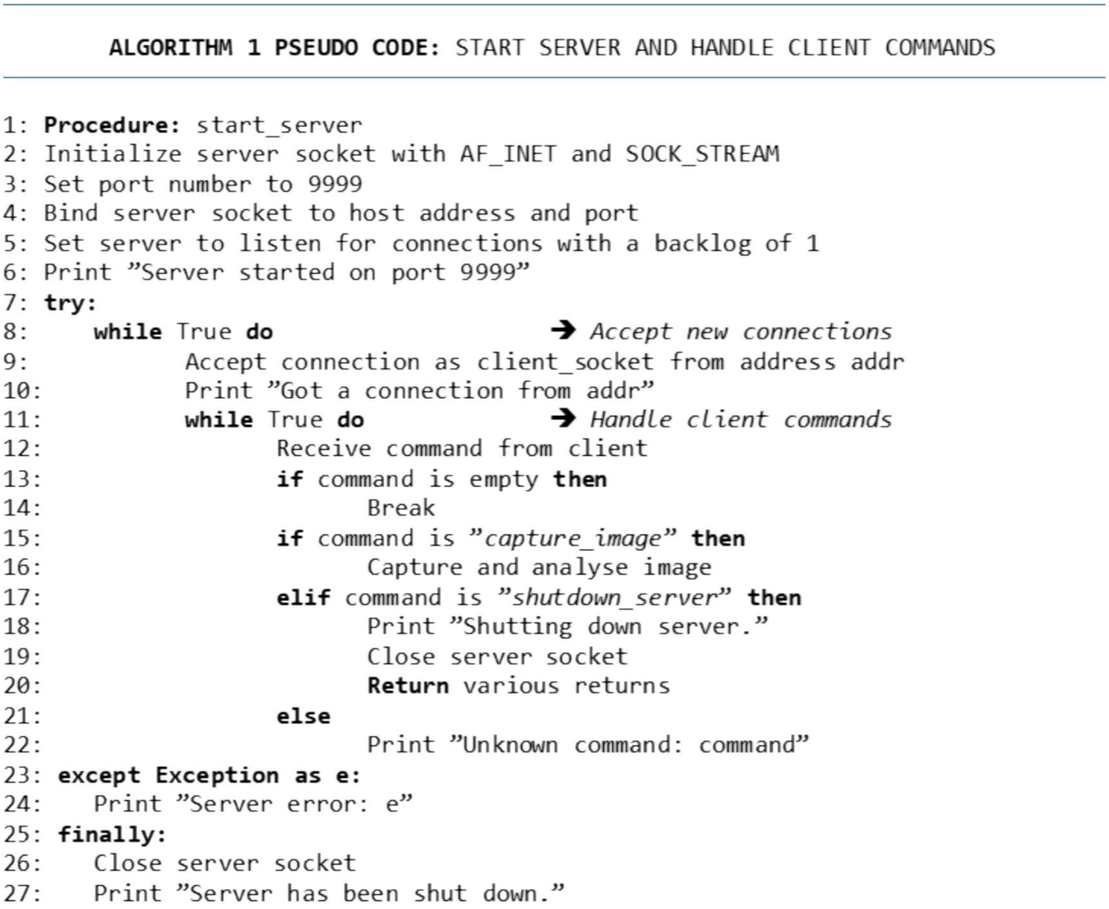
**Fig. 4** Client–Server interaction workflow in a distributed processing setup for real-time image analysis and error handling in robotic pipetting

intensive tasks such as object detection and image analysis. The Raspberry Pi functions as a client, sending requests to the server to inspect the robotic pipetting arm after each cycle.

The server responds to commands issued by the client (OT-2). When the client aspirates liquid into tips and positions its robotic arm to a predefined position on the robotics deck, it sends a 'capture_image' command to the server. Upon receiving this command, the server captures the image, applies a trained object detection model to identify tips and liquid contents, and then communicates its findings back to the client. If the server does not detect any errors, it sends a “go-ahead” signal to the Raspberry Pi, which then executes the next step in the robotics program. However, in cases where the server detects errors, missing tips or wrong liquid levels, it responds by requesting human intervention, or by log the anomalies for further analysis and then sending a go-ahead signal to the client (OT-2), that continues with it program.

The server further maintains structured data to track the detected tips and liquid volumes across consecutive images, continuously updating these data structures as new images are processed. When commanded with a 'shutdown_server' directive, the server gracefully ceases operations and prepares to transmit the accumulated data, ensuring a smooth shutdown process.

The client interacts with the server using two primary commands: 'capture_image' and 'shutdown_server'. When the robotic pipetting arm picks up tips and aspirates liquid, it positions itself for imaging. The client (OT-2) sends a 'capture_image' command to prompt the server for analysis. Subsequently, the 'shutdown_server' command instructs the server to conclude its operations, offering an effective means for the client to manage computational resources. This client–server arrangement allows for seamless exchange of commands and images, enabling the server to process and relay outcomes efficiently. This operational framework optimizes resource utilization and enhances overall processing efficiency. Figure 5**.** illustrates the algorithm in command



**ALGORITHM 1 PSEUDO CODE:** START SERVER AND HANDLE CLIENT COMMANDS

```
Procedure: start_server
Initialize server socket with AF_INET and SOCK_STREAM
Set port number to 9999
Bind server socket to host address and port
Set server to listen for connections with a backlog of 1
Print ”Server started on port 9999”
try:
    while True do                          ➔ Accept new connections
        Accept connection as client_socket from address addr
        Print ”Got a connection from addr”
        while True do                      ➔ Handle client commands
            Receive command from client
            if command is empty then
                Break
            if command is ”capture_image” then
                Capture and analyse image
            elif command is ”shutdown_server” then
                Print ”Shutting down server.”
                Close server socket
                Return various returns
            else
                Print ”Unknown command: command”
except Exception as e:
    Print ”Server error: e”
finally:
    Close server socket
    Print ”Server has been shut down.”
```

**Fig. 5** Pseudo code for client–server algorithm, the algorithm illustrates the interaction where the client (OT-2) commands the server to capture and analyze images, identifying tips and liquid contents, and maintaining structured data. The server processes 'capture_image' commands until it receives a 'shutdown_server' command, ensuring efficient management of computational tasks and resource utilization

flow and data exchange between the OT-2's Raspberry Pi and the external PC, for client–server setup.

## 2.4 AI-powered error detection and measurement techniques

### 2.4.1 Detecting missing pipette tip position

The algorithm for identifying missing pipette tips begins by assuming the presence of eight tips. If all eight tips are detected, the process is completed. If fewer than eight tips are detected, the algorithm proceeds to identify the missing tips. The detection algorithms assume that missing pipette tips or incorrect liquid levels can be reliably identified by locating expected features through bounding boxes around pipette tips and liquid volumes. This assumption simplifies the detection process and allows for efficient classification within common laboratory setups.

First, the algorithm calculates the expected horizontal distance between the tips, called 'section_width'. This is done by dividing the total width of the robotic arm ('arm_width') into ten equal sections, with the first and last sections serving as gaps before and after the tips. The 'section_width' is the distance between each tip. Using the bounding box information of the detected tips, the algorithm calculates the center point of each tip. Since the tips are evenly spaced and aligned horizontally, their center points should align as well.

To accurately check the presence of each tip, the algorithm defines a 'tolerance_range' around each expected position, which is half of the 'section_width'. This range allows for flexibility and improves detection accuracy. The algorithm checks whether a detected tip's center point falls within this expected range. If no tip is found within the range, it is marked as missing.

Finally, the algorithm then generates two lists: one indicating the presence of tips (with '1' for present and '0' for missing) and another listing the positions of the missing tips. This process ensures precise identification and documentation of missing tips, enhancing the reliability and error-handling capabilities of the OT-2 robot. By systematically verifying each expected tip position, the system improves the accuracy of automated liquid handling processes, contributing significantly to operational efficiency and consistency in laboratory workflows.

### 2.4.2 Measuring liquid levels in tips

The develop algorithm for accurately determining the liquid level (proxy for volume) inside pipette tips employs a systematic approach. Initially, the object detection model identifies both the tips, and the liquid contained within them. It next measures the vertical distance between the tip of pipette tip and the liquid surface to determine the liquid level, enabling calculation of the percentage of the tip filled with liquid. Using the known total capacity of the tip, we next compute the precise amount of liquid present.

However, due to the conical shape of the tip and the uneven distribution of liquid inside, our initial method required refinement. To address this challenge, we developed a polynomial regression model [23]. This model utilizes both the percentage of the tip filled with liquid and the actual measured amount of liquid inside the tip as parameters, based on the recorded images and volumes the robot was instructed to aspire. The polynomial regression model is represented in Eq. (1):

$$V = aP^2 + bP + c \tag{1}$$

where:

- V is the volume of liquid.
- $P$ is the percentage of the tip filled with liquid.
- $a$, $b$, $c$ are regression coefficients determined through training.

Implementing this approach significantly improved the accuracy of the liquid volume measurements. The refined algorithm ensures precise data essential for automated liquid handling tasks in practical laboratory settings. By integrating object detection with precise measurement techniques and advanced regression modeling, our algorithm expertly adapts to the complex geometries and variations encountered in real-world laboratory conditions.

# 3 Experimental results and analysis

This section provides a detailed account of the observed outcomes concerning the detection of pipette tips, liquids, and liquid levels, focusing on their performance in real-world laboratory settings. An essential component of this analysis is the evaluation of the processing time required for these detections. All these results and analyses are produced using a calibrated OT-2 liquid handler, ensuring the reliability and accuracy of the evaluations.

## 3.1 Evaluating model performance

In assessing our model's performance, we prioritize several critical factors. Initially, we analyze the losses incurred during both training and validation stages, including box loss, classification loss, and Distribution Focal Loss (DFL) [24]. These metrics reveal the model's accuracy in classifying detected objects, such as different types of liquids or fill levels. Specifically, DFL measures the precision in predicting the orientation and direction of objects, ensuring accurate positioning within the image frame (Fig. 6).

The selection of batch size, number of epochs, and learning rate were optimized through iterative testing. Initial trials indicated that batch sizes below 16 caused memory inefficiencies, while larger sizes led to a plateau in accuracy gains. A batch size of 16 was therefore chosen to balance memory utilization and performance. Similarly, we experimented with different numbers of epochs and determined that 50 epochs provided a satisfactory trade-off between model training time and accuracy, as indicated by the stable mAP@95 and recall scores. Any deviations from these parameters would likely impact the model's precision, particularly under varied lighting conditions and diverse liquid volumes present in the dataset.

We observe a consistent reduction in these losses over the course of training and validation. This trend indicates that the model has progressively enhanced its ability to recognize and categorize objects more accurately, thereby reducing errors and improving overall performance.

In addition to measure the accuracy of the ‘pipette tip’ and liquid detection, we calculate mean average precision (mAP) as our evaluation metric [25]. mAP is a standard evaluation metric in. object detection calculates the mean Average Precision (AP) across all classes at a predetermined Intersection over Union (IoU) threshold [26]. To define precision in object detection, the algorithm calculates the ratio of true positives to the total number of true positives and false positives. A true positive occurs when the IoU between the predicted box and the ground truth exceeds the set IoU threshold, while a false positive occurs when the IoU falls below that threshold. Precision is then calculated as represented in Eq. (2):

$$\frac{tp}{tp+fp} \tag{2}$$

For each class, we compute the mean precision by iterating over a set of IoU thresholds and averaging the results. For mAP50-95, we use IoU thresholds from 0.5 to 0.95 in increments of 0.05. The average precision over this range is considered the class AP. By averaging the APs of all classes, we obtain the mAP50-95. (Table 2.).

Following completion of the training, the model achieved a mAP@0.5 score of 0.995, indicating it's exceptional accuracy in identifying objects under less stringent criteria. Despite a slightly lower score of 0.88513 with

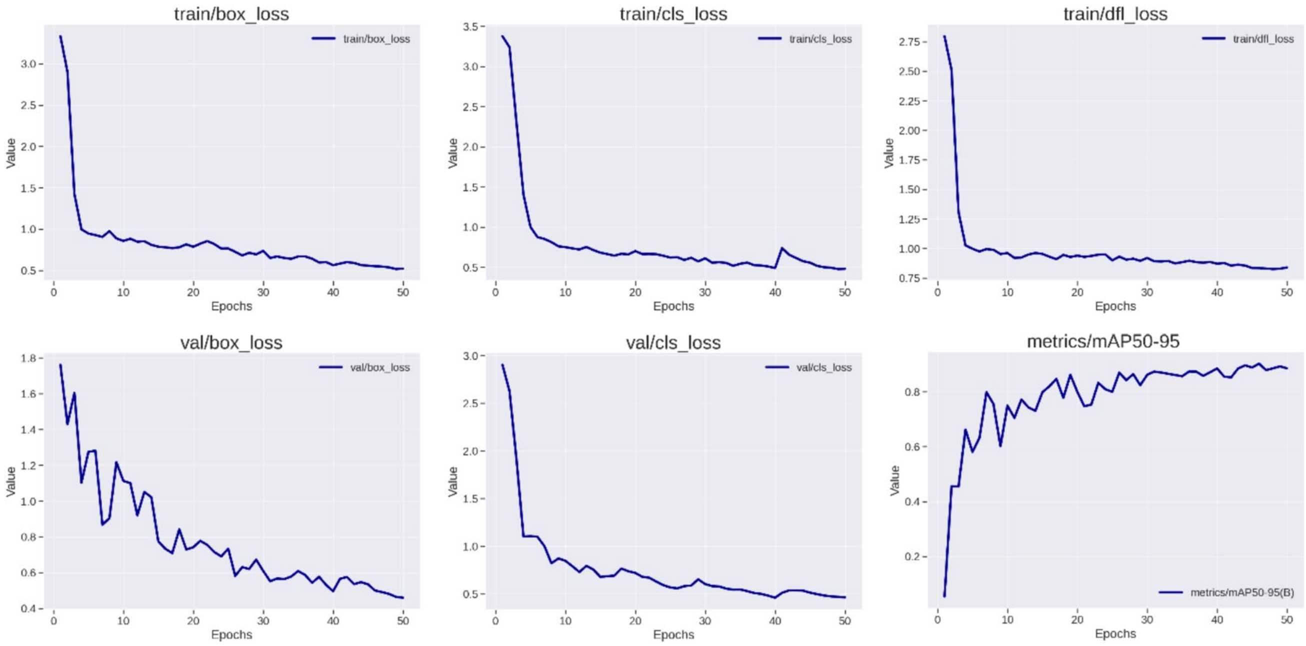


**Fig. 6** Loss and Performance Metrics During YOLOv8 Model Training. This graph displays key performance trends for box loss, classification loss (cls_loss), and Distribution Focal Loss (DFL) over the training and validation phases, each showing continuous decreases that indicate effective learning and refinement of object detection capabilities. Notably, the classification loss stabilizes below 0.5 during validation, demonstrating the model's capacity to achieve a low error rate when classifying new data and balancing false positives with false negatives. The DFL, which enhances the model's precision in predicting object orientation and detail, decreases substantially, highlighting improvements in fine-grained object detection. Additionally, the mean Average Precision (mAP) at IoU thresholds from 50 to 95% exhibits a consistent upward trend, signifying the model’s ability to accurately detect and localize objects in complex lab environments. This progression confirms the model’s robust learning and reliability for real-world laboratory applications

**Table 2** Mean Average Precision (mAP) Evaluation

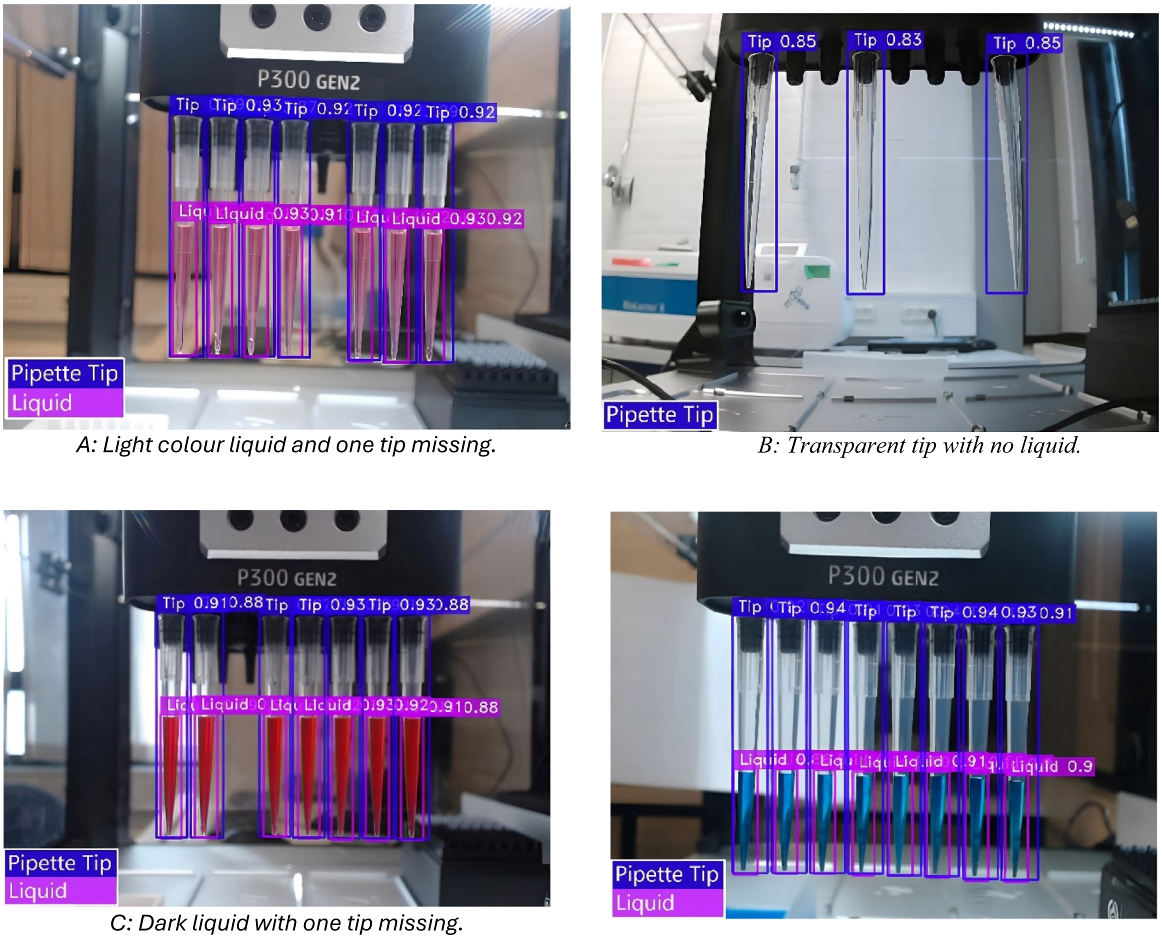
| Matric | Value |
|---|---|
| mAP@50 | 0.995 |
| mAP@95 | 0.88513 |
| Recall | 0.99 |

a stricter threshold (mAP@0.95), the model still demonstrates robust performance even in challenging conditions.

The perfect recall score of 0.99 underscores the model's ability to identify all relevant objects present in the tested images without missing any. The model strikes a balance between precision and recall, effectively distinguishing relevant objects while minimizing false identifications. This equilibrium is crucial for developing a reliable and practical model suitable for deployment in real-world applications.

## 3.2 Readings from detected tips and liquids

Following the testing of our system, we conducted 50 experiments to evaluate its performance. The experiments involved running the OT-2 robot in real-time, using live data directly from an actual pipetting process. During these experiments, intentional omissions of tip attachments were made, and a '12 × 8 Greiner 96-well plate' was used to assess the system (Fig. 7.). The system effectively captured images upon receiving a 'capture signal' from the robot, accurately delineating bounding boxes around detected tips and liquid in the images. This allowed for precise counting of empty and filled tips, as well as mapping their statuses within the 12 × 8 well plate layout. Our tests revealed a 98% accuracy in identifying missing tips and their respective locations, demonstrating high precision across various scenarios.



*A: Light colour liquid and one tip missing.*

*B: Transparent tip with no liquid.*

*C: Dark liquid with one tip missing.*

*D: liquid in all tips.*

**Fig. 7** Images of pipette tip detection outcomes, illustrating various scenarios used to evaluate the accuracy of the system in identifying missing tips and detecting liquid presence. Each scenario simulates a unique laboratory condition: (A) Light-colored liquid with one tip missing; (B) Transparent tip with no liquid; (C) Dark liquid with one tip missing; (D) All tips containing liquid. These images showcase the system's ability to delineate bounding boxes around tips and liquid levels, supporting high-accuracy detection and mapping of missing or empty tips within experimental setups

Figure 8. illustrates a specific instance of a missing tip, providing a reference for future refinements in this area. The left side depicts counts of detected tips and liquid, while the right side shows the layout of the 12 × 8 Greiner 96-well plate, indicating missing tips ('MISSING').

### 3.3 Measuring liquid

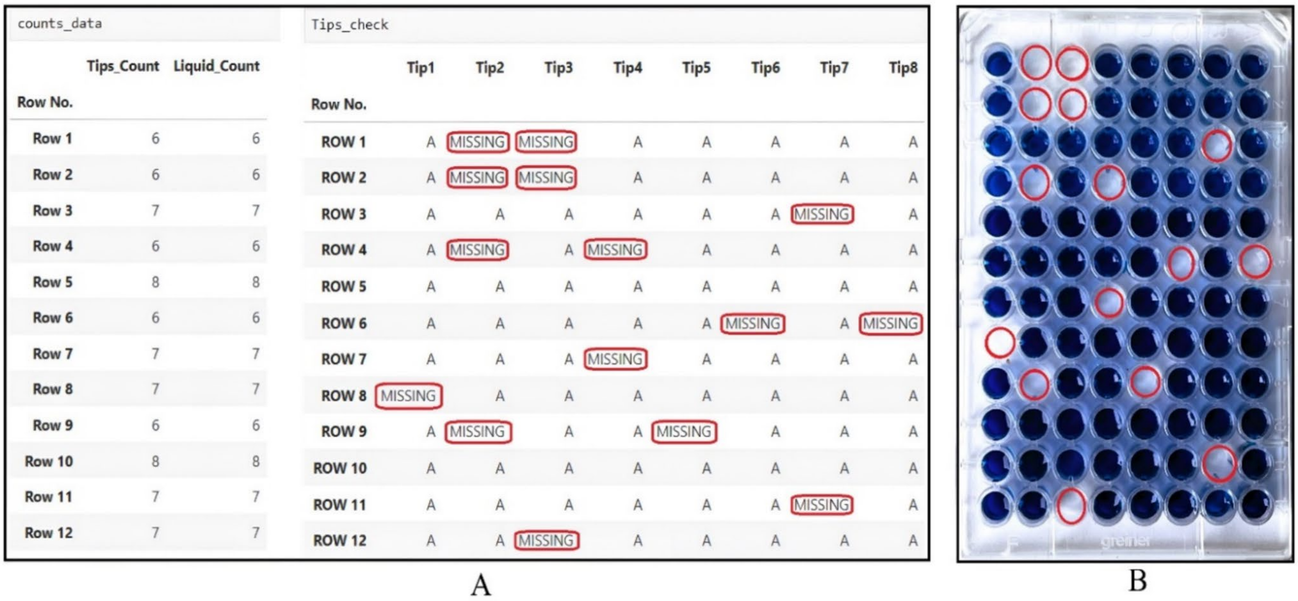
To determine the volume of liquid within the pipette tip, we conducted 100 tests using varying liquid quantities. By leveraging the fixed position of the camera and the precise movements programmed for the robotic pipetting arm, our regression model significantly enhanced the accuracy of our volume measurements. The model's precision correlates closely with the actual liquid volume within the tip, although it performs less reliably when the liquid volume is minimal. Optimal accuracy, in the prediction, is achieved when the tip contains at least 25% of its total capacity.

The polynomial regression model for volume estimation, as defined by Eq. (1), yielded the following optimized coefficients after training: $a = 0.0401$, $b = 1.3500$, and $c = -40.0435$. These coefficients were calibrated to minimize prediction error and align closely with observed liquid measurements. Figure 9**.** illustrates the polynomial regression curve, demonstrating how the model correlates the percentage of liquid in the pipette tip to the estimated volume. The figure also highlights the regression equation and key data points used during model training to achieve accurate predictions.

Our results demonstrate an average measurement accuracy of 95% in assessing the percentage of liquid within the tip. This suggests that our current approach is effective, but further improvements could be achieved by implementing liquid segmentation techniques to precisely mask and quantify the liquid inside the tip.

Table 3 displays a log screenshot of the detected liquid quantities for the 12 × 8 Greiner 96-well plate. In this figure, '0' indicates the absence of liquid due to either missing tips or no liquid present in those positions. Positive values reflect the measured volume of liquid in each tip, represented in microliters. For instance, in Row 1, our system detects that tips 1 and 2 are missing, while tips 3 and 4 contain approximately 50 and 51 µl, respectively. This allows precise tracking of liquid levels across multiple pipette tips, offering a clear representation of fill levels within the well plate.

To further validate the model's performance, Table 4 compares the actual volumes with the farthest predicted values identified in our dataset, providing a detailed analysis of the prediction error. The average error percentage across these examples is calculated to be 3.4%. This highlights the system's ability to deliver reliable measurements with minimal deviation.


counts_data

| Row No. | Tips_Count | Liquid_Count |
|---|---|---|
| Row 1 | 6 | 6 |
| Row 2 | 6 | 6 |
| Row 3 | 7 | 7 |
| Row 4 | 6 | 6 |
| Row 5 | 8 | 8 |
| Row 6 | 6 | 6 |
| Row 7 | 7 | 7 |
| Row 8 | 7 | 7 |
| Row 9 | 6 | 6 |
| Row 10 | 8 | 8 |
| Row 11 | 7 | 7 |
| Row 12 | 7 | 7 |

Tips_check

| Row No. | Tip1 | Tip2 | Tip3 | Tip4 | Tip5 | Tip6 | Tip7 | Tip8 |
|---|---|---|---|---|---|---|---|---|
| ROW 1 | A | MISSING | MISSING | A | A | A | A | A |
| ROW 2 | A | MISSING | MISSING | A | A | A | A | A |
| ROW 3 | A | A | A | A | A | A | MISSING | A |
| ROW 4 | A | MISSING | A | MISSING | A | A | A | A |
| ROW 5 | A | A | A | A | A | A | A | A |
| ROW 6 | A | A | A | A | A | MISSING | A | MISSING |
| ROW 7 | A | A | A | MISSING | A | A | A | A |
| ROW 8 | MISSING | A | A | A | A | A | A | A |
| ROW 9 | A | MISSING | A | A | MISSING | A | A | A |
| ROW 10 | A | A | A | A | A | A | A | A |
| ROW 11 | A | A | A | A | A | A | MISSING | A |
| ROW 12 | A | A | MISSING | A | A | A | A | A |

**Fig. 8** Evaluating the pipette tip detection capability of the developed systems; A) Output from the pipette tip detection workflow, where 'A' indicate that a pipette tip was attached (detected), while 'MISSING' indicates that a specific pipette tip was absent. B) The resulting 96-well plate following pipetting of a blue liquid, showing perfect correlation with the expected pattern based on the pipette tip detections in panel A

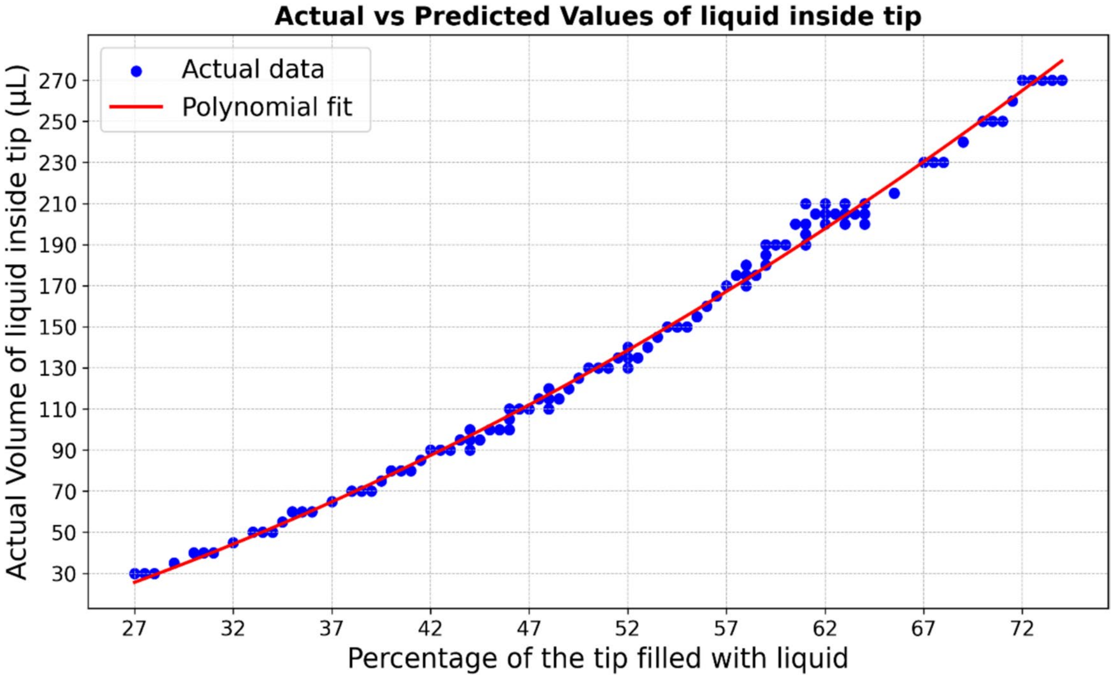


**Fig. 9** Polynomial regression model for liquid volume estimation. The curve illustrates the relationship between the percentage of the pipette tip filled with liquid and the predicted liquid volume. Key data points and the regression equation are annotated to demonstrate the accuracy of the model in correlating liquid levels to volumes

**Table 3** Log screenshot showing detected liquid volumes in the tips. The number '0' indicates the absence of liquid due to missing tips, while other numbers represent the quantity of liquid present in each tip, measured in microliters

| Tips_Volume | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Tip1 | Tip2 | Tip3 | Tip4 | Tip5 | Tip6 | Tip7 | Tip8 |
| Row No | | | | | | | | |
| Row 1 | 0 | 0 | 50 | 51 | 52 | 49 | 51 | 49 |
| Row 2 | 71 | 0 | 0 | 69 | 70 | 70 | 68 | 70 |
| Row 3 | 88 | 92 | 91 | 91 | 90 | 91 | 0 | 89 |
| Row 4 | 113 | 0 | 109 | 0 | 110 | 112 | 108 | 109 |
| Row 5 | 132 | 129 | 130 | 131 | 133 | 133 | 130 | 132 |
| Row 6 | 147 | 152 | 148 | 153 | 150 | 0 | 149 | 0 |
| Row 7 | 167 | 174 | 168 | 0 | 167 | 166 | 166 | 170 |
| Row 8 | 0 | 187 | 188 | 187 | 194 | 188 | 191 | 193 |
| Row 9 | 215 | 0 | 210 | 211 | 0 | 212 | 207 | 211 |
| Row 10 | 229 | 232 | 233 | 233 | 234 | 235 | 226 | 234 |
| Row 11 | 246 | 244 | 247 | 255 | 251 | 256 | 0 | 253 |
| Row 12 | 266 | 275 | 0 | 274 | 276 | 275 | 264 | 268 |

## 4 Discussion

The development of computer vision and machine learning (ML) based error detection system created the foundation for implementing automated control system in cost-effective liquid handling robots, such as the Opentrons OT-2. This system enables real-time quality control, providing precise detection of pipette tips, liquids, and their levels in a real-world laboratory environment. This solution not only enhances the operational efficiency of the Opentrons OT-2 robots in liquid handling but also reduces or eliminates the need for human intervention during laboratory procedures, which is essential for fully realizing the potential of automation.

Our work establishes a novel foundation for integrating real-time quality control into affordable laboratory robotics. By leveraging YOLOv8, the system significantly enhances

**Table 4** Example of a comparison between actual volumes and farthest predicted values by our system along with their error percentages (Mean Error Percentage: 3.4%). This table compares actual volumes with their most extreme predicted values from Fig. 9., which shows predictions for Tips 1 to 8. For each row, the farthest predicted value that deviates the most from the actual volume is highlighted. The error percentages emphasize how much the farthest predicted value differs from the actual volume

| Actual Volumes | Farthest Predicted Value | Error Percentage |
|---|---|---|
| 50µl | 52µl | 4.0% |
| 70µl | 68µl | 2.9% |
| 90µl | 88µl | 2.2% |
| 110µl | 113µl | 2.7% |
| 130µl | 133µl | 2.3% |
| 150µl | 147µl | 2.0% |
| 170µl | 174µl | 2.4% |
| 190µl | 194µl | 2.1% |
| 210µl | 215µl | 2.4% |
| 230µl | 235µl | 2.2% |
| 250µl | 244µl | 2.4% |
| 270µl | 276µl | 2.2% |

error detection capabilities while maintaining low processing times. Unlike traditional high-end systems, our approach democratizes laboratory automation, providing an accessible alternative to costly proprietary solutions. The application of a closed-loop feedback mechanism ensures operational consistency, which is crucial for maintaining accuracy in experimental workflows. These advancements not only address immediate laboratory needs but also pave the way for broader adoption of intelligent automation in academic and resource-limited environments.

The intention of most efforts to automate laboratory work is to improve the quality of the produced results, by reducing the error rate in experiments, and secondly to increase the throughput of the given laboratory. Meaning that a workable solution should fulfil both goals. The analysis of images with the aim of detecting objects and evaluation of the observed conditions is a computational heavy task when relying on AI based analysis. The original YOLOv5 algorithm, developed by Yu [6], required approximately 8 s per frame for processing, which is impractical for high-throughput laboratory operations, as it significantly extended the time it takes to perform an experiment that includes multiple pipetting steps. However, the recently developed YOLOv8 offers much faster image processing time, down to just 1.1 s per frame, which now makes it feasible to use AI-based image processing in laboratory automation systems. YOLOv8 was selected for this study due to its optimal balance of speed, accuracy, and ease of deployment, which aligns with the demands of high-throughput laboratory automation. Unlike multi-stage algorithms such as Faster R-CNN, YOLOv8 processes images in a single pass, enabling real-time processing essential for detecting fine details in pipette tips and liquid volumes. Its user-friendly design and streamlined deployment make it accessible for integration without requiring extensive computational resources, as seen with frameworks like Mask R-CNN or EfficientDet [27, 28]. Additionally, YOLOv8's reliable performance in detecting small objects and variations, even in complex laboratory backgrounds, makes it an ideal choice for life science automation. This capability supports real-time monitoring of laboratory tasks, providing immediate visual feedback on the robot's performance while maintaining high accuracy.

In the current study, we chose to work with the reasonable praised Opentrons OT-2 platform, as the addition of a real-time quality monitoring system would significantly improve it capabilities and application scenarios. The limited processing power of the OT-2's on-board controller, a Raspberry Pi, mean that we had to distribute the needed data processing between the Raspberry Pi and the external PC work in series, where the Raspberry Pi requests image processing services from the PC. This setup ensures that the system remains robust and consistently operational, as the Raspberry Pi is freed from heavy processing tasks that it wasn't originally designed to handle. Additionally, the use of the external server further solved that challenge that the OT-2 Pi only allows for standard Python libraries, which were insufficient for object detection and image analysis tasks.

To train the YOLOv8 model for use in the detection of pipette tips and volume estimation, we first collected the needed image training dataset. In the data collection, we ensured that images were captured under different lighting conditions, including bright daylight and standard laboratory light. This approach aimed to simulate the range of lighting conditions the robot might encounter during typical laboratory operations. By capturing detailed and varied images, we trained our machine learning model on a comprehensive dataset, enhancing its accuracy in object detection and robustness in real-world applications. This thorough data collection process is foundational to the development of our AI-driven vision system, facilitating precise monitoring and control of the OT-2 robot's operations. We initiated training of the YOLOv8 model by testing various image batch sizes, and found the optimal size to be 16, as it allowed for efficient use of GPU memory while maintaining robust training. The model was subsequently trained for 50 epochs, which provided a good balance between training time and model accuracy. Our dataset assumes consistency in pipette type, liquid color, and volume variations, typical of most laboratory workflows. The developed model was found to perform well overall, but certain limitations were identified through an analysis of failure cases. Specifically, the pipette tips' long, thin, cylindrical shape led to instances where the model misidentified similar shapes in the background—such as curtain folds or hanging wires—as pipette tips. This mislabeling contributed to an increase in false positive detections, which impacted precision and reduced accuracy at higher IoU thresholds, particularly affecting mAP@95 scores. An error source we expect that be reduce, and possibly completely eliminated, by controlling the image background, e.g. by introducing a standardized background in the region of the robot where the image is recorded. Additionally, the model encountered challenges in detecting minimal liquid volumes within pipette tips, as the small liquid level sometimes blended visually with empty tips. This occasional misclassification impacted the detection accuracy in near-empty scenarios. These limitations point to potential areas for improvement, including enhanced background filtering and segmentation techniques to better differentiate pipette tips from similarly shaped objects. Future work may also focus on improving the model's ability to distinguish between minimal liquid levels and empty tips, which could further enhance its reliability in diverse laboratory settings.

Table 5 summarizes the significant improvements in key performance metrics, including mAP and operational capabilities, achieved by our computer vision system compared to the existing method [6]. The results reveal notable advancements in processing time, mAP@95, and recall, underscoring the effectiveness of our approach in automating quality control for the Opentrons OT-2 liquid handling robot. These enhancements not only increase the accuracy and efficiency of pipette tip and liquid detection but also ensure the system's practical applicability in real-world laboratory settings.

By integrating robust object detection models and implementing a server-client communication setup and relying on a training set captured under a series of real-life conditions, we ensured that our system could operate seamlessly in dynamic laboratory settings. The incorporation of real-time communication methods between the Raspberry Pi of the OT-2 robot and an external Python server facilitated efficient data exchange and command execution, which are essential for responsive and adaptive robotic operations.

**Table 5** Performance Comparison between Previous and Developed Methods

| Metrics | Previous method | Developed method |
|---|---|---|
| mAP@50 | 0.99 | 0.99 |
| mAP@95 | 0.81 | 0.88 |
| Recall | 0.98 | 0.99 |
| Processing time | 8 Sec | 1.1 Sec |
| Working in real life Lab | X | YES |
| Communication method | X | Real-time python server and client setup |
| Generation of log file | X | YES |

The pipette tip detection module offers potential for further development into a QC system capable of evaluating tip alignment. This could be particularly valuable when working with high-density SBS plates (e.g., 384-well plates), where the smaller well diameter increases the risk of misaligned tips colliding with the plate.

The current volume detection system achieves 95% accuracy, sufficient for detecting large pipetting errors. However, since most pipette manufacturers guarantee 99% accuracy for newly calibrated pipettes, additional training of the ML model or optimization of imaging conditions may be required to meet this standard.

In its current implementation of the developed quality control system simply stops the OT-2 robot, or requests human interventions, or log the incidence, if an error is detected. However, the developed system lays the ground for the creation of a reaction control-loop, where the detection of an error is followed by an automatic error correction response by the OT-2 robot (self-healing system). In the case where the volume is found to deviate too much from the expected, the OT-2 should repeat the pipetting step using the same source tubes. In the case of missing tips, the ideal would be to allow the system to check for tips before any pipetting is performed, and in cases where tips are missing to send instructions to the OT-2 to repeat the step where it collected the tips. By introducing the automated QC systems, we hope to reduce the number of situations, and time spent where scientist watch a robot work to ensure it correct operation.

In comparison to other high-end liquid handling systems from manufacturers such as Tecan and Hamilton, which often come with sophisticated and expensive quality control mechanisms, our developed method offers a more affordable solution. Commercial systems typically feature advanced sensors and proprietary software for quality control, requiring specialized expertise for operation and maintenance. Our approach, however, utilizes a web-camera and a PC to establish a closed-loop feedback control system. By enhancing the OT-2 with this cost-effective method, we offer a practical alternative for budget-conscious laboratories and low-cost pipetting robots that lack built-in quality control. This democratization of advanced liquid handling automation facilitates broader adoption and accessibility in various research settings.

## 5 Conclusion and future direction

This study presents a novel AI-driven computer vision solution that significantly enhances quality control within the OT-2 liquid handling platform, marking a notable advancement in life science laboratory automation. Our key findings demonstrate a mean average precision (mAP@50) of 0.995 and recall of 0.99, with processing times reduced to 1.1 s per frame, making the system viable for high-throughput tasks. These results underscore the AI model's robustness in accurately detecting and classifying objects, even in complex laboratory conditions. This work not only refines the operational capacity of existing robotics but also contributes theoretical insights into the application of CNN-based object detection for reliable quality control in life science automation.

Despite its promising performance, this study acknowledges certain limitations, particularly the dataset's focus on specific pipette types and liquid volumes. Future work will address these limitations by expanding the dataset to include a broader range of experimental conditions and additional liquid handling tasks. Furthermore, we aim to extend the system's quality control capabilities to detect challenges such as air bubbles, which could adversely affect liquid handling accuracy. We also plan to develop a web-based graphical user interface (GUI) for user-friendly operation and remote monitoring, along with a self-healing mechanism that allows the OT-2 to detect and correct errors autonomously. Further integration with AI-driven decision-making systems may broaden applications across various scientific fields, enhancing the usability and impact of affordable laboratory automation. This study demonstrates the adaptability of YOLOv8 for quality control in laboratory settings, extending the applicability of deep learning for real-time monitoring in life sciences R&D. The results contribute to a deeper understanding of closed-loop automation, laying a foundation for future advancements in robotic control, quality assurance, and AI-driven lab automation.

**Funding** Open access funding provided by Technical University of Denmark.

### Declaration

**Conflict of interest** The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

## References

1. Thurow K (2022) Automation for Life Science Laboratories. Adv Biochem Eng Biotechnol 182:3–22. https://doi.org/10.1007/10_2021_170
2. Preuße S (2013) Technologies for engineering manufacturing systems control in closed loop, Vol 10. Logos Verlag Berlin GmbH
3. Ferreira EC, Mesbah NM, Hillson N, Holland I, Davies JA (2020) Automation in the life science research laboratory. Front Bioeng Biotechnol. https://doi.org/10.3389/fbioe.2020.571777
4. Archetti C, Montanelli A, Finazzi D, Caimi L, Garrafa E (2017) Clinical laboratory automation: a case study. J Public Health Res 8:23–27
5. Christo BB, Jebarani MRE (2016) An automatic liquid dispensing robot with database management and biometric security systems. ARPN J Eng Appl Sci 11:1819–6608 (**Accessed 16 July 2024**)
6. Yu H, Møller VK, Frandsen RJN, Mansourvar M (2023) Image processing for robotic control in life science laboratory automation. In: 2023 5th International Conference on Robotics and Computer Vision (ICRCV). pp 305–309. IEEE. https://doi.org/10.1109/ICRCV59470.2023.10329251
7. Liu B, Che C, Hu H, Yu L, Lin Q, Zhao X (2024) Integration and performance analysis of artificial intelligence and computer vision based on deep learning algorithms. arXiv preprint arXiv:2312.12872
8. Hassaballah M, Awad AI (Eds) (2020) Deep learning in computer vision: principles and applications. CRC Press
9. Che C, Zheng H, Huang Z, Jiang W, Liu B (2024) Intelligent robotic control system based on computer vision technology. Preprint at https://arxiv.org/abs/2404.01116
10. Robinson N, Tidd B, Campbell D, Kulić D, Corke P (2023) Robotic vision for human-robot interaction and collaboration: a survey and systematic review. ACM Trans Hum Robot Interact 12:12. https://doi.org/10.1145/3570731
11. Ghoson NH, Hakam N, Shakeri Z, Meyrueis V, Loubère S, Benfriha K (2023) Towards remote control of manufacturing machines through robot vision sensors. Proc Des Soc 3:3601–3610. https://doi.org/10.1017/pds.2023.361
12. Ettalibi A, Elouadi A, Mansour A (2024) AI and computer vision-based real-time quality control: a review of industrial applications. Procedia Comput Sci 231:212–220. https://doi.org/10.1016/J.PROCS.2023.12.195
13. Chen Y, Goorden MC, Beekman FJ, Du J (2018) Understanding of object detection based on CNN family and YOLO. J Phys 12029. https://doi.org/10.1088/1742-6596/1004/1/012029
14. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
15. Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E (2018) Deep learning for computer vision: a brief review. Comput Intell Neurosci 2018:7068349. https://doi.org/10.1155/2018/7068349
16. Nazarkevych M, Kostiak M, Oleksiv N, Vysotska V, Shvahuliak A-T (2024) A YOLO-based method for object contour detection and recognition in video sequences. In CPITS pp 49–58
17. Reis D, Kupec J, Hong J, Daoudi A (2023) Real-time flying object detection with YOLOv8. Preprint at https://arxiv.org/abs/2305.09972
18. Hussain M (2023) YOLO-v1 to YOLO-v8, the rise of YOLO and Its complementary nature toward digital manufacturing and industrial defect detection. Machines 11:677. https://doi.org/10.3390/MACHINES11070677
19. Li H, Wu D, Zhang W, Xiao C (2024) YOLO-PL: helmet wearing detection algorithm based on improved YOLOv4. Digit Signal Process 144:104283. https://doi.org/10.1016/J.DSP.2023.104283

20. Moya V, Quito A, Pilco A, Vásconez JP, Vargas C (2024) Crop detection and maturity classification using a YOLOv5-based image analysis. Emerging Sci J 8:496–512. https://doi.org/10.28991/ESJ-2024-08-02-08
21. Ji CL, Yu T, Gao P, Wang F, Yuan RY (2024) Yolo-tla: an efficient and lightweight small object detection model based on YOLOv5. J Real Time Image Process 21:1–16. https://doi.org/10.1007/S11554-024-01519-4/TABLES/8
22. Guillermo M, et al. (2020, November). Implementation of automated annotation through mask RCNN object detection model in CVAT using AWS EC2 instance. In 2020 IEEE region 10 conference (TENCON) (pp. 708-713). IEEE, Osaka, Japan, 16-19 November 2020. https://doi.org/10.1109/TENCON50793.2020.9293906
23. Cheng X, Khomtchouk B, Matloff N, Mohanty P (2018) Polynomial regression as an alternative to neural nets. Preprint at https://arxiv.org/abs/1806.06850
24. Li X et al (2020) Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection. Adv Neural Inf Process Syst 33:21002–21012
25. Powers DM (2020) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Preprint at https://arxiv.org/abs/2010.16061
26. Padilla R, Passos WL, Dias TL, Netto SL, Da Silva EAB (2021) A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics 10:279. https://doi.org/10.3390/ELECTRONICS10030279
27. Hermens F (2024) Automatic object detection for behavioural research using YOLOv8. Behav Res Methods 56:7307–7330. https://doi.org/10.3758/S13428-024-02420-5/FIGURES/22
28. Sohan M, Sai Ram T, Reddy R, Venkata C (2024) A review on yolov8 and its advancements. In: international conference on data intelligence and cognitive informatics. Springer, Singapore, pp 529–545

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Sana Ullah Khan** possesses expertise in data analytics, artificial intelligence, and the automation of data-driven processes in life sciences. He earned his Master's degree in Business Data Analytics from the Technical University of Denmark (DTU). His research focuses on applying machine learning to life sciences automation, enhancing decision-making, optimizing data workflows, and improving process efficiency. Passionate about AI-driven solutions, he develops intelligent models to analyze and structure complex data while integrating automation systems. With a strong background in data analytics, he strives to contribute to innovative technologies that enhance accuracy and sustainability across various sectors.

**Vilhelm Krarup Møller** is an automation specialist focused on developing laboratory automation solutions to enhance throughput and reproducibility in microbiological workflows. He integrates liquid handling robots, LIMS software, and control systems to streamline operations and empower microbiologists in data-driven research.

**Rasmus John Normand Frandsen** Associate Professor Rasmus Frandsen is heading the Section for Synthetic Biology at DTU' s Department for Bioengineering. His work focuses on reducing the development of new biosolutions addressing the global resource crises caused by current agriculture and industrial production methods. Developing smarter and new agile laboratory automation solutions is key to achieving this goal. Dr Frandsen is also the founder and head of DTU's Arena for Life Science Automation (Dalsa/dtu.dk), an interdisciplinary research centre focused on new automation solutions for life science R&D and manufacturing.

**Marjan Mansourvar** is an Assistant Professor and leader of the Bio Data Science & Digitalization research group at DTU Bioengineering. With a background in computer science and data science, her research focuses on applying AI and automation to life sciences and biotechnology. Her work spans deep learning, bioinformatics, and laboratory automation, contributing to advancements in synthetic biology and bioprocess optimization. She has experience as a PI and co-PI on funded projects and actively collaborates with academia and industry to drive innovation in biotechnological research.