2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Detection of Jamming and Low Harvesting Height in Automated Cabbage Harvesting

Masaki Asano[1] and Takanori Fukao[2]

***Abstract*— Agricultural labor shortages have increased the demand for automation in farming. In cabbage harvesting, automated harvesters rely on a side-mounted camera for detection to control harvesting height, but occlusion from outer leaves can cause errors and lead to failures. This paper presents a robust detection and control framework that integrates YOLO-based cabbage detection, trajectory tracking, LSTM-based motion classification, and LiDAR point cloud analysis. The system functions as a fail-safe while also providing redundancy, enabling recovery when side-mounted camera detection fails, and addresses two critical failure modes: cabbage jamming during extraction and low harvesting height. Temporal motion features are classified by an LSTM, while LiDAR-based trajectory analysis of the cabbage head point cloud centroid identifies low harvesting height. When both jamming and low harvesting height are detected, the system issues a raising command to the harvester. Experiments on real-world data demonstrated 95.3% accuracy in jamming detection and 95% in low harvesting height detection. Field experiments confirmed real-time operation at 10 Hz and effective prevention of severe blockages, achieving an overall control accuracy of 97.0%. These results demonstrate the feasibility of the proposed method for robust automated cabbage harvesting.**

I. INTRODUCTION

The global agricultural sector faces a decline in workers due to an aging workforce and fewer new entrants. This trend is especially evident in developed countries such as Japan, Europe, and the United States, where the proportion of elderly agricultural workers continues to rise. In Japan, about 70% were aged 65 or older in 2019 [1], while in the EU many farm managers are over 55, with a large fraction 65 or older [2]. To address these shortages, agricultural automation has been actively pursued. Among these efforts, cabbage production has received significant attention, with studies on automated transplanting [3] and harvesting using side-mounted camera [4], although research specifically targeting automated cabbage harvesting remains limited.

A cabbage harvester (Fig. 1) employs a hydraulically controlled star-wheel mechanism (Fig. 2) to grip and extract cabbages by their stalks, enabling vertical movement for height adjustment. In automated systems, a side-mounted camera (Fig. 1a) detects cabbages using deep learning–based object detectors such as SSD [5], together with a red cross marker. The harvesting height is then controlled so that the vertical difference h between the marker bottom edge and that of a cabbage remains constant, aligning the unit with the stalk. This method generally works but can fail under outer leaf occlusion, and even a deviation of about 5 cm from the proper height can cause harvesting failure (Fig. 2b). Although newer detectors such as YOLO [6] improve accuracy, the side view remains fundamentally limited by occlusion.

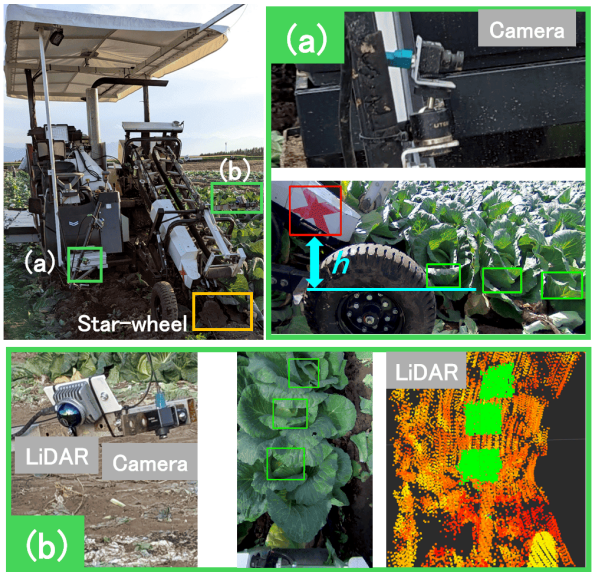


Fig. 1: Cabbage harvester with side and overhead cameras and LiDAR, together with cabbage detection results from each view.

(a) Star-wheel extracts cabbages by gripping their stalks.

(b) Harvesting unit stuck in the soil.

Fig. 2: Star-wheel mechanism and height control failure.

Thus, a reliable mechanism is required to detect low harvesting height and potential blockages before severe failures occur. To address this issue, we propose a robust approach that integrates image-based motion analysis with LiDAR-

[1]Masaki Asano is with the Department of Mechano-Informatics at the Graduate School of Information Science and Technology, The University of Tokyo, Japan asano@ynl.t.u-tokyo.ac.jp
[2]Takanori Fukao is with the same department.

based height estimation. The system functions as a fail-safe while also providing redundancy, enabling continued operation even when side-mounted camera detection fails. The side-mounted camera (Fig. 1a) is typically used for stalk alignment and height control, but its detection can be unstable under outer leaf occlusion. To overcome this, we introduce a complementary overhead camera (Fig. 1b), which provides a clearer view of the cabbage head for reliable tracking and failure detection.

The pipeline has two stages. First, cabbages observed by the overhead camera are tracked, and their motion trajectories are analyzed. A Long Short-Term Memory (LSTM) [7] network classifies these trajectories as normal or jamming, capturing temporal motion patterns beyond single-frame detection. Second, when jamming is detected, LiDAR point clouds are used to estimate the vertical trajectory of the cabbage head point cloud. A shallow trajectory slope indicates that the harvesting unit is operating too low, in which case a height raising command is issued to prevent soil entrapment and blockage. While RGB-D cameras can also provide depth information, their measurements are significantly affected by outer leaf occlusion, resulting in unstable height estimation. LiDAR, in contrast, provides more robust point cloud data under occlusion, which is why it was adopted in this study.

The key contributions of this work are:

- A vision–LiDAR fusion framework that integrates overhead-camera tracking with LiDAR analysis for robust failure detection, validated on datasets and in real harvesting experiments.
- An LSTM-based motion analysis approach using optical flow features from tracked cabbages to reliably detect jamming events.
- A LiDAR point cloud filtering pipeline applied to cabbages flagged as jammed, enabling trajectory estimation and low harvesting height detection, with field experiments confirming effective height control and successful harvesting.

The remainder of this paper is organized as follows. Section II describes jamming and low harvesting height detection, Section III presents height control of the harvesting unit, Section IV reports experiments, and Section V concludes the paper.

II. Detection Methods of Jamming and Low Harvesting Height

This section describes the proposed method for detecting cabbage jamming during harvesting and identifying cases of low harvesting height operation. An overview of the processing flow is illustrated in Fig. 3. The method consists of two main components: (1) cabbage tracking and jamming detection based on image analysis, and (2) detection of abnormally low harvesting height using LiDAR-based point cloud analysis

A. Cabbage Jamming Detection

To detect harvesting failures, we first track the movement of cabbages as they pass through the harvesting process. The

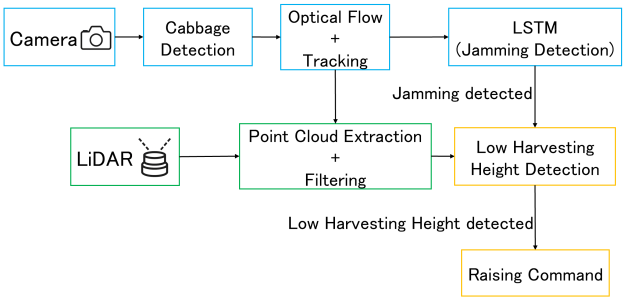


Fig. 3: Overview of the proposed method for detecting cabbage jamming and low harvesting height.

star-wheel mechanism should smoothly extract cabbages by gripping their stalks, but in some cases, cabbages exhibit movement that is not smooth and becomes jammed during extraction. This jamming-like behavior is characterized by irregular motion patterns, with cabbages seemingly stuck or hindered in their path, and by a noticeable reduction in the normal extraction speed at which the cabbage is pulled out for harvesting.

1) Relative Cabbage Motion Estimation Using Optical Flow: An overhead camera mounted above the harvesting unit (Fig. 1b) is used to detect and track cabbages as they pass through the harvesting process. Object detection is performed using a YOLOv11 model. The bounding boxes of detected cabbages are then linked across consecutive frames using the SORT (Simple Online and Realtime Tracking) algorithm [8], which provides a lightweight yet robust tracking framework. Fig. 4a shows an example of cabbage tracking. The tracking region is indicated by the cyan arrow bounded by white dashed lines, corresponds to the range in front of and behind the star-wheel ($-0.1 < x < 0.1$), within which the trajectories of cabbages are extracted and analyzed.

SORT employs a linear Kalman filter to predict the state of each track (center position, velocity, and bounding box size) from frame $t-1$ to t. Detections at frame t are then associated with predicted tracks using the Hungarian algorithm with an IoU-based cost. Unmatched tracks are updated by prediction only, and unmatched detections start new tracks. Apart from a few hyperparameters (e.g., maximum track age, minimum hit count), we used the default settings in [8]. While DeepSORT [9] incorporates appearance features for crowded scenes, such complexity is unnecessary here since cabbages are arranged in a single row. Thus, standard SORT provides an efficient and sufficient solution. Combined with YOLOv11 detection, it yields real-time cabbage trajectories for subsequent jamming detection and height analysis.

To evaluate the displacement of cabbages extracted by the star-wheel, their motion must be computed relative to the harvester; otherwise, vehicle speed would distort the measurements. We therefore estimate global harvester motion via optical flow and subtract it from the observed cabbage motion. Feature points are detected with Shi–Tomasi [12] and tracked using Lucas–Kanade [13], as shown in Fig. 4b. Points in the upper image region (white box in Fig. 4b), where

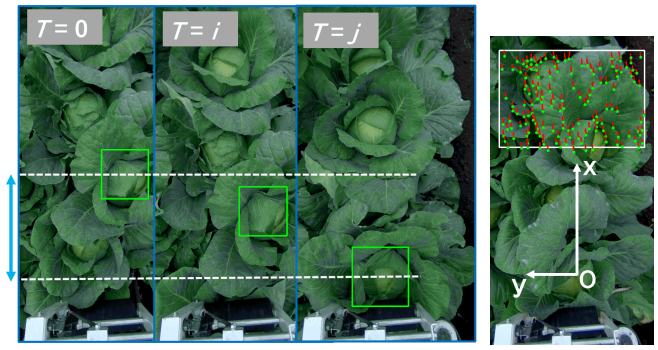
cabbages remain unharvested, serve as a stable reference for harvester motion. Although deep-learning methods such as FlowNet2 or RAFT achieve high accuracy, they require heavy computation and domain-specific training. In our setup with a fixed overhead camera and small inter-frame motion, the classical Shi–Tomasi + Lucas–Kanade approach is both sufficient and lightweight for real-time use. Given the optical flow displacement (u_i, v_i) of each tracked feature point i, the harvester's global motion $(dx_{\text{base}}, dy_{\text{base}})$ is computed as the average displacement of feature points in this stable region S:

$$(dx_{\text{base}}, dy_{\text{base}}) = \frac{1}{N} \sum_{i \in S} (u_i, v_i), \tag{1}$$

where N is the number of feature points in S. The relative displacement of a cabbage is then obtained by subtracting the estimated harvester motion from the displacement of the cabbage's feature points. Specifically, if (dx, dy) denotes the displacement of the centroid of a cabbage bounding box between consecutive frames, the relative motion is given by:

$$(dx_r, dy_r) = (dx - dx_{\text{base}}, dy - dy_{\text{base}}). \tag{2}$$

This relative displacement represents the true motion of the cabbage caused by the star-wheel mechanism pulling its stalk, independent of the harvester's forward movement. Here, (dx, dy) and $(dx_{\text{base}}, dy_{\text{base}})$ are defined in the LiDAR-aligned image coordinate system, where the origin is set at the center of the star-wheel and the x-axis runs along the cabbage extraction direction ($x = 0.1$ corresponds to the tip of the star-wheel).



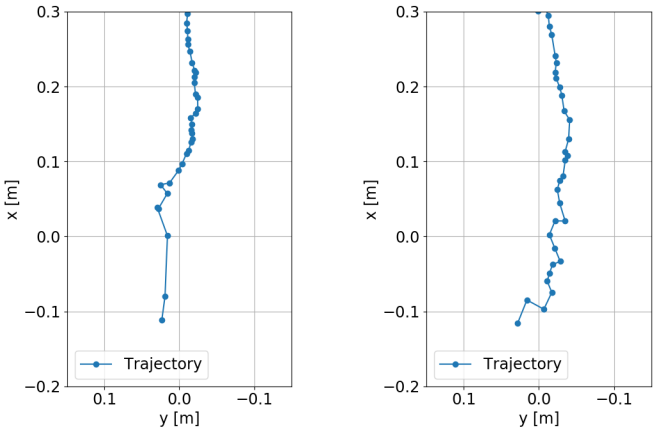
(a) Cabbage tracking. (b) Optical flow.

Fig. 4: Cabbage tracking and optical flow visualization. (a): The values T represent timestamps, where $0 < i < j$. (b): The red lines represent motion vectors over multiple frames.

To quantify jamming, we introduce the motion-based variable L, derived from the relative displacement as follows:

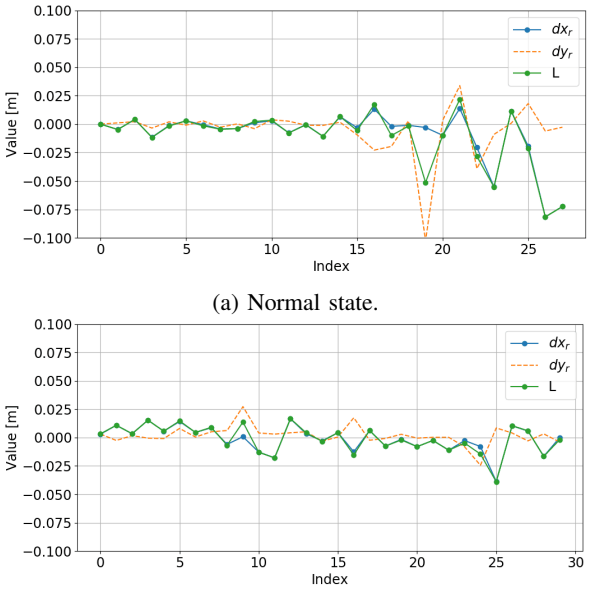
$$L = \text{sgn}(dx_r) \cdot \|(dx_r, \omega \cdot dy_r)\| \tag{3}$$

where $\text{sgn}(\cdot)$ is the sign function that returns 1 for positive input, -1 for negative input, and 0 otherwise. Here, ω is a weighting factor introduced because the primary extraction motion occurs in the x-direction. The tracked cabbage trajectory is shown in Fig. 5, where the centroids of the detected cabbage bounding boxes are plotted for each frame. The cabbage moves in the negative x-direction. In the normal case (Fig. 5a), when the cabbage reaches the star-wheel position ($x = 0.1$), it is correctly grasped and smoothly extracted, leading to noticeable acceleration and an increase in relative displacement. In contrast, in the jamming case (Fig. 5b), even after reaching the star-wheel, the cabbage is not smoothly extracted. Its forward displacement in the x-direction remains small, so the relative motion L stays close to zero, indicating that the cabbage is not being effectively pulled out.



(a) Normal state. (b) Jamming state.

Fig. 5: Visualization of cabbage trajectory.



(a) Normal state.

(b) Jamming state.

Fig. 6: Inter-frame cabbage displacement.

2) Jamming Detection using LSTM: The tracked cabbage trajectory is analyzed using an LSTM network to classify whether the cabbage is being smoothly extracted or in a jamming state. LSTM is chosen for its ability to capture temporal dependencies while mitigating vanishing gradient problems, which affect standard RNNs [10]. While 1D-CNNs [11] are computationally efficient, they have a limited receptive field, making them less suitable for sequential

dependencies. In this study, the tracked trajectory length is limited to about 30 frames, resulting in a relatively short input sequence. Given this sequence length, LSTM provides a good balance between accuracy and efficiency, making it suitable for jamming detection. The system learns motion patterns over time and detects abnormal extraction behavior indicative of cabbage jamming.

In this study, we used L as the primary feature for jamming detection. Fig. 6a and Fig. 6b illustrate the calculated values of L, shown as the green plot. The input to the LSTM model is the sequence of computed L values:

$$X = \{L_1, L_2, ..., L_T\}, \tag{4}$$

where L_t represents the motion-based jamming feature at time step t. The LSTM model processes the extracted motion sequence and predicts whether the cabbage is in a normal state or in a jamming state. The model consists of several components. First, the input layer receives the extracted feature sequence X. The LSTM layers then capture temporal dependencies in the motion sequence, allowing the model to learn patterns over time. The output from the LSTM layers is passed through a fully connected layer, which maps the learned representations to a binary classification output. Finally, a softmax function is applied to produce the final probability $P(jamming)$, indicating the likelihood of jamming occurrence.

The LSTM cell updates are defined as:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \tag{5}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \tag{6}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c), \tag{8}$$

$$h_t = o_t \odot \tanh(c_t), \tag{9}$$

where i_t, f_t, o_t are the input, forget, and output gates, c_t is the cell state, h_t is the hidden state, and W_*, U_*, b_* are the weight matrices and biases. The function $\sigma(\cdot)$ is the sigmoid activation function, mapping any input to a value between zero and one. It regulates the gate activations i_t, f_t, o_t, controlling information flow in and out of the LSTM cell. The final classification decision is made using the output probability:

$$P(jamming) = \text{Softmax}(W_h h_T + b_h), \tag{10}$$

where W_h and b_h denote the weight matrix and bias vector of the output layer, respectively. These parameters are optimized during training through backpropagation based on the classification loss. If $P(jamming)$ exceeds a predefined threshold θ, the cabbage is classified as jamming. The decision threshold was determined by cross-validation, selecting the point on the Precision–Recall curve that maximized the F1 score. This procedure provides a criterion that balances recall and precision.

B. Low Harvesting Height Detection

Jamming detected by image-based trajectory analysis does not necessarily imply that the harvesting unit is positioned too low. Even when the unit height is appropriate, outer leaves can obstruct the harvesting mechanism and cause jamming-like behavior. Therefore, after the LSTM-based detection flags a cabbage as being in a jamming state, the system performs an additional check to determine whether the cause is truly an abnormally low harvesting height. For this purpose, we analyze the point cloud trajectory of cabbages using a LiDAR sensor mounted above the harvesting unit. In normal cases, cabbages move rapidly and the point cloud of the cabbage head is unstable, making reliable trajectory estimation difficult. In contrast, when jamming is detected, the motion is slower and less affected by outer leaves, allowing the cabbage head point cloud to be stably extracted. By estimating the slope of this trajectory, the system distinguishes between normal operation and cases where a low harvesting height is observed.

1) Point Cloud Extraction and Filtering: To extract cabbage point clouds, LiDAR points are projected into the camera image plane using calibrated extrinsic and intrinsic parameters. A LiDAR point $P_L = (X_L, Y_L, Z_L)^\top$ is first transformed to the camera frame:

$$P_c = R_{LC} P_L + T_{LC}, \tag{11}$$

where $R_{LC} = R_{CL}^{-1}$ and $T_{LC} = -R_{CL}^{-1} T_{CL}$. The transformed coordinates $P_c = (X_c, Y_c, Z_c)^\top$ are then projected with the intrinsic matrix K. Points with image coordinates (u_c, v_c) that fall inside the detected bounding box ($u_{\min} \le u_c \le u_{\max}$, $v_{\min} \le v_c \le v_{\max}$) are retained as valid cabbage points. This projection-based filtering extracts only LiDAR points within the detected region (Fig. 1b).

To ensure robust tracking, the cabbage head point cloud is refined by filtering out outer leaf points, which otherwise destabilize the centroid due to their irregular structure and motion. Targeted filtering improves centroid consistency and yields more accurate trajectory estimation. Examples of cabbage and corresponding point clouds from different viewpoints are shown in Fig. 7. The following methods are applied (Fig. 8):

1) **Voxel-based filtering**: Points are voxelized, and only those with the lowest Z are retained to remove upper-layer noise such as outer leaves (red points in Fig. 8).
2) **Hemispherical filtering**: A hemisphere is defined to exclude peripheral points. Let $z_{\min}$ be the minimum Z of the cloud and (x_c, y_c) the centroid of the XY range. The hemisphere is centered at $(x_c, y_c, z_{\min})$ with radius

$$r = \tfrac{1}{2}\sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2}. \tag{12}$$

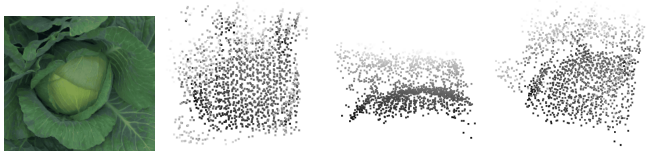
Points are retained if

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_{\min})^2 \le r^2, \quad z \ge z_{\min}. \tag{13}$$

Since the cabbage head can be approximated as an ellipsoid with a nearly circular top view, this hemispherical filtering reliably retains the core region while excluding peripheral leaves, thereby improving tracking accuracy (green points in Fig. 8).

3) **Euclidean clustering**: The filtered cloud is segmented using Euclidean clustering [14], and only the largest cluster is kept as the cabbage head point cloud, discarding small clusters from noise or detached leaves (blue points in Fig. 8).

Finally, the centroid of the cabbage head point cloud can be obtained as shown by the pink points in Fig. 8. The parameters of these filtering steps were empirically selected to match the typical scale of cabbage heads and the spatial resolution of the LiDAR.



(a) Cabbage (b) Top view (c) Side view (d) Oblique view

Fig. 7: Visualization of cabbage and corresponding point clouds from different viewpoints.

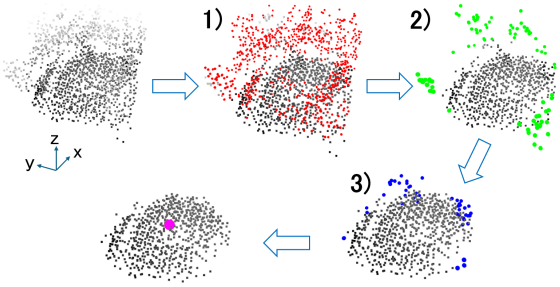


Fig. 8: Cabbage head point cloud filtering (steps 1-3). Red, green, and blue points indicate those removed at each step.

2) Low Harvesting Height Detection from Trajectory Slope: Fig. 9 shows an overview of the original cabbage point clouds together with the centroid points used for tracking. The 3D visualization demonstrates the trajectories of multiple cabbages after the Point Cloud Extraction and Filtering process. By focusing on the centroid points of the cabbage head point cloud, the system achieves stable trajectory estimation that is less affected by outer leaf noise, which is essential for the subsequent slope-based low harvesting height detection. The centroid trajectory is then analyzed using a RANSAC-based linear regression [15] on the XZ-plane, as shown in Fig. 10. The trajectory is modeled as

$$z = mx + b, \tag{14}$$

where the slope m represents the ratio $\Delta Z/\Delta X$. This slope serves as an indicator of the harvesting height. A steep slope indicates that cabbages move upward as they are extracted, whereas a small slope suggests that the harvesting unit is positioned too low, limiting upward motion and increasing the risk of soil entrapment. If $m < m_{th}$, the unit is considered too low. Here, m_{th} is a tunable threshold parameter. As shown in Fig. 10a, when the harvesting height is normal, the slope of the cabbage trajectory is steep. Conversely, as shown in Fig. 10b, the slope becomes small when the harvesting unit is too low.

By combining jamming detection with slope-based low harvesting height detection, the system can determine whether a detected jamming case is truly caused by the harvesting unit being positioned too low.

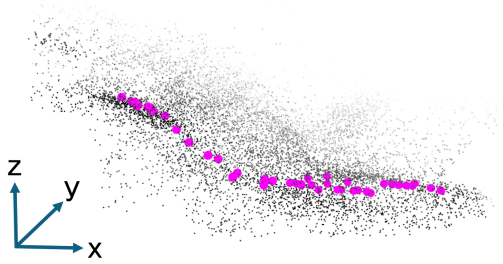
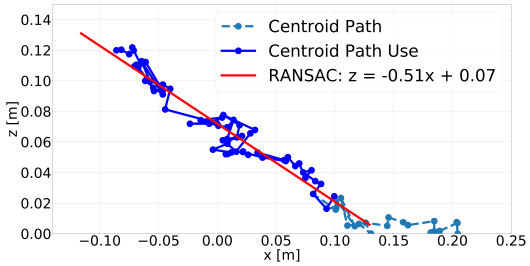
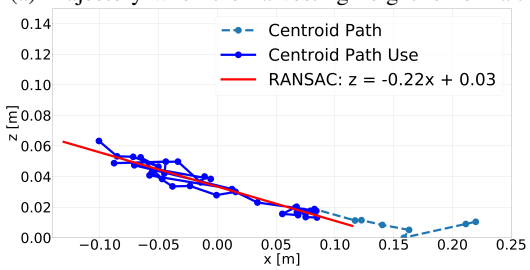


Fig. 9: 3D visualization of cabbage point clouds and head centroids after filtering.



(a) Trajectory when the harvesting height is normal.



(b) Trajectory when the harvesting height is too low.

Fig. 10: Centroid trajectory of the cabbage head point cloud in the XZ plane.

III. Height Control of Harvesting Unit

This section describes the height control method of the harvesting unit that is applied when a low harvesting height is detected. Using the proposed detection method, when a cabbage is judged to indicate a low harvesting height, a command to raise the harvesting unit is issued in order to prevent harvesting failure. The decision is made for each cabbage based on its tracked trajectory and point cloud information. To improve robustness, the control command is not triggered by a single detection result alone. We define the detection result at frame t as

$$d_t \in \{0, 1\}, \tag{15}$$

where $d_t = 0$ indicates normal height and $d_t = 1$ indicates that the unit is too low. The control output is defined as

$$u_t \in \{0, 1\}, \tag{16}$$

where $u_t = 1$ corresponds to issuing a raising command

and $u_t = 0$ corresponds to no command. Here, the control focuses only on correcting cases where the harvesting unit is judged to be too low. Therefore, the output is binary (0 or 1): either to raise the unit or to keep it unchanged. Lowering of the harvesting unit is not considered in this framework.

The control law is expressed as a state transition rule:

$$u_t = \begin{cases} 1 & \text{if } (d_{t-1} = 1 \wedge d_t = 1), \\ 1 & \text{if } (d_{t-2} = 1 \wedge d_{t-1} = 0 \wedge d_t = 1), \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

A raising command is issued when low harvesting height is detected in consecutive frames ($1 \rightarrow 1$) or when a single normal detection is sandwiched between two low detections ($1 \rightarrow 0 \rightarrow 1$); otherwise, no command is given. This temporal aggregation suppresses false positives that could cause unnecessary height increases and potential damage to the cabbage head.

The harvesting unit is actuated by hydraulic control, which enables vertical adjustment. Although the height command is binary due to actuator limitations, repeated commands are used to approximate a quasi-continuous adjustment. As the harvester grips and pulls the stalk without cutting, precise attitude control is not required and is outside the scope of this work. Fig. 11 illustrates the control scheme, where multiple consecutive raising commands are issued to achieve the required vertical displacement.

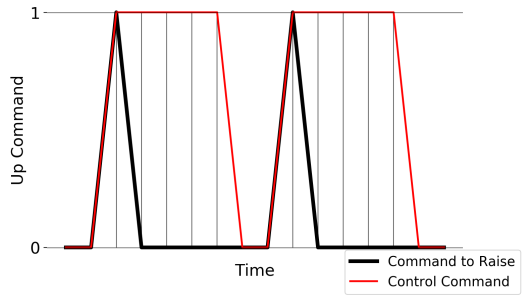


Fig. 11: Height control scheme: black = raw raising signals, red = raising commands sent to the harvester.

IV. Experiments and Field Validation

To validate the effectiveness of the proposed methods, we conducted both offline evaluation using recorded datasets and real-time field experiments on an actual cabbage harvester. The experiments were carried out in a cabbage field located in Shikaoi, Hokkaido, Japan, where the proposed system was integrated into a commercial harvesting machine. The overall hardware configuration and sensor installation are shown in Fig. 1.

A. Experimental Setup

The experiments used a Yanmar HC-1400 harvester equipped with a Leopard Imaging LI-AR0231-AP0200-GMSL2-60H RGB camera and a Livox Mid-360 LiDAR mounted above the harvesting unit (Fig. 1). The RGB camera provided cabbage detection and tracking, while the LiDAR supplied point clouds for low harvesting height detection. The Mid-360 provides high-density point clouds with about 20,000 points per frame at 10 Hz.

Onboard computation was distributed across a Jetson AGX Orin (64 GB) for YOLO-based detection and LSTM classification, and an Intel NUC 14 Pro (Core Ultra 7) for trajectory tracking, LiDAR processing, height control decisions, and command transmission. Control signals were sent via CAN to the harvester's hydraulic system to adjust unit height. The system ran on ROS, enabling real-time communication between perception, decision, and actuation modules.

B. Validation with Recorded Dataset

1) Cabbage Jamming Detection: Cabbage detection was conducted using the YOLOv11s model provided by Ultralytics, pre-trained on the COCO dataset and fine-tuned on a custom dataset collected in Shikaoi and Furano (Hokkaido) and at Fukuhara Farm (Shiga), consisting of 1,214 annotated images. The detector achieved a precision of 0.948, a recall of 0.911, and an mAP@0.5 of 0.97, providing reliable inputs for downstream motion analysis.

For jamming detection, the motion-based feature L was computed with a weighting factor $\omega = 0.5$, emphasizing displacements along the harvester's moving x-direction. The extracted sequences were fed into a three-layer LSTM with hidden state size 512 and sequence length $T = 30$. The model was trained on 1000 tracked cabbage trajectories collected in Shikaoi, Hokkaido, and the decision threshold was determined via stratified 5-fold cross-validation, yielding an operational value of $\theta = 0.4$. Evaluation on the held-out test set (197 normal, 101 jamming) resulted in an accuracy of 95.3%, precision of 0.922, recall of 0.941, and F1-score of 0.931, as summarized in Table I.

TABLE I: Confusion matrix for LSTM jamming detection at $\theta = 0.4$.

Actual / Predicted	Normal	Jamming
Normal	190	7
Jamming	6	95

The LSTM classifier achieved high performance overall. The few misclassified cases exhibited specific characteristics. As shown in Fig. 12, a cabbage in a jamming state was incorrectly classified as non-jamming. These failures mainly occurred when the cabbage posture changed while being grasped and lifted by the star-wheel. Such posture variations altered the apparent centroid trajectory and made the inter-frame displacement appear larger or more irregular than the actual motion, distorting the computed L features and leading to misclassification.

For reference, we also implemented a vanilla RNN and a 1D-CNN. The RNN underperformed (F1 $\approx$ 0.73, recall $\approx$ 0.76) due to unstable training, while the 1D-CNN achieved balanced results (F1 $\approx$ 0.83) but lower recall than the LSTM. Given that high recall is critical to avoid overlooked jamming events, the LSTM was selected as the primary classifier.

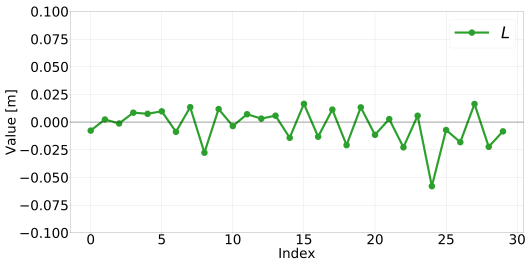


Fig. 12: Example of a failed cabbage jamming detection.

Other lightweight temporal models such as TCNs may also be effective for jamming detection and will be investigated in future work.

2) Low Harvesting Height Detection for Jamming Cases: Camera–LiDAR calibration was performed with a checkerboard target. Camera intrinsics were obtained by standard calibration, and extrinsics (R_{LC}, T_{LC}) by aligning checkerboard corners with LiDAR planes, enabling accurate projection of LiDAR points into the image plane for extracting cabbage point clouds within detected bounding boxes.

Low harvesting height detection was applied only to cabbages first classified as jamming by the LSTM module. For each cabbage, the extracted point cloud was filtered to remove outer leaves, and the centroid trajectory in the XZ-plane was fitted by RANSAC-based linear regression. The slope m of the fitted line indicated harvesting height, with a threshold of $m_{\text{th}} = 0.3$, corresponding to the star-wheel trajectory angle (about 20°) under normal height. Among 102 jamming cases (57 low, 45 normal), the method achieved 95% accuracy and 96% recall for low-height cases (Table II). Failure cases mainly arose from (1) complete occlusion of the cabbage head by outer leaves, and (2) posture changes during extraction by the star-wheel, which distorted the head point cloud and produced inconsistent slopes (Fig. 13). Although a force sensor at the harvesting unit tip could be considered, it would also react to outer leaves and soil, making low harvesting height detection unstable.

TABLE II: Low harvesting height detection results.

Metric	Performance
Accuracy	95% (97/102)
Recall (Low harvesting height detection)	96% (55/57)

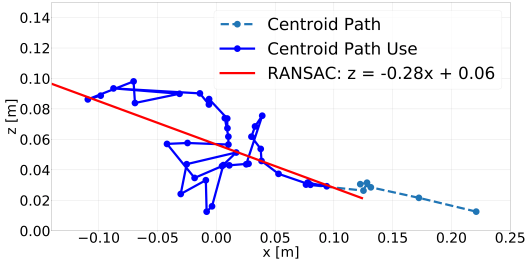


Fig. 13: Failed low harvesting height estimation due to posture change and leaf occlusion.

C. Real-time Field Experiments on Harvester

To validate the proposed system in real harvesting operations, We implemented the pipeline on the harvester and evaluated its real-time performance, motion estimation, and height control while keeping the harvesting height low and without using side-mounted camera detection, to simplify validation.

1) Execution Time Analysis: All modules operated in a 10 Hz pipeline. YOLOv11s-based detection took 35 ms per frame, while LSTM classification and low harvesting height detection required 15 ms and 30 ms, respectively. The total cycle time remained well below 100 ms, satisfying the 10 Hz requirement. Given the harvester speed of 0.1 m/s and plant spacing of 0.35 m, this cycle rate provides sufficient responsiveness for real-time harvesting.

2) Optical Flow Validation: To validate optical-flow-based vehicle motion estimation, we compared the estimated inter-frame displacement with GNSS velocity measurements. As shown in Fig. 14, the optical flow results (blue solid) closely matched the GNSS profile (orange dashed), demonstrating reliable frame-to-frame displacement estimation and global motion compensation.

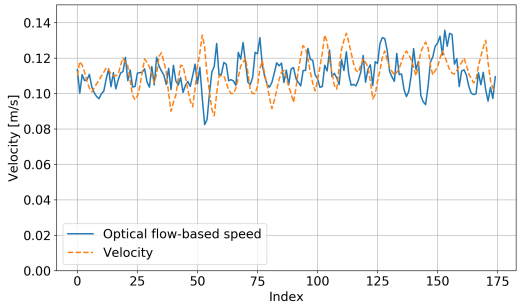


Fig. 14: Comparison of vehicle velocity estimated by optical flow (blue solid) and GNSS (orange dashed).

3) Height Raising Control on Harvester: The proposed system was integrated with the harvester's hydraulic actuator to raise the harvesting unit when both jamming and low harvesting height were detected. Fig. 15 illustrates an example of the harvesting unit being raised from a low state to a proper height. For clarity, a blue dot is drawn at the center of the marker, and the unit is controlled to move upward by approximately 5 cm in the direction indicated by the red arrow. We evaluated a total of 165 events, consisting of 102 positive cases (low harvesting height) and 63 negative cases (normal height). The confusion matrix and performance metrics are summarized in Table III. From these

TABLE III: Performance of harvesting unit height raising control (165 events).

Actual / Predicted	Raise	No Raise
Low (Positive)	99	3
Normal (Negative)	2	61

results, the system achieved an overall accuracy of 97.0% and an F1-score of 97.5%, demonstrating that the proposed

method can reliably detect low harvesting height and issue effective raising commands in real time using detection results from multiple cabbages. Fig. 16 shows the event-wise results of low harvesting height detection (0 or 1) together with the timing of raising commands. The red vertical lines indicate the moments when a raising command was issued to the harvester. The horizontal axis represents the index of each detected cabbage. This visualization confirms that the system correctly triggered height raising at appropriate events while ignoring normal cases. This indicates a fail-safe behavior while also providing redundancy: LSTM-based jamming detection combined with LiDAR analysis restores proper harvesting height and enables continuous operation. Moreover, simultaneous operation with side-mounted camera height control caused no adverse effects, and over 100 cabbages were successfully harvested.

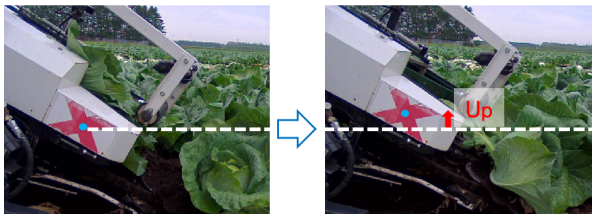


Fig. 15: Example of harvesting unit height control: raised from low to proper height in the field.

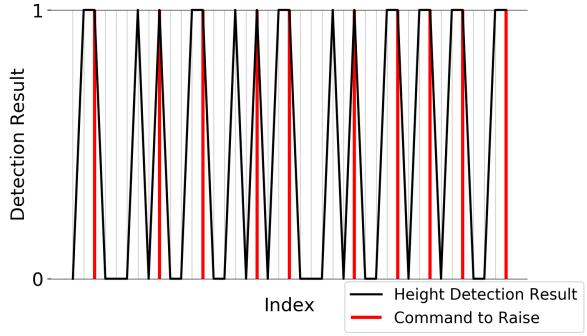


Fig. 16: Event-wise detection (0: normal, 1: low height) and timing of raising commands (red lines).

V. CONCLUSIONS

This study proposed a robust detection and control framework for automated cabbage harvesting, addressing two critical conditions: jamming during extraction and low harvesting height. The system integrates YOLO-based cabbage detection, trajectory tracking with LSTM-based motion analysis, and LiDAR-based trajectory analysis of the cabbage head point cloud centroid to issue timely raising commands. Experiments with recorded datasets and real-world field experiments demonstrated real-time operation at 10 Hz, achieving 95.3% accuracy in jamming detection, 95% in low harvesting height detection, and an overall control accuracy of 97.0%.

The proposed framework functions as a fail-safe while also providing redundancy, restoring proper harvesting height even when side-mounted camera detection fails and enabling continuous operation. The method targets severely low harvesting states that cause jamming, while moderately low cases that still allow smooth harvesting are judged as normal. Although this limits sensitivity, it is sufficient to prevent complete blockages that could immobilize the harvester. Failure cases were mainly due to outer leaf occlusion and posture changes during extraction. Future work includes enhancing height control stability by incorporating stalk position estimation beyond side-view cabbage detection.

Although validated in cabbage harvesting, the proposed framework is not crop-specific and can be extended to other vegetables with similar harvesting challenges, such as broccoli and radish, indicating its broader applicability to automated harvesting systems.

ACKNOWLEDGMENT

This work was conducted as part of the MAFF-funded research program “Smart Production System SOP Development”, implemented by the National Agriculture and Food Research Organization (NARO), and was supported by the Project of the Bio-oriented Technology Research Advancement Institution (BRAIN), Japan (Grant No. JPJ011397).

REFERENCES

[1] USDA Foreign Agricultural Service, “Foreign Farm Labor ’ s Role Growing in Japan,” GAIN Report Number JA2020-0066, Tokyo, Japan, 2020.
[2] Eurostat, “Farmers and the agricultural labour force statistics,” European Commission, Statistics Explained, 2020.
[3] M. Asano and T. Fukao, “LiDAR-based Robot Transplanter,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 15818–15824, 2024, doi: 10.1109/ICRA57147.2024.10611087.
[4] M. Asano, K. Onishi, and T. Fukao, “Robust cabbage recognition and automatic harvesting under environmental changes,” *Advanced Robotics*, vol. 37, pp. 960–969, 2023.
[5] W. Liu, D. Anguelov, D. Erhan, et al., “SSD: Single shot multibox detector,” in *Proc. European Conf. Computer Vision (ECCV)*, pp. 21–37, Oct. 2016.
[6] G. Jocher and J. Qiu, “Ultralytics YOLO11,” version 11.0.0, Ultralytics, 2024. [Online]. Available: `https://docs.ultralytics.com/models/yolo11/`
[7] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[8] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pp. 3464–3468, 2016.
[9] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pp. 3645–3649, 2017.
[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in *NIPS Workshop on Deep Learning*, 2014.
[11] S. Kiranyaz, T. Ince, O. Abdeljaber, et al., “1-D convolutional neural networks for signal processing applications,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8360–8364, 2019.
[12] J. Shi and C. Tomasi, “Good features to track,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 593–600, 1994, doi: 10.1109/CVPR.1994.323794.
[13] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, pp. 674–679, 1981.
[14] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 1–4, 2011.
[15] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.