https://doi.org/10.17221/29/2026-RAE

Enhancing the performance of lotus seed kernel and shell sorting after dehulling: A real-time approach based on deep learning and image processing (YOLOv8)

Dinh-Tu Nguyen[1]*, *Huu-Phat Tran*[2], *Duc-Tin Le*[1], *Hoai-Tan Nguyen*[3], *Thanh-Thuong Huynh*[3]

[1]*Faculty of Mechanical Engineering, Can Tho University of Technology, Can Tho, Vietnam*
[2]*Department of Mechanical Engineering, College of Engineering, National Central University, Tao Yuan, Taiwan*
[3]*Faculty of Mechanical Engineering, College of Engineering, Can Tho University, Can Tho, Vietnam*

**Corresponding author: ndtu@ctuet.edu.vn*

Citation: Nguyen D.T., Tran H.P., Le D.T., Nguyen H.T., Huynh T.T. (2026): Enhancing the performance of lotus seed kernel and shell sorting after dehulling: A real-time approach based on deep learning and image processing (YOLOv8). Res. Agr. Eng., 72: 181–194.

Abstract: Lotus seeds represent a high-value agricultural commodity, widely utilised in the food and pharmaceutical industries for their nutritional and medicinal benefits. However, post-harvest processing, specifically the removal of shells, often relies on manual labour or rudimentary equipment that results in low productivity. This study proposes a computer vision system based on the YOLOv8 deep learning framework to automate and enhance the efficiency of post-decortication sorting. The model is designed to detect and classify three distinct components: unshelled seeds, kernels, and shell fragments. Based on these predictions, the system controls pneumatic actuators to perform precise, real-time separation. Trained on a dataset of more than 30 000 images, the model achieved a mean average precision (*m*AP@0.5) of 92.6%. These results validate the model's capability to accurately identify lotus seed components under high-speed processing conditions. Consequently, this research presents a feasible technological solution to mitigate the labour dependence and optimise the shell residue removal stage in industrial lotus seed processing.

Keywords: computer vision; deep learning; fresh lotus seed shelling; post-harvest automation; shell-kernel lotus seed separation; smart and precision agriculture; YOLOv8n

The lotus (*Nelumbo nucifera*) is an economically important crop in many Asian countries, particularly Vietnam, where it is widely cultivated for its seeds, rhizomes, and flowers (Paudel and Panth 2015; Tu et al. 2020). Lotus seeds are widely acknowledged for their nutritional attributes and their longstanding use in traditional medical practices across Asia (Paudel and Panth 2015), as well as for their role as a food ingredient. They provide substantial amounts of proteins, carbohydrates, and essential minerals, supporting their contribution to the dietary quality (Zhang et al. 2015, 2023; Sneh et al. 2022). In addition, lotus seeds contain diverse bioactive constituents that are associated with potential health-related effects, including antioxidant, anti-inflammatory, and immunomodulatory activities (Liao and Lin 2011). Owing to their broad culinary applicability and prospective use in functional food formulations, lotus seeds have become a relevant plant-based resource for both food and health-oriented applications (Jaha et al. 2026). Although official statistics

https://doi.org/10.17221/29/2026-RAE

on the lotus seed yield are often aggregated under aquatic vegetables, making specific data scarce, local reports from major production hubs in Vietnam confirm the consistent expansion in the cultivation area driven by a high market demand (Nguyen et al. 2024). As of 2023, the lotus cultivation area in Dong Thap Province – the centre with the largest lotus cultivation area in Vietnam – had reached over 1 200 hectares, with plans to expand to 1 400 hectares by 2025 (Huynh 2022).

Since the demand for health-oriented plant-derived products continues to increase, the efficiency of post-harvest processing has become a key determinant of the economic value of lotus cultivation. Post-harvest handling typically involves two key operations: dehulling (removal of the seed coat) and embryo removal (extraction of the bitter embryo). The initial step involves the removal of the hard outer shell encasing the lotus kernel, a phase that now typically employs mechanical machining techniques instead of traditional manual methods (Rong et al. 2022). However, upon completion of this stage, a mixture comprising the lotus seed kernels, lotus seed pods, kernel fragments, and a minor quantity of inadequately peeled lotus seeds is produced as shown in Figure 1 (Huynh et al. 2019). Given that shells and contaminants can obstruct the subsequent embryo removal stage, it is critical to isolate the peel from the core. Although pneumatic separation methods are commonly integrated into agricultural separation equipment (Ji et al. 2025), the recovered kernel stream often remains insufficiently clean, with a relatively high proportion of entrained impurities persisting despite the ongoing technical refinements (Ma et al. 2015).

To address the limitations of mechanical separation, recent advancements in computer vision have paved the way for the automation of agricultural product sorting (Patrício and Rieder 2018). Several traditional methods have been implemented, such as a strawberry sorting system based on the shape and colour (Xu and Zhao 2010); image segmentation techniques for classifying raisins (Abbasgholipour et al. 2011); and methods for detecting foreign matter in post-harvest soybeans (Momin et al. 2017). However, these studies rely primarily on conventional image processing techniques, which require complex pre-processing and manual feature extraction (Blasco et al. 2009), often leading to time-consuming computations and low stability under varying lighting conditions.

Consequently, deep learning (DL) has been increasingly applied in agricultural engineering due to its superior accuracy and speed in automatic feature extraction (Fan et al. 2020; Fu et al. 2020). Foundation models like convolutional neural networks (CNNs) have proven effective in classification tasks (Nasiri et al. 2019, 2020; Jahanbakhshi et al. 2021; Parvathi and Sankar 2021). For complex tasks requiring both identification and localisation (object detection), algorithms typically employ CNNs as a backbone. While two-stage algorithms like Faster R-CNN offer high accuracy (Wan and Goudos 2020), their processing speed is often limited (Fu et al. 2020). In contrast, one-stage algorithms such as YOLO offer significantly faster processing speeds (Lu et al. 2022) making them

Figure 1. Unsorted mixture of lotus seed components after the shelling stage

https://doi.org/10.17221/29/2026-RAE

well-suited for real-time requirements. For instance, YOLOv3 was successfully integrated into a robotic system for the precise identification and sorting of jujubes (Lu et al. 2021).

The main aim of this paper is to offer a novel strategy that combines YOLOv8 deep learning architecture with a real-time computer vision system, focusing on recognising distinct lotus seed components after mechanical dehulling. In order to sort these components with higher productivity and accuracy, effectively overcoming the high impurity rates and low efficiency of traditional mechanical methods, a comprehensive integrated engineering framework is suggested.

MATERIAL AND METHODS

Image acquisition system. To construct the dataset for the training and validation of the deep learning models, an experimental image acquisition system was established to simulate a real-world sorting scenario, as shown in Figure 2. This apparatus was positioned immediately at the output of the discharge outlet of the mechanical lotus seed sheller to ensure a continuous material flow.

The hardware architecture comprises the primary components: a receiving unit for the post-decortication kernel-shell mixture, a rotary feeder, a mechanical conveying unit, and a digital camera. Specifically, upon exiting the vibratory cutting and pressing mechanism, the mixed stream of lotus kernels and shells flows into the hopper of a feeding mechanism. This feeder subsequently distributes the materials into three randomised streams onto a continuous conveyor belt (Figure 2). The conveyor utilises a non-reflective rubber belt, moving at a constant velocity (1.5 m·min^{-1}). The background of the belt surface was selected to maximise the contrast, thereby facilitating efficient object detection and segmentation.

Images were acquired using a high-resolution camera equipped with a CMOS sensor (XW180, Rapoo, China), offering a resolution of 1 080 × 720 pixels and 30 frames per second (FPS). The camera was mounted in a fixed nadir orientation perpendicular to the conveyor belt surface. A constant working distance was maintained to ensure scale invariance across the captured frames. To mitigate the effects of ambient lighting variability and eliminate shadows, the system is placed in a uniform and fixed lighting environment. This setup provides uniform illumination across the field of view (FOV). The acquisition system was interfaced with a central processing unit via a high-speed USB connection, facilitating real-time video stream acquisition. Individual frames were subsequently extracted to identify primary target classes: unshelled seeds, kernels, and shells.

Dataset preparations. To ensure the generalisability and robustness of the detection model, the

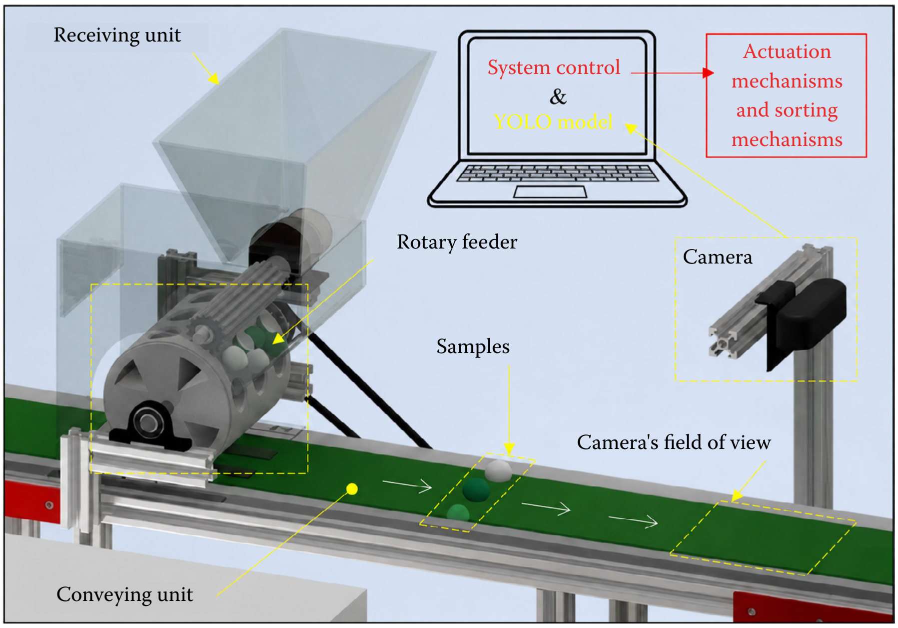

Figure 2. Hardware layout of the experimental apparatus

https://doi.org/10.17221/29/2026-RAE

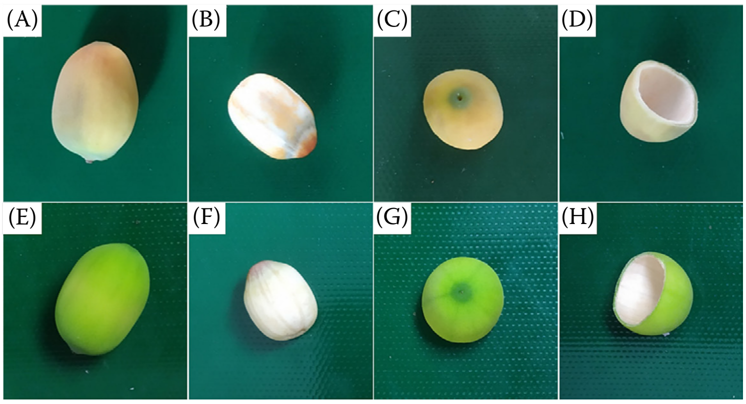

Figure 3. Representative images of the three object classes: (A, E) unshelled seeds, (B, F) kernels, (C, D, G, H) shells; panels A–D show withered samples, whereas panels E–H show fresh samples

Table 1. Statistical distribution of the original collected dataset

Category image	Description	Quantity	Percentage
Unshelled seed (Hat)	whole seeds before processing or unshelled	1 000	20
Kernel (Com)	white peeled kernels	1 500	30
Shell (Vo)	broken shell fragments/halves	1 500	30
Mixed	scenes containing multiple object types	1 000	20

data acquisition and processing procedures were rigorously conducted. Given that input image quality is a critical step for the deep learning algorithm performance, a specialised dataset comprising post-decortication components was constructed.

A total of 5 000 original images were acquired using the experimental setup described in the previous section. To accurately reflect the object diversity in actual production environments, images were captured under various scenarios, including variations in the object orientation and rotation. The target objects were categorised into primary classes: unshelled seeds, kernels and shells as illustrated in Figure 3. The detailed quantitative breakdown of the original collected dataset, including single-class captures and mixed-object scenarios, is presented in Table 1.

The image annotation was performed manually using open-source annotation tools to ensure the accuracy of the ground truth labels, while agricul-

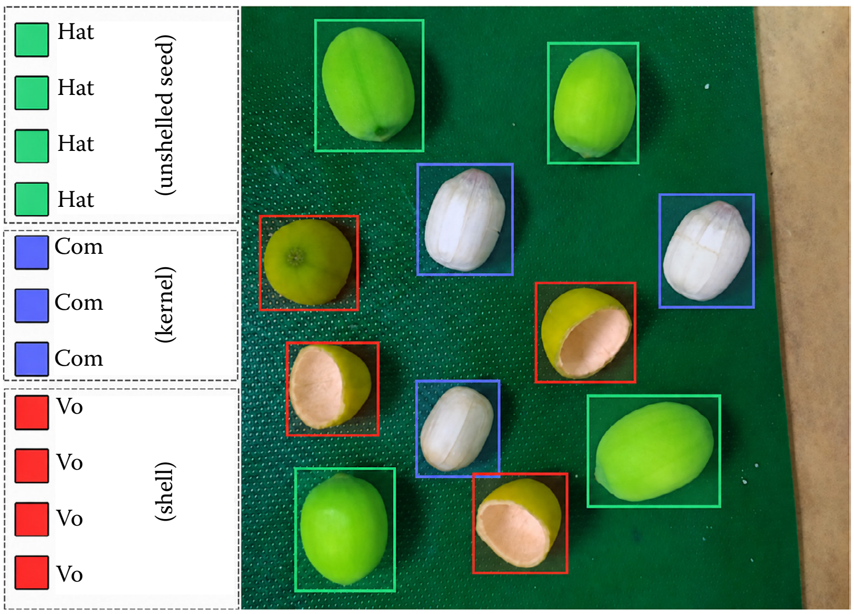

Figure 4. Data annotation with bounding boxes for the classes: unshelled seeds, kernels and shells

https://doi.org/10.17221/29/2026-RAE

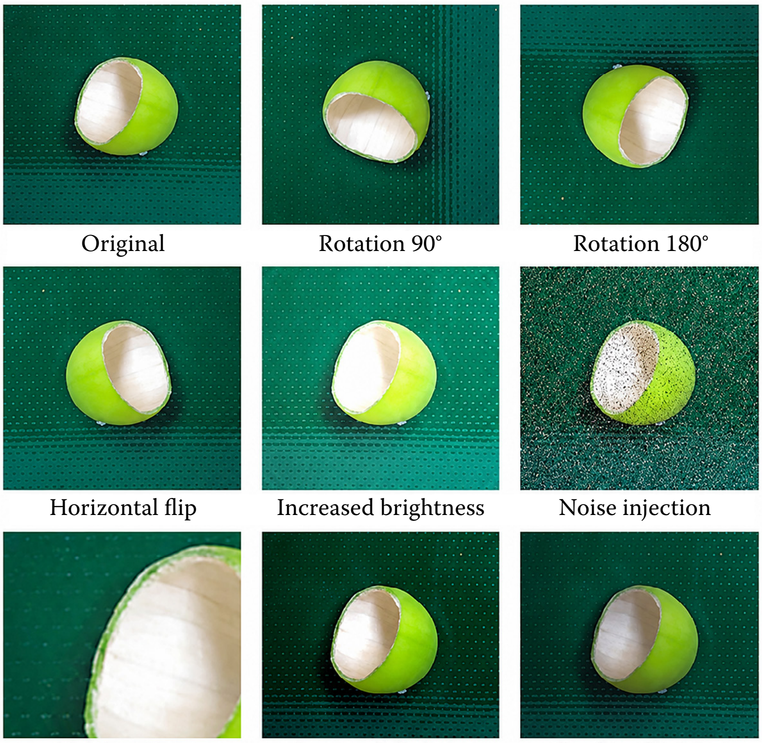

Figure 5. Geometric and photometric transformations utilised for the data augmentation

tural experts participated in a cross-verification process. Figure 4 below shows that the objects within the images were localised using a bounding box and assigned a label corresponding to one of the aforementioned target classes. The annotation format adhered to the standard YOLO structure with normalised coordinates.

Prior to training, the original dataset was randomly partitioned into three subsets: a training set (70%), a validation set (20%), and a test set (10%). To mitigate the risk of overfitting and enhance the model's robustness against morphological variations, data augmentation techniques were applied, including: geometric transformations (rotation, horizontal/vertical flipping), and random cropping were utilised to simulate different viewing angles of the seeds as they roll on the conveyor belt (Figure 2). Photometric transformations, adjustments to brightness and contrast, along with noise injection, were employed to simulate non-ideal working environmental conditions (Figure 5). Following augmentation, the dataset was expanded to ensure a relative balance among classes, establishing a rich, diverse, and highly representative dataset for each data class before using it in model training.

Establishing a rich, diverse, and highly representative dataset for each data class before using it in model training, the quantitative distribution of the final augmented dataset across the three subsets is detailed in Table 2. This expansion results in a well-balanced distribution among the unshelled seed, kernel, and shell classes, providing a sufficient volume of data to minimise the bias and ensure generalisation capabilities.

YOLOv8 architecture selection. To meet the rigorous demands of real-time classification on a high-speed conveyor system, the YOLOv8 architecture was selected for its superior balance between its computational inference speed and detection precision. As illustrated in Figure 6, the network comprises three functional compo-

Table 2. Distribution of images in the augmented dataset used for training, validation and testing

Training set (70%)	Validation set (20%)	Test set (10%)	Total images
31 500	9 000	4 500	45 000

https://doi.org/10.17221/29/2026-RAE

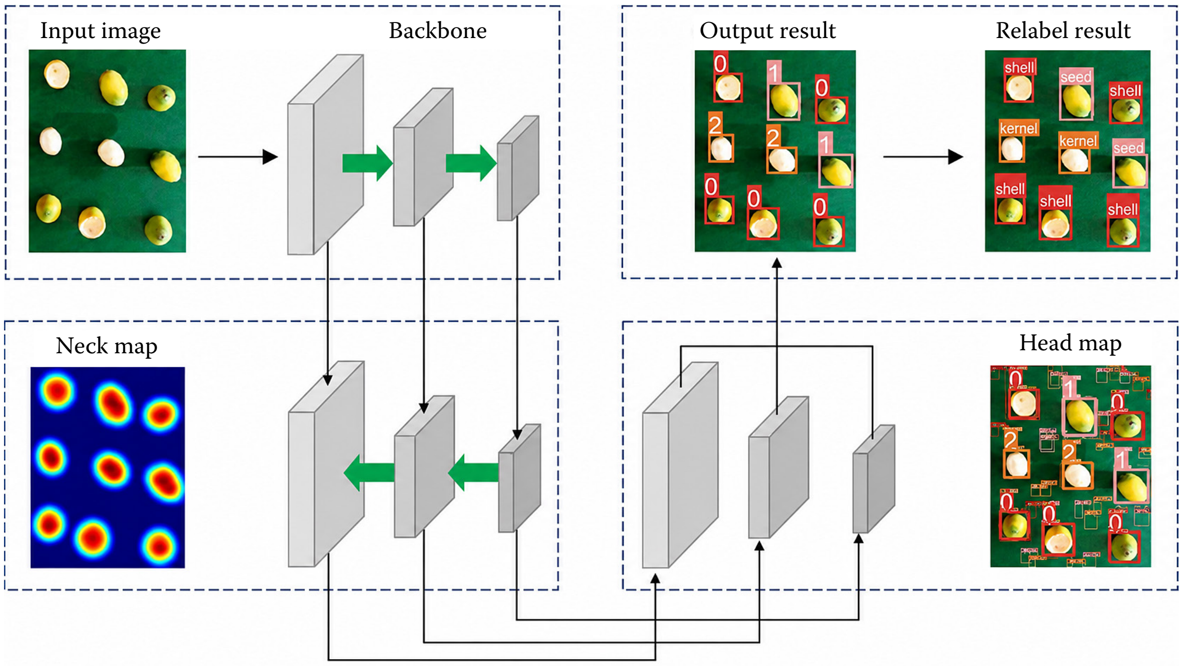

Figure 6. The overall architecture of the proposed YOLOv8 model

nents optimised for the specific morphological characteristics of lotus seeds: the backbone, the neck, and the head.

The backbone is responsible for extracting visual features from the input stream. YOLOv8 improves upon the CSPDarknet53 architecture by introducing the C2f (cross-stage partial bottleneck with two convolutions) module. In the context of lotus seed sorting, this module enhances the gradient flow and feature representation, allowing the model to effectively distinguish distinct visual features such as the bright, uniform surface of peeled kernels versus the distinctive pigmentation of the shells, even under varying lighting conditions or potential motion blur caused by the conveyor movement.

The dataset presents a challenge regarding the object orientation and scale. While whole seeds are uniform, the shell halves often appear in various rotations (cup-up or cup-down). To address this, the neck component utilises a feature pyramid network (FPN) integrated with a path aggregation network (PAN). This structure facilitates multi-scale feature fusion, ensuring that semantic information is preserved across different resolutions. Consequently, the system maintains high detection sensitivity for both the whole seeds and the smaller, hollow shell halves.

Departing from anchor-based predecessors, YOLOv8 employs a decoupled head structure with an anchor-free mechanism. This is particularly advantageous for this application. Although the shell fragments typically split into hemispherical halves, their orientation on the conveyor belt is unpredictable (e.g., the hollow inside may be visible or hidden). The anchor-free approach, which directly predicts object centres, allows the model to robustly generalise across these varying viewing angles without being constrained by rigid pre-defined anchor boxes.

To optimise the model for the three target classes (unshelled, kernel, shell), the training process minimises a composite loss function:

$$\text{Loss} = \lambda_{\text{box}} L_{\text{CIoU}} + \lambda_{\text{cls}} L_{\text{BCE}} + \lambda_{\text{dfl}} L_{\text{DFL}} \tag{1}$$

where: λ_{box}, λ_{cls}, λ_{dfl} – the weights (hyperparameters) that help balance the importance of each loss component during the training process; this prevents the model from over-focusing on a single factor (for example, only prioritising correct classification while the bounding box remains misaligned); L_{CIoU} – (complete IoU loss) ensures the predicted bounding boxes tightly fit the object boundaries (Equation 2); L_{BCE} – (binary cross-entropy) used for accurately classifying the distinct categories (Equation 3); and L_{DFL} – (distribution focal loss) specifically helps in refining the box boundaries, which is crucial when separating the convex shape of a seed from the similar concave shape of a shell half (Equation 4).

https://doi.org/10.17221/29/2026-RAE

These components are mathematically defined as follows:

$$L_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^{2}\left(b, b^{gt}\right)}{c^{2}} + \alpha v \quad (2)$$

where: IoU – the inter-section over the union between the predicted box and ground truth; $\rho^2(b,b^{gt})$ – the Euclidean distance between the central points of the predicted box (b) and the ground truth box (b^{gt}); c – the diagonal length of the smallest enclosing box that covers both boxes; αv – the aspect ratio penalty term, where c measures the consistency of the aspect ratio and v is a trade-off parameter.

$$L_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\widehat{y_i}\right) + \left(1 - y_i\right)\log\left(1 - \widehat{y_i}\right)\right] \quad (3)$$

where: $y_i \in [0,1]$ is the ground truth label for class i; $\hat{y}_1$ – the predicted probability for class I; N – the total number of samples.

$$L_{\text{DFL}}\left(S_i, S_{i+1}\right) = -\left(\left(y_{i+1} - y\right)\log\left(S_i\right) + \left(y - y_i\right)\log\left(S_{i+1}\right)\right) \quad (4)$$

where: y – the continuous ground truth label for the coordinate, y_i, y_{i+1} – the nearest integer values to y ($y_i \leq y \leq y_{i+1}$); S_i, S_{i+1} – the probabilities estimated by the network at locations y_i and y_{i+1}.

Set up the training environment. The training hyperparameters of the YOLOv8-nano model (YOLOv8n) were selected based on the characteristics of the dataset and the requirement for real-time inference. An input resolution of 640 × 480 pixels was adopted to balance the detection accuracy and computational efficiency. Pretrained weights were used to improve the convergence and robustness, while the default optimisation parameters of YOLOv8 were retained due to their proven effectiveness.

Early stopping was employed to mitigate overfitting. This configuration ensures sufficient detection accuracy while maintaining low inference latency for industrial applications. The training hyperparameters were configured based on empirical recommendations for the model as detailed in Table 3.

The experiments were executed on a high-performance personal computer equipped with an Intel Core i5 CPU (11th Gen) and 8GB of RAM. The model training tasks, which require intensive matrix computations, were accelerated by a dedicated NVIDIA GeForce RTX 3050 GPU with 4GB of Video RAM (VRAM). The algorithm was developed on the Windows 11 (64-bit) operating system.

The PyTorch framework served as the underlying deep learning backend, integrated with the CUDA 11.8 parallel computing library and CuDNN (CUDA deep neural network library) to maximise the GPU performance. The YOLOv8 model architecture was implemented using the latest ultralytics library, with Python 3.10 as the primary programming language.

RESULTS AND DISCUSSION

Model training performances. The model training process was executed over 50 epochs. Figure 7 presents the training and validation dynamics

Table 3. Experimental configuration and hyperparameter specifications

Hyperparameter	Setting	Description
Image resolution	640 × 480	normalised input size for feature extraction
Batch size	16	adjusted based on GPU memory constraints
Total epochs	50	total number of complete passes via datasets
Warmup epochs	3.0	initial epochs for stabilising gradients
Optimiser	SGD	stochastic gradient descent
Momentum	0.937	accelerates gradient vectors in the right direction
Weight decay	0.0005	L2 regularisation to prevent overfitting
Initial learning rate	0.01	starting step size for weight updates
Scheduler	Cosine	gradually reduces LR to fine-tune convergence
Final LR factor	0.01	final learning rate fraction of the initial LR
Augmentation	mosaic	enhances the detection of small objects

GPU – graphics processing unit; SGD – stochastic gradient descent; LR – learning rate

https://doi.org/10.17221/29/2026-RAE

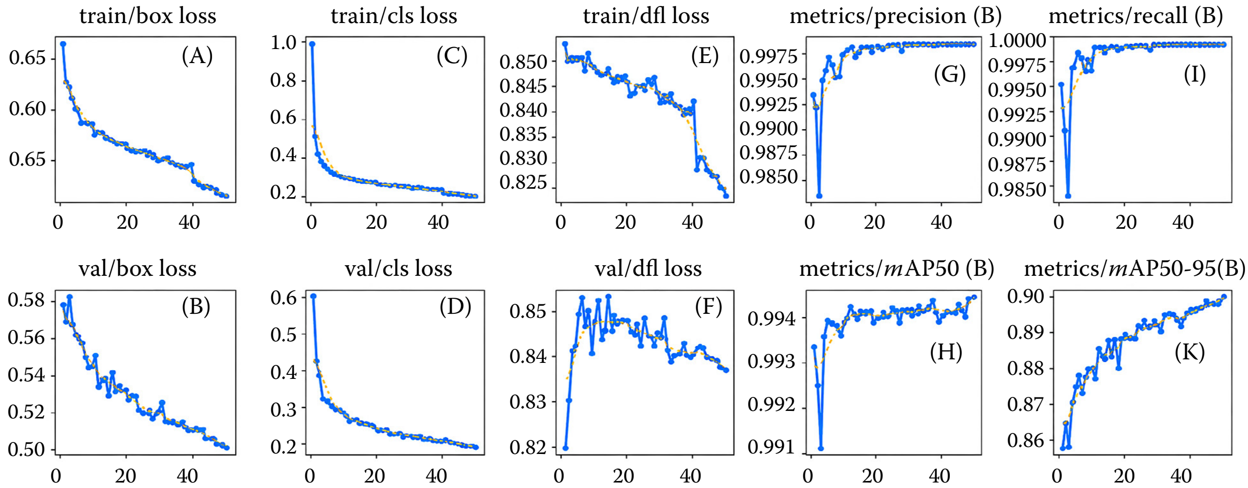

Figure 7. Evolution of the loss functions and performance metrics during the training of the model

cls – classification loss; dfl – distribution focal loss; train – training; val – validation; B – bounding boxes; *m*AP – mean average precision

of the YOLOv8 detector, including the box loss, classification loss, and distribution focal loss, together with the learning curves of precision, recall, mean average precision (*m*AP) *m*AP@0.5, and *m*AP@0.5-0.95. Overall, the curves indicate a stable and well-behaved optimisation process.

As in Figure 7A and C, during the early stage (approximately the first 20 epochs), both the train/box loss and train/cls-loss decrease sharply, reflecting fast acquisition of core visual features and effective gradient-driven learning. After this period, the loss reduction becomes more gradual, and the curves approach a steady regime toward the later epochs, suggesting that the model has largely converged under the current training configuration. Similarly, the validation losses (Figure 7B and D) follow a consistent downward trend and remain close to their corresponding training losses, implying no widening train-validation gap and therefore no evident overfitting. While the val/dfl-loss (Figure 7E and F) exhibits higher variability than the other losses (including a transient rise in the early-to-mid epochs), it stabilises afterwards without sustained divergence, which is consistent with normal localisation refinement behaviour during detector training.

Regarding the training metrics, the precision and recall (Figure 7G and I) improve rapidly and reach a high, saturated region early in training, then remain stable with only minor fluctuations, indicating that the detector quickly learns discriminative decision boundaries and maintains consistent detection quality across the epochs. Meanwhile, *m*AP@0.5 (Figure 7H) increases steeply and plateaus relatively early, whereas *m*AP@0.5-0.95 (Figure 7K) continues to improve more gradually throughout the later epochs. This pattern is typical for object detectors: coarse detection capability saturates first, while tighter localisation performance (captured by the stricter IoU thresholds in *m*AP@0.5-0.95) keeps benefiting from continued fine-tuning. Overall, the combined loss and metric trajectories confirm stable convergence and good generalisation behaviour during training, with the most beneficial learning occurring in the early-to-mid training phase and progressively diminishing gains in the ten final epochs.

Real class-wise performance analysis. The training accuracy and performance of the model are quite high; however, these results are a reflection of the overall average. It is necessary to further scrutinise the model classification accuracy and identify any potential misclassifications among a morphologically similar class – the confusion matrix was computed on the test set (Figure 8). This visualisation allows for the calculation of key performance metrics, including the precision (P), recall (R), and average precision (AP), defined as follows:

$$\text{Precision}(\%) = \frac{TP}{TP+FP} \times 100 \tag{5}$$

$$\text{Recall}(\%) = \frac{TP}{TP+FN} \times 100 \tag{6}$$

$$AP = \int_0^1 P(R)dR \tag{7}$$

https://doi.org/10.17221/29/2026-RAE

$$mAP = \frac{1}{M}\sum_{k=1}^{M} AP(k) \quad (8)$$

where: *TP* (true positives) – the number of positive instances that the model correctly predicted as positive; *FP* (false positives) – the number of negative instances (or background/irrelevant objects) that the model incorrectly predicted as positive; *FN* (false negatives) – the number of actual positive instances that the model failed to detect; *R* – the precision-recall function, representing the precision-recall curve used to calculate the area for a specific class; *M* – the total number of classes (the shell, kernel and unshelled seeds) in this study; *AP*(*k*) – the average precision value for the k^{th} classes; *mAP* – the mean of the *AP*s across all the classes.

As observed in the normalised confusion matrix, the high concentration of values along the main diagonal indicates that the majority of instances for the unshelled seeds, kernels and shells were correctly classified. The off-diagonal elements, representing prediction errors, remain low. These preliminary results suggest that the YOLOv8n model is capable of performing quite well under experimental conditions. While the accuracy of the unshelled seed and kernel classifications are very high (respectively 99.2% and 98.4%), a significant confusion in the shell classification was observed. Specifically, 6.4% of the actual shell instances were misclassified as unshelled seeds, and a considerable 13.4% were incorrectly identified as kernels. This suggests that the model currently struggles to robustly differentiate certain morphological features of shells with an accuracy of 80.2% – significantly lower than the remaining classes. This limitation is likely due to the complex morphological characteristics of the shell after mechanical processing. Unlike the relatively uniform elliptical shape of the nucleus and lotus seed, the fragmented shells have irregular geometries and unpredictable orientation.

Furthermore, the YOLOv8-nano's feature-frame-based detection mechanism poses a challenge for such fragmented objects. For small shell fragments, a rectangular frame will encompass a large proportion of background pixels relative to the actual target area. This inclusion of too much irrelevant information weakens the ability to represent features within the frame, leading to lower IoU scores and sometimes causing classification confusion. This observation has also suggested a future improvement direction - semantic segmentation methods that precisely separate pixel-level boundaries of shell fragments, overcoming the inherent limitation of the bounding box for small, irregularly shaped shell fragments.

Comparative analysis of YOLOv8 variants. Comparative experiments were conducted on four common variants of the YOLOv8 architecture YOLOv8n (nano), YOLOv8 s (small), YOLOv8 m (medium), and YOLOv8 l (large) to survey the optimal configuration that balances the detection accuracy and computational efficiency for the real-

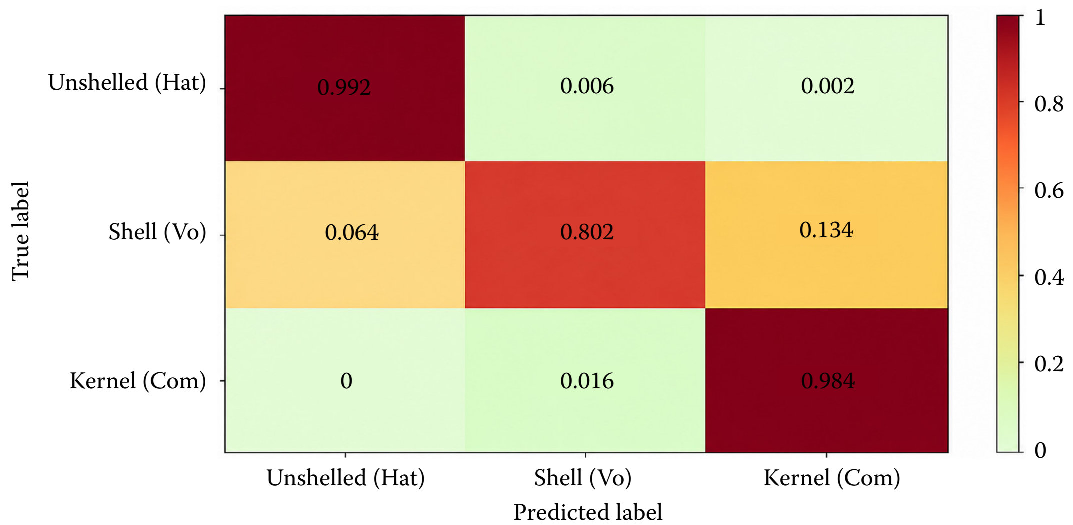

Figure 8. Normalised confusion matrix evaluating the classification performance of the YOLOv8 model on the test set across shell (Vo), seed (Hat), and kernel (Com) classes

https://doi.org/10.17221/29/2026-RAE

Table 4. Experimental performance of YOLOv8 models on the same test dataset

Version	Nano	Small	Medium	Large
FPS	9.70	4.64	2.54	1.40
*m*AP@0.5 (%)	99.4	99.5	99.4	99.5
*m*AP@0.5-95 (%)	91.4	91.5	91.4	91.1

FPS – frames per second; *m*AP – mean average precision

time sorting system. These models were trained and tested on the identical lotus seed dataset under uniform hyperparameter settings. Table 4 summarises the key performance metrics, including FPS and *m*AP.

The experimental results reveal a distinct trend regarding detection accuracy. All the variants exhibited exceptional performance, with the *m*AP@0.5 values plateauing at a high range of 99.4% to 99.5%. Increasing the network depth and width from the Nano version (YOLOv8n) to the large version (YOLOv8 l) slightly improves in accuracy. Specifically, the *m*AP@0.5-95 of the largest variant (YOLOv8 l) was 91.1%, which is marginally lower than that of the smallest variant (YOLOv8n) at 91.4%. This saturation in accuracy suggests that the morphological features of the target objects (lotus seeds and shells) are sufficiently distinct and distinct from the background, enabling a lightweight model like YOLOv8n to effectively extract features and classify objects without necessitating the extensive computational capacity of deeper networks.

In the inference speed, the measurement results demonstrate a significant difference with the model size. The YOLOv8n variant achieved the highest processing speed of 9.7 FPS, which is approximately seven times faster than the YOLOv8 l variant (1.40 FPS) under the same experimental hardware conditions. The significant reduction in FPS observed in the larger models (medium and large models) renders them impractical for real-time applications on resource-constrained devices.

For the post-decortication lotus seed sorting system, the prerequisite is the capability for an instantaneous response to accurately trigger the pneumatic mechanism as the conveyor belt moves at high velocity. Although YOLOv8 l offers marginally better accuracy, the reduction in FPS could induce latency in the control signal. Therefore, YOLOv8n was identified as the optimal configuration for the automated lotus seed sorting system. In this context, it not only maintains a good accuracy (*m*AP@0.5 of 99.4%), comparable to larger models, but also delivers high processing speed,

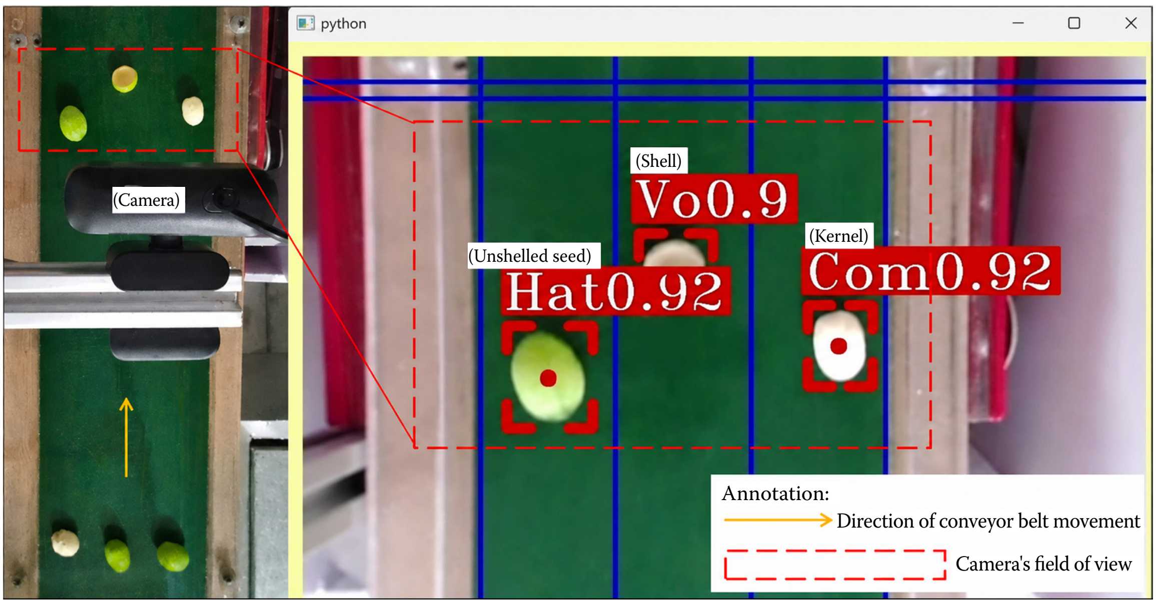

Figure 9. Experimental setup and real-time object detection results on the conveyor system

https://doi.org/10.17221/29/2026-RAE

thereby best satisfying the real-time response requirements of the actuation mechanism.

Calculating real-time system deployment. The YOLOv8n variant was integrated into a hardware prototype (Figure 9) designed to simulate the lotus seed sorting process. The primary objective of this experimental setup was to validate the operational considerations of the deep learning model in a simulated production environment. Experiments were conducted with the conveyor belt velocity set at 1.5 m·min$^{-1,}$ creating a simulated workflow that mirrors potential production scenarios.

The total processing latency, defined as the time interval from image acquisition to the generation of the actuation signal, was considered to assess the synchronisation capabilities of the prototype. Denoting the end-to-end latency as the total processing latency, defined as the time interval from image acquisition to the generation of the actuation signal, was monitored to assess the synchronisation capabilities of the prototype. Denoting the end-to-end latency as t_{lat}, it can be expressed as the sum of key processing components:

$$t_{\mathrm{lat}} = t_{\mathrm{acq}} + t_{\mathrm{pre}} + t_{\mathrm{inf}} + t_{\mathrm{nms}} + t_{\mathrm{comm}} + t_{\mathrm{ctrl}} \tag{9}$$

where: t_{acq} – image acquisition time; t_{pre} – pre-processing time; t_{inf} – model inference time; t_{nms} – NMS post-processing time; t_{comm} – communication delay time; t_{ctrl} – control computation time.

With the YOLOv8n model running at approximately 10 FPS, the system achieved an average response time of 6 ms per frame. The processing rate is related to the per frame time t_{frame} as:

$$\mathrm{FPS} \approx t_{\mathrm{frame}}{}^{-1} \tag{10}$$

In practice, there are differences in the installation and integration depending on the conveyor length and computer processing speed. Therefore, it is necessary to integrate position-based calibration computing, allowing the control unit to correctly compensate for processing delays. This ensures that the pneumatic nozzles were triggered precisely when the identified shell residues reached the ejection zone. Specifically, let L denote the belt-direction distance from the vision observation region to the sorting zone and v the conveyor-belt speed. The travel time (t_{travel}) of an object is:

$$t_{\mathrm{travel}} = L \times v^{-1} \tag{11}$$

In addition, the pneumatic actuation delay (t_{act}) can be represented as:

$$t_{\mathrm{act}} = t_{\mathrm{valve}} + t_{\mathrm{air}} + t_{\mathrm{jet}} \tag{12}$$

where: t_{valve} – the electromechanical delay of the solenoid valve; t_{air} – the pneumatic propagation delay; and t_{jet} – the jet formation delay.

Accordingly, a sufficient synchronisation condition for real-time triggering is:

$$t_{\mathrm{lat}} + t_{\mathrm{act}} \leq t_{\mathrm{travel}} \tag{13}$$

Using position-based compensation, if a residue is located at a belt coordinate at a detection time and the ejection zone is at x_e, the predicted arrival time (t_{arr}) will be:

$$t_{\mathrm{arr}} = (x_e - x_0) \times v^{-1} \tag{14}$$

and the trigger time (t_{trig}) can be scheduled as:

$$t_{\mathrm{trig}} = t_{\mathrm{arr}} - t_{\mathrm{act}} \tag{15}$$

The experimental sorting efficiency in terms of the sorting performance for fresh lotus seeds with a density of 2.5 ~ 3 g·seed^{-1}, the experimental apparatus demonstrated a processing throughput of approximately 60 kg·h^{-1}. The throughput Q can be defined as:

$$Q = \frac{N \times \overline{w}_{\mathrm{gram}}}{T} \times 10^{-3} \tag{16}$$

where: Q – the processing throughput of the system (kg·h^{-1}); N – the total number of lotus seeds counted and processed by the system within the time period; $\overline{w}_{\mathrm{gram}}$ – the average mass of a single fresh lotus seed (g·seed^{-1}); T – the duration of the processing operation (h).

Furthermore, the purity rate after sorting can be lower than the detection accuracy of the YOLO model. Therefore, it is necessary to carefully consider and integrate compensation calculations based on the instantaneous speed of the whole system, which may fluctuate during the actual operation and affect the real sorting performance of the final actuators.

https://doi.org/10.17221/29/2026-RAE

Table 5. Performance comparison between the proposed method and existing systems

Study	Objectives	Model and methods	Performances
Blasco et al. (2009)	sort pomegranate arils by colour and remove unwanted materials	computer vision using R/G ratio thresholding combined with air ejectors	accuracy: 90%; image processing: 15 ms per frame; throughput: 75 kg·h^{-1}
Abbasgholipour et al. (2011)	automatic sorting system to assess raisin quality and remove defective fruits	genetic algorithm in HIS-colour space with pneumatic valves	sensitivity: 87.9% ~ 95.9%; conveyor speed: 250 mm·s^{-1}
Momin et al. (2017)	detect and classify split, contaminated, defect beans, and stem/pods in harvested soybeans	openCV-based image processing using dual lighting, HIS-colour space and watershed transformation	accuracy: normal seed: 100%, split: 96%, contaminated: 75%, defect and stem/pods: 98%
Wan and Goudos (2020)	multi-class fruit detection for robotic harvesting	improved Faster R-CNN model and an image acquisition system	*m*AP: 90.72% (apple: 92.51%, orange: 90.73%, mango: 88.94%); processing speed: 58 ms per frame
Lu et al. (2021)	grading jujubes based on maturity for automated post-harvest systems	YOLOv3, the system consists of a conveyor belt, camera and actuators	accuracy: 97%; image processing: 0.17 s per image; sorting time: 1.39 s per fruit
Our method in this study	automate post-decortication sorting to classify and separate lotus seed components	YOLOv8 (nano) model, the system consists of a camera and mechanical conveyor	*m*AP: 92.6% (unshelled: 99.2%, kernel: 98.4%, shell: 80.2%); processing speed: 6 ms per frame; throughput: 60 kg·h^{-1}

To provide a comprehensive context for the performance of the proposed system, a comparative analysis with existing research in agricultural sorting and detection is presented in Table 5. Various methodologies, spanning from traditional image processing to modern deep learning architectures, are summarised to reflect the diversity of objectives in the field. Notably, while numerous agricultural products have been extensively studied, the specific challenge of sorting lotus seed kernels and shells after dehulling has not been addressed in the previous literature. Therefore, the advancements of the proposed YOLOv8-based approach are highlighted, particularly regarding its superior classification accuracy and real-time processing throughput for this novel case study.

CONCLUSION

This study successfully developed and validated a real-time computer vision system using the YOLOv8 nano architecture to automate the sorting of fresh lotus seed components (kernels, shells, and unshelled seeds) after mechanical shelling. The experimental results demonstrate that the system operates with high accuracy and stability, achieving an average accuracy (*m*AP@0.5) of 92.6% and maintaining a processing capacity of approximately 60 kg·h^{-1}.

The core novelty of this research lies in the first implementation of deep learning to sort a complex mixture of fresh lotus seeds – an unresolved challenge in automatic post-harvest processing. Furthermore, the proposed system offers highly practical retrofitting capabilities, allowing easy integration into existing small and medium-sized shelling machines without major mechanical changes, thereby optimising the investment costs for small and medium-sized enterprises.

However, the actual sorting purity may be affected by the instantaneous conveyor speed fluctuations and electromechanical hysteresis, and the detection of extremely small, obscured shell fragments can be a challenge. Future research can overcome these issues by diversifying the dataset and integrating semantic segmentation to improve the shell fragment detection capabilities. The integration of lightweight segmentation architectures (such as YOLOv8-seg or Nano-Seg) for pixel-level label of shell fragments should be considered. Besides that, optimising the hardware by deploying high-frequency solenoid valves and rotary encoders to minimise the elec-

https://doi.org/10.17221/29/2026-RAE

tromechanical latency and precisely synchronise the pneumatic mechanism with real-time conveyor speed should also be considered.

REFERENCES

Abbasgholipour M., Omid M., Keyhani A., Mohtasebi S.S. (2011): Color image segmentation with genetic algorithm in a raisin sorting system based on machine vision in variable conditions. Expert Systems with Applications, 38: 3671–3678.

Blasco J., Cubero S., Gómez-Sanchís J., Mira P., Moltó E. (2009): Development of a machine for the automatic sorting of pomegranate (*Punica granatum*) arils based on computer vision. Journal of Food Engineering, 90: 27–34.

Fan S., Li J., Zhang Y., Tian X., Wang Q., He X., Zhang C., Huang W. (2020): On line detection of defective apples using computer vision system combined with deep learning methods. Journal of Food Engineering, 286: 110102.

Fu L., Majeed Y., Zhang X., Karkee M., Zhang Q. (2020): Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RBG and depth features for robotic harvesting. Biosystems Engineering, 197: 245–256.

Huynh B. (2022): Dong Thap enhances the value of lotus products. Vietnam National Authority of Tourism. Available at https://vietnamtourism.gov.vn/post/43861

Huynh T.-T., Nguyen H.-T., Tran N.P.L., Huynh Q.-K., Le P.-H. (2019): Design and fabrication of cutting part for fresh lotus seed peeling machine. Journal of Science and Technology Issue on Information and Communications Technology, 17: 33–39.

Jaha A.F., Kasireddy C., Roy M. (2026): Lotus seeds (*Nelumbo nucifera*). In: Underutilized Seeds: Sustainable Solutions for Health, Industry, and Environment. Singapore, Springer Nature Singapore: 141–179.

Jahanbakhshi A., Momeny M., Mahmoudi M., Radeva P. (2021): Waste management using an automatic sorting system for carrot fruit based on image processing technique and improved deep neural networks. Energy Reports, 7: 5248–5256.

Ji S., Li A., Han Y., Li H., Sun Z., Zhao D., Gao H. (2025): Numerical study on the influence mechanism of feed rate on separation quality in a multi-baffle separation duct. Biosystems Engineering, 258: 104258.

Liao C.H., Lin J.Y. (2011): Lotus (*Nelumbo nucifera* Gaertn.) plumule polysaccharide protects the spleen and liver from spontaneous inflammation in non-obese diabetic mice by modulating pro-/anti-inflammatory cytokine gene expression. Food Chemistry, 129: 245–252.

Lu S., Chen W., Zhang X., Karkee M. (2022): Canopy-attention-YOLOv4-based immature/mature apple fruit detection on dense-foliage tree architectures for early crop load estimation. Computers and Electronics in Agriculture, 193: 106696.

Lu Z., Zhao M., Luo J., Wang G., Wang D. (2021): Design of a winter-jujube grading robot based on machine vision. Computers and Electronics in Agriculture, 186: 106170.

Ma Q., Lu A., Gao L., Wang Z., Tan Z., Li X. (2015): Aerodynamic characteristics of lotus seed mixtures and test on pneumatic separating device for lotus seed kernel and contaminants. Transactions of the Chinese Society of Agricultural Engineering (Nongye Gongcheng Xuebao), 31: 297–303.

Momin M.A., Yamamoto K., Miyamoto M., Kondo N., Grift T. (2017): Machine vision based soybean quality evaluation. Computers and Electronics in Agriculture, 140: 452–460.

Nasiri A., Taheri-Garavand A., Zhang Y.-D. (2019): Image-based deep learning automated sorting of date fruit. Postharvest Biology and Technology, 153: 133–141.

Nasiri A., Omid M., Taheri-Garavand A. (2020): An automatic sorting system for unwashed eggs using deep learning. Journal of Food Engineering, 283: 110036.

Nguyen D.K., Dinh T.K.O., Truong T.Q., Nguyen T.P., Nguyen C.D., Pham X.H., Dung T.Q. (2024): Farmer-trader vertical coordination: Drivers and impact on the lotus-grain value chains in central Vietnam. Cogent Economics and Finance, 12: 2357154.

Parvathi S., Sankar T.S. (2021): Detection of maturity stages of coconuts in complex background using faster R-CNN model. Biosystems Engineering, 202: 119–132.

Patrício D.I., Rieder R. (2018): Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. Computers and Electronics in Agriculture, 153: 69–81.

Paudel K.R., Panth N. (2015): Phytochemical profile and biological activity of *Nelumbo nucifera*. Evidence-Based Complementary and Alternative Medicine, 2015: 789124.

Rong C., Hou Q.-X., Chen Y.-X., Wang Y., Zhao D., Zhang G. (2022): Design and experiment of multi-channel integrated shelling mechanism for fresh lotus seed. Transactions of the Chinese Society for Agricultural Machinery, 53: 79–88.

Sneh P.B., Dunno K., Kumar M., Mostafa H., Maqsood S. (2022): A comprehensive review on lotus seeds (*Nelumbo nucifera* Gaertn.): Nutritional composition, health-related bioactive properties, and industrial applications. Journal of Functional Foods, 89: 104937.

Tu Y., Yan S., Li J. (2020): Impact of harvesting time on the chemical composition and quality of fresh lotus seeds. Horticulture, Environment, and Biotechnology, 61: 735–744.

https://doi.org/10.17221/29/2026-RAE

Wan S., Goudos S. (2020): Faster R-CNN for multi-class fruit detection using a robotic vision system. Computer Networks, 168: 107036.

Xu L., Zhao Y. (2010): Automated strawberry grading system based on image processing. Computers and Electronics in Agriculture, 71: S32–S39.

Zhang Y., Lu X., Zeng S., Huang X., Guo Z., Zheng Y., Tian Y., Zheng B. (2015): Nutritional composition, physiological functions and processing of lotus (*Nelumbo nucifera* Gaertn.) seeds: A review. Phytochemistry Reviews, 14: 321–334.

Zhang Y., Xu Y., Wang Q., Zhang J., Dai X., Miao S., Lu X. (2023): The antioxidant capacity and nutrient composition characteristics of lotus (*Nelumbo nucifera* Gaertn.) seed juice and their relationship with color at different storage temperatures. Food Chemistry: X, 18: 100669.

Received: February 8, 2026
Accepted: May 19, 2026
Published online: July 23, 2026