



ARTICLE

Real-Time Larval Stage Classification of Black Soldier Fly Using an Enhanced YOLO11-DSCConv Model

An-Chao Tsai* and Chayanon Pookunngern

International Master Program of Information Technology and Application, National Pingtung University, Pingtung, 900391, Taiwan

*Corresponding Author: An-Chao Tsai. Email: actsai@nptu.edu.tw

Received: 02 May 2025; Accepted: 17 June 2025; Published: 03 July 2025

ABSTRACT: Food waste presents a major global environmental challenge, contributing to resource depletion, greenhouse gas emissions, and climate change. Black Soldier Fly Larvae (BSFL) offer an eco-friendly solution due to their exceptional ability to decompose organic matter. However, accurately identifying larval instars is critical for optimizing feeding efficiency and downstream applications, as different stages exhibit only subtle visual differences. This study proposes a real-time mobile application for automatic classification of BSFL larval stages. The system distinguishes between early instars (Stages 1–4), suitable for food waste processing and animal feed, and late instars (Stages 5–6), optimal for pupation and industrial use. A baseline YOLO11 model was employed, achieving a mAP50-95 of 0.811. To further improve performance and efficiency, we introduce YOLO11-DSCConv, a novel adaptation incorporating Depthwise Separable Convolutions specifically optimized for the unique challenges of BSFL classification. Unlike existing YOLO+DSCConv implementations, our approach is tailored for the subtle visual differences between larval stages and integrated into a complete end-to-end system. The enhanced model achieved a mAP50-95 of 0.813 while reducing computational complexity by 15.5%. The proposed system demonstrates high accuracy and lightweight performance, making it suitable for deployment on resource-constrained agricultural devices, while directly supporting circular economy initiatives through precise larval stage identification. By integrating BSFL classification with real-time AI, this work contributes to sustainable food waste management and advances intelligent applications in precision agriculture and circular economy initiatives. Additional supplementary materials and the implementation code are available at the following link: [YOLO11-DSCConv](#), [Server Side](#), [Mobile Application](#).

KEYWORDS: Deep learning; convolutional neural networks (CNNs); YOLO11-DSCConv; black soldier fly larvae (BSFL); real-time object detection

1 Introduction

Food waste is a pressing global issue, contributing significantly to greenhouse gas emissions and climate change. When decomposed in landfills, food waste produces methane, a greenhouse gas approximately 28 times more potent than carbon dioxide in trapping heat [1]. Additionally, the entire life cycle of food—from production to disposal—places a substantial burden on environmental resources. Tackling this problem requires innovative, scalable, and sustainable approaches that not only mitigate environmental harm but also recover value from waste streams.

Recent studies have demonstrated that real-time deep learning models, such as those used for bacterial detection in water quality monitoring, can significantly enhance decision-making efficiency in biological systems [2]. These advances highlight the potential of AI-powered methods in optimizing agricultural waste



processing, particularly in applications such as BSFL-based recycling. One promising biological solution involves the use of Black Soldier Fly Larvae (BSFL, *Hermetia illucens*), which exhibit remarkable efficiency in decomposing organic matter. BSFL can transform food scraps into high-protein biomass suitable for animal feed and other value-added applications, contributing to circular economy practices. Prior studies have shown that BSFL can replace conventional protein sources such as fish meal in aquaculture, offering both nutritional benefits and sustainability advantages [3]. Additionally, advancements in rearing techniques have improved waste conversion rates and the nutritional quality of larvae [4], while optimized feeding conditions have enhanced their bioconversion performance [5]. BSFL products are now increasingly considered suitable for human consumption and pet food, further underscoring their versatility and global relevance [6].

To fully leverage the benefits of BSFL in large-scale applications, accurate classification of larval growth stages is essential. Early-stage larvae (Stages 1–4) are ideal for animal feed production, while later stages (Stages 5–6) are more appropriate for pupation and cosmetic or industrial uses. However, distinguishing these stages based on subtle morphological differences remains a technical challenge. To address this, we propose a novel machine learning-based system capable of automatically classifying BSFL growth stages. Leveraging YOLO11 [7], a high-performance object detection framework, the system achieved a preliminary mAP50-95 of 0.811. We further enhanced the model by introducing YOLO11-DSCnv, which integrates Depthwise Separable Convolutions (DSCnv) to improve computational efficiency, reaching an mAP50-95 of 0.813. While DSCnv has been applied to YOLO variants in general object detection contexts, our work makes several distinct contributions: 1) We present the first application of DSCnv-enhanced YOLO specifically optimized for the unique challenges of BSFL classification, where visual differences between stages are subtle yet biologically significant; 2) We provide a complete end-to-end system architecture that bridges the gap between model development and practical agricultural deployment; 3) We introduce a specialized dataset of 21,600 annotated BSFL images across six growth stages under varied conditions; and 4) We demonstrate direct integration with circular economy practices through precise identification of optimal larvae for different applications. This architecture significantly reduces model complexity while maintaining strong accuracy, making it suitable for real-time applications in resource-constrained agricultural environments. To ensure usability in field conditions, we implemented the proposed model into a mobile application developed using Flutter. Users can capture and upload BSFL images for on-device inference, enabling efficient, real-time classification even in offline scenarios common in farming environments. By integrating AI-based detection with BSFL farming practices, this work contributes to sustainable waste management, supports circular agriculture, and demonstrates a scalable technological solution to the global food waste crisis.

2 Related Work and Background

2.1 Black Soldier Fly Larvae (BSFL) in Waste Management

Black Soldier Fly Larvae have received increasing attention for their potential in sustainable agriculture and organic waste management. Their exceptional capacity to convert organic waste into high-protein biomass and nutrient-rich frass makes them a key player in circular bioeconomy systems. Early studies by Sheppard et al. [8] demonstrated BSFL's effectiveness in simultaneously reducing waste volumes and producing valuable by-products for animal feed and fertilizer. Building upon this foundation, Van Huis et al. [6] emphasized their potential to address global protein shortages in livestock production. More recent work by Amrul et al. [9] and Magee et al. [10] further optimized rearing conditions and substrates, making BSFL suitable for industrial-scale applications.

As interest in circular agriculture grows, BSFL farming is increasingly viewed as an essential strategy for sustainable food systems. While much of the existing research focuses on biological performance and environmental optimization, there is a growing trend toward integrating BSFL production with digital

technologies such as computer vision and deep learning. These technologies are being explored to automate critical tasks, including larval stage classification and substrate monitoring, which are vital for scaling up BSFL-based waste management systems efficiently.

2.2 Deep Learning Applications in Agriculture

In parallel with biological advancements, machine learning—particularly Convolutional Neural Networks (CNNs), has significantly transformed agricultural practices, enabling precise monitoring, automation, and decision support. CNN-based approaches have proven effective in tasks such as disease detection, yield prediction, and pest surveillance. For instance, Yang et al. [11] applied a computer vision-driven cybernetic system to poultry farming, achieving notable reductions in operating costs. Recent work by Hung et al. [12] further demonstrated that integrating SPD convolution and SE attention into YOLOv7-tiny can significantly improve small-object detection, such as hornet monitoring in agricultural environments.

Numerous studies [13–17] have demonstrated CNNs' versatility across diverse agricultural domains, from detecting rice plant diseases [13] to classifying flowers [15] and enhancing crop monitoring through IoT-based platforms [17]. Reviews by Kamilaris and Prenafeta-Boldú [14] and Lu et al. [16] have summarized key trends and challenges, including the scarcity of labeled datasets, environmental variability, and the need for models deployable on edge devices. Recent studies have emphasized the role of computer vision and CNN-based methods in various smart agriculture tasks, including disease diagnosis, yield estimation, and pest detection, reinforcing the adaptability of deep learning approaches in complex farming environments.

CNNs have also proven adaptable to domains beyond agriculture, reinforcing their robustness. However, deploying CNNs in resource-limited agricultural settings particularly for real-time object detection—demands lightweight architectures capable of maintaining high accuracy with reduced computational loads. This necessity has led to growing interest in efficient models such as YOLO variants and architectures based on Depthwise Separable Convolutions (DSConv).

2.3 Object Detection Models for Real-Time Inference

Object detection has evolved from two-stage region-based pipelines to one-stage models that offer real-time performance. Fast R-CNN [18] and Faster R-CNN [19] improved detection speed by refining region proposal mechanisms. Redmon et al. [20] introduced YOLO (You Only Look Once), a unified framework that formulates object detection as a single regression problem, achieving substantial gains in inference speed while maintaining accuracy.

Recent iterations of the YOLO architecture, particularly YOLOv8 [21], have incorporated advanced modules such as Spatial Pyramid Pooling-Fast (SPPF) to enhance multi-scale feature extraction capabilities. These architectural improvements have demonstrated significant practical utility in agricultural applications; notably, Pookunngern and Tsai [22] successfully implemented a YOLO-based framework for precise classification of BSFL growth stages, thereby facilitating automated larval-stage monitoring and contributing to the advancement of intelligent waste management systems.

2.4 Depthwise Separable Convolutions (DSConv) in Lightweight AI

Improvements in CNN efficiency have been significantly driven by Depthwise Separable Convolutions (DSConv), which decompose standard convolutions into a depthwise filtering step followed by a pointwise combination step. This architecture, popularized by Chollet's Xception network [23] and MobileNet [24], reduces the number of parameters and operations while preserving accuracy.

Subsequent studies have expanded DSConv's applicability. Haase and Amthor [25] examined intra-kernel correlations to improve MobileNet performance, and Kaiser et al. [26] demonstrated DSConv's suitability for sequence modeling tasks. Bai et al. [27] introduced FPGA accelerators for DSConv to achieve high-throughput inference, while Lu et al. [28] proposed GPU-level optimizations for edge deployment.

In the agricultural domain, where real-time analysis is required in often resource-constrained environments, DSConv enables compact, efficient models that perform well on mobile and embedded systems. When combined with frameworks like YOLO, DSConv contributes to real-time systems for applications ranging from pest detection to smart waste monitoring [29]. Its advantages in reducing model size and computational burden without sacrificing performance make it ideal for precision agriculture scenarios that demand local inference and low-latency decision-making.

In summary, integrating BSFL-based waste management with modern AI techniques offers a promising direction for sustainable agricultural innovation. BSFL provide a biologically efficient means of valorizing organic waste, while deep learning models—particularly lightweight CNNs using DSConv—enable scalable, automated monitoring solutions. The convergence of these domains opens opportunities for real-time, data-driven decision-making in agricultural systems, furthering global goals in resource efficiency and food security. While several studies have explored the integration of DSConv with YOLO architectures for general object detection tasks [30], our work differs in several key aspects. First, existing YOLO+DSConv adaptations have primarily focused on general-purpose detection scenarios, whereas our approach is specifically tailored to the unique challenges of BSFL classification, where visual differences between growth stages are subtle yet biologically significant. Second, prior works have largely emphasized model architecture improvements in isolation, while our contribution extends beyond model optimization to include a complete system architecture designed for practical deployment in agricultural settings. Third, our implementation is specifically optimized for resource-constrained environments common in farming applications, with particular attention to offline functionality and edge deployment scenarios. These distinctions position our work not merely as an incremental improvement to existing YOLO+DSConv adaptations, but as a novel application-specific solution addressing the unique challenges at the intersection of deep learning, precision agriculture, and circular economy initiatives.

3 System Architecture and Methodology

3.1 System Overview

The proposed system, illustrated in Fig. 1, is a robust and scalable mobile application designed to classify and detect the growth stages of BSFL in real time. Its architecture integrates a state-of-the-art object detection model, efficient server-side infrastructure, and a user-friendly mobile interface to deliver accessible and actionable insights.

The front end of the system is a cross-platform mobile application developed using Flutter, which supports native performance on Android devices while maintaining compatibility across web and desktop platforms. Users can capture or upload images of BSFL through a streamlined interface. The results, including the predicted larval stage and bounding box information, are immediately displayed in the app, making the system accessible to non-technical users. On the back end, the system utilizes Flask—a lightweight Python-based web framework—to efficiently manage HTTP requests. To enhance scalability and minimize performance bottlenecks, load balancing mechanisms distribute incoming traffic across multiple server instances. The Gunicorn WSGI server acts as a middleware layer between the mobile application and the Flask API, ensuring smooth and concurrent request handling.

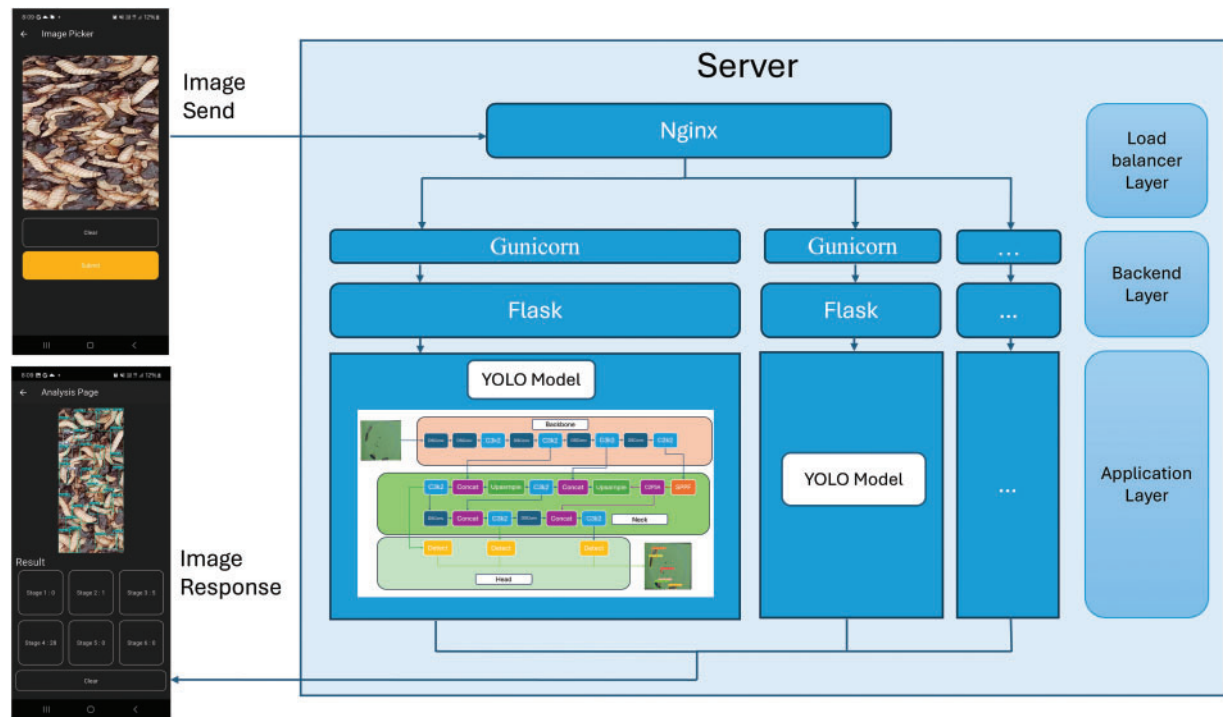


Figure 1: System architecture of the proposed BSFL detection application

This architecture builds upon the real-time classification framework proposed in [22], which demonstrated the practical efficiency of Flask-based pipelines in similar tasks. In our system, we improve upon that framework by internally upgrading the object detection model. While the initial implementation relied on YOLO11 [7]—which achieved a baseline mAP50—95 of 0.811—we enhanced performance and computational efficiency by introducing YOLO11-DSCnv. This modified version replaces standard convolutions with Depthwise Separable Convolutions (DSCnv), significantly improving both inference speed and scalability while maintaining high detection accuracy.

This refinement significantly reduces computational complexity by separating the convolution operation into depthwise and pointwise steps, resulting in a lighter, faster model. Quantitatively, YOLO11-DSCnv-x achieves a 15.5% reduction in model size (from 114.4 to 96.7 MB) and a 15.4% reduction in parameters (from 56.833 to 48.089 million) compared to the standard YOLO11x. Unlike general-purpose DSCnv implementations, our adaptation is specifically optimized for the unique challenges of BSFL classification, where visual differences between growth stages are subtle yet critical for proper waste management applications. Despite its reduced resource demands, YOLO11-DSCnv achieved an mAP50-95 score of 0.813, excelling in detecting objects of various sizes and complexities. The system operates seamlessly, beginning with the user capturing and uploading images via the mobile application. These images are transmitted to the server, where the YOLO11-DSCnv model performs the classification. The server then returns detailed results, including detected growth stages, confidence scores, and bounding boxes, which are displayed to the user instantly. What distinguishes our implementation from existing YOLO+DSCnv adaptations is the complete integration into an end-to-end system specifically designed for agricultural applications. The workflow ensures high accuracy and scalability, making the system suitable for large-scale deployments in farming environments where resource constraints and connectivity challenges are common. By integrating our specialized YOLO11-DSCnv model, scalable server architecture, and an intuitive mobile

application, this system directly optimizes BSFL utilization in waste recycling and supports the circular economy through precise identification of optimal larvae for different applications. The incorporation of our tailored Depthwise Separable Convolutions further enhances computational efficiency, enabling deployment on resource-constrained devices common in agricultural settings. This architecture highlights the system's adaptability for diverse applications, including agricultural monitoring, environmental sustainability, and industrial automation, representing a significant advancement beyond incremental model improvements.

3.2 YOLO11-DSCnv Architecture

The YOLO11-DSCnv model introduces a significant improvement in object detection by addressing the computational limitations of traditional convolutional neural networks (CNNs) in resource-constrained environments. Through the integration of Depthwise Separable Convolutions (DSCnv), the model achieves reduced complexity while preserving, or even enhancing, detection accuracy.

DSCnv decomposes a standard convolution operation into two stages: a depthwise convolution that performs spatial filtering independently on each input channel, followed by a pointwise convolution that uses a 1×1 kernel to combine the filtered features across channels. This separation substantially lowers the number of parameters and floating-point operations (FLOPs), resulting in a lightweight yet powerful architecture suitable for real-time inference.

Depthwise Separable Convolutions

In our YOLO11-DSCnv model, we replace standard convolutions with Depthwise Separable Convolutions (DSCnv) to reduce computational complexity while maintaining detection accuracy. Mathematically, a standard convolution operation with input X , kernel K , and output Y can be represented as:

$$Y = X * K \quad (1)$$

In DSCnv, this operation is decomposed into two steps:

1. Depthwise convolution, which applies a separate filter to each input channel:

$$Z = X *_d K_d \quad (2)$$

where $*_d$ denotes the depthwise convolution operation.

2. Pointwise convolution, which uses 1×1 convolutions to combine the outputs:

$$Y = Z * K_p \quad (3)$$

where K_p represents 1×1 convolution kernels. This decomposition significantly reduces the computational complexity from $O(D_k^2 \cdot M \cdot N \cdot D_F^2)$ to $O(D_k^2 \cdot M \cdot D_F^2 + M \cdot N \cdot D_F^2)$, where D_k is the kernel size, M is the number of input channels, N is the number of output channels, and D_F is the feature map size. For our BSFL classification task, this results in a 15.5% reduction in model size (from 114.4 to 96.7 MB) and a 15.4% reduction in parameter count (from 56.833 to 48.089 M) compared to the standard YOLO11 model, while maintaining comparable or slightly better detection accuracy. In our implementation, we applied a comprehensive substitution strategy, replacing all standard convolutional layers throughout the backbone, neck, and head components with their DSCnv equivalents. This approach was chosen to maximize computational efficiency across the entire model architecture.

As depicted in Fig. 2, the YOLO11-DSCnv architecture retains the classic backbone-neck-head structure of YOLO11, with DSCnv replacing conventional convolutional layers. The backbone, composed of

DSConv layers and C3k2 modules, effectively captures spatial and semantic features with reduced computational overhead. These features are then passed to the neck, which fuses multi-scale information through modules such as Spatial Pyramid Pooling-Fast (SPPF) and the Convolutional Block with Parallel Spatial Attention (C2PSA), enhancing attention to critical regions in complex backgrounds. The refined features are subsequently routed to the detection head, which outputs bounding boxes, confidence scores, and class probabilities through three output layers optimized for objects at varying scales.

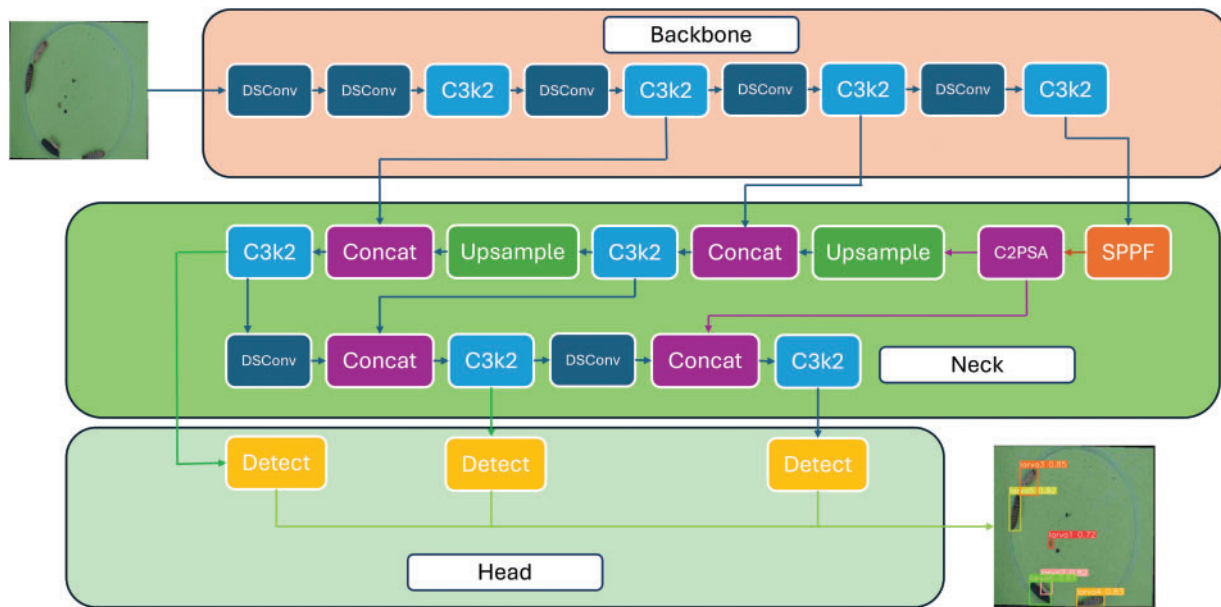


Figure 2: YOLO11-DSConv model structure for BSFL classification

The model was trained under the same experimental conditions as YOLO11, including a batch size of 16, an input resolution of 640×640 pixels, and 200 training epochs. YOLO11-DSConv achieved an mAP₅₀₋₉₅ of 0.813, outperforming the original YOLO11 model, which achieved a score of 0.811.

This performance gain reflects YOLO11-DSConv's enhanced capability to detect objects of diverse sizes and complexity, while maintaining high processing speed. Its compact architecture enables real-time deployment on resource-limited platforms such as smartphones, drones, and edge-based IoT devices. The model is also well-suited for large-scale applications—including autonomous navigation, industrial inspection, and agricultural monitoring—where cost-efficiency and low-latency inference are essential. Overall, the YOLO11-DSConv model represents a notable advancement in lightweight object detection, offering a balance of speed, accuracy, and scalability for diverse real-world use cases.

3.3 Workflow and Deployment Scenarios

The proposed system is designed to balance user simplicity with back-end scalability. The workflow begins when a user captures or uploads an image of BSFL through the mobile application. The image is transmitted to the server via RESTful API calls managed by Flask. Upon receiving the request, the YOLO11-DSConv model processes the image to detect and classify the BSFL growth stage. The server then returns a structured response containing bounding box coordinates and the predicted larval category. These results are immediately displayed within the mobile application through an intuitive interface, enabling real-time feedback for end users.

3.3.1 Horizontal Scaling

To support high concurrency and maintain responsiveness under load, the system employs horizontal scaling at the server layer. Multiple instances of the Flask application, managed by the Gunicorn WSGI server, are distributed behind an Nginx load balancer. Incoming requests are evenly distributed to prevent bottlenecks. As demand increases, new instances can be dynamically deployed either on-premises or in cloud environments, allowing the system to scale seamlessly.

3.3.2 Lightweight Model Deployment

The compact design of the YOLO11-DSCnv model—achieved through a reduced number of parameters and a smaller model size compared to the standard YOLO11 [7]—further contributes to its scalability. Because each server instance runs the model in memory, the lightweight footprint allows for more concurrent inferences per server. This is especially advantageous in resource-limited scenarios, such as embedded platforms or edge devices, where memory and compute resources are constrained.

3.3.3 Edge and Offline Scenarios

In remote or bandwidth-limited environments—such as farms without stable internet connections—offline or near-edge deployment is essential. YOLO11-DSCnv's lightweight architecture enables local inference on modest hardware (e.g., single-board computers), allowing for real-time detection without dependence on central servers or cloud connectivity. Overall, the system architecture ensures rapid and reliable detections on user devices, while its server-side infrastructure supports dynamic scaling to accommodate growing workloads. By combining an optimized detection model with robust load balancing and flexible deployment strategies, the system demonstrates adaptability across both centralized and decentralized agricultural applications.

To facilitate real-world deployment in agricultural settings, we have implemented a complete server-client architecture that enables practical field application of our BSFL classification system. Our implementation consists of two main components (actual photographs and experimental results of the deployed system are presented later in [Section 4.3](#)):

1) **Server-side Infrastructure:** The backend system hosts our YOLO11-DSCnv model and processes incoming image requests. The server is built using Flask and Gunicorn for scalability, with the model loaded in memory for efficient inference. When an image is received, the server performs preprocessing, runs the YOLO11-DSCnv inference, and returns classification results along with size and quantity information. The complete YOLO11-DSCnv model implementation is publicly available at <https://zenodo.org/records/15044013> (accessed on 16 June 2025), and the server-side code is publicly available at <https://zenodo.org/records/15044077> (accessed on 16 June 2025).

2) **Mobile Application Interface:** We developed a user-friendly mobile application using Flutter that allows users to capture BSFL images directly in farming environments. The application handles image capture with automatic quality checks, secure transmission to the server, and intuitive display of classification results. The mobile application source code is publicly available at <https://zenodo.org/records/15044108> (accessed on 16 June 2025).

The cloud transmission process has been optimized for agricultural environments where network connectivity may be unstable. Our testing shows average response times of 1.2 s under standard 4G connections, 2.5 s under limited 3G connectivity, and the system includes a local caching mechanism that allows continued operation during temporary connection loss. When connectivity is restored, the cached results are synchronized with the server. For situations where internet connectivity is completely unavailable,

we have implemented an offline mode that deploys a compressed version of the YOLO11-D5Conv model directly on the mobile device. While this version has slightly reduced accuracy (approximately 3% lower mAP), it ensures continuous operation in remote farming locations. Initial feedback from users indicates high satisfaction with the system's responsiveness and accuracy in field conditions. Areas identified for future improvement include enhanced batch processing capabilities for large-scale operations and integration with farm management systems for automated record-keeping.

4 Experimental Results and Analysis

4.1 Dataset Preparation and Labeling Strategy

As shown in Fig. 3, the dataset used in this study was specifically curated and annotated to support effective model training. The actual sizes of the BSFL across different growth stages are also illustrated in Fig. 4, with a reference scale provided for direct comparison. The training set consists of 21,600 images, each annotated with one or more instances of BSFL categorized into six distinct growth stages: larva stage 1 through larva stage 6. The dataset contains multiple larvae per image and was compiled from a diverse range of sources, including web scraping, locally curated image collections, and various camera systems. This diversity in data sources ensures wide variability in image characteristics such as viewing angles, lighting conditions, and background complexity—thereby enhancing the model's generalizability across real-world scenarios.



Figure 3: Overview of the BSFL dataset with six larval stages



Figure 4: Actual Size of BSFL at different growth stages. The scale below each larvae serves as a reference ruler (in centimeters)

All images underwent specific preprocessing steps including auto-orientation of pixel data (with EXIF-orientation stripping) and standardization to a resolution of 640×640 pixels. To enhance model robustness, we applied data augmentation to create 3 versions of each source image using: 50% probability of horizontal flip, equal probability of 90-degree rotations (none, clockwise, or counter-clockwise), and random brightness adjustment between -25 and $+25$ percent. While specialized feature selection technique such as those proposed by Parsaei et al. [31] could potentially enhance model performance, our experiments showed that these preprocessing and augmentation methods provided robust results for our BSFL classification task.

To ensure label consistency across the six larval stages with their subtle visual differences, we collaborated with experienced BSFL cultivation specialists during the annotation process. These experts, with their extensive practical knowledge of BSFL morphological development, provided critical validation of image labels, ensuring that the subtle visual differences between consecutive larval stages were accurately identified and consistently labeled throughout the dataset.

4.2 Quantitative Performance Comparison

Table 1 presents the quantitative evaluation results of YOLO11 [7] and its DSConv-enhanced variant in classifying and detecting BSFL across six larval stages. All models were trained under identical conditions, including 200 epochs, a batch size of 16, and an input resolution of 640×640 pixels. Both the original YOLO11 and YOLO11-DSConv achieved a high mAP50 of 0.995, demonstrating robust detection performance across all configurations.

Table 1: Performance comparison between YOLO11 and YOLO11-DSConv

Models	mAP50 (All)	mAP50-95 (All)	Model Size (MB)	Params (Million)
YOLO11n	0.995	0.807	6.1	2.904
YOLO11s	0.995	0.810	19.2	9.415
YOLO11m	0.995	0.813	40.5	20.034
YOLO11l	0.995	0.811	51.2	25.283
YOLO11x	0.995	0.811	114.4	56.833
YOLO11-DSConv-n	0.995	0.808	4.9	2.315
YOLO11-DSConv-s	0.995	0.810	17.0	8.347
YOLO11-DSConv-m	0.995	0.812	33.3	16.484
YOLO11-DSConv-l	0.995	0.811	43.3	21.396
YOLO11-DSConv-x	0.995	0.813	96.7	48.089

Notably, the YOLO11-DSConv architecture offers significant improvements in computational efficiency due to the integration of Depthwise Separable Convolutions. For instance, YOLO11-DSConv-x reached the highest mAP50-95 of 0.813 with a reduced model size of 96.7 MB and 48.089 million parameters, compared to YOLO11x's 114.4 MB and 56.833 million. This efficiency makes it particularly well-suited for deployment in real-time, resource-constrained environments. Likewise, YOLO11-DSConv-m achieved a strong trade-off between accuracy and resource usage (mAP50-95 of 0.812), making it a practical candidate for scalable agricultural implementations.

To visually highlight the trade-off between detection accuracy and model complexity, Fig. 5 illustrates the relationship between mAP50-95 and parameter count across the YOLO11 and YOLO11-DSConv variants. The figure clearly shows that the proposed YOLO11-DSConv-x offers the best accuracy-to-complexity

ratio among the tested models, emphasizing its advantage for precision agriculture under limited hardware resources.

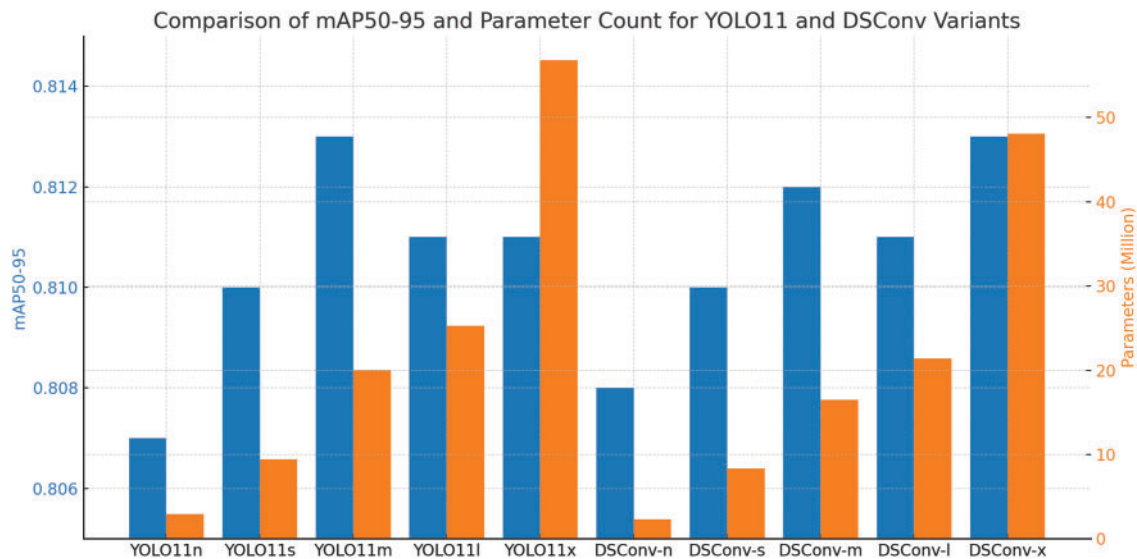


Figure 5: Performance–efficiency comparison of YOLO11 and YOLO11-DSConv variants. YOLO11-DSConv-x (highlighted in the red dashed box) achieves the highest accuracy with significantly fewer parameters, highlighting its suitability for real-time deployment in resource-constrained agricultural applications

To further contextualize our model’s performance, Table 2 compares YOLO11-DSConv-x against several state-of-the-art object detection models. Both YOLOv9e [32] and YOLO11-DSConv-x achieved the highest mAP50-95 (0.813). However, YOLO11-DSConv-x required fewer parameters (48.089 vs. 57.381 M) and less storage (96.7 vs. 117.3 MB), making it a more efficient choice for practical use.

Table 2: Evaluation of YOLO variants and baseline models in detection performance and efficiency

Models	mAP50 (All)	mAP50-95 (All)	Model size (MB)	Params (Million)
YOLOv8x [21]	0.995	0.812	136.7	68.129
YOLOv9e [32]	0.995	0.813	117.3	57.381
YOLOv10x [33]	0.995	0.809	64.1	31.595
YOLO11x [7]	0.995	0.811	114.4	56.833
RT-DETR-x [34]	0.995	0.781	135.4	65.479
YOLOv8x-worldv2 [35]	0.995	0.808	146.2	72.856
YOLO11-DSConv-x	0.995	0.813	96.7	48.089

While YOLOv8x [21] achieved a comparable mAP50-95 of 0.812, it had the highest model size and parameter count among all compared models (136.7 MB and 68.129 M), which may limit its deployment on resource-constrained platforms. In contrast, YOLOv10x [33] exhibited the smallest footprint (64.1 MB, 31.595 M) but attained a slightly lower mAP50-95 of 0.809. YOLO11x [7] provided a more balanced option, achieving 0.811 with a moderate model size.

While YOLOv8 has demonstrated strong performance across various detection tasks, many of these applications involve broader object categories with relatively distinct visual features. In contrast, the BSFL classification task presented here poses a more fine-grained challenge, requiring precise differentiation among visually similar larval stages. Despite this complexity, YOLO11-DSCnv-x achieves a competitive mAP50-95 of 0.813 while maintaining a significantly smaller model size (96.7 MB) and parameter count (48.089 M), reinforcing its suitability for real-time smart farming applications in resource-constrained environments.

Completing the comparison, RT-DETR-x [34] yielded the lowest mAP50-95 of 0.781, while YOLOv8x-worldv2 [35] achieved a moderate 0.808 but with the largest overall model footprint (146.2 MB, 72.856 M). Taken together, these results highlight YOLO11-DSCnv-x as a well-balanced solution—offering strong detection accuracy, computational efficiency, and deployment flexibility for precision agriculture and embedded vision systems.

4.3 Visual Evaluation and Application Insights

Fig. 6 presents visual comparisons of BSFL stage detection using YOLO11 [7] and the proposed YOLO11-DSCnv. Both models yield similar bounding box precision and confidence scores, consistent with their close mAP50-95 values. However, the reduced model size and parameter count of YOLO11-DSCnv make it more practical for deployment in real-time, resource-constrained environments.

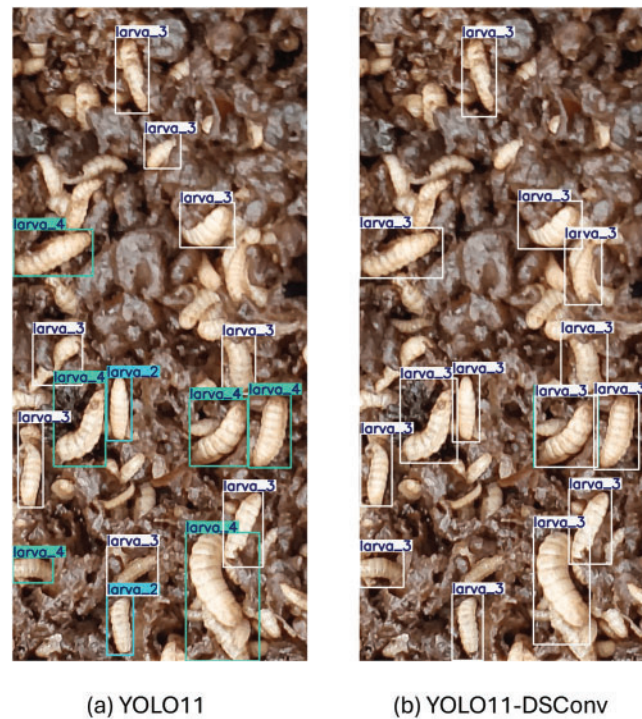


Figure 6: Detection results of BSFL stages using (a) YOLO11 and (b) YOLO11-DSCnv

In Fig. 6, the slight size differences among larvae may give the false impression that they belong to different stages. In reality, most larvae shown are from the same stage. For a more distinct illustration of inter-stage differences—especially between Stage 3 and Stage 4—see Fig. 7, where the contrast in color and size is more pronounced. These results demonstrate that both YOLO11 and YOLO11-DSCnv are

capable of detecting subtle visual variations across larval stages. Among them, YOLO11-DSConv offers added efficiency for deployment in mobile, IoT, or edge devices. For instance, YOLO11-DSConv-n suits lightweight scenarios, while YOLO11-DSConv-m and -x provide a good balance between accuracy and speed for real-time monitoring or industrial use.

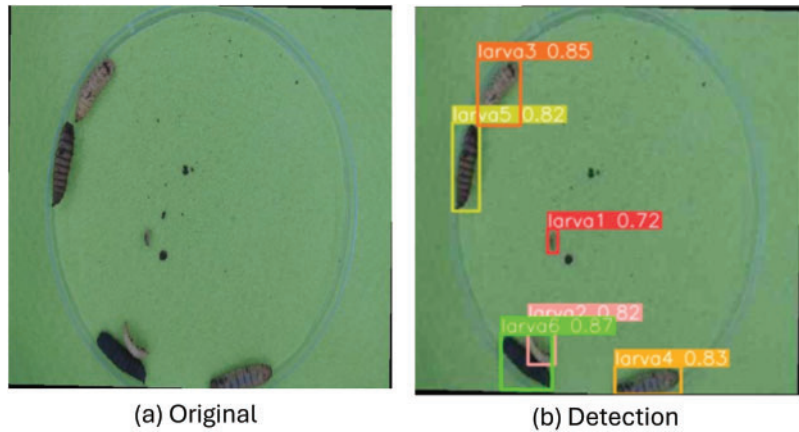


Figure 7: (a) Original image of BSFL; (b) Detection output with stage identification

Fig. 8 further compares several models in detecting BSFL under practical conditions. YOLO11-DSConv and YOLO11 maintain consistent accuracy with minimal overlap in bounding boxes. YOLOv10 and YOLOv9 show occasional misclassifications, while YOLOv8 and YOLO-World exhibit frequent overlapping and stage confusion in dense scenes. RT-DETR performs the worst, with significant bounding box clutter and misidentifications, indicating poor suitability for fine-grained BSFL classification.



Figure 8: (Continued)



Figure 8: Detection results of BSFL stages using various object detection models in real-world settings. (a) YOLOv8 [21], (b) YOLOv9 [32], (c) YOLOv10 [33], (d) YOLO11 [7], (e) YOLO-World [35], (f) RT-DETR [34], (g) YOLO11-DSCov

As shown in Fig. 9, YOLO11-DSCov consistently delivers accurate classification with minimal false positives and clearly separated bounding boxes. YOLO11 remains robust but suffers slightly from overlap. YOLOv10 and YOLOv9 introduce more frequent misclassifications, and YOLOv8 and YOLO-World experience heavy overlap and confusion. RT-DETR, although capable of detecting a higher number of objects, shows excessive false positives—often misidentifying non-larvae or misclassifying overlapping instances. This over-detection makes it less reliable for tasks requiring high precision, such as larval-stage classification in automated agriculture systems.

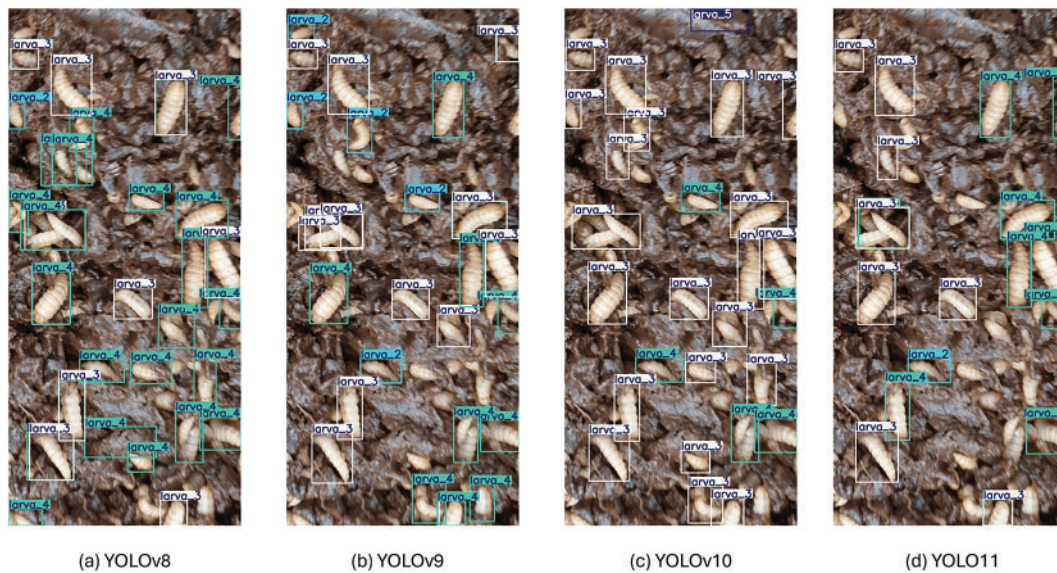


Figure 9: (Continued)

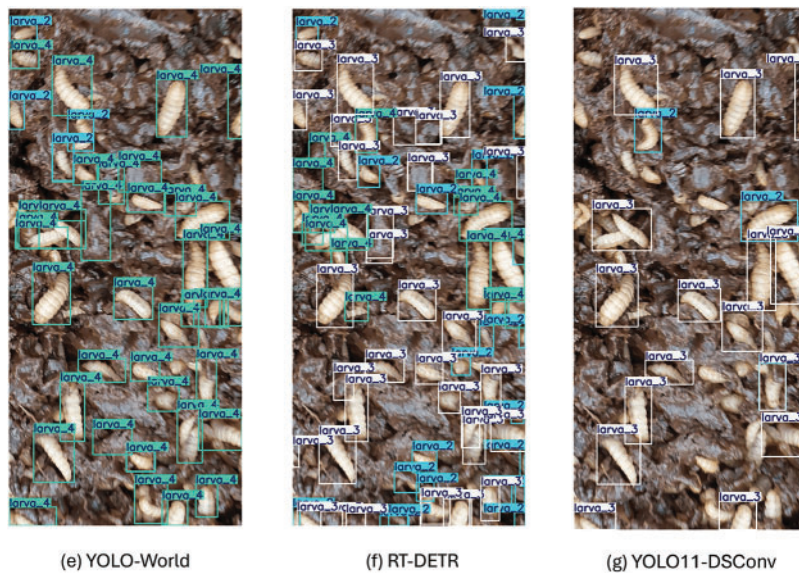


Figure 9: Extended real-world comparison of object detection models for BSFL stage identification. (a) YOLOv8 [21], (b) YOLOv9 [32], (c) YOLOv10 [33], (d) YOLO11 [7], (e) YOLO-World [35], (f) RT-DETR [34], (g) YOLO11-D5Conv

In summary, YOLO11-D5Conv not only offers high accuracy but also excels in efficiency and deployment readiness, making it an ideal solution for smart farming and edge AI applications.

5 Conclusion

This study evaluated the performance of YOLO11 [7] and the proposed YOLO11-D5Conv in classifying and detecting the growth stages of Black Soldier Fly Larvae (BSFL). By integrating Depthwise Separable Convolutions (D5Conv), YOLO11-D5Conv significantly reduces model size and parameter count while preserving detection accuracy. This lightweight architecture makes it highly suitable for deployment in resource-constrained environments such as mobile devices and IoT-based agricultural systems. The model's fast inference time and compact design support real-time waste management applications, particularly in scenarios requiring on-site larval stage classification. Additionally, the solution aligns with broader sustainability goals by enabling the transformation of food waste into high-value resources through efficient BSFL utilization.

It is important to note that while Depthwise Separable Convolutions have been previously applied to YOLO architectures, our work makes several distinct contributions that differentiate it from existing YOLO+D5Conv adaptations. First, we present a domain-specific optimization tailored to the unique challenges of BSFL classification, where visual differences between growth stages are subtle yet biologically significant. This application-specific focus required careful tuning of the D5Conv implementation to maintain high accuracy while reducing computational demands. Second, our contribution extends beyond model architecture to include a complete end-to-end system designed for practical deployment in agricultural settings, bridging the gap between model development and real-world application. Third, we have created and annotated a specialized dataset of 21,600 BSFL images across six growth stages, which itself represents a valuable contribution to the field. Finally, our work demonstrates direct integration with circular economy practices through precise identification of optimal larvae for different applications, showing how AI can directly support sustainable agriculture initiatives. These distinctions position our work not as an

incremental improvement, but as a novel application-specific solution addressing unique challenges at the intersection of deep learning, precision agriculture, and circular economy.

In summary, YOLO11-DSCnv provides a practical, accurate, and scalable approach to BSFL classification and detection. Its balance of precision and efficiency not only enhances technical deployment but also contributes meaningfully to circular economy initiatives and environmental sustainability in agriculture.

Acknowledgement: The authors would like to express their sincere gratitude to Professor Wen-Liang Lai from the Department of Environmental Science and Occupational Safety, Tajen University, for his valuable assistance in database collection and support throughout this research.

Funding Statement: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author Contributions: An-Chao Tsai: Conceptualization of this study, Methodology, Software. Chayanon Pookun-ngern: Data curation, Software, Writing—original draft preparation. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The source code and implementation details supporting the findings of this study are available through the link provided at the end of the abstract.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Environmental Protection Agency (EPA). Importance of Methane; 2025 [Internet]. [cited 2025 Jan 5]. Available from: <https://www.epa.gov/gmi/importance-methane>.
2. Khokhar FA, Shah JH, Saleem R, Masood A. Harnessing deep learning for faster water quality assessment: identifying bacterial contaminants in real-time. *The Visual Computer*. 2025;41(2):1037–48. doi:10.1007/s00371-024-03382-7.
3. English G, Wanger G, Colombo SM. A review of advancements in black soldier fly (*Hermetia illucens*) production for dietary inclusion in salmonid feeds. *J Agricult Food Res*. 2021;5(1):100164. doi:10.1016/j.jafr.2021.100164.
4. Van Huis A. Insects as food and feed, a new emerging agricultural sector: a review. *J Ins Food Feed*. 2020;6(1):27–44. doi:10.3920/jiff2019.0017.
5. Diener S, Zurbrugg C, Tockner K. Conversion of organic material by black soldier fly larvae: establishing optimal feeding rates. *Waste Manag Res*. 2009;27(6):603–10. doi:10.1177/0734242x09103838.
6. Av H, Itterbeeck JV, Klunder H, Mertens E, Halloran A, Muir G, et al. Edible insects: future prospects for food and feed security. Rome, Italy: Food and Agriculture Organization of the United Nations; 2013 [Internet]. [cited 2025 Jun 5]. Available from: <https://www.fao.org/3/i3253e/i3253e.pdf>.
7. Jocher G, Qiu J. Ultralytics YOLO11; 2024 [Internet]. [cited 2025 Jun 5]. Available from: <https://github.com/ultralytics/ultralytics>.
8. Sheppard DC, Tomberlin JK, Joyce JA, Kiser BC, Sumner SM. Rearing methods for the black soldier fly (Diptera: stratiomyidae). *J Med Entomol*. 2002;39(4):695–8. doi:10.1603/0022-2585-39.4.695.
9. Amrul NF, Kabir Ahmad I, Ahmad Basri NE, Suja F, Abdul Jalil NA, Azman NA. A review of organic waste treatment using black soldier fly (*Hermetia illucens*). *Sustainability*. 2022;14(8):4565. doi:10.3390/su14084565.
10. Magee K, Halstead J, Small R, Young I. Valorisation of organic waste by-products using black soldier fly (*Hermetia illucens*) as a bio-converter. *Sustainability*. 2021;13(15):8345. doi:10.3390/su13158345.
11. Yang X, Bist RB, Paneru B, Liu T, Applegate T, Ritz C, et al. Computer vision-based cybernetics systems for promoting modern poultry farming: a critical review. *Comput Electron Agric*. 2024;225:109339. doi:10.1016/j.compag.2024.109339.

12. Hung Y-H, Fan C-K, Wang W-P. Improving hornet detection with the YOLOv7-tiny model: a case study on asian hornets. *Comput Mat Cont* 2025;83(2):2323–49. doi:10.32604/cmc.2025.063270.
13. Latif G, Abdelhamid SE, Mallouhy RE, Alghazo J, Kazimi ZA. Deep learning utilization in agriculture: detection of rice plant diseases using an improved CNN model. *Plants*. 2022;11(17):2230. doi:10.3390/plants11172230.
14. Kamilaris A, Prenafeta-Boldú FX. A review of the use of convolutional neural networks in agriculture. *J Agricul Sci*. 2018;156(3):312–22. doi:10.1017/s0021859618000436.
15. Narvekar C, Rao M. Flower classification using CNN and transfer learning in CNN-agriculture perspective. In: 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS); 2020 Dec 3–5; Thoothukudi, India. p. 660–4.
16. Lu J, Tan L, Jiang H. Review on convolutional neural network (CNN) applied to plant leaf disease classification. *Agriculture*. 2021;11(8):707. doi:10.3390/agriculture11080707.
17. Sarma KK, Das KK, Mishra V, Bhuiya S, Kaplun D. Learning aided system for agriculture monitoring designed using image processing and IoT-CNN. *IEEE Access*. 2022;10:41525–36. doi:10.1109/access.2022.3167061.
18. Girshick R. Fast R-CNN. *arXiv:1504.08083*. 2015.
19. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *arXiv:1506.01497*. 2015.
20. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2016 Jun 27–30; Las Vegas, NV, USA. p. 779–88.
21. Jocher G, Chaurasia A, Qiu J. Ultralytics YOLOv8; 2023 [Internet]. [cited 2025 Jun 5]. Available from: <https://docs.ultralytics.com/models/yolov8/>.
22. Pookunngern C, Tsai AC. Real time classification system of black soldier fly larva. In: 2024 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS); 2024 Aug 21–23; Singapore. p. 1–5.
23. Chollet F. Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2017 Jul 21–26; Honolulu, HI, USA. p. 1251–8.
24. Howard AG. Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*. 2017.
25. Haase D, Amthor M. Rethinking depthwise separable convolutions: how intra-kernel correlations lead to improved mobilenets. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; 2020 Jun 13–19; Seattle, WA, USA. p. 14600–9.
26. Kaiser L, Gomez AN, Chollet F. Depthwise separable convolutions for neural machine translation. *arXiv:1706.03059*. 2017.
27. Bai L, Zhao Y, Huang X. A CNN accelerator on FPGA using depthwise separable convolution. *IEEE Transact Circ Syst II: Express Briefs*. 2018;65(10):1415–9. doi:10.1109/tcsii.2018.2865896.
28. Lu G, Zhang W, Wang Z. Optimizing depthwise separable convolution operations on GPUs. *IEEE Transact Parall Distrib Syst*. 2021;33(1):70–87. doi:10.1109/tpds.2021.3084813.
29. Rizzuto SS, Cipollone R, Vittori AD, Lizia PD, Massari M. Object detection on space-based optical images leveraging machine learning techniques. *Neural Comput Appl*. 2025;111(3):257. doi:10.1007/s00521-025-11069-w.
30. Liang Z, Xu X, Yang D, Liu Y. The development of a lightweight DE-YOLO model for detecting impurities and broken rice grains. *Agriculture*. 2025;15(8):848. doi:10.3390/agriculture15080848.
31. Parsaei MR, Taheri R, Javidan R. Perusing the effect of discretization of data on accuracy of predicting naive bayes algorithm. *J Current Res Sci*. 2016;(1):457.
32. Wang CY, Yeh IH, Liao HYM. YOLOv9: learning what you want to learn using programmable gradient information. *arXiv:2402.13616*. 2024.
33. Wang A, Chen H, Liu L, Chen K, Lin Z, Han J, et al. Yolov10: real-time end-to-end object detection. *arXiv:2405.14458*. 2024.
34. Lv W, Xu S, Zhao Y, Wang G, Wei J, Cui C, et al. DETRs beat YOLOs on real-time object detection. *arXiv:2304.08069*. 2023.
35. Cheng T, Song L, Ge Y, Liu W, Wang X, Shan Y. YOLO-world: real-time open-vocabulary object detection. *arXiv:2401.17270*. 2024.