Autonomous Tomato Harvesting With Top–Down Fusion Network for Limited Data

Xingxu Li, Yiheng Han, Nan Ma, *Senior Member, IEEE*, Yongjin Liu, *Senior Member, IEEE*, Jia Pan, *Senior Member, IEEE*, Shun Yang, and Siyi Zheng

***Abstract*—Using robots for tomato truss harvesting represents a promising approach to agricultural production. However, incomplete acquisition of perception information and clumsy operations often results in low harvest success rates or crop damage. To addressthis issue, we designed a new method for tomato truss perception, an autonomous harvesting method, and a novel circular rotary cutting end-effector. The robot performs object detection and keypoint detection on tomato trusses using the proposed top–down fusion network, making decisions on suitable targets for harvesting based on phenotyping and pose estimation. The designed end-effector moves gradually from the bottom up to wrap around the tomato truss, cutting the peduncle to complete the harvest. Experiments conducted in real-world scenarios for robotic perception and autonomous harvesting of tomato trusses show that the proposed method increases accuracy by up to 11.42% and 22.29% for complete and limited dataset conditions, compared to baseline models. Furthermore, we have implemented an automatic tomato harvesting system based on TDFNet, which reaches an average harvest success rate of 89.58% in the greenhouse.**

***Index Terms*—Agriculture robot, autonomous manipulation, deep learning, plant phenotyping, pose estimation, precision agriculture.**

Received 31 December 2024; revised 24 March 2025; accepted 16 April 2025. Date of publication 6 May 2025; date of current version 13 June 2025. This work was supported in part by the National Key R&D Program of China under Grant 2023YFF0615800, Grant 2023YFD2001200, and Grant 2024YFD2000800, in part by the National Natural Science Foundation of China under Grant 62371013 and Grant 62461160309, in part by the Beijing Natural Science Foundation under Grant 4222025, in part by the Beijing Postdoctoral Science Foundation under Grant 2024-ZZ-23, in part by the NSFC-RGC Joint Research Scheme under Grant N_HKU705/24, in part by the QIYUAN LAB Innovation Foundation for Innovation Research under Grant S20210201107. This article was recommended for publication by Associate Editor Andrea Cherubini and Editor Sven Behnke upon evaluation of the reviewers' comments. *(Xingxu Li and Yiheng Han are co-first authors.) (Corresponding author: Nan Ma.)*

Xingxu Li is with the School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China, and also with the Beijing AIForce Technology Company Ltd., Beijing 100102, China (e-mail: lixingxu@emails.bjut.edu.cn).

Yiheng Han and Nan Ma are with the School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China (e-mail: hanyiheng@bjut.edu.cn; manan123@bjut.edu.cn).

Yongjin Liu is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: liuyongjin@tsinghua.edu.cn).

Jia Pan is with the Department of Computer Science, University of Hong Kong, Hong Kong, SAR 999077, China (e-mail: jpan@cs.hku.hk).

Shun Yang and Siyi Zheng are with the Beijing AIForce Technology Company Ltd., Beijing 100102, China (e-mail: yangshun@aiforcetech.com; zhengsiyi@aiforcetech.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TRO.2025.3567544, provided by the authors.

Digital Object Identifier 10.1109/TRO.2025.3567544

NOMENCLATURE

Related Notations for the Data Structure of Tomato Trusses and Fruits

Symbol	Description.
t^i	Represents the ith detected tomato truss target.
b_t, b_f	2-D bounding boxes for the tomato truss target and the fruit.
K_p, K_f	Represents the keypoint sets, including the peduncle structure points and the fruit center points.
k_p, k_f	Key points for the peduncle and fruits.
f^v	vth fruit in the set of fruits contained in a target.
rip_t, rip_f	Ripeness of the tomato truss and the fruit. rip_t is derived during the phenotyping stage by aggregating the rip_f values of all fruits $(f^v)^{v=1,\ldots,u}$ associated with the target.
d	Diameter of the fruit.

I. INTRODUCTION

IN THE future, agricultural production will increasingly rely on automation to address labor shortages caused by aging populations, urbanization, and other factors, while also ensuring sustainable development and expanding production scale [1]. Harvesting robots offer a promising solution to supplement labor in labor-intensive production segments and reduce the physical burden on workers [2].

In robotic manipulation, accurate perception and autonomous trajectory setting are the core challenges [3]. Traditional industrial applications, such as robotic palletizing and welding [4], are performed in static and well-defined environments with minimal uncertainties. In contrast, agricultural work environments present dynamic and continuously changing characteristics. Over long time scales, plant growth continuously alters the spatial layout. During operation, changes in lighting conditions and the robot's actions further influence the environment. Furthermore, the targets in agricultural tasks are fundamentally different from rigid industrial components. They are soft and deformable crops with diverse shapes and textures, making them significantly more challenging to perceive and manipulate in dynamic environments. These variations in the environment and target uncertainty introduce challenges to the key steps of crop perception, trajectory setting, and execution for harvesting robots.

To achieve autonomous operation in such environments, robust vision-based systems are essential. However, agricultural

tasks introduce unique challenges. The diversity in crop morphology and the dynamic nature of the environment significantly increase the difficulty of accurate perception [3]. Lightweight models designed to meet real-time performance constraints often lack the capacity to effectively process complex features. Moreover, the scarcity of high-quality annotated datasets restricts the ability of models to generalize, particularly when dealing with diverse tomato trusses and visually similar backgrounds. Existing studies have primarily employed deep learning methods [5], [6] for single- or multistage perception tasks. However, these methods typically require task-specific datasets, further exacerbating data demands in resource-constrained agricultural contexts.

Unlike the palletization applications of robotical arms in traditional industrial scenarios, and single-fruit harvesting crops, such as strawberries [7], apples [8], [9], tomatoes [10], and bell peppers [11], cluster-grown crops represented by grape clusters [12] and tomato trusses cannot be harvested by grabbing and pulling. Instead, it is necessary to cut the peduncle of the crop to separate the target from the plant. To avoid damage to the crop plant or surrounding facilities, which can lead to loss of commodity value or production accidents, more accurate and comprehensive perception methods and reasonable robotic harvesting methods are needed.

To enhance the success rate and safety of autonomous continuous harvesting of tomato trusses, we have developed a comprehensive set of solutions that address key stages in the process; see Fig. 1.

In the perception stage, we propose top–down fusion network (TDFNet) for object detection and keypoint detection of tomato trusses. By integrating features from different processing stages, the keypoint detection task in the downstream phase directly utilizes the rich feature representations extracted from the object detection task in the upstream phase. This approach effectively enhances fitting ability and learning efficiency by exploiting the correlations between different tasks. In addition, the upstream and downstream models can remain independent, allowing for the replacement or addition of datasets corresponding to the upstream and downstream tasks, enhancing scenario adaptability. This is particularly crucial for achieving high precision and robustness in intelligent systems within agricultural applications, where crop morphology is diverse, sample availability is limited, and annotation costs are high. Postprocessing methods, such as grouping and matching, are used to generate the feature encoding of the tomato truss and restore its posture in the 3-D space, providing input for subsequent decision-making and trajectory design.

In the trajectory setting and execution stages, we designed a novel circular end-effector and a harvesting method based on the target posture. The action is similar to catching insects with a net. The robotic arm guides the end-effector from below the target, gradually enclosing it before positioning the blade at the peduncle for precise cutting. After separating the tomato truss from the vine, it falls into the net, completing the harvest. The robot calculates collision risks and adjusts its trajectory based on the target posture, minimizing contact between the end-effector and the target, thereby improving the success rate of harvesting and reducing damage to the target.

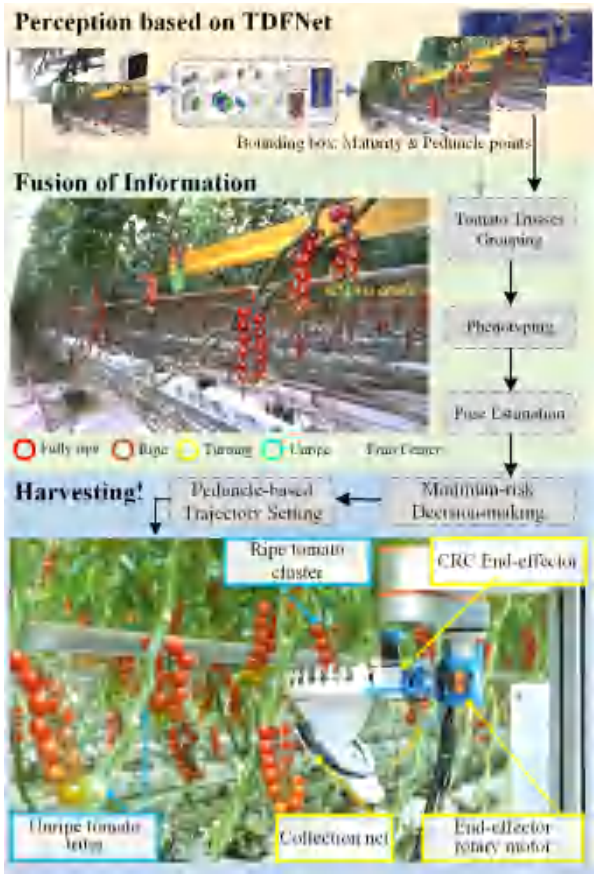


Fig. 1. System overview of the tomato harvesting robot. In the perception stage, a depth camera captures RGB images and depth data from the scene. The TDFNet extracts essential metadata from the RGB images. The fusion stage integrates this information to determine the phenotype parameters and spatial posture of tomato trusses through grouping mechanisms. The harvesting stage employs a decision-making framework to plan a collision-minimizing trajectory and utilizes the proposed circular rotary cutting (CRC) end-effector to accurately sever the peduncle and complete the harvesting process. This integrated system ensures efficient and minimally damage harvesting in real-world environments.

This article builds upon and significantly extends the robotic harvesting research [13] presented at International Conference on Robotics and Automation (ICRA) 2024, introducing the following key enhancements and expansions.

1) TDFNet is proposed to improve model accuracy while maintaining a moderate scale of network parameters, specifically under conditions of lightweight models and limited datasets.
2) A new harvesting method, along with an end-effector design based on target posture and volume, has been refined and detailed.
3) Expanded unit and system experiments in commercial greenhouse environments have been conducted to further validate the practicality and robustness of the proposed methods.

II. Related Work

Manipulation is a fundamental skill for robots to interact with the physical world [14]. Agricultural harvesting tasks impose stricter constraints and more complex motion requirements compared to industrial applications. For tomato clusters, the process

extends beyond simple grasping to include separating the target from the plant without causing damage. In addition, entanglement of the robotic arm with vines remains a critical challenge, as such scenarios are difficult to resolve autonomously, significantly impacting efficiency.

This study focuses on designing a robotic system for the autonomous and continuous harvesting of tomato clusters, addressing key challenges in perception, end-effector design, and motion execution under complex agricultural conditions.

A. Crop Perception

Crop detection and maturity estimation are critical initial steps in robotic harvesting pipelines. Afonso et al. [15] employed the Mask R-CNN network in greenhouse environments to detect tomatoes and classify them as mature or immature, outperforming traditional image processing methods based on manual feature extraction, such as color and contour. However, binary classification of maturity presents challenges due to ambiguous class boundaries. To address computational constraints, Tian et al. [16] proposed TF-YOLOv5s, enabling real-time detection of tomato flowers and fruits using edge computing platforms with minimal resources.

Beyond detection, robotic harvesting requires additional information to support decision-making and motion planning, mimicking human decision processes. This includes phenotypic traits, such as growth status used in precision agriculture, and 3-D pose estimation. Weyler et al. [17] introduced a single-stage model based on CenterNet for instance detection and leaf counting in field crops, while Marks et al. [18] used unmanned aerial vehicles (UAVs) to perform high-precision leaf instance segmentation of plant point clouds for phenotypic analysis. Although these studies did not target tomatoes or harvesting, their insights into phenotypic analysis highlight how robots can evaluate harvestability and quality by assessing traits, such as fruit count and volume.

Grasp pose estimation is another critical aspect, requiring spatial localization of key parts of the target for action planning. Li et al. [19] used a multiview system and a deep convolutional neural network (DCNN) model to detect apple targets, reconstruct occluded regions using visible point clouds, and estimate grasp poses. Tafuro et al. [20] applied segmentation models to extract the point cloud of strawberry targets, estimate their weight, and identify contact points for cutting and grasping. Yin et al. [21] employed Mask R-CNN to segment grape cluster regions and used the random sample consensus (RANSAC) algorithm to fit point clouds into cylindrical shapes, obtaining 6DOF information for harvesting tasks. For tomatoes, detecting, segmenting, and locating cutting points is crucial. Rong et al. [22] used YOLOv4-Tiny to detect tomato clusters and peduncles, followed by YOLACT++ for segmentation and least-squares curve fitting to extract three key points. Zhang et al. [23] proposed a tomato pose model (TPM) integrating prior geometric models, a cascaded multitask network, and 3-D reconstruction to represent relationships between fruits, clusters, and plants through key points. However, TPM imposes strict requirements on the number of fruits in a cluster, reducing its applicability.

In agricultural robotics, data scarcity and class imbalance in datasets present additional challenges. Application requirements, such as semantic segmentation, object detection, and pose estimation, demand diverse datasets. For instance, Tafuro et al. [20], [24] used datasets containing 3100 and 1588 samples for strawberry segmentation and keypoint estimation, respectively. Rong et al. [22] employed datasets with 828 and 700 samples for peduncle detection and segmentation, while Kim et al. [25] used datasets with 443 and 447 samples for cherry tomato detection/segmentation and keypoint estimation, respectively.

Maximizing the utility of limited datasets is critical in agricultural applications. Nesteruk et al. [26] proposed an augmentation framework that generated scene-consistent training samples by placing image masks on random backgrounds. Riou et al. [27] introduced a data-carrying strategy to better convey contextual information during training. While augmentation is a valuable auxiliary tool [28], it often fails to fully address the challenges of data scarcity, particularly in improving model generalization to unseen scenarios. Nuthalapati and Tunga [29] addressed this limitation by combining feature extractors with transformers and employing Mahalanobis distance to classify plants and diseases with limited samples.

The development of deep learning has significantly advanced perception and harvest pose planning, addressing the complexity and uncertainty of these tasks. Previous studies laid the foundation for understanding crops with high structural diversity. However, few studies have combined perception with vision-based grasping to demonstrate how perception methods directly support autonomous harvesting. Furthermore, the reliance on independent datasets for each perception task creates a "bottleneck effect," limiting system functionality, increasing the research burden, and hindering the scalability of agricultural robots.

B. Autonomous Harvesting Systems

In 2017, Bac et al. [30] developed a sweet pepper harvesting robot with two end-effectors and tested its autonomous harvesting performance in both human-supervised and unsupervised greenhouse environments. The report identified several reasons for failures, including perception errors, localization inaccuracies, motion planning failures, end-effector deficiencies, and extreme fruit positions. Although dated, these conclusions remain relevant. With the maturity of commercial robotic arms and depth cameras, localization and motion accuracy have improved. Researchers have shifted their focus to end-effectors and decision-making systems. Sa et al. [31] proposed a harvesting robot using a suction-based end-effector to fix sweet peppers before cutting the peduncle, but challenges, such as complex lighting conditions and leaf occlusions, led to grasping difficulties, cutting failures, and fruit damage. Miao et al. [32] emphasized the need to detect target orientation for tomatoes, as occluded peduncles often resulted in harvesting difficulties. Li et al. [8] designed a three-finger gripper that achieved separation by quickly rotating the target after grasping, reporting a maximum harvesting success rate of 80.46% with failures mainly attributed to the complexity of target shapes.

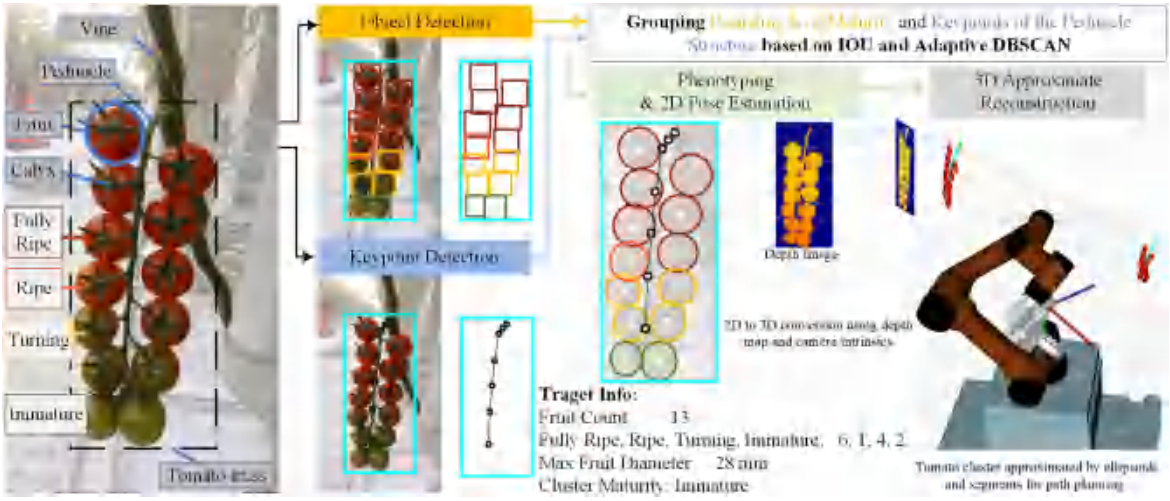


Fig. 2. Perception workflow for tomato phenotyping and pose estimation. The leftmost image illustrates the basic biological structure of a tomato truss, where peduncles extend from the vine and form clusters of fruits (target for harvesting, enclosed by the black dashed box). Object detection identifies tomato clusters and classifies fruits into four maturity categories, each indicated by a color-bounding box. Keypoint detection locates the structural points of the peduncle. Using IoU-based matching in 2-D images and adaptive DBSCAN clustering in point clouds, target information t is obtained. Fruit diameter d (in meters) and 3-D points k_f and k_p are calculated using depth information and camera intrinsics. The cluster is approximated with spheres (fruits) and cylinders (peduncles) to reconstruct its 3-D pose.

End-effectors tailored to crop-specific biological characteristics are critical for improving harvesting success rates. However, the success rate remains a bottleneck limiting large-scale deployment of harvesting robots. In addition, few studies address whether end-effectors damage the plant. For crops, such as tomato clusters, which can be harvested multiple times, damage is unacceptable. Experimental tests often focus on individual objectives without demonstrating the continuous, autonomous harvesting capabilities needed for large-scale applications. Drawing on prior research [13], entanglement with plant vines remains a persistent challenge for robotic harvesting systems, making it difficult to achieve uninterrupted operations without human intervention. This inefficiency restricts scalability and practical deployment.

In summary, this study focuses on three aspects: perception of tomato clusters with limited data, design of crop-specific end-effectors, and development of a harvesting process aimed at achieving autonomous, continuous, and safe operations.

III. Methodology

A. Overview

The core of the proposed autonomous harvesting algorithm is visual understanding, with its main process illustrated in Fig. 2. The perception of potential tomato truss targets T in the environment begins with 2-D detection, progressively extracting bounding box b, ripeness rip, and keypoints Kps as metainformation. Through grouping and matching, the attribute information t^i of each target is generated, encompassing phenotypic traits and 3-D pose. The attribute information of a tomato truss target is summarized as follows:

$$T = (t^i)^{i=1,\dots,m} \tag{1}$$

$$t^i = \{b_t, K_p, K_f, (f^v)^{v=1,\dots,u}, u, \mathrm{rip}_t\} \tag{2}$$

$$K_p = (k_p^1, \dots, k_p^7), K_f = (k_p^1, \dots, k_p^u) \tag{3}$$

$$f^v = \{b_f, \mathrm{rip}_f, d\}. \tag{4}$$

Due to the lack of widely verified detection models capable of outputting an indefinite number of keypoints, the indefinite number of fruit keypoints in the tomato truss is derived from the object detection results. The number of keypoints in the peduncle structure is fixed at 7, which are provided by the keypoint model. The definition and selection of these seven peduncle keypoints are closely related to the picking operation and will be explained in detail in the next section. Commonly used keypoint detection methods are categorized into top–down and one-stage approaches. The top–down method first performs object detection and subsequently estimates keypoints within the detected bounding boxes. A representative model is Lite-HRNet [33]. In contrast, the one-stage approach directly outputs bounding boxes and keypoints. Illustrations of both methods are shown in Fig. 3.

Autonomous operation in robotic systems requires diverse sensory information, which can be efficiently achieved by dividing the perception system into multiple tasks. Each task is supported by a dedicated dataset and neural network model. Due to differences in precision requirements, data collection strategies, and annotation complexity, the dataset sizes for each task are often uneven. This phenomenon is particularly pronounced in agricultural applications.

Taking the two perception tasks in this study as examples: object detection and keypoint estimation. The differences in annotation costs and focuses are evident. Under the same sample size, the annotation cost for pose estimation using keypoints is significantly higher than for object detection. Object detection, as a general-purpose visual task, benefits from advanced methods and tools, such as segment anyting model (SAM)[34], which can perform zero-shot preannotation to reduce labor costs. In contrast, keypoint estimation is often designed for specific tasks, such as human action recognition, gesture detection, or the tomato truss pose estimation

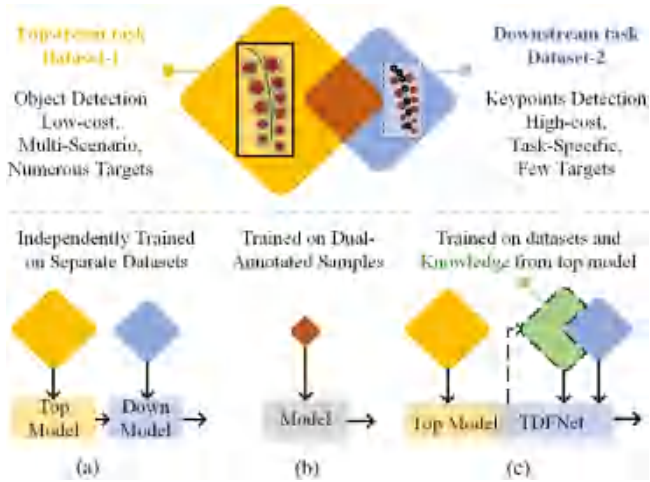


Fig. 3. Illustrations of different pose estimation methods and their use of datasets during training. Generally, different tasks have distinct types of datasets with varying sample sizes, and there may be common samples among them. (a) Top–down approach: Upstream and downstream tasks use separate datasets. (b) One-stage approach: Trains with dual-annotated samples. (c) Proposed TDFNet: Extracts knowledge from upstream, indirectly utilizing upstream datasets to enhance downstream model learning, especially when downstream task data samples are limited—Top–down fused.

proposed in this work, and typically requires full manual annotation.

The focus between these two tasks also differs. Object detection aims to localize targets within scenes, requiring datasets with images taken at various distances, from close to medium and long ranges, to cover all growth stages of tomatoes. On the other hand, keypoint detection emphasizes detailed structural analysis of mature targets ready for harvesting, favoring close-up images in both training and application scenarios. To achieve robust and generalized performance, expanding the training dataset to cover all potential application scenarios of the robot is a critical goal for data-driven methods.

As datasets grow, the divergence between object detection and keypoint estimation datasets becomes more apparent. The proportion of overlapping samples between the two datasets decreases, as each dataset increasingly specializes in its respective task. The relationship between dataset sizes and the overlap ratio of samples is illustrated in Fig. 3. The specifics of the dataset will be elaborated in the subsequent sections.

In this case, the top–down approach is more suitable than the one-stage approach. The reason is that the two task models can be independently trained with their corresponding datasets, making fuller use of the datasets. The one-stage model can only use overlapping samples from different datasets. Even from the opposite perspective, the conclusion remains unchanged. If a one-stage model underperforms in a certain task, increasing the corresponding annotated dataset would also require expanding the dataset of the other task. This is difficult to accept in the field of agricultural robotics, where annotation costs are high and data volume is limited. On the other hand, top–down object detection and keypoint detection can independently replace datasets or models for enhancement. This modular approach prevents a single underperforming task from constraining the overall functionality of the robotic system, thereby improving adaptability and robustness.

B. Top–Down Fusion Network

We found that although the top–down approach uses every sample in the dataset, the utilization efficiency of the dataset can be further optimized. Given an image $\boldsymbol{I}$ containing multiple targets, the goal of top–down pose estimation is to estimate the positions of pose keypoints from each cropped target image $\mathcal{I}$, which can be represented as

$$B = \{b_t^1, b_t^2, \ldots, b_t^m\} = \mathrm{DET}(\boldsymbol{I}) \tag{5}$$

$$\mathcal{I}^i = \mathrm{crop}(\boldsymbol{I}, b_t^i) \tag{6}$$

$$K_p^i = (k_p^1, k_p^2, \ldots, k_p^n) = \mathrm{KPE}(\mathcal{I}^i) \tag{7}$$

where $\mathrm{DET}(\cdot)$ denotes the upstream detection model and $\mathrm{KPE}(\cdot)$ denotes the downstream keypoint estimation model. m and N represent the number of targets in $\mathcal{I}$ and the number of pose keypoints for each target, respectively. b_t^i denotes the ith bounding box of the target and k_p^n denotes the nth peduncle point of the target.

Only detection results B are transferred between tomato detection and keypoint estimation. The keypoint estimation model needs to relearn every image of the targets $\mathcal{I}^i$ independently. This limitation hinders its fitting ability and learning efficiency, thereby reducing accuracy and robustness when applied in natural scenes. Tasks with limited datasets or lightweight architectures face significant challenges due to reduced model capacity and data availability, which can hinder the robustness and accuracy of keypoint estimation models. To address this issue, we aim to improve the performance of keypoint estimation by leveraging the inherent correlation between object detection and pose estimation tasks. Specifically, we hypothesize that there is an overlap in the data domains and features learned between detection and pose estimation models. In particular, we exploit intermediate feature maps $\mathcal{F}$ generated by the detection model, which encode abstract knowledge $\mathcal{K}$—including texture, size, and spatial relationships—as prior information for keypoint prediction. These feature maps provide rich semantic information that bridges the two tasks, serving as a transferable representation to accelerate the fitting process, improve predictive accuracy, and enhance learning efficiency under few-shot conditions. This approach, referred to as top–down fusion $\mathrm{TDF}(\cdot)$, can be formally described as follows:

$$B, \mathcal{K} = \mathrm{DET}(\boldsymbol{I}) \tag{8}$$

$$K_p^i = \mathrm{TDF}(\mathcal{I}^i, \mathcal{K}). \tag{9}$$

To optimize the keypoint estimation process, ground truth heatmaps are generated for each annotated keypoint using the MSRA method [35]. This method encodes each keypoint as a 2-D Gaussian distribution centered at its ground truth location, providing a probabilistic representation to guide the model's predictions. During training, the predicted keypoint heatmaps are supervised using the keypoint mean-squared error loss, which minimizes the pixelwise squared error between the predicted heatmaps and their ground truth counterparts. The loss $\mathcal{L}$

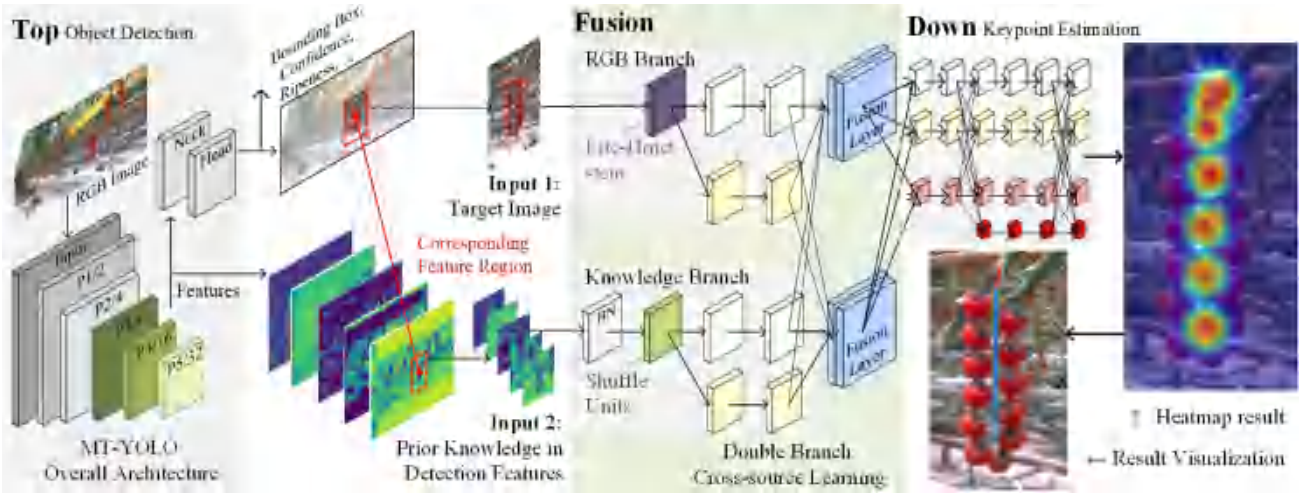


Fig. 4. TDFNet network architecture. The object detection model generates feature maps encoding abstract knowledge (e.g., texture, structural relationships) about the target. These features, along with the detected regions, are mapped and extracted as inputs for the downstream pose estimation model. Through the DBCL framework, target RGB images and features are transformed, learned, and fused to enhance the prediction of structural information.

function is defined as

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N} w_n \left(\hat{H}_n - H_n\right)^2 \tag{10}$$

where N denotes the total number of keypoints, which is $N = 7$ in our research for tomato peduncle keypoint localization. w_n represents the weight assigned to the nth keypoint, $\hat{H}_n$ is the predicted heatmap for the nth keypoint, and H_n is the corresponding ground truth heatmap. The loss function averages the pixelwise mse across all keypoints, weighted by w_n, to optimize keypoint localization accuracy.

Naive TDFNet was built to achieve this goal. To retain the characteristic that the object detection and keypoint estimation components in the top–down approach can be independently enhanced and trained, TDFNet features a dual-layer network architecture. The object detection stage utilizes a multitask YOLOv5 (referred to as MT-YOLO), which includes an additional branch for determining the maturity of the fruit. This model can detect tomato trusses and fruits at four different stages of maturity. Since object detection is not the main focus of this article, it is only briefly introduced here. For the keypoint estimation stage, we use the Lite-HRNet [33] backbone due to its advantage of fewer parameters and faster inference speed.

Convolutional neural networks (CNNs) predict by progressively extracting hierarchical features from RGB images, a process that forms the foundation of their architecture. Lower layers in a CNN typically capture fundamental visual features, such as edges and textures, while higher layers represent more complex and semantic information about objects in the scene [36], [37]. Object detection and keypoint estimation, although distinct in their forms and objectives, both rely on these hierarchical features, including edges, contours, colors, and textures. As shown in Fig. 4, the feature maps generated by the backbone of an object detection model visually highlight the extracted features at various levels of abstraction. The abstract knowledge $\mathcal{K}$ derived from object detection is materialized through these feature representations $\mathcal{F}$. In scenarios with limited model capacity or sparse annotated data, leveraging or augmenting the features generated during object detection can effectively alleviate the learning challenges faced by keypoint estimation models.

We design TDFNet not just to relay detection results between object detection and keypoint estimation but to extract, refine, and integrate detection-derived knowledge as prior information to enhance keypoint estimation. Therefore, we incorporate the detection model's output features $\mathcal{F}$ as a second input to the keypoint estimation model. These features undergo depthwise separable convolutions to adjust the feature map's scale and dimensions and are ultimately concatenated into the first stage output of Lite-HRNet. The goal is for the network to autonomously fit and fuse features from both the RGB image and the detection model.

Preliminary experimental results suggest that this method offers slight improvements over baseline models, although the enhancements are not substantial. This indicates that the current downstream network, with its limited number of parameters, struggles to associate, transform, and deeply fit the two inputs. Therefore, it is necessary to design additional structures to reduce the difficulty of the network in transforming and learning from upstream features, thereby facilitating more effective upstream and downstream fusion. The final testing of the optimized TDFNet is primarily accomplished through the correspondence feature region (CFR) and the double branch cross-source learning (DBCL) structure.

CFR precisely extracts target features during preprocessing to remove irrelevant information that may interfere with downstream tasks. In many cases, the target occupies only a small portion of the scene image, and regions far from the target pixels often do not contain useful information for keypoint estimation. Retaining such irrelevant information in the downstream feature map not only fails to enhance performance but may also introduce noise, reducing prediction accuracy. To mitigate this issue, CFR leverages the bounding box b_t generated by the detection stage as prior positional knowledge to crop and extract the corresponding target regions from the feature map.

Specifically, based on the receptive field calculations of the detection model, CFR determines the dimensions of the bounding box within the feature map and crops this segment to retain only the target features. Common data augmentation techniques, such as scaling and rotation, are also applied within CFR to ensure that the extracted features are consistent with the target regions in the RGB image and maintain the same aspect ratio. The operations of CFR are mathematically described as follows:

$$\mathcal{R}_t = R(b_t) \tag{11}$$

$$\Phi = \mathcal{C}(\mathcal{F}, \mathcal{R}_t) \tag{12}$$

$$\mathcal{I}', \Phi' = \mathrm{ST}(\mathcal{I}, \Phi, \theta) \tag{13}$$

where $R(\cdot)$ denotes the receptive field transformation based on the tomato truss bounding box b_t, and $\mathcal{C}(\mathcal{F}, \mathcal{R}_t)$ denotes the operation of extracting the target feature Φ from the target region $\mathcal{R}_t$ in the feature map $\mathcal{F}$ generated by the detection stage. $\mathrm{ST}(\cdot)$ denotes synchronous transformation $\mathcal{I}, \Phi$ in similar transformation method and parameters θ to maintain the same proportions. For example, if the input of the RGB branch is H_w, H_h, the model feature map is h_w, h_h. If use linear resize, θ are the transformation ratios of the image width and height, denoted as kw, kh. Respectively, the features should also be scaled accordingly before inputted into the DBCL.

DBCL is designed to enable the effective cross-source learning and fusion of two distinct input sources: RGB images $\mathcal{I}'$ and upstream feature maps Φ. While CFR eliminates irrelevant information before input, DBCL focuses on ensuring that features from the RGB branch and the knowledge branch (derived from the detection model) correspond spatially and semantically, allowing them to enhance each other during fusion rather than interfere. The core challenge lies in maintaining the correspondence between the two independent feature maps, ensuring that the learned representations from both sources are compatible and complementary.

To address this, DBCL adopts a dual-branch design in which both branches follow similar convolutional architectures and learning mechanisms, ensuring spatial and semantic consistency in the extracted features. Specifically, both branches are designed to maintain consistent convolutional operations, layer depths, and convolution kernel parameters, ensuring that the intermediate feature representations are spatially aligned. This design minimizes the risk of feature misalignment and facilitates effective pointwise fusion in later stages, all while avoiding a significant increase in the model's parameter count.

Both branches share the commonality of having a stem for preextracting input data and generating high- and low-resolution feature maps for separate learning. The difference lies in that the RGB branch follows the Lite-HRNet stem [33], which has been proven effective. In the knowledge branch, the input is first normalized to minimize the differences caused by the varying data scales between upstream and downstream features. Batch normalization layers are used within the network structure to simplify this operation. Subsequently, shuffle units [38] are used as a stem module to process the input through two distinct convolutional paths. The outputs are then merged and channel shuffling is applied. This method effectively reduces and refines the number of channels of the upstream features. The generated multiscale features are summed pointwise across branches, followed by an additional convolution layer for optimization, ultimately achieving the fusion of upstream and downstream features. This process can be represented as follows:

$$F_{\mathrm{rb}} = \{F_{\mathrm{rb}}^1, F_{\mathrm{rb}}^2\} = \mathcal{B}_{\mathrm{rgb}}(\mathcal{I}') \tag{14}$$

$$F_{\mathrm{kb}} = \{F_{\mathrm{kb}}^1, F_{\mathrm{kb}}^2\} = \mathcal{B}_{\mathrm{kno}}(\Phi') \tag{15}$$

$$F_{\mathrm{dbcl}} = \{F_{\mathrm{dbcl}}^1, F_{\mathrm{dbcl}}^2\} = \left\{ \bigoplus_{s=1}^{2} F_{\mathrm{rb}}^s \oplus F_{\mathrm{kb}}^s, \bigoplus_{s=1}^{2} F_{\mathrm{rb}}^s \oplus F_m^s \right\} \tag{16}$$

where $\mathcal{B}_{\mathrm{rgb}}(\cdot)$ and $\mathcal{B}_{\mathrm{kno}}(\cdot)$ denote the convolutional operation sequences in the RGB and knowledge branches, respectively, generating two-scale feature maps F_{rb} and F_{kb}. The fused feature maps F_{dbcl} are obtained through the cross-scale fusion operation $\bigoplus$, which integrates F_{rb} and F_{kb} at both resolution levels, where $s = 1, 2$ indexes the resolution levels. The operation $\oplus$ represents pointwise summation, while $\bigoplus$ includes upsampling or downsampling to align spatial resolutions before summation.

With the integration of CFR and DBCL, the keypoint estimation (pose estimation) stage effectively utilizes the feature representations generated by the detection stage. CFR ensures that only the relevant target features are extracted from the detection model's output, preventing unnecessary information from interfering with the downstream task. DBCL further enables the alignment and fusion of the RGB image features with the detection features, allowing the pose estimation stage to leverage these two sources of information in a complementary manner. Together, these modules allow the downstream model to better associate, transform, and integrate the knowledge from the detection stage without significantly increasing the number of parameters. Compared to the baseline Lite-HRNet, the network remains lightweight while meeting the computational complexity requirements of robotics perception systems, achieving a balance between efficiency and performance in resource-constrained scenarios.

C. Grouping, Phenotyping, and Pose Estimation

Phenotypic and pose information of target crops is essential for autonomous robotic decision-making and trajectory setting. While TDFNet provides preliminary detections of potential targets, further postprocessing is required to infer complete phenotypic characteristics and pose information of tomato trusses.

1) Tomato Truss Grouping: Tomato truss grouping is the process of associating detected fruits with their respective trusses, a step not directly provided by TDFNet's output. A simple intersection over union (IoU)-based approach is employed in most cases, where fruit bounding boxes are matched with the corresponding truss bounding boxes by applying a threshold. This method is highly effective when fruit bounding boxes are fully contained within truss bounding boxes.

However, challenges arise in cases where multiple trusses overlap due to occlusions or reduced detection accuracy from certain viewpoints. As shown in Fig. 5(a), IoU-based grouping may fail to correctly assign some fruit targets, such as the

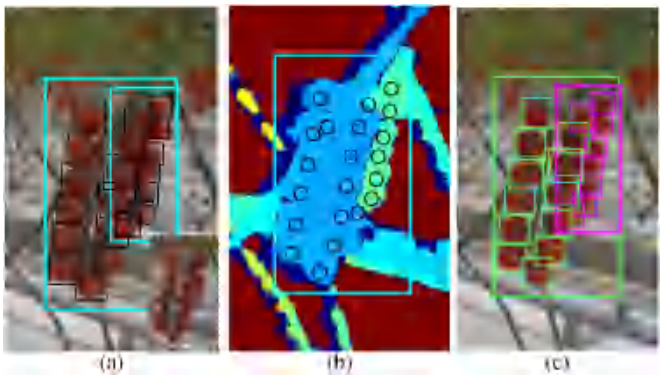


Fig. 5. Challenges in grouping certain fruit targets using IoU-based grouping. The algorithm struggles to resolve the ownership of targets located in the overlapping regions between bounding boxes, as shown by the white target in (a). (b) Visualization of the depth map, where black circles represent the center pixel positions of the fruit targets to be grouped. Fruits belonging to the same tomato truss are expected to exhibit similar depth values. DBSCAN clustering is applied within the bounding box, using each fruit center as a seed point. (c) Grouping results based on the clustering output, with bounding boxes displayed in green and magenta to represent different groups.

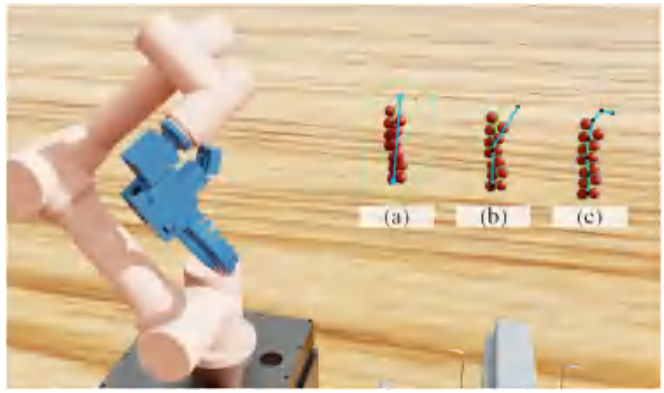


Fig. 6. Comparison of tomato truss posture representations. (a) Traditional methods simplify the truss as a rectangular cuboid or cylindrical model, providing 6DOF pose information but failing to capture irregular shapes. (b) Model with three keypoints approximates the peduncle as piecewise linear segments, losing smooth curvature and prone to generalization issues, such as undetectable keypoints in real-world scenarios. (c) Our proposed method with seven keypoints enables quadratic curve fitting, accurately representing the peduncle's 3-D curvature and supporting precise trajectory planning.

white target in the example. To address this, depth information is utilized to refine the grouping process. Specifically, a density-based clustering algorithm, DBSCAN, is applied within the bounding boxes of detected trusses using fruit centers as clustering seeds. This approach effectively resolves ambiguities by leveraging spatial information, as demonstrated in Fig. 5(b), where overlapping fruits are assigned to distinct groups (green for one group and magenta for another).

To ensure real-time performance, the input point cloud is trimmed to include only regions within detected truss bounding boxes, significantly reducing the computational load of proximity searches. In addition, clustering thresholds are adaptively adjusted based on the prior structural knowledge of tomato trusses. This pipeline ensures accurate grouping even in complex scenarios, providing a robust foundation for subsequent pose estimation and phenotyping tasks.

2) Phenotyping and Pose Estimation: Phenotyping in agricultural robotics involves characterizing biological traits of crops to support decision-making, such as growth assessment or trait selection. For tomato trusses, this includes determining overall maturity, fruit count, and spatial arrangement. Unlike prior studies relying on simple color thresholds or deep learning for maturity estimation, we propose a hybrid approach combining individual fruit maturity and truss-level aggregation to enhance explainability and adaptability.

Pose estimation involves reconstructing the 3-D structure of a tomato truss, including its curved peduncle and spherical fruits (see Figs. 2 and 6). The truss structure is modeled using fruit centers and seven keypoints along the peduncle. Fruit contours are approximated as circles in 2-D images, with diameters calculated as the mean of bounding box dimensions. Depth information is then used to obtain the 3-D center and diameter of each fruit.

Traditional methods often simplify the truss as a rectangular cuboid or cylindrical model, providing only 6DOF pose information. While sufficient for basic grasping tasks, these models fail to represent irregularly shaped trusses, limiting their effectiveness in precision harvesting. To address this, we explicitly model the peduncle's 3-D curvature using keypoints. As shown in Fig. 6, the choice of keypoint quantity significantly impacts representation accuracy. Using only three keypoints results in a piecewise linear approximation of the peduncle, failing to capture its smooth curvature accurately. In real-world scenarios, generalization issues further degrade accuracy due to potential keypoint detection failures. In contrast, our method [see Fig. 6(c)] employs seven keypoints to enable quadratic curve fitting, accurately capturing the peduncle's curvature. This ensures precise pose estimation, robust performance, and reliable trajectory planning for robotic harvesting.

IV. Robot System

A. System Overview

After completing the perception phase, which involves identifying, phenotyping, and estimating the pose of potential crops, the next critical challenge is how the robot makes decisions, plans actions, and executes harvesting based on the perception results. Building upon the analysis of related work and our previous research [13], we identified two major causes of harvesting failures: incorrect harvesting trajectory setting and unintended displacement of the target caused by end-effector contact during motion. Furthermore, attempts to harvest tomato trusses in extreme positions often result in the end-effector becoming entangled with vines, requiring human intervention to disentangle the robotic arm. This significantly limits the robot's ability to perform continuous autonomous operations. In fact, enabling the robotic arm to autonomously escape from entanglement in dense vines proves to be an even greater challenge than the harvesting process itself. Rather than attempting to harvest every target in the environment, we emphasize enabling the robot to achieve multiple successful harvests efficiently. Once the robot demonstrates the capability to replace part of human labor, it could pave the way for large-scale industrial adoption.

Fig. 7. Robots designed and utilized in this research. Left: The initial version [13] with SCARA robotic arm. Right: The extended version with a 6-DOF robotic arm.

We propose that improving the robot's autonomy, efficiency, and operational continuity requires two primary strategies: during the decision-making phase, excluding targets that are unfavorable for the robotic arm to harvest can enhance motion safety and consistency, and during the action planning and execution phases, minimizing contact with the target during cutting can increase the success rate of truss separation while reducing potential risks. To address these challenges, we designed an autonomous harvesting pipeline and corresponding robotic platform, as shown in Fig. 1. This process consists of three key stages: "top–down fusion perception," "fusion of information," and "harvesting."

B. Hardware Design

1) Basic Architecture: Our robotic system comprises five primary hardware components: a depth camera, a robotic arm, an end-effector, a computing unit, and a mobile chassis. In addition to the previously developed AHPPEBOT [13], this study introduces a second robotic system, with the key distinction being the use of different robotic arms, as illustrated in Fig. 7. The earlier version uses a 4-DOF SCARA robotic arm from Huiling-tech Robotic Co., Ltd., while the new version employs a 6-DOF robotic arm from Aubo Robotics and a lifting platform. Since this research does not involve joint-level motion planning or control of the robotic arms, the difference in robotic arm configurations does not affect the autonomous system's perception, decision-making, or path planning processes. Both robotic platforms are used in this study, and all systems and algorithms proposed here are applied to both; therefore, we do not distinguish between them in the subsequent discussions.

The depth camera used is an Intel RealSense D405 stereo camera. Its compact baseline enables it to achieve high-precision depth acquisition (error $\pm$ 1 cm) for fine structures, such as tomato truss peduncles, within a range of 25–75 cm. The computing unit is an industrial PC running Ubuntu 20.04 with an Intel

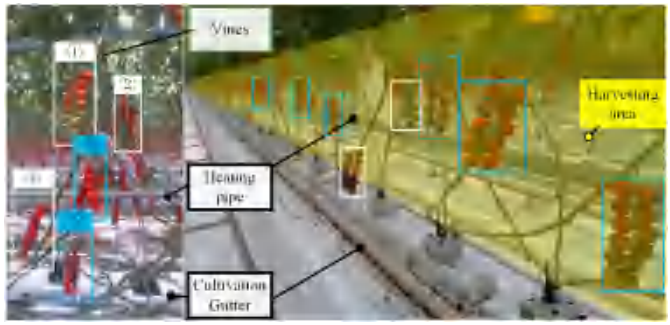


Fig. 8. Illustration of the robot's operational scene and manipulable area. White boxes represent unharvestable targets, while blue boxes indicate suitable targets. Target ① is not ripe, with an unsuitable pose (facing left). Targets ② and ④ are also unharvestable due to their poses (facing away, facing left). Targets ③ and ⑤ meet ripeness and pose requirements, allowing collision-free harvesting based on the robot's heuristic strategy.

i7-10700E CPU and an Nvidia 3050 GPU. The robotic arm's end is equipped with an end-of-arm tool specifically designed for harvesting. This includes an additional rotary motor to provide greater degrees of freedom and a novel CRC end-effector.

2) CRC End-Effector Design: End-effectors are typically crop-specific and are difficult to generalize across different crops [39]. In previous studies on harvesting robots, end-effectors can be broadly categorized into two types: scissor-based end-effectors, as represented in [10] and [40], which achieve separation by cutting the peduncle, and rotation-and-traction-based end-effectors, as represented in [30], which rely on gripping and applying rotational force to break the peduncle.

Scissor-based end-effectors are advantageous due to their simple actuation, typically achieved through motors or pneumatic systems, resulting in short operation times. However, these end-effectors require precise localization of the cutting point and accurate pose alignment, which can be challenging when the peduncle is occluded, the viewpoint is suboptimal, or the cutting point is misaligned. In addition, foreign objects, such as leaves, vines, or structural elements, can obstruct the opening, leading to failures.

Rotation-and-traction-based end-effectors are unsuitable for tomato truss harvesting. Unlike apples, where gripping and pulling may suffice, tomatoes require precise cutting to separate the truss from the plant. Furthermore, these end-effectors are sensitive to the physical characteristics of the fruit, such as moisture content and skin hardness, making it difficult to control grip force, rotational torque, and speed precisely. This often results in damage to the fruit's surface, leading to harvesting failures.

To address these challenges, we propose a CRC end-effector, which offers efficient cutting performance while maintaining high fault tolerance and safety. The structure of the CRC, as shown in Figs. 7 and 9, features a circular outer frame with multiple sharp guiding slots. These slots are precisely sized based on the typical widths of tomato vines and peduncles, ensuring that only peduncles can pass through for cutting while preventing foreign objects, such as vines or debris, from entering. This design effectively resolves the issue of scissor-based end-effectors being obstructed by foreign objects, significantly

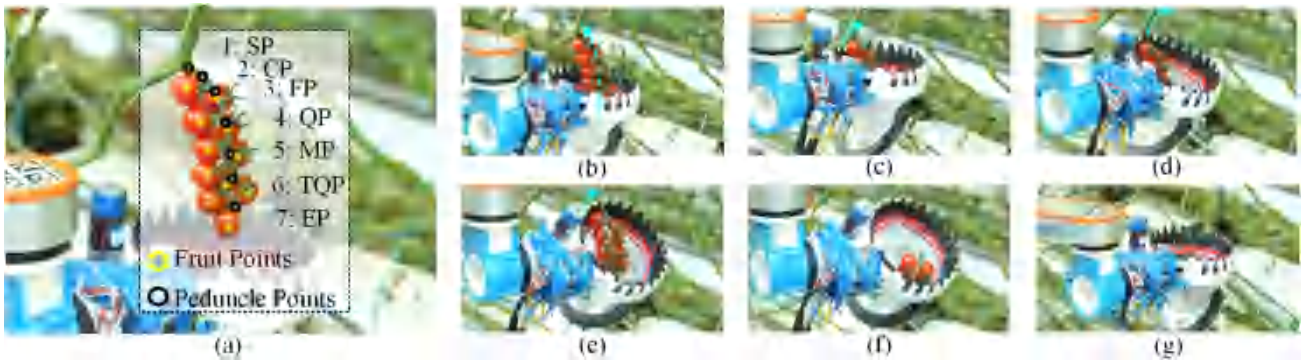


Fig. 9. "Bottom–up wrapping" harvesting process involves the following steps. (a) Robotic arm's end-effector positions itself below the target truss, vertically beneath the end point (EP). (b) End-effector gradually ascends, adjusting its position based on the tomato's volume and key points, including the third quarter point (TQP), midpoint (MP), quarter point (QP), first point (FP), and cut point (CP), to avoid contact. (c) End-effector wraps around the truss, positioning its cutting tool near the start point (SP). (d) and (e) End-effector rotates, guiding the peduncle into the cutting slot. (f) Saw blade cuts through the peduncle, and the tomato cluster is held in the net. (g) Robotic arm follows a predefined lifting path to return to its original position.

enhancing operational safety. The CRC has 17 guiding slots evenly distributed around its frame, enabling it to cut the target peduncle from multiple angles. This reduces the precision requirements for the end-effector's pose adjustment, making it suitable for use with lower DOF robotic arms, such as the SCARA arm. This multiangle cutting capability is particularly advantageous for harvesting in complex planting environments, improving the robot's flexibility and operational efficiency.

Inside the frame, a motor-driven circular saw blade performs the cutting task, separating the peduncle from the plant. The rotation speed of the blade is monitored in real time by an independent encoder and controlled using a PID algorithm to maintain a constant speed under varying loads. This design prevents jamming during cutting, enhancing system reliability. Once the peduncle is cut, the tomato truss naturally falls into a soft net positioned below the end-effector, which cushions the impact and prevents damage to the fruit's surface caused by hard gripping. When the robotic arm moves to the designated unloading position, a servo mechanism opens the net, allowing the tomato truss to slide out under gravity into the collection container. This process eliminates the need for additional mechanical gripping, simplifying the harvesting mechanism and improving overall efficiency.

C. Harvesting Strategy and Trajectory Design

Automated harvesting in greenhouse environments presents unique challenges due to the structural intricacies of tomato vines and the delicate nature of the crops. Potential risks, such as entanglement of the robotic arm with vines, unintended contact with fruits or peduncles, and inaccurate cutting, can disrupt the process, causing crop damage and necessitating manual intervention. These challenges reduce efficiency, increase labor costs, and compromise task success. To address these issues, we propose an integrated risk-minimization strategy that combines heuristic decision-making and trajectory design. This combined strategy ensures operational safety and efficiency by prioritizing low-risk targets and generating trajectories that minimize entanglement and contact.

1) Decision-Making Phase: The decision-making phase focuses on identifying harvesting targets that minimize operational risks while ensuring efficiency. Fig. 8 illustrates a typical greenhouse scenario, where tomato vines grow on trellises with limited operational space. Despite manual adjustments to plant spacing and pruning, tangled vines and constrained environments pose significant challenges for robotic harvesting.

To evaluate and prioritize targets, we employ a heuristic approach that incorporates key spatial and positional characteristics of each truss. This approach rapidly eliminates high-risk targets based on the following factors.

1) *Entanglement risks:* Overlapping trusses or obstructed peduncles increase the likelihood of the robotic arm becoming entangled, potentially causing vine breakage or manual intervention.
2) *Proximity to infrastructure:* Trusses near heating pipes or planting troughs pose risks to both the robot and greenhouse infrastructure.
3) *Target displacement:* Contact with nontarget trusses or fruits during harvesting can cause positional shifts, reducing success rates.

Trusses located in extreme positions, such as those oriented inward toward planting troughs or positioned near infrastructure, are automatically excluded. The relative positions of surrounding trusses are also analyzed to ensure the selected target lies within the safe operational space of the robotic arm. This heuristic evaluation ensures that only low-risk, high-success-rate targets are prioritized for harvesting. Fig. 8 demonstrates how the system evaluates ripeness and pose suitability to identify manipulable targets.

2) Trajectory Design Phase: Once a low-risk target is identified, the trajectory of the robotic arm is planned based on the geometric structure of the peduncle. Specifically, the center of the circular end-effector moves sequentially along the peduncle's curved path. The success of the harvesting process relies on precise trajectory execution, which minimizes the risk of entanglement and avoids contact with the fruit-bearing regions of the truss. Fig. 9 illustrates the sequential steps of the trajectory design and execution process.

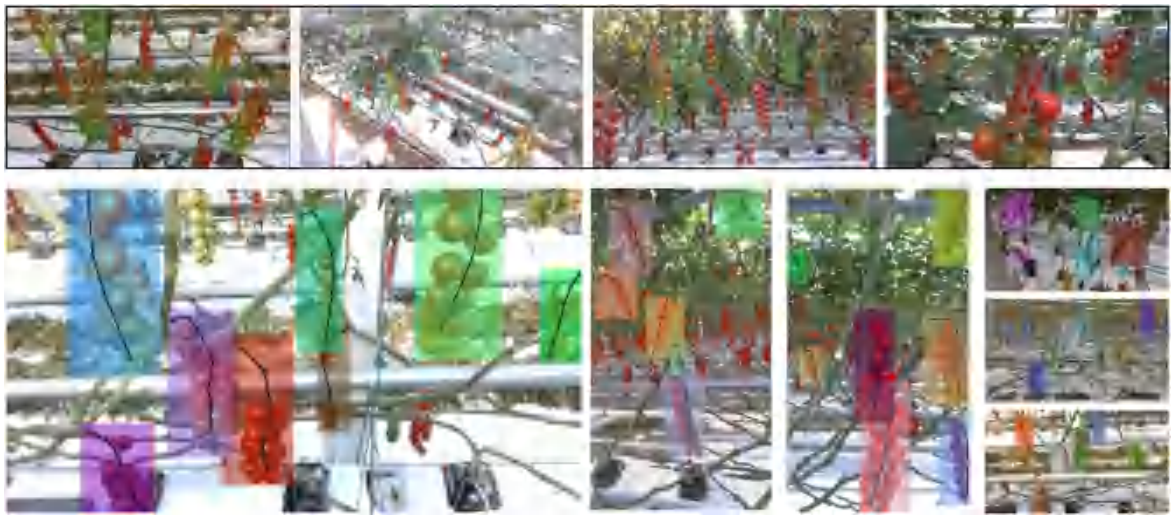


Fig. 10. Sample images and annotations from the *DetectRipen* and *PedunclePos* datasets. *(Up)*: The *DetectRipen* dataset was collected to capture diverse tomato trusses under varying lighting conditions, truss densities, and maturity stages. The camera's yaw, pitch, and roll angles were adjusted to simulate the robot's mobile vision. Four fruit maturity stages were annotated with bounding boxes: green fruit stage (green boxes), where the fruit is fully grown but white-green; breaker stage (yellow boxes), where ripening begins at the apex; turning stage (orange boxes), where over 75% of the surface is red or yellow; and full-ripe stage (red boxes), where the entire surface is red. All visible targets larger than 10×10 pixels were annotated, with bounding box alignment errors limited to within 5 pixels. *(Down)*: The *PedunclePos* dataset focuses on the structural geometry of mature, harvest-ready tomato trusses. Images were captured at close range (within 50 cm) to emphasize peduncle details, with seven structural keypoints annotated (SP, CP, FP, QP, MP, TQP, and EP) to capture peduncle curvature and positional relationships. Each truss is marked with a distinct rectangular region using color-coded boxes, and the black circles represent annotated keypoints. The connections between keypoints approximate the peduncle's curvature and structure.

TABLE I
STATISTICAL INFORMATION OF THE DATASETS

Dataset	Number of images	Number of targets	Annotation details
DetectRipen	2 000	112,000	Bounding boxes, maturity classification
PedunclePos	1051	5 432	Seven peduncle, key points

To represent the peduncle's natural curvature, a quadratic fitting process is applied to the detected keypoints, including seven peduncle points (from SP to EP) and multiple fruit points. This fitting process removes noise and outliers, resulting in a smooth peduncle skeleton that serves as a geometric reference for trajectory planning. The peduncle skeleton is divided into the following two functional segments.

1) *SP–CP segment (cutting region):* This region guides the end-effector to position its cutting tool precisely at the SP, ensuring clean separation of the truss from the vine. Note that the CP, denoting the cut point, is defined as the ideal cutting location for scissor-type end-effectors. However, in the proposed method, the actual cutting process does not occur precisely at the CP but rather at a position between the SP and CP.
2) *CP–EP segment (fruit-bearing region):* This segment represents the main body of the truss and is avoided during motion planning to prevent contact with the fruits.

The trajectory is generated to follow the peduncle skeleton, ensuring a smooth path that adheres to its natural curvature. This minimizes interaction with surrounding trusses and reduces the likelihood of entanglement or fruit displacement. Volume-based adjustments are incorporated to avoid contact with nontarget objects, further enhancing safety and precision.

V. TDFNET EXPERIMENTS AND VALIDATION

A. Dataset Collection and Annotation

Two specialized datasets, *DetectRipen* and *PedunclePos*, were collected from commercial greenhouses in Beijing to train and evaluate TDFNet for tomato truss pose estimation. These datasets were designed for distinct tasks: *DetectRipen* focuses on tomato maturity classification, while *PedunclePos* captures the structural geometry of peduncles in harvest-ready trusses. Annotations for both datasets were performed by experts with agricultural backgrounds and engineers specializing in precision agriculture in greenhouses.

Detailed statistical information, including the number of images, annotated targets, and annotation details, is provided in Table I. The *DetectRipen* dataset simulates the robot's mobile vision under varying environmental conditions, while the *PedunclePos* dataset provides high-resolution peduncle annotations to support pose estimation tasks. Fig. 10 illustrates representative samples and corresponding annotations from both datasets.

B. Model Training

We utilized the MMPose [41] framework and PyTorch for model training and experimentation. Due to the lack of a publicly available tomato cluster keypoint dataset, we performed pretraining on the COCO human pose estimation dataset. During this pretraining phase, the input layer of the knowledge branch in TDFNet was modified to accept RGB images as input. Once

the pretrained model was obtained, training on the PedunclePos dataset commenced.

At this stage, the MT-YOLO model had already been trained on the DetectRipen dataset, equipping it with the ability to detect targets and assess ripeness. To accelerate the training process and eliminate the influence of upstream detection model variations on downstream pose estimation, TDFNet adopted an offline training strategy. During this process, MT-YOLO was excluded from training, with its parameters kept frozen. Instead, it was utilized to perform inference on all images in the PedunclePos dataset, generating feature map files for each sample. These feature maps, along with the corresponding RGB images, were loaded by the DataLoader for training the pose estimation component of TDFNet. During this process, common data augmentation techniques for pose estimation tasks, such as scaling, rotation, and flipping of RGB images, were applied. The same transformations were also applied to the input features from the tomato detection model, which were then processed by the CFR module.

The PedunclePos dataset was divided into a training set with 820 images and a validation set with 231 images. The validation set was reused for testing purposes, serving both as a validation and test set. The batch size was set to 12. The initial learning rate was set to 0.005 and optimized using the Adam algorithm. The learning rate was reduced to 20% of its original value after the 40th and 80th epochs. To reflect the varying importance of different keypoints, specific weights were assigned to each keypoint, with joint_weights w_n set to [3.5, 4.5, 2.0, 0.5, 0.5, 0.5, 0.5]. Data augmentation included a rotation range of $\pm 20^\circ$, scaling between 85% and 150%, and a translation ratio range of $\pm$ 10%. Training was conducted over 100 epochs, with validation performed after each epoch. Only the model with the highest accuracy was retained.

C. Metric

Similar to the COCO human keypoint dataset [21], the accuracy assessment for peduncle keypoints is based on the object keypoint similarity (OKS) metric

$$\mathrm{OKS} = \exp\left(-\frac{d_j^2}{2s^2\sigma_j^2}\right) \tag{17}$$

where d_j represents the Euclidean distance between the predicted and ground truth keypoint, s is the object scale factor, and σ_j is the standard deviation of the jth keypoint.

The parameter σ is calculated by determining the standard deviation of multiple annotations for the same target compared to the expert-annotated ground truth. Unlike human keypoint detection, the importance of the seven keypoints on the peduncle varies in harvesting tasks: the accuracy requirements for the first two keypoints (SP and CP) are the highest. If these keypoints are not detected precisely on the peduncle, the harvesting process is likely to fail. The remaining points are primarily used to fit the curve of the tomato truss, reducing the potential for collisions with the end-effector. To account for these differences, the σ values for SP and CP were manually adjusted to impose stricter accuracy constraints, ensuring that the overall OKS metric better reflects the precision of these critical keypoints. In addition, the OKS threshold was set to 0.75 to align with the high accuracy requirements of the robotic harvester for target pose estimation.

TABLE II
EXP. 1: PERFORMANCE COMPARISON OF MODELS ON THE VALIDATION SET UNDER THE FULL TRAINING DATASET

Model	Input size	Params(M)	GFLOPs	AP (%)
Top–down				
Hrnet-w32	256 × 192	28.54	7.65	71.23
Hrnet-w32	384 × 288	28.54	17.29	74.77
ResNet-50	256 × 192	35.92	6.71	73.44
ResNet-50	384 × 288	35.92	15.09	74.08
MobileNetV2	256 × 192	9.57	1.57	66.56
MobileNetV2	384 × 288	9.57	3.54	69.79
ShuffleNetV2	256 × 192	7.55	1.36	57.63
ShuffleNetV2	384 × 288	7.55	3.06	58.86
Lite-HRNet-18	256 × 192	1.13	0.27	60.61
Lite-HRNet-18	384 × 288	1.13	0.60	70.41
Lite-HRNet-30	256 × 192	1.76	0.42	63.06
Lite-HRNet-30	384 × 288	1.76	0.95	71.77
One-stage				
RTMO-s	640 × 640	9.53	5.10	55.23
RTMO-tiny	416 × 416	6.57	3.38	43.49
YOLOxpose-s	640 × 640	10.72	18.30	58.66
YOLOxpose-tiny	416 × 416	6.04	4.39	44.52
Our method				
TDFNet-18	256 × 192	1.21	0.41	73.74
TDFNet-18	384 × 288	1.21	0.83	76.10
TDFNet-30	256 × 192	1.85	0.58	74.48
TDFNet-30	384 × 288	1.85	1.21	76.71

D. Benchmarking Experiment

1) Experiment 1: First, we trained representative models of common paradigms in the field of pose estimation (top–down and one-stage) using the complete training set of PedunclePos and evaluated their accuracy, parameter count, and GFLOPs. Detailed results are shown in Table II and Fig. 11(a).

Experimental results indicate that models with more parameters and larger input sizes achieve higher accuracy, consistent with neural network scaling laws. Larger models, such as HRNet-w32 [42] and ResNet-50 [36], outperform lightweight models, such as Lite-HRNet [33] and MobileNetV2 [43], in terms of accuracy. However, this improvement comes at the expense of significantly increased computational costs, posing challenges for deployment in resource-constrained environments. For instance, with an input resolution of 384 × 288, the GFLOPs of HRNet-w32 reach as high as 17.29.

Notably, TDFNet demonstrated remarkable advantages by maintaining low parameter and computation costs while achieving performance comparable to or even exceeding standard models with larger parameter counts. For example, with an input resolution of 256 × 192, TDFNet with widths of 18 and 30 only increased GFLOPs by 0.142 and 0.155 compared to Lite-HRNet of the same size, while improving AP by 8.58% and 9.52%, respectively. At a resolution of 384 × 288, the GFLOPs of TDFNet-30 are approximately 7% and 8% of HRNet-w32 and ResNet-50 at the same resolution, yet its AP surpasses both models. This demonstrates that TDFNet achieves an optimal balance between accuracy and computational cost, making it well suited for real-time and resource-constrained scenarios in automated harvesting tasks.

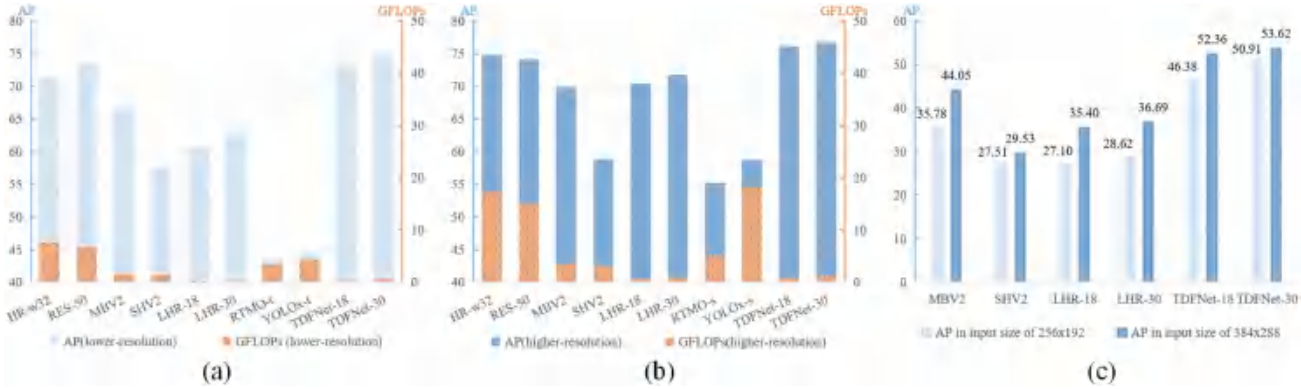


Fig. 11. Comparison of model complexity and accuracy. (a) and (b) Accuracy results and computational complexity of models across different input sizes on the full dataset. (c) AP of models across different input sizes on the few-shot dataset. Top–down methods are evaluated at input resolutions of 256 × 192 (lower resolution) and 384 × 288 (higher resolution). One-stage methods are evaluated at 416 × 416 (lower resolution, "-t(iny)" models) and 640 × 640 (higher resolution, "-s" models). HR-w32 = HRNet-w32, RES-50 = RESNet-50, MBV2 = MobileNetV2, LHR = LiteHRNet, RTMO-s/t = RTMO-s, RTMO-tiny, and YOLOx-s/t = YOLOxpose-s, YOLOxpose-tiny.

It is worth noting that among the tested models, top–down models generally achieve higher accuracy on the validation set compared to one-stage models (RTMO [44] and YOLOx-Pose [45]). One-stage models jointly learn both bounding box detection and keypoint estimation, enabling them to perform detection and output keypoints simultaneously. We believe this performance gap is caused by the specific characteristics of the PedunclePos dataset. Not all objects in the images are labeled, as illustrated in the examples in Fig. 10. The reasons for this dataset inconsistency have been previously discussed in Section III. The varying requirements and costs of different tasks lead to discrepancies in dataset labeling. This inconsistency introduces additional challenges for training one-stage models, especially when they need to simultaneously learn detection and regression tasks. Moreover, the total size of the PedunclePos dataset, even in its complete form, is not large enough to support the training of multitask one-stage models effectively.

2) Experiment 2: To enhance the applicability of the experiments and provide references for further research on harvesting robots, we adjusted the experimental conditions. We trained and evaluated models with potential for deployment in robotic harvesting applications under conditions of limited training samples. During the training phase, only 25% of the samples from the training set were used, while the validation set size remained unchanged. This setup was designed to test the models' generalization abilities under limited data conditions.

In the preliminary experiments, we observed that multiple tomato clusters often appeared simultaneously in the scenes. Models, such as HRNet-w32 and ResNet-50, exhibited insufficient computational speed, which negatively impacted the real-time performance of the robotic perception system. RTMO and YOLOxPose showed poor accuracy in our test greenhouse scenarios and were consequently excluded. Therefore, in Experiment 2, only lightweight top–down methods were evaluated, and inference speeds were tested on a CPU with a batch size of 10. The experimental results are shown in Table III and Fig. 11(b).

The results demonstrate that TDFNet maintains significant advantages under limited sample conditions. For instance, with an input resolution of 256 × 192, TDFNet with widths of 18 and 30 outperformed Lite-HRNet in accuracy by 19.21% and 19.62%, respectively. When the training set size was reduced from 100% to 25%, TDFNet's accuracy dropped by 27.36 percentage points at an input resolution of 256 × 192, whereas Lite-HRNet exhibited a larger decline of 33.51 percentage points. This result partially reflects the robustness of the models when encountering environmental changes, with TDFNet demonstrating stronger adaptability.

TABLE III
EXP. 2: DETECTION ACCURACY AND REAL-TIME PERFORMANCE COMPARISON FOR ROBOTIC APPLICATIONS UNDER LIMITED TRAINING DATA

Model	Input size	AP (%) under limited data condition	FPS*
MobileNetV2	256 × 192	35.78	2.0
MobileNetV2	384 × 288	44.05	1.5
ShuffleNetV2	256 × 192	27.51	3.4
ShuffleNetV2	384 × 288	29.53	2.6
Lite-HRNet-18	256 × 192	27.10	8.7
Lite-HRNet-18	384 × 288	35.40	3.2
Lite-HRNet-30	256 × 192	28.62	5.4
Lite-HRNet-30	384 × 288	36.69	2.2
Our method			
TDFNet-18	256 × 192	46.38	5.6
TDFNet-18	384 × 288	52.36	2.4
TDFNet-30	256 × 192	50.91	3.9
TDFNet-30	384 × 288	53.62	1.5

*Inference time and FPS tested under CPU AMD Ryzen5 3600 × 4.25 GHz, batch size = 10.

E. Ablation Study

To evaluate the effectiveness of the proposed TDFNet architecture, we conducted a comprehensive ablation study by incrementally integrating key components into the baseline model and analyzing their impact on pose estimation performance. The baseline model was Lite-HRNet-18, with an input resolution of 256 × 192. The AP at 0.75 metric was used to quantify model accuracy. We adopted the same two training conditions as in the previous experiment: the complete training set and the limited training set. Experimental results are summarized in Table IV.

First, the knowledge branch was added to the baseline model, while only using RGB images as input. Under both the complete and limited training set conditions, this intermediate model

TABLE IV
ABLATION STUDY RESULTS ON TDFNET USING AP AT 0.75

Variants	Components	Input	AP (%)	
			Full	Limited
Baseline		RGB $\mathcal{I}$	60.61	27.10
	+ KB	RGB $\mathcal{I}$	64.45	30.87
TDFNet	+ KB + CFR	RGB $\mathcal{I}$, DF $\mathcal{F}$	73.74	46.38

KB: Knowledge Branch, CFR: Correspondence Feature Region, DF: Detection Features.

TABLE V
THREEFOLD CROSS-VALIDATION RESULTS OF TDFNET

Fold	Complete dataset AP (%)	Limited dataset AP (%)
Fold 1	71.13	46.10
Fold 2	71.91	45.08
Fold 3	71.17	46.92
Average	71.40	46.03
Std. Dev.	0.32	0.76

achieved improvements of 3.84% and 3.77%, respectively, compared to the baseline. These gains were attributed to the optimized and expanded network structure. Subsequently, the model leveraged features from the tomato detection model and used the CFR module to precisely extract target-relevant regions, enabling upstream-downstream fusion inference. Under the complete and limited training set conditions, this approach achieved improvements of 13.13% and 19.28%, respectively, compared to the baseline. These results indicate the complementary nature of RGB and detection knowledge, and demonstrate that the proposed fusion method significantly enhances accuracy and the model's learning capability.

F. Conclusion on TDFNet

The results of benchmarking and ablation experiments jointly validate the effectiveness of the TDFNet architecture, particularly the advantages of its top–down fusion design. The experimental findings confirmed the initial hypothesis: for datasets with limited sample sizes, models with fewer parameters struggle to efficiently train on raw images and fail to serve as effective feature extractors, while models with more parameters, despite their stronger feature extraction and prediction capabilities, suffer from low real-time performance. The key challenge lies in how downstream models can extract sufficient features to support high-precision predictions. By leveraging knowledge generated from upstream tasks, TDFNet provides a promising solution to this challenge, reducing the difficulty of directly learning from target images.

By utilizing knowledge learned from upstream tasks, TDFNet significantly improves the accuracy of downstream tasks, particularly under limited dataset conditions. The features extracted by the upstream model indirectly leverage multiple potentially related datasets. TDFNet transforms data from various tasks into "knowledge" usable for specific tasks while maintaining the advantages of the top–down paradigm, enabling modularity between upstream and downstream models. In the agricultural domain, where annotated data are scarce and task scenarios are diverse, the ability to independently enhance or replace upstream models or datasets becomes particularly important.

G. k-Fold Cross-Validation Experiment

To further validate the stability and generalization capability of TDFNet, we conducted a threefold cross-validation experiment on the complete training set. Results are given in Table V. The average AP on the complete training set was 71.40%, with a standard deviation of only 0.32, indicating stable and consistent performance across different data partitions. These results demonstrate that the training effectiveness of TDFNet is not dependent on a specific data partitioning, further enhancing the credibility and applicability of the model in agricultural scenarios.

In addition, a supplementary cross-validation experiment was conducted under a limited dataset condition. The results, also presented in Table V, show that the average AP under limited data conditions dropped to 46.03%, with a slightly higher standard deviation of 0.76. Although performance declines with less training data, TDFNet still maintains reasonable detection accuracy. The primary focus of this study, however, remains on the results obtained using the complete dataset, as they better reflect the model's potential for practical applications.

VI. HARVESTING EXPERIMENTS

To validate the effectiveness of the proposed method in perception, path planning, and execution, we designed a series of experiments. These experiments comprehensively analyze and evaluate the impact of perception on action outcomes, single-target harvesting tests under controlled conditions, and autonomous continuous harvesting performance.

A. Validation Experiment on the Impact of Pose Estimation Accuracy on Harvesting Success Rate

The performance of the visual perception module, particularly pose estimation, is typically evaluated independently of the subsequent harvesting operation. However, this isolated evaluation limits the assessment of the perception module's contribution to overall system-level harvesting success. To address this issue, this experiment was designed to validate the relationship between pose estimation accuracy and the harvesting success rate under different precision thresholds.

The experiment was conducted in a high-fidelity simulation environment, NVIDIA Isaac Sim, to ensure reproducibility and strict control of variables. In natural environments, the uniqueness of each target and the changes in pose after a harvesting attempt make it difficult to restore initial conditions for comparative experiments. By contrast, the simulation environment allowed for the digital modeling of 42 tomato clusters, replicating real-world tilt angles and curvatures, as illustrated in Fig. 12. The same model (TDFNet) was used, with different versions saved during the training process, each exhibiting varying pose estimation accuracies on the validation dataset. All other processing steps, including grouping and 3-D pose estimation, remained unchanged to isolate the impact of pose estimation accuracy.

A harvesting attempt was considered successful if the end-effector gradually enveloped the tomato cluster from the bottom,

Fig. 12. Built the basic greenhouse structure in Isaac Sim and conducted picking experiments on virtual tomato clusters replicated from real-world scenarios.

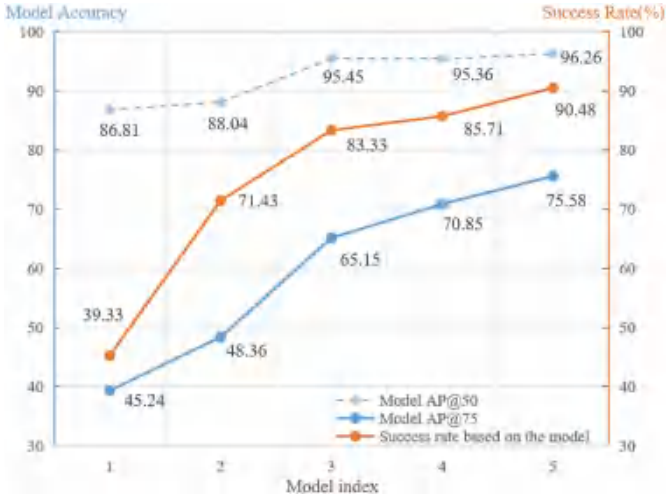


Fig. 13. Relationship between pose estimation accuracy (AP50 and AP75) and harvesting success rate. The results demonstrate strong positive correlations, with $r_{AP50} = 0.9009$ and $r_{AP75} = 0.9438$. AP75 exhibits slightly higher explanatory power, as indicated by $R^2_{AP75} = 0.8907$ compared to $R^2_{AP50} = 0.8116$.

with the fruit stem ultimately aligning with the guiding slot of the end-effector. Failure was defined as any collision that displaced the target by more than 1 cm. To facilitate the detection of collisions, targets were set to a zero-gravity state, making them more responsive to external forces.

To evaluate the contribution of pose estimation accuracy to harvesting success rate, two standard precision thresholds, AP50 and AP75, were analyzed. Pearson correlation coefficients (r) and coefficients of determination (R^2) were calculated to quantify the relationships between these variables. The analysis, shown in Fig. 13, revealed that AP50 and AP75 are both strongly positively correlated with harvesting success rate, with $r_{AP50} = 0.9009$ and $r_{AP75} = 0.9438$. The corresponding coefficients of determination, $R^2_{AP50} = 0.8116$ and $R^2_{AP75} = 0.8907$, indicate that AP75 provides a slightly stronger explanatory power under simulation conditions.

These results suggest that while both thresholds are effective predictors of harvesting success, AP75 offers stricter evaluation criteria, making it more suitable for model selection during training. However, in real-world conditions, where environmental disturbances, such as occlusions, lighting variations, and mechanical noise, are prevalent, AP50 may provide more robust performance by mitigating failures caused by overly strict thresholds. This tradeoff highlights the importance of leveraging different precision thresholds in simulation and real-world applications to balance precision and robustness.

B. Automated Harvesting Experiment

The proposed method was integrated into a robotic system, and two types of experiments were conducted in a greenhouse. The first experiment evaluated the performance of pose estimation using TPFNet, the effectiveness of end-effector path planning based on pose estimation, and the functionality of the end-effector under controlled conditions. The second experiment assessed the system's autonomous capability in an environment without human intervention.

1) Experiment 1—Single-Target Harvesting Test in Controlled Conditions: We intentionally adjusted the environmental conditions to ensure the safety and controllability of the experiment. Specifically, the target tomato peduncle was not occluded by other similar fruits, and no other greenhouse structures or vines, apart from the vine connected to the target, interfered with the process. Each stage of the experiment was manually verified for correctness. The accuracy of pose estimation was evaluated by manually annotating keypoints on the target peduncle in real time by an operator. Unlike the model prediction accuracy tests described in the previous section, we calculated the OKS between the 3-D virtual peduncle curve generated by the perception system and its 2-D projection on the image against the ground truth.

In natural scenes, the accuracy of the perception model is often affected by factors, such as lighting conditions and crop texture, leading to most predictions being considered failures under the default 0.75 threshold, thereby preventing the perception structure from being output. As a result, the threshold had to be lowered to 0.5. Since the success rate of the end-effector's cutting operation is highly dependent on the accuracy of SP point prediction, manual evaluation was further introduced. If the end-effector contacted the target without causing a displacement of more than 1 cm, the contact was considered successful. Each pose target category contained 20 items, with a maximum of three attempts per target.

The experimental results are summarized in Table VI. They demonstrate that the perception method based on TDFNet performs effectively in greenhouse environments. The overall accuracy of pose estimation reached 91.25%, with SP point determination achieving an accuracy of over 86.25%. The results reveal a strong correlation between the orientation of the crops and the difficulty as well as the success rate of harvesting. Targets oriented directly toward the robot were the easiest to harvest, followed by those oriented to the right. Conversely, targets facing away from the robot posed the greatest challenge. The primary factor contributing to reduced keypoint detection accuracy was the occlusion of the peduncle by crop vines, which subsequently affected the motion planning of the robotic arm.

TABLE VI
SUCCESS COUNTS AND RATES IN CONTROLLED SCENARIOS

Direction	Pose Estimation	SP Judgment	Bottom–Up Wrapping	Detach
Based on SP and 6DOF bounding box				
Left	-	19, 95%	8, 40%	3, 15%
Front	-	18, 90%	17, 85%	16, 80%
Right	-	19, 95%	12, 60%	10, 55%
Back	-	14, 70%	4, 20%	1, 5%
All	-	70, 87.5%	41, 51.25%	30, 37.5%
Based on proposed method				
Left	19, 95%	20, 100%	13, 65%	10, 50%
Front	19, 95%	17, 85%	19, 95%	18, 90%
Right	19, 95%	19, 95%	19, 95%	19, 95%
Back	16, 80%	13, 70%	9, 45%	8, 40%
All	73, 91.25%	69, 86.25%	60, 75%	55, 68.75%

Harvesting failures were categorized into three main causes: 1) extreme target poses, 2) inaccuracies in pose estimation, and 3) performance limitations of the end-effector. During the trials, it was observed that the end-effector often made contact with either the target or the surrounding vines, causing displacement. This was primarily due to vine occlusion and interference, compounded by the absence of vine perception capabilities. For instance, the net of the end-effector might lift the target by contacting its bottom, displace the target by touching the fruit's edge, or make unintended contact with crop vines, preventing proper alignment of the peduncle with the cutting blade. However, when harvesting was limited to targets oriented toward the front or right, the success rate of peduncle cutting increased significantly to 92.5%. This observation highlights the importance of focusing future experiments on these two orientations to further optimize performance.

In summary, successful harvesting relies heavily on precise path planning and pose-based adjustments to minimize unintended contact with the target. Using the proposed method, the final separation rate was improved by 31.25%. Notably, the fault-tolerant design of the end-effector played a crucial role: once the target was enveloped, successful harvesting was highly likely. Even when there were slight deviations in SP point predictions, the peduncle could still fall into the guiding slot and be successfully cut.

2) Autonomous Continuous Harvesting Evaluation: This experiment aimed to evaluate the robot's capability for autonomous and continuous harvesting in a commercial greenhouse environment. Unlike previous tests, this experiment required no manual intervention or pruning, with the robot autonomously selecting harvesting targets based on its perception of the environment.

The robot moved at a constant speed along a predefined track, continuously scanning for harvestable targets that met predefined criteria. Upon detecting a target, the robot stopped, re-estimated the phenotype and posture of the target, planned its actions, and attempted the harvesting process. Each target was attempted only once, and the robot proceeded to the next target regardless of success. The experiment was halted only in cases of safety concerns.

A section of the greenhouse planting area was divided into five blocks, with the robot traversing approximately 200 m to complete the harvesting. Human operators recorded the number of harvesting attempts and successful harvests in each block, as summarized in Table VII. The results show an average harvesting success rate of 89.58%.

TABLE VII
RECORD OF CONTINUOUS AUTONOMOUS HARVESTING EXPERIMENTS

Experiment ID	Number of attempts	Successful harvests	Success rate
1	16	14	87.50%
2	22	20	90.90%
3	14	13	92.85%
4	26	23	88.46%
5	18	16	88.89%

The experimental results demonstrate that the proposed autonomous decision-making and harvesting method, integrating phenotype and posture estimation, enables the robot to achieve continuous, safe, and efficient harvesting. The primary failure causes, including perception and mechanical limitations, have been discussed in detail in the dedicated failure case analysis section. Notably, despite occasional positioning errors or operational challenges, no safety incidents occurred during the experiment. This can be attributed to the fault-tolerant design of the end-effector, which ensured secure operation even under challenging conditions.

VII. DISCUSSION

A. System-Level Integration and Failure Analysis

In this study, we adopt a system-level approach to analyze and validate the harvesting robot, integrating perception, planning, and execution as a cohesive framework. This is in contrast to prior research, which often evaluates these components in isolation. While our approach provides a holistic understanding of the system, it also introduces challenges in quantifying the individual contribution of each module to the overall performance of autonomous harvesting.

The proposed harvesting system demonstrated high success rates during experiments; however, specific failures were observed, attributed to perception errors, end-effector limitations, target properties, and environmental obstacles.

Perception errors were a primary source of failure. Occlusion, dense fruit arrangements, and lighting variations frequently caused detection inaccuracies. Furthermore, errors in depth estimation led to misalignments between the reconstructed tomato cluster and its actual position, resulting in unintended collisions between the end-effector and the target. These issues often displaced the target, preventing the end-effector from properly applying pressure on the peduncle, as illustrated in Fig. 14(①–④).

End-effector design limitations also impacted performance. Flexible or thick peduncles introduced additional movement, making precise alignment and cutting more difficult. Large or irregularly shaped clusters often failed to fit into the end-effector, leading to incomplete cuts or displacement. Overripe fruit was particularly prone to detachment or damage during collection, especially when falling into the net pouch.

Fig. 14. Examples of harvesting failures. In ①–④, perception errors caused collisions between the harvesting action and the target, leading to displacement. Although the tomato cluster accidentally fell into the end-effector, the robot failed to complete the cutting operation along the designed path. In ⑤, the end-effector collided with the vine in the scene, posing a potential risk of damage if the action proceeded. In ⑥, the peduncle of the target was too long, and due to its flexibility and additional movement allowance, it could not be severed when pressure was applied. In ⑦, the end-effector grazed the vine, causing target displacement. Consequently, the peduncle did not align with the end-effector blade, leading to a cutting failure.

TABLE VIII
PERFORMANCE COMPARISON OF ROBOT AND HUMAN OPERATORS IN HARVESTING TASKS

Harvesting step	Robot time (s)	Human time (s)
Movement and Decision-Making	Robot <Human	
Harvesting Planning	Robot <Human	
Arm Movement	6	2
Cutting (Separation) Action	6	0.5
Placement and Reset	20	4

Environmental obstacles, such as intertwined leaves and vines, further complicated the harvesting process. Collisions with these elements posed risks of damaging the robot or surrounding plants, occasionally requiring human intervention to ensure safe operation.

These findings highlight the need for further advancements in three key areas: perception algorithms to improve target detection and depth estimation, end-effector design to enhance adaptability and precision, and motion planning to mitigate environmental interference and ensure reliable operation.

B. Efficiency Comparison Between Robot and Human Operators

The proposed autonomous harvesting system supports continuous operation, but its speed and adaptability remain lower than those of human workers. The harvesting process is divided into the following five stages:

1) movement and inspection, where the robot scans for ripe tomato clusters;
2) target identification, where it identifies targets and plans the harvesting operation;
3) arm movement, where the arm positions itself to prepare for cutting;
4) cutting and separation, where the peduncle is severed; and
5) placement and reset, where the harvested fruit is deposited, and the robot prepares for the next cycle.

Table VIII compares the time consumption of robots and humans for each step.

In tasks requiring perception and decision-making, the robot is significantly slower than humans due to limitations in real-time processing speed. While the arm movement speed of the robot can approach human speed, it is intentionally restricted for safety reasons. For cutting, humans using scissors can complete the task almost instantly, whereas the robot requires gradual application of force to ensure proper separation. Furthermore, during placement and reset, the robot moves cautiously to avoid entanglement or damage, resulting in a significant time difference compared to humans, who complete this task effortlessly. In addition, the robot foregoes a large portion of extreme targets in the scene due to strategy constraints, further affecting harvesting completeness.

Despite these limitations, robots possess unique advantages in repetitive tasks and harsh environments. Robots are capable of operating continuously in hot greenhouses without fatigue, unaffected by skill level, weather, or mood. Although human workers currently outperform robots in terms of speed, precision, and adaptability, robots have the potential to complement human labor in the future, especially with advancements in technology and human–robot collaboration models.

C. TDFNet Limitations and Scalability

The limited sample size and high annotation difficulty of crop datasets pose significant challenges, as different crops and tasks require distinct annotation strategies. In this study, completing the harvesting task based on pose estimation required both target detection and the annotation of seven peduncle keypoints, making data collection particularly demanding. To address this, TDFNet leveraged correlations between detection and keypoint tasks, significantly improving keypoint detection performance with limited data.

To evaluate TDFNet's scalability, experiments were conducted on a publicly available grape cluster dataset [46], which includes 2-D bounding box annotations and approximately 500 images with manually annotated keypoints. As shown in Fig. 15, the model can outline the overall trajectory of the target's main

Fig. 15. Illustration of TDFNet applied to grape clusters. The model utilizes keypoint detection to estimate the main stem's trajectory, as demonstrated with limited training data. Increasing the dataset size or applying transfer learning techniques could further enhance detection precision and robustness.

stem, albeit with limited precision. Under an OKS threshold of 0.75, Lite-HRNet-18 achieved a keypoint detection accuracy of 50.5%, while TDFNet-18 achieved 68.4%, demonstrating its superior performance and generalizability for clustered crops with similar structural features.

Nevertheless, the limited size of the annotated dataset restricts the full exploration of TDFNet's capabilities. Future work should focus on expanding datasets, validating performance across diverse crops, and exploring domain adaptation and transfer learning techniques to improve adaptability to different crop morphologies and environmental conditions. In addition, establishing quantitative methods to measure task correlations and determining thresholds for effective knowledge transfer remain key research directions.

D. Future Directions for Improvement

While TDFNet has shown promising results in detecting keypoints for tomato cluster harvesting, its reliance on high-quality annotated datasets limits its scalability, particularly for crops with diverse morphologies or in data-scarce environments. Future research should explore transfer learning and domain adaptation to enhance model adaptability and reduce dependency on extensive annotations.

Beyond TDFNet, the overall system performance must be improved to address challenges identified in this study. Perception errors, end-effector limitations, and environmental obstacles remain key factors contributing to harvesting failures. Enhancing the perception module with multitarget recognition and environmental sensing could enable obstacle detection and intelligent path planning. Similarly, optimizing the end-effector design with modular or reconfigurable mechanisms could improve adaptability to different crop structures, reducing failure rates and increasing efficiency.

The future of robotic harvesting systems lies in achieving human-level performance and reliability while maintaining scalability across diverse agricultural applications. Advancements in dynamic path planning, real-time environmental awareness, and collaborative robotics are crucial for enabling robots to operate effectively in unstructured environments. Addressing these challenges will position robotic harvesting systems as a sustainable and scalable solution to meet the growing demands of global food production.

VIII. Conclusion

In this article, we introduced the TDFNet to address the challenges of phenotyping and pose estimation of tomato trusses in agricultural environments. By leveraging features learned from upstream object detection tasks, TDFNet enhances the accuracy, robustness, and few-shot learning capability of downstream pose estimation tasks while preserving the independence of upstream and downstream models. This approach is particularly advantageous in agricultural scenarios where annotated data are scarce. The results indicate that TDFNet, in comparison to baseline models, achieves significant performance enhancements with only a minor increase in parameters, improving accuracy by up to 11.42% and 22.29% on complete and limited sample datasets, respectively.

Building on the capabilities of TDFNet, we developed a robotic harvesting system that integrates innovative target perception, decision-making, and end-effector path planning methods. Real-world experiments conducted in greenhouse environments demonstrated that our robot, guided by vision-based posture estimation and equipped with a specially designed end-effector, achieved a high success rate of 89.58% in harvesting tasks and showcased the potential for continuous operation.

Acknowledgment

This research was conducted during Xingxu Li's doctoral internship at AIForceTech. The AIForceTech team developed the robotic hardware platform (including the end-effector) and provided agricultural technical guidance. Special thanks to Yunpeng Song, an intern at AIForceTech, for his dedicated contributions to the TDFNet extension experiments.

References

[1] A. Ghobadpour, G. Monsalve, A. Cardenas, and H. Mousazadeh, "Off-road electric vehicles and autonomous robots in agricultural sector: Trends, challenges, and opportunities," *Vehicles*, vol. 4, no. 3, pp. 843–864, 2022.

[2] C. Cheng, J. Fu, H. Su, and L. Ren, "Recent advancements in agriculture robots: Benefits and challenges," *Machines*, vol. 11, no. 1, 2023, Art. no. 48.

[3] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—A survey," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 289–309, Apr. 2014.

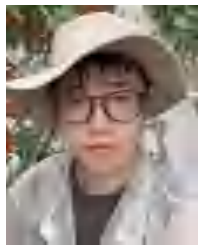
[4] M. Abbas, J. Narayan, and S. K. Dwivedy, "A systematic review on cooperative dual-arm manipulators: Modeling, planning, control, and vision strategies," *Int. J. Intell. Robot. Appl.*, vol. 7, no. 4, pp. 683–707, 2023.

[5] J. Liu and Z. Liu, "The vision-based target recognition, localization, and control for harvesting robots: A review," *Int. J. Precis. Eng. Manuf.*, vol. 25, no. 2, pp. 409–428, 2024.

[6] Z. Wang, Y. Xun, Y. Wang, and Q. Yang, "Review of smart robots for fruit and vegetable picking in agriculture," *Int. J. Agricultural Biol. Eng.*, vol. 15, no. 1, pp. 33–54, 2022.

[7] S. Parsa, B. Debnath, M. A. Khan, and A. G. E, "Modular autonomous strawberry picking robotic system," *J. Field Robot.*, vol. 41, no. 7, pp. 2226–2246, 2024.

[8] T. Li, F. Xie, Z. Zhao, H. Zhao, X. Guo, and Q. Feng, "A multi-arm robot system for efficient apple harvesting: Perception, task plan and control," *Comput. Electron. Agriculture*, vol. 211, 2023, Art. no. 107979.
[9] A. Kumar and L. Behera, "Design, localization, perception, and control for GPS-denied autonomous aerial grasping and harvesting," *IEEE Robot. Autom. Lett.*, vol. 9, no. 4, pp. 3538–3545, Apr. 2024.
[10] J. Jun, J. Kim, J. Seol, J. Kim, and H. I. Son, "Towards an efficient tomato harvesting robot: 3D perception, manipulation, and end-effector," *IEEE Access*, vol. 9, pp. 17631–17640, 2021.
[11] C. Lehnert, A. English, C. McCool, A. W. Tow, and T. Perez, "Autonomous sweet pepper harvesting for protected cropping systems," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 872–879, Apr. 2017.
[12] T. Yi, D. Zhang, L. Luo, and J. Luo, "View planning for grape harvesting based on active vision strategy under occlusion," *IEEE Robot. Autom. Lett.*, vol. 9, no. 3, pp. 2535–2542, Mar. 2024.
[13] X. Li, N. Ma, Y. Han, S. Yang, and S. Zheng, "AHPPEBot: Autonomous robot for tomato harvesting based on phenotyping and pose estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2024, pp. 18150–18156.
[14] R. Newbury et al., "Deep learning approaches to grasp synthesis: A review," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3994–4015, Oct. 2023.
[15] M. Afonso et al., "Tomato fruit detection and counting in greenhouses using deep learning," *Front. Plant Sci.*, vol. 11, 2020, Art. no. 571299.
[16] S. Tian, C. Fang, X. Zheng, and J. Liu, "Lightweight detection method for real-time monitoring tomato growth based on improved YOLOv5s," *IEEE Access*, vol. 12, pp. 29891–29899, 2024.
[17] J. Weyler, A. Milioto, T. Falck, J. Behley, and C. Stachniss, "Joint plant instance detection and leaf count estimation for in-field plant phenotyping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3599–3606, Apr. 2021.
[18] E. Marks et al., "High precision leaf instance segmentation for phenotyping in point clouds obtained under real field conditions," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 4791–4798, Aug. 2023.
[19] T. Li, F. Xie, Q. Feng, and Q. Qiu, "Multi-vision-based localization and pose estimation of occluded apple fruits for harvesting robots," in *Proc. Youth Academic Annu. Conf. Chin. Assoc. Autom.*, 2022, pp. 767–772.
[20] A. Tafuro, A. Adewumi, S. Parsa, G. E. Amir, and B. Debnath, "Strawberry picking point localization ripeness and weight estimation," in *Proc. 2022 Int. Conf. Robot. Autom.*, 2022, pp. 2295–2302.
[21] W. Yin, H. Wen, Z. Ning, J. Ye, Z. Dong, and L. Luo, "Fruit detection and pose estimation for grape cluster–harvesting robot using binocular imagery based on deep neural networks," *Front. Robot. AI*, vol. 8, 2021, Art. no. 626989.
[22] J. Rong, G. Dai, and P. Wang, "A peduncle detection method of tomato for autonomous harvesting," *Complex Intell. Syst.*, vol. 8, no. 4, pp. 2955–2969, 2022.
[23] F. Zhang, J. Gao, H. Zhou, J. Zhang, K. Zou, and T. Yuan, "Three-dimensional pose detection method based on keypoints detection network for tomato bunch," *Comput. Electron. Agriculture*, vol. 195, 2022, Art. no. 106824.
[24] A. Tafuro, B. Debnath, A. M. Zanchettin, and E. A. Ghalamzan, "dPMP-deep probabilistic motion planning: A use case in strawberry picking robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 8675–8681.
[25] T. Kim et al., "2D pose estimation of multiple tomato fruit-bearing systems for robotic harvesting," *Comput. Electron. Agriculture*, vol. 211, 2023, Art. no. 108004.
[26] S. Nesteruk, D. Shadrin, and M. Pukalchik, "Image augmentation for multitask few-shot learning: Agricultural domain use-case," 2021, *arXiv:2102.12295*.
[27] K. Riou, J. Zhu, S. Ling, M. Piquet, V. Truffault, and P. L. Callet, "Few-shot object detection in real life: Case study on auto-harvest," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, 2020, pp. 1–6.
[28] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020.
[29] S. V. Nuthalapati and A. Tunga, "Multi-domain few-shot learning and dataset for agricultural applications," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 1399–1408.
[30] C. W. Bac, J. Hemming, B. A. J. van Tuijl, R. Barth, E. Wais, and E. J. van Henten, "Performance evaluation of a harvesting robot for sweet pepper," *J. Field Robot.*, vol. 34, no. 6, pp. 1123–1139, 2017.
[31] I. Sa et al., "Peduncle detection of sweet pepper for autonomous crop harvesting—Combined color and 3-D information," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 765–772, Apr. 2017.
[32] Z. Miao et al., "Efficient tomato harvesting robot based on image processing and deep learning," *Precis. Agric*, vol. 24, no. 1, pp. 254–287, 2023.
[33] C. Yu et al., "Lite-HRNet: A lightweight high-resolution network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10435–10445.
[34] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 3992–4003.
[35] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 472–487.
[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[37] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2023.
[38] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.
[39] Y. Park, J. Seol, J. Pak, Y. Jo, J. Jun, and H. I. Son, "A novel end-effector for a fruit and vegetable harvesting robot: Mechanism and field experiment," *Precis. Agriculture*, vol. 24, no. 3, pp. 948–970, 2023.
[40] J. Jun, J. Seol, and H. I. Son, "A novel end-effector for tomato harvesting robot: Mechanism and evaluation," in *20th Int. Conf. Control, Autom. Syst.*, 2020, pp. 118–121.
[41] mmpose Contributors, "OpenMMLab pose estimation toolbox and benchmark," 2020.
[42] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.
[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
[44] P. Lu, T. Jiang, Y. Li, X. Li, K. Chen, and W. Yang, "RTMO: Towards high-performance one-stage real-time multi-person pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 1491–1500.
[45] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOx: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[46] A. Blekos et al., "A grape dataset for instance segmentation and maturity estimation," *Agronomy*, vol. 13, no. 8, 2023, Art. no. 1995.



Xingxu Li received the B.Eng. degree in measurement and control technology and instruments from the Institute of Disaster Prevention, Langfang, China, in 2016. He is currently working toward the Ph.D. degree in control science and engineering with the School of Information Science and Technology, Beijing University of Technology, Beijing, China.

His research interests include artificial intelligence, robotics, and autonomous systems, particularly in agricultural applications.

Yiheng Han received the B.Eng. degree in computer science and technology from Jilin University, Changchun, China, in 2018, and the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2023.

He is currently an Assistant Professor with the School of Information Science and Technology, Beijing University of Technology, Beijing. His research interests include visual-point cloud fusion, robotics, deep reinforcement learning, and artificial intelligence in agricultural systems.

Nan Ma (Senior Member, IEEE) received the Ph.D. degree in computer application technolog from the University of Science and Technology Beijing, Beijing, China in 2013.

She is a Professor with the School of Information Science and Technology, Beijing University of Technology, Beijing. Her research interests include machine vision, interactive cognition, and multimedia content analysis. She has authored or coauthored more than 50 SCI/EI-indexed papers.

Dr. Ma is currently a Senior Member of CCF and CAAI. She is also a Member of the Editorial Board of the "*Journal of Intelligent Systems*."

Yongjin Liu (Senior Member, IEEE) received the B.Eng. degree in mechatronics from Tianjin University, Tianjin, China in 1998, and the M.Eng. and Ph.D. degrees in mechanical engineering from the Hong Kong University of Science and Technology, Hong Kong, China, in 2000 and 2004. He has been with the Department of Computer Science and Technology, Tsinghua University, Beijing, China, since 2006, where he is currently a Professor.

In the past five years, he has authored or coauthored nearly 100 papers in leading journals and conferences, with six recognized as ESI Highly Cited or Hot Papers. He has also transferred six patented technologies, valued at 11 million RMB. His research interests include computational geometry, computer vision, and pattern recognition.

Jia Pan (Senior Member, IEEE) received the B.S. degree in automation from Tsinghua University, Beijing, China, in 2005, and the Ph.D. degree in computer science from the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, in 2013.

He conducted postdoctoral research on robotics and machine learning with the University of California, Berkeley, CA, USA. He is currently a tenured Associate Professor with the Department of Computer Science, University of Hong Kong, Hong Kong. His research focuses on on creating algorithms that enable robots to interact efficiently and intelligently with the world and collaborate with people. These general-purpose sensing, control, planning, and manipulation algorithms are applicable to robots operating in homes, factories, laboratories, and agricultural fields.

Shun Yang received the Ph.D. degree in Vehicle Engineering from Jilin University, Changchun, China, in 2019.

He is the CTO of AiforceTech, Beijing, China, where he leads the development of autonomous transport robots, agricultural harvesting robots, and laser weeding robots. His research interests include robotics, autonomous driving perception, and reinforcement learning in the context of smart agriculture.

Siyi Zheng received the Ph.D. degree in Engineering from the University of Science and Technology Beijing, Beijing, China, in January 2013.

She is a Partner and the Director of Administration & Intellectual Property with Beijing AiforceTech Technology Company Ltd., where she has contributed to multiple national research projects, including China's National Key R&D Program and collaborations with the Chinese Academy of Sciences. She owns 40+ patents and authored or coauthored 30+ papers. Her research interests include AI and wheeled robotics, particularly autonomous driving in smart agriculture.

Dr. Zheng was the recipient of the two provincial/ministerial sci-tech awards.