Computers and Electronics in Agriculture 210 (2023) 107878

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

TPMv2: An end-to-end tomato pose method based on 3D key points detection

Fan Zhang [1], Jin Gao [1], Chaoyu Song, Hang Zhou, Kunlin Zou, Jinyi Xie, Ting Yuan, Junxiong Zhang [*]

College of Engineering, China Agricultural University, Beijing 100083, PR China

ARTICLE INFO

Keywords:
Harvesting robot
Deep learning
Tomato bunch
3D pose estimation
3D keypoints detection

ABSTRACT

The automatic harvesting of tomatoes has been achieved for many years in the laboratory. The new research topic is harvesting the tomato more flexibly and nondestructively at any tomato bunch pose according to the agronomic demands. Although the tomato pose can be predicted by keypoints detection, the poor data quality of commercial RGBD cameras, occlusion between plant organs, various tomato poses, and unstructured working environments pose some challenges to the tomato bunch pose detection. Therefore, our research proposed an improved version of the Tomato Pose Method (TPM), namely TPMv2, which is a two-stage end-to-end multi-task network. This network provides comprehensive information on the tomato bunch, including the positions and poses of the stem, peduncle, and fruits, by predicting the two-dimensional bounding box (2D BBox), three-dimensional bounding box (3D BBox), two-dimensional key point (2D Kpt), and three-dimensional key point (3D Kpt). Aiming at the problems of occlusion and poor-quality point cloud, this paper specially designs a key point network (KPN) for tomatoes, where a keypoints processing pipeline was innovatively proposed, improving the accuracy of key point positioning and reducing abnormal prediction effectively. TPMv2 makes it possible to detect tomato bunch pose precisely with an economical camera, avoiding dangerous situations caused by abnormal prediction. The precision of 2D BBox and 3D BBox reached 0.9372 and 0.8700, and the Percentage of correct Keypoints (PCK) of 2D Kpt and 3D Kpt reached 0.8882 and 0.7836. About 78.36 % of 3D Kpts' positioning errors are less than 20 mm, sufficient to describe a correct pose trend based on the 3D Kpt, benefiting the manipulator to plan a more reasonable trajectory for non-destructive harvesting.

1. Introduction

Robots are able to handle more delicate work smartly, while computer vision is boosted by artificial intelligence. Harvesting robots, the main form of agricultural robots, obtain the ability to recognize fruits in the natural environment, where light condition varies and plants grow densely. Sweet pepper, tomato, strawberry, and apple are the most common research objects of agricultural robots. The SWEEPER (Arad et al., 2020) is a successful sweet pepper harvesting robot, which reaches a success rate of 61 %, and 24 s per picking cycle. Its detection algorithm, end-effector (Eizicovits et al., 2016), and path-planning (Bac et al., 2017) were detailed researched. Zhao et al. (2016) designed a tomato harvesting robot with an Adaboost classifier (Zhao et al., 2016) as the tomato detection algorithm and a dual-arm manipulator to pick tomatoes cooperatively. Liu et al. (2020) detect tomatoes with a circle bounding box based on the YOLOv3 network. The strawberry harvesting robot (Yuanyue et al., 2020) localizes the strawberries with a center deviation of 8.9 mm by point cloud construction, segmentation, and shape completion. However, the detection algorithms of the above-mentioned sweet pepper, tomato, and strawberry harvesting robot are only capable of the single fruit and less suitable for fruits clustered in a bunch.

The research object of this paper is tomato bunches containing four to six fruits in one bunch, which are harvested in a bunch. Many factors are influenced the tomato bunch pose, including the deformation of the peduncle, the gravity of fruits, and the interaction forces between fruits, stem, and peduncle. As a result, tomato bunches exist in various poses, such as four special poses, the left, right, front, and back pose. For tomato harvesting in a bunch, identifying a cutting point in the peduncle and cutting it simply is the main-stream paradigm. In light of this,

* Corresponding author.
E-mail address: cau2007@cau.edu.cn (J. Zhang).
[1] These authors contributed equally to this work and should be considered co-first authors.

https://doi.org/10.1016/j.compag.2023.107878
Received 19 November 2022; Received in revised form 13 April 2023; Accepted 21 April 2023
Available online 12 May 2023

peduncles and stems become important organs for tomato bunch pose detection. The harvesting robot (Qin et al., 2021) detects the tomato bunch's bounding box (BBox) by Yolov4 and then identifies the cutting points on the peduncle by their depth and morphology information. However, only a location of one point is not sufficient for a delicate grasping action, especially for agricultural grasping task, which is challenging in a natural environment. Because the variety of tomato bunch poses and the plants interlacing causes the occlusion, these cutting points are often invisible. Incomplete information about the tomato bunch and the surroundings causes a collision between the plant and the robot. For this challenge, the poses information can be a good complementary of tomato localization information to improve the success rate of harvesting robots. The pose estimation is an essential step in many fields highly related to picking (He et al., 2020), especially for these complicated working environments. Haan et al., 2021 segments the whole peduncle of the tomato bunch to find the center of mass as the target grasping location. Kim et al. (2022) proposed a deep learning algorithm Deep-ToMaTo to estimate the 6D pose of the single fruit and its sub-peduncle (side-stem) as an entirety.

The above-mentioned research presents some reasonable methods to estimate the pose of the tomato and its peduncle, benefiting the estimation of robot grasping. There are still two significant but unsolved problems: (1) the peduncle and sub-peduncle is not always existence as a straight line; (2) the peduncle is not always visible at different pose. These problems hinder the tomato bunch pose estimation and successively hinder grasp pose planning for harvesting robots. Our previous research (Zhang et al., 2022) modeled tomato bunch as an eleven-keypoints model. In them, three key points are designed to describe the pose of the peduncle. Three key points can define the peduncle's more flexible status instead of roughly defining the curve peduncle as a straight line or representing the peduncle with a single cutting point. In addition, it can estimate the pose of tomato bunch via detecting the keypoints precisely in image, and obtain their spatial locations with point cloud. Unfortunately, the coordinates of the keypoint are unable to be localized when this point is invisible in the camera's projective plane. The RealSense is one of the common commercial RGBD Cameras, which has many advantages, lightweight, high integration, good real-time performance, and inexpensive. But the main weakness is that the quality of the point cloud generated by this camera is not good enough, where null and infinite values often come. It brought severe negative impacts, leading to the false key points positioning and false tomato bunch pose estimation. The incorrect positioning can guide the robot arm to move in the wrong trajectory, which causes a collision with the tomato plants and damages the stem or fruit.

Although the greenhouse is built and managed by standard, the severely occluded, clustering, multi-pose scenarios are very common here. It brings the dilemma to delicate processing and non-destructive harvesting, which is required precise pose estimation. Therefore, it is necessary and valuable to research further the precise tomato bunch pose and peduncle pose estimation when the organs of the tomato bunch are partially visible, and the quality of the point cloud is limited.

In summary, computer vision and deep learning were applied in harvesting robots widely, including tomato harvesting robots. Although a few research has concerned with the pose estimation of tomato bunch, the pose estimation of flexible tomato bunch, whose peduncle is invisible and curved, is still challenging. To solve these problems, this paper made three main contributions: (1) proposed an end-to-end network TPMv2 (Tomato Pose Method version 2) to estimate the two-dimensional, three-dimensional bounding box of tomato bunch, two-dimensional, three-dimensional coordinates of eleven key points, even if a few keypoints are invisible; (2) proposed a Keypoint Prediction Network, containing DP + DC + KCU processing pipeline, for our end-to-end and multi-task deep learning network, which benefits the partially visible keypoints detection; (3) simulated tomato bunch pose based on the key points prediction, flexibly and correctly described the pose of partially visible tomato bunch and curved peduncle. This paper is structured as follows: Section 2 building datasets, explaining the structure of our TPMv2 and the experimental setup; Section 3 shows the results of three experiments, evaluating the performance of TPMv2 and each processing unit statistically and visually, discussing it qualitatively and quantitatively; finally, the conclusions are presented in Section 4.

2. Materials and methods

List of abbreviation

Abbreviation	Definition	Description
2D BBox	Two-dimensional Bounding Box	A 2D BBox containing one tomato bunch image
3D BBox	Three-dimensional Bounding Box	A 3D BBox containing one tomato bunch point cloud
2D Kpt	Two-dimensional Keypoint	A coordinate (u_P, v_P) in image
3D Kpt	Three-dimensional Keypoint	A coordinate (x_P, y_P, z_P) in camera coordinate system
RPN	Region Proposal Network	A sub network for candidate BBox prediction
KPN	Keypoints Prediction Network	A sub network for keypoints prediction
HG	Hourglass	A net structure in backbone
ZCF Unit	Zero Center Fusion Unit	A feature processing unit for zero centre transform and fusion
KFP	Keypoint-focused Processing	A special feature processing unit
KCU	Keypoints Concatenate Unit	A post processing unit for keypoints prediction
DP	Deprojection processing	A data processing unit for 3D Kpt calculation from 2D Kpt
DC	Decode processing	A data processing unit for 3D Kpt prediction from heatmap
3DBH	3D BBox Delta prediction head	A prediction head for refinement of 3DBBox prediction
PCK	Percentage of Correct Keypoints	A metric to evaluate the quality of predicted Kpt
GT	Ground Truth	–
PD	Prediction	–
GCS	Global Coordinate System	–
LCS	Local Coordinate System	–

2.1. Data acquisition

The dataset for this research consists of 2000 RGBD images captured with camera Realsense D435i (Intel, Santa Clara, America) at a resolution of 640 × 480 pixels. The location of the greenhouse is Hongfu Agricultural Tomato Production Park in Daxing District (116.283601 E, 39.603538 N), Beijing, China. The tomato of the variety "HongZhenZhu" grows in clusters. Each cluster contains six fruits, but 1–2 fruits are often absent, and only 4–6 fruits are visible, stem and peduncle partially visible at different tomato bunch poses. This greenhouse follows the Dutch standard planting and management mode, where the plants are planted uniformly and regularly manually defoliated. The camera is placed on a tripod 400–600 mm horizontal away from the fruits Fig. 1 and 200–400 mm vertical away from the platform. To ensure the diversity of pictures, the camera position is uniformly distributed within the range mentioned above, and the picture acquisition time covers morning, noon, and afternoon.

The image quality can be relatively good at this distance, but there is still a problem the point cloud is too sparse, and the quality of the point cloud at some patches is poor. These extremely poor-quality images need to be filtered out, such as serious defects of depth value, severely occluded by obstacles, and no object in the visual field. One color point cloud can be generated for one RGBD image through the pinhole camera model and the camera inter parameter Fig. 1(e).

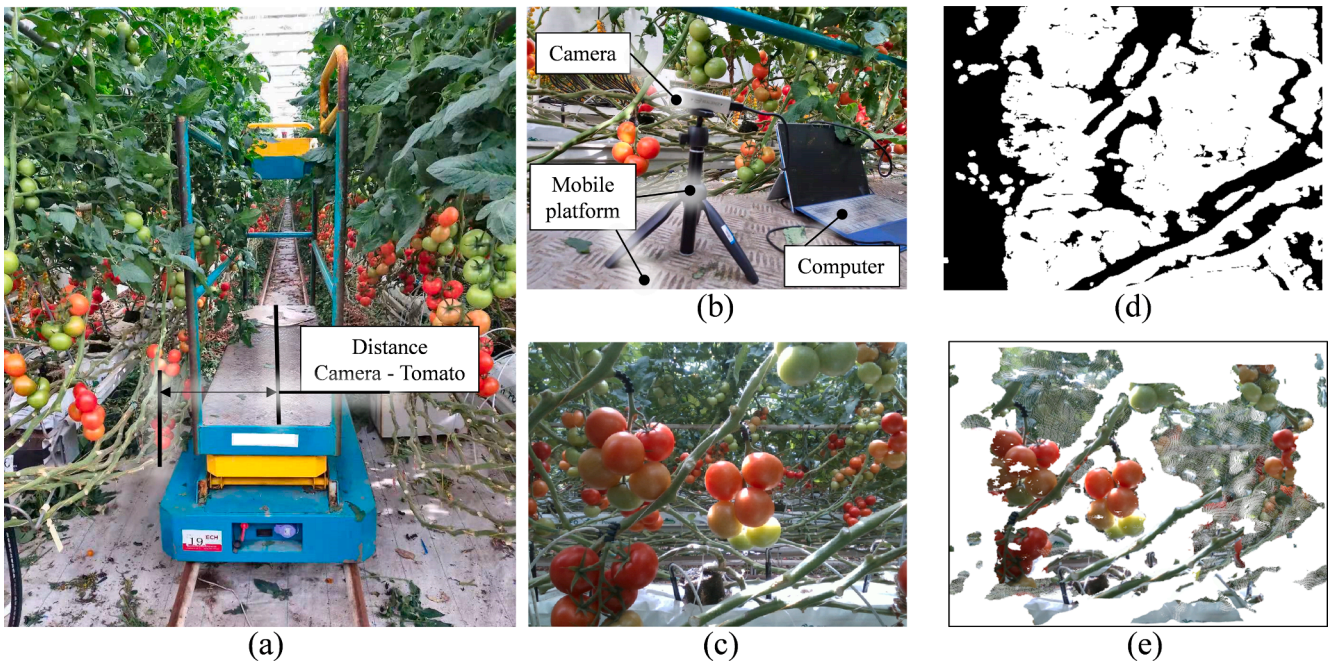

Fig. 1. The device to capture pictures. (a) The tomato bunch in the greenhouse. (b) The capture system consists of a camera, tripod, and computer. (c) The RGB image. (d) The depth image. (e) The point cloud.

2.2. Dataset preparation

Data preparation is an important part of a deep learning task. The data preparation consists of problem modeling and label processing. Some example data, labels and one visualization software can be downloaded in the link https://github.com/123sandrachen/TPMv2.

2.2.1. Problem modelling

Problem modeling is the basis of scientific research, which is the same for deep learning. Tomato bunch, a natural object, is almost impossible to grow following human's desire thoroughly. Inspired by the human pose detection method, an eleven-keypoints model Fig. 2(a) is designed for the tomato bunch, which can meet the challenge of natural object modeling.

The eleven key points are selected according to the agronomical demands, the detailed reasons refer to our previous research (Zhang et al., 2022). The sub-peduncle are not defined as any key points because they are often invisible and too thin to get the depth information by the RGBD camera. Combining the name of the plant organ and the symbol of key points, these eleven key points were named Kpt S1, S2, P3, P4, P5, OA, OB, OC, OD, OE, and OF in the text, where "S" means stem, "P" means peduncle, and "O" means circle fruit.

2.2.2. Label processing

In deep learning, the label content is determined by the modeling of the target. And the label is always used as the ground truth (GT) in the

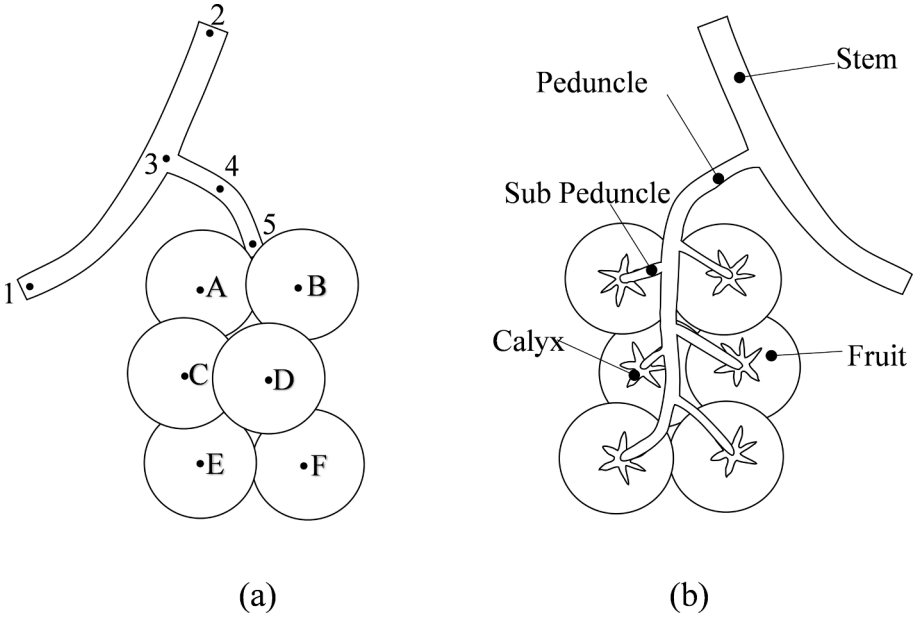

Fig. 2. The eleven-keypoints model of tomato bunch and organ names. Points 1 to 5 are to describe the stem and peduncle. Points A to F are to describe six fruits. (a) Front view of tomato bunch. (b) Back view of tomato bunch.

self-learning algorithms. For our multi-task networks, the two-dimensional bounding box (2D BBox), three-dimensional bounding box (3D BBox), two-dimensional key point (2D Kpt), and three-dimensional key point (3D Kpt) need to be labeled. They were labeled by our own label software, which can label 2D Kpts and 3D Kpts simultaneously. This software was programmed in python language with the pyQt, Vtk, and PCL packages. The 2D BBox is required to be a rectangle enclosing all 2D Kpts of one tomato bunch, while the 3D BBox is required to be a cuboid enclosing all 3D Kpts of one tomato bunch. Because the RGBD image is only from one observation angle, the RGBD image shows partial information about the tomato bunch. It needs to drag and observe the point cloud in the software firstly, combine human reasoning secondly, draw or input the location of each point thirdly, and finally check the registration between the label and truth point cloud. In this process, the human prior knowledge about the structure of tomatoes was inducted into the labeling data serving to network learning.

Four types of labels, namely 2D BBox, 3D BBox, 2D Kpt, and 3D Kpt, are shown in Fig. 3. Each type of label must be defined as a set of parameters. The 2D Kpt is defined by (u_P, v_P) for each point, and the subscript "P" is for the name of the point, such as "P1". The 2D BBox is defined by (cu_B, cv_B, uw_B, vh_B), and (cu_B, cv_B) as the center and (uw_B, vh_B) as the width and height of 2D Bbox. The 3D Kpt is defined by (x_P, y_P, z_P), and the subscript "P" is for the name of the point. The 3D BBox is defined by $(cx_B, cy_B, cz_B, w_B, h_B, l_B)$, and (cx_B, cy_B, cz_B) as the center and (w_B, h_B, l_B) as the width, height and length of 3D BBox. These label data are stored in JSON files.

2.3. The structure of network

Our TPMv2 is an end-to-end multi-task network, serving to predict 2D BBox, 3D BBox, 2D Kpts, and 3D kpts for tomato bunch. Taking Faster RCNN (Ren et al., 2017), Mask RCNN (He et al., 2017) as reference, the TPMv2 consist with two main parts: Region Proposal Network (RPN) and Keypoints Prediction Network (KPN). The RPN generates candidate object boxes, which are used as the Region of Interest, and the features in this region are extracted for the next stage, KPN stage. The overview of the network structure is illustrated in Fig. 4.

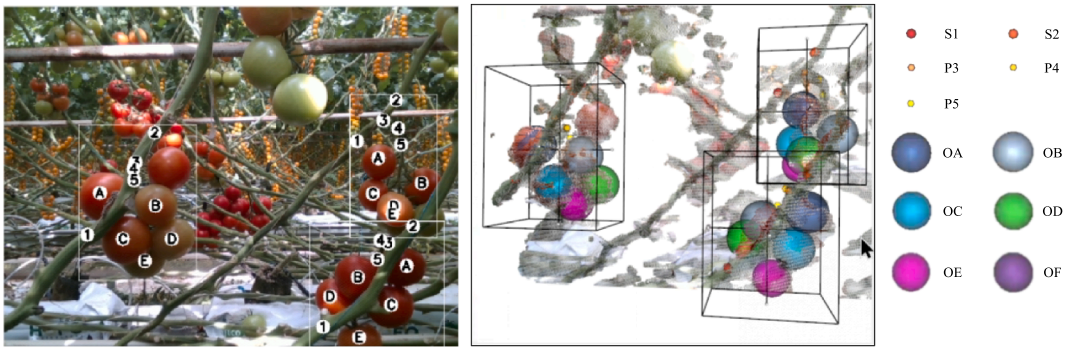

Fig. 3. Four types of label for tomato bunch pose. (a) Labels of 2D BBox and 2D Kpts. (b) Labels of 3D BBox and 3D Kpts.

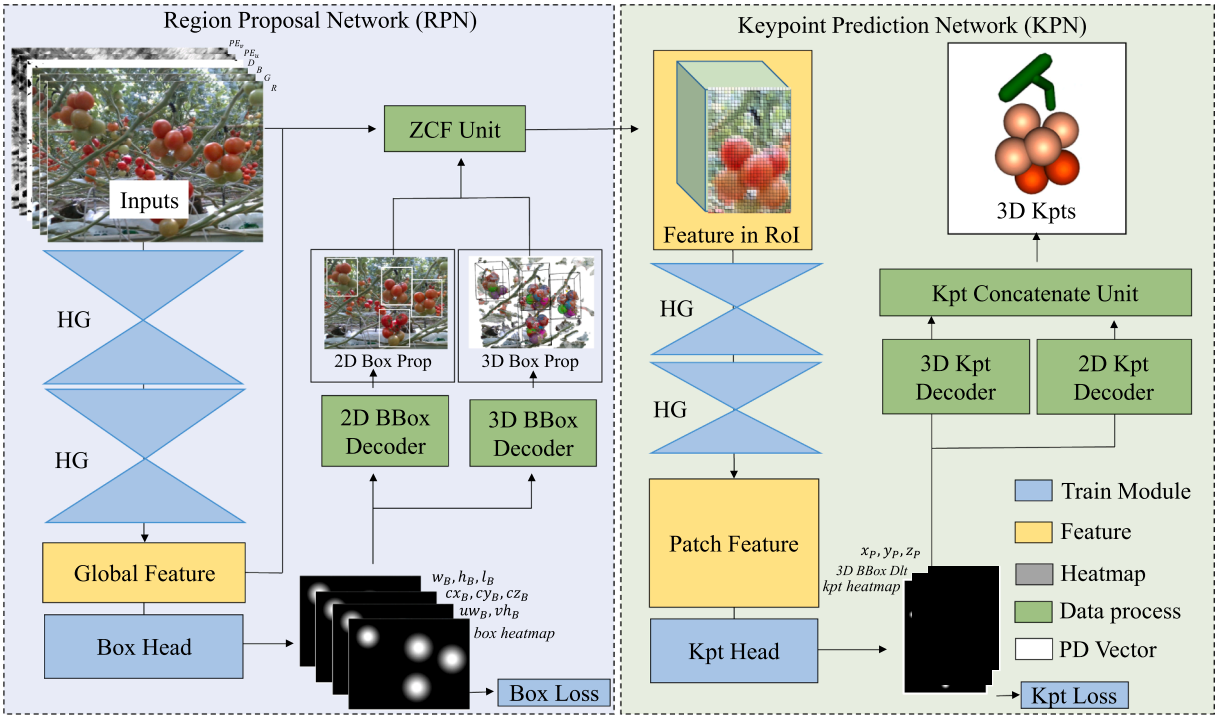

Fig. 4. The overview of our network TPMv2.It consists of two parts: RPN and KPN. HG means the hourglass structure. 2D/3D BBox Prop means 2D/3D box proposal. "2D/3D Box/Kpt Deco" and "2D/3D Kpt" means 2D/3D Bounding box/keypoints decoder.

2.3.1. The region proposal network (RPN)

The main task of RPN is to generate candidate regions, where one tomato bunch is enclosed in one box, and crop and fuse the feature map in this region. While the RPN of the Faster RCNN and the Mask RCNN proposes candidate object BBox, our RPN generate both candidate 2D BBox and candidate 3D BBox. This network includes a preprocessing module, a feature extractor, a Box Head, two Decoders, and one ZCF Unit, taking five-channel image data as inputs and a 262-channel feature map as outputs.

The Inputs: The inputs in Fig. 4 consists of five channels: R, G, B, D, PE_u, and PE_v channel. The R, G, B, and D channels provide color and depth information, respectively. Inspired by the ViT (Dosovitskiy et al., 2021), the position-embedding in our network is D, PE_u, and PE_v channel, which can bring the three-dimensional spatial information into network training. The value of $PE_u = u_p - u_0$ and $PE_v = v_p - v_0$ are calculated by the index of each pixel and the intern parameters of the camera. Where u_P and v_P are each pixel's column index and row index, and the (u_0, v_0) is the projection of optical centers into the image.

The inputs were padded from the shape of 640 × 480 into the shape of 640 × 640 and then were resized into the shape of 256 × 256. This processing is to keep consistency with the shape of the input interface of the feature extractor. Because the padding processing causes the offset of the location of key points, the value of the parameter in the original labels must be represented in this new system.

The Feature extractor: Taking the CenterNet (Duan et al., 2019) and the Hourglass (Newell et al., 2016) Network as a reference, the feature extractor is built of two parts, pre-processing module and a stacked hourglass module. The preprocessing module is a convolution module, including three times residual module and three times pooling, and its output shape is 64 × 64. A basic hourglass structure has a six-step (128, 64, 32, 16, 8, 4) down-sampling and a six-step up-sampling part, which are connected to each other by skip connection. Between two hourglasses, there is a merging layer, which merges the heatmaps of the last stage and the middle feature into a new feature with a shape of 64 × 64 × 256. This merging module is designed to fuse the feature from the previous hourglass into the next hourglass, helping realize the middle supervision.

The 2D BBox and the 3D BBox Head and decoder: The prediction head consists of two parts: a heatmap generator and a loss calculator. Prediction head design is based on the encoding of the prediction target. The centers of 2D BBox (cu_B, cv_B) were encoded as a gaussian heatmap GT_{2DBoxC} , and 2D BBox's (uw_B, vh_B), 3D BBox's $(cx_B, cy_B, cz_B; w_B, h_B, l_B)$ stored in the (cu_B, cv_B)-index cell in the respective heatmap $GT_{2DBoxWH}$ and GT_{3DBox}. The corresponding decoder is used to analyze the heatmap to get the desired parameter set, such as (cu_B, cv_B). For each heatmap mentioned above, a convolutional module was constructed as PD heatmap generators for each parameter set, and the output channel was consistent with the channel of encoded GT heatmap. Their outputs are the predicted heatmap, and the loss is calculated from the predicted heatmap and the ground truth heatmap. Because this is a multi-task network, a weighting factor was assigned for each loss in Equation (1). Where α presents the weighting factor, PD and GT mean Predicted and ground truth heatmap.

$$\begin{aligned} Loss_{BBox} =& \alpha_{2DBoxC} * L_2(PD_{2DBoxC}, GT_{2DBoxC}) \\ &+ \alpha_{2DBoxWH} * L_{r1}(PD_{2DBoxWH}, GT_{2DBoxWH}, Mask) \\ &+ \alpha_{3DBox} * L_{r1}(PD_{3DBox}, GT_{3DBox}, Mask) \end{aligned} \tag{1}$$

Zero Center Fusion Unit (ZCF Unit): The ZCF Unit is a connection module between RPN and KPN, obtaining the feature from the previous stage, processing it, and feeding it into the next stage. As we all know, the deep network is an expert in extracting abstract features, which is useful for importing abstract global information into the next stage. In the RPN in the Faster RCNN network, the feature maps are cropped as the input of the next stage. However, it is insufficient for our 2D and 3D Kpt prediction, which needs abstract global information and also needs

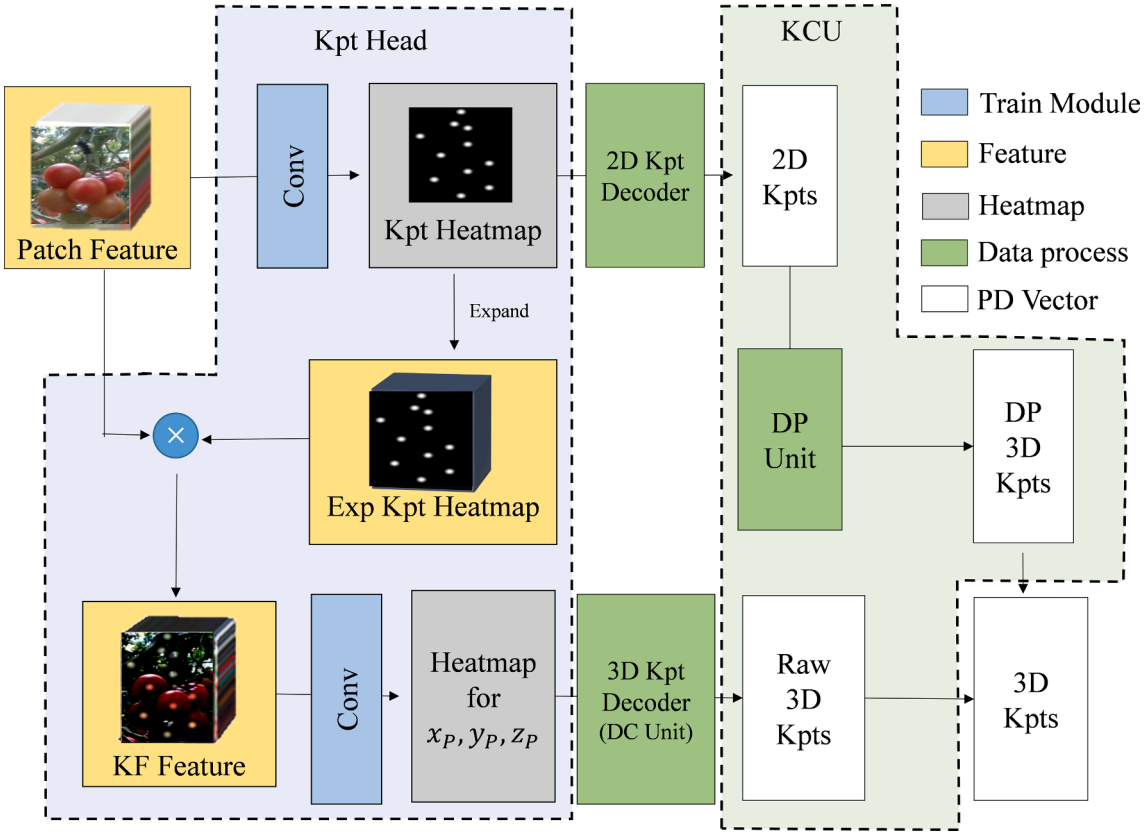

Fig. 5. The detailed schematic of the Kpt Head and Keypoints Concatenate Unit (KCU).

local precise information. Therefore, the ZCF Unit is specifically designed for our task. The ZCF Unit selects the qualified bounding box firstly, crops its corresponding feature map secondly, performs zero center transformation around the center of 3D BBox thirdly, and finally concatenates these data and features into the Patch Feature for KPN. The feature extracted from this ZCF Unit contains the global feature, the local spatial information of a single tomato bunch, which benefits the refinement of 3D BBox and the key points prediction. In addition, the zero-center transformation eliminates the negative influence of the overall offset of the tomato bunch, focusing on the key points distribution inner 3D BBox of the tomato bunch.

2.3.2. Keypoints prediction network (KPN)

The main task of KPN is to predict the 2D Kpts, and 3D Kpts, and the refinement offset of 3D BBox. The preprocessing module, feature extractor, Kpt Head, 2D/3D Kpt decoder and Kpt Cat Unit composed the KPN. The inputs of KPN are the feature in ROI, processed by a 262-channel-in and 256-channel-out preprocessing module, fed into an hourglass module to obtain Patch Feature. The additional goal of our KPN is to tackle the problem of poor-quality source data. Thus, the keypoint-focused processing (KFP) in Kpt Head and Keypoints Concatenate Unit (KCU) were designed. Its input is Patch Feature in Fig. 5.

The 2D and 3D Kpt Head and decoder: There are some delicate designs in the Kpt head and decoder, while their base frame is similar to the BBox Head, BBox decoder, and the loss in RPN. (1) A multi loss function: Like the BBox Head, our Kpt Head also predicts 3D based on 2D. The Kpt head predicts 2D Kpts (u_p, v_P), 3D Kpts (x_P, y_P, z_P), which are encoded via the same encoding method for BBox in RPN. As a result, the decode method of the Kpt Head is similar to the BBox Head, a heatmap PD_{2DKpt} for 2D Kpt, a heatmap PD_{3DKpt} for 3D Kpt. In addition, because the 3D BBox is closely relative to the keypoints location, the 3D BBox Delta Head for $(\delta_{Bcx}, \delta_{Bcy}, \delta_{Bcz}, \delta_{Bw}, \delta_{Bh}, \delta_{Bl})$ prediction is necessary to set in the KPN as the refinement of the 3D BBox. The Loss of Kpt Head is defined in Equation (2).

$$Loss_{Kpt} = \alpha_{Puv} * L_2\left(PD_{2DKpt}, GT_{2DKpt}\right) + \alpha_{3DKpt} * L_{r1}\left(PD_{3DKpt}, GT_{3DKpt}, Mask\right) + \alpha_{B3DH} * L_{r1}\left(PD_{B3DH}, GT_{B3DH}, Mask\right) \quad (2)$$

(2) Keypoint-focused processing (KFP): Coming to precise keypoints detection, the quality of the point cloud is critical, so the property of our RGBD camera needs to be concerned. Although this camera can provide RGBD images in real-time, the point cloud is still sparse, and the null value, infinite value, and false value exist very often, especially on the boundary of objects or on some occluded objects. In principle, only the concrete information around the key points and the abstract information of the whole tomato bunch is crucial information for key points prediction. And this kind of information, which is non-relative to the key points, and has null, infinite, or false values, is harmful to 3D Kpt prediction. Therefore, the keypoint-focused processing (KFP) was designed in the Kpt Head to reduce the negative influence of the poor-quality and distractive point cloud on the 3D Kpt prediction. The operation process was shown in Fig. 5. The Kpt heatmap, as the prediction of 2D Kpt gassian heatmap, was obtained through a convolutional module, then expand as Exp Kpt Heatmap according to shape of Patch Feature. Further, the Exp Kpt Feature times the Patch Feature to get the KF Feature (keypoint-focused Feature), where features around the key points are augmented, and the non-relative features are weakened.

The Keypoints Concatenate Unit (KCU): The KCU is seen as the post-processing of 3D Kpts. The 3D Kpts can be predicted in two ways: deprojection based on 2D Kpts (Deprojection processing, DP) and decoding from the heatmap (Decode processing, DC). The DP method is more direct and highly fits with the source data, but unable to correct the key point whose source data is poor quality or null value. The DC method takes advantage of the abstract ability of deep-layer features, which is an expert in estimating the distribution trend of key points and benefiting the keypoints prediction, even if the key point is occluded. To make a trade-off, the Keypoints Concatenate Unit generates 3D Kpts through two methods, selects sub-sets from them, and combines them as the final output.

2.4. Network training

In terms of multi-task networks, the training process is very crucial. Our TMPv2 is a two-stage multi-task network, so that a three-phase training strategy can be helpful in its performance. The main difference is in the weighting factors' values in the three training phases. In the first phase (0–40 epoch), the weight factor of the Box Loss and Kpt Loss was set as 1.0 and 0.1, focusing on the training of RPN to generate proposals. In the second phase (40–80 epoch), the weight factor of the Box Loss, 3D BBox Delta Loss, and Kpt Loss was set as 0.5, 1.0, and 0.5, focusing on the refinement of the 3D BBox. In the third phase (80–120 epoch), the weight factor of the Box Loss, 3D BBox Loss, and Kpt Loss was set as 0.1, 0.1, and 1.0, focusing on the training of KPN to predict the keypoints. In this stage, the parameters in RPN are frozen, and untrainable, which is to keep the BBox prediction ability. The loss-epoch curve is shown in Fig. 6.

A five-fold cross-validation experiment was performed to validate the generalization capability and robustness of our network. The dataset was split at a ratio of 4:1, with 1600 images for training and 400 images for testing. There is no overlapping between each training dataset and their paired testing dataset. In the five-fold cross-validation experiment, each of the five subsets was used as the testing dataset, and the rest four subsets were collected as the training dataset.

3. Results and discussion

This section shows the result of our TPMv2 in these respects: the result of four tasks, an ablation test of the keypoints-focused module, and the illustration of the prediction results. Each task has its evaluation metrics.

3.1. The results of four tasks

The four tasks of our TPMv2 are the 2D BBox, 3D BBox, 2D Kpt, and 3D Kpt prediction. For convenience, the evaluation metrics are mainly divided into two types, metrics for BBox and Kpts. Further, there are some slight differences in metrics for 2D and 3D.

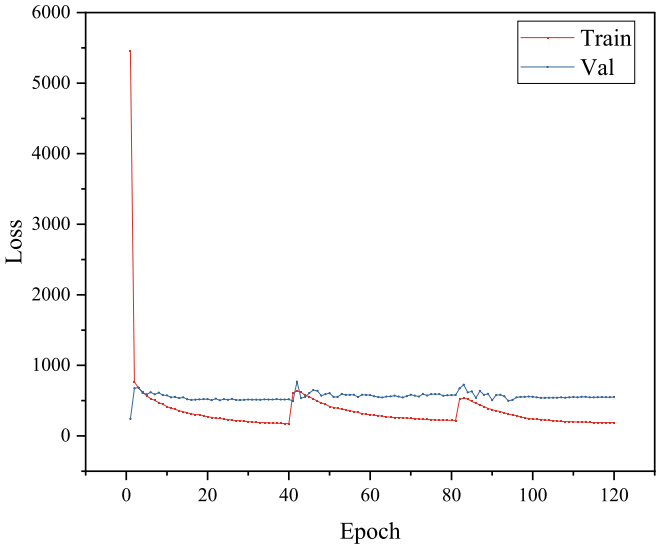

Fig. 6. The loss-epoch curve.

3.1.1. Metrics for BBox

Precision and Recall are typical metrics for bounding boxes in papers (Ali et al., 2018; Liu et al., 2016). To ensure the quality of the bounding boxes, the IoU (Inter-over-union) and the error of the bounding boxes' center were as e complementary metrics.

Precision and Recall: The prediction of bounding boxes was classified into true-positive boxes (TP) and false-positive boxes (FP). The unpredicted bounding boxes were false-negative boxes (FN). And the precision and recall are defined by Equation (3) and Equation (4).

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

IoU: The IoU is the value of intersection-over-union between two boxes, defined by Equation (5). The 2D IoU and 3D IoU were the IoU of area and volume separately Fig. 7.

$$IoU = \frac{Intersection}{Union} \tag{5}$$

***error$_c$*:** The error of the center point ($error_c$) was chosen to evaluate the accuracy of bounding box localization.

$$error_c = \frac{\|PD_{BBoxC} - GT_{BBoxC}\|}{Ref} \tag{6}$$

The $error_c$ is defined by Equation (6), where PD_{BBoxC} is the prediction and GT_{BBoxC} is the ground truth of center point, Ref is related with the height, width, or length of the bounding box. The center of 2D BBox is a 2D coordinate (cu_B, cv_B), while the center of 3D BBox is a 3D coordinate (cx_B, cy_B, cz_B).

3.1.2. Metrics for keypoints

The keypoints detection was widely applied in human pose estimation, so their metrics for keypoints evaluation were taken as reference for our keypoints on tomato bunch.

The δ: The δ is defined by Equation (7), where PD_{ij} means the predicted location of j-th keypoint on the i-th tomato bunch, and GT means ground truth.

$$\delta_{ij} = \frac{\|PD_{ij} - GT_{ij}\|}{Ref_j} \tag{7}$$

PCK: The PCK (Percentage of Correct Keypoints) is a mainstream metric for keypoints detection in papers (ShihEn et al., 2016; Z et al., 2019; Andriluka et al., 2014). This metric counts the Percentage of Correct Keypoints in predicted keypoints. And the judging criteria of one correct key point is based on the value δ, which represents the distance between predicted Kpt and the ground truth Kpt. Using PCK@40, when δ_{ij} is less than 40, the j-th keypoint on the i-th tomato bunch was seen as correctly predicted. Where 40 can be another threshold value, with which the tomato pose can be correctly visualized.

$$\text{PCK@40} = \frac{\sum_i^M \sum_j^N (\delta_{ij} < 40)}{\text{M*N}} \tag{8}$$

The PD_{ij} in the formula can be substituted by 2D Kpt coordinate (cu_P, cv_P) or 3D Kpt coordinate (x_P, y_P, z_P). In addition, the δ shows the positioning error of each keypoints prediction.

Our TPMv2 is a top-down key points detection algorithm that detects the tomato bunch first and estimates the key points on the predicted tomato bunch second. As a result, the key points' positioning accuracy in the global coordinate system (GCS) is affected by the positioning accuracy of the local coordinate system (LCS) in the GCS and the key points' positioning accuracy in the LCS in Fig. 8. In the following, these metrics were analyzed in 2D/3D and LCS/GCS.

3.1.3. Results of multi-tasks

This section analyzed the comprehensive performance of the TPMv2 on four tasks, including the performance in 2D BBox, 3D BBox, 2D Kpt, and 3D Kpt prediction. The Precision and Recall curve and ROC curve of TPMv2 were shown in Fig. 9. The Precision and Recall of 2D BBox were 0.8705 and 0.6084, where the threshold of confidence and IoU of 2D BBox were set as 0.7 and 0.5. The IoU thresholds in Faster R-CNN (Ren et al., 2017), SSD (Liu et al., 2016) and YOLO3D (Ali et al., 2018) are set to 0.7, 0.5 and 0.5, respectively. The confidence thresholds in SSD and FasterR-CNN are set to 0.01 and 0.6, respectively. Although setting a lower confidence threshold of 0.2 can get a higher Recall of 0.92437 to detect more tomato bunch, the Precision becomes low. Because our TPMv2 predicts the other three tasks based on 2D BBox, the Precision and Recall of 2D BBox show the detection performance on tomato bunch prediction and influence other tasks. Hence, the confidence threshold was set to 0.7 to ensure the other three tasks during network inference runtime.

The quality of detected 2D BBox and 3D BBox: The quality of detected 2D BBox and 3D BBox were analyzed from two perspectives: the IoU and the $error_c$.

The Fig. 10 shows that all the predicted 2D BBoxes's IoU are higher than 0.5, $error_c$ less than 0.2, while 12.79 % of 3D BBoxes' IoU are lower than 0.5, 19.21 % of 3D BBoxes' $error_c$ are higher than 0.2. In fact, when the $error_c < 0.3$, a reasonable 3D BBox can be visualized, but it is not accurate enough and brought some negative influence on keypoints

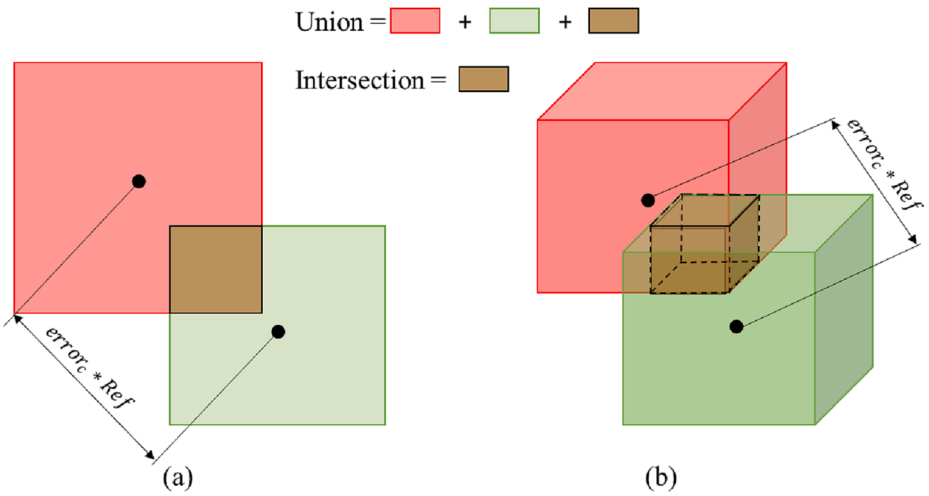

Fig. 7. The schematic of 2D IOU and 3D IOU.

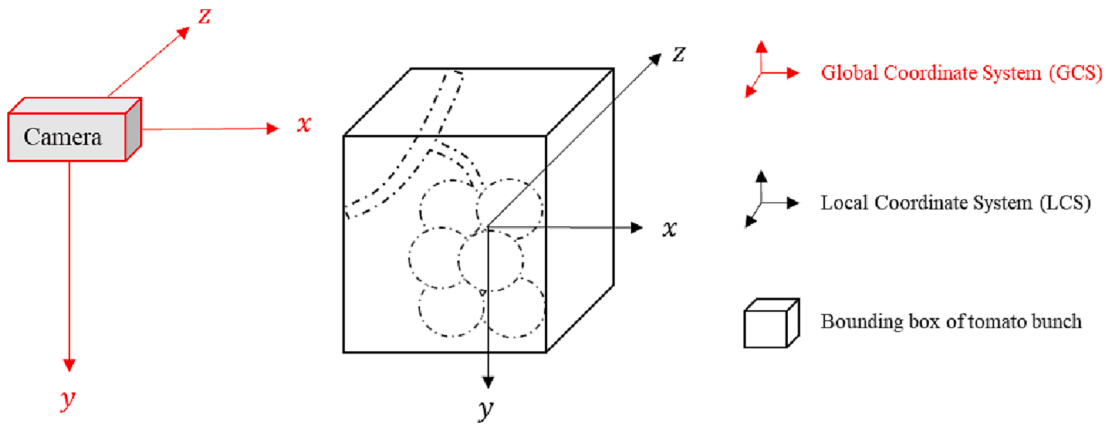

Fig. 8. The schematic of the local coordinate system and the global coordinate system.

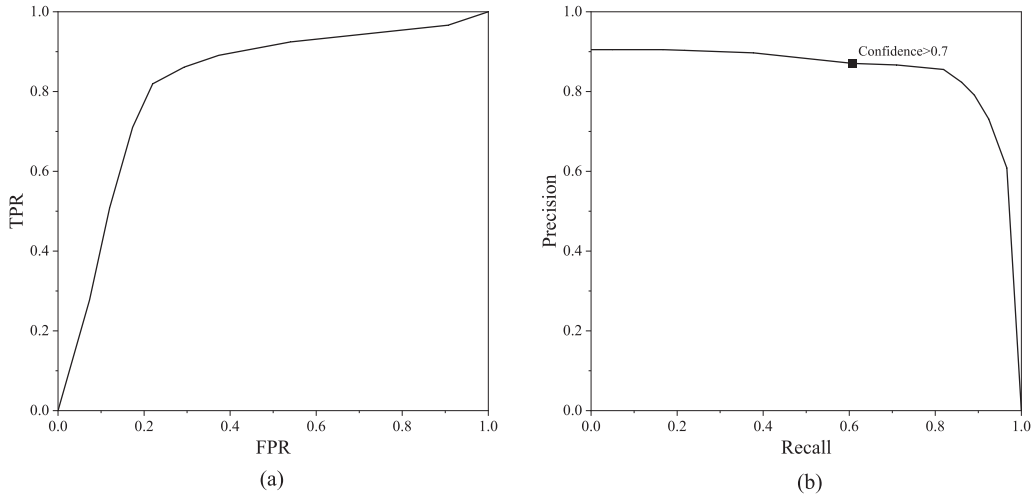

Fig. 9. (a) The Precision - Recall Curve and (b) Receiver Operating Characteristic curve (ROC).

prediction. About 87.00 % of predicted 3D BBox are with $IoU > 0.5$ and $error_c < 0.4$, and 93.72 % of them have the same high quality as the 2D BBox.

The results of 2D/3D Kpt: The distribution of 2D Kpt and 3D Kpt in Fig. 11 and Fig. 12. Both 2D Kpt's medians in GCS are larger than median in LCS. Compared with the 2D Kpt and 3D Kpt in GCS, the KPts in LCS are more concentrated around the median. Because the δ in GCS accumulated the error from bounding box prediction and keypoints prediction. The physical meaning of δ in a 2D Coordinate System is the pixel distance between PD Kpt and GT Kpt, while δ in a 3D Coordinate System means the spatial distance between PD Kpt and GT Kpt. The 90th percentile lines of all 3D Kpts in GCS and LCS are located around $\delta = 30$, which means a positioning error of less than 30 mm. And more than half of the predicted points' positioning errors are about within 10–20 mm.

3.2. Ablation tests

3.2.1. Ablation test of KPN

Except for our KPN, the 3D Kpt also can be predicted by the Deprojection Processing, DP, which inferred 3D Kpt from the RGBD image with 2D Kpt as the index, referring to TPMv1. For this reason, the results of two methods, TPMv2 (RPN + KPN), and TPMv1 (RPN + 2DKPN + DP), were compared in Fig. 13. The error of each 3D Kpt was drawn as a point in the error coordinate system, whose distribution shows the distribution of errors. The blue transparent sphere, with a radius of 50 mm, is a boundary for the outlier, where the normal value inner the sphere and the abnormal value outer the sphere. The errors of RPN + KPN in Fig. 13(a) are more concentrated and have fewer outliers, while the errors of RPN + 2DKPN + DP in Fig. 13(b) have 25.7 % outliers caused by the poor quality, data occlusion of RGBD data source. Only four out of 1738 points are outliers by the RPN + KPN method (our TPMv2).

The Fig. 14 takes a typical example to show the abnormal value repairing effect of our KPN. This abnormal value belongs to the key points S2 and P3, the end point of the stem and the start point of the peduncle, which are very important for tomato pose prediction. This abnormal prediction was caused by the poor quality of the source data, the null value source data. It can be seen in Fig. 14(b) that the absence of the point cloud around the S2 and P3. As a result, it is impossible to get the right value by Deprojection Processing. In contrast, our TPM can predict these keypoints successfully. The reason lies in the structure of our KPN, which consist of the DP, DC, and KCU module, taking advantage of deep learning and deprojection processing. The DP is adept at indexing the position of the key point in a source-data-closely-relative way, while the deep learning is adept at trend prediction as a constraint to the DP to avoid the abnormal-value indexing. It is a noticeable improvement in Kpt prediction and positioning and benefits the subsequent robot operation of non-destructive harvesting.

3.2.2. Ablation test of four processing units

The KPN of our TPMv2 are specifically designed for our task, including some special processing, the Keypoint-focused Processing (KFP), Keypoints Concatenate Unit (KCU), Deprojection Processing (DP), and 3D BBox Dlt prediction head (3DBH). The effects of each

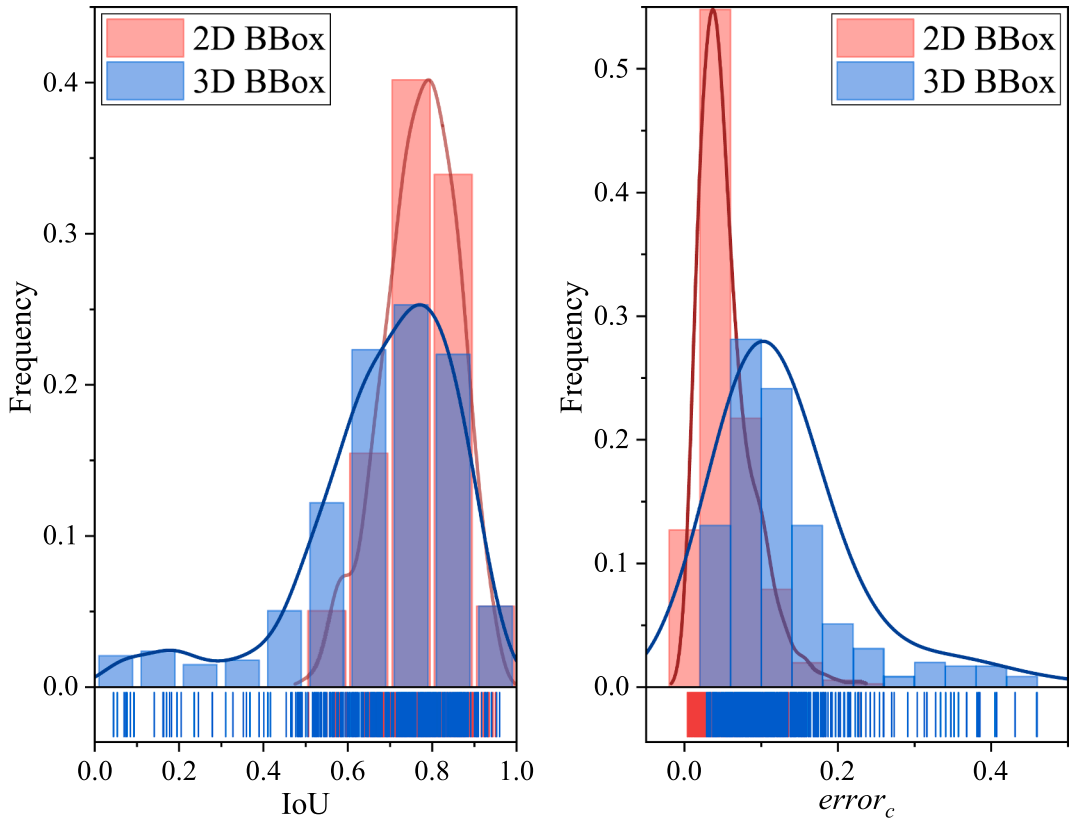

Fig. 10. The histogram of the IoU and the $error_c$ of the predicted 2D BBoxes and 3D BBoxes.

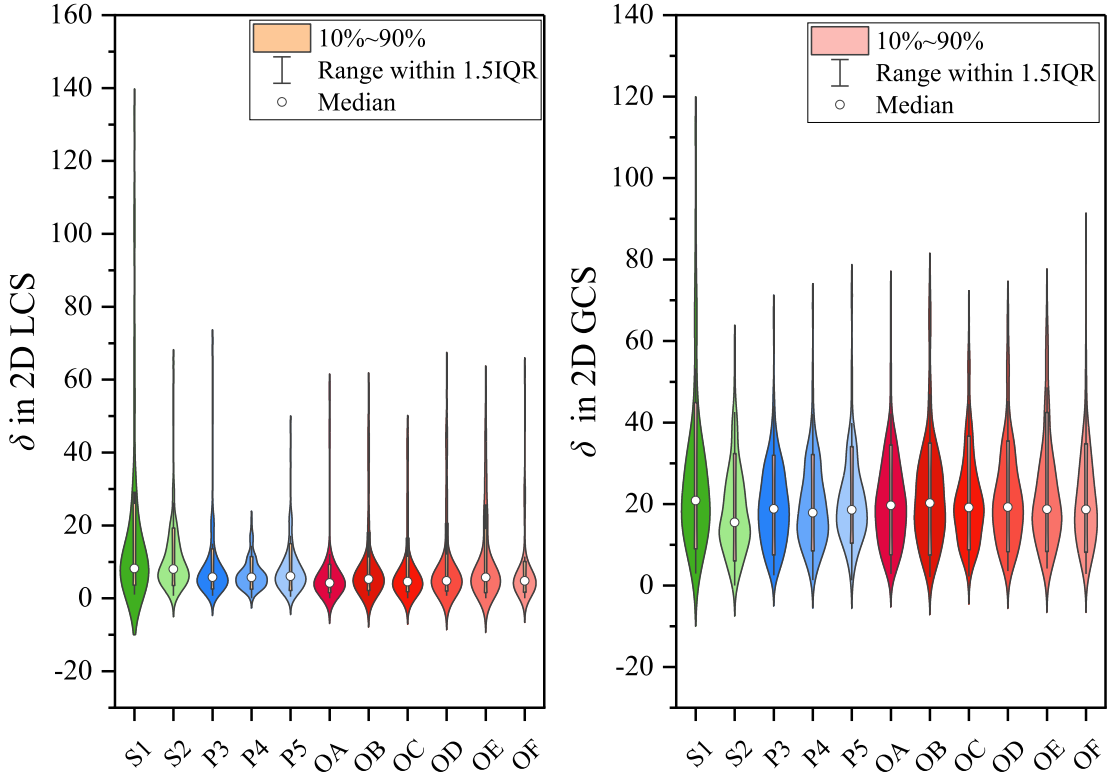

Fig. 11. The distribution of 2D Kpts' δ in the Local Coordinate System (LCS) and the Global Coordinate System (GCS).

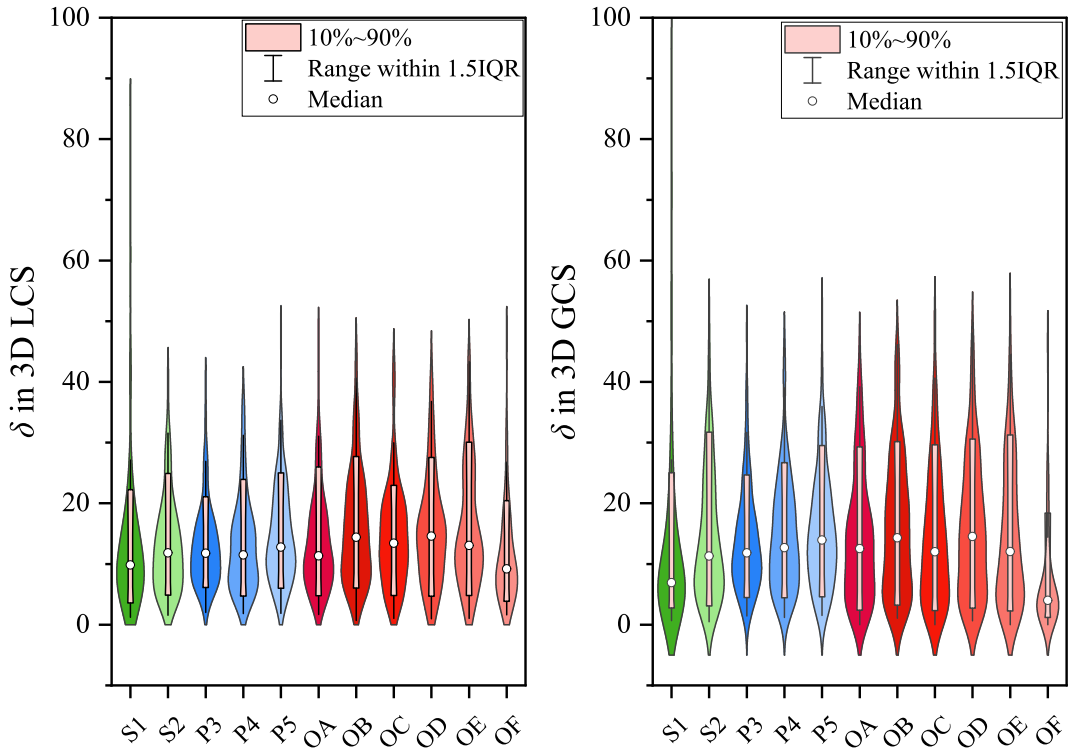

Fig. 12. The distribution of 3D Kpts' δ in the Local Coordinate System (LCS) and the Global Coordinate System (GCS).

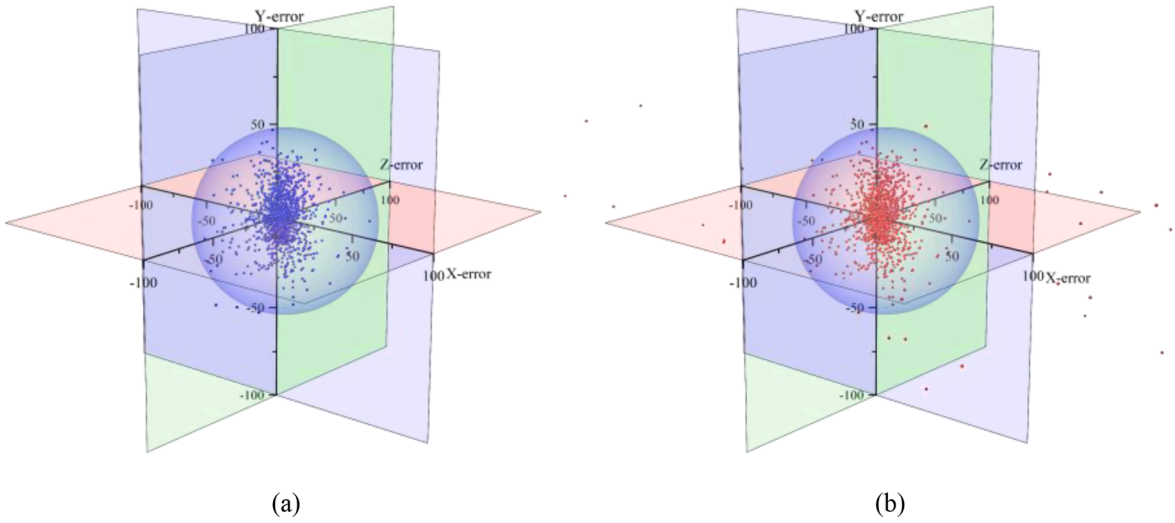

Fig. 13. The positioning error of 3D Kpt in x-, y-, z-axis. Each error is drawn as a point. The blue transparent sphere is a boundary for the outlier. (a) The positioning error of blue 3D Kpts predicted by RPN + KPN. (b) The positioning error of red 3D Kpts predicted by RPN + 2DKPN + DP. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

processing module were tested in this comparative experiment, whose results were shown in Table 1. For the convenience of comparison, the Precision, Qualified Percent and PCK were calculated under the same condition on different networks. The precision of 2D BBox is calculated under the condition: Confidence > 0.7, and IoU > 0.5. The qualified percent of 3D BBox is calculated under the condition: $IoU > 0.5$ and $error_c < 0.4$. The PCK of 2D Kpt and 3D Kpt was calculated under the conditions respectively: $\delta < 40$ and $\delta < 20$, namely taking the PCK@40 of eleven 2D key points and PCK@20 of eleven 3D key points as indicators.

In addition, TPMv1 was chosen as a comparative network, which predicts 2D keypoints from one RGB image, inferring 3D keypoints by deprojection. The PCKs of 3D keypoints are calculated in GCS and LCS. While the PCKs of 3D keypoints in GCS are suitable to revel the accuracy of positioning, the PCKs of 3D keypoints in LCS are suitable to reveal the correctness of pose trend prediction for a single bunch.

To robustify the experimental analysis, the *t*-test statistical method was selected to test the difference between two means with different variances. The result of six comparative experiments are shown in Table 2. The Experiment Net II - Net I and Net III - Net II are designed to evaluate the 3DBH module and the KFP module, respectively. The Experiment Net IV - Net III, TPMv2 - Net III and TPMv2 - Net IV are designed to compare the effect of the DC, the DP and the DC + DP + KCU on 3D Kpt prediction, on 3D Kpt prediction both in GCS and LCS.

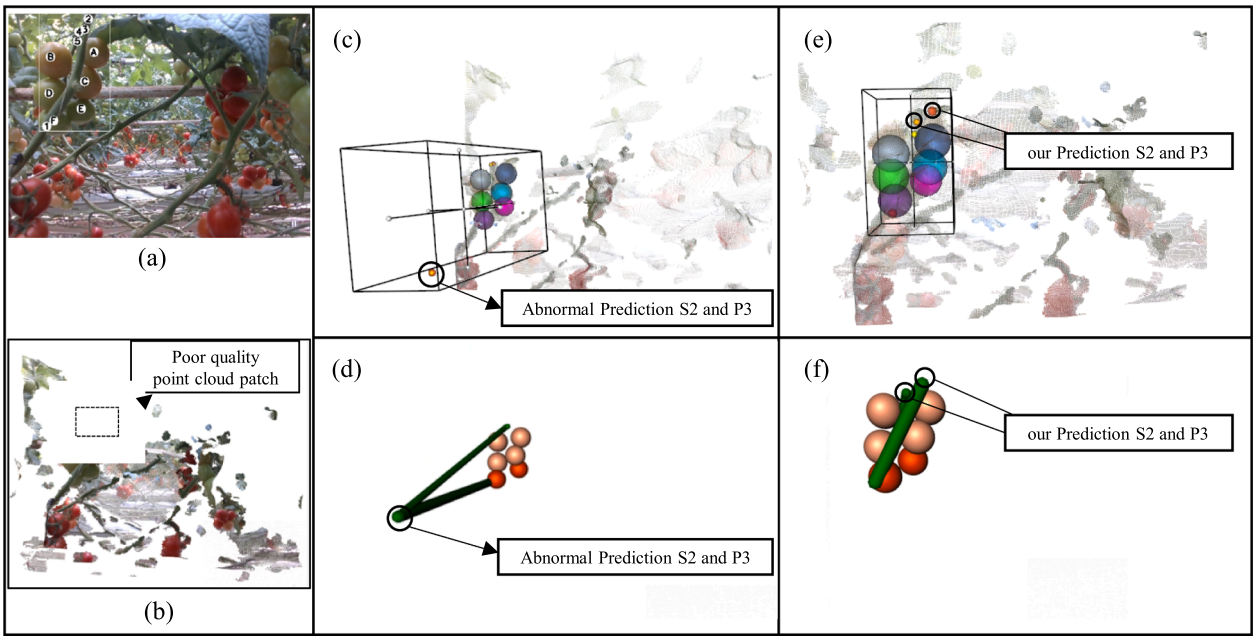

Fig. 14. The abnormal value repairing effect of our KPN. (a) RGB image and (b) point cloud are the source data. The prediction of TPMv1 is shown in (c) and (d), where P3 and S2 were predicted with an abnormal value (Null value). The prediction of our TPM v2 (RPN + KPN) is shown in (e) and (f), where S2 and P3 were predicted correctly. Our KPN contains the DP + DC + KCU module. The (c) and (d) were the visualization of 3D Kpts in the point clouds. The (d) and (f) are the simulation tomato bunch based on predicted 3D Kpts.

Table 1
There are five network with diefrent structure. The Keypoint-focused Processing (KFP), Keypoints Concate Unit (KCU), Deprojection Processing (DP), Decode Processing (DC) and 3D Kpt Dlt prediction head (3DBH) are optional for each network.The Precision of the 2D BBox and qualified percent of 3D BBox, the Percentage of Correct Keypoints (PCK) of 2D Kpt and 3D Kpt were taken as the indicators. The format of data is Means (Standard Deviations).

	KFP	KCU	DC	DP	3DBH	2D BBox	3D BBox	2D Kpt	3D Kpt in GCS	3DKpt in LCS
TPMv1				√		**0.9782(0.01)**	–	**0.9473(0.02)**	0.6467(0.01)	0.5168(0.08)
I			√			0.9524(0.04)	0.4634(0.19)	0.9161(0.05)	0.4511(0.10)	0.6083(0.10)
II			√		√	0.9564(0.03)	0.6654(0.10)	0.8605(0.05)	0.3840(0.12)	0.5484(0.11)
III	√		√		√	0.9614(0.02)	0.7840(0.05)	0.9470(0.03)	0.6152(0.09)	**0.8065(0.04)**
IV	√			√	√	0.9096(0.02)	0.5320(0.07)	0.8517(0.03)	0.7246(0.03)	0.4384(0.03)
V (TPMv2)	√	√	√	√	√	0.9372(0.02)	**0.8700(0.02)**	0.8882(0.01)	**0.7836(0.02)**	0.7187(0.02)

Table 2
The *t*-test result of a difference between the two networks' means. The format of data is Difference (P-Value). The results, whose P-value is less than 0.05, are marked in bold.

	2D BBox	3D BBox	2D Kpt	3D Kpt in GCS	3DKpt in LCS
Net II - Net I	0.0040 (0.44)	**0.2020 (0.04)**	−0.0556 (0.06)	−0.0670 (0.18)	−0.0600 (0.19)
Net III - Net II	0.0051 (0.40)	**0.1187 (0.03)**	0.0864 **(0.01)**	**0.2312 (0.00)**	**0.2582 (0.00)**
Net IV – Net III	**−0.0519 (0.00)**	**−0.2521 (0.00)**	**−0.0953 (0.00)**	**0.1093 (0.03)**	**−0.3681 (0.00)**
TPMv2 - Net IV	**0.0276 (0.03)**	**0.3380 (0.00)**	**0.0365 (0.04)**	**0.0591 (0.01)**	**0.2803 (0.00)**
TPMv2 - Net III	**−0.0243 (0.04)**	**0.0859 (0.01)**	**−0.0588 (0.00)**	**0.1684 (0.01)**	**−0.0878 (0.00)**
TPMv2 - TPMv1	**−0.0410 (0.00)**	–	**−0.0591 (0.00)**	**0.1370 (0.00)**	**0.2019 (0.00)**

From Net I to Net II, the precision of 3D BBox rise, which shows that the module 3DBH benefits the prediction of 3D BBox. From Net II to Net III, the BBox 3D, 3D Kpt in GCS, and 3D Kpt in LCS improved significantly with 0.1187, 0.2312, and 0.2582 because of the introduction of the KFP. The KFP module focused more on the feature around the keypoints and eliminated the negative impacts from other non-relative features. In Net III and Net IV, the effect of DP and DC were compared. Compared with Net III, the PCK value of 3D Kpt in GCS on Net IV is higher at 0.1093, which means better in positioning in the global coordinate system. But the Net IV performs poorly in 3D BBox and 3D Kpt in LCS prediction, which means poor pose trend prediction. The reason lies in the DP module in Net IV, which lacks the ability to avoid the abnormal value in the point cloud. From Net IV to TPMv2, the performance is improved in all aspect, because the DP + DC + KCU inherit the advantage of DC and fix the weakness of DP. In contrast, our TPMv2 contains the DP + DC + KCU module, which improves the 3D Kpt significantly. From the comparative experiment TPMv2 - Net III, it can be inferred that some abilities in 2D BBox and 3D Kpt in LCS are sacrificed, while it predicts the 3D Kpt in GCS with higher PCK. Compared with TPMv1, our TPMv2 sacrificed some performance on the 2D BBox and 2D Kpt prediction to significantly improve 3D Kpt prediction, both in GCS and LCS, namely higher accuracy in positioning and pose trend prediction.

3.3. Visualization of the results

The pose types of the tomato bunch are infinite because the tomato bunch is a natural object. To illustrate the performance of our TPMv2, four typical pose types were selected in Fig. 15, (I) Right, (II) Left, (III) Front, (IV) Back Pose. In the natural state, the common tomato pose is an intermediate pose between any two typical poses. From the source data

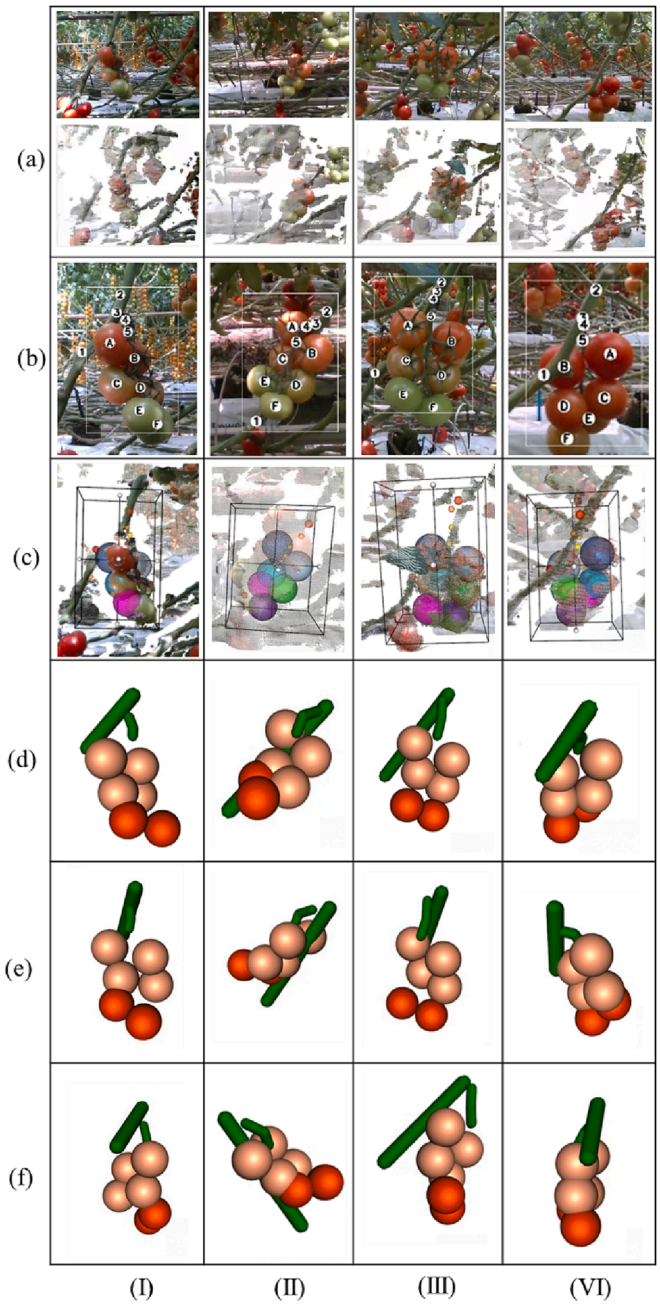

Fig. 15. Four typical tomato bunch poses, one pose in one column, (I) Right, (II) Left, (III) Front, (IV) Back. Row (a) is the source data, including RGB and point cloud. Row (b) is the 2D BBox and 2D Kpt prediction results. Row (c) is the 3D BBox and 3D Kpt prediction results. Row (d) (e) (f) is the front view, right view, and left view of the visualization of predicted 3D Kpts.

in row (a), it is worth noticing the coarse quality of the point cloud, where the infinite/null value, hole, coarse boundary, and sparse density often come. In the Right Pose, the stem and three fruit are partially occluded. In the Left Pose and Right Pose, the stem is partially occluded. In the Back Pose, two keypoint of the peduncle and one fruit are occluded by other plant organs. Row (b) shows the correct detection of 2D BBox and 2D Kpt, while Row (c) shows the correct detection of 3D BBox and 3D Kpt even with the above-mentioned negative influence. Row (d; e), and (f) are aimed to illustrate the predicted 3D pose comprehensively, which shows the position and pose of each plant organ, especially the pose of the peduncle curve. It is obvious that the curves of the peduncle in four pose bend in different directions, which is clearly and correctly visualized by two small cylinders based on three key points. The mechanism of our TPMv2 to find the peduncle is more reasonable than the other methods, which find the cutting point on the peduncle in a fixed region of interest (ROI) (Kenta et al., 2009; Lufeng et al., 2016). In addition, our TPMv2 provides more comprehensive and structured information about the tomato bunch, including the stem, the peduncle curve, and the fruit. In addition, our TPMv2 provides more comprehensive and structured information about the tomato bunch, including the stem, the peduncle curve, the fruit, the position of the whole tomato bunch, and the position and pose of each plant organ. It outperforms these algorithms, which simplify the tomato bunch as one or two bounding boxes (Qin et al., 2021).

Although the pose trend can be predicted correctly and the abnormal prediction can be avoided, the positioning accuracy of our algorithm needs further improvement. Columns (II) and (II) in Fig. 15 show the correct prediction of the pose trend, but the key point P3 is not predicted at a high positioning accuracy. Therefore, the simulated stem and peduncle are not closely junction at key point P3. Due to the not high positioning accuracy of fruits, there are some inferences between the stem and fruit in Fig. 15 column (II) row (e). In future research, the main task is to improve positioning accuracy by enlarging the dataset scale, balancing multiple tasks better, and optimizing the network structure.

4. Conclusions and prospection

To cope with the poor data quality of economical RGBD cameras, occlusion between plant organs, various tomato poses, and unstructured working environments, our research proposed an improved version of the Tomato Pose Method (TPM), namely TPMv2, which is an end-to-end multi-task network. This network provides comprehensive information on the tomato bunch, including the positions and poses of the stem, peduncle, and fruits, by predicting the 2D BBox, 3D BBox, 2D Kpt, and 3D Kpt. The KPN, containing the innovatively proposed DP + DC + KCU, solved the problem of abnormal prediction caused by poor-quality source data. The Precision of 2D BBox and the Qualified Percent of 3D BBox reached 0.9372 and 0.8700, and the Percentage of correct detected Keypoints (PCK) of 2D Kpt and 3D Kpt reached 0.8882 and 0.7836. About 78.36 % of 3D Kpts' positioning errors are less than 20 mm, which is sufficient to describe a correct pose trend based on the 3D Kpt. It benefits the manipulator to plan a more reasonable trajectory for non-destructive harvesting. Even though this algorithm achieves some improvement, there is still room to improve the positioning accuracy for delicate harvesting at all tomato poses. Updating the camera, enlarging the dataset scale, balancing multiple tasks better, and optimizing the network structure could be some further research directions to improve positioning accuracy.

CRediT authorship contribution statement

Fan Zhang: Conceptualization, Methodology, Software, Investigation, Formal analysis, Data curation, Writing – original draft. **Jin Gao:** Conceptualization, Methodology, Investigation, Formal analysis, Data curation. **Chaoyu Song:** Data curation, Investigation, Validation. **Hang Zhou:** Data curation, Investigation. **Kunlin Zou:** . **Jinyi Xie:** Data curation, Investigation, Validation. **Ting Yuan:** Funding acquisition, Resources, Supervision. **Junxiong Zhang:** Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

Funding: This research is supported by Ministry of Science and Technology of the People's Republic of China, "Key Technologies Research and Development Program" [grant numbers CN 2016YFD0701501].

The greenhouse was provided by Beijing Hongfu International Agricultural Science and Technology Co., Ltd. All authors contributed to the work. Fan Zhang made the dataset, proposed the algorithm, performed the experiments, and analyzed the data. Junxiong Zhang guided the research work. Jin Gao, Jinyi Xie captured and labeled the images in greenhouse. Ting Yuan, Wei Li, Suzhou Botian Co, Ltd, and Hongfu Co, Ltd provided the device and green house. Fan Zhang edited the paper; Junxiong Zhang, Hang Zhou, Chaoyu Song, Kunlin Zou proofread the paper. All authors have read and agreed to the published version of the manuscript.

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service.

References

Ali, W., Abdelkarim, S., Zahran, M., Zidan, M., Sallab, A., 2018. YOLO3D: end-to-end real-time 3D oriented object bounding box detection from LiDAR point cloud. CoRR abs/1808.02350. https://doi.org/10.1007/978-3-030-11015-4_54.

Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B., 2014. 2D human pose estimation: new benchmark and state of the art analysis. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3686–3693. https://doi.org/10.1109/CVPR.2014.471.

Arad, B., Balendonck, J., Barth, R., Ben-Shahar, O., Edan, Y., Hellström, T., Hemming, J., Kurtser, P., Ringdahl, O., Tielen, T., Tuijl, B., 2020. Development of a sweet pepper harvesting robot. J. Field Robot. 37, 1027–1039. https://doi.org/10.1002/rob.21937.

Bac, C.W., Hemming, J., van Tuijl, B.A.J., Barth, R., Wais, E., van Henten, E.J., 2017. Performance evaluation of a harvesting robot for sweet pepper: performance evaluation of a harvesting robot for sweet pepper. J. Field Robot. 34, 1123–1139. https://doi.org/10.1002/rob.21709.

Haan, T. de, Kulkarni, P., Babuska, R., 2021. Geometry-Based Grasping of Vine Tomatoes. ArXiv abs/2103.01272.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. CoRR abs/2010.11929.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q., 2019. CenterNet: Keypoint Triplets for Object Detection . 2019 IEEE/CVF International Conference on Computer Vision (ICCV) 6568 – 6577. https://doi.org/10.1109/ICCV.2019.00667.

Eizicovits, D., van Tuijl, B., Berman, S., Edan, Y., 2016. Integration of perception capabilities in gripper design using graspability maps. Biosyst. Eng. 146, 98–113. https://doi.org/10.1016/j.biosystemseng.2015.12.016.

He, Z., Feng, W., Zhao, X., Lv, Y., 2020. 6D pose estimation of objects: recent technologies and challenges. Appl. Sci. 11, 228. https://doi.org/10.3390/app11010228.

He, K., Georgia, G., Dollár, R.P., Girshick, R., 2017. Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV) 2980 – 2988. https://doi.org/10.1109/ICCV.2017.322.

Kenta, S., Satoshi, Y., Ken, K., Yasushi, K., Junzo, K., Mitsutaka, K., 2009. Evaluation of a strawberry-harvesting robot in a field test. Biosyst. Eng. 2, 160–171.

Kim, J., Pyo, H., Jang, I., Kang, J., Ju, B., Ko, K., 2022. Tomato harvesting robotic system based on Deep-ToMaToS: Deep learning network using transformation loss for 6D pose estimation of maturity classified tomatoes with side-stem. Comput. Electron. Agric. 201, 107300 https://doi.org/10.1016/j.compag.2022.107300.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: Single Shot MultiBox Detector, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.

Liu, G., Nouaze, J.C., Touko Mbouembe, P.L., Kim, J.H., 2020. YOLO-tomato: a robust algorithm for tomato detection based on YOLOv3. Sensors 20, 2145. https://doi.org/10.3390/s20072145.

Lufeng, L., Yunchao, T., Xiangjun, Z., Min, Y., Wenxian, F., Guoqing, L., 2016. Vision-based extraction of spatial information in grape clusters for harvesting robots. Biosyst. Eng. 151, 90–104.

Newell, A., Yang, K., Deng, J., 2016. Stacked Hourglass Networks for Human Pose Estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 483–499. https://doi.org/10.1007/978-3-319-46484-8_29.

Qin, Z., Jianmin, C., Bin, L., Can, X., 2021. Method for recognizing and locating tomato cluster picking points based on RGB-D information fusion and target detection. Trans. Chin. Soc. Agric. Eng. 37, 143–152.

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.

ShihEn, W., Varun, R., Takeo, K., Yaser, S., 2016. Convolutional pose machines. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4732. https://doi.org/10.1109/CVPR.2016.511.

Yuanyue, G., Ya, X., F.P., J., 2020. Symmetry-based 3D shape completion for fruit localisation for harvesting robots. Biosyst. Eng. 197, 188–202.

Zhang, F., Gao, J., Zhou, H., Zhang, J., Zou, K., Yuan, T., 2022. Three-dimensional pose detection method based on keypoints detection network for tomato bunch. Comput. Electron. Agric. 195, 106824 https://doi.org/10.1016/j.compag.2022.106824.

Zhao, Y., Gong, L., Liu, C., Huang, Y., 2016a. Dual-arm robot design and testing for harvesting tomato in greenhouse. IFAC-PapersOnLine 49, 161–165. https://doi.org/10.1016/j.ifacol.2016.10.030.

Zhao, Y., Gong, L., Zhou, B., Huang, Y., Liu, C., 2016b. Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis. Biosyst. Eng. 148, 127–137. https://doi.org/10.1016/j.biosystemseng.2016.05.001.

Z, C., GHidalgo, M., T, S., S, W., YA, S., 2019. OpenPose: Realtime MultiPerson 2D Pose Estimation using Part Affinity Fields. IEEE Trans. Patt. Anal. Mach. Intell.