Cognitive Computation (2025) 17:132
https://doi.org/10.1007/s12559-025-10488-0

RESEARCH

Test-Time Adaptation of a Multi-Class Object Localization and Size Estimation Framework for Smart Agriculture Applications

Zixu Liu[1] · Qinhao Wu[2] · Yuan Chai[3] · Huan Yu[1]

Received: 13 August 2024 / Accepted: 20 July 2025 / Published online: 28 July 2025

Abstract
Smart agriculture brings massive amounts of real-time images generated via modern information and communication technology. Promptly providing accurate estimates of fruit/vegetable information, such as location, quantity, and size, is worth studying. Therefore, we focus on exploring a deep learning-based backbone model for heatmap regression to capture the yield information. This singular and lightweight architecture effectively addresses the unified challenge of object counting, location detection, and size estimation for fruits/vegetables. However, when dealing with real-world applications, the data distribution shift would happen in response to the collection of new data. Moreover, some unseen fruits/vegetables often appear during the training process. All of these give rise to the open set recognition (OSR) problem. In such an OSR environment, a test-time domain adaptation approach based on deep learning is proposed for multi-class object localization and size estimation. This is the first attempt at unsupervised domain adaptation for heatmap regression tasks. Furthermore, to overcome the drawback of lacking a public dataset, a new benchmark dataset (including synthetic and real image data) has been created and collected to train, test, and evaluate our approach. Extensive experimental evaluations prove that our approach can achieve accurate predictions in the OSR setting within a single epoch of test-time optimization without altering the training process.

Keywords Domain adaptation · Object counting · Size estimation · Open set recognition · Adaptive receptive fields · Synthetic dataset

Introduction

The Fourth Industrial Revolution (Industry 4.0) created opportunities for companies from different sectors to adopt new technologies and gain competitive advantages in domestic and global market digital transformation [1]. This digital revolution introduces modern technologies and innovations, such as the Internet of Things, to the agricultural field, which forms "Smart Agriculture" [2]. Food production is estimated to decrease from 7 to 23% due to climate change [3]. Smart agriculture with growth monitoring is one of the scalable solutions for sustainable crop production. Aiming to guarantee crop production while reducing resource use, artificial intelligence has been widely applied in agriculture to automate traditional farming processes [4]. Smart sensors are now commonly used to capture real-time images of crops, allowing for effective monitoring and prediction of their growth. With integrated Semantic Image Segmentation algorithms, it can automatically pre-process these photos to remove the noise information, such as background or occlusion of stems and leaves, for the convenience of the latter stage of data analysis. However, massive amounts of data will be generated every day. One of the main challenges in smart agriculture is how to process the data efficiently and effectively. For example, it is important to provide adequate yield information for farmers to plan harvesting operations in a competitive

✉ Qinhao Wu
q.wu@lacdr.leidenuniv.nl

✉ Yuan Chai
y.chai@gdut.edu.cn

Zixu Liu
zixu.liu@soton.ac.uk

Huan Yu
huan.yu@soton.ac.uk

[1] Southampton Business School, University of Southampton, University Road, Southampton SO17 1BJ, Hampshire, UK

[2] Leiden Academic Centre for Drug Research, Leiden University, Rapenburg 70, Leiden 2311 EZ, Netherlands

[3] School of Computer Science and Technology, Guangdong University of Technology, Panyu District, Guangzhou 510006, Guangdong, China

market. In this scenario, real-time growth monitoring to capture the number, size, and surface outlook information of each individual fruit/vegetable from the images sent from the field is worth studying. This paper proposes a top-down deep learning model for detecting, classifying, counting, and estimating the size and count of various types of fruits and vegetables. This approach aims to circumvent the shortcomings of the classical bottom-up method that relies heavily on accurate segmentation. This study aims to address the problem of identifying the target object from images with many irrelevant objects, such as leaves and stems, and capturing its size. Moreover, there are various kinds of fruits and vegetables in the real world, resulting in distribution shifts when new data is collected. Such shifts may be caused by changes in the properties or domain of the new data [5, 6]. The distribution shift can also occur because the training data fails to cover all aspects of the distribution [5, 7, 8]. All of these give rise to the open set recognition (OSR) problem, a more challenging and realistic setting where test samples are from unseen classes during training [9]. Therefore, how to achieve such localization, counting, and size estimation tasks in an OSR setting is our main challenge.

Although deep learning (DL) can be very useful for this problem and has already been used in smart farming and other real-world applications, due to the lack of proper datasets, most existing research mainly focuses on scenarios where: (1) only one type of object exists in the dataset; (2) only a single object in each image is considered. Providing such individual object information (i.e., size and position) for the images by hand annotation is time- and effort-consuming, especially since there are huge numbers of objects (fruits/vegetables) in the image and/or multiple classes of objects exist in the dataset. The lack of a proper training dataset is the reason why most current research separately studies the counting and size estimation problems. In addition, none of the research considers an OSR setting. The detail of related literature is explained in detail in the "Literature Review and Methodology" section.

Contributions and Paper Structure

The contributions of this paper are summarized as follows:

- We propose a deep learning-based backbone model predicated on heatmap regression. This is a singular and lightweight architecture which effectively addresses the unified challenge of object counting, location detection, and size estimation.
- A testing-time adaptation-based approach is further developed to fine-tune the trained backbone model to handle previously unseen classes of objects. This approach is the first attempt to handle source-free domain adaptation on a heatmap regression task. The best strategies and parameter settings for different scenarios and tasks in this approach are tested and discussed.
- A new benchmark dataset (which includes synthetic and real image data) is created and collected to train, test, and evaluate the approaches mentioned above for the open set object counting, localization, and size estimation problems. It could be customized and used as a benchmark or testbed for future studies.
- We further conduct experiments to compare the performance of our model with prior arts, and ours can achieve state-of-the-art performance on the proposed benchmark dataset.

The rest of the paper is organized as follows: In the "Literature Review and Methodology" section, state-of-the-art related works from aspects of automated object counting, object size estimation, and TTA are reviewed. From these, the challenges and research questions of this area are identified. The details of the proposed framework for multi-class object counting and size estimation in the OSR setting for the smart agriculture application are depicted in the "A TTA-Based Approach for Multi-Class Object Localization and Size Estimation" section. In the "Experiments for TTA-Based Approach in OSR" section, we illustrate the designed experiments for our TTA-based approach for multi-class object counting and size estimation in an OSR setting. The "Conclusion" section discusses the management implications of our work and concludes the article.

Literature Review and Methodology

In this section, we review related works from the aspects of object counting and size estimation datasets, as well as the TTA, to identify the challenges and research questions addressed in this paper. Our methodology for such research questions is also depicted.

Challenges and Research Questions

Compared to machine learning methods, deep neural networks (DNNs) have shown great capabilities for classification and regression in image processing. Therefore, it could achieve higher accuracy while better utilizing computing resources in a shorter time [10–12]. Nowadays, DNN is increasingly used for target detection and recognition in the computer vision area [10, 13–15]. Inspired by these, we recognize that using deep learning techniques for image-based object localization and size estimation can be highly beneficial for addressing our research problem. This approach can help farmers know more fruit/vegetable yield information and enable unmanned aerial vehicles (UAVs) to harvest autonomously.

Most existing works focus on counting a single class of objects in the image. For example, crowd counting has been studied extensively in recent years [11, 14]. Most works in this area have employed DL techniques and achieved impressive progress in terms of counting accuracy [11]. Although many benchmark datasets have been published, such as ShanghaiTech [16], UCF_CC_50 and UCF_QNRF [17], only one class is annotated in the images of these datasets. Even there exist some multi-class object counting datasets, such as Tobacco leaf counting from [18], KR-GRUIDAE collected by Go et al. [19] which contain 4 different species of Crane and Anser albifrons, these datasets are not fruit/vegetable related and cannot be applied to applications in smart agriculture. When multiple object classes exist in the dataset to be counted separately in different images (which happens in smart agriculture, i.e., different species of fruits on one farm), multiple models must be trained and deployed. It is imperative to have one unified model for multi-class object counting and object size estimation. However, the lack of proper fruit/vegetable datasets containing multiple object classes in images and proper datasets for estimating object size has been the main barrier to such research.

Most previous studies use traditional detection algorithms for size estimation. Such methods are based on hand-crafted pixel counting [20–22], which has lower performance and cannot return size and position information for the individual object instance. Classical bottom-up approaches are dependent on the accuracy of object segmentation. DL-based Semantic Image Segmentation algorithms such as DeepLab [23] and U-Net [24] faced the challenge in the field of identifying individual objects from a group of fruit/vegetables close to each other and output-related position and size information. Segment anything model (SAM) is designed and trained based on 11 M images (annotated with 1B masks) to segment objects of interest in an image based on a given prompt-based definition of the tasks, e.g., points and boxes [25]. This requirement of training datasets and high computing resources is a barrier to applying SAM in smart agriculture to address the aforementioned research problems. For most of the current object size estimation datasets, there is only one single object in each image or one kind of item, such as strawberry [21], boiled-rice [26], and orange [4]. Even though there are some works related to fruits/vegetables (which are analyzed in Table 1), they mostly focus on either single-class object counting or single object size estimation. Most of these datasets cannot be used in the application of multi-object counting and size estimation problems in smart agriculture. Therefore, considering the limited computing resources in the application scenario of the farm in smart agriculture, how to train a lightweight DL-based model to detect and identify individual objects for multiple objects in object counting and size estimation remains an open problem. To the best of our knowledge, no existing work focuses on this. Therefore, the lack of a public dataset containing multiple object classes and proper information for object numbers and size information for each of its images, and the method to train appropriate lightweight DL-based models for multi-object counting and individual size estimation problems in smart agriculture are the **first two challenges** we faced.

As mentioned above, there are a huge number of different kinds of fruits & vegetables in the real world. In real-world tasks, the data distribution usually shifts along with the newly collected data. Such shifts may be due to changes in the properties or domain of the new data [5, 6]. For example, a new kind of fruit/vegetable on the tree/crop that has not been seen in the training dataset needs localization and size estimation. The distribution shift can also occur because the training data fails to cover all aspects of the distribution [5, 7, 8]. This gives rise to the OSR problem, a more challenging and realistic setting where test samples are from unseen classes during training [9]. However, most of the aforementioned works do not consider the new classes of objects beyond their training datasets. Hence, the generalization ability to the out-of-distribution (OoD) samples of this approach is questioned. Even SAM can segment objects of types that it has not seen without any additional training, but this is achieved based on the specific model architecture in a high-computational resource environment and a uniquely large dataset. Consequently, our primary research objective is identified: to address our first two challenges in an open-set learning context, which is also our third challenge. Domain Adaptation is a promising solution that extrapolates from training data to test data from a different distribution.

Domain adaptation-based methods aim to generalize a model to new distributions while assuming some knowledge about the test distribution, such as unlabeled examples or a few labelled examples known [5, 6, 9]. Liang et al.

Table 1 Comparison between current fruit/vegetable counting and size estimation datasets and E-MOCSE13

Datasets/works	Purpose	No. of classes	Object counting	Size estimation	Extensible	Use in OSR
ImageNet [10, 31]; VegFru [32]	Object Classification	Single/multi	✓	×	×	×
Kiwi [33]; Orange[34]; Strawberry [35]	Object Detection	Single	✓	×	×	✓
Starberry [21]; UAV-Orange [4]	Size Estimation	Single	×	✓	×	×
E-MOCSE13 (Ours)	OSR	**Multi**	✓	✓	✓	✓

[27] proposed a method based on the assumption that the source and target domain data have the same classifier in a common latent space, in which the source encoding module is fine-tuned by maximizing the mutual information between the latent layer features and the classifier output layer. This adapts the classification knowledge obtained from the source domain to the target domain, making the target data classification output closer to the source data classification output. These methods that combine data augmentation and self-supervised learning can effectively circumvent the limitations of the source domain, and improve the model's generalization ability for open domains, regardless of whether the source data is accessible or not [28]. Therefore, domain adaptation-based methods provide a more widely applicable solution for open-domain problems in practical scenarios by minimizing local risk.

Methodology

In object detection tasks, object counting, localization, and size estimation are conventionally addressed as separate regression tasks, necessitating the regression of three real-valued parameters. Considering this and to solve the aforementioned second challenge, we propose a backbone model that aims to consolidate these three tasks by formulating them as a unified heatmap regression problem. Specifically, we convert the input images into heatmaps using annotations, which serve as the targets for regression. Before implementing a domain adaptation strategy for OSR problems, it is essential to design a compatible model that serves as the adaptation target and undergoes supervised training while ensuring differentiability. Focusing on this, our proposed approach utilizes HRNet [29] as the backbone architecture and employs multi-scale fusion to enhance high-resolution representations. In addition, we design two heads for jointly estimating object location and size. Mean squared error (MSE) loss functions are used for heatmap comparison. The number of objects can be obtained by counting the peaks of the locations. The resulting compatible model is trained in a supervised manner to ensure the necessary differentiability for stable adaptation and optimization during unsupervised testing. Unlike SAM which is a bottom-up approach (segment targets in the images based on input prompt first, then do the object size estimation, and make a final decision at the end of the clustering), our approach's objective function is checked in each iteration to see whether it is improved or NOT with the addition of new sub-clusters: (1) each sub-cluster is further partitioned into two new sub-clusters if the fitness value is improved in the last split; (2) reject the split if the objective function does not improve. Given this internal heuristic for cluster acceptance or rejection, the most probable number of states is typically identified within a few iterations, compared to the N-1 iterations required by most bottom-up methods. This simple implementation and accelerated computational speed make divisive clustering an attractive alternative that only requires fewer training datasets and computing resources [30].

To train and test such designed models while solving the aforementioned first challenge, we create a new dataset named E-MOCSE13, which includes **synthetic** and **real** image data. A detailed description of E-MOCSE13 is included in the "Evaluation Metrics and Experiment Dataset" section. Table 1 summarizes the comparative analysis of state-of-the-art fruit/vegetable object classification, counting, and size estimation datasets/works with our extended E-MOCSE13, which shows the advantages of our dataset. This builds fundamentals for our research of object counting and size estimation problems in OSR.

The reason we are not creating real fruit/vegetable image datasets is that obtaining image annotations would be time-consuming and labor-intensive, especially when there are many objects in each image. Unity 3D is a cross-platform game engine that can create a 2D or 3D customized game scene and object, and set the physics properties for all 3D objects to simulate the physical effect in the real world [36]. It has been demonstrated that the data created by a gaming engine can be used in real applications and experiments [37, 38]. Useful prior knowledge can be gained during the experiment since the generated dataset can simulate the real-world scenario, so that the problem domain can be consistent with the real application [13]. Thus, we could say it is promising to import synthetic pictures instead of real images in our experiments to solve the problem of multi-class object counting and size estimation. Besides, the **real images part** of our dataset can be used to evaluate the performance of our model (trained from the synthetic dataset), and TTA fine-tuning strategies can prove the feasibility of our approach in real-world application scenarios, which is conducted in the "The Experiment on the Real Image Dataset" section.

After getting the experiment datasets, we aim to solve the third challenge, that is to build a unified model and domain adaptation strategy. In this way, the problems of target counting, localization, and size estimation can be addressed simultaneously. The previously unseen species of fruits and vegetables, i.e., new images that (only) contain a new object class but do not include any labeling information, can be handled. With the image preprocessing algorithm (i.e., DeepLab and U-Net) integrated with the smart sensor, some of the noise information, such as background or even stems and leaves, can be removed. As a result, our problem can be formulated as shown in Fig. 1.

Traditional algorithms for unsupervised domain adaptation tasks heavily rely on the accessibility of source domain data [39–41]. In practice, factors such as privacy protection, data storage and transmission costs, and computational burden often limit the implementation of these method [42]. To

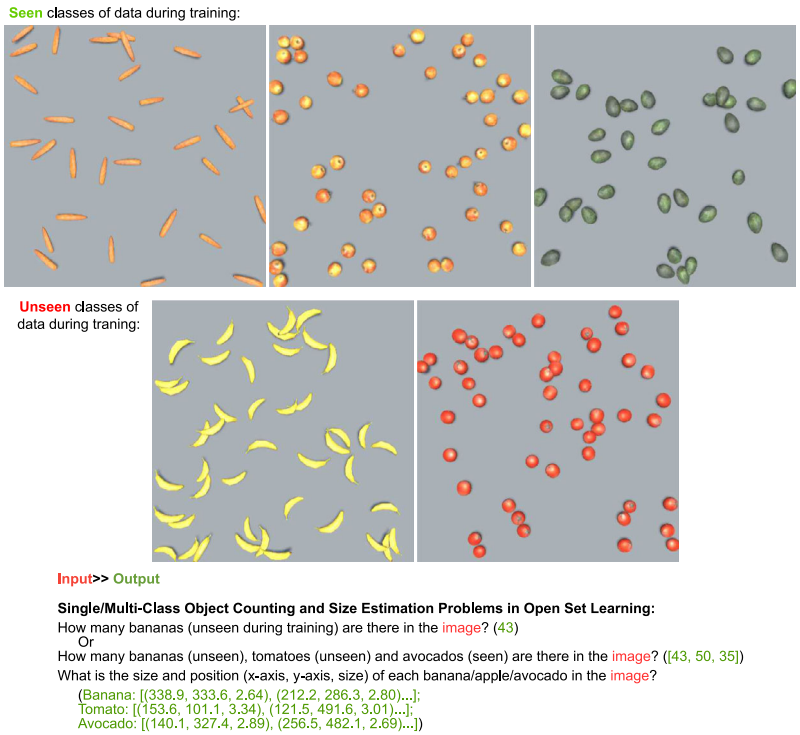

Fig. 1 An illustration of the unified problem of multi-class object counting, position detection, and size estimation in OSR

overcome this problem, the Source-Free UDA task has gradually attracted attention. Under this setting, models trained on source domain data can only solve unsupervised domain adaptation problems with unlabeled target domain data. Considering the restricted access to source domain data, learning domain-invariant features becomes more difficult. During testing/inference time, the model needs to generalize for unseen data from different domains [42]. In many application scenarios, such as Medical Image Segmentation, access is restricted to only test data and pre-trained models [43]. As a result, the given model must be fine-tuned with the test samples to overcome the distribution discrepancies between the training and test data, which is commonly known as testing-time adaptation (TTA) [7]. To achieve this, TTA optimizes the model for confidence, measured by the entropy of its predictions, and updates the model's parameters online on each batch. The best result is achieved in one epoch of test-time optimization without altering the training process. Moreover, Niu et al. investigated the instability of TTA and found that the batch norm layer is a crucial factor. They further proposed removing partial noisy samples with large gradients and encouraging model weights to approach a flat minimum [44].

Although TTA has many advantages, Sun et al. noted that small unlabeled batches of shifted data can potentially be accessed just before prediction time [29]. This observation leads to a simple yet effective method called Test-time batch normalization (BN) to leverage test batch statistics instead of training statistics. BNadapt and α-BN [45, 46] correct the BN statistics for reducing covariate shift. Based on this, Su et al. [47] found that the inexact target statistics largely stem from the substantially reduced class diversity in batches and also introduced the test-time exponential moving average (TEMA) method to bridge the class diversity gap between training and testing batches. However, these methods all focus only on the model's capability of generalization to a specific task, such as classification or segmentation. Also, certain noisy test samples can still disturb model adaptation and result in collapsed trivial solutions [44]. To address these issues, we propose a method to fine-tune the BN layer of the source encoding module, with the objective of Entropy Minimization. We further select partial samples with different levels of entropy values and optimize the model for confidence. This strategy thus adapts the acquired knowledge from the source domain to the target domain, thereby enhancing the model's generalization capability that encompasses the whole target domain. The results are achieved in one epoch of test-time optimization without altering the training process. Moreover, in contrast with the aforementioned work, our proposed model is capable of predicting object counting, localization, and size estimation simultaneously. To the best of our knowledge, we are the first to attempt to design TTA strategies for heatmap regression.

Note: In this paper, similar to [4], we do not estimate the actual diameter, length, or volume of the object. Still, they can be derived by multiplying a constant factor (can be obtained

by measuring a reference real-object in an image, also this value depends on camera distance to the objects and camera focal lengths, and these parameters are adjustable in our datasets generating framework; for a real dataset from the field in smart agriculture, these two corresponding parameters can be known from the distance between the smart sensor and the tree and the camera focal lengths of the smart sensor) with the estimated sizes [48]. In other words, we focus on estimating the relative scale of an object and refer to this relative scale as object size, without causing ambiguity. According to the object's relative size, the weight of this object can be obtained by comparing it to a reference object from the same class.

A TTA-Based Approach for Multi-Class Object Localization and Size Estimation

In this section, we introduce our proposed deep learning-based framework for extracting information about product size and quantity in the product bundle image with the task of the OSR. Figure 2 depicts the diagram of the proposed TTA-based approach for multi-class Object Localization and Size Estimation problems.

We use the following notations to express the testing-time adaptation settings. Let $F_\theta(x)$ denote a model trained on seen domain images $D^{Seen} = \{(x_i, y_i^s, y_i^c)\}_{i=1}^N$ with location and size labels, where θ is the model parameter. The goal of test-time adaptation is to enhance the performance of $F_\theta(x)$ on unseen OoD test samples $D^{Unseen} = \{x_j\}_{j=1}^M$ without labels. To introduce novel tasks and data, we propose a deep model with high-resolution net (HRNet) [29] as the backbone. However, at the training phase, the model is trained exclusively on the seen data with labels and does not implement any domain adaptation pipeline. Entropy minimization and Batch Normalization operations are solely executed during the testing phase.

Backbone Model

In this work, we treat the problem of Localization and Size Estimation as regression. The architecture is illustrated in Fig. 3. HRNet [29] serves as the backbone architecture, which is a general-purpose convolutional neural network for tasks like semantic segmentation, object detection, and image classification, and could be modified and utilized to achieve our tasks. The pre-trained HRNet is used to extract multi-resolution feature maps $h(x)$, shown in Fig. 3. In addition, the repeated multi-scale fusion pipeline is employed to enhance high-resolution representations, thereby improving the accuracy of the predicted heatmaps. As we aim to estimate the object's location and size jointly, we have designed two heads accordingly. We regress the heatmaps simply from the output of HRNet concatenated representations. Two loss functions, defined as the mean squared error L^c_{MSE}, L^s_{MSE}, are compared with the predicted heatmaps $\hat{y}^c$ and $\hat{y}^s$ with their corresponding ground truth heatmaps y^c and y^s, respectively. Note that c and s represent the tasks of position detection and size estimation, respectively. The ground truth heatmaps for object location y^c are generated by applying a 2D Gaussian:

$$y^c \sim N(\mu^c, \sigma^c), \quad \mu^c = object\ coordinates, \quad \sigma^c \propto \frac{object\ size}{\xi}, \tag{1}$$

with the standard deviation being proportional to the size of the object ($\xi = 3$ empirically) and centered at the object's location. Similarly, the ground-truth heatmaps for object size y^s are generated by:

$$y^s \sim N(\mu^s, \sigma^s), \quad \mu^s = object\ coordinates, \quad \sigma^s = object\ size. \tag{2}$$

Since the location and quantity of objects are positively correlated, our approach does not involve the case of multiple overlapping layers. Accordingly, the number of objects can be obtained by counting the peaks of their respective location maps, called peak maps.

As a result, we obtain a compatible model that serves as the adaptation target and must undergo supervised training while ensuring differentiability. This is because during testing, we need to optimize the model in an unsupervised manner. Therefore, the model has to be trained while keeping differentiability in mind to ensure stable adaptation and optimization.

TTA Based Approach

During the training phase, our proposed model $F_\theta(x^s)$ is trained with normalized heatmaps y^c and y^s to regress the location and size of an image x^s. Once a trained model $F_\theta(x^s)$ with parameters θ is obtained, we focus solely on the unlabeled test set X^t during the test phase, which can include OoD data (as shown in Fig. 3). Inspired by [7], we use test entropy as the test-time objective. Since our model output is a heatmap, we need to apply $p(\hat{y}) = Softmax(\hat{y} > 0)$ to the output to obtain probabilities.

During testing, we optimize our model by minimizing the predicted entropy by adjusting its BN layer. Specifically, the target mini-batch input $X_t^{(i)}$ is used to passively calculate the corresponding target statistics $\mu_t^{(i)} = \mathbb{E}[X^t]$, $\sigma_t^{2(i)} = \mathbb{E}[(\mu_t^{(i)} - X_t^{(i)})^2]$, and mix both source and target statistics before forwarding i^{th} BN layer:

$$\mu^{(i)} = \alpha\mu_s^{(i)} + (1-\alpha)\mu_t^{(i)}, \tag{3}$$

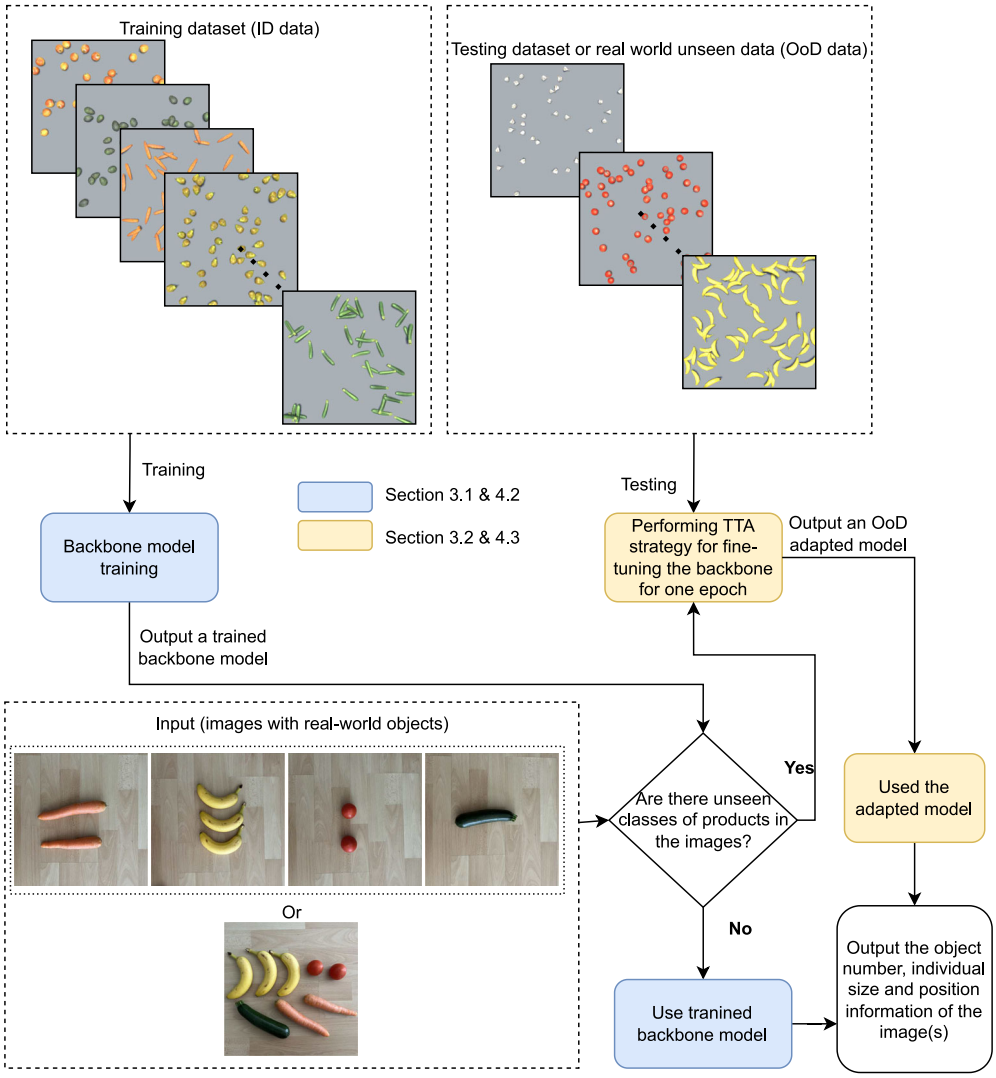

Fig. 2 Diagram of the proposed OSR approach for multi-class object counting, position detection, and size estimation

$$\sigma^{i} = \alpha\sigma_{s}^{(i)} + (1-\alpha)\sigma_{t}^{(i)}, \tag{4}$$

$$y^{(i)} = \gamma^{(i)}\frac{x^{(i)} - \mu^{(i)}}{\sigma^{(i)} + \beta^{(i)}}, \tag{5}$$

where μ_s and σ_s are source statistics, and α is a hyper-parameter to alleviate the estimated error caused by the small batch size. This process normalizes the input data $x^{(i)}$ using the mixed mean and standard deviation, and then applies an affine transformation using scale parameter γ and shift parameter β to produce the corresponding output $y^{(i)}$. The statistics $(\mu_t^{(i)}, \sigma_t^{2(i)})$ are calibrated from the target data, while the affine parameters $\gamma^{(i)}$, $\beta^{(i)}$ in BN layers are optimized with the loss function:

$$\mathcal{L}_H = -\sum_{n=1}^{H\times W} p(\hat{y}_n)\log p(\hat{y}_n). \tag{6}$$

This approach is fully self-supervised, and it updates the parameters for all BN layers only once for each test batch. Pseudo-labeling tunes a confidence threshold, assigns predictions over the threshold as labels, and then optimizes the model to these pseudo-labels before testing. For unlabeled target data, we calculate the entropy of the predicted heatmap and divide the samples by different entropy levels based on a set threshold value. Pseudo-label sampling method makes use of stratified samples and associated predictions as self-supervised samples for tuning BN layers, resulting in further optimization of testing performance.

Experiments for TTA-Based Approach in OSR

In this section, our experiments for the TTA-based approach (described in the "A TTA-Based Approach for Multi-Class Object Localization and Size Estimation" section) are pre-

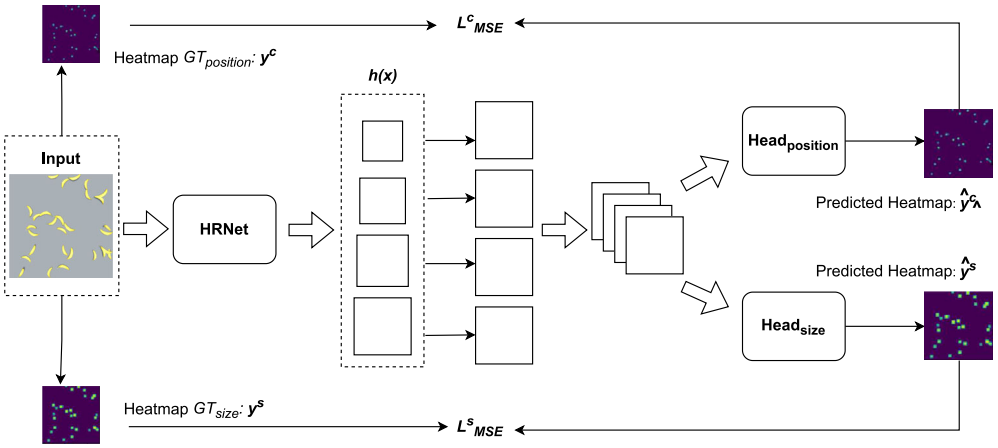

Fig. 3 The framework of the proposed model to multi-class object counting, position detection, and size estimation

sented. This evaluates the performance of our approach for extracting information on fruit/vegetable size and quantity from images in the OSR. The results prove that our approach can work as an enabling technology in Smart Agriculture. In the following, we first show the experiment of how we train a deep learning-based backbone model for multi-class object localization, size estimation, and counting problems. The optimal strategies and parameter settings of the proposed TTA-based approach are evaluated and discussed for tasks under different scenarios. Lastly, we test our proposed models, trained on the synthetic datasets, on the collected real image dataset. A comparative experiment is further conducted between our model and SAM on the real-image dataset. All experiments are implemented on an Nvidia Volta V100 GPU, and our source code and data are publicly available (link is provided in Appendix 1).

Evaluation Metrics and Experiment Dataset

We describe the evaluation metrics and data used for the three tasks shown in Fig. 1 (object counting, position detection, and size estimation) in our experiment.

From the aspect of localization and size estimation, we use root mean square error (RMSE) to evaluate the detected position and size estimation errors of each object. The reason for using the RMSE in our experiments is that the distance between the object's ground-truth position and the predicted one is naturally calculated by the root mean square. The RMSE for object size estimation is calculated over all object instances in all the test images.

$$RMSE_{size} = \sqrt{\frac{1}{\sum_{i=1}^{N_t} n_i} \sum_{i=1}^{N_t} \sum_{j=1}^{n_i} (y_{ij}^s - \hat{y}_{ij}^s)^2} \tag{7}$$

$$RMSE_{distance} = \sqrt{\frac{1}{\sum_{i=1}^{N_t} n_i} \sum_{i=1}^{N_t} \sum_{j=1}^{n_i} ((x_{ij}^c - \hat{x}_{ij}^c)^2 + (y_{ij}^c - \hat{y}_{ij}^c)^2)} \tag{8}$$

where y_{ij}^s and $\hat{y}_{ij}^s$ are the ground truth and predicted sizes; x_{ij}^c (y_{ij}^c) and $\hat{x}_{ij}^c$ ($\hat{y}_{ij}^c$) are the ground truth and predicted position, in the x-axis (y-axis) of the j-th object instance in the i-th test image.

For object counting, we also use RMSE to evaluate the performance:

$$RMSE_{count} = \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} (y_i^{peak} - \hat{y}_i^{peak})^2} \tag{9}$$

where y_i^{peak} and $\hat{y}_i^{peak}$ are the ground truth and predicted peak map for the i-th test image respectively, N_t is the number of test images.

The **synthetic part** of our E-MOCSE13 dataset for the unified problem of object counting and size estimation in OSR has the following settings:

- 390 images in total, and 30 images per class, with a resolution of 512*512. The ground truth information for each image is provided in a text file, which includes the total amount, label, relative size, and position of each object.
- 13 classes of fruits or vegetables: apple, artichoke, avocado, banana, carrot, courgette, garlic, melon, onion, orange, pear, tomato, and shallot. Based on the shapes of these objects, the banana, tomato, and garlic are selected as the OoD dataset used in the experiment. The rest of the classes are set as the in-distribution (ID) dataset.
- Each class in our dataset contains 6 groups, which are combinations of counts and sizes in different ranges.

Table 2 Detail of six groups of images in the dataset

	Small amount	Large amount
Small size	Count range: 30–70; Size range: 1.8–2.4	Count range: 70–110; Size range: 1.8–2.4
Medium size	Count range: 30–70; Size range: 2.4–3.0	Count range: 70–110; Size range: 2.4–3.0
Large size	Count range: 30–70; Size range: 3.0–3.6	Count range: 70–110; Size range: 3.0–3.6

Table 2 gives the details of all groups. Each group has 5 images, three of which are used as the **training-set** in the ID dataset, and the rest of the two images of each group are used as the **validation-set**. The OoD dataset is used as the **test-dataset** in our experiment. We also select two images from each group for each class in OoD to build a **smaller-test-dataset**. The mixture of ID and OoD can be seen as the noises added to create a joint distribution, so that it increases the complexity of the OSR to test the generalization performance.

The **real part** of E-MOCSE13 dataset is created from Google Images and COCO dataset [49]. Figure 4 shows the thumbnails of them. They contain adjacent or non-adjacent or occluded *apples*, *oranges*, and *tomatos* of different sizes and numbers. The settings of this real image dataset are listed as follows:

- It contains 3 classes (*apple*, *orange*, and *tomato*) and 11, 7, and 5 images for each class correspondingly, in which *apple* and *orange* are set as ID classes and *tomato* is set as OoD class in our synthetic dataset.
- To increase the complexity of counting tasks, we duplicate and combine these source images to make each of our test images contain 15 to 30 objects. The labels and positions of all objects in our real image dataset are hand-annotated.
- To get the relative size of each object, we first assume the parameters of the camera distance to the objects and camera focal lengths for all source images are the same as those in our synthetic dataset. Under this assumption, a ratio exists for each class that measures the number of pixels per given scale in the source images of this class [4]. This ratio is equivalent to the ratio of images of this class in our synthetic dataset. This ratio for each class can be calculated based on the average total pixel number and the average relative size of all objects in any image of this class within our synthetic dataset. Then, we preprocess each source image to count the total pixel number for each object within it (the details are shown in Appendix 2) and, based on the corresponding class ratio, determine its relative size.

The Experiment of Backbone Model Training

In this section, we depict our experiment on how to train a backbone model to output the three tasks (object counting, localization, and size estimation) simultaneously.

Experiment Setting

The experiment setting of the backbone model training is discussed. A pre-trained HRNet [29] is implemented in PyTorch [50] in this experiment. The Adam optimizer is employed for training. The initial learning rate (LR) is set to 1e-3, the minimum LR is set to 5e-7, and the cosine annealing schedule is employed during the training process. The learning rate in this schedule starts high and decreases rapidly to a minimum value near zero before increasing again to the maximum, which is a typical aggressive learning rate schedule [51]. The batch size is set to 16. Since the model can always converge after around 400 epochs, we set the total number of training epochs to 500. For every 10 epochs after 150 epochs, we set it as the checkpoint to save the trained model. In the experiment, we use the whole training set of the ID dataset

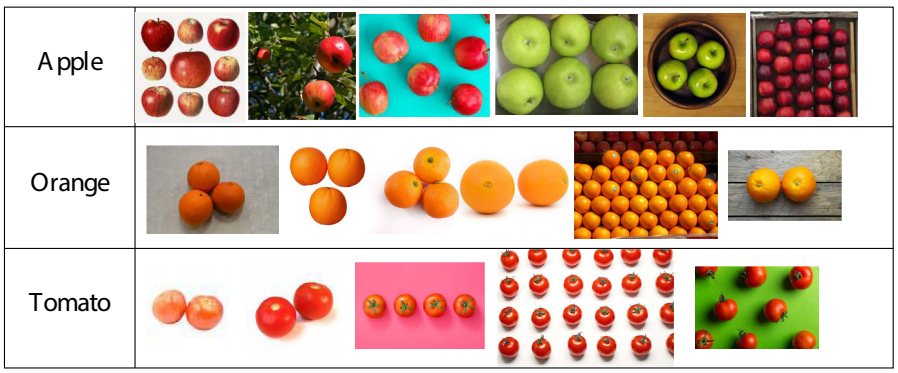

Fig. 4 Collected sources images (thumbnails)

described before to train a model to get more prior knowledge for different kinds of fruits/vegetables.

According to the proposed TTA strategy, the training process aims to generate a compatible model that can achieve strong OoD generalization, specifically for examples drawn from distributions that differ from the training set. The output position heatmap and size heatmap have a size of 128*128, which is 1/4 of the input image. The method we use to adapt the ground truth position heatmap and size heatmap with the prediction output during the training is the loss function we used "MSEloss", they all have the same size. For testing, a position heatmap, a size heatmap, and a peak map of the test figure could be output by feeding the whole image to the trained model. The peak map is used to count the objects in the image by aggregating all the pixel values. We use the function "MaxPool2d" from PyTorch to get the peak map from the output heatmap. Through segmentation, the individual-identified object with its position and size can be obtained. Figure 5 shows an example of the output heatmap, size heatmap, and peak map.

Experiment Training Result

During the experiment, a training backbone model is saved as a checkpoint every 10 epochs. The aims of the experiment in this section are (1) validate these checkpoint models by using the **validation-set**; (2) examine the OoD generation on the unseen classes of data by using the **smaller-test-dataset**.

Figure 6 shows the experiment results of the training model from all checkpoints. The upper left subfigure demonstrates the change in the counting RMSE of the trained model with the increased number of epochs in the training process. Under two different inputs of the **validation-set** or the **smaller-test-set**, both the RMSE decrease with the increased number of epochs and finally converge after 400 epochs. The best RMSE under the **validation-set** and the **smaller-test-set** is 3.121 and 4.381 at 400 epochs, respectively. Since the range of object numbers in images of the dataset is 30 to 110 (listed in Table 2), 3.121 (4.381) is a relatively small number compared with numbers in this range, with the counting error at most 10.4% (14.6%) in this scenario. All these mean that the training model can get enough prior knowledge through feature representation learning, and proves it has good OoD generalizations on the object counting task.

The upper right subfigure in Fig. 6 shows the changes in distance RMSE for each object of the **validation-set** and the **smaller-test-dataset** with the increased number of epochs. Unlike the decreasing trend of the counting RMSE, the distance RMSE for each object fluctuated with values of 10 or 4 for both input datasets, respectively. Compared to the size of the heatmap 128*128, the distance error on the object detection task is relatively small, around 7.81% or 3.1%. The fluctuation of the distance RMSE suggests that the model has converged after 150 training epochs. Increasing the number of epochs does not have a positive effect on the results of the localization task. Furthermore, the distance RMSE in the OoD dataset is almost 2.5 times higher than in the ID dataset. All these mean the OoD generalizations of the trained model for the task of object detection are acceptable but could be further explored to improve.

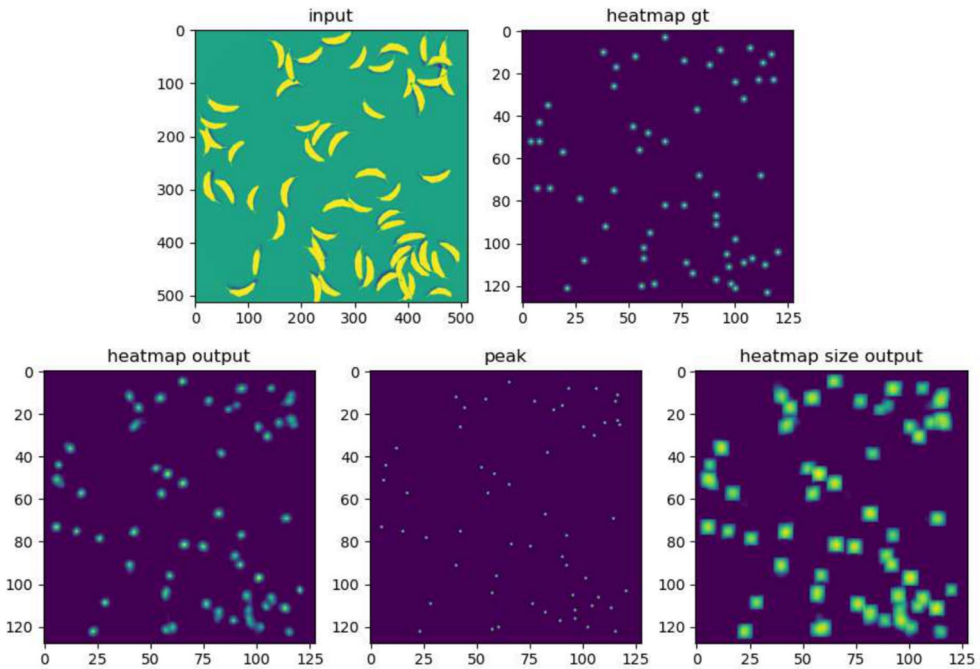

Fig. 5 The example output results of banana (unseen class of data during the training) for one of its test images: input image (upper left), heatmap ground truth (upper right), output heatmap (lower left), output peak map (lower middle), and output size heatmap (lower right)

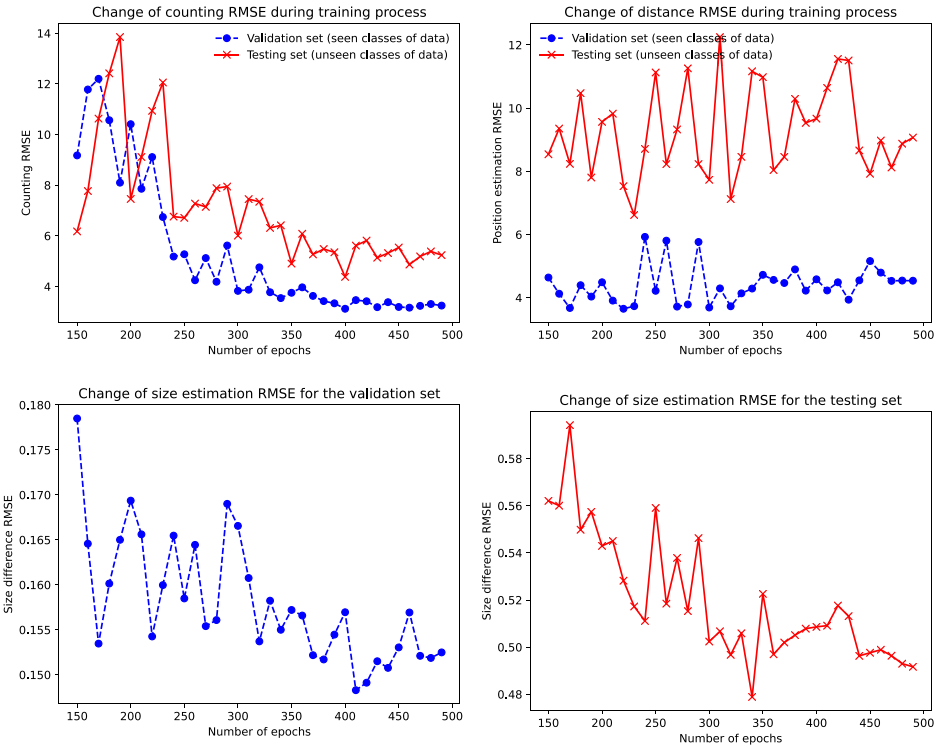

Fig. 6 Validation is performed for trained models at different checkpoints, with a focus on changes in counting RMSE (upper left), distance RMSE (upper right), and size estimation RMSE (lower left and right)

The lower two subfigures in Fig. 6 depict the decreasing trend of size estimation RMSE for each object under the input of the **validation-set** or the **smaller-test-dataset** with the increasing number of epochs, which is similar to the results of the object counting task. From the figure, we can see that the test results converge after 450 epochs with values around 0.15 and 0.52. From Table 2, we observe that the objects in the test images have a size range of 1.8 to 3.6. The RMSE values of 0.15 and 0.52 (which denote prediction errors of at most 8.33% and 28.8%, respectively) are relatively small compared to this range. Therefore, we can conclude that the training model performs well in terms of size estimation tasks under the ID dataset. However, the RMSE value for size estimation in the OoD dataset is higher than that of the ID dataset, indicating the potential for improvement via the incorporation of the TTA strategy.

In this experiment, we test the performance of the trained backbone model under the **validation-set** and the **smaller-test-dataset** on tasks of object counting, localization, and size estimation. It performs well for all tasks under the ID dataset. However, the performance of the trained model on the tasks of object localization and size estimation under the OoD dataset could be further improved, which is implemented in the next section.

Note: For the scenario of multiple classes of objects that exist in the same figure, we slightly modify the framework by adding a classification head in Fig. 3 to output multiple heatmaps/size heatmaps (each for one class), while keeping our backbone of the deep model backbone for feature representation learning. A similar approach has been achieved and implemented in our previous work [13]. Since our experiments mainly focus on object counting, localization, and size estimation for OoD Generalization issues, and the application scenario is smart agriculture (normally one type of fruit/vegetable exists in one crop/tree), we will not present or discuss the corresponding experiment results for this scenario.

The Experiment of Fine-Tuning the Backbone Models

In this section, we experiment with different strategies and parameter settings under TTA-primal and TTA-ft strategies to improve our backbone predictive model when training and test data with different distributions.

Experiment Setting

We use the trained models from the "The Experiment of Backbone Model Training" section under checkpoints of epochs 350 and 380 as the backbone model to test TTA-primal and TTA-ft domain adaptation strategies. TTA-primal calibrates the statistics $(\mu_t^{(i)}, \sigma_t^{2(i)})$ for all BN layers, utiliz-

ing test data. Notably, it does not update any of the affine parameters in the BN layers, thereby negating the need to fine-tune the model. However, the use of Pseudo-label sampling can be toggled to update these statistics for data exhibiting 'Top k minimum' entropy. In TTA-ft, the aim is to optimize the affine parameters $\gamma^{(i)}$ and $\beta^{(i)}$ in BN layers with a loss function by fine-tuning the model. The statistics calibration feature is an optional choice, as is the option to incorporate Pseudo-label sampling for selecting Top k minimum or Top k maximum entropy data for fine-tuning.

The testing OoD datasets are the **smaller-test-dataset** and **test-dataset** described in the "Evaluation Metrics and Experiment Dataset" section. To adapt different domains or distributions during testing, entropy loss is used to minimize the entropy of model predictions. To minimize entropy, our approach normalizes and transforms inference on target data by estimating statistics and optimizing affine parameters batch-by-batch. The times of affine transformation for both TTA-primal and TTA-ft during the testing can be set as 1 or 2. Specifically, TTA-ft will update the affine parameters during the testing with a learning rate. The optimal learning rate needs to be tested, and the batch size is set to 16 (the same as the backbone model training). During testing, we select the top k percentage of samples with the maximum or minimum entropy loss values to do the adaptation for the model. The possible value of k could be 10, 20, 50, and 100. The combinations of the parameter settings mentioned above are listed in Table 3, where *BS* is short for batch size and '-' means the parameter is not available in this setting. The test results of trained models from epochs 350 and 380 on two test datasets under these parameter settings are shown in the "Experiment Result" section. All the results shown in the following section are the mean values of experimental results from multiple runs (2–3 times). The reason for this is to avoid possible bias. The results from the different runs are quite close, and the variance in the results is small. The potential reason is that the DL-based models and TTA we implemented are well-designed, robust, and confident enough to make similar decisions in every run.

Experiment Result

The experiment results of the fine-tuned models under different combinations of settings are shown and discussed. Due to the space limitation, Tables 4, 5, 6, and 7 show the results fine-tuned on the checkpoint 380 backbone model. The parameter setting with the best $RMSE_{distance}$, $RMSE_{size}$ or $RMSE_{count}$ in each block of the table is in **bold**, and the second best is underscored *italic*.

Table 4 lists the detailed results of different LR settings for adopting the TTA strategy on two different testing datasets under sample selection ratio 1 and batch size 16. The test results on two datasets from the original backbone model (from checkpoint epochs 380) are also listed in the first row for each block as a comparison. From the table, we can see the best $RMSE_{distance}$ on the **smaller-test-dataset** is lower from 10.3 to 6.0 under the TTA-ft approach with setting LR to 5e-3 and switching off statistics calibration, which is quite close to 4.90 - $RMSE_{distance}$ of the original backbone model testing on ID dataset (**validation-set**). The results $RMSE_{size}$ of different settings are quite close to the original backbone model, around 0.51. It may suggest further testing in a larger range of parameter settings. This similar result of size may indicate that our synthetic images may lack the reflection towards the real-world heterogeneity of the fruit size. The best $RMSE_{count}$ from the table is 3.29, which is improved compared with the $RMSE_{count}$ of the original backbone model testing on the same OoD dataset 5.48. Except for the best values, from Table 4, we can find that different setting strategies may bring better performance for the different tasks under this groups of parameter settings: i.e., switching on the statistics calibration can improve models' performance on the task of position detection, but switching off it can get better counting results. Moreover, the results indicate that fine-tuning all BN layers of the model with testing data is advantageous in achieving domain generalization.

Based on the number of images in two test datasets and the results from Table 4, for the following experiments, we set the LRs used to update the affine under the TTA-FT approach

Table 3 The combinations of parameter settings

Methods	Max or Min entropy	Sample selection ratio (k)	Statistics calibration ($\mu_t^{(i)}$, $\sigma_t^{2(i)}$)	LR to optimize affine	Affine transformation times
TTA-ft	–	1 (BS= 16)	True, false	1e-2, 1e-3,1e-4,1e-5, 5e-3,5e-4	–
	–	1	True, false	1e-3	1, 2
	Top k minimum	0.1, 0.2, 0.5	True	1e-3	1, 2
	Top k minimum	0.1, 0.2, 0.5	False	1e-3	1, 2
	Top k maximum	0.1, 0.2, 0.5	True	1e-3	1, 2
	Top k maximum	0.1, 0.2, 0.5	False	1e-3	1, 2
TTA-primal	–	1	True	–	1, 2
	Top k minimum	0.1, 0.2, 0.5	True	–	1, 2

Table 4 Experiment results of different LR under batch size 16

Test dataset	Fine-tune approach	Sample selection ratio (k)	Statistics Calibration	LR to update affine	$RMSE_{distance}\downarrow$	$RMSE_{size}\downarrow$	$RMSE_{count}\downarrow$
Smaller test dataset	–	–	–	–	10.30	*0.51*	5.48
	TTA-ft	1 (BS= 16)	False	1e-2	9.44	0.58	**3.29**
	TTA-ft	1 (BS= 16)	False	5e-3	*6.42*	0.52	*3.56*
	TTA-ft	1 (BS= 16)	False	1e-3	7.24	*0.51*	4.30
	TTA-ft	1 (BS= 16)	False	5e-4	6.80	*0.51*	4.76
	TTA-ft	1 (BS= 16)	False	1e-4	10.29	*0.51*	5.27
	TTA-ft	1 (BS= 16)	False	1e-5	10.30	*0.51*	5.45
	TTA-ft	1 (BS= 16)	True	1e-2	8.59	0.52	10.18
	TTA-ft	1 (BS= 16)	True	5e-3	**6.00**	**0.50**	7.20
	TTA-ft	1 (BS= 16)	True	1e-3	8.05	*0.51*	5.96
	TTA-ft	1 (BS= 16)	True	5e-4	8.08	*0.51*	5.57
	TTA-ft	1 (BS= 16)	True	1e-4	7.72	**0.50**	5.50
	TTA-ft	1 (BS= 16)	True	1e-5	7.75	*0.51*	5.45
	TTA-primal	1 (BS= 16)	True	–	7.75	*0.51*	5.50
Test dataset	–	–	–	–	9.15	0.46	5.59
	TTA-ft	1 (BS= 16)	False	1e-2	9.20	0.55	8.71
	TTA-ft	1 (BS= 16)	False	5e-3	11.76	0.56	12.31
	TTA-ft	1 (BS= 16)	False	1e-3	9.80	0.47	**4.20**
	TTA-ft	1 (BS= 16)	False	5e-4	9.68	*0.46*	*4.48*
	TTA-ft	1 (BS= 16)	False	1e-4	8.80	**0.45**	5.43
	TTA-ft	1 (BS= 16)	False	1e-5	9.15	*0.46*	5.59
	TTA-ft	1 (BS= 16)	True	1e-2	12.79	0.60	9.22
	TTA-ft	1 (BS= 16)	True	5e-3	8.66	0.53	6.91
	TTA-ft	1 (BS= 16)	True	1e-3	*8.44*	0.47	5.72
	TTA-ft	1 (BS= 16)	True	5e-4	**8.35**	*0.46*	5.66
	TTA-ft	1 (BS= 16)	True	1e-4	8.46	0.47	5.77
	TTA-ft	1 (BS= 16)	True	1e-5	8.47	0.47	5.76
	TTA-primal	1 (BS= 16)	True	–	8.47	0.47	5.79

as $1e-3$ when testing the **smaller-test-dataset** and $5e-4$ when testing the **test-dataset**. Table 5 lists the results of different parameter combinations of maximum or minimum loss value, different sample selection ratios, and affine transformation times under the TTA-ft approach. In this experiment, we set the statistics calibration to false for all runs. Based on the best and second-best values of each metric from the table, the best strategies under this group can be found: select the top 50% (10%) of samples with the maximum (minimum) entropy loss values to do the adaptation for the task of object position detection (object counting). Compared with the results in Table 4, we can conduct the following truths: (1) both of the best values of $RMSE_{distance}$ and $RMSE_{count}$ in this table are lower than the best values in Table 4 respectively; (2) same with the previous group of parameter settings, the performance of the fine-tuned backbone model under this group of parameters is also not improved on the task of object size estimation.

Upon enabling statistics calibration, we conducted the same experiment for the aforementioned parameter combinations as listed in Table 5, and recorded their respective results in Table 6. Interestingly, the best strategies within this group are reversed compared to the previous group: select the top 50% (50%) of samples with the minimum (maximum) entropy loss values to do the adaptation for the task of object position detection (object counting). The best results on metrics $RMSE_{distance}$ are lower than the best values in Table 4 when using the full test dataset, which indicates that the confidence level of pseudo-labels decreases with an increasing number of test samples. In this scenario, enabling statistics calibration would be beneficial. However, concerning the optimal result for the $RMSE_{count}$ metric, it is noted that while the predicted coordinates may experience some deviation, the counting threshold is relatively lenient. Thus, incorporating more test data would have a positive effect on its performance, as demonstrated

Table 5 Testing results of different parameter combinations of maximum or minimum loss value, different sample selection ratios, and affine transformation times on the TTA-ft approach(1)

Test dataset and general settings	Min or Max Entropy	Sample selection ratio (k)	Affine transformation times	$RMSE_{distance}\downarrow$	$RMSE_{size}\downarrow$	$RMSE_{count}\downarrow$
Smaller test dataset, LR=**1e-3**, Statistics Calibration: **True**	Top k min	0.1	1	7.87	*0.51*	*5.03*
	Top k min	0.1	2	8.77	*0.51*	**4.70**
	Top k min	0.2	1	9.21	*0.51*	5.46
	Top k min	0.2	2	9.19	*0.51*	5.45
	Top k min	0.5	1	10.34	**0.50**	5.81
	Top k min	0.5	2	9.26	*0.51*	5.74
	–	1	1	10.29	*0.51*	5.61
	–	1	2	9.12	*0.51*	5.63
	Top k max	0.1	1	9.13	*0.51*	5.77
	Top k max	0.1	2	9.14	*0.51*	5.70
	Top k max	0.2	1	10.20	**0.50**	5.56
	Top k max	0.2	2	9.00	*0.51*	5.64
	Top k max	0.5	1	**7.71**	*0.51*	5.76
	Top k max	0.5	2	*7.70*	0.52	6.04
Test dataset, LR=**5e-4**, statistics calibration: **True**	Top k min	0.1	1	*8.09*	*0.46*	*5.16*
	Top k min	0.1	2	8.73	0.47	**4.97**
	Top k min	0.2	1	8.68	*0.46*	5.37
	Top k min	0.2	2	8.93	*0.46*	5.21
	Top k min	0.5	1	9.06	**0.45**	5.75
	Top k min	0.5	2	8.61	**0.45**	5.61
	–	1	1	8.92	*0.46*	5.67
	–	1	2	8.52	*0.46*	5.61
	Top k max	0.1	1	8.42	*0.46*	5.75
	Top k max	0.1	2	8.54	0.47	5.69
	Top k max	0.2	1	8.87	**0.45**	5.72
	Top k max	0.2	2	8.27	*0.46*	5.87
	Top k max	0.5	1	**8.05**	0.46	5.71
	Top k max	0.5	2	*8.09*	0.46	5.85

by the discrepancy compared to the optimal values shown in Table 4.

Table 7 lists the testing results from the TTA-primal approach with different combinations of sample selection ratios and affine transformation times. The best strategies for the object position detection and counting tasks are the same: select the top 10% of samples with the minimum entropy loss values for adaptation. But still, the best results on metrics $RMSE_{distance}$ and $RMSE_{count}$ are lower than the best values in Table 4 on these two metrics, respectively. This outcome aligns with our expectations.

In this section, we extensively evaluated the performance of fine-tuned models using varied parameter settings. When using OoD data, our TTA-based approach demonstrated noteworthy enhancement in both object localization and counting tasks. The metric of $RMSE_{size}$ remained relatively constant, potentially attributed to the greater complexity involved in assessing size features of novel fruits and vegetables, particularly under limited testing data. Our way of finding the best strategy on the TTA approach to fine-tune the model for domain adaptation under different target tasks can be a reference for future similar work. It is concluded as follows.

Parameter Tuning Approach In scenarios with a smaller dataset (such as **smaller-test-dataset** in this paper), we prefer to use a batch size of 16, updating only the affine parameters with a learning rate between 5e-3 and 5e-4. This approach

Table 6 Testing results of different parameter combinations of maximum or minimum loss value, different sample selection ratios, and affine transformation times on the TTA-ft approach(2)

Test dataset and general settings	Min or Max Entropy	Sample selection ratio (k)	Affine transformation times	$RMSE_{distance}\downarrow$	$RMSE_{size}\downarrow$	$RMSE_{count}\downarrow$
Smaller test dataset, LR=**1e-3**, statistics calibration: **False**	Top k min	0.1	1	9.04	0.51	5.10
	Top k min	0.1	2	8.90	0.51	4.96
	Top k min	0.2	1	**8.13**	0.51	5.22
	Top k min	0.2	2	8.94	0.51	4.90
	Top k min	0.5	1	**8.13**	0.51	5.07
	Top k min	0.5	2	*8.17*	0.52	4.71
	–	1	1	10.30	0.51	5.43
	–	1	2	9.14	**0.50**	5.35
	Top k max	0.1	1	10.26	0.51	4.89
	Top k max	0.1	2	10.28	0.51	*4.63*
	Top k max	0.2	1	10.30	0.51	4.84
	Top k max	0.2	2	8.94	0.52	4.90
	Top k max	0.5	1	9.00	0.51	4.83
	Top k max	0.5	2	9.06	0.51	**4.62**
Test dataset, LR=**5e-4**, statistics calibration: **False**	Top k min	0.1	1	8.94	0.46	5.44
	Top k min	0.1	2	8.26	0.46	5.34
	Top k min	0.2	1	**8.05**	0.46	5.52
	Top k min	0.2	2	8.46	0.46	5.38
	Top k min	0.5	1	**8.05**	0.46	5.40
	Top k min	0.5	2	*8.09*	0.46	5.40
	–	1	1	9.20	0.46	5.18
	–	1	2	8.89	**0.45**	**4.78**
	Top k max	0.1	1	9.66	0.46	5.29
	Top k max	0.1	2	9.17	0.46	5.03
	Top k max	0.2	1	9.14	0.46	5.34
	Top k max	0.2	2	9.56	0.46	5.06
	Top k max	0.5	1	9.29	0.46	5.23
	Top k max	0.5	2	8.63	0.46	*4.89*

ensures more accurate localization estimation on OoD data. This preference is due to the limited testing data, which results in a relatively minor covariance shift caused by the unseen set. Conversely, with a relatively larger dataset (like the full test dataset we used), selecting the maximum top samples yields better localization accuracy compared to the minimum top samples. Since maximum top samples bring significant covariance shifts, it is necessary to update both the statistics calibration and affine parameters. Regardless of the testing set size, the optimal counting performance is achieved when the batch size is set to 16 and only affine parameters are updated. This is because statistics calibration does not influence the counting process. Furthermore, since size regression is less dependent on the shape of the detected objects, the backbone model demonstrates the best generalizability for this task. Consequently, it is also less likely to be affected by statistics calibration.

The Experiment on the Real Image Dataset

In this section, we evaluate the performance of our model (trained from the synthetic dataset) and TTA fine-tuning strategies by using the collected **real image dataset** from the **E-MOCSE13**. The feasibility of our approach to real-world application scenarios is validated.

Experiment Results

We first test the trained backbone model obtained from the "The Experiment of Backbone Model Training" section by

Table 7 Testing results of different parameter combinations of different sample selection ratios and affine transformation times on the TTA-primal approach

Test dataset and general settings	Min or Max loss	Sample selection ratio (k)	Affine transformation times	$RMSE_{distance}\downarrow$	$RMSE_{size}\downarrow$	$RMSE_{count}\downarrow$
Smaller test set, statistics calibration: **True**	Top k min	0.1	1	**7.88**	0.51	*5.02*
	Top k min	0.1	2	*8.99*	0.52	**4.61**
	Top k min	0.2	1	9.30	0.51	5.55
	Top k min	0.2	2	9.26	0.51	5.38
	Top k min	0.5	1	9.12	0.51	5.44
	Top k min	0.5	2	9.28	0.51	5.38
	–	1	1	10.30	0.51	5.45
	–	1	2	9.14	**0.50**	5.35
Test dataset, statistics calibration: **True**	Top k min	0.1	1	**8.26**	0.46	*5.09*
	Top k min	0.1	2	8.90	0.47	**5.00**
	Top k min	0.2	1	8.69	0.46	5.23
	Top k min	0.2	2	8.70	0.47	5.17
	Top k min	0.5	1	9.06	0.46	5.53
	Top k min	0.5	2	8.69	0.45	5.42
	–	1	1	9.14	0.46	5.62
	–	1	2	*8.63*	0.45	5.46

using the two sets of images: *apple+orange* (ID) and *tomato* (OoD), respectively. In this experiment, we compare our models to the segment anything (SAM) [25]. A bounding box has to be offered to generate the segmentation mask from SAM. We include rough- and fine-bounding boxes to test the result, shown in Fig. 7. Specifically, the rough-bounding box covers a pile or a gathering of fruits, while the fine-bounding box is generated by each fruit's location to cover a single fruit. Therefore, we do not measure the distance for SAM as the location information is already used in the bounding box. Moreover, as we use the pre-trained SAM model in this comparison, we do not label the "ID" or "OoD" in the table.

For the results of all checkpoints, we only show the best results in the first two rows in Table 8 due to space limitations. For the ID test, our model achieves the best $RMSE_{distance}$ and $RMSE_{size}$ values: 5.22 and 0.17, respectively, at epoch 300. Our model with the checkpoint from epoch 350 gets the best $RMSE_{distance}$ of 7.06 and $RMSE_{size}$ of 0.43 on the OoD test. All these results are compatible with the best ones on the synthetic dataset. According to Fig. 6, the best $RMSE_{distance}$ scores for synthetic ID and OoD are 3.68 and 6.62, and the best $RMSE_{size}$ for ID and OoD are 0.14 and 0.47, respectively. In Table 8, the $RMSE_{counting}$ scores of 0 on OoD test data are obtained due to the relatively small number of objects contained in each image (15–30) of this dataset, compared to the synthetic dataset (30–110). The images of ID test data contain more objects (15–50) and have a large number of adjacent/partly-covered objects, the $RMSE_{count}$ shows as 2.57 is still much better than the best test corresponding results of the synthetic dataset in Fig. 6: 3.12. Another reason to affect the $RMSE_{count}$ of ID test data is that the range of object relative scales in it is 1.4–9.4, which was much wider than the range of object relative scales in the synthetic dataset 1.8–3.6 (shown in Table 2). The aforementioned results prove the capability for good generalization of our backbone model. The results of SAM show that the fine-bounding box offers a significantly advanced result compared to the rough-bounding box. The $RMSE_{size}$ of the fine-bounding box SAM remains inferior to that of our proposed models for both datasets. The $RMSE_{count}$ of the fine-bounding box SAM for the *apple+orange* dataset leads the backbone model by 0.64. On the other hand, the result of SAM for the *tomato* dataset is behind both the backbone model and the one with TTA-ft. The SAM model is designed as a generalized solution approach to segmentation. When handling images in our scenario, it may lack the capability of precise segmentation due to the fine-bounding box SAM. It requires a specific design of the neural networks to obtain precise information on the number, size, and location.

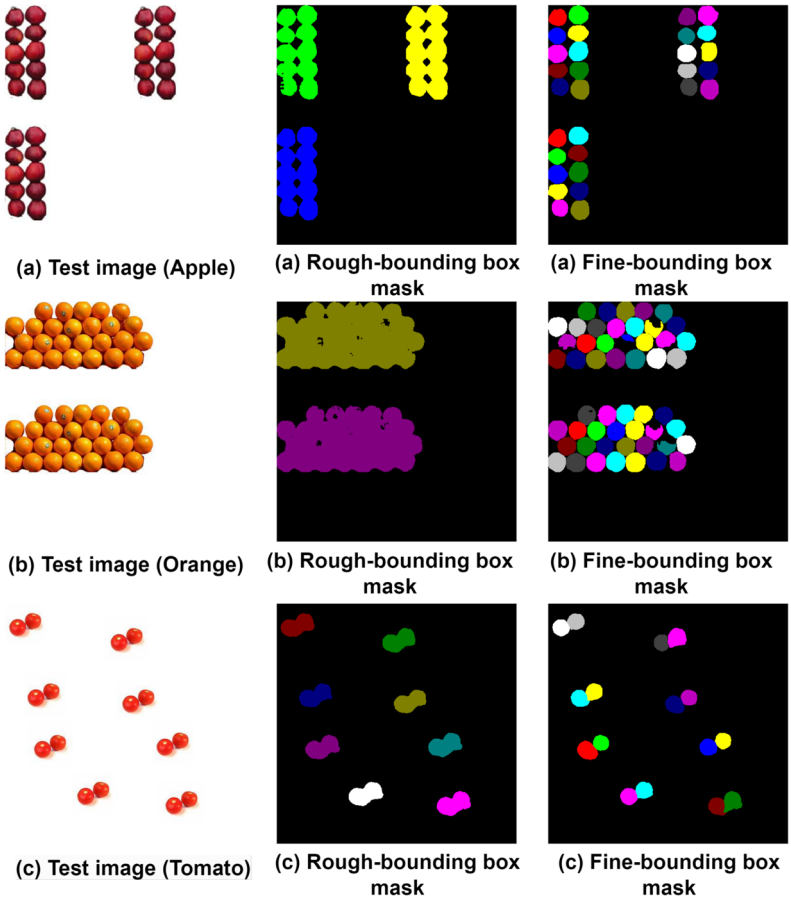

Fig. 7 SAM segmentation results on *apple+orange* and *tomato* dataset

Similar to the "The Experiment of Fine-Tuning the Backbone Models" section, we further experiment with different strategies and parameter settings under TTA-primal and TTA-ft for improving our backbone predictive model when tested on *tomato*. The best results, along with corresponding strategies and parameter settings, are listed in the third row in Table 8. The $RMSE_{distance}$ and $RMSE_{size}$ are improved by 0.16 and 0.02, respectively, which are improved compared with before. It is noticed that the $RMSE_{count}$ for the ID classes is 2.57 compared to the rest OoD results. The backbone model may fail to separate the pile fruit cases in the real image dataset, as shown in the apple and orange images in Fig. 7. It is a limitation that we can add the occlusion cases to further improve the E-MOSE13 dataset in the future. All these results show that useful prior knowledge can be gained when using our synthetic data in the experiment, since our dataset simulates the real-world scenario to keep the problem domain the same as the real application problem. This demonstrates the feasibility and high prediction accuracy of our approach, which is trained and tested using a synthetic dataset for real-world application scenarios without requiring further training.

Table 8 Experiment results on real image dataset

Test data	Model	Pareameters	$RMSE_{distance}\downarrow$	$RMSE_{size}\downarrow$	$RMSE_{count}\downarrow$
Apple+Orange (ID)	backbone	checkpoint of epochs 300	5.22	0.17	2.57
Tomato (OoD)	backbone	checkpoint of epochs 350	7.06	0.43	0.00
Tomato (OoD)	backbone+TTA-ft	Sample selection ratio 1, affine time 1, LR 0.001 (Table 5)	6.90	0.41	0.00
Apple+Orange	SAM	With rough-bounding box	–	5.19	22.76
Tomato	SAM	With rough-bounding box	–	3.82	7.60
Apple+Orange	SAM	With fine-bounding box	–	1.00	1.93
Tomato	SAM	With fine-bounding box	–	0.95	1.41

Conclusion

To capture the number, size, and location information of the individual fruit/vegetable from the images, this paper proposes a test-time domain adaptation approach based on deep learning for multi-class object counting, localization, and size estimation in an OSR environment. A new benchmark dataset (which includes a synthetic dataset generated from Unity 3D and a collected real image dataset) is used to train and test our approach. This dataset could be customized and further used as a testbed for future studies. Extensive experimental evaluations are also conducted on this dataset, and the high prediction accuracies for tasks such as object localization, size estimation, and counting in the OSR setting prove the feasibility and effectiveness of our approach in real-world application scenarios. Our framework is proposed for smart agriculture to automate the traditional farming process and help the farmer make decisions through growth monitoring and yield estimation. There are still some limitations regarding the generated E-MOSE13, such as the light and shade variation, etc. Such weaknesses can be further customized for wider and better usage, e.g., adding trees, crops, leaves, or other backgrounds to better simulate real scenarios of the farm field (we do not do this in this paper for the reason of framework and model parameter tuning approach testing purposes).

In our forthcoming research, we intend to emphasize more complex scenarios and settings, such as 3D object segmentation, counting, and size estimation, which rely on 3D point cloud datasets. To support these future scenarios, we plan to increase the variety of the E-MOCSE13, such as the variety of the fruit sizes. Additionally, to address the challenges associated with testing in domains involving both synthetic and real-world data, we aim to explore the use of fully test-time adaptive techniques, such as domain adaptation methods and real-time parameter adjustment of trained models. In this case, the performance and generalization of deep learning models will be improved.

Appendix 1. Source Code and Data Link

All experiments are implemented on an Nvidia Volta V100 GPU, and our source code and data are publicly available at: https://github.com/liulei1260/TTA-of-a-Multi-Class-Object-Localization-and-Size-Estimation-Framework

Appendix 2. Fruit Total Pixel Number Annotation for Background Removed Pictures

The preprocessing of real fruit images was implemented on MATLAB with the Image Processing Toolbox. For this study, the annotation process was led by authors, who coordinated two annotators from both Southampton and Eindhoven to systematically categorize and analyze the dataset. The total pixel number of each fruit in the image was calculated by partial automation processing and verified by the annotator. Afterward, the authors checked the annotation and the pixel numbers. Therefore, the annotation of the real fruit image represented the best knowledge from the annotator. The annotation process was performed as follows:

Figure 8 illustrates the annotation processing of an example image. After the background removal, only bare fruits were left within the given picture, shown in Fig. 8a. The orange segmentation could be achieved by thresholding in the YCbCr space. The color segmentation would generate a binary mask, shown in Fig. 8b. Afterwards, if there were several noises on the target objects, such as the stem or pedicle, a median filter was applied to remove the unexpected area. For the example case, there was no need to apply the filter.

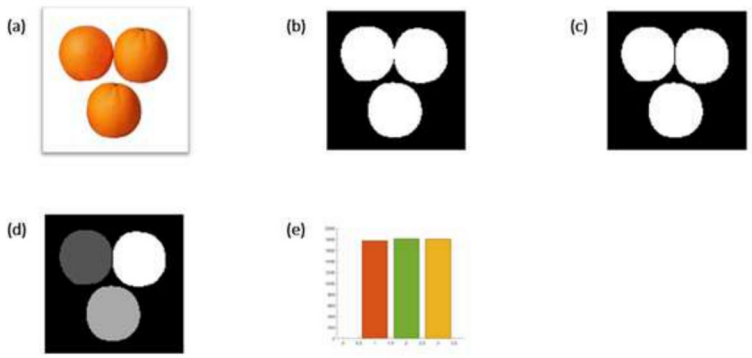

Fig. 8 The example of the pixel annotation for the background removed fruits. **a** The input orange image. **b** The binary mask after color segmentation on the YCbCr color space. **c** The binary mask after the watershed segmentation. **d** The mask for each individual orange. **e** The pixel histogram for each individual orange

With the connection of multiple objects (the top row oranges in Fig. 8a), the watershed algorithm was used to separate the objects, shown in Fig. 8c. By checking the completeness of each object's boundary and separating the objects, the separated orange mask was given, shown in Fig. 8d. The separated mask for each individual orange was accumulated, resulting in Fig. 8e. After the automated processing, the annotator checked the intermediate and final masks for each image. The annotated mask would offer the ground truth of each fruit. Therefore, the accurate pixel numbers for the real fruit images were generated.

Acknowledgements The authors would like to respect and thank all reviewers for their constructive and helpful review.

Author Contributions Z. L: Conceptualization, Methodology, Formal analysis, Coding, Visualization, Investigation, Data Curation, Writing - Original Draft. Q. W: Conceptualization, Methodology, Coding, Validation, Data Curation, Writing - Review & Editing. Y. C: Conceptualization, Methodology, Coding, Writing - Review & Editing, Supervision. H. Y: Writing - Review & Editing, Data Curation, Resource Management.

Funding This research received no specific grant from any funding agency in the public.

Data Availability The source code of our approach and the experiment image datasets used in this paper are publicly available: https://github.com/liulei1260/TTA-of-a-Multi-Class-Object-Localization-and-Size-Estimation-Framework

Declarations

Competing Interests The authors declare no competing interests.

References

1. Liu Z, Sampaio P, Pishchulov G, Mehandjiev N, Cisneros-Cabrera S, Schirrmann A, Jiru F, Bnouhanna N. The architectural design and implementation of a digital platform for industry 4.0 SME collaboration. Comput Ind. 2022;138:103623.
2. Shaikh TA, Mir WA, Rasool T, Sofi S. Machine learning for smart agriculture and precision farming: towards making the fields talk. Arch Comput Methods Eng. 2022;29(7):4557–97.
3. Rezaei EE, Webber H, Asseng S, Boote K, Durand JL, Ewert F, Martre P, MacCarthy DS. Climate change impacts on crop yields. Nat Rev Earth environ. 2023;4(12):831–46.
4. Apolo-Apolo O, Martínez-Guanter J, Egea G, Raja P, Pérez-Ruiz M. Deep learning techniques for estimation of the yield and size of citrus fruits using a UAV. Eur J Agron. 2020;115:126030.
5. Sun Y, Wang X, Liu Z, Miller J, Efros A, Hardt M. Test-time training with self-supervision for generalization under distribution shifts. In: International conference on machine learning. PMLR; 2020. p 9229–9248
6. Su Y, Xu X, Jia K. Revisiting realistic test-time training: sequential inference and adaptation by anchored clustering. In: Oh AH, Agarwal A, Belgrave D, Cho K editors. Advances in neural information processing systems; 2022
7. Wang D, Shelhamer E, Liu S, Olshausen B, Darrell T. Tent: fully test-time adaptation by entropy minimization. In: International conference on learning representations; 2021
8. Zhao P, Xue H, Ji X, Liu H, Han L. Zero-shot learning via visual feature enhancement and dual classifier learning for image recognition. Inf Sci. 2023;642:119161.
9. Geng C, Huang S-J, Chen S. Recent advances in open set recognition: a survey. IEEE Trans Pattern Anal Mach Intell. 2020;43(10):3614–31.
10. Zhu L, Li Z, Li C, Wu J, Yue J. High performance vegetable classification from images based on AlexNet deep learning model. Int J Agricu Biol Eng. 2018;11(4):217–23.
11. Sindagi VA, Patel VM. A survey of recent advances in CNN-based single image crowd counting and density estimation. Pattern Recogn Lett. 2018;107:3–16.
12. Murugaiyan S, Uyyala SR. Aspect-based sentiment analysis of customer speech data using deep convolutional neural network and BiLSTM. Cogn Comput. 2023;15(3):914–31.
13. Liu Z, Wang Q, Meng F. A benchmark for multi-class object counting and size estimation using deep convolutional neural networks. Eng Appl Artif Intell. 2022;116:105449.
14. Wang Q, Breckon TP. Crowd counting via segmentation guided attention networks and curriculum loss. IEEE Trans Intell Transp Syst. 2022;23(9):15233–43.
15. Chougule A, Bhardwaj A, Chamola V, Narang P. AGD-Net: attention-guided dense inception U-Net for single-image dehazing. Cogn Comput. 2024;16(2):788–801.
16. Zhang Y, Zhou D, Chen S, Gao S, Ma Y. Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 589–97
17. Idrees H, Tayyab M, Athrey K, Zhang D, Al-Maadeed S, Rajpoot N, Shah M. Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 532–46
18. Aich S, Stavness I. Leaf counting with deep convolutional and deconvolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) workshops; 2017
19. Go H, Byun J, Park B, Choi M-A, Yoo S, Kim, C.: Fine-grained multi-class object counting. In: 2021 IEEE International Conference on Image Processing (ICIP). IEEE; 2021. p. 509–13
20. Gongal A, Karkee M, Amatya S. Apple fruit size estimation using a 3D machine vision system. Inf Process Agric. 2018;5(4):498–503.
21. Oo LM, Aung NZ. A simple and efficient method for automatic strawberry shape and size estimation and classification. Biosys Eng. 2018;170:96–107.
22. Gené-Mola J, Sanz-Cortiella R, Rosell-Polo JR, Escolà A, Gregorio E. In-field apple size estimation using photogrammetry-derived 3D point clouds: comparison of 4 different methods considering fruit occlusions. Comput Electron Agric. 2021;188:106343.
23. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Deeplab: semantic image segmentation with deep convolutional

nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell. 2017;40(4):834–48.
24. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer; 2015. p. 234–41
25. Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg AC, Lo W-Y, et al. Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision; 2023. p. 4015–26
26. Ege T, Ando Y, Tanno R, Shimoda W, Yanai K. Image-based estimation of real food size for accurate food calorie estimation. In: 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). IEEE; 2019. p 274–9
27. Liang J, Hu D, Feng J. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In: International conference on machine learning. PMLR; 2020. p. 6028–39
28. Zhao Y, Zhong Z, Luo Z, Lee GH, Sebe N. Source-free open compound domain adaptation in semantic segmentation. IEEE Trans Circ Syst Video Technol. 2022;32(10):7019–32.
29. Sun K, Xiao B, Liu D, Wang J. Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019. p. 5693–703
30. White DS, Goldschen-Ohm MP, Goldsmith RH, Chanda B. Top-down machine learning approach for high-throughput single-molecule analysis. Elife. 2020;9:53357.
31. Deng J, Dong, W, Socher R, Li L-J, Li K, Fei-Fei L. Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE; 2009. p. 248–55
32. Hou S, Feng Y, Wang Z. VegFru: a domain-specific dataset for fine-grained visual categorization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2017
33. Williams HA, Jones MH, Nejati M, Seabright MJ, Bell J, Penhall ND, Barnett JJ, Duke MD, Scarfe AJ, Ahn HS, et al. Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. Biosyst Eng. 2019;181:140–56.
34. Ganesh P, Volle K, Burks T, Mehta S. Deep orange: Mask R-CNN based orange detection and segmentation. IFAC-PapersOnLine. 2019;52(30):70–5.
35. Yu Y, Zhang K, Yang L, Zhang D. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. Comput Electron Agric. 2019;163:104846.
36. Kim SL, Suk HJ, Kang JH, Jung JM, Laine TH, Westlin J. Using unity 3D to facilitate mobile augmented reality game development. In: 2014 IEEE World Forum on Internet of Things (WF-IoT). IEEE; 2014. p. 21–6
37. Cook DJ, Holder LB, Youngblood GM. Graph-based analysis of human transfer learning using a game testbed. IEEE Trans Knowl Data Eng. 2007;19(11):1465–78.
38. Shaker N, Abou-Zleikha M. Transfer learning for cross-game prediction of player experience. In: 2016 IEEE Conference on Computational Intelligence and Games (CIG). IEEE; 2016. p. 1–8
39. Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW. A theory of learning from different domains. Mach Learn. 2010;79:151–75.
40. Yao K, Su Z, Huang K, Yang X, Sun J, Hussain A, Coenen F. A novel 3D unsupervised domain adaptation framework for cross-modality medical image segmentation. IEEE J Biomed Health Inform. 2022;26(10):4976–86.
41. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V. Domain-adversarial training of neural networks. J Mach Lear Res. 2016;17(1):2096–2030.
42. Li J, Lü S, Li Z. Unsupervised domain adaptation via softmax-based prototype construction and adaptation. Inf Sci. 2022;609:257–75.
43. Su Z, Yao K, Yang X, Huang K, Wang Q, Sun J. Rethinking data augmentation for single-source domain generalization in medical image segmentation. In: Proceedings of the AAAI conference on artificial intelligence; 2023, vol 37, p. 2366–74
44. Niu S, Wu J, Zhang Y, Wen Z, Chen Y, Zhao P, Tan M. Towards stable test-time adaptation in dynamic wild world. 2023. arXiv:2302.12400
45. Nado Z, Padhy S, Sculley D, D'Amour A, Lakshminarayanan B, Snoek J. Evaluating prediction-time batch normalization for robustness under covariate shift. 2020. arXiv:2006.10963
46. Schneider S, Rusak E, Eck L, Bringmann O, Brendel W, Bethge M. Improving robustness against common corruptions by covariate shift adaptation. Adv Neural Inf Process Syst. 2020;33:11539–51.
47. Su Z, Guo J, Yao K, Yang X, Wang Q, Huang K. Unraveling batch normalization for realistic test-time adaptation. In: Proceedings of the AAAI conference on artificial intelligence; 2024. vol 38, p. 15136–44
48. Zhang H, Gu D. Deep multi-task learning for animal chest circumference estimation from monocular images. Cogn Computat. 2024; 1–11
49. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft coco: common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer; 2014. p. 740–55
50. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer, A. Automatic differentiation in Pytorch. In: Thirty-first conference on neural information processing systems; 2017
51. Liu Z. Super convergence cosine annealing with warm-up learning rate. In: CAIBDA 2022; 2nd international conference on artificial intelligence, big data and algorithms. VDE; 2022. p. 1–7