Iterative Knowledge Distillation for Automatic Check-Out

Libo Zhang, Dawei Du*, Congcong Li, Yanjun Wu, and Tiejian Luo

***Abstract*—Automatic Check-Out (ACO) provides an object detection based mechanism for retailers to process the purchases of customers automatically. However, it suffers a lot from the domain shift problem because of different data distribution between the single item in training exemplar images and mixed items in testing checkout images. In this paper, we propose a new iterative knowledge distillation method to solve the domain adaptation problem for this task. First, we develop a new augmentation data strategy to generate synthesized checkout images. It can extract segmented items from the training images by the coarse-to-fine strategy and filter items with unrealistic poses by pose pruning. Second, we propose a dual pyramid scale network (DPSNet) to exploit the multi-scale feature representation in joint detection and counting views. Third, the iterative knowledge distillation training strategy is developed to make full use of both image-level and instance-level samples to narrow the semantic gap between source domain and target domain. Extensive experiments on the large-scale Retail Product Checkout (RPC) dataset show the proposed DPSNet can achieve state-of-the-art performance compared with existing methods. The source codes can be found at https://isrc.iscas.ac.cn/gitlab/research/dpsnet.**

***Index Terms*—automatic check-out, domain adaptation, data augmentation, dual pyramid scale network, iterative knowledge distillation**

I. Introduction

AUTOMATIC Check-Out (ACO) attracts increasingly interest of researchers due to its practical applications in our daily life such as supermarkets and grocery stores. It can recognize all kinds of shopping items chosen by the costumers and output the final price by determining the categories and count of these items automatically and efficiently, resulting in better user experience for customers and lower operational costs for retailers. However, limited research focuses on this topic [1]. With the fast development of deep learning, it can be formulated as object detection problem using deep neural network [2]–[4].

Generally speaking, good performance for object detection relies on large-scale training and testing data in similar scenes.

This work was supported by the National Natural Science Foundation of China, Grant No. 61807033, the Key Research Program of Frontier Sciences, CAS, Grant No. ZDBS-LY-JSC038. Libo Zhang was supported by Youth Innovation Promotion Association of the Chinese Academy of Sciences (2020111), and Outstanding Youth Scientist Project of ISCAS. (Corresponding author: Dawei Du)

Libo Zhang and Yanjun Wu are with the State Key Laboratory of Computer Science, Institute of Software Chinese Academy of Sciences, Beijing 100190, China (e-mail: {libo,yanjun}@iscas.ac.cn).

Dawei Du is with the Computer Science Department, University at Albany, State University of New York, Albany, NY, USA (e-mail: cvdaviddo@gmail.com).

Congcong Li and Tiejian Luo are with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101400, China (licongcong18@mails.ucas.edu.cn, tjluo@ucas.ac.cn).

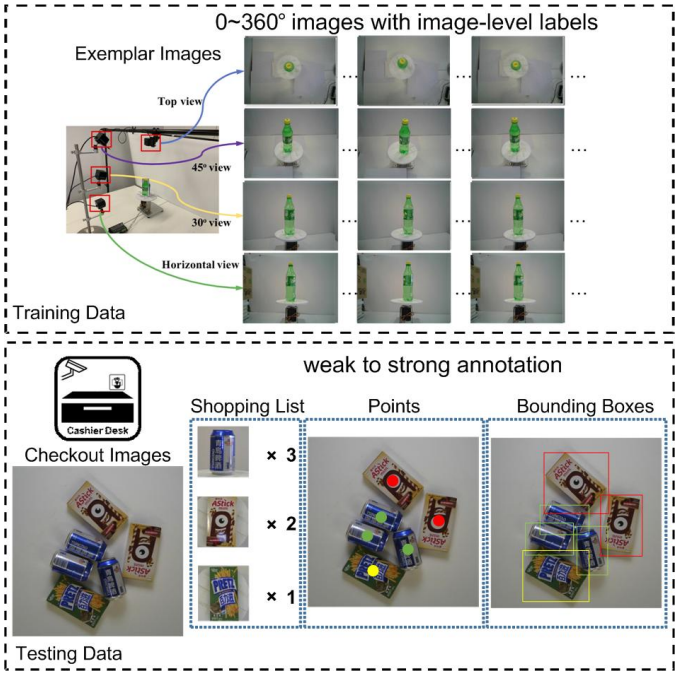

Fig. 1. The Retail Product Checkout (RPC) dataset for the Automatic Check-Out (ACO) system. The training data and testing data have different distribution.

However, unlike general object detection, we need to deal with the shift from source domain to target domain in the ACO task. As shown in Fig. 1, the training data include isolated items with different viewing angles captured from the turntable; while the testing data are several items piled together over a surface. Given the different distribution between training exemplar images and testing checkout images, the question then arises, "*how can we perform domain adaptation between the source domain with the isolated item and the target domain with mixed items in an image*?"

To this end, Wei *et al.* [1] propose a baseline object detection framework. Specifically, it first synthesizes the checkout images based on the images with isolated items using a random copy-and-paste strategy. Then, the CycleGAN method [5] is used to render the synthesized checkout images for more realistic lighting condition and shadows. Finally, the Feature Pyramid Network (FPN) [3] is trained based on rendered synthesized images and recognizes the category and count of items based on testing images. However, the previous work [1] mainly focuses on generating training samples similar to the testing data. The performance is still not satisfactory for two reasons. First, simple copy-and-paste using inaccurate

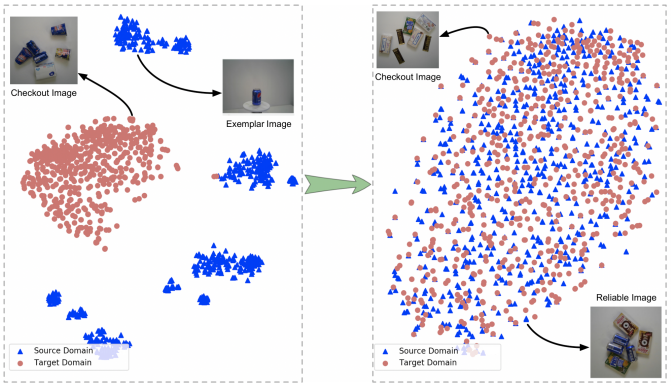

Fig. 2. The illustration of domain adaptation. There exists a huge gap between the isolated item in training exemplar images and mixed items in testing checkout images (left), and domain adaptation from reliable images to checkout images (right).

object segmentation will introduce background noises and affect detection accuracy. Second, there still exists considerable domain shift between training synthesized images and testing checkout images.

To address the above issues, we first propose a new data augmentation strategy to narrow the gap between training data and testing data. As shown in the left-hand part of Fig. 2, the data distribution of source domain and target domain is quite different[1]. We extract the foreground region from the training images of isolated objects using the coarse-to-fine saliency detection method. Moreover, we develop the pose pruning method to select images only with consistent configurations of the target domain as candidates, and generate synthesized images of checked out items with realistic poses.

However, there still exists a huge gap between synthesized images and testing checkout images. The solution is to distil knowledge from unlabeled testing checkout images directly. Inspired by the work in [7] that assigns pseudo-labels to unlabelled target samples, we develop the dual pyramid scale network (DPSNet) to exploit the multi-scale feature representation in joint detection and counting views. Note that the two views are derived from the same backbone to learn a common but different representation of data. If the outputs from two views are consistent, it is reliable to assign the corresponding testing checkout samples to pseudo labels.

Given the aforementioned synthesized samples and pseudo-labelled target samples, we develop the iterative knowledge distillation strategy to train the DPSNet to narrow the gap between source domain and target domain gradually (see the right-hand part of Fig. 2). Specifically, we select some testing samples to learn a common feature representation in each iteration and fine-tune the whole network based on the data from different domains. In this way, the target-discriminative representation can be learned to improve the accuracy on the target domain gradually. We conduct the experiments on the large-scale Retail Product Checkout (RPC) dataset [1].

[1]To visualize the data distribution, we employ the ResNet-101 network to calculate the high-dimensional feature of each sample and then embed it in a low-dimensional space of two dimensions by t-Distributed Stochastic Neighbor Embedding (t-SNE) [6].

Compared with existing methods, our method with the same backbone achieves the best 88.14%, 94.28%, 88.56%, and 81.59% checkout accuracy in terms of averaged, easy, medium, and hard difficulty levels, respectively. The main contributions of this work are summarized as follows.

- We propose the dual pyramid scale network (DPSNet) with joint counting and detection views to learn a common but different representation of data, and distinguish then reliable testing samples for pseudo label assignment.
- We develop the iterative knowledge distillation learning strategy to train the DPSNet model, which narrows the huge gap between synthesized images and testing checkout images gradually.
- Extensive experiments demonstrate the effectiveness of the proposed method compared with existing methods on the automatic check-out task.

A preliminary conference version of this work appeared in [8] where Data Priming Network (DPNet) was developed. It proposes a new data priming network with rough combination of detection head and counting head, and then selects some reliable testing images to narrow the gap between source domain and target domain. However, it does not make full use of multi-scale features of samples with high confidence in each image. In this work, we enhance DPNet [8] substantially on the aspects of network architecture, domain adaptation learning strategy and experimental evaluation, which is summarized in the following aspects:

- Instead of rough combination of detection and counting head, we improve the framework of DPNet by considering the multi-scale representation for detection and counting simultaneously;
- The iterative detection and counting collaborative learning strategy can make full use of reliable testing samples at both image-level and instance-level;
- We provide more ablative studies and qualitative analyses to demonstrate the superiority of DPSNet comprehensively, including different detection backbones, the computational complexity of network, and convergence of iterative learning scheme.

The remainder of this paper is organized as follows. A brief review of related works is presented in Section II. The details of the proposed DPSNet and learning strategy are given in Section III. Extensive experiments and ablation studies are given in Section IV. We conclude our method in Section V.

II. Related Work

A. Domain Adaptation

Deep neural networks perform not well in source and target domains with different distribution. To solve this issue, many domain adaptation training methods have been proposed. Szegedy *et al.* [9] develop a new training strategy by adding auxiliary classifiers to intermediate layers and discarding these auxiliary networks at inference time. In [10], the proposed collaborative learning framework adds multiple classifier heads of the same network and optimizes them collaboratively. In [11], a multi-task collaborative learning method is proposed to solve the facial landmark detection problem based on

auxiliary training and geometric constraints. Chen *et al.* [12] employ Faster R-CNN detection network [2] and design two domain adaptation components, on image level and instance level, to reduce the domain discrepancy. Wang *et al.* [13] develop a manifold embedded distribution alignment method to learn a domain-invariant classifier in Grassmann manifold with structural risk minimization, which is the first attempt to perform dynamic distribution alignment for manifold domain adaptation. Jing *et al.* [14] develop the heterogeneous hashing network to learn compact hash representations of both face images and videos for face retrieval across image and video domains. In terms of unsupervised domain adaptation, annotations are only available for source domain and no labels for target domain. To solve this problem, Lee [15] trains the target domain classifier using pseudo-labels with maximum predicted probability from the source domain trained classifier. Saito *et al.* [7] develop an asymmetric tri-training method for unsupervised domain adaptation, where unlabeled samples are assigned to pseudo labels selected by two asymmetric classifier heads.

Domain adaptation is widely used in cross-domain data (*e.g.*, image, video and text) analysis. Qian *et al.* [16] propose a generic cross-domain collaborative learning framework via a discriminative nonparametric Bayesian dictionary learning model. In [17], multimodal domain adaptation neural networks are proposed to learn domain-invariant features by constraining single modal features, fused features, and attention scores. In [18], the multi-kernel sparse representation-based domain-adaptive discriminative projection method is proposed to learn the discriminative features of the data in the two domains with the dictionary. Ma *et al.* [19] use the multi-modality adversarial network to learn semantic multi-modality representations to reduce domain discrepancy. In this work, we propose an iterative learning strategy to learn a common feature representation for different detection and counting views, based on reliable testing samples in both image-level and instance-level.

B. Knowledge Distillation

Domain adaptation methods usually use knowledge distillation to transfer representations between data domains. The goal of knowledge distillation is simple: train a student model that achieves better performance by knowledge transfer from the teacher model than it would if trained directly. It is first proposed in [20], which distills the knowledge in an ensemble of models into a single smaller model. Romero *et al.* [21] use intermediate representations learned by the teacher as hints to improve the training process and final performance of the student. Fukuda *et al.* [22] propose two new strategies for knowledge distillation using multiple teachers. In [23], a weighted cross-entropy loss is developed to address the problem of class imbalance and a teacher bounded loss is used to handle the regression component, where the adaptation layers can learn from intermediate teacher distributions effectively. Bagherinezhad *et al.* [24] introduce an iterative procedure that updates the ground truth labels after examining the entire dataset. Refining the labels while training enables the model to generate soft, informative, collective, and dynamic labels which results in major improvements. Furlanello *et al.* [25] develop a simple re-training procedure: after the teacher model converges, they initialize a new student identical to the teacher model and train it with the dual goals of predicting the correct labels and matching the output distribution of the teacher. This procedure is called **self-distillation**, which gains significant improvement in both computer vision and language modeling tasks. Knowledge distillation has been adapted to other tasks. Li *et al.* [26] mimick feature maps between the student and the teacher pooled from the same region proposal and discarded those from uninterested regions. Ning *et al.* [27] use learned projections that impose proper prior to inject external knowledge into the deep neural networks for the guidance of its training process in human pose estimation. Huang and Peng [28] propose the two-level progressive cross-media knowledge transfer method to transfer knowledge from large-scale cross-media data. Recently, Tang *et al.* [29] design an adaptive knowledge distillation loss, which is able to pay more attention to teacher-defined hard samples. Besides, they use a simple data filtering mechanism to train on unlabeled data in the semi-supervised setting. Heo *et al.* [30] provide a new perspective based on a decision boundary, *i.e.*, the generalization performance of a classifier is closely related to the adequacy of its decision boundary. Based on this idea, they train a student classifier based on the adversarial samples supporting the decision boundary to transfer more accurate information about the decision boundary. In this work, our framework can be regarded as iterative knowledge distillation between source domain and target domain.

C. Grocery Product Datasets

Recently, emerging interest occurs in integrating computer vision technology into the retail industry. However, there only exist few datasets for grocery product classification [31], [32], recognition [33]–[36], segmentation [37] and tallying [1]. This is because the retail industry requires a huge amount of human labor and a large percentage of the workload is spent on recognizing products. Klasson *et al.* [32] establish the dataset from fruit and vegetable sections and refrigerated sections in 18 different grocery stores, which consists of 5,125 images and 81 fine-grained classes. Jund *et al.* [36] collect 5,021 images of 25 grocery classes using smartphone cameras at various stores, apartments and offices in Freiburg, Germany, including 4,947 training images that contain one or more instances of one class, and 74 testing images of 37 clutter scenes that contain objects of multiple classes. Follmann *et al.* [37] propose the MVTec D2S dataset for instance semantic segmentation, which contains 21,000 images of 60 object categories with pixel-wise masks. However, the aforementioned datasets are relatively not challenging, resulting in a poor representation of checkout scenarios in real life.

In this work, we use the largest scale Retail Product Checkout (RPC) dataset [1] to evaluate the proposed algorithm. It includes 200 product categories and 83,739 images for both training data and testing data. It is further divided into 17 sub-categories, *i.e.*, `Puffed Food`,

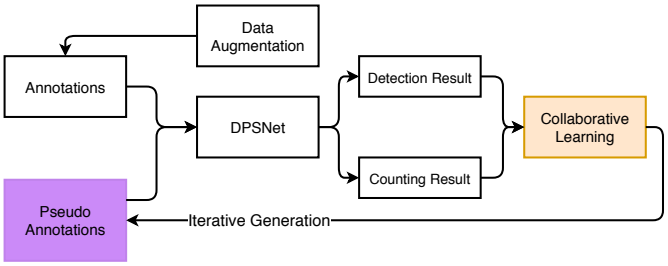

Fig. 3. The overall architecture of our proposed DPSNet.

`Dried Fruit`, `Dried Food`, `Instant Drink`, `Instant Noodles`, `Dessert`, `Drink`, `Alcohol`, `Milk`, `Canned Food`, `Chocolate`, `Gum`, `Candy`, `Seasoner`, `Personal Hygiene`, `Tissue`, and `Stationery`.

The training data contain 53,739 single-product images in total, where each image has a particular instance of a type of product. As shown in the top of Fig. 1, there are four cameras to collect exemplar images on a turntable for the top horizontal, 30° and 45° views. Meanwhile, each image is extracted every 9 degrees when the turntable rotating. The resolution of training images is 2592×1944. The *bounding boxes* annotation is provided to show the location and category of items in each image.

The testing data contain three sub-sets with different difficulty modes, each containing 10,000 images. As shown in the bottom of Fig. 1, we use the camera mounted on top to generate testing checkout images by putting random items on a $80cm \times 80cm$ whiteboard. The resolution of testing images is 1800×1800. Notably, according to the number of items in an image, the three difficulty modes in the RPC dataset consist of easy ($3 \sim 5$ categories and $3 \sim 10$ instances), medium ($5 \sim 8$ categories and $10 \sim 15$ instances), and hard ($8 \sim 10$ categories and $5 \sim 20$ instances). Three different annotations are provided for weak to strong supervision:

- *shopping lists* including the category and count of each instance,
- *point-level annotations* including the central position and the category of each item,
- *bounding boxes* including the bounding box and the category of each item in the checkout image.

III. Methodology

In this section, we present our data augmentation method, the architecture of the proposed DPSNet, and iterative training scheme to distill knowledge from source domain $\mathcal{S}$ to target domain $\mathcal{T}$ in detail. Specifically, we first employ the data augmentation strategy to narrow the gap between training data and testing data. Then, our DPSNet is used to generate pseudo annotations by collaborative learning of detection and counting heads. After that, we use the iterative training strategy to fine-tune the whole network based on the data from different domains gradually. The overall architecture is illustrated in Fig. 3 for better understanding.

A. Data Augmentation

To generate random rendered synthesized images for training, we collect training images of segmented items to remove those with irrelevant poses. It consists of three steps including *background removal*, *pose pruning* and *checkout images synthesis*.

1) Background Removal: Since the exemplar images in the RPC dataset [1] are captured on the turntable, the provided bounding box annotations contain background noise. We remove the background noise to narrow the gap between training images and testing images by coarse-to-fine refinement strategy [8]. As shown in Fig. 4, we first use the edge detector [38] to extract the contour of the item and then remove the edges with low confidence (*i.e.*, confidence score less than 0.1). Secondly, we fill the holes inside the contour and remove small isolated regions by mathematical morphology (*i.e.*, dilation and erosion operations). Thirdly, the edges of the item mask are smoothed by the median filter. After that, the saliency detection network [39] is used to generate fine masks based on the coarse masks, which is trained on the MSRA-B salient object dataset [40]. Notably, the saliency model is further fine-tuned on the coarse masks.

2) Pose Pruning: To generate synthesized checkout images, we randomly select multiple segmented items and paste them on a prepared background image. However, not all the poses of the isolated items are viable in checkout images. As shown in Fig. 5, it is not stable to put bag-like or can-like items on the checkout table with the view from bottom to top, which is called unrealistic pose. To remove them, we introduce a simple metric based on the ratio of areas, *i.e.*, $\mathcal{R}_{k,v} = \frac{\mathcal{A}_{k,v}}{\max_v \mathcal{A}_{k,v}}$. $\mathcal{A}_{k,v}$ denotes the area of the item mask captured by the v-th view in the k-th category and $\max_v \mathcal{A}_{k,v}$ the maximal area of pose. If the ratio is less than a pre-set threshold θ_m, it indicates that the area of this pose is too small to be put on the checkout table stably; otherwise, we regard this pose as a realistic pose.

3) Checkout Images Synthesis: Given the segmented items with realistic pose, the checkout images are synthesized by using the method in [1]. Specifically, segmented items are randomly selected and freely placed (*i.e.*, random angles from 0 to 360 and scales from 0.4 to 0.7) on a prepared background image such that the occlusion rate of each instance less than 50%. Thus the synthesized images are similar to true checkout images in terms of item placement (see the first row of Fig. 6). As discussed before, the synthesized checkout images (see the second row of Fig. 6) still lack lighting and shadow characteristics of true checkout images (see the third row of Fig. 6). Therefore, we employ the Cycle-GAN method [5] to render synthesized checkout images for more realistic lighting condition and shadows.

B. Network Architecture

As discussed in the introduction, the *source domain* of rendered checkout images is still different from the *target domain* of real checkout images after data augmentation. Inspired by [7], we propose the Dual Pyramid Scale Network (DPSNet) to learn a common feature representation from two different views, *i.e.*, detection and counting. As shown in

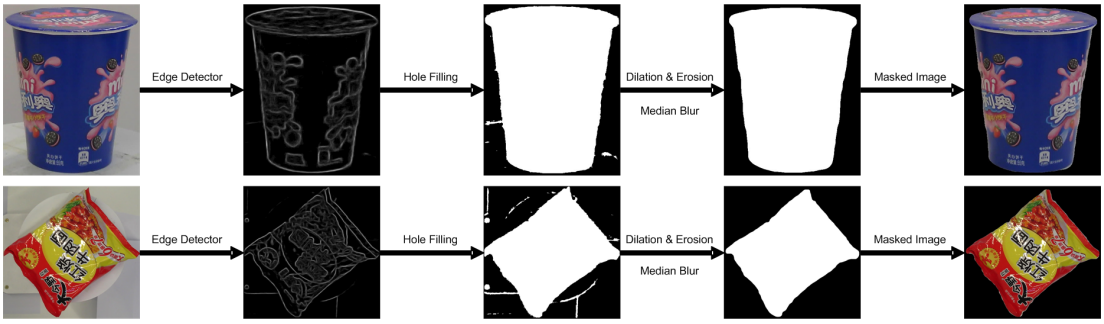

Fig. 4. Background removal by mathematical morphology.

Fig. 5. Comparison between the items with unrealistic and realistic poses.

Fig. 6. Comparison of synthesized checkout images (first row), rendered checkout images (second row) by Cycle-GAN [5] and true checkout images (third row).

Fig. 7, the detection view $\mathcal{D}$ produces a set of objects O, which is a 5-tuple in the form of (c, x, y, w, h), where c is the index of object category and x, y, w, h are central coordinates and scales of the object. The counting view C predicts a density map $\hat{\Theta} \in \mathcal{R}^{H \times W \times C}$, where C, H and W indicate the density category, density map height and width, respectively.

According to the work in [3], low-level features are with more location information while high-level features are with strong semantical information. To make full use of the features, we use all level feature maps from the backbone of feature pyramid network (FPN) [3], *i.e.*, $\mathcal{P} = \{P2, P3, P4, P5, P6\}$. For the low-resolution feature map, we upsample the spatial resolution using the bilinear method. The upsampled map is followed by one 1×1 convolutional layer (without ReLU layer) to learn to select useful information, then added to the next level feature. This upsample procedure is stopped as $P3$, and then we use the same procedure to down-sample $P2$ and add it to $P3$. The fused feature map has the same shape as that in the original $P3$ feature maps. Finally, the counting head consists of one 3×3 convolutional layer and one 1×1 convolutional layer to predict density map; while the detection head contains fully connected layers to calculate regression and classification results of detection proposals. Compared to our preliminary work [8], it is worth mentioning that the proposed network considers the multi-scale representation for counting and detection views simultaneously. On the other hand, the counting head has been greatly simplified compared with [41]. Therefore, we train the detection and counting network based on training data in end-to-end fashion more efficiently (see the results in Table IV).

C. Iterative Knowledge Distillation

After the detection model is initialized on the training data (source domain), we learn a common feature representation from source domain to target domain gradually via the iterative knowledge distillation scheme. Before that, we revisit the previous representative knowledge distillation methods briefly.

Fig. 8(a) shows the conventional knowledge distillation procedure [20]. The class probabilities produced by the cum-

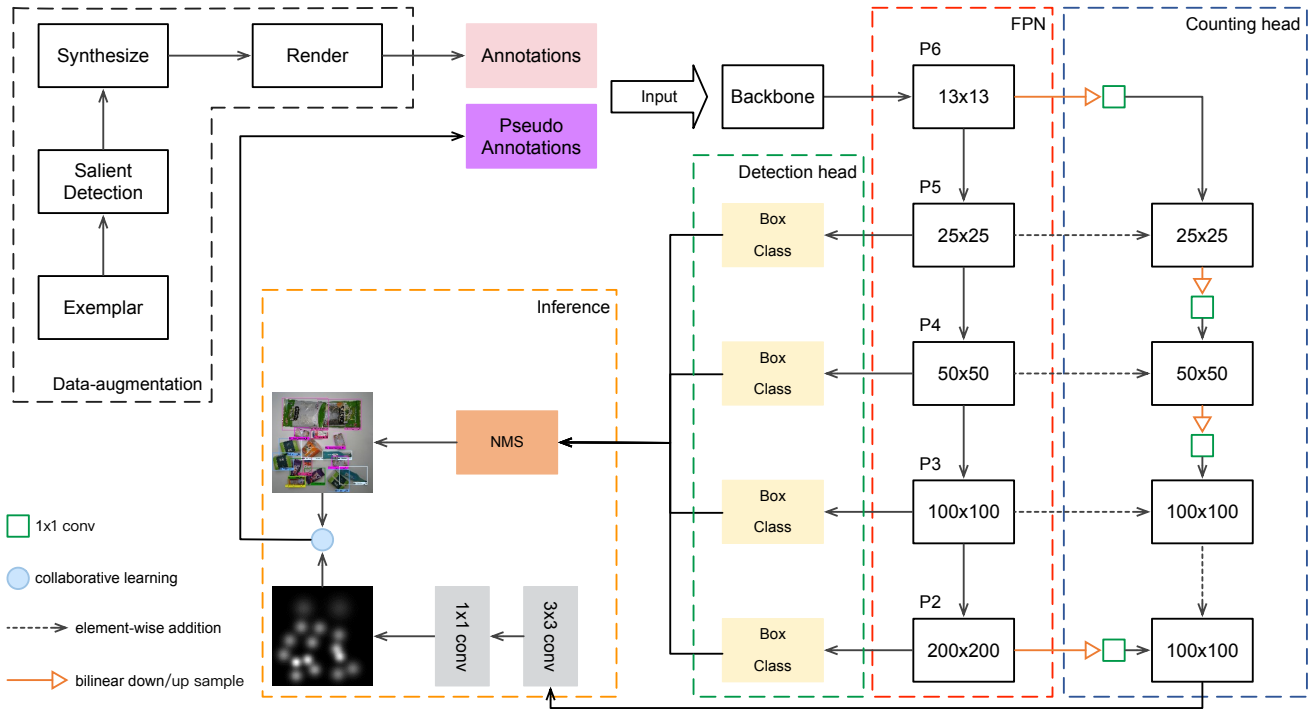

Fig. 7. The detailed framework of the proposed network. The red dashed bounding box indicates feature pyramid network (FPN) [3] fuse module. The blue and green dashed bounding boxes correspond to the counting and detection heads, respectively. The orange dashed bounding box measures the target consistency.

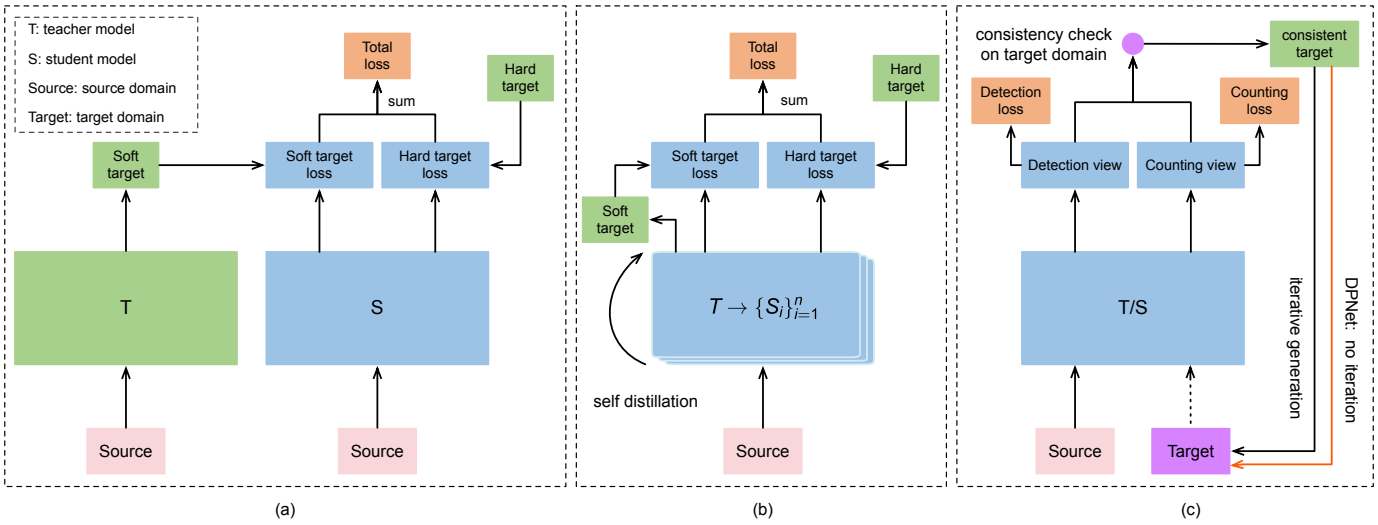

Fig. 8. Different knowledge distillation frameworks. (a) traditional knowledge distillation method; (b) self-distillation training method; (c) our proposed iterative knowledge distillation method.

bersome teacher model T are used as soft targets to train the small student model S. If the soft targets have high entropy, they provide much more information per training case than hard targets and much less variance in the gradient between training cases. Thus the small model can be trained on much less data than the original cumbersome model by using a much higher learning rate. However, this procedure is often limited to the same domain where the teacher model T and the student model S are trained on the dataset with the same distribution. Besides, unlike discrete categories, it is difficult to use soft targets for regression tasks like object detection directly. This is because the teacher's regression outputs may provide wrong guidance toward the student model and the variance and mean of regression targets may vary greatly.

On the other hand, the self-distillation procedure [25] is shown in Fig. 8(b), where the student network is identical to the teacher model in terms of the network graph. This distillation process can be performed consecutively several times. At each consecutive step, a new identical model is initialized from a different random seed and trained from the supervision of the earlier generation. At the end of the procedure, additional gains can be achieved with an ensemble

of multiple students generations. An ensemble of multiple student generations increases the inference time accordingly as well.

Different from the aforementioned knowledge distillation methods, our method is based on two views with the common backbone, as shown in Fig. 8(c). Specifically, the detection view learns to localize objects while the counting view focuses on counting objects. However, both two views learn from the shared backbone and back-propagates gradients independently during training. To distill knowledge from source domain to target domain, we use **consistent targets** instead of soft and hard targets. Consistent targets are defined as predicted outputs from the detection view that passes the **consistency check** such that

$$\Big\lceil \sum_{i=0}^{H}\sum_{j=0}^{W}\hat{\Theta}(i,j,c)\Big\rfloor = \sum_{d\in D}\mathbb{I}(\mathcal{P}_c(d)>\theta_p), \forall c\in C, \qquad (1)$$

where (i, j) enumerates all positions in the density map $\hat{\Theta}\in\mathcal{R}^{H\times W\times C}$. $\lceil\cdot\rfloor$ indicates the rounding operation. $\mathcal{P}_c(d)$ is the probability of detection d belonging to the c-th category, where D is the set of detections and C is the set of categories. θ_p is the threshold of detection confidence. $\mathbb{I}(\cdot)=1$ if its argument is true, and 0 otherwise.

Iterative learning. We perform the consistency check on the unlabelled target domain to select reliable images using pseudo labels. If the number of objects with high confidence equals to the count number estimated by density map in all density categories, we assign the sample to valid pseudo label and regard it as a reliable image; otherwise, we discard the outputs. To guide the training procedure converging on target domain, we use both source domain data and selected target domain data to fine-tune the initialized detection network iteratively. That is, we train the network using the testing data with pseudo labels generated in the previous step, and then generate new samples with pseudo labels after this step stops. Thus this training procedure is iterative as shown in Fig. 8(c).

Pseudo labels at instance-level. After iterative training, to make full use of training samples, we re-collect the instances with high confidence and ignore others in the discarded unreliable images. Specifically, we remove the counting head C of the DPSNet, and then fill background pixels in the bounding boxes of the objects with low confidence (*i.e.*, lower than the threshold θ_p). As shown in Fig. 9, there are three types of ignored instances: (A) the objects with similar appearance as that of neighbouring objects, (B) occluded objects in crowded scenes, (C) non-product objects in check-out images of the RPC dataset. Finally, we fine-tune the network using filtered unreliable testing images. In such an iterative training procedure, the knowledge flows from source domain to target domain at both image-level and instance-level. For better understanding, the whole procedure of iterative training is presented in Algorithm 1.

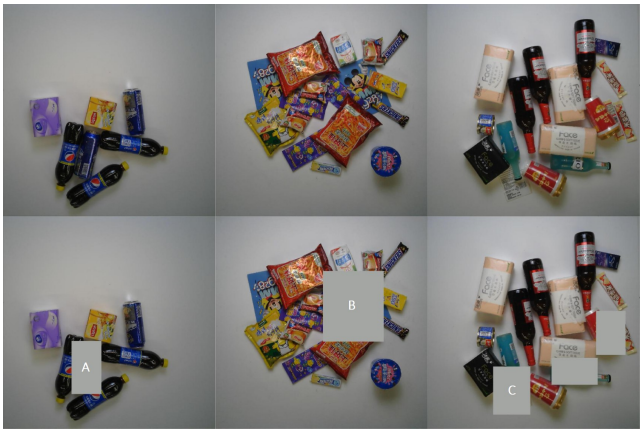

Fig. 9. Comparison between original checkout images and filtered images without low-confidence instances (from top to bottom). There exist three types of ignored objects with low-confidence (instance A,B,C).

Algorithm 1 Iterative Knowledge Distillation

Input: rendered training data $\mathcal{S}$ with annotations, unlabelled testing data $\mathcal{T}$
Output: the model of DPSNet

for $j = 1$ `to` N_{iter} **do**
  Train DPSNet with counting head C and detection head $\mathcal{D}$ using rendered training data $\mathcal{S}$.
end for
Assign reliable testing data $\hat{\mathcal{T}}$ to pseudo labels based on Eq. (1).
for $i = 1$ `to` N_{step} **do**
  for $j = 1$ `to` N'_{iter} **do**
    Train DPSNet based on training data and reliable testing data $\mathcal{S}\cup\hat{\mathcal{T}}$.
  end for
  Assign reliable testing data $\hat{\mathcal{T}}$ to pseudo labels based on Eq. (1).
end for
Remove the counting head C of DPSNet.
Filter unreliable testing images by ignoring the instances with low confidence.
Fine-tune DPSNet using filtered unreliable testing images.

D. Loss Function

For iterative training, we consider loss function of both counting and detection heads. For the counting head, the squared Euclidean distance between the ground-truth map and the estimated density map is computed. For the detection head, the standard cross-entropy loss for classification and the smooth L1 loss for regression [3] are computed. The whole loss function is calculated as

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{\mathbb{N}}\Big(&\sum_{\ell}|\hat{\Theta}(\ell;x_i)-\Theta(\ell;x_i)|^2\\ &+\sum_{d}(\mathcal{L}_{\text{cls}}(\hat{p}_d,p_d;x_i)+\mathbb{I}(p_d>0)\cdot\mathcal{L}_{\text{reg}}(\hat{t}_d,t_d;x_i))\Big),\end{aligned} \qquad (2)$$

where x_i denotes the input checkout image with the size of 800×800 and $\mathbb{N}$ is the batch size. $\hat{\Theta}(\ell;x_i)$ and $\Theta(\ell;x_i)$ are estimated and ground-truth density values of location ℓ in image x_i, respectively. Notably, the density maps are $1/8$ size of the input image, *i.e.*, 100×100. $\hat{p}_d$ and p_d are predicted and ground-truth category label of detection d in image x_i, respectively. $\mathbb{I}(p_d>0)$ means that we only calculate the

regression loss of objects. We have $\mathbb{I}(p_d > 0) = 1$ if its argument is true (objects), and $\mathbb{I}(p_d > 0) = 0$ otherwise (background). $\hat{t}_d$ and t_d are the regression vectors that represent the 4 parameterized coordinates of the predicted and ground-truth bounding box of detection d in the image x_i, respectively. In terms of generating ground-truth density maps, we generate a map with the normalized Gaussian kernel based on the central locations of each object, and then sum up all the density maps to produce the final ground-truth density map, similar to the work in [41].

IV. Experiment

We evaluate our method on the RPC dataset [1] compared with two existing methods [1], [8]. We use several metrics following [1], including Checkout Accuracy (cAcc), Average Counting Distance (ACD), Mean Category Counting Distance (mCCD), Mean Category Intersection of Union (mCIoU), and two mean Average Precision scores (*i.e.*, mAP50 and mmAP), which are defined as follows.

- Checkout Accuracy (cAcc$\in [0, 1]$) is the accuracy when the complete shopping list is predicted correctly, which is computed as

$$\text{cAcc} = \frac{\sum_{i=1}^{N} \delta(\sum_{k=1}^{K} \text{CD}_{i,k} = 0)}{N}. \tag{3}$$

  In Eq.(3), $\text{CD}_{i,k} = |P_{i,k} - \text{GT}_{i,k}|$ is the counting error for a specific category in an image, where $P_{i,k}$ and $\text{GT}_{i,k}$ correspond to the predicted and ground-truth number of items in the k-th category in the i-th image, respectively. cAcc = 1 means that all items are predicted accurately, *i.e.*, $\sum_{k=1}^{K} \text{CD}_{i,k} = 0$. This is the primary metric.
- Mean Category Intersection of Union (mCIoU$\in$ [0, 1]) is the overlap between the predicted and ground-truth shopping list, which is defined as

$$\text{mCIoU} = \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i=1}^{N} \min(\text{GT}_{i,k}, P_{i,k})}{\sum_{i=1}^{N} \max(\text{GT}_{i,k}, P_{i,k})}. \tag{4}$$

- Average Counting Distance (ACD) is the average number of counting errors for each image:

$$\text{ACD} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \text{CD}_{i,k}. \tag{5}$$

- Mean Category Counting Distance (mCCD) calculates the average ratio of counting errors for each category:

$$\text{mCCD} = \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i=1}^{N} \text{CD}_{i,k}}{\sum_{i=1}^{N} \text{GT}_{i,k}}. \tag{6}$$

- The mean Average Precision (mAP) metrics including mAP50 and mmAP are used to evaluate the object detection performance. mAP50 is the average precision based on the Intersection over Union (IoU) threshold 0.50 over all the categories, while mmAP is the mean average precision of all item categories based on all 10 IoU thresholds from the interval [0.50, 0.95] in steps of 0.05. Please refer to evaluation protocols in MS COCO [42] and the ILSVRC 2015 challenge [43] for more details.

A. Implementation Details

The propose DPSNet is implemented by PyTorch [44] on a workstation with 4 Nvidia TITAN Xp GPU cards. We set the batch size to 8 in the training phase, where each image includes at most 256 detection proposals. Then, the network is trained using SGD optimization method with 0.9 momentum and 0.0001 weight decay. We set the initial learning rate to 10^{-7} and 0.01 for counting and detection heads, respectively. Besides, the channel dimension of density map in the counting head is set as $C = 1$. In Section III-A2, the threshold for pose pruning is set as $\theta_m = 0.45$. In Section III-C, the threshold for consistency check is set as $\theta_p = 0.95$.

In terms of iterative knowledge distillation training, the initial learning rate is 0.001 for the first 7 training steps, which decays by a factor of 10 at the 10-th step and 12-th step. The training steps are 14 in total in our experiment. For each step, we train 10,000 iterations using the pseudo labels generated in the previous step, then we generate new pseudo labels after this step stops.

B. Results and Analysis

We evaluate the proposed DPSNet compared with the Wei *et al.* method [1] and Data Priming Network (DPNet) [8]. For each compared method, we conduct the experiments on two variants including *Render* and *Syn+Render*. The *Render* method is trained based on rendered synthesized images using Cycle-GAN [5]. The *Syn+Render* method is trained based on both synthesized and rendered images. Note that we generate 100,000 synthesized checkout images by copying and pasting the segmented isolated items to the background. DPNet [8] is an improved FPN detector with both detection and counting heads, which is optimized by collaborative detection and counting learning. Besides, we construct additional *Instance+Render* variant for the proposed DPSNet, where we train the network on both instance-level samples and rendered images.

As shown in Table I, the *Render* baseline method achieves only 45.60% cAcc score on averaged clutter mode. This may be due to considerable domain shift between rendered training samples and real checkout images. Based on both synthesized and rendered images, the domain shift problem is solved to some degree, resulting in better 56.68% checkout accuracy. After further introducing reliable testing samples in the training phase, the checkout accuracy of DPNet [8] is improved from 45.60% to 77.91% significantly. Similarly, the performance reaches 80.51% when we train DPNet [8] based on both synthesized and rendered images.

Compared with the previous methods, our DPSNet employs an iterative training scheme to learn a common feature representation for both source domain and target domain more effectively. That is, better 86.54% checkout accuracy is obtained on averaged clutter mode. It is worth mentioning that the "DPSNet (Syn+Render)" method achieves slightly inferior performance than the "DPSNet (Render)" method (*i.e.*, 85.98% vs. 86.54%). We speculate that DPSNet can make full use of rendered training data while synthesized data introduce additional domain shift. In terms of other metrics

TABLE I
EXPERIMENTAL RESULTS ON THE RPC DATASET.

Clutter mode	Methods	cAcc (↑)	ACD (↓)	mCCD (↓)	mCIoU (↑)	mAP50 (↑)	mmAP (↑)
Easy	Wei *et al.* [1] (Render)	63.19%	0.72	0.11	90.64%	96.21%	77.65%
	Wei *et al.* [1] (Syn+Render)	73.17%	0.49	0.07	93.66%	97.34%	79.01%
	DPNet [8] (Render)	89.74%	0.16	0.02	97.83%	98.52%	82.75%
	DPNet [8] (Syn+Render)	90.32%	0.10	0.02	97.87%	98.60%	83.07%
	DPSNet (Render)	93.00%	0.10	0.01	98.57%	99.10%	84.69%
	DPSNet (Syn+Render)	93.01%	0.10	0.01	98.54%	99.03%	84.96%
	DPSNet (Instance+Render)	94.28%	0.08	0.01	98.94%	99.23%	85.18%
Medium	Wei *et al.* [1] (Render)	43.02%	1.24	0.11	90.64%	95.83%	72.53%
	Wei *et al.* [1] (Syn+Render)	54.69%	0.90	0.08	92.95%	96.56%	73.24%
	DPNet [8] (Render)	77.75%	0.35	0.03	97.04%	97.92%	76.78%
	DPNet [8] (Syn+Render)	80.68%	0.32	0.03	97.38%	98.07%	77.25%
	DPSNet (Render)	87.10%	0.19	0.02	98.38%	98.85%	79.34%
	DPSNet (Syn+Render)	86.28%	0.21	0.02	98.20%	98.83%	79.69%
	DPSNet (Instance+Render)	88.56%	0.16	0.01	98.69%	98.86%	79.85%
Hard	Wei *et al.* [1] (Render)	31.01%	1.77	0.10	90.41%	95.18%	71.56%
	Wei *et al.* [1] (Syn+Render)	42.48%	1.28	0.07	93.06%	96.45%	72.72%
	DPNet [8] (Render)	66.35%	0.60	0.03	96.60%	97.49%	74.67%
	DPNet [8] (Syn+Render)	70.76%	0.53	0.03	97.04%	97.76%	74.95%
	DPSNet (Render)	79.65%	0.32	0.02	98.16%	98.45%	77.32%
	DPSNet (Syn+Render)	78.71%	0.34	0.02	98.06%	98.51%	77.60%
	DPSNet (Instance+Render)	81.59%	0.26	0.02	98.49%	98.51%	77.88%
Averaged	Wei *et al.* [1] (Render)	45.60%	1.25	0.10	90.58%	95.50%	72.76%
	Wei *et al.* [1] (Syn+Render)	56.68%	0.89	0.07	93.19%	96.57%	73.83%
	DPNet [8] (Render)	77.91%	0.37	0.03	97.01%	97.74%	76.80%
	DPNet [8] (Syn+Render)	80.51%	0.34	0.03	97.33%	97.91%	77.04%
	DPSNet (Render)	86.54%	0.21	0.02	98.33%	98.56%	79.18%
	DPSNet (Syn+Render)	85.98%	0.22	0.02	98.24%	98.61%	79.46%
	DPSNet (Instance+Render)	88.14%	0.17	0.01	98.66%	98.64%	79.75%

such as ACD, mCIoU and mmAP, our method achieves the best performance in different difficulty levels.

Moreover, to verify the robustness of our method applied in more advanced detectors, we also evaluate DPSNet with different detection backbones including FPN [3], Mask R-CNN [45], Libra R-CNN [46] and Cascade R-CNN [47]. For a fair comparison, all experiments settings are the same except the detectors. As shown in Table II, the cAcc scores are increased along with the improvement of detection accuracies of different backbones. Specifically, the "Mask R-CNN (Instance+Render)" method performs slightly better than the "FPN (Instance+Render)" method by using another segmentation head. The "Libra R-CNN (Instance+Render)" method tries to reduce the imbalance at sample, feature, and objective level, resulting in 89.67% cAcc score. The "Cascade R-CNN (Instance+Render)" method employs a sequence of detectors trained with increasing IoU thresholds to achieve over 2% improvement on the cAcc score (*i.e.*, 90.74% vs. 88.14%) and approximate 3% gain on the mmAP score (*i.e.*, 82.72% vs. 79.75%). It indicates that our method can further boost the performance when using advanced detectors. Besides, we provide visual examples in different difficulty levels of the RPC dataset in Fig. 10.

C. Ablation Study

In this section, the ablation study is conducted to explore the complexity of network and convergence procedure of iterative learning scheme. we first study the influence of the number of density categories C on the performance. In addition, as shown in Table III, we construct several DPSNet variants and evaluate them on the RPC dataset to show the effectiveness. Notably, we use the same parameter settings including input size (800×800) and training data.

1) Influence of Density Categories: To analyze the influence of different density categories C in Section III-B, we enumerate C token 1, 17 and 200, as shown in Fig. 11. If $C = 1$, we obtain 86.54% cAcc score, which means that we regard all the products as one category. If $C = 17$, we obtain 80.05% cAcc score by using 17 super categories as density categories. If $C = 200$, we only obtain 77.22% cAcc score by considering all the categories. In terms of other metrics including MAE (mean absolute error), label accuracy and the number of selected samples, we can see the similar trend. The MAE score is decreased as the training steps increase, which is defined as

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\left|M_{x_i,c} - \hat{M}_{x_i,c}\right|, \tag{7}$$

where N is the number of images in the testing set. $\hat{M}_{x_i,c}$ and $M_{x_i,c}$ are the estimated and ground-truth counts of image x_i for the c-th density category respectively, *i.e.*, $\hat{M}_{x_i,c} = \sum_{i=0}^{H}\sum_{j=0}^{W}\hat{\Theta}(i,j,c;x_i)$. The label accuracy means the percentage of correct label assignment for instances in all the selected instances, and the number of selected samples is the total number of instances for training. In summary, it is more difficult to converge based on more density categories. Therefore, we set $C = 1$ in our network.

2) Effectiveness of Dual Pyramid Scale Network: To capture the appearance of various scales of objects, we propose the dual pyramid scale network (DPSNet, see Fig. 7). From Table III, DPSNet performs similarly compared with the variant

Fig. 10. Detection results of the proposed DPSNet for easy, medium, and hard modes (from top to down). Different color bounding boxes correspond to different predicted categories. Best view in color.

without a dual pyramid scale representation (*i.e.*, DPNet [8]) if no collaborative learning of detection and counting heads is performed for domain adaptation (*i.e.*, 70.08% vs. 70.80%). If we the use collaborative learning strategy based on the dual pyramid scale network, the cAcc score grows 1.44%, *i.e.*, 79.35% vs. 77.91%.

Moreover, we report the complexity and running time of existing methods in Table IV. Compared with Wei *et al.* [1], DPNet [8] has 2.73 million more parameters and achieves a promising cAcc score of 77.91%. However, the training time also increases from 0.36 second to 1.20 second per iteration. The proposed DPSNet increases only 0.85 million parameters than Wei *et al.* [1] but achieves more than 20% improvement in checkout accuracy. It indicates that our method is efficient and effective. It is worth mentioning that the compared methods have the same inference speed. This is because all of them are based on the FPN detector, where the only difference is the training strategy. In other words, only the original FPN backbone is used in the inference phase and all other additional modules (*e.g.*, the counting head and consistency check module) are removed.

3) Effectiveness of Collaborative Learning: If we only consider the multi-scale feature representation in the network, the accuracy is slightly degraded. However, the performance is significantly improved by using the proposed detection and counting collaborative learning (70.80% vs. 79.35%). This is maybe because detection and counting heads are mutually affected without collaborative learning, resulting in local optimum of two different views in the training phase. The experiment results indicate the importance and effectiveness of collaborative learning.

4) Effectiveness of Iterative Learning: From Table III, it can be seen that the significant improvement in cAcc score is obtained by using iterative learning strategy, *i.e.*, 79.35% vs. 86.54%. To further show the effectiveness of iterative training scheme, we provide the iteration results in Fig. 12. It is observed that the checkout accuracy increases along with the increase of training steps and reaches the maximal value at the 12-th step (see the blue line in Fig. 12(a)). It means that we have taken full use of testing samples to facilitate training our model. Moreover, the red line denotes the performance of the iterative training scheme based on selected samples with ground-truth annotations, which shows the upper bound of the

TABLE II
COMPARISON BETWEEN DIFFERENT BACKBONES INCLUDING FPN [3], MASK R-CNN [45], LIBRA R-CNN [46] AND CASCADE R-CNN [47].

Clutter mode	Methods	cAcc (↑)	ACD (↓)	mCCD (↓)	mCIoU (↑)	mAP50 (↑)	mmAP (↑)
Easy	FPN (Render)	93.00%	0.10	0.01	98.57%	99.10%	84.69%
	FPN (Instance+Render)	94.28%	0.08	0.01	98.94%	99.23%	85.18%
	Mask R-CNN (Render)	93.54%	0.09	0.01	98.63%	99.15%	85.01%
	Mask R-CNN (Instance+Render)	94.63%	0.08	0.01	99.01%	99.32%	85.89%
	Libra R-CNN (Render)	95.32%	0.07	0.01	99.05%	99.36%	86.69%
	Libra R-CNN (Instance+Render)	95.84%	0.06	0.01	99.08%	99.41%	87.51%
	Cascade R-CNN (Render)	96.71%	0.05	0.01	99.12%	99.46%	88.77%
	Cascade R-CNN (Instance+Render)	97.22%	0.04	0.01	99.35%	99.63%	89.45%
Medium	FPN (Render)	87.10%	0.19	0.02	98.38%	98.85%	79.34%
	FPN (Instance+Render)	88.56%	0.16	0.01	98.69%	98.86%	79.85%
	Mask R-CNN (Render)	87.77%	0.18	0.01	98.46%	98.85%	79.44%
	Mask R-CNN (Instance+Render)	89.01%	0.15	0.01	98.84%	98.98%	82.11%
	Libra R-CNN (Render)	89.87%	0.14	0.01	98.97%	99.01%	82.18%
	Libra R-CNN (Instance+Render)	90.33%	0.13	0.01	99.01%	99.09%	82.21%
	Cascade R-CNN (Render)	90.81%	0.12	0.01	99.05%	99.12%	82.34%
	Cascade R-CNN (Instance+Render)	91.64%	0.09	0.01	99.17%	99.28%	82.97%
Hard	FPN (Render)	79.65%	0.32	0.02	98.16%	98.45%	77.32%
	FPN (Instance+Render)	81.59%	0.26	0.02	98.49%	98.51%	77.88%
	Mask R-CNN (Render)	80.12%	0.30	0.01	98.32%	98.47%	77.46%
	Mask R-CNN (Instance+Render)	82.05%	0.23	0.01	98.75%	98.83%	79.31%
	Libra R-CNN (Render)	82.11%	0.22	0.01	98.78%	98.84%	79.42%
	Libra R-CNN (Instance+Render)	83.16%	0.20	0.01	98.82%	98.86%	79.75%
	Cascade R-CNN (Render)	83.22%	0.19	0.01	98.89%	98.89%	79.86%
	Cascade R-CNN (Instance+Render)	85.06%	0.16	0.01	99.01%	99.01%	80.15%
Averaged	FPN (Render)	86.54%	0.21	0.02	98.33%	98.56%	79.18%
	FPN (Instance+Render)	88.14%	0.17	0.01	98.66%	98.64%	79.75%
	Mask R-CNN (Render)	86.96%	0.20	0.01	98.39%	98.58%	79.35%
	Mask R-CNN (Instance+Render)	88.59%	0.15	0.01	98.73%	98.79%	81.24%
	Libra R-CNN (Render)	88.79%	0.14	0.01	98.80%	98.83%	81.30%
	Libra R-CNN (Instance+Render)	89.67%	0.13	0.01	98.89%	98.85%	81.56%
	Cascade R-CNN (Render)	89.81%	0.13	0.01	98.96%	98.88%	81.90%
	Cascade R-CNN (Instance+Render)	90.74%	0.10	0.01	99.12%	99.02%	82.72%

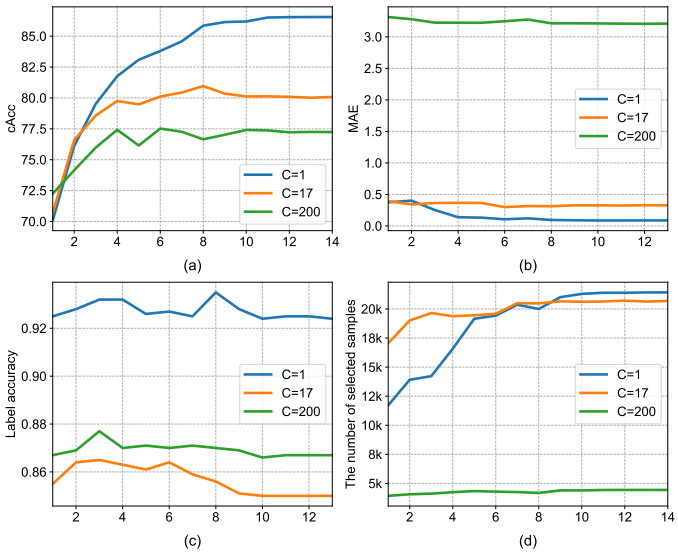

Fig. 11. Comparison of different number of density categories $C = \{1, 17, 200\}$ in terms of cAcc, MAE, label accuracy and the number of selected samples.

TABLE III
EFFECTIVENESS OF VARIOUS DESIGNS. ALL MODELS ARE TRAINED AND TESTED ON THE RPC DATASET.

Component	DPSNet					
dual pyramid scale?		✔	✔	✔	✔	
collaborative learning?	✔		✔	✔	✔	
iterative learning?				✔	✔	
instance-level sample?					✔	
cAcc	70.80%	77.91%	70.08%	79.35%	86.54%	88.14%

TABLE IV
COMPARISON OF THE COMPLEXITY OF DIFFERENT METHODS.

Method	# of Extra Params	cAcc	Training	Inference
Wei *et al.* [1]	0	56.68%	0.36 s/iter	0.16 s/img
DPNet [8]	2.73M	77.91%	1.20 s/iter	0.16 s/img
DPSNet	0.85M	79.35%	0.38 s/iter	0.16 s/img

accuracy. The decreased MAE score indicates the counting head can predict more accurate number of items. Fig. 12(c) shows the accuracy of selected pseudo labels. It can be seen that the label accuracy remains stable. Specifically, based on the consistency check equation (1), 92.70% of pseudo labels are assigned correctly, which shows the effectiveness of our training strategy. As shown in Fig. 12(d), more reliable testing samples are used in the training phase with the increase of training steps. In summary, both the accuracy of pseudo labels and the number of training samples are crucial to the check-out accuracy.

5) Effectiveness of Instance-level Samples: As discussed before, DPNet [8] only collects reliable testing samples at image-level, which easily ignores reliable samples with high confidence in the discarded images. Therefore, the reliable samples are not taken into consideration for domain adaptation

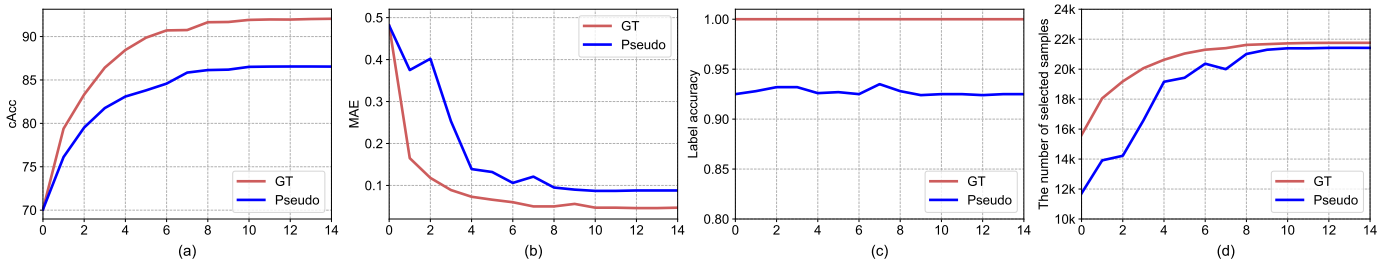

Fig. 12. The comparison between training samples with pseudo and ground-truth labels in terms of (a) cAcc, (b) MAE, (c) label accuracy, (d) the number of selected samples in each training step.

fully, resulting in limited performance. To this end, as shown in Fig. 9, we only remove the low-confidence instances in each image by filling background pixels in the bounding boxes of these objects. As presented in Table III, the aAcc score increases from 86.54% to 88.14% by considering instance-level testing samples. Meanwhile, the experiment in Table I demonstrates the effectiveness of instance-level knowledge distillation (*i.e.*, Instance+Render). Training on additional instance-level samples can gain 1.60%, 1.28%, 1.46% and 1.94% cAcc score improvement in averaged, easy, medium and hard levels, respectively.

V. Conclusion

In this paper, we develop a dual pyramid scale network with an iterative knowledge distillation scheme for automatic check-out. Different from the previous method, we extract the multi-scale representation in both the detection and counting heads of the proposed network, resulting in better efficiency and accuracy. Then, we make full use of image-level and instance-level reliable testing samples to learn the common representation of both training data and testing data gradually. The experiment on the large scale RPC dataset shows that our method outperforms than existing methods, *i.e.*, 88.14% checkout accuracy in the averaged level. For future works, we can explore more efficient iterative training strategy and object detection network with multi-view heads.

References

[1] X. Wei, Q. Cui, L. Yang, P. Wang, and L. Liu, "RPC: A large-scale retail product checkout dataset," *CoRR*, vol. abs/1901.07249, 2019.

[2] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Neural Information Processing Systems*, 2015, pp. 91–99.

[3] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 936–944.

[4] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4203–4212.

[5] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 2242–2251.

[6] L. van der Maaten, "Accelerating t-sne using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[7] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *International Conference on Machine Learning*, 2017, pp. 2988–2997.

[8] C. Li, D. Du, L. Zhang, T. Luo, Y. Wu, Q. Tian, L. Wen, and S. Lyu, "Data priming network for automatic check-out," in *ACM International Conference on Multimedia*, 2019, pp. 2152–2160.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[10] G. Song and W. Chai, "Collaborative learning for deep neural networks," in *Neural Information Processing Systems*, 2018, pp. 1837–1846.

[11] G. Mu, Q. She, Z. Tian, H. Gan, and P. Jiang, "A multi-task collaborative learning method based on auxiliary training and geometric constraints," in *IEEE Industrial Cyber-Physical Systems*, 2018, pp. 79–84.

[12] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool, "Domain adaptive faster R-CNN for object detection in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3339–3348.

[13] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *ACM International Conference on Multimedia*, 2018, pp. 402–410.

[14] C. Jing, Z. Dong, M. Pei, and Y. Jia, "Heterogeneous hashing network for face retrieval across image and video domains," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 782–794, 2019.

[15] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *International Conference on Machine Learning Workshop*, vol. 3, 2013, p. 2.

[16] S. Qian, T. Zhang, and C. Xu, "Cross-domain collaborative learning via discriminative nonparametric bayesian model," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 2086–2099, 2018.

[17] F. Qi, X. Yang, and C. Xu, "A unified framework for multimodal domain adaptation," in *ACM International Conference on Multimedia*, 2018, pp. 429–437.

[18] Y. Zheng, X. Wang, G. Zhang, B. Xiao, F. Xiao, and J. Zhang, "Multi-kernel coupled projections for domain adaptive dictionary learning," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2292–2304, 2019.

[19] X. Ma, T. Zhang, and C. Xu, "Deep multi-modality adversarial networks for unsupervised domain adaptation," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2419–2431, 2019.

[20] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015.

[21] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *International Conference on Learning Representations*, 2015.

[22] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers," in *International Speech Communication Association*, 2017, pp. 3697–3701.

[23] G. Chen, W. Choi, X. Yu, T. X. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Neural Information Processing Systems*, 2017, pp. 742–751.

[24] H. Bagherinezhad, M. Horton, M. Rastegari, and A. Farhadi, "Label refinery: Improving imagenet classification through label progression," *CoRR*, vol. abs/1805.02641, 2018.

[25] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born-again neural networks," in *International Conference on Machine Learning*, 2018, pp. 1602–1611.

[26] Q. Li, S. Jin, and J. Yan, "Mimicking very efficient network for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7341–7349.

[27] G. Ning, Z. Zhang, and Z. He, "Knowledge-guided deep fractal neural networks for human pose estimation," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1246–1259, 2018.

[28] X. Huang and Y. Peng, "TPCKT: two-level progressive cross-media knowledge transfer," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2850–2862, 2019.

[29] S. Tang, L. Feng, W. Shao, Z. Kuang, W. Zhang, and Y. Chen, "Learning efficient detector with semi-supervised adaptive distillation," *CoRR*, vol. abs/1901.00366, 2019.
[30] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge distillation with adversarial samples supporting decision boundary," in *AAAI Conference on Artificial Intelligence*, 2019, pp. 3771–3778.
[31] A. Rocha, D. C. Hauagge, J. Wainer, and S. Goldenstein, "Automatic fruit and vegetable classification from images," *Computers and Electronics in Agriculture*, vol. 70, no. 1, pp. 96–104, 2010.
[32] M. Klasson, C. Zhang, and H. Kjellström, "A hierarchical grocery store image dataset with visual and semantic labels," in *IEEE Winter Conference on Applications of Computer Vision*, 2019, pp. 491–500.
[33] D. Koubaroulis, J. Matas, J. Kittler, and C. CMP, "Evaluating colour-based object recognition algorithms using the soil-47 database," in *Asian Conference on Computer Vision*, vol. 2, 2002.
[34] M. Merler, C. Galleguillos, and S. J. Belongie, "Recognizing groceries in situ using in vitro training data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
[35] M. George and C. Floerkemeier, "Recognizing products: A per-exemplar multi-label image classification approach," in *European Conference on Computer Vision*, 2014, pp. 440–455.
[36] P. Jund, N. Abdo, A. Eitel, and W. Burgard, "The freiburg groceries dataset," *CoRR*, vol. abs/1611.05799, 2016.
[37] P. Follmann, T. Böttger, P. Härtinger, R. König, and M. Ulrich, "Mvtec D2S: densely segmented supermarket dataset," in *European Conference on Computer Vision*, 2018, pp. 581–597.
[38] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *IEEE International Conference on Computer Vision*, 2013, pp. 1841–1848.
[39] Q. Hou, M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 815–828, 2019.
[40] J. Wang, H. Jiang, Z. Yuan, M. Cheng, X. Hu, and N. Zheng, "Salient object detection: A discriminative regional feature integration approach," *International Journal of Computer Vision*, vol. 123, no. 2, pp. 251–268, 2017.
[41] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1091–1100.
[42] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *European Conference on Computer Vision*, 2014, pp. 740–755.
[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Neural Information Processing Systems Workshop*, 2017.
[45] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.
[46] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: towards balanced learning for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 821–830.
[47] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6154–6162.

Libo Zhang received the Ph.D. degree in computer software and theory from University of Chinese Academy of Sciences, Beijing, China, in 2017. He is currently an Associate Research Professor with the Institute of Software Chinese Academy of Sciences, Beijing. He is selected as a member of Youth Innovation Promotion Association, Chinese Academy of Sciences, and Outstanding Youth Scientist of Institute of Software Chinese Academy of Sciences. His current research interests include image processing and pattern recognition.

Dawei Du received his B.S. and M.S. degrees from University of Electronic Science and Technology of China, Chengdu, China, in 2010 and 2013, respectively. He received the Ph.D. degree with the University of Chinese Academy of Sciences, Beijing, China, in 2018. He is currently a Post-Doctoral Researcher with University at Albany, State University of New York, Albany, NY, USA. His current research interests include visual tracking, object detection, video segmentation and digital forensics.

Congcong Li received the B.Eng. degree from University of Electronic Science and Technology of China in 2018. He is currently a master student with University of Chinese Academy of Sciences. His research interests include computer vision and deep learning, particularly focusing on object detection, and domain adaptation.

Yanjun Wu received his B.Eng. degree in computer science from Tsinghua University in 2006, and received the Ph.D. degree in computer science from the Institute of Software Chinese Academy of Sciences (ISCAS), Beijing, China. He is currently a Research Professor with ISCAS. Also, he is the director of Intelligent Software Research Center, ISCAS. His current research interests include computer vision, operating systems and system security.

Tiejian Luo received the Ph.D. degree in computer software and theory from the Graduate University of Chinese Academy of Sciences (UCAS), Beijing, China, in 2001. He is currently a full professor with the School of Computer Science and Technology, UCAS, also he is a Research Professor with Institute of Software Chinese Academy of Sciences. He is the director of information dynamics and engineering application laboratory, UCAS. His current research interests include web mining and deep learning.