Automatic Check-Out via Prototype-Based Classifier Learning from Single-Product Exemplars

Hao Chen[1,2], Xiu-Shen Wei[1,2,3(✉)], Faen Zhang[4], Yang Shen[1], Hui Xu[4], and Liang Xiao[1(✉)]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
weixs@njust.edu.cn, xiaoliang@mail.njust.edu.cn

[2] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China

[3] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

[4] Qingdao AInnovation Technology Group Co., Ltd, Qingdao, China

Abstract. Automatic Check-Out (ACO) aims to accurately predict the presence and count of each category of products in check-out images, where a major challenge is the significant domain gap between training data (single-product exemplars) and test data (check-out images). To mitigate the gap, we propose a method, termed as PSP, to perform Prototype-based classifier learning from Single-Product exemplars. In PSP, by revealing the advantages of representing category semantics, the prototype representation of each product category is firstly obtained from single-product exemplars. Based on the prototypes, it then generates categorical classifiers with a background classifier to not only recognize fine-grained product categories but also distinguish background upon product proposals derived from check-out images. To further improve the ACO accuracy, we develop discriminative re-ranking to both adjust the predicted scores of product proposals for bringing more discriminative ability in classifier learning and provide a reasonable sorting possibility by

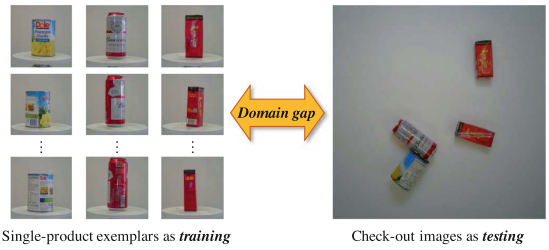
X.-S. Wei and Y. Shen are also with Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, Nanjing University of Science and Technology, China. This work is supported by National Key R&D Program of China (2021YFA1001100), Natural Science Foundation of China under Grant (61871226), Natural Science Foundation of Jiangsu Province of China under Grant (BK20210340), the Fundamental Research Funds for the Central Universities (No. 30920041111, No. NJ2022028), CAAI-Huawei MindSpore Open Fund, Beijing Academy of Artificial Intelligence (BAAI), and Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX22_0464).

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19806-9_16.

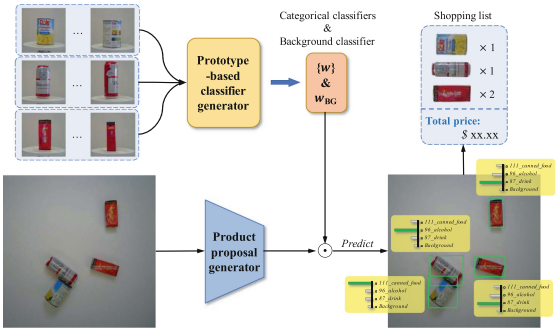
© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
S. Avidan et al. (Eds.): ECCV 2022, LNCS 13685, pp. 277–293, 2022.
https://doi.org/10.1007/978-3-031-19806-9_16

considering the fine-grained nature. Moreover, a multi-label recognition loss is also equipped for modeling co-occurrence of products in check-out images. Experiments are conducted on the large-scale RPC dataset for evaluations. Our ACO result achieves 86.69%, by 6.18% improvements over state-of-the-arts, which demonstrates the superiority of PSP. Our codes are available at https://github.com/Hao-Chen-NJUST/PSP.

Keywords: Automatic check-out · Prototype · Classifier learning



(a) Domain gap between training and testing in ACO



(b) Basic idea of our PSP method

Fig. 1. (a) shows the significant domain gap between single-product exemplars (as training) and check-out images (as testing) in Automatic Check-Out (ACO). (b) illustrates the basic idea of our Prototype-based classifier learning with Single-Product exemplars (PSP) method. To mitigate this gap, we hereby develop a prototype-based classifier generator to learn both product categorical classifiers and a background classifier based product prototypes to recognize product proposals derived from the input check-out image, and finally return a shopping list.

1 Introduction

As a crucial field of intelligent retail [7,8,17,29,35,50], Automatic Check-Out (ACO) [26,39,41,46] has been evolving and promoting in recent years. The basic

design requirement for an automatic check-out system is to automatically identify and count the products purchased by customers. Thus, how to accurately count the fine-grained subordinate categories and quantities of products in check-out images becomes a key task in ACO [21,41,46].

There exist three main challenges in ACO [41], *i.e.*, the large-scale data of products images, the domain gap between training and test (cf. Fig. 1 (a)), and the fine-grained [42,43] characteristics of product categories. To deal with these issues, Wei *et al.* [41] proposed a baseline approach based on object detection to achieve domain adaptation by synthesizing and rendering check-out images. This baseline approach provides a way for dealing with the domain gap in ACO. Then, two latter works [21,46] modified the synthesized-and-rendered strategy of [41] for achieving better domain adaptation ability, and thus improved ACO accuracy. Different from these methods [21,41,46], in this paper, we attempt to mitigate the significant domain gap between single-product exemplars (as training) and check-out images (as test) *by learning corresponding categorical classifiers directly from the categorical prototype of each product*, cf. Fig. 1.

More specifically, the so-called prototype refers to a vector representation that can accurately represents the semantics of the category (*i.e.*, genuine class representation) in the visual space, which is usually realized by the feature center of a specific class [38]. For ACO, beyond the potential of handling domain gap, another advantage of using product prototypes is to avoid the multi-view issue of single-product exemplars. Thanks to the generalization and robustness, compared with using single view or several views from exemplar images, categorical prototypes can more accurately represent category semantics of products. As shown in Fig. 2, we first feed the single-product exemplars $\{I_s^{k,i}\}$ into the proposed prototype-based classifier generator to obtain the categorical classifier $\boldsymbol{w}_k$ corresponding to product category k. Meanwhile, a background classifier is also generated based on all categorical prototypes to learn general patterns and thus distinguish product proposals containing background or incomplete products in check-out images. After that, the learned categorical classifiers and the background classifier can be employed for recognizing product proposals returned by a traditional proposal generation process, *e.g.*, RPN [34].

To further improve the discriminative ability of these learned classifiers, we develop a discriminative re-ranking approach by adjusting the predicted scores of these product proposals, cf. Fig. 2. In concretely, we rank the predicted score of the ground truth category as the highest to improve the prediction confidence, and meanwhile re-rank the score of background as the second highest due to the characteristics of the background classifier (cf. Sect. 3.3). Moreover, by further considering the fine-grained nature of products, we introduce a slack variable as a soft re-ranking strategy to provide a reasonable sorting possibility w.r.t. the predicted scores of fine-grained products. Additionally, we also equip a multi-label recognition loss with PSP for modeling co-occurrence of products in check-out images, which could further bring improvements of the ACO accuracy.

For empirical comparisons, we conduct experiments on the large-scale benchmark dataset in ACO, *i.e.*, RPC [41], by comparing with competing meth-

ods [21,41,46] in the literature. In experiments, we evaluate our PSP method on two detection backbones, *i.e.*, Faster RCNN [34] and Cascade RCNN [2] with Feature Pyramid Network (FPN) [24] for justifying its generalization ability. Our check-out accuracy of Faster RCNN and Cascade RCNN are 86.69% and 92.05%, which are improved by 6.18% and 11.54% over state-of-the-arts. The results of ablation studies are also reported for validating the effectiveness of the main components of PSP.

The main contributions of our work are three-fold. (1) We propose a novel method, *i.e.*, prototype-based classifier learning from single-product exemplars, for dealing with the ACO task, especially for the domain gap. (2) We design a discriminative re-ranking approach to enhance the discriminative ability of these prototype-based classifiers. (3) Experimental results on the benchmark RPC dataset show that our PSP achieves significant improvements over competing methods.

2 Related Work

2.1 Automatic Check-Out

Automatic check-out [41] is a branch of intelligent retail that aims to automatically and accurately detect the products selected by customers and complete the check-out. The popular used ACO datasets in the related works include SOIL-47 [18], Grozi-120 [28], Grocery Products Dataset [9], Freiburg Groceries Dataset [17], MVTec D2S [7] and RPC [41]. To drive the development of ACO, RPC [41] also proposed an object detection baseline. This baseline approach enabled domain adaptation at the data level by segmenting single-product exemplar images to synthesize and render check-out images. Afterwards, Li *et al.* [21] improved the aforementioned baseline method, which restricted the pose of single-product images to create synthesized check-out images. Moreover, Li *et al.* [21] proposed DPNet to accurately detect single-product from the check-out image by collaborating detection and counting. Then, based on DPNet [21] as a pipeline, Yang *et al.* [46] proposed IncreACO with photorealistic exemplar augmentation (PEA). PEA was developed to generate physically reliable and photorealistic check-out images from typical samples scanned for each product, with an incremental learning strategy to match the updated nature of the ACO system and reduce the training effort. Different from the existing ACO methods, our PSP applies prototype-based classifiers derived from single-product exemplars to directly mitigate the domain gap between training and test in ACO. These prototype-based classifiers are able to both recognize fine-grained products in check-out scenario and distinguish background bounding box proposals, without requiring synthesized-rendered process or complicated data augmentation.

2.2 Object Detection

Object detection is one of the basic tasks of computer vision, which includes two sub-tasks of object localization [12] and recognition [27]. The current mainstream

object detection methods include both single-stage [1,3,5,25,27,31–33,49] and two-stage [2,11,12,14,22,24,34,36,45] detection paradigms. Faster RCNN [34] introduced Region Proposal Network (RPN) to generate proposals. FPN [24] built a top-down architecture with lateral connections to extract features across multiple layers. Cascade RCNN [2] constructed a sequence of detection heads trained with increasing Intersection of Union (IoU) thresholds. Double-Head RCNN [45] rethought classification and localization for object detection and proposed the double-head network with a fully-connected head for classification and a convolution head for bounding box regression. I^3Net [3] aimed at learning instance-level features with invariance from different layers for one-stage cross-domain detection. Different from these previous object detection methods, we improve the product proposal classifier for the object detection. Particularly, We develop a prototype-based classifier generator to handle the multi-view issue of single-product exemplars and the domain adaptation issue between exemplars with check-out images.

2.3 Classifier Boundary Transformation

Classifier boundary transformation is a kind of strategies for learning discriminative classifier boundaries, which is always conducted within the meta learning framework. Several related work [19,40,44] involved the boundary transformation where a mapping from a decision boundary (or a sample) to another decision boundary is learned. Specifically, in early years, Wang and Hebert [40] proposed a model regression network which learns how to transform from a classifier trained with large-scale data to another one trained with easy small samples. Wei *et al.* [44] utilized the meta learning method to create piecewise classifiers for few-shot fine-grained classification. In CLEAR [19], the authors proposed a cumulative learning approach by meta learning for the one-shot one-class classification task. Different from these previous work, our PSP method is able to directly generate both categorical classifiers and the background classifier, rather than requiring the meta learning process, which shows its efficiency and practicality.

3 Methodology

3.1 Overall Framework and Notations

As shown in Fig. 2, our proposed PSP method consists of two crucial components, *i.e.*, prototype-based classifier generator and discriminative re-ranking. More specifically, the prototype-based classifier generator is developed to map product prototypes derived from single-product exemplars towards both categorical classifiers and a background classifier. Such a mapping is designed to handle not only the multi-views issue of exemplar images but also the domain change issue between training and test of ACO, cf. Fig. 1 (a). The discriminative re-ranking component could adjust the predicted scores of the corresponding

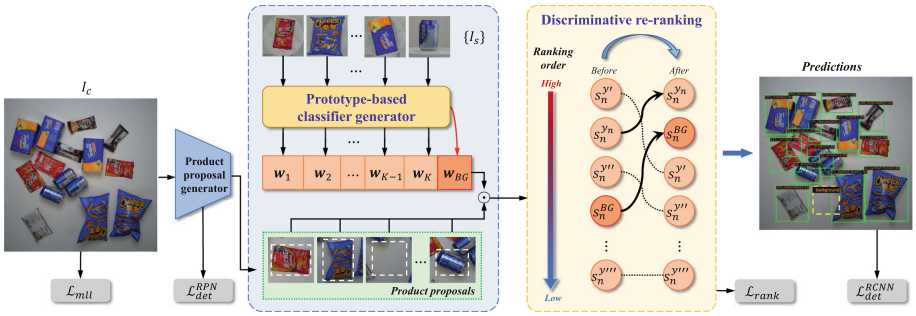


Fig. 2. Overall framework of our PSP method. Regarding the check-out image I_c, after product proposal generation, a set of product proposals is obtained, which involves both effective proposals and non-effective proposals (*i.e.*, background or incomplete products). By performing the proposed prototype-based classifier generator based on the prototype of single-product exemplars $\{I_s^{k,i}\}$, we have categorical classifiers (*i.e.*, $\boldsymbol{w}_k$, $k \in \{1, \ldots, K\}$) for recognizing these fine-grained products of K classes in I_c, as well as a background classifier (*i.e.*, $\boldsymbol{w}_{BG}$) for distinguishing background proposals containing no products. Based on these learned classifiers, for the n-th proposal, a series of corresponding predicted scores is returned as $s_n^{\cdot}$. To further improve the discriminative ability of $\{\boldsymbol{w}\}$, we develop discriminative re-ranking to push the score $s_n^{y_n}$ highest and the score s_n^{BG} the second highest (cf. Sect. 3.3). Particularly, $s_n^{y_n}$ is the predicted score of its ground truth label y_n, and s_n^{BG} represents the probability as predicted as background. (Best viewed in colors.) (Color figure online)

proposals, which further improves the discriminative ability of the learned classifiers.

In the following subsections, we present these components in details, and introduce the notations as follows.

Let $\mathcal{S} = \{I_s^{k,i}\}$ $(k \in \{1, \ldots, K\}, i \in \{1, \ldots, N_s^k\})$ denote the set of the single-product exemplar images, where $I_s^{k,i}$ indicates the i-th single-product exemplar of the k-th product category. K is the number of product categories and N_s^k is the number of exemplar images belonging to category k. Also, we notate the check-out image set as $\mathcal{C} = \{(I_c, Y_c)\}$, where Y_c includes the category label and the bounding box coordinate parameters of each product. Let $\bar{\boldsymbol{x}}_k$ denote the product prototype representation of the k-th category, and $\bar{\boldsymbol{x}}_{BG}$ represents the prototype feature of the background. Regarding the learned classifiers, let $\boldsymbol{w}_k$ denote the classifier weights of the k-th categorical prototype-based classifier, and $\boldsymbol{w}_{BG}$ is the classifier corresponding to the background. In addition, $\boldsymbol{m}_n$ indicates the feature of the n-th product proposal from an check-out image I_c.

3.2 Prototype-Based Classifier Generation

As illustrated in Fig. 2, given an input check-out image I_c, a product proposal generator firstly returns product proposals $\boldsymbol{m}_n$ from I_c. These product proposals (including proposals containing background or incomplete products) are then

categorized by the generated prototype-based classifiers $\{\boldsymbol{w}\}$. Regarding $\{\boldsymbol{w}\}$, in concretely, we first obtain the categorical prototypes $\bar{\boldsymbol{x}}_k$ of category k based on single-product exemplars $\{I_s^{k,i}\}$ of each product category as training, which is formulated as

$$\bar{\boldsymbol{x}}_k = \frac{1}{N_s^k} \sum_{i=1}^{N_s^k} \left[f_{GMP}(f_{CNN}(I_s^{k,i}; \Theta) \right] , \tag{1}$$

where $f_{CNN}(\cdot; \Theta)$ represents the backbone CNN model (a pre-trained ResNet-50 [15] by feeding with merely N_s^k product exemplars for category k) for extracting features of $\{I_s^{k,i}\}$, $f_{GMP}(\cdot)$ is the global max-pooling function to aggregate and obtain the categorical prototypes, and Θ indicates the model parameters. These prototypes are used for both training and testing processes. Then, we can generate categorical classifiers $\boldsymbol{w}_k$ based on $\bar{\boldsymbol{x}}_k$.

Beyond $\boldsymbol{w}_k$ for recognizing fine-grained products, it is also desirable to learn a classifier for distinguishing non-effective proposals, *i.e.*, bounding box proposals containing no or incomplete products (aka "background proposals"). However, there is no exemplar of "background proposals" in the training data $\{I_s^{k,i}\}$ to learn such a classifier. Therefore, we average the category prototypes of all product categories as the "background prototype", since the background prototype should correspond to a meta concept with general patterns, rather than specific product concepts with discriminative patterns. The background prototype $\bar{\boldsymbol{x}}_{BG}$ can be represented by

$$\bar{\boldsymbol{x}}_{BG} = \frac{1}{K} \sum_{k=1}^{K} \bar{\boldsymbol{x}}_k \, . \tag{2}$$

After obtaining both $\bar{\boldsymbol{x}}_k$ and $\bar{\boldsymbol{x}}_{BG}$, they are fed into the prototype-based classifier generator $f_{generator}(\cdot; \theta)$ to learn the corresponding categorical classifiers $\boldsymbol{w}_k$ and the background classifier $\boldsymbol{w}_{BG}$ as outputs:

$$\boldsymbol{w}_k = f_{generator}(\bar{\boldsymbol{x}}_k; \theta) \, , \tag{3}$$

$$\boldsymbol{w}_{BG} = f_{generator}(\bar{\boldsymbol{x}}_{BG}; \theta) \, , \tag{4}$$

where θ is the parameter of the generator and it is realized as a multi-layer perceptron with two fully connected layers plus ReLU as the activation function.

Finally, the weight matrix $\boldsymbol{W} = [\boldsymbol{w}_1; \boldsymbol{w}_2; \ldots; \boldsymbol{w}_K; \ \boldsymbol{w}_{BG}]$ is obtained and employed for recognizing the proposals $\boldsymbol{m}_n$ as fine-grained product categories or background by

$$\boldsymbol{s}_n = \boldsymbol{W} \cdot \boldsymbol{m}_n \in \mathbb{R}^{K+1} \, , \tag{5}$$

where the elements in $\boldsymbol{s}_n$ are the predicted scores belonging to the product categories or as background.

3.3 Discriminative Re-Ranking

Based on the predicted scores $\boldsymbol{s}_n$, the predictions of these proposals can be obtained by ranking $\boldsymbol{s}_n$, where the corresponding category with the highest score indicates the predicted category.

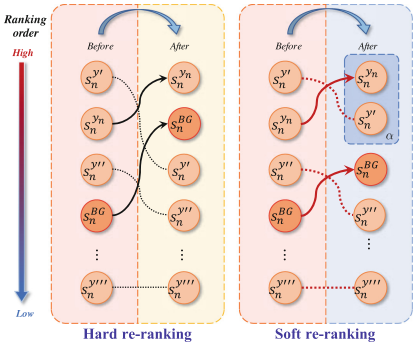


Fig. 3. Proposed discriminative re-ranking. The original ranking order of predicted scores is directly obtained by Eq. (5). Compared with hard re-ranking, our soft re-ranking not only re-ranks the predicted scores for bringing more discriminative ability in classifier learning, but also provides a more reasonable sorting possibility for the predicted scores of fine-grained products.

In order to further achieve a better discriminative ability during training, we propose to adjust the score of the ground truth category (*i.e.*, $s_n^{y_n}$) as the highest and the score of background (*i.e.*, s_n^{BG}) as the second highest, while other scores of the rest categories are smaller than $s_n^{y_n}$ and s_n^{BG}. The process can be formulated by

$$\begin{aligned} s_n^{y_n} &> s_n^k, \quad \forall k \in \Omega - \{y_n\}, \\ s_n^{BG} &> s_n^j, \quad \forall j \in \Omega - \{y_n, BG\}, \end{aligned} \tag{6}$$

where y_n is the ground truth category of the n-th product proposal, and $\Omega = \{1, 2, \ldots, K-1, K, BG\}$.

In particular, it is straightforward to rank $s_n^{y_n}$ as the highest. The reason why we set up the constraint of ranking s_n^{BG} as the second highest is that: the background classifier $\boldsymbol{w}_{BG}$ is derived from the *meta* prototype $\bar{\boldsymbol{x}}_{BG}$ (containing information of all categories). Thus, regarding a proposal $\boldsymbol{m}_n$, its value of dot product upon $\boldsymbol{w}_{BG}$ is natural expected to be larger than the values upon other categorical classifiers, since the ability of discriminative classifiers is to clearly distinguish different product categories, even similar categories.

Another advantage of Eq. (6) is that: for the background proposal, its highest predicted score is s^{BG}. Therefore, we can use it as a clue to filter out the false positive bounding box predictions in check-out images during testing.

However, in ACO, products are always belonging to subordinate categories, aka *fine-grained* categories. That is to say that, there exist product categories from one meta-category are extremely visually similar, cf. Fig. (2) in the supplementary materials. For these sub-categories, the re-ranking constraint in Eq. (6) might be not strictly obeyed. Thus, in order to not only deal with the fine-grained nature of products, but also remain the discriminative ability of re-ranking, we modify Eq. (6) into a ***soft re-ranking*** by introducing a slack variable ϵ, which

is as follows:

$$\begin{aligned} & s_n^{y_n} > s_n^k, \quad \forall k \in \Omega - \{y_n\}, \\ & s_n^{j'} > s_n^{BG} - \epsilon, \quad \forall j' \in \Omega', \\ & s_n^{BG} > s_n^{j''}, \quad \forall j'' \in \Omega - \{y_n, BG\} - \Omega', \end{aligned} \tag{7}$$

where $\Omega' = \{j' | s_n^{j'} > \alpha \cdot \max s_n^j, \forall j\}$ and α is a hyper-parameter in the range of $(0, 1)$. For more clearly presentations, we show the proposed discriminative re-ranking approach in Fig. 3. Specifically, compared with our soft re-ranking, the re-ranking constraint in Eq. (6) can be termed as ***hard re-ranking***.

Collaborated with Eq. (7), the discriminative re-ranking process can be written as a loss function by

$$\mathcal{L}_{rank} = \sum_{n=1}^{N} \sum_{k=1}^{K+1} y_{cls}^{k,n} s_n^k + (1 - y_{cls}^{k,n}) \left[s_n^{BG} - s_n^k \right]_+ + C\epsilon, \tag{8}$$

where $y_{cls}^{k,n} \in \{0, 1\}$ is the k-th category ground truth label of the n-th proposal, $[\cdot]$ indicates the hinge function $\max(0, \cdot)$, and C is the penalty factor on the slack variable. In addition, except for the re-ranking loss function, the whole model of PSP is further driven by the traditional regression/classification loss in detection, as well as a multi-label learning loss for modeling the correlation of purchased products (*i.e.*, the co-occurrence), cf. Sect. 3.4.

3.4 Loss Functions

Detection Loss. According to the general two-stage object detection framework [13,34], the detection loss $\mathcal{L}_{det}$ in our PSP consists of two parts as well, i.e., the region proposal network (RPN) [34] detection loss $\mathcal{L}_{det}^{RPN}$ in the first stage and the RCNN detection loss $\mathcal{L}_{det}^{RCNN}$ in the second stage. In particular, the detection loss of each stage contains both losses of classification and regression for proposals. For more details of the detection loss, please refer to [13,34] for detailed techniques.

Multi-label Loss. Multi-label recognition is a fundamental computer vision topic in the literature [20,47], which aims to model the label dependencies among multiple labels associated with images to improve the recognition performance. Recently, multi-label recognition shows its advantages in object detection tasks, *e.g.*, [6]. We hereby introduce the multi-label loss upon the check-out images I_c to model the correlation of co-occurrence products in the check-out scenario. More specifically, we denote $\boldsymbol{F}$ as the feature map of the last layer from I_c, and utilize a fully connected layer $f_{FC}(\cdot; \phi)$ with its parameter ϕ to conduct multi-label recognition, which can be presented by

$$\hat{\boldsymbol{y}}_{mll} = f_{FC}(f_{GAP}(\boldsymbol{F}) + f_{GMP}(\boldsymbol{F}); \phi), \tag{9}$$

where $f_{GAP}(\cdot)$ and $f_{GMP}(\cdot)$ are global average- and max-pooling for aggregating $\boldsymbol{F}$ into a single vector. In general, GAP captures common information while

GMP captures the most salient information. In Eq. (9), the sum of GAP and GMP can jointly exploit these two complementary information. $\hat{\boldsymbol{y}}_{mll}$ is the multi-label predictions and its each element is a confidence score. We assume that the ground truth label of an image is $\boldsymbol{y}_{mll}$, where $y^k_{mll} = \{0, 1\}$ denotes whether product of category k appears in the image or not. Thus, the multi-label loss $\mathcal{L}_{mll}$ can be formulated as

$$\mathcal{L}_{mll} = \sum_{k=1}^{K} y^k_{mll} \log(\sigma(\hat{y}^k_{mll})) + (1 - y^k_{mll}) \log(1 - \sigma(\hat{y}^k_{mll})) \,, \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function.

Finally, the whole network of our PSP is driven by

$$\mathcal{L} = \mathcal{L}_{det} + \mathcal{L}_{rank} + \mathcal{L}_{mll} \,, \tag{11}$$

where the weight of each loss is the same because they are equally important and be set as the default setting. Additionally, for performing post-processing, we apply a class-agnostic NMS [37] to keep the bounding boxes with the top confidence score for similar locations, and filter out the false positives.

4 Experiments

4.1 Dataset

We adopt the large-scale Retail Product Check-out (RPC) dataset [41] as benchmarks to evaluate our method. It includes 200 categories and 83,739 images, where these 200 fine-grained categories belong to 17 meta-categories. It means that the RPC dataset is currently the largest dataset in ACO tasks, which is practical and challenging. The training data of RPC contains 53,739 single-product images in total, where each image has a particular instance of a product category. The RPC test data contains three sub-sets with different clutter modes, each containing 10,000 images. Notably, according to the number of items in an image, the three clutter modes in the RPC dataset consist of `easy` (3∼5 categories and 3∼10 instances), `medium` (5∼8 categories and 10∼15 instances), and `hard` (8∼10 categories and 15∼20 instances). In our experiments, the single-product image set is applied to obtain the prototype-based classifiers, and 6,000 (RPC valid set) and 24,000 (RPC test set) check-out images are used for training and test, respectively. Meanwhile, we apply the four evaluation metrics [41] to measure the advantages and disadvantages between our method and the comparison methods, including check-out accuracy (cAcc), average counting distance (ACD), mean category counting distance (mCCD), mean category intersection of union (mCIoU). The specific formulas are given in the supplementary materials. In addition, other related datasets such as Products-6K [10] and Product1M [48] lack the necessary conditions to satisfy the ACO task or our PSP design, *e.g.*, check-out images or bounding box annotations.

4.2 Baseline Methods and Implementation Details

Baseline Methods. Specifically, we adopt the method of [41], DPNet [21] and IncreACO [46] as the baselines. [41] first proposed an object detection based ACO pipeline to locate and count products by detecting them in check-out images. Specifically, the backbone of object detection is Faster RCNN [34] with Feature Pyramid Networks (FPN) [24]. To reduce the gap where the check-out image has multiple objects and the training exemplar has only one, [41] proposed a *copy and paste* method to synthesize the image, denoted as *Syn*. In addition, *Render* was used to represent rendered synthesized images using Cycle-GAN [51] for more realistic lighting and shading effects. Then, DPNet [21] modified and improved this baseline. DPNet [21] was an improved FPN [24] and Mask RCNN [13] detector with both detection and counting heads, which is optimized by collaborative detection and counting learning. In particular, DPNet designed a pose pruning step when synthesizing the images to remove exemplars with too small relative areas. IncreACO [46] improved the DPNet by incremental learning LwF [23] and knowledge distillation [16], where the output of the teacher provides the student with the soft label of the old categories. All three approaches utilize a large amount of synthesized and rendered data to train the detection network to achieve domain adaptation at the data level.

Implementation Details. Our proposed method is implemented by PyTorch [30] and MMDetection [4]. The relevant parameters in experiments basically follow the default settings in [4] and the backbone network we use is ResNet-50 [15]. In concretely, we set the size of input images to 800×800. The number of N_s^k is all set to 8. The total number of training epochs is 36. While, different from existing methods [21,41,46], our method only requires a smaller iteration time until convergence. The learning rate, momentum and weight decay of the stochastic gradient descent optimizer are initialized to 0.0025, 0.9 and 10^{-4}, respectively. α, C and ϵ are set in 0.97, 5 and 10^{-6} in Eq. (8), respectively. Our training strategy is to decrease the learning rate as 0.1 time of the previous one at the 24-th and 33-th epoch. All experiments are conducted with a GeForce RTX 2080 Ti GPU card (11G memory).

4.3 Main Results

Table 1 presents the four evaluation metrics of our method and the existing methods. All the three existing methods have the *Syn+Render* training setting. Specifically, the *Syn+Render* training setting trains the detector with 100,000 synthesized images and 100,000 rendered images simultaneously. Meanwhile, "*Faster RCNN + FPN*" and "*Cascade RCNN + FPN*" represent the Faster RCNN [34] and Cascade RCNN [2] frameworks with the addition of the FPN module, respectively. We apply our proposed method on basis of these two object detection frameworks. In addition, the structure of "*Faster RCNN + FPN*" is the closest to that of existing methods. For fair comparisons, we report the results of different clutter modes. As shown in this table, our method achieves

the best experimental results in all clutter modes and evaluation metrics. For example, in the average clutter mode, the cAcc measure of our PSP method is 86.69%, which is an improvement of 6.18% compared to the previous optimal result. The cAcc results of "*Faster RCNN + FPN*" in easy, medium, and hard modes are improved by 1.80%, 6.60%, and 9.91%, over the results of the previous state-of-the-art methods when analyzed for each of the three clutter modes. The amount of improvement, especially in the hard and medium modes, is very significant. In addition, we use our method on a more powerful object detection framework, *i.e.*, "*Cascade RCNN + FPN*", with a cAcc result of 92.05%, which is 5.36% higher than the result of "*Faster RCNN + FPN*".

4.4 Ablation Studies

Table 1. Experimental results on the RPC dataset. "↑" indicates that the larger the value, the better, and "↓" indicates that the smaller the value, the better. The best results are marked in **red** and the second best results are in **blue**.

Clutter mode	Methods	cAcc↑	ACD↓	mCCD↓	mCIoU↑
Easy	Wei *et al.* [41] (Syn+Render)	73.17%	0.49	0.07	93.66%
	IncreACO [46] (Syn+Render)	88.06%	0.21	0.03	96.95%
	DPNet [21] (Syn+Render)	90.32%	**0.10**	**0.02**	97.87%
	PSP (Faster RCNN + FPN)	**92.12%**	0.11	**0.02**	**98.40%**
	PSP (Cascade RCNN + FPN)	**94.90%**	**0.08**	**0.01**	**98.88%**
Medium	Wei *et al.* [41] (Syn+Render)	54.69%	0.90	0.08	92.95%
	IncreACO [46] (Syn+Render)	77.31%	0.40	0.03	96.82%
	DPNet [21] (Syn+Render)	80.68%	0.32	0.03	97.38%
	PSP (Faster RCNN + FPN)	**87.28%**	**0.18**	**0.01**	**98.59%**
	PSP (Cascade RCNN + FPN)	**92.94%**	**0.11**	**0.01**	**99.12%**
Hard	Wei *et al.* [41] (Syn+Render)	42.48%	1.28	0.07	93.06%
	IncreACO [46] (Syn+Render)	66.14%	0.64	0.04	96.35%
	DPNet [21] (Syn+Render)	70.76%	0.53	0.03	97.04%
	PSP (Faster RCNN + FPN)	**80.67%**	**0.29**	**0.02**	**98.29%**
	PSP (Cascade RCNN + FPN)	**88.33%**	**0.19**	**0.01**	**98.88%**
Averaged	Wei *et al.* [41] (Syn+Render)	56.68%	0.89	0.07	93.19%
	IncreACO [46] (Syn+Render)	77.15%	0.41	0.03	96.72%
	DPNet [21] (Syn+Render)	80.51%	0.34	0.03	97.33%
	PSP (Faster RCNN + FPN)	**86.69%**	**0.19**	**0.02**	**98.44%**
	PSP (Cascade RCNN + FPN)	**92.05%**	**0.13**	**0.01**	**98.98%**

Effectiveness of Our Proposed PSP. Beyond the main results, we also perform ablation studies to determine the effectiveness of main components of PSP with "*Faster RCNN + FPN*". Table 2 reports the results in the averaged clutter mode. Specifically, "Prototype" indicates that the inputs of our proposed prototype-based classifier learning are either single-product "Exemplars" or ground truth product instances in "Check-out" images. Comparisons on "Exemplars" and "Check-out" can reveal the ability of our PSP on handling the domain gap challenge. Besides, "Re-ranking" in Table 2 denotes our discriminative re-ranking strategy, including hard re-ranking or soft re-ranking, cf.

Table 2. Experimental results of ablation studies. "↑" indicates that the larger the value, the better, and "↓" indicates that the smaller the value, the better. "Exemplars" denotes that a prototype-based classifier is generated from single-product exemplars, while "Check-out" denotes a prototype-based classifier generated from ground truth product instances in check-out images. Best results are marked in **bold**.

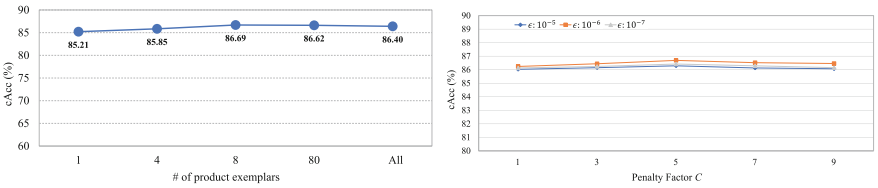
Methods				cAcc↑	ACD↓	mCCD↓	mCIoU↑
	Prototype	Re-ranking	Multi-label loss				
FC	–	–	–	82.87%	0.31	0.02	98.04%
# 1	Check-out	–	–	85.31%	0.22	0.02	98.26%
# 2	Exemplars	–	–	85.12%	0.22	0.02	98.28%
# 3	Check-out	Hard	–	85.60%	0.21	0.02	98.34%
# 4	Check-out	Soft	–	85.56%	0.21	0.02	98.30%
# 5	Exemplars	Hard	–	85.75%	0.21	0.02	98.34%
# 6	Exemplars	Soft	–	85.93%	0.20	0.02	98.37%
# 7	Check-out	Hard	✓	85.90%	0.20	0.02	98.39%
# 8	Check-out	Soft	✓	85.65%	0.21	0.02	98.31%
# 9	Exemplars	Hard	✓	85.90%	0.20	0.02	98.40%
# 10	Exemplars	Soft	✓	**86.69%**	**0.19**	0.02	**98.44%**

Sect. 3.3. In addition, "Multi-label loss" represents whether using the multi-label loss for capturing the co-occurrence of products in check-out images. Meanwhile, the "FC" row is the baseline with a fully-connected layer as the bounding box classifier.

As the results shown in that table, comparing the results of "FC" and row #2, the effectiveness of our prototype-based classifier learning is validated, *i.e.*, cAcc is improved by 2.25%. Similarly, we can justify the effect of "Re-ranking" and "Multi-label loss" by comparing the results of row #2, #6 and #10, which can improve the cAcc results by 0.81% and 0.76%, respectively. Such improvements of cAcc are significant. In addition, comparing the results in row #9 and #10, it can be found that soft re-ranking performs 0.79% higher than hard re-ranking, which shows the effectiveness and rationality of considering the fine-grained nature by introducing the slack variable. Moreover, comparing #8 with #10, the classifiers generated from the single-product exemplars have 1.04% improvement with these from the check-out instances.

Overall, our PSP improves the cAcc results by 3.82% over the baseline methods in Table 2. Furthermore, we also verify the effectiveness of PSP for mitigating the domain gap challenge. In particular, we generate another prototype-based classifier by feeding the ground truth product instances from the check-out images. Comparing these cAcc results from row #3 to #10, it can be observed the learned classifiers based on "Exemplar" perform equally or even slightly better than the classifiers based on "Check-out" in different empirical settings. The phenomenons demonstrate that our PSP can effectively mitigate the domain gap between the source and target domains. In addition, we discuss the domain adaptation mechanism of PSP in the supplementary materials.

Sensitivity to Hyper Parameters N_s^k, C and ϵ. It is also noteworthy to know that how the exemplars and their number are collected will affect the per-



(a) N_s^k (b) C and ϵ

Fig. 4. The analysis results of these three hyper parameters N_s^k, C and ϵ. (a) is the cAcc results of different numbers of exemplars N_s^k and (b) is the cAcc results of different settings of the penalty factor C and the slack variable ϵ.

formance of the prototype-based classifiers. Thus, we compare the cAcc results of different numbers of exemplars N_s^k (*i.e.*, 1, 4, 8, 80 and all exemplars) collected by certain rules, which is reported in Fig. 4 (a). The specific sampling rules for these five cases are as follows. (1) $N_s^k = 1$: Only one exemplar is randomly selected to form the k category prototype. (2) $N_s^k = 4$: Four exemplars at a certain angle are selected based on the front and back of the product and two of the camera viewpoints. (3) $N_s^k = 8$: All the four camera viewpoints at the same angle are considered to sample 8 exemplars compared to $N_s^k = 4$. (4) $N_s^k = 80$: Based on the rule of $N_s^k = 8$, a total of 80 exemplars with 10 randomly selected angles are used to obtain the k-th category prototype. (5) $N_s^k = All$: All exemplars of the k-th category are utilized to generate the prototype. It is apparent from Fig. 4 (a) that the prototype-based classifiers perform best when N_s^k is set to 8. As analyzed, the lack of exemplar features (*i.e.*, $N_s^k \leq 8$) may result in that the prototypes are under-represented for distinguishing subtle visual differences, while redundant multi-angle exemplar features (*i.e.*, $N_s^k > 8$) might caused redundancies in exemplar feature representation, leading to a reduction in classifier performance. Those might be the reasons why N_s^k is too small or large will cause slightly cAcc drops. Moreover, the cAcc results of different N_s^k values show that our PSP is not sensitive to N_s^k. Except for using only 1 exemplar, there is no significant difference between 4, 8, 80 and all exemplars (the precise cAcc gap is less than 0.8%). Meanwhile, even using 1 exemplar, the cAcc score (85.21%) is greatly larger than state-of-the-art results (*e.g.*, 80.51% of DPNet).

Subsequently, we similarly analyze the sensitivity of the penalty factor C and the slack variable ϵ in Eq. (8). As shown in Fig. 4 (b), the values of C and ϵ are $1, 3, 5, 7, 9$ and $10^{-5}, 10^{-6}, 10^{-7}$, respectively. We can see that our soft re-ranking is not sensitive to the values of C and ϵ, and the difference between the best and worst cAcc should not exceed 0.7%. Moreover, we show some other qualitative results and the analyses of working mechanism in the supplementary materials.

5 Conclusion

In this paper, we proposed a prototype-based classifier learning from single-product exemplars with a discriminative re-ranking for Automatic Check-Out.

The prototype-based classifier learning was developed to mitigate the domain gap between exemplars as training and check-out images as test. Furthermore, we designed a discriminative re-ranking to improve ACO accuracy by bringing more discriminative ability in classifier learning. Also, the multi-label loss was applied to model the co-occurrence of products in check-out images. On the large-scale benchmark RPC dataset, our PSP achieved 86.69% ACO accuracy, which outperformed previous competing methods by 6.18%. In the future, we will investigate PSP for class incremental learning to meet the practical and realistic requirement in ACO.

Acknowledgments. The authors would like to thank the anonymous reviewers for their critical and constructive comments and suggestions. We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
2. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. In: CVPR, pp. 6154–6162 (2018)
3. Chen, C., Zheng, Z., Huang, Y., Ding, X., Yu, Y.: I3Net: implicit instance-invariant network for adapting one-stage object detectors. In: CVPR, pp. 12576–12585 (2021)
4. Chen, K., et al.: MMDetection: open MMLab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
5. Chen, Q., Wang, Y., Yang, T., Zhang, X., Cheng, J., Sun, J.: You only look one-level feature. In: CVPR, pp. 13039–13048 (2021)
6. Chen, Z.M., Jin, X., Zhao, B., Wei, X.S., Guo, Y.: Hierarchical context embedding for region-based object detection. In: ECCV, pp. 633–648 (2020)
7. Follmann, P., Bottger, T., Hartinger, P., Konig, R., Ulrich, M.: MVTec D2S: densely segmented supermarket dataset. In: ECCV, pp. 569–585 (2018)
8. Frontoni, E., Raspa, P., Mancini, A., Zingaretti, P., Placidi, V.: Customers' activity recognition in intelligent retail environments. In: ICIAP, pp. 509–516 (2013)
9. George, M., Floerkemeier, C.: Recognizing products: a per-exemplar multi-label image classification approach. In: ECCV, pp. 440–455 (2014)
10. Georgiadis, K., et al.: Products-6K: a large-scale groceries product recognition dataset. In: PETRA, pp. 1–7 (2021)
11. Girshick, R.: Fast R-CNN. In: CVPR, pp. 1440–1448 (2015)
12. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR, pp. 580–587 (2014)
13. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: ICCV, pp. 2961–2969 (2017)
14. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE TPAMI **37**(9), 1904–1916 (2015)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR abs/1503.02531 (2015)

17. Jund, P., Abdo, N., Eitel, A., Burgard, W.: The freiburg groceries dataset. CoRR abs/1611.05799 (2016)
18. Koubaroulis, D., Matas, J., Kittler, J.: Evaluating colour-based object recognition algorithms using the SOIL-47 database. In: ACCV, pp. 840–845 (2002)
19. Kozerawski, J., Turk, M.: CLEAR: cumulative learning for one-shot one-class image recognition. In: CVPR, pp. 3446–3455 (2018)
20. Lapin, M., Hein, M., Schiele, B.: Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification. IEEE TPAMI **40**(7), 1533–1554 (2018)
21. Li, C., Du, D., Zhang, L., Luo, T., Wu, Y., Tian, Q., Wen, L., Lyu, S.: Data priming network for automatic check-out. In: ACM MM, pp. 2152–2160 (2019)
22. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Light-head R-CNN: in defense of two-stage object detector. arXiv preprint arXiv:1711.07264 (2017)
23. Li, Z., Hoiem, D.: Learning without forgetting. IEEE TPAMI **40**(12), 2935–2947 (2018)
24. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR, pp. 936–944 (2017)
25. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: ICCV, pp. 2980–2988 (2017)
26. Liu, A., Wang, J., Liu, X., Cao, B., Zhang, C., Yu, H.: Bias-based universal adversarial patch attack for automatic check-out. In: ECCV, pp. 395–410 (2020)
27. Liu, W., et al.: SSD: single shot multibox detector. In: ECCV, pp. 21–37 (2016)
28. Merler, M., Galleguillos, C., Belongie, S.: Recognizing groceries in situ using in vitro training data. In: CVPR, pp. 1–8 (2007)
29. Paolanti, M., Liciotti, D., Pietrini, R., Mancini, A., Frontoni, E.: Modelling and forecasting customer navigation in intelligent retail environments. JINT **91**(2), 165–180 (2018)
30. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: NeurIPS, pp. 8026–8037 (2019)
31. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only look once: unified, real-time object detection. In: CVPR, pp. 779–788 (2016)
32. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: CVPR, pp. 7263–7271 (2017)
33. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
34. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NeurIPS, pp. 91–99 (2015)
35. Sciucca, L.D., Manco, D., Contigiani, M., Pietrini, R., Bello, L.D., Placidi, V.: Shoppers detection analysis in an intelligent retail environment. In: ICPR, pp. 534–546 (2021)
36. Tan, Z., Nie, X., Qian, Q., Li, N., Li, H.: Learning to rank proposals for object detection. In: ICCV, pp. 8273–8281 (2019)
37. Tychsen-Smith, L., Petersson, L.: Improving object localization with fitness NMS and bounded iou loss. In: CVPR, pp. 6877–6885 (2018)
38. Vieville, T., Crahay, S.: Using an hebbian learning rule for multi-class SVM classifiers. J. Comput. Neurosci. **17**(3), 271–287 (2004)
39. Wang, Q., Liu, X., Liu, W., Liu, A.A., Liu, W., Mei, T.: MetaSearch: incremental product search via deep meta-learning. IEEE TIP **29**, 7549–7564 (2020)
40. Wang, Y.X., Hebert, M.: Learning to learn: model regression networks for easy small sample learning. In: ECCV, pp. 616–634 (2016)

41. Wei, X.S., Cui, Q., Yang, L., Wang, P., Liu, L., Yang, J.: RPC: a large-scale and fine-grained retail product checkout dataset. Sci. China Inf. Sci. (2022). https://doi.org/10.1007/s11432-022-F3513-y
42. Wei, X.S., Shen, Y., Sun, X., Ye, H.J., Yang, J.: A^2-Net: Learning attribute-aware hash codes for large-scale fine-grained image retrieval. In: NeurIPS, pp. 5720–5730 (2021)
43. Wei, X.S., et al.: Fine-grained image analysis with deep learning: a survey. IEEE TPAMI (2021). https://doi.org/10.1109/TPAMI.2021.3126648
44. Wei, X.S., Wang, P., Liu, L., Shen, C., Wu, J.: Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples. IEEE TIP **28**(12), 6116–6125 (2019)
45. Wu, Y., et al.: Rethinking classification and localization for object detection. In: CVPR, pp. 10186–10195 (2020)
46. Yang, Y., Sheng, L., Jiang, X., Wang, H., Xu, D., Cao, X.B.: IncreACO: incrementally learned automatic check-out with photorealistic exemplar augmentation. In: WACV, pp. 626–634 (2021)
47. Yeh, M.C., Li, Y.N.: Multilabel deep visual-semantic embedding. IEEE TPAMI **42**(6), 1530–1536 (2020)
48. Zhan, X., et al.: Product1M: towards weakly supervised instance-level product retrieval via cross-modal pretraining. In: ICCV, pp. 11782–11791 (2021)
49. Zhang, X., Wan, F., Liu, C., Ji, R., Ye, Q.: FreeAnchor: learning to match anchors for visual object detection. In: NeurIPS, pp. 147–155 (2019)
50. Zhao, L., Yao, J., Du, H., Zhao, J., Zhang, R.: A unified object detection framework for intelligent retail container commodities. In: ICIP, pp. 3891–3895 (2019)
51. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV, pp. 2223–2232 (2017)