







# Split-Check: Boosting Product Recognition via Instance-Level Retrieval

Chengxu Liu , Graduate Student Member, IEEE, Zongyang Da , Yuanzhi Liang , Yao Xue , Guoshuai Zhao , Member, IEEE, and Xueming Qian , Member, IEEE

**Abstract**—AI-based methods are shining across a variety of industries, especially unmanned retail. Product recognition is the problem of recognizing the category and quantity of products (e.g., beverages and mineral water) in intelligent unmanned vending machines (UVMs) to automatic checkout during purchase. However, for similar products in hundreds of categories, the existing method is not accurate enough. Besides, they cannot be extended for new products without retraining. In this article, we propose a product recognition approach based on intelligent UVMS, called **Split-Check**, which first splits the region of interest of products by detection and then check product by instance-level retrieval. **Split-Check** is the combination of two important components. The preliminary detection distinguishes items that contain the different coarse-grained features, then locates items, and classifies them into coarse-grained categories as a candidate. The retrieval further distinguishes the candidate items that contain the different fine-grained features. Besides, we reconstruct a large-scale categories product dataset **GOODS-85** based on actual UVMS scenarios, in which the number of categories of items is larger than the existing dataset. Experimental results demonstrate

the effectiveness of the proposed approach. Our method significantly improves the recognition performance of hundreds of products and increases the scalability of products.

**Index Terms**—Detection, product recognition, retrieval, unmanned retail.

## I. INTRODUCTION

VARIOUS methods based on deep learning have been applied to industry recently, such as face payment, intelligent security, and so on. While intelligent unmanned vending machines (UVMS) and unmanned supermarkets have brought increasing profits to many companies, the emerging concept of “unmanned retail” has attracted increasing attention.

Among them, unmanned retail based on the intelligent UVMS scene has been gradually accepted by the public, and occupies an increasing market share. However, traditional UVMS rely on automation technology and mechanical sensors to deliver drinks when a customer pressing buttons. It is an inconvenient nontouched purchase experience, and it has high operating costs. Unlike them, the core technology of unmanned retail based on intelligent UVMS scene is to recognize the products in the image collected by the camera [1], [2] mounted on top of the container. There are the following three main advantages why intelligent UVMS are popular among the public.

- 1) In the process of customer purchase, it is combined with deep learning technology to ensure high accuracy, and at the same time has the superiority of interaction and selectivity for products.
- 2) At the end of each purchase, the supervisor can monitor the quantity of products and customize the unique replenishment scheme to reduce a lot of operating costs.
- 3) Merchants can record and analyze customer purchase data to boost potential commercial applications and gain additional revenue.

Therefore, the research related to the intelligent UVMS is very valuable, and it is necessary to construct a proper recognition approach for intelligent UVMS. In this work, aiming at the product recognition based on intelligent UVMS scene, we proposed a product recognition approach, called **Split-Check**. Our method is applied for the recognition of large categories [stock keeping units (SKUs)] of products in intelligent UVMS. The overview is shown in Fig. 1.

Generally speaking, good performance for recognition relies on high similarities within classes and large differences between

Manuscript received 14 December 2022; revised 3 April 2023; accepted 14 August 2023. Date of publication 12 September 2023; date of current version 23 February 2024. This work was supported in part by the NSFC under Grant 62272380 and Grant 62103317, in part by the Science and Technology Program of Xi’an, China under Grant 21RGZN0017, and in part by the SHAANXI KEY R&D under Grant 2022QFY01-17 and Grant 2022FP-40. Paper no. TII-22-5089. (Corresponding author: Xueming Qian.)

Chengxu Liu and Yao Xue are with the School of Information and Communication Engineering, Xi’an Jiaotong University, Xi’an 710049, China, and also with the Shaanxi Yulan Jiuzhou Intelligent Optoelectronic Technology Company, Ltd., Xi’an 710000, China (e-mail: liuchx97@gmail.com; xueyao@xjtu.edu.cn).

Zongyang Da is with the School of Information and Communication Engineering, Xi’an Jiaotong University, Xi’an 710049, China (e-mail: dzy1134483011@stu.xjtu.edu.cn).

Yuanzhi Liang is with the School of Software Engineering, Xi’an Jiaotong University, Xi’an 710049, China (e-mail: liangyzh13@stu.xjtu.edu.cn).

Guoshuai Zhao is with the School of Software Engineering, Xi’an Jiaotong University, Xi’an 710049, China, and also with the Shaanxi Yulan Jiuzhou Intelligent Optoelectronic Technology Company, Ltd., Xi’an 710000, China (e-mail: guoshuai.zhao@xjtu.edu.cn).

Xueming Qian is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security, School of Information and Communication Engineering, and SMILES LAB, Xi’an Jiaotong University, Xi’an 710049, China, and also with the Shaanxi Yulan Jiuzhou Intelligent Optoelectronic Technology Company, Ltd., Xi’an 710000, China (e-mail: qianxm@mail.xjtu.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TII.2023.3308771>.

Digital Object Identifier 10.1109/TII.2023.3308771

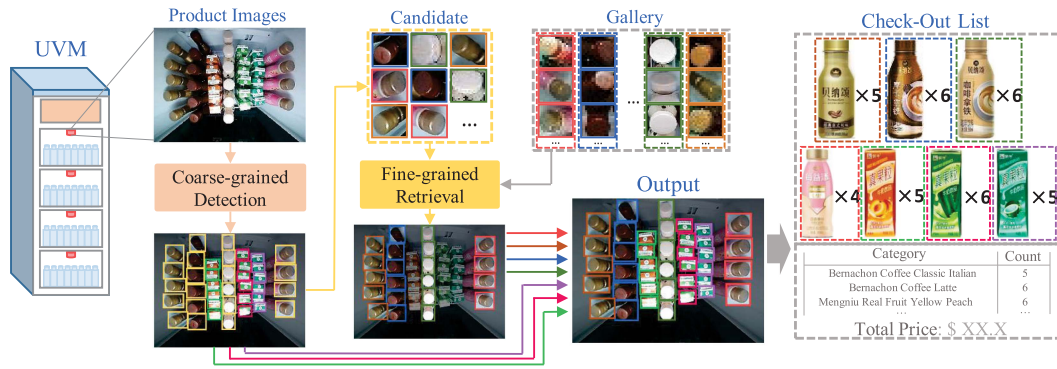


Fig. 1. Overview of our approach. It is the combination of two important components. The preliminary coarse-grained detection splits items, and classifies them into coarse-grained categories as a candidate. The fine-grained retrieval further checks the candidate items and boosts the performance.

classes. However, in the task of product recognition in UVMs, we find two main challenges: the **high intraclass variance** due to the angle and position, and the **low interclass variance** due to the appearance, especially when there is a variety in SKU. Based on our actual experience, these challenges lead to the following three main problems.

- 1) *Products with a similar appearance need to be treated separately due to the low inter-class variance:* For products that are very similar in appearance, often have most of the same coarse-grained features and subtle fine-grained features, this suggests that it is difficult to learn the differences between them mindlessly. For example, the top contour of mineral water is white and round, with only a distinctive fine-grained feature on the subtle logos. Compared with those products in a box with a completely different structure, it is intuitively believed that their differences should not be treated equally, otherwise it will affect the stability of learning, thus leading to poor performance.
- 2) *It is difficult to explore the subtle fine-grained differences between products, due to the large intraclass variance, especially for large-scale SKUs:* For example, as shown in Fig. 2(a), vitamin water, C100 lemonade, and Tea pi have the same white bottle cap and similar drink color. Three different flavors of Bernachon coffee have exactly the same top structure, as shown in Fig. 2(b). Focusing on the subtle differences between products is the key to differentiating them.
- 3) As products become more abundant in practice, what is worth considering is how to expand the category with minimal cost, especially for large-scale SKUs, and still maintain performance.

In the field of unmanned retail, existing works focus their efforts on smart unstaffed retail shop [3], [4], abnormal detection [5], and automatic checkout system [6]. There are additional benchmark are all based on completely different scenarios [3], [7]. In addition, product recognition based on intelligent UVMs has also flourished. Zhang et al. [1] combined the customer's purchase behavior from the beginning to the end to recognize product. Li et al. [2] proposed DrtNet, combined with deformable convolution, focal loss, and other technologies to

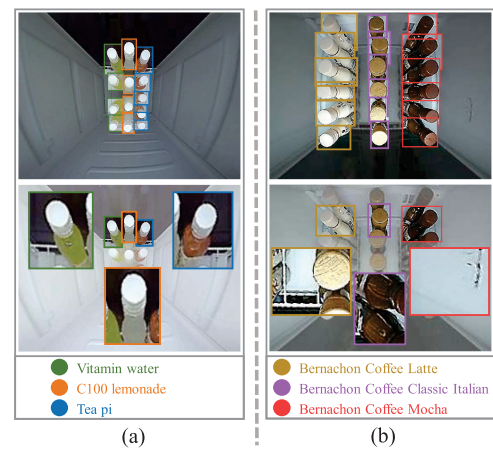


Fig. 2. Illustration of the similar products with the subtle fine-grained differences. The top of indicates the position and category of products and the bottom is the enlarged items.

assist more accurate product recognition. Almost all the existing intelligent UVMs contain only ten distinct categories, which is very limited and the categories are few and sparsely distributed. More importantly, it is difficult for them to deal with a various-category dataset. Therefore, in the existing work, there are the following issues worthy of attention.

- 1) The scale of SKUs directly determines the feasibility of the work in practice, and when the product variety is poor will greatly reduce the customer's purchasing experience.
- 2) Due to low interclass variance, the multigranularity features of the product are very important. Generally, the high similarity between different classes leads to poor performance; meanwhile fine-grained classification is also a necessity. Existing methods do not address this problem at all.
- 3) The use of detection or classification makes output units fixed and not extensible, and many additional data collection and retraining are often needed to supplement a product.

To address the above issues, we propose a product recognition approach **Split-Check** for boosting large-scale categories recognition based on intelligent UVMs. We design a brand-new

coarse-to-fine framework for recognition, which integrates the two components of preliminary coarse-grained detection and fine-grained retrieval: 1) The preliminary coarse-grained detection splits the items that contain the different coarse-grained feature or distinct color differences, and the rotation angle is robust. For example, products with different top contours and different colors. In addition, according to the characteristics of the product, the detection is improved to make it perform well on small items. 2) The fine-grained retrieval checks the candidate items that have been split, and its candidate contains subtle fine-grained features. For example, the products have the same bottle and similar drink color or only the subtle logo on the bottle is different. Moreover, the expression ability of fine-grained features is enhanced by combining pooling and tripletloss [8] in this section. The fine-grained retrieval is very extensible for new products and does not require retraining. To demonstrate the effectiveness of our approach, we experimented on the GOODS-85 datasets, which we collected a dataset of 85 products based on the actual UVMs scenario. Our main contributions are as follows.

- 1) We propose an approach **Split-Check** for boosting large-scale categories product recognition based on intelligent UVMs. It combines the two components of preliminary coarse-grained detection and fine-grained retrieval. Effectiveness of the approach proposed by us is proved through the comparative experiments on GOODS-85 dataset.
- 2) The preliminary coarse-grained splits distinguish the items that contain the different coarse-grained feature or distinct color differences. It allows for better positioning and is improved to make it perform well on small objects.
- 3) Fine-grained retrieval checks the candidate items that have been split, and it contains subtle fine-grained features. The fine-grained representations of the product are enhanced by combining pooling and tripletloss [8]. More importantly, the retrieval is very extensible for new products and does not require retraining.

The rest of this article is organized as follows. Related work is reviewed in Section II. The proposed method is elaborated in Section III. Experimental evaluation and analysis are presented in Section IV. Finally, Section V concludes this article.

## II. RELATED WORK

In this section, we mainly introduce the related work on product recognition based on intelligent UVMs. Then, we give a brief overview of object detection and retrieval.

### A. Product Recognition Based on Intelligent UVMs

The core technology of unmanned retail based on intelligent UVMs scene is to recognize the products collected by the camera [1], [2] mounted on top of the container. We present the datasets of relevant products and the latest methods of product recognition based on intelligent UVMs, respectively.

There are increasing datasets about products, but not all of them apply to UVMs. Some of them are used for classification [3], others for detection [7]. They are completely different from the application scenarios of our work and are not helpful for product recognition based on UVMs. Zhang et al. [1] considered

the real-world scenarios of UVMs, and constructed a dataset for beverage detection. The datasets comprise ten categories of beverages in the market of China, with an average of 4.56 instances per image. Pujol et al. [9] collected more than 30 000 images of unmanned retail containers including ten kinds of beverages and 155 153 instances. These two are different from the dataset we used, we collected a dataset covering a total of 85 SKUs. The products are densely laid out, with an average of 22.97 instances per image.

Much work has been done to contribute to the task of product recognition based on intelligent UVMs. Li et al. [2] proposed DrtNet, combined with deformable convolution, focal loss, and other technologies to assist more accurate product recognition. Almost all the existing intelligent UVMs contain only ten distinct beverages, which is very limited considering the business application. Furthermore, some recognition work makes output units fixed and not extensible, which increases the difficulty of adding new category. Different from the existing methods, our work focuses on proposing a cascaded product recognition method based on intelligent UVM with larger SKUs, and improves the performance effectively, increasing the possibility of expansion for additional products. Furthermore, our work focuses on large-scaled dataset, which has more practical and commercial value.

### B. Object Detection

The key technology of coarse-grained detection is object detection. With the development of deep learning recently, a series of detection methods emerge in an endless stream and are widely used in the industrial field. Whereas one-stage detectors have emerged as a popular paradigm, such as SSD [10], YOLO [11], and RetinaNet [12], many top-performing frameworks still adopt the proven two-stage pipeline, such as Faster R-CNN [13] and FPN [14]. Then, with the advent of CornerNet [15], object detection entered the era based on anchor free, and more advanced methods CenterNet [16], CentripetalNet [17], and FCOS [18] achieve better performance.

In addition to the general object domain, it is also very important in the industrial field, including medical cancer cell detection, pedestrian detection, and so on. In the field of product recognition of intelligent UVMs, Zhang et al. [1] applied object detection and classification method for product recognition. Li et al. [2] proposed DiffNet and DrtNet to assist in identifying products that changed before and after purchase for more accurate product recognition. These approaches perform well in certain scenarios. However, when expanding the SKU category range these approaches are ineffective. It is because general object detection methods neglect the high intraclass variance, which makes the general detector difficult to learn the feature difference between different categories. For our proposed approach, we use preliminary coarse-grained detection is used to better split and distinguish coarse-grained categories, which reduces stress on the detector.

### C. Image Retrieval

The key technology of fine-grained retrieval is to explore the representative fine-grained features of products with similar



appearance and output the retrieval results by feature matching. Zheng et al. [19] provided a comprehensive survey of instance retrieval over the last decade. Deep convolutional network can better extract the features and capture the focus areas in the image. VGGNet [20], GoogleNet [21], ResNet [22], and DenseNet [23] are widely used. For the unsupervised image retrieval, Kalantidis et al. [24] proposed a way of creating image representations via cross-dimensional weighting and aggregation of network outputs.

Image retrieval is combined with the requirements of various fields. Wei et al. [25] conducted fine-grained feature image retrieval through deep learning. For the medical image retrieval system, Qayyum et al. [26] proposed a framework for a content-based medical image retrieval system. In product recognition task, we use a retrieval module to deal with the low inter-class variance problem. For Split-Check, fine-grained retrieval enhances the expressive ability of features. More importantly, retrieval is very extensible for new products and require no retraining, which is more convenient for practical use.

### III. METHOD

#### A. Overview

In this section, we describe the framework of our Split-Check. As shown in Fig. 1, Split-Check consists of two major parts: preliminary coarse-grained detection and fine-grained retrieval. 1) Preliminary coarse-grained detection is used to split products with obvious structural information or distinct color differences. In addition, in view of the characteristics of product distribution, we improved the detection to achieve product performance in the positioning and classification of small objects. 2) Fine-grained retrieval is used to extract the fine-grained feature of products. At the same time, we adopted the combinatorial pooling features and tripletloss [8] to enhance the feature expression ability of fine-grained features.

More details are shown in Fig. 1. The recognition steps of our Split-Check are as follows.

- 1) The coarse-grained detection outputs two parts: one part is the candidate for further retrieval, and the other is the recognition result. The recognition result includes the category and bounding box information. Among the recognition result, for the products with similar appearances, we consolidated them into a single category, which are represented by the candidates. They will be further distinguished through the fine-grained retrieval.
- 2) Fine-grained retrieval takes candidates as inputs and then extracts the feature embedding of each item in the candidate. Then, retrieval network compares each item with the existing gallery, which includes feature embeddings of standard instances of all kinds of products. The network selects the final class by calculating Euclidean distance minimization.
- 3) The final recognition results are determined by the recognition result from the preliminary coarse-grained detection and the retrieval result from the fine-grained retrieval.

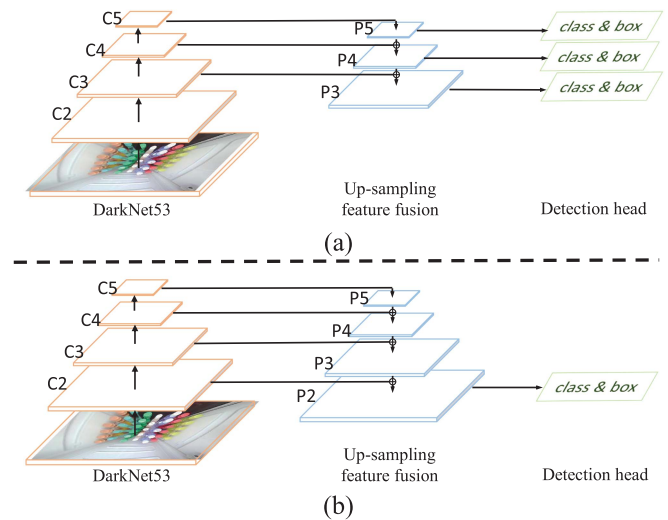


Fig. 3. Illustration of the backbone. (a) Structure for the original YOLOv3. (b) Structure for Split-Check.

#### B. Coarse-Grained Detection

In the sight of cameras in real UVM scenes, products only occupy a small area, and all of them are tiled in the whole image to a certain extent. YOLOv3 [27] has a good performance for small items, so we chose it as the basic network and revised the framework for small object detection. At the same time, to ensure the effect of classification and positioning, we used a separated prediction head structure (SPHS). The detailed methods are as follows.

1) *Optimization for Small Object Detection:* The backbone of Split-Check is improved on the basis of DarkNet53 in YOLOv3 [27]. As shown in part (a) of Fig. 3, convolutional layers of DarkNet53 are used as the basic feature extractor to get the semantic information. The image has been downsampled for five times in total, and the size of the output feature map is reduced by 32 times. Original DarkNet53 has a good performance on feature extraction, but not suitable for product recognition in UVMs because of small object. Therefore, we designed and optimized the feature pyramid network. As shown in part (b) of Fig. 3, the size of the output feature map is increased by three up-sampling layers; after each up-sampling layer, the feature map is fused with the shallow feature map from the base network. Up-sampling and feature fusion integrates the deep semantic and shallow texture information, which greatly improves the feature extraction capability of Split-Check. Furthermore, our proposed dataset based on the actual UVMs has a small area of instance and low bounding box overlap, and our optimization also increases the size of the feature map for small object detection. More analyses can be found in the Supplementary Material.

2) *Separated Prediction Head:* As shown in Fig. 4, for YOLOv3, the prediction head is used to predict the category and position, and the parameters of this two parts are shared. However, YOLOv3 neglects that classification for the category relies on semantic features while regression for the position

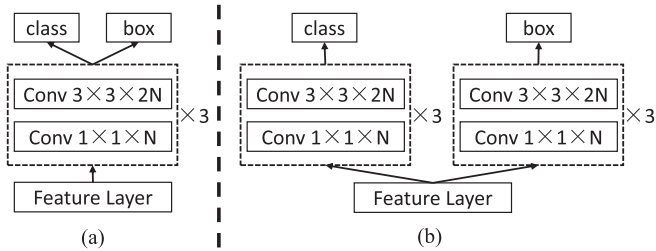


Fig. 4. Illustration of the prediction head structure. (a) Prediction head structure for the original YOLOv3. (b) SPHS we improved.  $N$  is the number of channels in the feature map.

more relies on textural features. It means shared parameters will trigger unstable learning and slow convergence speed. Besides, due to large intraclass variance, the instances in small object detection have larger diversity in the aspect of position, thus parameters-shared prediction head will influence the performance of small object detection. Therefore, we used SPHS and use different parameters to adapt different tasks, respectively, which enhances the stability of the learning process. More analyses can be found in the Supplementary Material.

### C. Fine-Grained Retrieval

Some products tend to have similar appearances, especially for large-scale SKUs. Fine-grained retrieval is used to extract the fine-grained features of products, which have similar structures or appearances. It distinguishes the candidate items detected by the coarse-grained detection, which only has subtle interclass differences. To reconcile the extensibility of new products and the ability to capture fine-grained features, we used CNN to extract the feature embeddings. Besides, to enhance the fine-grained feature expression, we used combinatorial pooling features and tripletloss during training. The detailed methods are as follows.

1) *Backbone*: It is crucial to extract product features with strong expression ability in retrieval. Compared with traditional methods, deep convolutional network can better extract the features and capture key areas in the image. VGGNet [20], GoogleNet [21], ResNet [22], and DenseNet [23] are widely used. Among them, ResNet performs well in terms of speed and capacity. ResNet with different depths has different inference speeds and capacities, and the deeper ResNet often has a better ability on feature extraction.

In detail, we cropped out the candidates detected in the coarse-grained detection. Input candidates share the same scale of  $H \times W$  (i.e.,  $32 \times 32$ ). Then, we feed the candidate into the retrieval component consisting of a deep convolutional network to extract high-dimensional fine-grained product features. Where the scale of the fine-grained features for each candidate is  $d \times h \times w$  (i.e.,  $1024 \times 1 \times 1$ ), and the ratio between  $H$  and  $h$  are the same with the one between  $W$  and  $w$ .  $d$  is the number of dimensions of the output feature maps. For image retrieval, we generated the feature embeddings of some standard instances and grouped the embeddings as the gallery. The category of each candidate will be determined by calculating Euclidean distances

between the candidate and each embedding in the gallery. We use retrieval instead of classification to make this extensible for new products: we only need to output feature embeddings of new products. It is beneficial to commercial updating.

2) *Combinatorial Pooling*: Different pooling methods focus on different feature information. For example, global average pooling (GAP) focuses on all regions, whereas global maximum pooling (GMP) focuses on the corresponding strongest region. They have advantages in different situations. We adopted combinatorial pooling methods to combine GAP and GMP. In detail, we use GAP and GMP to obtain two different feature vectors with length of  $d$  from the output feature embedding, and then concatenate them together to form a vector with a length of  $2 \times d$ , which well represents the fine-grained structure or appearance differences. For an input image, the input  $v_{\text{Avg}}(i) \in R^d$  and  $v_{\text{Max}}(i) \in R^d$  represent the GAP feature and GMP feature, respectively. The vector  $e(i) \in R^{2 \times d}$  after combinatorial pooling can be represented as

$$e(i) = [v_{\text{Avg},1}, \dots, v_{\text{Avg},d}, v_{\text{Max},1}, \dots, v_{\text{Max},d}]. \quad (1)$$

We assume  $G$  as the gallery and  $g(i) \in G$  is the embeddings in  $G$ . Our goal is to find the  $g(i)$  that corresponds to the smallest Euclidean distance, which can be expressed as

$$\arg \min_{g(i) \in G} \sqrt{\sum_{i=1}^{2 \times d} [e(i) - g(i)]^2} \quad (2)$$

where the category corresponding to  $g(i)$  is our final retrieval result. Finally, we use the index category of minimum Euclidean distance to measure the final category. More analyses can be found in the Supplementary Material.

3) *Tripletloss*: Retrieval is aimed at distinguishing candidates subtle fine-grained features, so we added tripletloss [8] to enhance the extraction ability of fine-grained interclass features for network. Different from the original tripletloss, we do not use normalization for the input data in order to gap the difference between classes.

In detail, we assume that  $\mathcal{T}$  is the set of all possible triplets in the training set and it has  $N$  triplets. For an input image, the feature embedding is represented as  $e(i) \in R^{2 \times d}$ .  $e^p(i) \in R^{2 \times d}$  and  $e^n(i) \in R^{2 \times d}$  are the feature embedding of an instance from the same category and from any other categories, respectively. We want to ensure that the  $e(i)$  is closer to all other  $e^p(i)$  than any  $e^n(i)$ . Therefore, the loss function can be represented as

$$L = \sum_i^N \{d[e(i), e^p(i)] - d[e(i), e^n(i)] + \alpha\} \quad (3)$$

where  $(e(i), e^p(i), e^n(i)) \in \mathcal{T}$ .  $d(\cdot)$  represents the Euclidean distance.  $\alpha$  represents the margin parameter. The larger the value of  $\alpha$ , the greater the difference between category will be, but too large  $\alpha$  will also affect the stability. More analyses on retrieval can be found in the Supplementary Material.

## IV. EXPERIMENTS

In this section, we first briefly introduce our dataset and experimental settings. Furthermore, in the results and analysis section, we compare our Split-Check and other methods.

**TABLE I**  
SAMPLE DISTRIBUTION OF IMAGES AND OBJECTS IN DATASET (OBJECT DENOTES THE NUMBER OF INSTANCES)

Type	Distribution		Category	Objects per image
	trainval	test		
Image	733	314	85	22.97
Object	17295	6756		

### A. Dataset

In this section, to demonstrate the superiority of our method, we reconstruct a dataset that includes large-scale kinds of SKUs, namely GOODS-85 [28]. The distribution of images and instances in the dataset is given in Table I. For the detection part, we grouped all bottled waters into one candidate category, with 28 categories for coarse-grained detection. For the retrieval part, we specifically divided the bottled water categories detected by the coarse-grained detection part, with 68 categories for fine-grained retrieval, which are difficult to distinguish by the detection network. We covered all areas in the UVMs with different angles of SKU targets to build the gallery of retrieval dataset. Each category has 50 standard samples distributed in different locations in the UVMs.

Moreover, we stretch the fisheye image from the center position to the surrounding position by spherical isometric projection correction model, so as to avoid the serious overlapping of bounding boxes caused by the dense spatial position. We have labeled the top of the item area, and the bounding box tries to cover the area visible to the items as much as possible. A total of 24 051 instances were labeled with category labels and bounding boxes. Meanwhile, these instances are labelled with the retrieved category for the retrieval network training. There are about 100 to 1000 instances per category for learning of fine-grained retrieval features.

### B. Experimental Settings

1) *Compared Methods*: To demonstrate the superiority of our Split-Check for product recognition, we compare the method with some classical detection algorithms and product recognition methods. They can be categorized into the following paradigms.

- 1) One-stage detection algorithms, i.e., SSD [10], YOLOv3 [27], RetinaNet [12], YOLOv5 [29], YOLOX [30], and YOLOv7 [31].
- 2) Two-stage detection algorithms, i.e., Faster R-CNN [13] and Dynamic R-CNN [32].
- 3) Anchor-free detection algorithms, i.e., CornerNet [15], CenterNet [16], CentripetalNet [17], FCOS [18], and SABL [33].
- 4) Label assignment algorithm, i.e., ATSS [34].
- 5) Product recognition algorithm, i.e., PRUVM [28].

2) *Performance Evaluation*: For performance evaluation, a widely used metric mean average precision (mAP) was calculated for products in our experiment, formulated as

$$\text{mAP} = \frac{1}{m} \sum_{i=1}^m \text{AP}_i \quad (4)$$

where  $m$  represents the number of categories of products. The idea of average precision (AP) can be conceptually regarded as calculating the area under the precision and recall curve of each product. The calculation formula of AP is

$$\text{AP} = \frac{1}{11} \sum_{r \in \{0,0.1,\dots,1\}} p_{\text{interp}}(r) \quad (5)$$

where  $p_{\text{interp}}(r) = \max_{\tilde{r}: \tilde{r} \geq r} p(\tilde{r})$  represent the maximum precision when recall equals to  $r$ . Among them, the precision and recall rate can be expressed as follows:

$$p = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (6)$$

$$r = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (7)$$

where TP, FP, and FN are the true positive, false positive, and false negative sample quantity of products, respectively. Unless otherwise stated, we used the result of setting the intersection of union threshold value as 0.5 when calculating the

$$m\text{AP}.$$

3) *Implementation Details*: The proposed method is implemented by PyTorch. The base detection and retrieval adopt the premodel of DarkNet53 and ResNet50 in ImageNet, respectively. Also, we removed the last two layers of downsampling in the retrieval to match the feature size. In Section III-C2, the feature vectors with length of  $d$  is 512. In Section III-C3, the margin parameter  $\alpha$  is 15.0. Unless otherwise specified, we will use images of size 416 in detection, and images of size 56 in retrieval. We trained the detection and retrieval in the GOODS-85 dataset, respectively. For the detection network training, we used the Adam optimization algorithm to train the network, and the set initial learning rate is 0.001, and training stops after 60 epochs. For the retrieval network training, we used the stochastic gradient descent (SGD) optimization algorithm to train the network, and set the weight decay to be 0.0001 and momentum is set to be 0.9. The initial learning rate is 0.001 for the first 24 epochs, which decays by a factor of 10 for the next ten and six epochs, and training stops after 40 epochs. All the experiments are conducted on a workstation with 8 GTX-1080Ti GPUs.

### C. Results and Analysis

1) *Objective Comparison*: In Table II, we compare the classic detection methods of one stage, two stage, and anchor free in recent years on the GOODS-85 dataset. Compared with the classic methods for general object detection, such as SSD [10] and Faster R-CNN [13], the two-stage method has a better effect due to its strong adaptability and dense anchor to small-scale items. Furthermore, the high similarity between categories also limits the effect of the one-stage methods. The method based on anchor free can also achieve better performance, such as CentripetalNet [17] and SABL [33], reaching 93.2% and 93.0% of the mAP. However, when an image contains many similar products, and they are placed densely, it will bring a certain difficulty to the selection of the center point and the matching of



TABLE II

QUANTITATIVE RESULTS IN TERMS OF MAP AND NUMBER OF PARAMETERS IN COMPARISON OF STATE-OF-THE-ART OBJECT DETECTION METHODS

Method	Backbone	Size	mAP	#Params
SSD [9]	VGG16	512×512	91.2%	36.04
YOLOv3 [26]	Darknet53-FPN	608×608	89.1%	63.00
RetinaNet [11]	ResNet50-FPN	600×600	84.9%	37.74
Faster R-CNN [12]	ResNet50-FPN	600×600	91.9%	41.53
CornerNet [14]	Hourglass-104	511×511	92.1%	201.04
CenterNet [15]	Hourglass-104	511×511	90.4%	190.70
CentripetalNet [16]	Hourglass-104	511×511	93.2%	205.76
FCOS [17]	ResNet50-FPN	600×600	90.4%	32.02
PRUVM [27]	ResNet50-FPN	600×600	93.7%	47.95
SABL [32]	ResNet50-FPN	512×512	93.0%	41.99
ATSS [33]	ResNet50-FPN	512×512	93.2%	32.07
Dynamic R-CNN [31]	ResNet50-FPN	512×512	90.7%	41.53
YOLOv5-L [28]	Darknet53-FPN	608×608	91.8%	46.50
YOLOX-L [29]	Darknet53-FPN	608×608	91.9%	54.20
YOLOv7-X [30]	Darknet53-FPN	608×608	94.0%	71.30
<b>Split-Check(Ours)</b>	Darknet53-FPN+R50	416×416 608×608	<b>95.8%</b> <b>96.1%</b>	62.93

#Params denotes the number of parameters (M).  
The bold values mean the best performance.

TABLE III

QUANTITATIVE RESULTS IN COMPARISON OF STATE-OF-THE-ART OBJECT DETECTION MODELS ON SMARTUVM [1] DATASET

Method	Backbone	Size	mAP
Faster R-CNN [12]	VGG16	600×600	90.8%
SSD [9]	VGG16	512×512	91.2%
YOLOv3 [26]	Darknet-53	608×608	91.0%
RetinaNet [11]	ResNet50-FPN	600×600	90.9%
CornerNet [14]	Hourglass-104	511×511	91.3%
CenterNet [15]	Hourglass-104	511×511	91.4%
FCOS [17]	ResNet50-FPN	600×600	91.6%
PRUVM [27]	VGG16	600×600	92.1%
<b>Split-Check(Ours)</b>	DarkNet-53-FPN+ResNet50	416×416 608×608	<b>92.3%</b> <b>92.6%</b>

The bold values mean the best performance.

corner points. More importantly, in practice, the methods based on anchor free are difficult to ensure adequate recall capacity, and it is not extensible for new products and require retraining.

According to the characteristics of the products and the actual demand, our method can better mine the fine-grained features of products with similar appearance and achieves the best performance on the GOODS-85. Even for lower input sizes 416, Split-Check improves by 1.8% compared with the best one-stage method YOLOv7 [31], improves by 2.6% compared with the best anchor-free method CentripetalNet [17]. This high performance further demonstrates the generalization ability and superiority of the Split-Check with different input sizes.

On the aspect of computational complexity, Split-Check also makes some difference. First of all, compared with original YOLOv3 [27], the parameters of Split-Check are smaller than the original network parameters, even when adding the retrieval network. It is because after up-sampling feature fusion Split-Check only retains one detection head structure, and thus the number of parameters has been largely reduced. Besides, compared with anchor-free and two-stage networks, Split-Check can achieve a better tradeoff between accuracy and computational resource.

In Table III, we also compare our method with other typical detection methods. However, SmartUVM [1] dataset is different

TABLE IV

QUANTITATIVE RESULTS IN TERMS OF RECALL@1 AND RECALL@5 IN COMPARISON OF STATE-OF-THE-ART RETRIEVAL METHODS

Method	Recall@1	Recall@5
Triplet Loss [34]	97.07%	98.72%
Triplet LogExp Loss [34]	97.10%	99.01%
Angular Loss [35]	97.01%	99.15%
AP Loss [36]	95.72%	98.22%
Tie-aware AP Loss [36]	95.74%	98.45%
Ours	97.89%	99.45%

from ours, which only contains ten beverages with obvious differences. We can see that on SmartUVM, compared with Faster R-CNN [13], the one-stage-based and anchor free-based methods have better performance. It can be found that although the dataset only contains ten categories and the advantages of our method cannot be exploited, it can still bring about an improvement of about 0.4%. Experimental results show that our method has a better performance in a smaller image size, and can be extended for new products without retraining, which is in line with practical applications.

2) *Comparison on Retrieval*: The second part of the results and analysis aims to prove the effectiveness of retrieval module in our method. The comparison methods include Triplet Loss [35], Triplet LogExp Loss [35], Angular Loss [36], AP Loss [37], and tie-aware AP loss [37]. All the experiments are conducted on ResNet50 [22] and we share the same hyperparameters described in Section IV-B3. For retrieval performance evaluation, we use the widely-used Recall@1 and Recall@5, which mean the accuracy of the top 1 and top 5 results from the retrieval list sorted by score.

The experimental results are given in Table IV, the second and third rows are the recall results of training with each method. Compared with Triplet Loss and Triplet LogExp Loss, our method has an advantage of 0.8% in Recall@1. Compared with other methods our retrieval method is also leading. The reason for that can be indicated in two aspects: 1) we use triplet loss to enlarge the feature gap between similar categories, which is useful when low interclass variance; 2) combinatorial pooling we used further enhances the feature representation ability.

3) *Ablation Experiments on Each Component*: The third part of the results and analysis aims to prove the effectiveness of each component in our method, different components are used for detection and retrieval in the ablation experiment. In this section, we share the same hyperparameters and training strategy to make sure no influence from other factors. Specific experimental details are described in Section IV-B3.

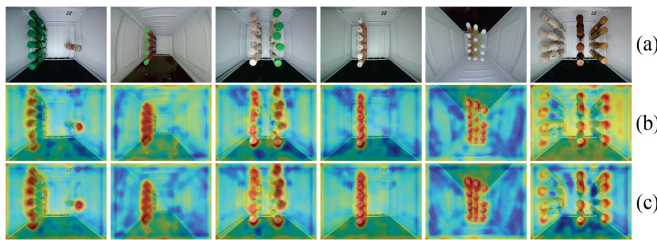
The experimental results are given in Table V, the front four rows are the results of adding each component separately, and we use the GAP to obtain the feature embeddings. Compared with SPHS and tripletloss, optimization of small object (OSO) has better performance. The front seven rows indicates that each component contributes to the recognition. More importantly, OSO make the mAP improved more than 3.5%, due to the small size of the product, and SPHS and Tripletloss make the mAP improved more than 0.1% and 0.7%, respectively. The results of the last three rows indicate that GAP can be used to extract

**TABLE V**  
RESULTS OF ABLATIONS ON GOODS-85 DATASET

OSO	SPHS	Tripletloss	CPF		mAP
			GAP	GMP	
			✓		90.6%
✓			✓		94.3%
	✓		✓		90.9%
		✓	✓		91.5%
✓	✓		✓		94.3%
	✓	✓	✓		91.6%
✓		✓	✓		95.2%
✓	✓	✓	✓		95.6%
✓	✓	✓		✓	95.1%
✓	✓	✓	✓	✓	<b>95.8%</b>

OSO: Optimization of small object. SPHS: Separated prediction head structure. CPF: Combinatorial pooling features. GAP: Global average pooling. GMP: Global maximum pooling.

The bold value means the best performance.



**Fig. 5.** Schematic of the heatmap visualization in detection. (a) Original images. (b) Heatmap of the original YOLOv3. (c) Heatmap of the coarse-grained detection part in the Split-Check.

fine-grained features better, while GMP is useful, but of minimal effect.

The results demonstrate that adding them can effectively improve the mAP of product recognition. In particular, although our method is not comparable to the best methods when these components are not added, comprehensive considerations of them can bring a degree of improvement in this task.

4) **Visualization:** It is difficult to classify and regression simultaneously using a single feature extraction network due to the large categories of products and low interclass variance. To explain the mechanism of our method intuitively, we visualized the feature map of the detection part, as shown in Fig. 5. Among them, (a) shows the original images. (b) represents the heatmap of the original YOLOv3. In this case, the detection network extracts the features for classification and regression at the same time, learning not only the fine-grained features of each product but also their location. Instead, (c) represents the heatmap of the coarse-grained detection part in the Split-Check, and we consolidate a group of products into candidate categories, which allows the network to learn more general features, and reduces the burden on the detection.

As shown in Fig. 5, we artificially selected some different products that belong to the same candidate category and visualized them. In Fig. 5(c), they are classified into the same category in the process of training, whereas in Fig. 5(b), each product is learned separately. For each image, Fig. 5(c) pays more attention to the product location than Fig. 5(b), and has a stronger response to each item. It is worth noting that there are two reasons for this result. 1) In the case of the large category of products and low interclass variance, it is more reasonable for detection to

learn more generalized features that indicate in Fig. 5(c) than learn the fine-grained features of each product that indicate in Fig. 5(b). 2) The proposed Split-Check is used to put the process of fine-grained feature extraction into the retrieval, which greatly reduces the burden of detection and improves the recall.

## V. CONCLUSION

In this article, we propose a coarse-to-fine approach for large-scale categories of product recognition. It integrates preliminary coarse-grained detection and fine-grained retrieval. Preliminary detection outputs classes and positions for instances, and it also generates candidates for further recognition. The detection allows for better positioning and is improved to make it perform well on small objects. Fine-grained retrieval takes candidates as input and extracts the fine-grained features. Retrieval allows for better features so that it can perform the classification well and it is extensible for new products. Effectiveness of the approach proposed by us is proved through the experiments on GOODS-85 dataset, which we collected based on the actual UVMs scenario. Our method significantly improves the recognition performance of hundreds of products and increases the scalability of products.

## REFERENCES

- [1] H. Zhang, D. Li, Y. Ji, H. Zhou, W. Wu, and K. Liu, "Towards new retail: A benchmark dataset for smart unmanned vending machines," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7722–7731, Dec. 2020.
- [2] D. Li, H. Zhou, G. Li, B. Yang, F. Gao, and H. Zhang, "DrtNet: An improved RetinaNet for detecting beverages in unmanned vending machines," in *Proc. IEEE Int. Symp. Product Compliance Eng.-Asia-CN*, 2020, pp. 1–6.
- [3] P. Follmann et al., "MVTec D2S: Densely Segmented Supermarket Dataset," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 569–585.
- [4] L. Liu, B. Zhou, Z. Zou, S.-C. Yeh, and L. Zheng, "A smart unstaffed retail shop based on artificial intelligence and IoT," in *Proc. IEEE 23rd Int. Workshop Comput. Aided Model. Des. Commun. Links Netw.*, 2018, pp. 1–4.
- [5] Z. Da et al., "Anomaly detection framework for unmanned vending machines," *Knowl.-Based Syst.*, vol. 262, 2023, Art. no. 110251.
- [6] L. Zhang, D. Du, C. Li, Y. Wu, and T. Luo, "Iterative knowledge distillation for automatic check-out," *IEEE Trans. Multimedia*, vol. 23, pp. 4158–4170, 2020.
- [7] E. Goldman, R. Herzig, A. Eisenschat, J. Goldberger, and T. Hassner, "Precise detection in densely packed scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5227–5236.
- [8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.
- [9] V. Casamayor-Pujol, B. Gastón, S. López-Soriano, A. A. Alajami, and R. Pous, "A simple solution to locate groups of items in large retail stores using an RFID robot," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, pp. 767–775, Feb. 2022.
- [10] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
- [11] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.
- [12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
- [13] S. Ren et al., "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
- [15] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.



- [16] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6569–6578.
- [17] Z. Dong, G. Li, Y. Liao, F. Wang, P. Ren, and C. Qian, "CentripetalNet: Pursuing high-quality keypoint pairs for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10519–10528.
- [18] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9627–9636.
- [19] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1224–1244, May 2018.
- [20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.
- [21] C. Szegedy et al., "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [23] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.
- [24] Y. Kalantidis et al., "Cross-dimensional weighting for aggregated deep convolutional features," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 685–701.
- [25] X.-S. Wei, J.-H. Luo, J. Wu, and Z.-H. Zhou, "Selective convolutional descriptor aggregation for fine-grained image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2868–2881, Jun. 2017.
- [26] A. Qayyum et al., "Medical image retrieval using deep convolutional neural network," *Neurocomputing*, vol. 266, pp. 8–20, 2017.
- [27] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [28] C. Liu et al., "Product recognition for unmanned vending machines," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 29, 2022, doi: [10.1109/TNNLS.2022.3184075](https://doi.org/10.1109/TNNLS.2022.3184075).
- [29] J. Glenn et al., "ultralytics/yolov5: v6.1 – TensorRT, TensorFlow Edge TPU and OpenVINO export and inference," Feb. 2022.
- [30] Z. Ge et al., "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
- [31] C.-Y. Wang et al., "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7464–7475.
- [32] H. Zhang et al., "Dynamic R-CNN: Towards high quality object detection via dynamic training," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 260–275.
- [33] J. Wang et al., "Side-aware boundary localization for more precise object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 403–419.
- [34] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9759–9768.
- [35] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. Int. Workshop Similarity-Based Pattern Recognit.*, 2015, pp. 84–92.
- [36] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2593–2601.
- [37] J. Revaud, J. Almazan, R. S. Rezende, and C. R. de Souza, "Learning with average precision: Training image retrieval with a listwise loss," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5107–5116.



**Chengxu Liu** (Graduate Student Member, IEEE) received the B.E. degree in information engineering in 2019 from Xi'an Jiaotong University, Xi'an, China, where he is currently working toward the Ph.D. degree in information and communication engineering.

From 2021 to 2022, he was an Intern with the Multimedia Search and Mining Group, Microsoft Research Asia, Beijing, China. His current research interests include fine-grained image classification, object detection, video super-resolution, video frame interpolation, and image enhancement.



**Zongyang Da** received the B.E. degree in information engineering and the M.E. degree in information and communication engineering from Xi'an Jiaotong University, Xi'an, China, in 2020.

His current research interests include single-image super-resolution, blind super-resolution, object detection.



**Yuanzhi Liang** received the B.E. degree in information engineering from Lanzhou University, Lanzhou, China, in 2017, and the M.E. degree in software engineering from Xi'an Jiaotong University, Xi'an, China. He is currently working toward the Ph.D. degree in computer science with Australian Artificial Intelligence Institute, University of Technology Sydney, Ultimo, NSW, Australia.

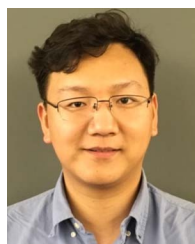
His current research interests include visual relationships, fine-grained image classification,

and object detection.



**Yao Xue** received the B.E. degree in information engineering from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2010, the M.E. degree in information and communication engineering from Xi'an Jiaotong University, Xi'an, China, and the Ph.D. degree in computer science from the University of Alberta, Edmonton, AB, Canada, in 2018.

He is currently a Lecturer with Xi'an Jiaotong University. His research interests include computer vision, medical image analysis, machine learning, and artificial intelligence.



**Guoshuai Zhao** (Member, IEEE) received the B.E. degree from Heilongjiang University, Harbin, China, in 2012, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2015 and 2019, respectively, all in software engineering.

He was an Intern with the Social Computing Group, Microsoft Research Asia, Beijing, China, in 2017, and was a Visiting Scholar with Northeastern University, Boston, MA, USA, from 2017 to 2018 and with MIT, Cambridge, MA, USA, in 2019. He is currently an Assistant Professor with Xi'an Jiaotong University. His research interests include social media Big Data analysis, recommender systems, and natural language generation.



**Xueming Qian** (Member, IEEE) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2008, all in electronics and information engineering.

From 2011 to 2014, he was an Associate Professor with Xi'an Jiaotong University, where he is currently a Full Professor and the Director of SMILES Lab. From 2010 to 2011, he was a Visiting Scholar with Microsoft Research Asia,

Beijing, China. His current research interests include social media Big Data mining and search.

Dr. Qian was the recipient of the Microsoft Fellowship in 2006 and the Outstanding Doctoral Dissertations of Xi'an Jiaotong University and Shaanxi Province, in 2010 and 2011, respectively.