Rethinking Object Detection in Retail Stores

Yuanqiang Cai[1,2*], Longyin Wen[3*], Libo Zhang[1†], Dawei Du[4], Weiqiang Wang[2]
[1]**Institute of Software Chinese Academy of Sciences, China**
[2]**University of the Chinese Academy of Sciences, China**
[3]**JD Finance America Corporation, Mountain View, USA**
[4]**University at Albany, SUNY, USA**

Abstract

The conventional standard for object detection uses a bounding box to represent each individual object instance. However, it is not practical in the industry-relevant applications in the context of warehouses due to severe occlusions among groups of instances of the same categories. In this paper, we propose a new task, *i.e.*, simultaneously object localization and counting, abbreviated as *Locount*, which requires algorithms to localize groups of objects of interest with the number of instances. However, there does not exist a dataset or benchmark designed for such a task. To this end, we collect a large-scale object localization and counting dataset with rich annotations in retail stores, which consists of $50,394$ images with more than 1.9 million object instances in 140 categories. Together with this dataset, we provide a new evaluation protocol and divide the training and testing subsets to fairly evaluate the performance of algorithms for Locount, developing a new benchmark for the Locount task. Moreover, we present a cascaded localization and counting network as a strong baseline, which gradually classifies and regresses the bounding boxes of objects with the predicted numbers of instances enclosed in the bounding boxes, trained in an end-to-end manner. Extensive experiments are conducted on the proposed dataset to demonstrate its significance and the analysis is provided to indicate future directions. Dataset is available at https://isrc.iscas.ac.cn/gitlab/research/locount-dataset.

Introduction

Object detection is one of the most fundamental tasks in the computer vision community, which aims to answer the question: "*where are the instances of the particular object classes?*". It is extremely useful in the retail scenarios, such as identifying commodity on the shelves to provide review or price information, and the navigation in supermarkets, to promote the sales. The conventional standard uses

a bounding box to represent object instance. However, it is not achievable in the industry-relevant applications in the context of warehouses due to the severe occlusions among groups of instances of the same categories. For example, as shown in Fig. 1(g), it is extremely difficult to annotate the stacked dinner plates even by a well-trained annotator. Meanwhile, it is almost impossible for object detectors to detect all stacked dinner plates accurately, even for the state-of-the-art detectors[1]. Thus, it is necessary to rethink the definition of object detection in such scenarios.

Inspired by the definitions of object detection (Dalal and Triggs 2005; Zhang et al. 2018) and crowd counting (Lempitsky and Zisserman 2010; Zhang et al. 2016), we propose a new task, *i.e.*, simultaneously object localization and counting, abbreviated as *Locount*, which requires algorithms to localize groups of objects of interest with the number of instances. Specifically, as shown in Fig. 1(g) and (h), if some object instances are severely occluded each other or belonging to the same commodity, we use the minimum enclosing box with a predicted instance number to indicate this group of instances. To the best of our knowledge, there does not exist a dataset or benchmark attempt to solve this issue in the retail scenarios. That is, object detection and crowd counting problems are considered individually by their own evaluation protocols.

To solve the above issues, we collect a large-scale object localization and counting dataset at 28 different stores and apartments, which consists of $50,394$ images with the JPEG image resolution of 1920×1080 pixels. More than 1.9 million object instances in 140 categories (including *Jacket*, *Shoes*, *Oven*, *etc.*) are annotated. To facilitate data usage, we divide the dataset into two subsets, *i.e.*, *training* and *testing* sets, including $34,022$ images for training and $16,372$ images for testing. Meanwhile, to fairly evaluate the performance of algorithms in the *Locount* task, we design a new evaluation protocol inspired by conventional object detection and counting protocols (Lin et al. 2014; Zhang et al. 2016). It can penalize algorithms for missing object instances, for duplicate detections of one instance, for false positive detec-

---

---

[1]Most of the state-of-the-art object detectors use non-maximal suppression (NMS) to post-process object proposals to produce final detections. Specifically, it filters the proposals based on intersection-over-union (IoU) between proposals and then most of the stacked dinner plates may fail to be detected.
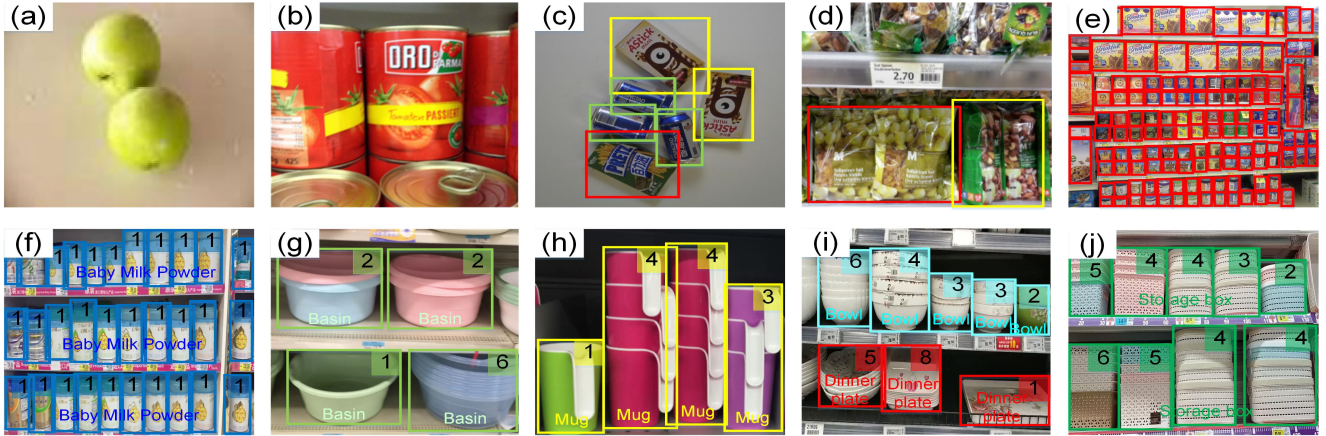
Figure 1: The previous object recognition datasets in grocery stores have focused on image classification, *i.e.*, (a) Supermarket Produce (Rocha et al. 2010) and (b) Grozi-3.2k (George and Floerkemeier 2014), and object detection, *i.e.*, (c) D2S (Follmann et al. 2018), (d) Freiburg Groceries (Jund et al. 2016), and (e) Sku110k (Goldman et al. 2019). We introduce the Loccount task, aiming to localize groups of objects of interest with the numbers of instances, which is natural in grocery store scenarios, shown in the last row, *i.e.*, (f), (g), (h), (i), and (j). The numbers on the right hand indicate the numbers of object instances enclosed in the bounding boxes. Different colors denotes different object categories. Best viewed in color and zoom in.

tions, and for false counting numbers of detections.

Moreover, we present a cascaded localization and counting network (CLCNet) as a strong baseline, to solve object localization and counting simultaneously. Specifically, inspired by Cascade R-CNN (Cai and Vasconcelos 2018), our CLCNet gradually classifies and regresses the bounding boxes of objects and counts the number of instances enclosed in the predicted bounding boxes with increasing IoU and count thresholds, respectively. As shown in Fig. 1(g), for the counting problem, it is challenging to predict the accurate numbers of instances enclosed in the bounding boxes due to similar appearance, especially for the stacked objects (*e.g.*, bowls and dinner plates). To that end, we design a coarse-to-fine multi-stage classification process to gradually narrow the ranges of instance numbers instead of directly regressing instance numbers, to generate accurate results. We define the quality of a hypothesis as its localization intersection-over-union (IoU) and counting accuracy (CA) with the ground-truth, and use the increasing IoU thresholds and more accurate counting partition to generate positives/negatives for training. The whole CLCNet is trained in an end-to-end manner with the multi-task loss, formed by three terms, *i.e.*, classification loss, regression loss, and counting loss. Extensive experiments are conducted on the proposed dataset to demonstrate its effectiveness for *Locount*. We also provide the analysis to indicate future directions and improvements.

Contributions. (1) We propose a new task, *i.e.*, *Locount*, which aims to localize groups of objects of interest with the numbers of enclosing instances. (2) We construct a large-scale object localization and counting dataset in retails stores and a new evaluation protocol to evaluate the performance of algorithms for *Locount*. (3) We present the CLCNet method to solve the *Locount* task, which uses a coarse-to-fine multi-stage process to gradually classify and regress the bounding

boxes of objects and narrow the ranges of instance numbers, instead of directly regressing them, to generate accurate results. (4) Extensive experiments are conducted on the proposed dataset to validate the effectiveness of the proposed methods, and the analysis is provided to indicate future directions.

Related work

Existing datasets. Commodity detection is critical for several applications in the retail scenarios. Several datasets are collected to boost the research and development in such field. The SOIL-47 dataset (Koubaroulis, Matas, and Kittler 2002) contains only 987 images with 47 product categories for object recognition. The Supermarket dataset (Rocha et al. 2010) focuses on recognizing fruits and vegetables, which consists of 2,633 images in 15 categories. The images in the dataset contains one or more items belonging to the same category with pure color backgrounds in various poses. D2S (Follmann et al. 2018) is designed for product detection and recognition, which includes 21,000 images in 60 categories. Each image contains several items belonging to different categories with various poses, illumination conditions, and backgrounds. The RPC dataset (Wei et al. 2019) considers commodity detection in the automatic checkout scenarios, which consists of 83,739 images in 200 categories. However, the aforementioned datasets focus on image classification or commodity detection in constrained scenarios, which are much easier than the commodity detection in supermarkets or shopping malls in the mobile shooting views.

Recently, some attempts focus on the commodity detection task in the supermarket or shopping mall scenarios in the mobile shooting views. Grozi-3.2k (George and Floerkemeier 2014) contains 8,350 images collected from the Internet for training, and 680 images acquired from the real-world supermarket shelves for testing. The SHORT dataset

Table 1: Summary of existing object detection benchmarks in retail stores. "C" indicates the image classification task, "S" indicates the single-class object detection task, and "M" indicates the multi-class object detection task.

datasets	#images	category	#instance	resolution	task	year
SOIL-47 (Koubaroulis, Matas, and Kittler 2002)	987	47	-	576×720	C	2002
Supermarket (Rocha et al. 2010)	2,633	15	-	640×480	C	2010
D2S (Follmann et al. 2018)	21,000	60	72,447	1920×1440	M	2018
RPC (Wei et al. 2019)	83,739	200	421,674	1800×1800	M	2019
Grozi-3.2k (George and Floerkemeier 2014)	9,030	80	11,585	640×450	M	2014
SHORT (Rivera-Rubio et al. 2014)	134,524	30	-	986×653	C	2014
Grocery Shelves (Varol and Kuzu 2015)	354	10	13,000	-	M	2015
Freiburg Groceries (Jund et al. 2016)	5,021	25	-	1920×1080	C	2016
Retail-121 (Karlinsky et al. 2017)	567	122	-	-	M	2017
GameStop (Karlinsky et al. 2017)	1,039	3,700	-	1200×900	C	2017
TGFS (Hao, Fu, and Jiang 2019)	38,027	24	38,027	480×640	M	2019
Sku110k (Goldman et al. 2019)	11,762	1	1,733,711	1920×2560	S	2019
Ours	50,394	140	1,905,317	1920×1080	M	2020

(Rivera-Rubio et al. 2014) contains several high-resolution grocery product images captured by hand-held smart phones under various illumination conditions. The Freiburg Groceries dataset (Jund et al. 2016) consists of 5,021 images covering 25 different classes of groceries, including 4,749 images for training and 74 images for testing. The grocery shelves dataset (Varol and Kuzu 2015) uses 4 cameras to acquire 354 images in 10 product categories from the shelves in approximate 40 stores, which includes 13,000 groceries. Karlinsky *et al.*(Karlinsky et al. 2017) collect two datasets, *i.e.*, the GameStop dataset, and the Retail-121 dataset, for fine-grained recognition. The former one consists of 5 video clips including 3,700 categories of game chunks acquired from retail stores, while the later one contains 2 video clips with several products in 121 retail product categories. The T-GFS dataset (Hao, Fu, and Jiang 2019) contains 38,027 images in 24 fine-grained categories, which is acquired in the self-service vending machines for automatic self-checkout. The Sku110k dataset (Goldman et al. 2019) provides 11,762 images with more than 1.7 million annotated bounding boxes captured in densely packed scenarios, including 8,233 images for training, 588 images for validation, and 2,941 images for testing. In contrast to the aforementioned datasets, our dataset focuses on commodity detection on the shelve, where some groceries are severely occluded each other and densely packed, such as the stacked basin in Fig. 1(g). Meanwhile, we focus on commodity detection of 140 different categories, which is much more challenging than the one-class groceries detection task in (Goldman et al. 2019). The detailed comparisons of the proposed dataset with other related datasets are presented in Table 1.

Object detection algorithms. Object detection requires algorithms to produce a series of bounding boxes with category scores, which can be roughly divided into two categories, *i.e.*, anchor-based approach and anchor-free approach. The anchor-based approach uses the anchor boxes to generate object proposals, and then determines the accurate object regions and the corresponding class labels using convolutional networks. For example, Faster R-CNN (Ren et al. 2017) designs the region proposal network to generate proposals and uses Fast R-CNN (Girshick 2015) to produce accurate

bounding boxes and class labels of objects. Cascade R-CNN (Cai and Vasconcelos 2018) proposes a multi-stage object detection architecture, which is formed by a sequence of detectors trained with increasing IoU thresholds. Considering the efficiency, SSD (Liu et al. 2016), RetinaNet (Lin et al. 2017), and RefineDet (Zhang et al. 2018) omit the proposal generation step and tile multi-scale anchors at different layers, which run very fast and produce competitive detection accuracy. Recently, the anchor-free approach attracts much attention of researchers, including CornerNet (Law and Deng 2018), CenterNet (Zhou, Wang, and Krähenbühl 2019), FCOS (Tian et al. 2019), RepPoint (Yang et al. 2019), which generally produces the bounding boxes of objects by learning the features of several object key-points. The anchor-free approach has shown great potential to surpass the anchor-based approach in terms of both accuracy and efficiency.

Object counting algorithms. Object counting methods aim to predict the total number of objects in different categories existing in images, such as pedestrian counting (Lempitsky and Zisserman 2010; Zhang et al. 2015, 2016; Liu, van de Weijer, and Bagdanov 2018), vehicle counting (Guerrero-Gómez-Olmedo et al. 2015; Zhang et al. 2017), goods counting (Li et al. 2019; Goldman et al. 2019) and general object counting (Chattopadhyay et al. 2017; Laradji et al. 2018; Cholakkal et al. 2019). In contrast to the *Locount* task, the aforementioned methods are always based on image-level statistics, which only require algorithms to produce the centers of objects. The count numbers associated with the bounding boxes in our dataset (see Fig. 1(g)) is used to indicate the number of instances enclosed in the bounding boxes, designing to bypass the severe occlusion challenge in real-world applications.

In addition, the most related work (Chen, Fern, and Todorovic 2015) introduce the person count localization task, aiming to produce detections covering both isolated individuals and cluttered groups of people with the counts. However, the definitions of the isolated and groups of people are ambiguous, bring the difficulties in annotating for evaluation and designing algorithms. In contrast, we clearly define the *Locount* task, which requires algorithms to localize and count
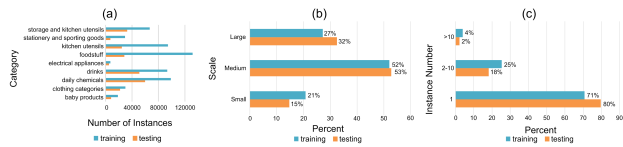
Figure 2: Attribute statistics of the *Locount* dataset. (a) The object category distribution, (b) the scale distribution of objects, and (c) the instance numbers of the annotated bounding boxes, in the *training* and *testing* subsets.

multi-class commodities in retail scenarios.

The Locount Dataset

The *Locount* dataset is formed by $50,394$ JPEG images with the resolution of 1920×1080 pixels. Notably, to ensure the diversity, we acquire the dataset at 28 different stores and apartments with various illumination conditions and shooting angles.

Data collection and annotation. As mentioned above, we acquire the dataset at 28 different stores and apartments. The dataset contains 140 common commodities, including 9 big subclasses, *i.e.*, Baby Stuffs (*e.g.*, *Baby Diapers* and *Baby Slippers*), Drinks (*e.g.*, *Juice* and *Ginger Tea*), Food Stuff (*e.g.*, *Dried Fish* and *Cake*), Daily Chemicals (*e.g.*, *Soap* and *Shampoo*), Clothing (*e.g.*, *Jacket* and *Adult hats*), Electrical Appliances (*e.g.*, *Microwave Oven* and *Socket*), Storage Appliances (*e.g.*, *Trash* and *Stool*), Kitchen Utensils (*e.g.*, *Forks* and *Food Box*), and Stationery and Sporting Goods (*e.g.*, *Skate* and *Notebook*). Please refer to the supplementary materials for more details. There are various factors challenging the performance of algorithms, such as scale changes, illumination variations, occlusion, similar appearance, clutter background, blurring and deformation, *etc*.

More than $1,905,317$ object instances are annotated in the proposed *Locount* dataset. Specifically, we hired 15 experts to label the bounding boxes with the instance numbers using the Colabeler tool [2] for 250 hours per person. If some object instances with the same category are severely occluded each other, *i.e.*, the overlap scores are larger than 0.5, we use the minimum enclosing box to indicate the group of instances with a instance number; otherwise, we annotate each instance using an individual bounding box. We conduct several rounds of cross-check to ensure high quality annotations. The *Locount* dataset is divided into two subsets, *i.e.*, *training* set and *testing* set. There are $34,022$ images with $1,437,166$ instances in the *training* subset, and $16,372$ images with $468,151$ instances in the *testing* subset. The images from these two subsets are captured in different locations, but share similar conditions and attributes. This setting is designed to reduce the chances of algorithms to overfit to particular scenarios. In addition, for better data usage, especially for the performance analysis of algorithms, we also annotate several attributes of objects, shown as follows.

- **Object categories.** We group the object categories in our *Locount* dataset in the hierarchical structure, which

---

[2]http://www.colabeler.com/

is formed by 9 big sub-groups including Baby Stuffs, Drinks, Foodstuff, Daily Chemicals, Clothing, Electrical Appliances, Storage Appliances, Kitchen Utensils, and Stationery and Sporting Goods. Each sub-group is further divided into several sub-classes, and the common products in retail stores are covered by our *Locount* dataset. The number of instances in these 9 sub-groups in the *training* and *testing* subsets are presented in Fig. 2(a). The detailed category distributions are summarized in the supplementary materials.

- **Object scales.** We use the square root of the area of bounding box in pixels to indicate its scale, and divide three subsets based on the scales of objects, *i.e.*, *small scale* subset ($< 150^2$ pixels), *medium scale* subset (150^2-300^2 pixels), and *large scale* subset ($> 300^2$ pixels). The distribution of object scales in the *training* and *testing* subsets are shown in Fig. 2(b).

- **Object numbers.** As described above, we associate an integer to each bounding box to indicate the number of instances enclosed in the bounding box, see Fig. 1 (i), (j), (k), and (l). To facility analysis, we divide the dataset into three subsets based on the instance numbers associated on the bounding boxes, *i.e.*, *individual number* subset (number equals to 1), *medium number* subset (number is $2 - 10$), and *large number* subset (number > 10). The instance number distribution in the *training* and *testing* subsets are presented in Fig. 2(c).

Evaluation protocol. To fairly compare algorithms on the *Locount* task, we design a new evaluation protocol, which penalizes algorithms for missing object instances, for duplicate detections of one instance, for false positive detections, and for false counting numbers of detections. Inspired by MS COCO (Lin et al. 2014), we design new metrics AP^{lc}, $AP^{lc}_{0.5}$, $AP^{lc}_{0.75}$, and $AR^{lc}_{max=150}$ to evaluate the performance of methods, which takes both the localization and counting accuracies into account. Specifically, a correct detection should satisfied two criteria, (1) the localization intersection over union, $IoU = \frac{\widehat{B} \cap B^*}{\widehat{B} \cup B^*}$, between the predicted bounding box $\widehat{B}$ and the ground-truth bounding box B^* is larger than the threshold θ_l, *i.e.*, $IoU \geq \theta_l$; and (2) the counting accuracy, $AC = \max\left(0, 1 - \frac{|\widehat{C} - C^*|}{C^*}\right)$, between the predicted instance number enclosed in the predicted bounding box $\widehat{C}$ and the ground-truth instance number C^* is larger than the threshold θ_c, *i.e.*, $AC \geq \theta_c$. After that, AP^{lc} is computed by averaging over all 10 IoU thresholds, *i.e.*, $\theta_l \in [0.50, 0.95]$ with the uniform step size 0.05, and 10 AC thresholds, *i.e.*, $\theta_c \in [0.50, 0.95]$ with the uniform step size 0.05, of all categories, which is used as the primary metric for ranking algorithms. Note that, to indicate the localization accuracy, we also report the results of evaluated methods using the MS COCO protocol for reference.

CLCNet

We design a cascaded localization and counting network (CLCNet) to solve the *Locount* task, which gradually classifies and regresses the bounding boxes of objects, and es-
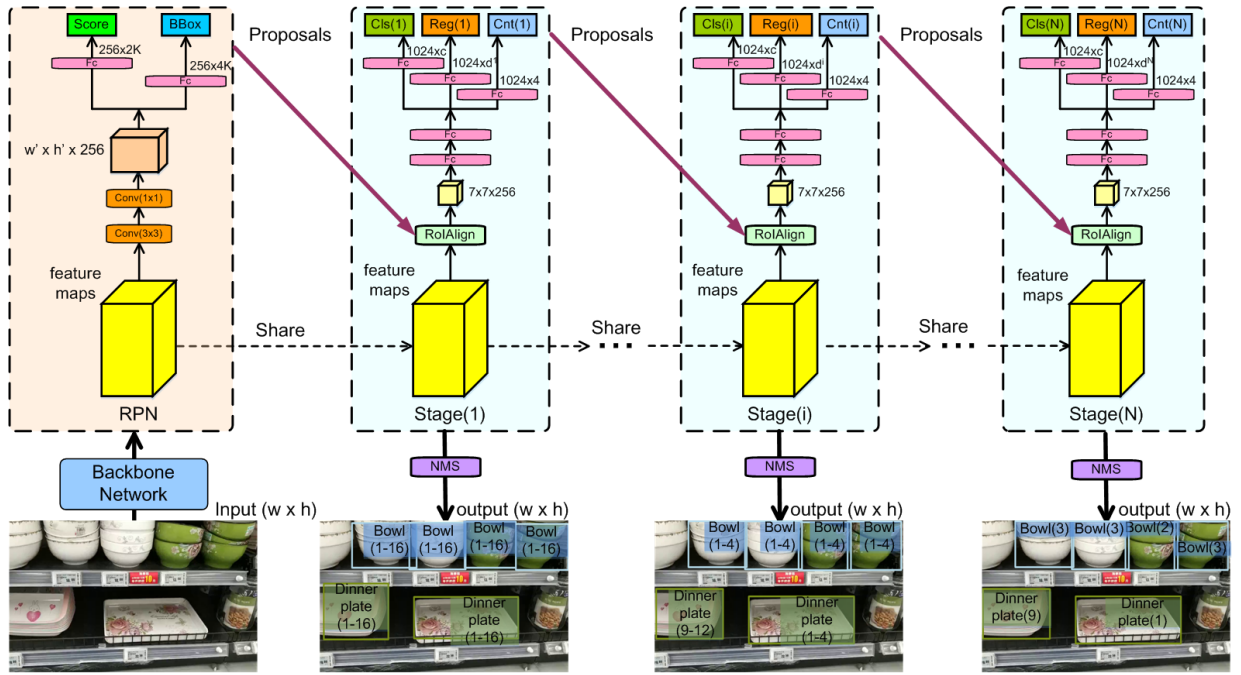
Figure 3: The architecture of our CLCNet for the *Locount* task. The cubes indicate the output feature maps from the convolutional layers or RoIAlign operation. The numbers in the brackets indicate the range of counting number in each stage.

timates the number of instances enclosed in the predicted bounding boxes, with the increasing IoU and count number threshold in training phase. The architecture of the proposed CLCNet is shown in Fig. 3. The entire image is first fed into the backbone network to extract features. A proposal sub-network (denoted as "S_0") is then used to produce preliminary object proposals. After that, given the detection proposals in the previous stage, multiple stages for localization and counting, *i.e.*, $S_1, \cdots, S_N$ are cascaded to generate final object bounding boxes with classification scores and the number of instances enclosed in the bounding box, where N is the total number of stages. For the i-th stage S_i, it takes the features generated by the ROIAlign operation (He et al. 2017) to produce the intermediate classification score, object bounding box, and the number of instances. That is, the features are fed into three sibling fully connected (FC) layers, *i.e.*, a box-regression layer, a box-classification layer, and an instance counting layer to generate the final results. Notably, the localization IoU threshold in the i-th stage used to generate the positive/negative samples in training phase is set to $0.5 + (i-1) \cdot v_l$, where v_l is a pre-defined increasing parameter. The counting accuracy threshold for the positive/negative sample generation is determined by the architecture design of CLCNet, which is described as follows.

We use the same architecture and configuration as (Cai and Vasconcelos 2018) for the box-regression and box-classification layers. For the instance counting layer, a direct strategy is to use a FC layer to regress a floating point number, indicating the number of instances, called *count-regression strategy*. However, the numbers of instances enclosed in the bounding boxes are integers, leading challenges for the network to regress accurately. For example,

if the ground-truth numbers of instances are 4 and 5 for two bounding boxes, and both of the predictions are 4.5, it is difficult for the network to choose the right direction in the training phase. To that end, we design a classification strategy to handle such issue, called *count-classification strategy*. Specifically, we assume the maximal number of instances is α and construct α bins to indicate the number of instances. Thus, the counting task is formulated as the multi-class classification task, which use a FC layer to determine the bin index for instance number.

Notably, as mentioned above, we use the cascade architecture to gradually estimate the instance number with more accurate counting partitions, *i.e.*, the network approaches the accurate number of instances in a coarse-to-fine process. We denote η_i to be the new divided number of classes in the i-th stage. We have $\prod_{i=1}^{k} \eta_i$ number of classes till the k-th stage, where $k = 1, \cdots, N$. To cover all possible numbers of instances, we need to ensure $\prod_{i=1}^{N} \eta_i >= \alpha$ in design. For convenience, we can use the digital base representation to determine the counting division (*i.e.*, the number of bins for the classification task) in each stage. We take the binary representation as an example. Let the maximal number of instances $\alpha = 50$, and $N = 3$ stages in our CLCNet. Thus, 6 digits are more than enough to cover all kinds of possibilities of instance numbers (*i.e.*, $2^6 = 64 > \alpha$). For each stage, we can gradually cover 2 more digits ($\eta_i = 4$, where $i = 1, 2, 3$), *i.e.*, partitioning the value space of the instance number into 4 more parts. To be specific, in the first stage, we only focus on the first 2 digits, *i.e.*, $00, 01, 10$, and 11, of the instance number to generate positive/negative samples. In the second stage, we cover 2 more digits, and
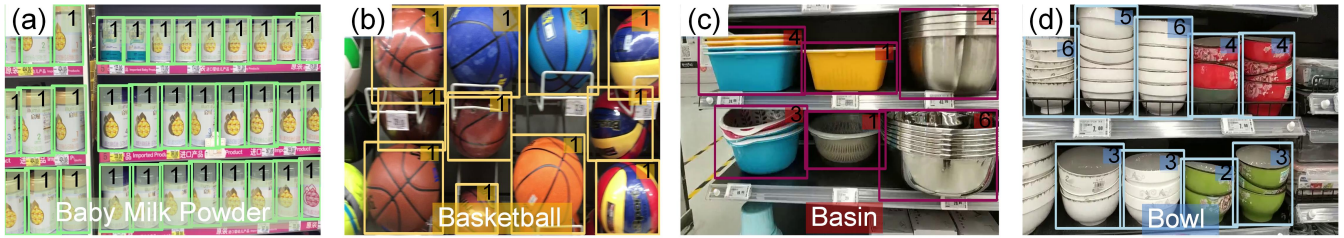
Figure 4: Qualitative results of the proposed CLCNet method on the *Locount* dataset. Best viewed in color and zoom in.

Table 2: Comparison results of the algorithms on the proposed dataset. Detection results of all comparison methods on the proposed dataset. The mark *lc* on the upper right corner indicates that its value is computed by the proposed metrics. Bold fonts indicate the best performance (%).

Method	MS COCO protocol				Proposed protocol			
	AP	$AP_{0.5}$	$AP_{0.75}$	$AR_{max=150}$	AP^{lc}	$AP^{lc}_{0.5}$	$AP^{lc}_{0.75}$	$AR^{lc}_{max=150}$
SSD (Liu et al. 2016)	32.4	54.4	35.6	47.1	27.9	47.5	30.7	42.2
FCOS (Tian et al. 2019)	40.6	56.5	47.5	59.2	37.2	52.2	43.5	55.9
RepPoints (Yang et al. 2019)	42.2	59.0	49.5	57.6	38.8	54.6	45.5	54.3
RetinaNet (Lin et al. 2017)	42.6	59.3	50.0	59.7	37.1	52.1	43.7	53.6
Faster R-CNN (Ren et al. 2017)	45.3	64.3	53.2	55.9	39.7	56.7	46.8	50.2
Cascade R-CNN (Cai and Vasconcelos 2018)	46.8	63.2	54.7	56.2	40.9	55.7	47.8	50.5
CLCNet-s(1)-reg	45.0	62.8	52.8	57.2	40.8	59.0	47.9	53.2
CLCNet-s(2)-reg	46.6	63.1	54.7	56.5	42.6	59.6	50.0	52.7
CLCNet-s(3)-reg	46.2	63.0	54.1	56.3	42.1	59.5	49.3	52.5
CLCNet-s(6)-reg	45.8	62.0	53.6	55.1	38.6	54.9	45.1	49.2
CLCNet-s(1)-cls(2)	45.6	63.2	53.5	57.5	42.3	60.3	49.8	54.4
CLCNet-s(2)-cls(2)	46.7	63.2	54.6	56.4	43.1	60.0	50.5	53.5
CLCNet-s(3)-cls(2)	46.8	63.5	**54.9**	56.2	43.1	60.3	50.7	52.9
CLCNet-s(6)-cls(2)	46.7	62.8	**54.9**	55.2	42.9	59.5	50.5	51.9
CLCNet-s(1)-cls(10)	45.4	62.9	53.4	56.9	42.0	59.8	49.5	53.5
CLCNet-s(2)-cls(10)	**46.9**	63.4	**54.9**	56.2	**43.5**	**60.6**	**51.0**	53.1

use the first 4 digits, *i.e.*, $0000, 0001, 0010, \cdots, 1111$, for sample generation. The rest can be done in the same manner. Along this way, the value space of the instance number can be partitioned into 4, 16, and 64 different parts, and the coarse-to-fine process can be constructed for more accurate counting results. Obviously, the octal, decimal or other base representations can also be used to determine the counting division in the cascade architecture.

Loss function. We use the multi-task loss to train the network in an end-to-end manner, which is formed by three terms, *i.e.*, the classification loss, the regression loss, and the counting loss. The overall loss function is computed as $\mathcal{L} = \frac{1}{N}\big(L_{cls} + \lambda_1 \cdot L_{reg} + \lambda_2 \cdot L_{cnt}\big)$, where L_{cls}, L_{reg}, and L_{cnt} are the classification, regression, and instance counting losses, N is the number of positive anchors in the training phase, and λ_1 and λ_2 are the predefined parameters used to balance these three loss terms. Similar to (Cai and Vasconcelos 2018), we use the cross-entropy loss and smooth L1 loss to compute the classification loss L_{cls} and the regression loss L_{reg}, respectively. Meanwhile, for the *count-regression* and *count-classification* strategies, the smooth L1 and cross-entropy losses are used to compute the counting loss, respectively.

Experiments

Experimental setup. All the evaluated methods are implemented based on the mmdetection platform[3]. For fair comparison, all the evaluated algorithms are trained on the *training* subset and evaluated on the *testing* subset of the proposed *Locount* dataset. For the proposed CLCNet method, we use ResNet-50 with the feature pyramid architecture as the backbone network. In the inference phase, the network outputs top 512 high confident proposals per image. After that, we use the non-maximum suppression with Jaccard overlap of 0.5 and retain the top 150 high confident detections per image to generate the final results. All the experiments are conducted on a machine with 1 NVIDIA Titan Xp GPU and a 2.80GHz Intel(R) Xeon(R) E5-1603 v4 processor. The batch size is set to 8 in the training phase. The whole network is trained using the stochastic gradient descent (SGD) algorithm with the 0.9 momentum and 0.0001 weight decay. The initial learning rate is set to 0.02. We set the incremental parameter v_l of the localization IoU threshold for positive/negation sample generation to 0.05 for six stages and 0.2 for two stages. The predefined parameters λ_1 and λ_2 in the loss function are set to 1.0 and 0.0001 for the count-regression strategy, and set to 1.0 and 0.1 for the count-classification strategy.

---

[3]https://github.com/open-mmlab/mmdetection.

Table 3: Quantitative results of the variants of our CLCNet method on the six subsets determined by the scales of objects (*i.e.*, *small scale*, *medium scale*, and *large scale* subsets), and the instance numbers of bounding boxes (*i.e.*, *individual*, *medium*, and *large* subsets). Bold fonts indicate the best performance (%).

Method	Object scale						Instance number					
	AP_{SS}^{lc}	AP_{MS}^{lc}	AP_{LS}^{lc}	AR_{SS}^{lc}	AR_{MS}^{lc}	AR_{LS}^{lc}	AP_{IN}^{lc}	AP_{MN}^{lc}	AP_{LN}^{lc}	AR_{IN}^{lc}	AR_{MN}^{lc}	AR_{LN}^{lc}
CLCNet-s(1)-reg	23.5	37.8	42.5	**31.4**	50.0	55.9	41.1	17.1	18.7	54.0	28.7	24.2
CLCNet-s(2)-reg	23.5	39.2	45.1	29.8	48.9	56.0	43.0	20.5	18.1	53.4	30.7	20.4
CLCNet-s(3)-reg	23.0	38.7	44.3	30.2	48.8	54.9	42.5	21.3	19.2	53.1	31.8	22.2
CLCNet-s(6)-reg	21.7	35.8	40.7	28.2	45.3	51.9	39.9	11.9	14.3	50.2	21.5	16.2
CLCNet-s(1)-cls(2)	22.1	39.4	44.4	30.5	**51.2**	**57.6**	42.4	26.4	19.1	**54.7**	37.5	23.9
CLCNet-s(2)-cls(2)	23.4	39.4	45.4	30.9	49.0	56.6	43.4	**26.9**	**21.6**	53.8	**37.8**	**26.0**
CLCNet-s(3)-cls(2)	22.6	39.8	45.4	29.1	49.1	56.0	43.3	25.3	17.7	53.4	36.0	23.0
CLCNet-s(6)-cls(2)	23.2	38.9	44.9	29.4	47.3	54.7	43.1	24.8	16.9	52.6	34.6	22.3
CLCNet-s(1)-cls(10)	23.2	38.4	43.7	**31.4**	50.2	56.5	42.0	26.8	21.0	54.1	37.6	25.4
CLCNet-s(2)-cls(10)	**23.6**	**39.9**	**46.3**	30.8	49.2	56.4	**43.7**	26.7	21.1	53.4	37.7	25.9

Quantitative results. As presented in Table 2, we compare our CLCNet method with the state-of-the-art object detectors (*e.g.*, FCOS (Tian et al. 2019), RepPoints (Yang et al. 2019), SSD (Liu et al. 2016), RetinaNet (Lin et al. 2017), Faster R-CNN (Ren et al. 2017), and Cascade R-CNN (Cai and Vasconcelos 2018)), for both the conventional object detection and the proposed *Locount* tasks. Notably, for the *Locount* task, each detected bounding box of the conventional detectors is regarded to enclose only one instance. We use CLCNet-s(N)-reg to denote the CLCNet method with N stages and the *count-regression* strategy for counting, and CLCNet-s(N)-cls(γ) to be the CLCNet method with N stages and γ digital representation in the *count-classification* strategy for counting. Notably, if we use only one stage, CLCNet is reduced to Faster R-CNN (Ren et al. 2017) with counting head.

For the conventional object detection task, we use the evaluation protocol in MS COCO (Lin et al. 2014) to indicate the localization accuracy. As shown in Table 2, our CLCNet method produces comparable localization accuracy compared to its baselines with the *count-classification* strategy, *e.g.*, CLCNet-s(3)-cls(2) *vs.* Cascade R-CNN (Cai and Vasconcelos 2018) and CLCNet-s(1)-cls(10) *vs.* Faster R-CNN (Ren et al. 2017). It indicates that the *count-classification* strategy does not affect the accuracy of object localization. Meanwhile, it worth mentioning that with the *count-regression* strategy, the localization accuracy is affected to some extent, *e.g.*, CLCNet-s(3)-reg *vs.* Cascade R-CNN (Cai and Vasconcelos 2018), and CLCNet-s(1)-reg *vs.* Faster R-CNN (Ren et al. 2017), demonstrating that the floating prediction of counting layer confusing the network to produce accurate results.

For the *Locount* task, we use the proposed protocol to evaluate the performance of algorithms, shown in Table 2. As shown in Table 2, the conventional object detection methods assume that there is only one instance enclosed in each bounding box, resulting in inferior accuracy in terms of AP^{lc}. Among them, Cascade R-CNN (Cai and Vasconcelos 2018) produces the best AP^{lc} score of 40.9%. Meanwhile, our CLCNet method based on either the *count-regression* strategy or the *count-classification* strategy can produce accurate number of instances in the bounding box in some scenarios. Notably, CLCNet-s($\cdot$)-reg perform worse than their counterpart CLCNet-s($\cdot$)-cls($\cdot$), which further validate the effectiveness of the proposed *count-classification* strategy. Overall, the CLCNet-s(2)-cls(10) method achieves the state-of-the-art results with AP^{ls} score 43.5% on our *Locount* dataset, surpassing all other methods.

Ablation study. We further perform experiments to study the influence of the number of stages in CLCNet in terms of object scales and object number attributes in Table 3. We can conclude that using multiple stages generally achieve better results. For example, CLCNet-s(2)-cls($\cdot$) performs better than CLCNet-s(1)-cls($\cdot$) in terms of the AP^{lc} scores in all subsets, see Table 3. It indicates the effectiveness of the coarse-to-fine process in our method. However, using too many stages (more than 2 stages) may cause the over-fitting issue since too many parameters are introduced in the network, resulting in inferior results. For example, CLCNet-s(3)-reg produces 23.0% AP_{SS}^{lc} compared to CLCNet-s(2)-reg with 23.5% AP_{SS}^{lc}. Meanwhile, as shown in Table 3, we find that it is much more difficult to detect smaller objects, *i.e.*, $AP_{SS}^{lc} < AP_{MS}^{lc} < AP_{LS}^{lc}$ for all variants of CLCNet, and also difficult to detect a large groups of instances, *i.e.*, $AP_{LN}^{lc} > AP_{IN}^{lc} > AP_{MN}^{lc}$ for all variants of CLCNet. In summary, there remains much room for improvement for detecting smaller objects and larger groups of instances in the *Locount* dataset.

Conclusions

In this paper, we define a new task *Locount* to localize groups of objects with the instance numbers, which is more practical in retail scenarios. Meanwhile, we collect a large-scale object localization and counting dataset, formed by 50, 394 images with more than 1.9 million annotated object instances in 140 categories. A new evaluation protocol is designed to fairly compare the performance of algorithms on the *Locount* task. We also present the CLCNet method, which uses a coarse-to-fine multi-stage process to gradually classify and regress the bounding boxes, and predict the instance numbers enclosed in the bounding boxes. Finally, we carry out several experiments on the proposed dataset to validate the effectiveness of the proposed method, and analyze the challenging factors of the proposed dataset to indicate future directions.

References

Cai, Z.; and Vasconcelos, N. 2018. Cascade R-CNN: Delving Into High Quality Object Detection. In *CVPR*, 6154–6162. 2, 3, 5, 6, 7

Chattopadhyay, P.; Vedantam, R.; Selvaraju, R. R.; Batra, D.; and Parikh, D. 2017. Counting Everyday Objects in Everyday Scenes. In *CVPR*, 4428–4437. 3

Chen, S.; Fern, A.; and Todorovic, S. 2015. Person count localization in videos from noisy foreground and detections. In *CVPR*, 1364–1372. 3

Cholakkal, H.; Sun, G.; Khan, F. S.; and Shao, L. 2019. Object Counting and Instance Segmentation with Image-level Supervision. *CoRR* abs/1903.02494. 3

Dalal, N.; and Triggs, B. 2005. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 886–893. 1

Follmann, P.; Böttger, T.; Härtinger, P.; König, R.; and Ulrich, M. 2018. MVTec D2S: Densely Segmented Supermarket Dataset. In *ECCV*, 581–597. 2, 3

George, M.; and Floerkemeier, C. 2014. Recognizing Products: A Per-exemplar Multi-label Image Classification Approach. In *ECCV*, 440–455. 2, 3

Girshick, R. B. 2015. Fast R-CNN. In *ICCV*, 1440–1448. 3

Goldman, E.; Herzig, R.; Eisenschtat, A.; Ratzon, O.; Levi, I.; Goldberger, J.; and Hassner, T. 2019. Precise Detection in Densely Packed Scenes. *CoRR* abs/1904.00853. 2, 3

Guerrero-Gómez-Olmedo, R.; Torre-Jiménez, B.; López-Sastre, R. J.; Maldonado-Bascón, S.; and Oñoro-Rubio, D. 2015. Extremely Overlapping Vehicle Counting. In *IbPRIA*, 423–431. 3

Hao, Y.; Fu, Y.; and Jiang, Y. 2019. Take Goods from Shelves: A Dataset for Class-Incremental Object Detection. In *ICMR*, 271–278. 3

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. B. 2017. Mask R-CNN. In *ICCV*, 2980–2988. 5

Jund, P.; Abdo, N.; Eitel, A.; and Burgard, W. 2016. The Freiburg Groceries Dataset. *CoRR* abs/1611.05799. 2, 3

Karlinsky, L.; Shtok, J.; Tzur, Y.; and Tzadok, A. 2017. Fine-Grained Recognition of Thousands of Object Categories with Single-Example Training. In *CVPR*, 965–974. 3

Koubaroulis, D.; Matas, J.; and Kittler, J. 2002. Evaluating Colour-Based Object Recognition Algorithms Using the SOIL-47 Database 2. 2, 3

Laradji, I. H.; Rostamzadeh, N.; Pinheiro, P. O.; Vázquez, D.; and Schmidt, M. W. 2018. Where Are the Blobs: Counting by Localization with Point Supervision. In *ECCV*, 560–576. 3

Law, H.; and Deng, J. 2018. CornerNet: Detecting Objects as Paired Keypoints. In *ECCV*, 765–781. 3

Lempitsky, V. S.; and Zisserman, A. 2010. Learning To Count Objects in Images. In *NeurIPS*, 1324–1332. 1, 3

Li, C.; Du, D.; Zhang, L.; Luo, T.; Wu, Y.; Tian, Q.; Wen, L.; and Lyu, S. 2019. Data Priming Network for Automatic Check-Out. In *ACM MM*. 3

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *ICCV*, 2999–3007. 3, 6, 7

Lin, T.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, 740–755. 1, 4, 7

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. E.; Fu, C.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. In *ECCV*, 21–37. 3, 6, 7

Liu, X.; van de Weijer, J.; and Bagdanov, A. D. 2018. Leveraging Unlabeled Data for Crowd Counting by Learning to Rank. In *CVPR*, 7661–7669. 3

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *TPAMI* 39(6): 1137–1149. 3, 6, 7

Rivera-Rubio, J.; Idrees, S.; Alexiou, I.; Hadjilucas, L.; and Bharath, A. A. 2014. Small Hand-held Object Recognition Test (SHORT). In *WACV*, 524–531. 3

Rocha, A.; Hauagge, D. C.; Wainer, J.; and Goldenstein, S. 2010. Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture* 70(1): 96–104. 2, 3

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. *CoRR* abs/1904.01355. 3, 6, 7

Varol, G.; and Kuzu, R. S. 2015. Toward Retail Product Recognition on Grocery Shelves. In *ICGIP*, volume 9443, 944309. 3

Wei, X.; Cui, Q.; Yang, L.; Wang, P.; and Liu, L. 2019. R-PC: A Large-Scale Retail Product Checkout Dataset. *CoRR* abs/1901.07249. 2, 3

Yang, Z.; Liu, S.; Hu, H.; Wang, L.; and Lin, S. 2019. RepPoints: Point Set Representation for Object Detection. *CoRR* abs/1904.11490. 3, 6, 7

Zhang, C.; Li, H.; Wang, X.; and Yang, X. 2015. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, 833–841. 3

Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; and Li, S. Z. 2018. Single-Shot Refinement Neural Network for Object Detection. In *CVPR*, 4203–4212. 1, 3

Zhang, S.; Wu, G.; Costeira, J. P.; and Moura, J. M. F. 2017. FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras. In *ICCV*, 3687–3696. 3

Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; and Ma, Y. 2016. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *CVPR*, 589–597. 1, 3

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as Points. *CoRR* abs/1904.07850. 3