Terrace-based Food Counting and Segmentation

Huu-Thanh Nguyen,[1] **Chong-Wah Ngo,** [2]
[1] Department of Computer Science, City University of Hong Kong
[2] School of Computing and Information Systems, Singapore Management University
tnguyenhu2-c@my.cityu.edu.hk, cwngo@smu.edu.sg

Abstract

This paper represents object instance as a terrace, where the height of terrace corresponds to object attention while the evolution of layers from peak to sea level represents the complexity in drawing the finer boundary of an object. A multi-task neural network is presented to learn the terrace representation. The attention of terrace is leveraged for instance counting, and the layers provide prior for easy-to-hard pathway of progressive instance segmentation. We study the model for counting and segmentation for a variety of food instances, ranging from Chinese, Japanese to Western food. This paper presents how the terrace model deals with arbitrary shape, size, obscure boundary and occlusion of instances, where other techniques are currently short of.

Introduction

Panoptic segmentation has recently been studied in (Kirillov et al. 2019), aiming to combine the strength of object detection (He et al. 2017) and semantic segmentation (Long, Shelhamer, and Darrell 2015). Specifically, the goal is to pixel-wise extract arbitrary shapes of semantic units, where the shape is beyond bounding box and the unit-of-interest is not only object category but also instance. Panoptic segmentation naturally supports applications like instance counting, which is often achieved by glance-based regression without having to know the object locations (Chattopadhyay et al. 2017). This paper studies panoptic segmentation for food instances on a plate as shown in Fig. 1. Food segmentation is a fundamental step towards portion size estimation for nutrition estimation (Myers et al. 2015). As studied in nutritional science (Khanna et al. 2010), there is correlation between the area and density of food. Under the situation where the scale of food can be estimated, for example by fiducial marker (He et al. 2013), counting the number of pixels per food item already provides basic information for calorie estimation.

Nevertheless, counting and segmenting food instances, as shown in Fig. 1, is difficult. First, instance occlusion is a common phenomenon in food presentation. When the instances occlude or touch each others, the boundaries might not be distinguishable, as shown in Fig. 1a and Fig. 1e. Second, food items are often decorated and topped with different ingredients (Fig. 1b and Fig. 1d). Regularization is required to constrain over-segmentation of toppings as instances. Third, as food images are usually captured in close distance, distortion of size and shape can always be observed due to perspective difference (Fig. 1c). Fourth, the unit of items can be a piece, slice or even a scoop of dish as shown in Fig. 1f. A combination of these factors in general results in a variety of visual appearance in terms of shape and size. To the best of our knowledge, there is yet to have any study in the context of food instance segmentation to address these challenges.

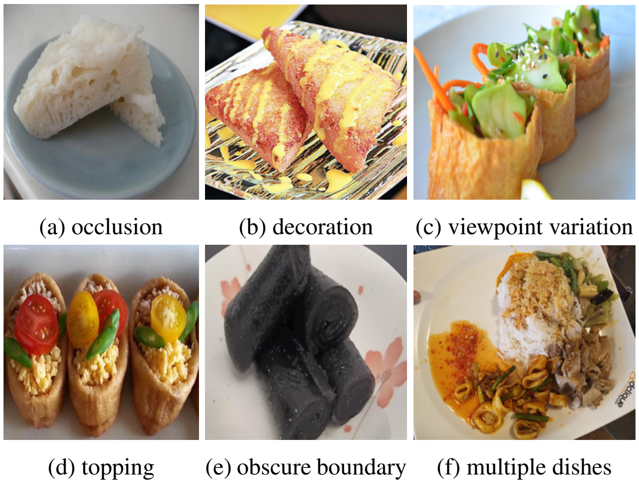

(a) occlusion (b) decoration (c) viewpoint variation
(d) topping (e) obscure boundary (f) multiple dishes

Figure 1: Challenges of counting and segmentation.

The contribution of this paper is proposal of a novel terrace based instance segmentation technique. The method represents instance as a multi-layer terrace (see Fig. 3 for example), where different layers signify the level of difficulty in segmentation. Each layer provides prior to the shape as well as size of an instance. An instance is segmented layer-wise in progressive manner to preserve the shape and size. To regularize instance segmentation to alleviate the adverse effect such as due to food decoration, the height of terrace is estimated as attention for glance-based counting. To this end, terrace is end-to-end learnt as a multi-task neural network for joint segmentation and counting. Last but not least, terrace is class agnostic. The terrace can be clustered into high-quality instance segmentation map as in (Kirillov et al.

	Annotation	Object category	Counting	Segmentation
(Chattopadhyay et al. 2017)	image-level	yes	regression	no
(Laradji et al. 2018)	point-level	yes	blob	partially
(Onoro and Lopez 2016)	point-level	no	density map	partially
(Cholakkal et al. 2019)	image-level	yes	density map	partially
(Tian et al. 2019)	bounding box	no	object detection	partially
(Bai and Urtasun 2017)	instance map	yes	clustering	yes
(Wang et al. 2019a)	instance map	yes	clustering	yes
(Neven et al. 2019)	instance map	yes	clustering	yes
(He et al. 2017)	bounding box + instance map	either	object detection	yes
Ours	terrace polygon	no	clustering	yes

Table 1: An overview of methods for object counting, detection and segmentation

2019; He et al. 2017; Neven et al. 2019) with minimum annotation effort. The ground-truth of an instance is a polygon, which can be created in about 20 times faster than pixel-wise instance labeling.

Related Works

Food instance counting and segmentation is seldom studied in the literature. Having said that, there exist off-the-shelf methods for this problem. Table 1 broadly classifies these methods based on different factors: labeling effort, class specific versus agnostic, counting algorithm and completeness of instance extraction. These methods operates either on everyday domain (e.g., COCO (Lin et al. 2014)), cityscapes (Cordts et al. 2016) or surveillance videos. Directly applying them to address the challenges in food domain is not seriously explored.

Regression is a powerful method and has been applied for counting cells in microscope images (Hernández, Sultan, and Pande 2018) and fruits on a tree (Chen et al. 2017). A general way is by modifying convolutional neural network (CNN) to output a continuous number as count. These approaches are domain-specific and category agnostic, demanding only image-level annotation and assuming the same or similar instance size. Category-specified counting is studied in (Chattopadhyay et al. 2017) to estimate count individually for each object category. Inspired by (Cutini and Bonato 2012), this approach divides an image into non-overlapping grids and performs counting in divide-and-conquer manner by sequential subtilizing on grids. As counting is performed “by glance” without localizing objects, these approaches are not applicable to food instance segmentation.

More advanced approaches are by generating density map (Onoro and Lopez 2016), blobs (Laradji et al. 2018) or instance segmentation (Bai and Urtasun 2017; Wang et al. 2019a; Neven et al. 2019), which indicate instance locations, as the basis of counting. Point and instance-level supervision that roughly annotates the object locations is required. Density map is popularly adopted for counting crowd and animals in surveillance videos. These approaches fit object sizes with Gaussian distribution. The object size often needs to be explicitly estimated such as by camera perspective map (Shi, Yang, and Chen 2019), which are not available in most applications. Blob-based detection (Laradji et al. 2018) is more applicable for general object detection. Nevertheless, only a small portion of an instance can be extracted.

Object detection techniques, such as (Law and Deng 2019; Tian et al. 2019), generally support instance counting. However, limited by bounding box representation, the arbitrary shapes of objects cannot be described. The problem is addressed by proposal-based instance segmentation (He et al. 2017; Bolya et al. 2019), which also produces a mask outlining the object shape inside a bounding box. Proposal-free approach, on the other hand, performs bottom-up image processing, by first labeling of pixels into semantic categories and then grouping them as instance masks (Neven et al. 2019; Wang et al. 2019a; Bai and Urtasun 2017). In SECB (Spatial Embeddings and Clustering Bandwidth (Neven et al. 2019)), each pixel value is embedded to predict the centroid and size of the instance that it belongs to for instance clustering. In PSENet (Wang et al. 2019a), starting from predicting multiple scales of instance seeds, pixels are progressively labeled across scales from the instance seeds towards borders by 8-neighbourhood connectivity analysis. Our proposed terrace model is also proposal-free and the most similar approach is Deep watershed (Bai and Urtasun 2017), which represents an instance as 16-level watershed for pixel labeling. To produce the watershed, each pixel predicts the direction that points to the nearest instance boundary. Terrace also adopts multi-layer instance representation. However, different from Deep watershed, terrace enjoys the simplicity in network design for not performing distance transform to predict direction for every pixel. Instead, a counting subnetwork is included to regulate the formation of terrace. Terrace is also computationally efficient with comparable speed as Yolact (Bolya et al. 2019) and enjoys higher accuracy in instance counting and segmentation than the existing approaches.

In food domain, instance localization are mostly based on off-the-shelf techniques, such as semantic segmentation

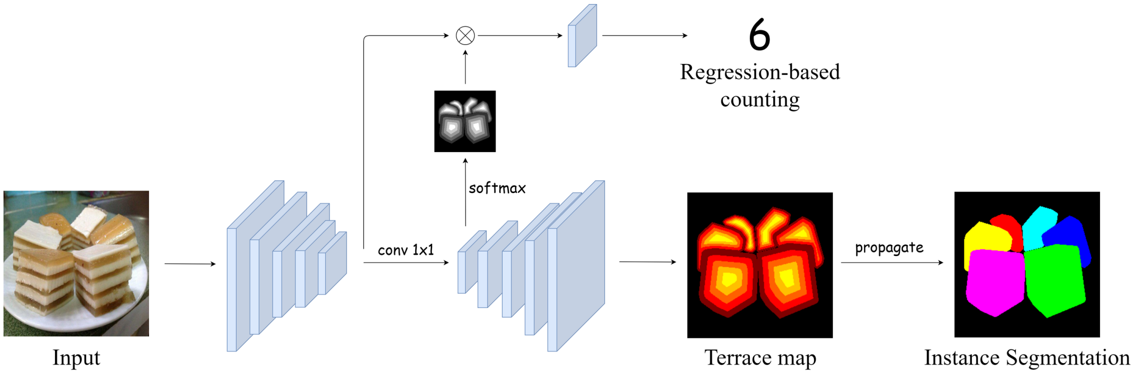


Figure 2: Terrace model for multi-task instance segmentation and counting.

(Aguilar et al. 2018), object detection (Ege et al. 2019; Deng et al. 2019) and DeepLab (Myers et al. 2015). These works do not investigate the problem of food counting. Although localization of bounding boxes inherently supports counting, the boxes cannot adequately describe the arbitrary shapes of food items. Panoptic segmentation (Kirillov et al. 2019), which locates and segments instances, is more applicable.

Terrace-based Instance Segmentation

Fig. 2 sketches the architecture of terrace model. Overall, the network consists of two branches extended from ResNet-50 (He et al. 2016) backbone, respectively, for a terrace map based on Fully Convolutional Network (FCN, (Long, Shelhamer, and Darrell 2015)) and a regression-based counting. The terrace will be post-processed as an instance map. The counting sub-network penalizes over- and under-creation of terrace instances.

Regression-based counting has limitation that food instances cannot be localized for downstream applications. Indeed, a perfect way is by extracting instances explicitly for counting. Nevertheless, counting by localization is particularly difficult due to factors such as occlusion and obscure food boundary. We model food item as a terrace with multiple layers, indicating how each region is far from "sea level", i.e., either other items or background clutter. The centre of a terrace is a "peak" that indicates the attended region for counting. On the other hand, the outer most terrace layer is a risky region that should be segmented in care. In sum, the terrace map representation aims to compromise counting by attending to peaks, and segmentation by preserving the shape of food as far as possible in a progressive manner.

Terrace Model

In the terrace pathway, terrace layers are categorized into $k+1$ labels, where k is the number of layers and the addition label is for background. Denote $p_{n,m} \in \mathbb{R}^{(k+1)\times 1}$ as output of the last deconvolution layer of FCN for a pixel m in an image n. The terrace height probability distribution at this pixel is:

$$p^*_{n,m} = softmax(p_{n,m}) \tag{1}$$

where $p^*_{n,m} \in \mathbb{R}^{(k+1)\times 1}$. Cross-entropy loss is employed to regulate the predicted class labels:

$$L_{CLASS} = \frac{-\sum_n^N \sum_m^M \mathrm{W}_k \log(p^*_{n,m,t_{n,m}})}{NM} \tag{2}$$

where N and M are the number of images and pixels per image respectively, $p^*_{n,m,t_{n,m}}$ is the probability score of pixel (n,m) belonging to ground-truth label $t_{n,m}$, W_k is the weight for label (or layer) k. In practice, the weight W_k should be decreased layer-wise from the inner most layer (i.e., peak) towards outside (i.e., sea level). We set the weight of inner most layer as 2.0, and the value is decremented by 0.2 after each subsequent layer. For example, a 5-layer terrace will have weights for $k+1$ labels as $[2.0, 1.8, 1.6, 1.4, 1.2, 0.5]$, where the weight 0.5 is for pixels classified as background. In the instance labeling stage, class label is picked as the final prediction for terrace layer.

A terrace map exhibits gradual increase or decrease in height level when crossing layers. For an actual height level $t_{n,m}$, a close prediction of $t_{n,m} \pm 1$ should be considered better than a further distance such as $t_{n,m} \pm 2$. Although L_{CLASS} penalizes misclassification errors, this relative distance has not been under consideration. We propose a regression loss function namely L_{HEIGHT} to address this gap. At each pixel, a regression height level $\hat{t}_{n,m}$ is modeled as the integration of its probability distribution over $k+1$ terrace levels:

$$\hat{t}_{n,m} = \mathrm{V}_{k+1} p^*_{n,m} \tag{3}$$

where $\mathrm{V}_{k+1} = [0, 1, .., k] \in \mathbb{R}^{1\times(k+1)}$ amplifies the probability of a layer by a constant equals to its layer id, while suppressing sea level (or background) to 0. Then, the height loss L_{HEIGHT} is computed as following:

$$L_{HEIGHT} = \frac{\sum_n^N \sum_m^M |t_{n,m} - \hat{t}_{n,m}|}{NM} \tag{4}$$

Overall, the loss function for terrace sub-network is formulated as:

$$L_{CONTOUR} = L_{HEIGHT} + L_{CLASS} \tag{5}$$

Instance Labeling

Creation of instance map is basically a clustering process. Starting from the terrace peaks as "seeds", each layer is piled sequentially by grouping pixels being most probably classified to that layer. Along the process, different terraces may compete for pixels. As the network already learns to capture instance shape by classifying pixels to different terrace layers, this mostly happens at the outermost layer. We adopt a simple first-come first-serve strategy for instance labeling. Specifically, a pixel is assigned to a terrace that first reaches it during clustering. Note that, different from the conventional clustering algorithms such as k-means, the terrace map provides prior knowledge about the number and shapes of instances. The shape can be arbitrary and is not necessarily an ellipse as in the density map (Onoro and Lopez 2016). The layer-wise representation of terrace, nevertheless, is similar to density map, which allows growing of instance shape in a "safely" manner. In such case, even when food items are severely overlapped, the first few inner layers of a terrace can be more "safely" piled before creeping into "high risk" region to delineate instance border.

Denote the label for the inner most layer of a terrace map as k. The label is decremented by $k-1$ when continuing to the next subsequent layer. The clustering algorithm is summarized as following:

- Step 1: Perform the connected component analysis to cluster pixels being labeled as layer k. Instance ids are then assigned to each cluster. The pixels neighbouring to the border of layer k is put in a queue S.
- Step 2: Set $k = k - 1$ and initialize an empty queue B.
  - 2a: Retrieve a pixel p from the queue S and traverse the neighbours of p.
  - 2b: Propagate the instance id of p to the pixels that are classified to layer k. Add these pixels to queue S.
  - 2c: If any of the neighbours are not classified to k^{th} layer, p is regarded as a pixel located at the border across two layers. Add p to the queue B.
  - 2d: Repeat step-2a until the queue S is empty.
- Step 3: Copy queue B to queue S.
- Step 4: Repeat step 2 until $k = 1$.

The algorithm, which is linear to the number of pixels in terms of time complexity, is simple and efficient to implement.

Modeling Terrace Height as Attention for Counting

By simply enumerating the number of instances in step-1 of the clustering algorithm, terrace map can be utilized for counting. Nevertheless, when scene complexity is high, false terraces could be predicted resulting in excessive number of counts. For regularization, a regression-based counting is plugged in for simultaneous counting and segmentation. To focus counting on the terrace peaks, a connection is created to fuse the feature maps of the last convolution layer and a small version of attention map predicted from the first deconvolution layer, as shown in Fig. 2. The learning of attention weights in this map is equivalent to estimating the height of terrace.

	Dimsum	Sushi	Cookie	Mixed dishes
#categories	27	11	100	135
#counting	6	6	9	6
#images	3,760	2,877	5,920	9,254

Table 2: Statistics on the number of food categories and images in four food datasets.

Denote the first deconvolution layer as $q_I \in \mathbb{R}^{(k+1)\times S}$ where S is the resolution. Similar intuition as Equation 1, the attention of each terrace layer is learnt by:

$$q_i^* = softmax(q_i) \tag{6}$$

where $q_i^* \in \mathbb{R}^{(k+1)\times 1}$. The terrace height is estimated by:

$$H_I = \mathrm{V}_{k+1} q_I^* \tag{7}$$

where $\mathrm{V}_{k+1} = [0, 1, .., k] \in \mathbb{R}^{1\times(k+1)}$ as defined in Equation 3. The terrace height is subsequently used to weight the feature map f_I of the last convolution layer, as following:

$$\hat{f_i^*} = \frac{H_i f_i}{k} \tag{8}$$

where k in the denominator is to normalize the terrace height. The transformed map $\hat{f_i^*}$ will be further undergone average pooling and passed through a fully-connected layer for regression counting. The loss function of regression-based counting (RC) minimizes the mean absolute error between the actual (r_n) and predicted ($\hat{r}_n$) counts over training examples, as following:

$$L_{RC} = \frac{1}{N}\sum_{n=1}^{N} |r_n - \hat{r}_n| \tag{9}$$

To this end, the loss function of multi-task terrace is:

$$L_{TERRACE} = \lambda_{RC} L_{RC} + \lambda_{CONTOUR} L_{CONTOUR} \tag{10}$$

where λ_{RC} and $\lambda_{CONTOUR}$ are trade-off parameters.

Experimental Setup

Dataset

The experiments are conducted on four datasets: Dimsum, Sushi, Cookie and Mixed dishes, with statistics summarized in Table 2. The first three datasets, represent Chinese, Japanese and Western food respectively, are constructed by crawling images from search engines. The images are manually screened to contain one to nine food items. In some of the images, the items are pieces cut from a whole food. The last dataset, Mixed dishes, is contributed by (Wang et al. 2019b), where each image is composed of multiple dishes placed on a plate. The images are collected from different canteens in a university. The details of datasets are provided in the supplementary document.

The terrace polygon of a food item is manually created for training and validation sets. An annotator only marks the corners of an instance. The line segments between corners are then automatically drawn to form a polygon that approximately encloses the item. The part of instance which is occluded will not be delineated by polygon. On each instance, we calculate the distance from a pixel to its nearest instance boundary. The distances for all the pixels in an instance are then quantized into k different layers, such that each layer has equal thickness. The pixel-wise instance map, labeled with the aid of GrabCut (Rother, Kolmogorov, and Blake 2004), is created for each testing image for evaluation purpose. Each dataset is split into the proportion of 70:20:10 for training, testing and validation respectively.

Performance Measures

The measure mean absolute error (MAE) proposed in (Onoro and Lopez 2016; Laradji et al. 2018) does not take into account the case when count is correctly predicted by chance. For instance, if the model results in a falsely detected instance and a missing instance, the number of count is still correct. We propose a new version of MAE, named MAE*, taking into account localization error. MAE* measures error by enumerating the number of false positives and negatives, as following

$$\text{MAE}^* = \frac{1}{N}\sum_{n=1}^{N}(|FP_n| + |FN_n|) \tag{11}$$

where N is the total number of testing images, FP_n and FN_n are respectively the sets of false positives and negatives in an image. To determine these two sets, one-to-one bipartite graph matching is performed to align the ground-truth and predicted instances based on IoU (Intersection of Union). The instances which is not matched are then identified as either false positives or negatives.

The performance of instance segmentation is measured by Panoptic Quality (PQ) proposed by (Kirillov et al. 2019). PQ first performs one-to-one matching to align ground-truth and segmented instances. A match is considered as a true positive (TP) if its IoU between two instances is more than 0.5. Otherwise, a ground-truth instance is regarded as a false negative (FN), and a segmented instance is treated as false positive (FP). Denoting p and g as the segmented and ground-truth instances respectively, PQ of an image is defined as

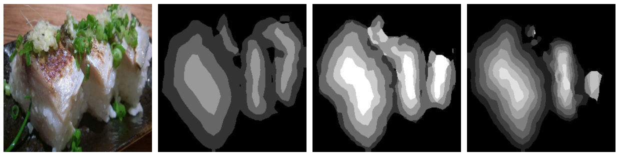
$$\text{PQ} = \frac{\sum_{(p,g)\in TP}\text{IoU}(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{12}$$

In the experiment, the performance is measured by the average of PQ values over all testing images.

Network Setting

All the proposed models are trained using ResNet-50 (He et al. 2016) as backbone. These models are pre-trained on ImageNet[1] dataset. Inspired by (Loshchilov and Hutter 2017), stochastic gradient descent with warm restarts strategy is employed to adjust the learning rate in the ranges of $[10^{-6}, 10^{-4}]$. The cycle length is set equal to 32 times higher than the batch size per epoch. All the models are trained with Adam optimizer and the batch size is set to 16. In the experiment, the model training is stopped after 512 epoches when training loss converges. The trade-off parameters in Equation 10 are set to $\lambda_{RC} = \lambda_{CONTOUR} = 1$.

[1] http://www.image-net.org/



(a) image (b) #layers = 3 (c) #layers = 5 (d) #layers = 7

Figure 3: The terrace maps with different number of layers.

	Single-task		Multi-task	
	MAE*	PQ	MAE*	PQ
Dimsum	0.22	86.13%	**0.18**	**87.29%**
Cookie	0.21	87.99%	**0.17**	**89.15%**
Sushi	0.27	83.35%	**0.22**	**84.98%**
Mixed dishes	0.52	67.92%	**0.45**	**69.30%**

Table 3: Performance comparison between terrace models without (single-task) and with a counting pathway integrated (multi-task) for instance counting (MAE*) and segmentation (PQ)

Experimental Results

Ablation Studies

The number of layers in a terrace is an ad-hoc parameter. Ideally, a larger value facilitates network to attend to the centre-of-mass for counting, while allowing finer levels of instance segmentation. Nevertheless, as each layer corresponds to one class, a larger number of layers can unnecessarily increase the complexity of learning. We experiment the single-task terrace model with the number of layers set to be {3,5,7}. When there is only 3 layers, the terrace suffers from accurate localization of instance border for case when instances are overlapped. By increasing the number to 7, on the other hand, the terrace struggles to classify pixels into different layers, making the post-processing step cumbersome and produces erroneous segments. Fig. 3 shows the terrace maps of different layers of a sushi image. A terrace map is sequentially grown from the inner layer towards outside. When the appearance of a food item is complex, the error in a layer could propagate to the next subsequent layers. Overall, the empirical studies show that setting the number of layers to 5 is a good tradeoff, and shows both the best MAE* and PQ in three of the datasets. In the remaining experiment, we set the number of layers to 5 for terrace model.

Table 3 compares the performances of counting and instance map segmentation between the terrace model without counting pathway (single-task) and the proposed terrace with a counting included (multi-task). In terms of counting, multi-task shows better MAE* by reducing the error from 4% to 11%. The panoptic quality is also improved for all datasets. The most significant segmentation improvements are observed on Sushi dataset, where the additional counting branch successfully constrains the terrace maps from overfitting on the ingredients and sauces on top of sushi items.

	Counting (MAE*)				Segmentation (PQ (%))			
	D	C	S	M	D	C	S	M
Multi-task Terrace	**0.18**	**0.17**	**0.22**	**0.45**	**87.29**	**89.15**	**84.98**	**69.30**
SECB (Neven et al. 2019)	0.39	0.29	0.52	0.62	84.38	87.88	80.22	65.56
PSENet (Wang et al. 2019a)	0.31	0.31	0.52	0.77	85.71	87.75	81.17	65.68
Deep watershed (Bai and Urtasun 2017)	0.57	0.37	0.61	0.75	78.62	85.36	76.00	61.48
Mask R-CNN (He et al. 2017)	0.33	0.42	0.55	1.02	82.81	81.10	78.87	54.52
Yolact (Bolya et al. 2019)	0.32	0.58	0.47	0.83	82.34	81.83	78.15	60.61
CornerNet (Law and Deng 2019)	0.45	0.84	0.63	1.36	NA	NA	NA	NA
FCOS (Tian et al. 2019)	0.35	0.25	0.46	0.75	NA	NA	NA	NA
LC-FCN (Laradji et al. 2018)	0.52	1.17	1.00	4.73	NA	NA	NA	NA
Glance-based (Chattopadhyay et al. 2017)	0.24	0.23	0.27	0.63	NA	NA	NA	NA
Density map (Onoro and Lopez 2016)	0.30	0.33	0.37	0.69	NA	NA	NA	NA

Table 4: Performance comparison with the existing approaches on Dimsum (D), Cookie (C), Sushi (S) and Mixed dishes (M) datasets. NA means not applicable. Note that glance-based and density map are not available for objects localization.

With reference to Fig. 4, we summarize the strength and weakness of terrace model. The generated terrace map manages to delineate item shape satisfactorily, even in case when the presentation of food is complex (Fig. 4a) and with background clutter (Fig. 4b). Both counting and segmentation are benefited from this representation. The map, nevertheless, could be sensitive to food items with different parts. Specifically, different parts of an item are enumerated separately, resulting in over count. This is particularly true in Cookie dataset where the shapes are diverse and decorated into parts, as the example shown in Fig. 4c. Multi-task terrace, with additional branch for regression counting, is effective in constraining the erroneous counting. Note that, for multi-task terrace, the predicted counts in counting and segmentation branches are not necessarily consistent. For complex dish placement, such as the example of mixed dishes in Fig. 4d, direct enumeration of items in instance map often yields better performance.

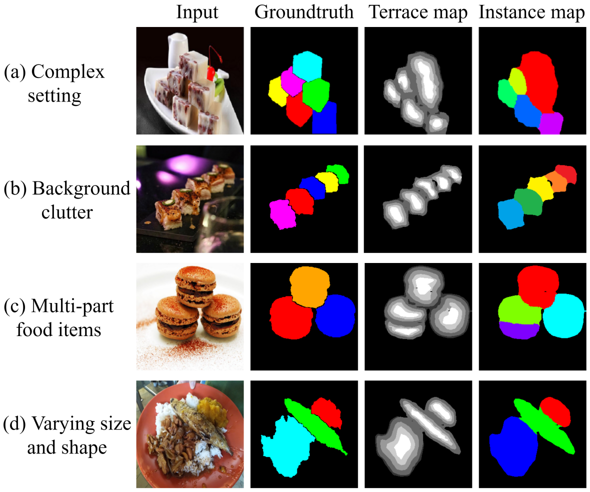


Figure 4: The results of multi-task terrace instance segmentation.

Performance Comparison

As no tailor-made method exists for both food counting and segmentation, we compare terrace model to state-of-the-art techniques in object counting, detection and segmentation. For instance segmentation, we compare to three strong proposal-free methods including SECB (Spatial Embedding and Cluster Bandwidth, (Neven et al. 2019)), PSENet (Wang et al. 2019a), Deep watershed (Bai and Urtasun 2017) and two proposal-based methods Mask R-CNN (He et al. 2017) and Yolact (Bolya et al. 2019). In SECB, each pixel is learnt to predict an offset vector pointing to the instance centre and a margin estimating the size of instance. Different from "terrace levels", PSENet models each instance with 6 masks of different scales ranging from 50% to 100% of the instance size. The mask is expanded progressively across the scales for pixel labeling. Deep watershed produces a contour mask of 16 floors to model an instance. Mask R-CNN requires object proposal. Each predicted proposal is regressed to output a mask for its instance. For counting, we compare to counting methods based on bounding box detection (CornerNet (Law and Deng 2019), FCOS (Tian et al. 2019)), blob detection (LC-FCN (Laradji et al. 2018)). Different from terrace, LC-FCN outputs one blob per instance, instead of estimating the shape and size of instance. Furthermore, glance-based (Chattopadhyay et al. 2017) and density map (Onoro and Lopez 2016) that only predict the number of objects but not localization are also compared.

Table 4 lists the performances of different approaches. As noted, multi-task terrace consistently outperforms all the methods across four different datasets. Fig. 5 gives a snapshot of how different methods address the various challenges in the food domain. LC-FCN works reasonably decent when items do not touch but is not effective on obscure boundary, stacking, close-up of food items and especially Mixed dishes. Bounding box based approaches are relatively poor in locating items that are stacked or with obscure food boundary. Nevertheless, they perform satisfactorily in locating decorated items and mixed dishes, as also reported in (Deng et al. 2019). Mask R-CNN, despite performing slightly better, indeed inherits the weakness of proposal-based method. Erroneous instance maps, either because of miss or false instances, are produced when bounding boxes fail to locate the arbitrary shapes of instances. Deep watershed fails to separate instances from occlusion and stacking. By enforcing pixels to point to instance centre, SECB handles almost as well as terrace model for effects of stacking and obscure boundary. Compared to terrace, nevertheless, false instance segmentation happens more often for multi-part, decorated food instances and mixed dishes. Particularly, when food items are presented in relatively complex setting, the nearby pixels will struggle pointing to different centres, resulting in relative worse PQ quality than terrace. PSENet also appears to be a strong competitor. However, the result is heavily dependent on the prediction of instance masks at the lowest scale. When food items are crowded in a plate, excessive number of masks will be predicted, resulting in over-segmentation when further expanding to masks of larger scale. Without considering object localization, glance-based and density map achieve competitive performances. However, density map, which performs counting based on a learnt Gaussian map, struggles on various shapes and poses of food instances. Occlusion and perspective change make the "divide and conquer" in the glance-based method fails. Our method preserves the shapes of food items by grouping instance pixels in the progressive way from the most confident (terrace peak) to the probable and uncertain regions. As shown in Fig. 5, terrace shows robustness in dealing with various situations than other methods.

Speed Comparison

Fig. 6 visualizes the speed efficiency of instance segmentation approaches. These deep learning networks are built on ResNet-50 backbone (He et al. 2016) and run on a single GPU of GeForce GTX 1080. Between the two proposal-based methods, Yolact (Bolya et al. 2019) optimizes the non-maximum suppression step, producing food instance masks 2.5 times faster than Mask R-CNN (He et al. 2017). For clustering approaches, the difference in computation time depends on not only the network complexity but also the post-processing algorithm. The designs of SECB (Neven et al. 2019), with two deconvolution branches for predicting seed maps and offset vectors, and Deep watershed (Bai and Urtasun 2017), with 2-phase direction net and watershed transform net, are considerably complex. Regarding to post-processing step, PSENet is slow in instance labeling for repeatedly performing pixel queueing and label propagation at six different scales of masks. Terrace is optimised by having a single deconvolution branch and a low-cost counting branch. The instance labeling stage only performs one scan of pixels by queuing the pixels at the borders of different layers for label propagation. The speed of terrace is comparable to Yolact, the fastest instance segmentation to date, at 18.37 frames per second.

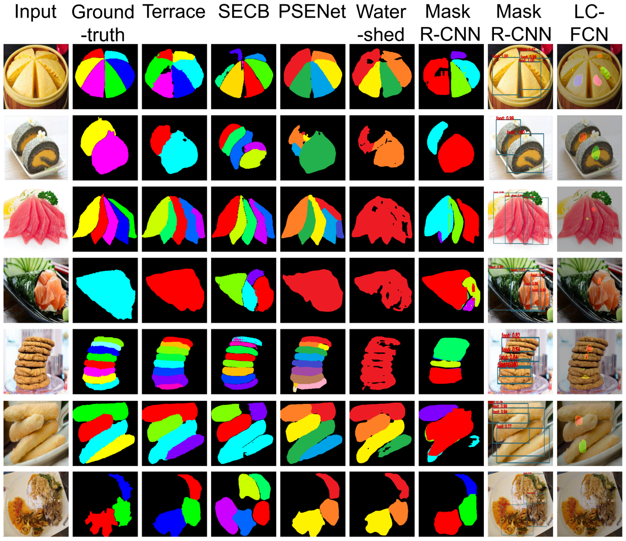


Figure 5: Comparison of panoptic segmentation with state-of-the-art techniques on various challenges. See the supplementary document for more results.

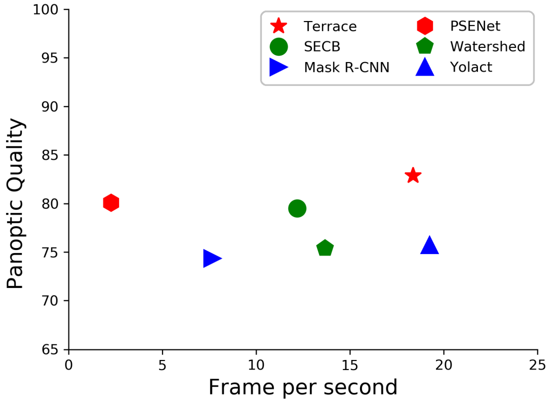


Figure 6: The accuracy versus speed efficiency of various techniques. Measurement is averaged over four datasets.

Conclusion

We have presented a terrace way of segmenting food instances. Empirical studies on four datasets, which covering a wide variety of shape, size and food presentation, justify the merit of this approach in counting and segmentation. The studies also pinpoint some limitations of the existing general-object segmentation techniques and verify the effectiveness of terrace model in dealing with various challenges of food counting and segmentation. Currently, we do not consider options such as adaptive number and width of layers in terrace for more effective representation. Such design is possible by inferencing from scene complexity, which will be our future work.

References

Aguilar, E.; Remeseiro, B.; Bolanos, M.; and Radeva, P. 2018. Grab, Pay, and Eat: Semantic Food Detection for Smart Restaurants. In *IEEE Transactions on Multimedia*, 3266–3275.

Bai, M.; and Urtasun, R. 2017. Deep Watershed Transform for Instance Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2858–2866.

Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019. YOLACT: Real-Time Instance Segmentation. In *The IEEE International Conference on Computer Vision (ICCV).*

Chattopadhyay, P.; Vedantam, R.; Selvaraju, R. R.; Batra, D.; and Parikh, D. 2017. Counting Everyday Objects in Everyday Scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Chen, S.; Skandan, S.; Dcunha, S.; Das, J.; Okon, E.; Qu, C.; Taylor, C.; and Kumar, V. 2017. Counting Apples and Oranges with Deep Learning: A Data Driven Approach. In *IEEE Robotics and Automation Letters*, 781–788.

Cholakkal, H.; Sun, G.; Khan, F. S.; and Shao, L. 2019. Object Counting and Instance Segmentation With Image-Level Supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3213–3223.

Cutini, S.; and Bonato, M. 2012. Subitizing and visual short-term memory in human and non-human species: a common shared system? In *Frontiers in Psychology*.

Deng, L.; Chen, J.; Sun, Q.; He, X.; Tang, S.; Ming, Z.; Zhang, Y.; and Chua, T. S. 2019. Mixed-Dish Recognition with Contextual Relation Networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, 112–120.

Ege, T.; Ando, Y.; Tanno, R.; Shimoda, W.; and Yanai, K. 2019. Image-Based Estimation of Real Food Size for Accurate Food Calorie Estimation. In *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 274–279.

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R. B. 2017. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

He, Y.; Xu, C.; Khanna, N.; Boushey, C. J.; and Delp, E. J. 2013. Food image analysis: Segmentation, identification and weight estimation. In *IEEE International Conference on Multimedia and Expo (ICME)*, 1–6.

Hernández, C. X.; Sultan, M. M.; and Pande, V. S. 2018. Using Deep Learning for Segmentation and Counting within Microscopy Data. *CoRR* abs/1802.10548. URL http://arxiv.org/abs/1802.10548.

Khanna, N.; Boushey, C. J.; Kerr, D.; Okos, M.; Ebert, D. S.; and Delp, E. J. 2010. An Overview of The Technology Assisted Dietary Assessment Project at Purdue University. In *IEEE International Symposium on Multimedia*, 290–295.

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; and Dollar, P. 2019. Panoptic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Laradji, I. H.; Rostamzadeh, N.; Pinheiro, P. O.; Vazquez, D.; and Schmidt, M. 2018. Where are the blobs: Counting by Localization with Point Supervision. In *The European Conference on Computer Vision (ECCV).*

Law, H.; and Deng, J. 2019. CornerNet: Detecting Objects as Paired Keypoints. In *International Journal of Computer Vision*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; and Zitnick, C. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV).*

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431–3440.

Loshchilov, I.; and Hutter, F. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations (ICLR).*

Myers, A.; Johnston, N.; Rathod, V.; Korattikara, A.; Gorban, A.; Silberman, N.; Guadarrama, S.; Papandreou, G.; Huang, J.; and Murphy, K. 2015. Im2Calories: Towards an Automated Mobile Vision Food Diary. In *IEEE International Conference on Computer Vision (ICCV)*, 1233–1241.

Neven, D.; Brabandere, B. D.; Proesmans, M.; and Gool, L. V. 2019. Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Onoro, D. R.; and Lopez, R. J. S. 2016. Towards Perspective-Free Object Counting with Deep Learning. In *European Conference on Computer Vision*, 615–629.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. In *ACM SIGGRAPH*.

Shi, M.; Yang, Z.; and Chen, Q. 2019. Revisiting Perspective Information for Efficient Crowd Counting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7271–7280.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *IEEE International Conference on Computer Vision (ICCV).*

Wang, W.; Xie, E.; Li, X.; Hou, W.; Lu, T.; Yu, G.; and Shao, S. 2019a. Shape Robust Text Detection With Progressive Scale Expansion Network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9328–9337.

Wang, Y.; Chen, J.-J.; Ngo, C.-W.; Chua, T.-S.; Zuo, W.; and Ming, Z. 2019b. Mixed Dish Recognition through Multi-Label Learning. In *Proceedings of the 11th Workshop on Multimedia for Cooking and Eating Activities*.