FoodMask: Real-time Food Instance Counting, Segmentation and Recognition

Huu-Thanh Nguyen[a], Yu Cao[b], Chong-Wah Ngo[b], Wing-Kwong Chan[a]

[a]*Department of Computer Science, City University of Hong Kong, Hong Kong, 999077, China*
[b]*School of Information Systems, Singapore Management University, 178902, Singapore*

Abstract

Food computing has long been studied and deployed to several applications. Understanding a food image at the instance level, including recognition, counting and segmentation, is essential to quantifying nutrition and calorie consumption. Nevertheless, existing techniques are limited to either category-specific instance detection, which does not reflect precisely the instance size at the pixel level, or category-agnostic instance segmentation, which is insufficient for dish recognition. This paper presents a compact and fast multi-task network, namely FoodMask, for clustering-based food instance counting, segmentation and recognition. The network learns a semantic space simultaneously encoding food category distribution and instance height at pixel basis. While the former value addresses instance recognition, the latter value provides prior knowledge for instance extraction. Besides, we integrate into the semantic space a pathway for class-specific counting. With these three outputs, we propose a clustering algorithm to segment and recognise food instances at a real-time speed. Empirical studies are made on three large-scale food datasets, including Mixed Dishes, UECFoodPixComp and FoodSeg103, which cover Western, Chinese, Japanese and Indian cuisines. The proposed networks outperform benchmarks in both terms of instance map quality and speed efficiency.

Keywords: Food counting, Food instance segmentation, Food recognition

Email addresses: tnguyenhu2-c@my.cityu.edu.hk (Huu-Thanh Nguyen), yu.cao.2022@msc.smu.edu.sg (Yu Cao), cwngo@smu.edu.sg (Chong-Wah Ngo), wkchan@cityu.edu.hk (Wing-Kwong Chan)

1. Introduction

Instance counting, segmentation and recognition are basic functions to understand visual world. While human is able to perform these functions "in a glance", devising a neural network for them is a functionally complex task. In the literature, these functions are investigated and composed either loosely as a multi-task learning problem [1] or tightly as a cascaded learning problem [2]. For example, Mask R-CNN [1] and CenterMask [3] are constructed posterior to the typical object detection networks for instance segmentation. Deep watershed [4] and SECB [5] couple together semantic and instance segmentation networks for pixel labelling. Terrace [6] and SibNet [7] train end-to-end multi-task neural networks for simultaneous instance counting and segmentation. HTC [2] and DSC [8] interleave the semantic and instance mask generation sequentially for progressive fine-tuning of results. Despite excellent performances reported by these approaches on various object datasets, it remains challenging to strike a balance between speed, segmentation and recognition accuracy in practice.

In general, using real-time detectors such as Yolact++ [12] can yield speedy and satisfactory performance. Nevertheless, when applying to domain-specific objects, such as curved text [13] and food [6], the accuracy is suboptimal as reported in [7, 14]. This is mainly due to reasons such as the complex shape and structure of an object, which cannot be represented with a generic regular box, resulting in sub-optimal performance of general object detectors [15, 16]. Fig. 1 shows some examples of food where different dishes and their instances are crowdedly stacked. This causes the shape of an instance to appear arbitrary due to occlusion and the shape being distorted due to perspective difference. In the literature, the problem is addressed by predicting the centroids of instances and carefully pushing their territories outward pixel-by-pixel until reaching the most plausible boundary pixels [4, 5]. Such box-free approaches are more robust to densely placed objects as demonstrated in [6, 17] for curved text and multi-dish food datasets. Nevertheless, these approaches are either computationally slow (e.g., Deep watershed [4]) or consider only class-agnostic instance segmentation without classification (e.g., TextMountain [17], SibNet [7]).

In this paper, we investigate a network architecture peculiar to counting, segmenta-

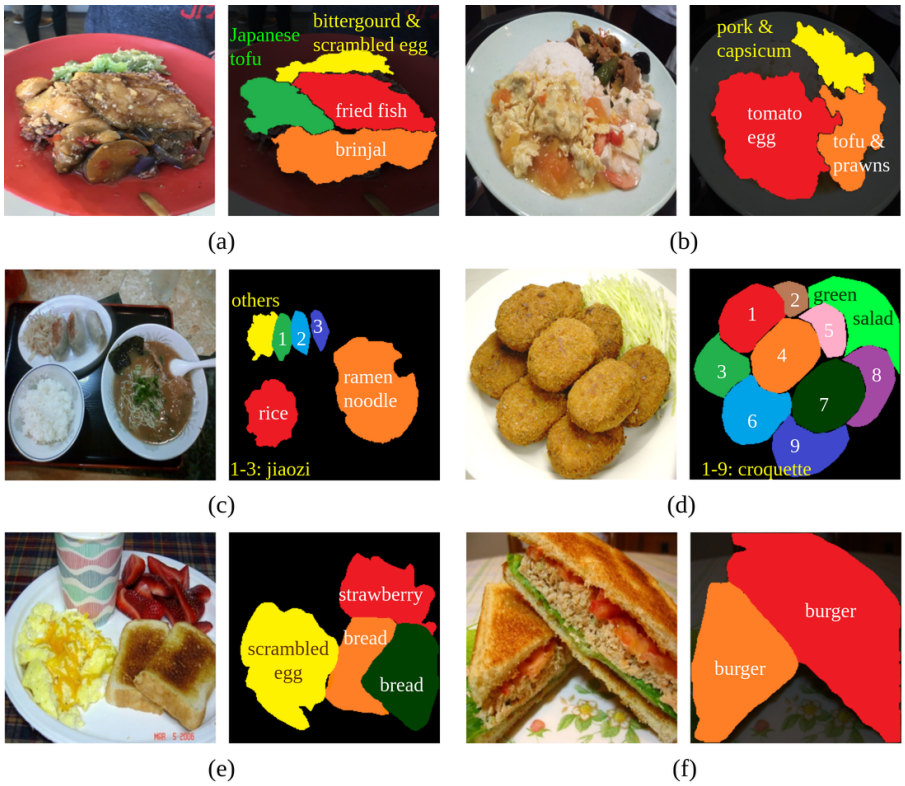

Figure 1: The challenges of food instance segmentation and recognition in Mixed Dishes [9], UECFood [10] and FoodSeg103 [11] datasets. The ground-truth food categories are displayed alongside the images.

tion and recognition of food instances on a plate. Particularly, the speed is targeted to be in real-time, processing at least 30 images per second. We envision such a network to be used for personalized food logging in a free-living environment and the batch processing of images for food packaging and delivery. Fig. 2 shows a plot of speed efficiency versus segmentation quality, which is split into four divisions based on a benchmark of 30 frames-per-second and 57% of panoptic quality [18]. FoodMask is one of the few networks that stays in the top-right division while the majority of the approaches are in the lower-left division. Food-specific networks like Terrace [6] and SibNet [7] manage to produce high panoptic quality but the speed is slower than FoodMask proposed in this paper.

Fig. 1 shows the examples of dishes experimented with in this paper. In Fig. 1a and 1b, the dishes are placed densely one after another. The boundaries between dishes are

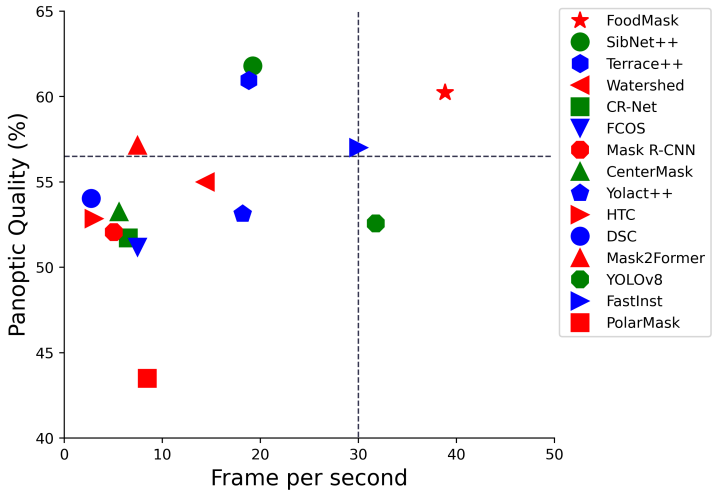

Figure 2: FoodMask strikes a good balance between segmentation quality and speed, and outperforms the existing approaches. Experiments are conducted on three multi-dish food datasets: Mixed Dishes [9], UECFood [10] and FoodSeg103 [11]. The displayed speed is averaged over all the testing images in these three datasets. The size of an input image is 256×256.

fuzzy to separate even with human eyes. These dishes are challenging to be recognized if seasoned with the same sauce on top. Fig. 1c shows a meal with four dishes of different shapes and sizes. One of them, "jiaozi", is severed with multiple items. In Fig. 1d, there are nine instances of "croquette" stacked beside a portion of "green salad". In Fig. 1e, the breakfast is composed of three dishes, including two slices of bread. In Fig. 1f, there are two pieces of "burger" with their ingredients partially visible. As shown in Fig. 1, FoodMask segment and count food instances, while labelling them with dish names.

The proposed FoodMask is a multi-task neural network with three branches for counting, semantic segmentation and contour map generation (see Fig. 3), respectively. To reduce processing time as well as memory consumption, these branches share a deconvolution sub-network. As the creation of a food instance map which involves pixel-level analysis is the most time-consuming component, FoodMask smartly post-processes the predictions from the three branches without parameter learning. In addition, pixel labelling is only performed for the selected regions of a food image. These simplifications of network design not only maintain similar segmentation quality

as in the more sophisticated domain-specific networks (e.g., Terrace [6] and SibNet [7]), but also result in significant speedup during inference time. To be specific, FoodMask accelerates the processing by deriving the semantic and instance labels of a pixel in a rather simplified way (Section 3.3) and enabling the selective clustering of instance labels (Section 3.4) by inferring category-level instance counts. These improvements do not only boost speed efficiency but also maintain high segmentation performance.

The rest of this paper is organized as follows. Section 2 discusses related works for instance counting, segmentation and recognition. Section 3 details the architecture of the proposed network for FoodMask. In addition, an instance labelling algorithm is proposed for rapid post-processing of FoodMask predictions to generate an instance map. Section 4 describes three large-scale food datasets and two metrics to evaluate recognition performances at the instance and pixel levels. Section 5 presents empirical findings to justify the performance of FoodMask in striking a good balance between speed and accuracy. Finally, Section 6 concludes this paper.

2. Related Works

2.1. Instance Counting

Instance counting is generally performed by regression methods such as with Convolutional Neural Network (CNN) [19, 20] that predicts a floating point number indicating the number of object instances in an image. These techniques have evolved from image-level [19] to region-level counting [20] that integrates the category distribution of counts estimated over regions. While being effective and simple in design for providing a glance at object distributions and their counts, the regression-based techniques are not capable of grounding the estimations with localized instances. This limitation inevitably hinders the analysis of instance shape and size which is useful for tasks such as food portion size estimation [21]. The techniques used in crowd counting are effective by labelling object positions as points and generating density maps to summarize object counts. While being highly successful in counting small and uniform objects, such as human head [22] and fish [23], these techniques cannot provide object shape in pixel-level accuracy. More explicit ways of counting are investigated by localizing

the object instances with bounding boxes [9] and masks of arbitrary shapes [7], which directly provide category-wise instance counts. These approaches are generally more sophisticated and computationally expensive for requiring exhaustive evaluation of object proposals [16], semantic segmentation [3], pixel-level connectivity analysis and clustering [7] for instance segmentation.

2.2. Proposal-based Instance Segmentation

Object detection networks, such as Faster R-CNN [15] and FCOS [16], are employed to enclose instances with bounding boxes for counting. These networks have been extended for food instance localization by addressing the specific challenges in the food domain. To explore the co-occurrence of dishes observed on a plate, CR-Net [9] proposes a relation module on top of Faster R-CNN to exploit such relationship for refinement of multi-dish categorization. As food images are susceptible to visual variations due to environmental change (e.g., lighting condition, viewing angle, cooking method), a domain adaptation network is also investigated in [24]. Owing to the bounding box representation, this line of approaches is not able to delineate the instance shapes which are wildly different across different dish categories. As a result, the bounding boxes detected on a plate can be highly occluded with overlapping regions.

Instance segmentation is more feasible in separating the food instances of arbitrary shapes. The approaches include Mask R-CNN [1] and CenterMask [3], which extend object detection techniques by performing semantic segmentation to craft the shape of an instance in a bounding box. A more popular approach is by Yolact++ [12], which detects object proposals and generates instance masks simultaneously with high efficiency. More sophisticated, but also more computationally expensive, approaches involve progressive refinement of instance localization. These include CD-Net [25] which employs graph convolutional network and the cascading networks (e.g., HTC [2], DSC [8]) which interleave bounding box regression and mask prediction for performance boosting. In addition, Mask encoding [26] presents a technique for mask representation at high resolution, ensuring high-quality mask reconstruction while minimizing computational overhead. More recently, Mask2Former [27], a transformer-based universal image segmentation architecture, replaces the cross-attention with masked attention

in the transformer decoder. Mask2Former [27] uses the learnable query features to provide region proposals, improving training efficiency [28] and model generalization to different image segmentation tasks. A faster version is proposed in FastInst [29], which leverages instance activation-guided queries to achieve comparable results with high efficiency using transformer decoder (versus a cascade of transformers of multi-scale processing in [27]). UniInst [30] proposes to learn a re-ranking map at image scale where the value for each map pixel is the quality score of the corresponding instance segmentation candidate. As the score is pixel-based, it suffers from the potential risk of over-segmentation on multi-part instances.

2.3. Proposal-free Instance Segmentation

Despite superior performance demonstrated for the general object detection and segmentation such as on COCO dataset [31], most of these approaches are incompetent in the settings specific for the object domains such as vehicle [32], text [13] and food [6]. To be resilient to instances of varying shapes that are densely placed in a complex setting, a general approach is to learn a multi-contour map to perform varying granularities of segmentation over different regions of an instance. Specifically, starting from a seed map, which depicts the instance centroids, pixel-level region growing is carried out from seeds to push the border of an instance based on various visual cues. For example, Deep Watershed [4] performs pixel labelling by predicting the direction of a pixel which points to the nearest instance boundary to generate a 16-level instance-wise watershed energy map for segmentation. PSENet [14] learns to progressively expand the shape and size of an object instance across different scales of an image. SECB [5], TextMountain [17] and SibNet [7] first predict the seed map and then grow the seeds (instance centroids) with pixel connectivity analysis, by various neural architectures for predicting pixel offset vectors [5], centre-direction [17] and sibling relation [7]. Terrace [6] performs end-to-end learning of the multi-layer contour map with attention for instance counting and with weights for robust instance labelling.

Most of the approaches (e.g., PSENet [14], SibNet [7], and Terrace [6]) perform class-agnostic instance segmentation. In other words, the category of an instance will not be predicted. Deep watershed [4] and SECB [5], on the other hand, take the

semantic map as input and perform instance segmentation for each semantic mask. The result is highly dependent on the quality of a semantic map. As these approaches have two network branches for semantic and instance segmentations, respectively, the computational cost and memory consumption is generally higher, for example, due to training of two deconvolution layers for up-sampling of feature maps [4] and storing of full-resolution seed maps for individual semantic mask [5]. In this paper, we propose FoodMask, a simple, efficient and yet effective neural architecture that seamlessly combines counting, semantic segmentation and contour map generation for instance labelling. The training of FoodMask is less time involved for learning only the parameters of shared deconvolution layers for different tasks and applying simple operators (e.g., max and sum) for semantic map and contour map generation. During testing, FoodMask is also efficient by skipping unnecessary processing in instance labelling and concentrating only on the regions with multiple instances for segmentation. All these result in at least a double speedup in terms of frames per second than any of the existing approaches in the literature.

In the literature, as FoodMask, these are also class-specific instance segmentation methods. The examples include single-shot instance segmentation (SSAP) [33], PolarMask [34] and PolarMask++ [35]. SSAP [33] learns the instance relation of neighbouring pixels within a window size over different image scales. As the instance extraction algorithm is implemented iteratively through multiple scales of feature maps, this method is computationally expensive. PolarMask [34] and PolarMask++ [35] model an instance by a centre mass with 9-16 pixels and a fixed number of rays from the centre of mass to the object boundary. As an instance can only be depicted with a polygon composed of 72 rays (or boundary points), the segmentation is not fine-grained, which is insufficient to extract food items of arbitrary shape.

3. Food Mask

Fig. 3 shows the architecture of FoodMask, which is composed of three branches for instance counting, segmentation, and recognition. The shared backbone of these branches is a Feature Pyramid Network (FPN) [36] that processes an input image of

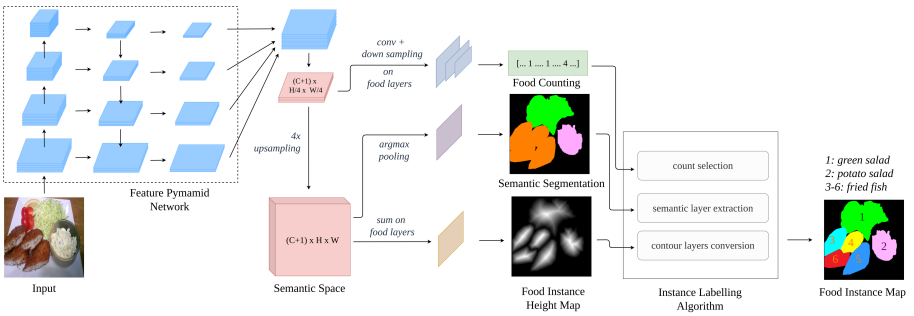

Figure 3: FoodMask: A multi-task neural architecture for instance counting, segmentation and recognition. The output is an instance map that labels the food category and instance of every pixel in an input image.

size $H \times W$ with convolutions at multiple scales of resolution. The resulting features at different scales are encoded into a $(C+1) \times H/4 \times W/4$ feature map of $C+1$ channels for semantic segmentation [37]. Each channel labels the pixels belonging to one of C food categories being considered. An additional channel is encoded for the labelling of background or non-food pixels. The feature map is leveraged by the counting task to produce a distribution of counts indicating the number of instances per category in an image. Meanwhile, deconvolution is performed to up-sample the feature map by four times via bilinear interpolation to obtain a semantic space, denoted as $\mathbb{F}$, of the original image size, i.e., $(C+1) \times H \times W$. The semantic space serves as input for the tasks of semantic and instance segmentation, which produce semantic and instance labels, respectively, for every pixel in an image.

3.1. Food Counting

Counting produces a vector $t \in \mathbb{R}^C$ to enumerate category-wise instances. We design a simple convolution network with three down-sampling stages to transform each channel of the FPN output, i.e., $(C+1) \times H/4 \times W/4$, to a feature map at 1/8 scale. Each stage consists of a 3×3 convolution, batch norm, ReLU, and $2\times$ bilinear down-sampling. The convolutional module is followed by an average pooling to output a single floating value. By repeating this process for all the C channels, the vector t of length C is formed, where each value t_c, $c \in [1, C]$, reflects the number of instances enumerated for a food category. Denote $t_{n,c}$ and $\hat{t}_{n,c}$ as the predicted and actual counts of a category c in the n^{th} image sample, we employ the mean square error loss function

to optimise the counting sub-network as following:

$$L_T = \frac{\sum_{n=1}^{N} \sum_{c=1}^{C} (t_{n,c} - \hat{t}_{n,c})^2}{NC} \tag{1}$$

where N is the total number of training samples.

3.2. Semantic Segmentation

We follow [37] which proposes to use FPN as a backbone and learn a semantic space $\mathbb{F}$ that encodes the category distribution at every pixel. Specifically, denote $F \in \mathbb{R}^{C+1}$ as a vector positioning at pixel (H_F, W_F) along the channel dimension of $\mathbb{F}$. The vector F captures the category distribution of a pixel, where its element F_c for food category c has the following three properties:

1. The probability for a category c^*:

$$P_{c^*} = \frac{e^{F_{c^*}}}{\sum_{c=0}^{C} e^{F_c}} \tag{2}$$

2. The classification decision for c^* is subjected to:

$$c^* = \arg\max_{c \in [0,C]} F_c \tag{3}$$

3. The semantic loss function is a negative log-likelihood on the ground-truth category $\hat{c}$:

$$L_F = -log(P_{\hat{c}}) \tag{4}$$

Based on Equation 4, cross-entropy loss is measured by summing the semantic losses over $H \times W$ pixels of the N training samples, as follows:

$$L_S = \frac{\sum_{n}^{N} \sum_{F}^{\mathbb{F}} L_{n,F}}{NHW} \tag{5}$$

It is worth noting that the semantic vector F enjoys the characteristic of scale invariance. Specifically, the aforementioned three properties are not affected by the

global scaling of value across all vector elements F_c. For example, by adding a constant value x, the probability distribution of a category remains the same:

$$P_{c_x^*} = \frac{e^{x+F_{c^*}}}{\sum_{c=0}^{C} e^{x+F_c}} = \frac{e^x e^{F_{c^*}}}{\sum_{c=0}^{C} e^x e^{F_c}} = \frac{e^{F_{c^*}}}{\sum_{c=0}^{C} e^{F_c}} = P_{c^*} \tag{6}$$

This scale-invariant characteristic will be leveraged for the estimation of the instance height map, which will be further elaborated in the next section.

3.3. Food Instance Height Map

Different from semantic segmentation, the instance height map aims to label individual food instances. With reference to Fig. 3, the pixels in the four pieces of fried fish are separately labelled. Separating food instances of the same food category, nevertheless, is challenging as these instances share similar visual appearance. Similar to the Terrace model [6], we model the pixels in an instance based on their spatial distances to the instance border. Specifically, the pixels centrally located in an instance will have higher values while the pixels closer to the border will have lower values. This forms a "height" map intuitively depicting different levels of challenge in labelling, where pixels closer to the border will have higher uncertainty in labelling decisions. As shown in Fig. 3, the height map visualizes peaks corresponding to centroid regions of different instances, which are served as "seeds" to be leveraged by the labelling algorithm (Section 3.4) for region growing of their respective instances.

Denote $\hat{Y} \in \mathbb{R}^{H \times W}$ as the ground-truth food instance height map which will be constructed as follows. On every pixel of an image, we assign a distance value, denoted as $D_{h,w}$, indicating the minimum number of pixels being traversed from the current pixel position (h, w) to a pixel belonging to background or a different instance of food. We consider the radius of an instance, $R_{\mathcal{I}}$, as the maximum distance value among all its pixels, as follows:

$$R_{\mathcal{I}} = \max_{(h,w) \in \mathcal{I}} D_{h,w} \tag{7}$$

The ground-truth height of a pixel $\hat{Y}_{h,w}$ is a normalized distance value in the range of

[0,1] by:

$$\hat{Y}_{h,w} = \frac{D_{h,w}}{R_{\mathcal{I}}} \tag{8}$$

By doing this, all instances have similar values of height at their peaks that facilitate the subsequence step of instance labelling.

Instead of constructing a new deconvolution network, instance labelling shares the semantic space, $\mathbb{F}$, with the semantic segmentation module for the learning of height map. A height map is derived by simply pooling the feature map along the channel direction, as follows:

$$Y_{h,w} = \sum_{c=1}^{C} \mathbb{F}_{h,w,c} \tag{9}$$

An advantage of this design is computational efficiency, where the formation of a height map does not require disparate learning of deconvolution layers from the semantic segmentation module. Capitalizing on the scale invariance of semantic space, as stated in Equation 6, learning to scale the semantic vector, $\mathbb{F}_{h,w}$, to reflect the height of a pixel at location (h, w) will not alter the original result of semantic segmentation. In other words, the network is trained to scale $\mathbb{F}_{h,w}$, such that the sum of all elements in $\mathbb{F}_{h,w}$ reflects the height of a pixel, while $\mathbb{F}_{h,w,c}$ still captures the probability belonging to a category after normalization (Equation 2). To this end, the pixel-wise mean square error loss function is employed to penalise the difference between the predicted and ground-truth height maps:

$$L_Y = \frac{\sum_n^N \sum_h^H \sum_w^W (\hat{Y}_{n,h,w} - Y_{n,h,w})^2}{NHW} \tag{10}$$

Overall, the loss function of the multi-task FoodMask is formulated as:

$$L = \lambda_T L_T + \lambda_S L_S + \lambda_Y L_Y \tag{11}$$

where λ_T, λ_S and λ_Y are trade-off hyper-parameters.

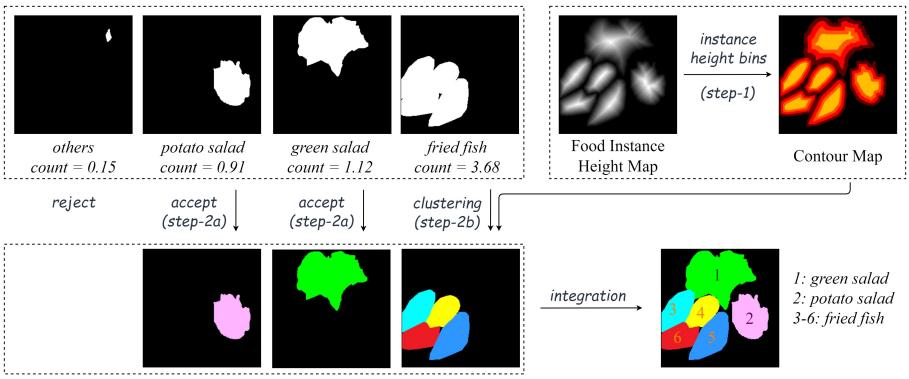

Figure 4: An example illustrating the instance labelling algorithm. The top-left block shows the semantic maps of individual categories with their predicted counts. Together with a contour map, which is converted from the height map, the semantic maps of these categories are either "rejected", "accepted", or "clustered" to produce an instance map. In this example, there are four pieces of fried fish being labelled as instances. Please refer to Algorithm 1 for the details of the corresponding steps labelled in this figure.

3.4. Instance Extraction and Labelling

There are various ways to leverage the per-category counting distribution, per-pixel probability distribution of categories and the image-level height map. For speed efficiency, we consider a pruning strategy which rapidly scrapes away noisy food instances while performing on-demand instance labelling. Specifically, we use per-category count distribution as a prior to determining whether to prune, keep or decompose the regions of pixels with the same category in a semantic map into separate instances. For a category with its count equal to zero, no instance will be generated while the corresponding region will be labelled as "background" in the instance map. If the value of count is equal to one, no instance labelling will be performed for that category, and instead, the entire region of the category in the semantic map will be copied over for the instance map. Only when the value of count is equal to or larger than two, the corresponding region in a semantic map will be decomposed into disjoint regions of different labels where each corresponds to a standalone instance. This strategy is more efficient than other options, such as performing instance labelling for every food category, as will be demonstrated in the experiments.

Algorithm 1 presents the instance labelling algorithm. In Step-1, the height map is converted into a contour map as in other proposal-free algorithms [4, 6, 14]. The

Algorithm 1 Instance labelling algorithm

Require: binarized version of semantic map: S, vector of per-category count: t, normalized version of height map: $\overline{Y}$, number of food categories: C

Ensure: Multi-instance map: $\mathbb{I}$

function INSTANCELABELLING(S, t, $\overline{Y}$, C)

Step-1: Convert $\overline{Y}$ into a contour map $\mathbb{Y}$ of different levels from 1 to $K+1$:

- $\mathbb{Y} \leftarrow K+1$ where $\frac{1}{2} < \overline{Y} \leq 1$
- $\mathbb{Y} \leftarrow k$ where $\frac{k-1}{2K} \leq \overline{Y} \leq \frac{k}{2K}$ and $k \leq K$

Step-2: Iterate over every food category $c \in [1, C]$:

- Step-2a: If $t_c \geq 0.5$: $\mathbb{I}_c \leftarrow S_c$
- Step-2b: If $t_c \geq 1.5$: construct a contour map Υ specifically for category c:

$$\Upsilon_{h,w} = \begin{cases} \mathbb{Y}_{h,w} & \text{if } S_{h,w,c} = 1 \\ 0 & \text{if } S_{h,w,c} = 0 \end{cases} \tag{12}$$

Perform clustering algorithm [6] on Υ

Map the resulting instance map on $\mathbb{I}_c$

return $\mathbb{I}$

end function

contour map facilitates region growth by dividing a height map into different levels of amplitude (or watershed), such that a region can be expanded efficiently in a level-wise rather than pixel-wise manner. The associated issue, nevertheless, is the number of levels required for region growing, which is a trade-off between the accurate localization of peaks for growing and the precise probing of instance boundaries, as discussed in [6]. For example, having an excessive number of levels is potentially helpful in depicting complex shapes but unnecessarily complicates the learning of a contour map and the post-processing step in instance labelling. In the implementation, the number of contour levels is empirically set as $K+1$. The highest $(K+1)^{th}$ level contour covers the pixels whose values are in the range of $(\frac{1}{2}, 1]$. The remaining pixels are uniformly grouped into different levels, with the k^{th} level contour covering the pixels with values $(\frac{k-1}{2K}, \frac{k}{2K}]$ and the lowest level contour includes all zero-height pixels. Unlike the equal-thickness contours used in other methods [4, 6, 14], preserving a larger innermost contour (50% in our case) speeds up labelling the next step. In Step-2, instance labelling is carried out by referring to the per-category count distribution. As count is a continuous number, a rounding operation is performed to gate the decision as shown in Algorithm 1. Only food categories with a count equal to or greater than two will be further processed by

Table 1: Statistics on three food datasets: Mixed Dishes [9], UECFood [10], and FoodSeg103 [11].

	Mixed Dishes [9]	UECFood [10]	FoodSeg103 [11]
Dish category	103	102	63
Instance	31,556	22,224	24,918
Training images	7,416	9,000	4,983
Testing images	1,838	1,000	2,135

Step-2b. Specifically, the clustering of pixels will be performed for instance labelling in Step-2b. While there are various algorithms that can be directly applied for clustering, such as energy cut [4] and scale expansion [14], the algorithm used in Terrace model [6] is adopted due to its efficiency and superior performance in terms of panoptic quality [18] on food datasets.

4. Experimental Setup

4.1. Datasets

The experiments are conducted on three food datasets Mixed Dishes [9], UECFood-PixComp (UECFood) [10] and FoodSeg103 [11] with statistics listed on Table 1.

Mixed Dishes images are collected from six canteens in a university. Each image contains a mixture of several dishes densely put together on a plate. Excluding "rice", this dataset has 102 major categories and 81 minor categories. A major category has more than 50 food instances in the training examples. Meanwhile, the minor categories are all grouped into a category called "others" due to insufficient training instances. With the assistance of a graphical annotation tool namely VIA [38], we manage to sketch a polygon line tightly surrounding the instance boundary.

UECFood dataset collects 10,000 images of Japanese food, covering 102 categories. Most images depict a daily meal which consists of one or several dishes presented on a bowl, plate, cup or canteen tray. Being composed of various ingredients, a dish in UECFood is rich in visual appearance. As UECFood provides only semantic maps, we also provide polygon labels to generate instance maps for all the images.

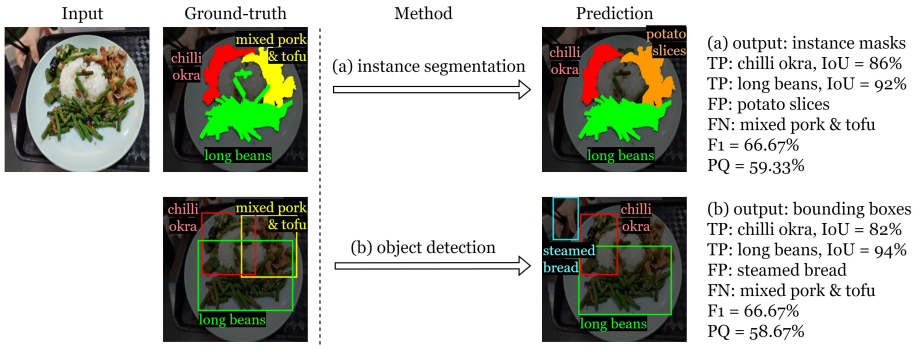

Figure 5: An example illustrates the measurement of performances for instance segmentation and object detection, respectively, on the region and box levels, using F1 and PQ (panoptic quality).

FoodSeg103 dataset includes around 7,000 images covering 103 ingredient categories. The photos are mostly close-up shots to better capture fine-grained ingredient details. Compared to Mixed Dish and UECFood, this dataset covers a large range of cuisines, such as Chinese, Indian and Western dishes, and a variety of meal types, such as breakfast, snack, drink, fruit and nut. However, the dataset is constructed for ingredient-based semantic segmentation. Ingredients such as "almond" (in "mixed nuts" dish) and "shiitake" (in "mushroom" dish) are overly fine-grained to be treated as dish categories and further segmented into instances. Hence, we merge the ingredient labels into 63 dish categories and provide polygons to label the instances of these categories. To this end, all images in the three datasets are prepared with polygon-based instance masks where each mask has both the instance and semantic labels.

4.2. Evaluation Metrics

Instance level. We evaluate the performance of instance recognition and counting based on the number of true, false and missing predicted instances. For the object-proposal techniques, a bounding box is considered as true positive (TP) if it is correctly classified and its Intersection over Union (IoU) with the ground-truth box is larger than 0.5. Otherwise, a detected bounding box is treated as a false positive (FP) while an undetected ground-truth box is regarded as a false negative (FN). Similar measures are used for the proposal-free approaches except that IoU is computed based on the generated and ground-truth instance maps. We use the same measure for the regression-

based counting, except that the predicted counts are rounded and then compared against the ground-truth per-category counts. Note that IoU is not measured in this case since only image-level counts are predicted by the regression approach. Finally, F1 evaluates the overall performance:

$$\text{F1} = \frac{1}{N}\sum_{n=1}^{N}\frac{|TP|_n}{|TP|_n + \frac{1}{2}|FP|_n + \frac{1}{2}|FN|_n} \tag{13}$$

where N is the total number of testing images, $|TP|_n$, $|FP|_n$ and $|FN|_n$ are respectively the number of true positives, false positives, and false negatives in the n^{th} image.

Pixel level. The instance-level segmentation is measured by Panoptic Quality (PQ) [18], which is similar to Equation 13 except that the numerator is evaluated with IoU. Formally, the PQ of an image is measured as:

$$\text{PQ} = \frac{\sum_{(p,g)\in TP}\text{IoU}(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{14}$$

where p and q denote the predicted true positive instance and the corresponding ground-truth instance, respectively. We also apply PQ for measuring the goodness-of-fit between predicted and ground-truth bounding boxes. It is expected that the PQ score measured on the bounding box level could not precisely reflect the goodness in capturing the irregular shape of an instance. Figure 5 depicts an example showing the F1 and PQ scores measured on two different units, i.e., region and box, predicted by FoodMask and CR-Net [24], respectively. While both the instance segmentation and object detection approaches could show a similar level of performance in terms of F1 and PQ scores, it should be noted that pixel-level labelling is more accurate, for example, in delineating the elongated shape of "chilli okra". A bounding box could include as much as more than 50% of pixels irrelevant to a food category or instance.

4.3. Network Training

We employed ResNet-50 [39] pre-trained on ImageNet [40] as the backbone of FoodMask. During training, we empirically set the hyper parameters $\lambda_T = \lambda_S = \lambda_Y = 1$ (Equation 11). The model was trained on a batch size of 16 on one GPU of GeForce

GTX 1080 Ti. The learning rate started from 2×10^{-4} and reduced 10% after every 32 epochs. The loss function converged after 512 epochs. In addition, we set the number of levels in a contour map as $(K + 1) = 9$.

5. Experimental Results

5.1. Ablation Studies

5.1.1. Various options for FoodMask implementation

The three different branches in FoodMask allow three different ways of deriving instance masks. The ablation studies aim to investigate the effectiveness of the pruning strategy adopted by FoodMask. We investigate three different options, as follows:

- Option A: Similar to Algorithm 1, except that the count distribution is not leveraged as a prior in Step-2 to selectively perform instance labelling. In other words, instance labelling is performed for all the food categories in Step-2b. This option is similar to the strategy used in Deep Watershed [4] and SECB [5].

- Option B: Similar to Option A, except that the instance labelling algorithm [6] is performed directly on the contour map produced in Step-1, instead of on the semantic-projected contour map generated by Equation 12. In other words, the Step-2 of Algorithm 1 is not performed. The semantic label of an instance is then determined by the majority voting of pixels. More specifically, each pixel votes for the food category that it belongs to based on the semantic map, and the entire region is then labelled to the majority vote.

- Option C is the pruning strategy used by FoodMask, as detailed in Algorithm 1.

Table 2 compares the performances of FoodMask variants under different implementation options. We also include the counting-only performance, i.e., the result of per-category counting based on the counting branch of FoodMask as a reference for comparison. Option-A tends to produce more false instances when the semantic maps are error-prone, resulting in an instance map with small erroneous regions. Option-B is effective in getting rid of these regions by directly performing pixel-level clustering on

Table 2: Ablation studies comparing the performances across different options of implementation for FoodMask on Mixed Dishes (MD), UEC and FoodSeg103 (FS) datasets. The speed performance (Frames Per Second) is also reported for three options of implementation.

Option	Instance level (F1 (%))			Pixel level (PQ (%))			FPS
	MD	UEC	FS	MD	UEC	FS	
FoodMask A	85.85	72.28	58.24	63.89	59.19	48.70	16.54
FoodMask B	86.34	72.85	60.03	66.06	**63.49**	**54.78**	18.32
FoodMask C	**87.02**	**72.91**	**60.81**	**66.99**	61.35	52.38	**38.85**
Counting-only	86.01	70.91	58.59	NA	NA	NA	-

the contour maps rather than the individual masks of a semantic map. Consequently, option-B exhibits higher F1 and PQ performances consistently across all three datasets. By the pruning strategy, option-C also manages to eliminate most of the small erroneous regions on a semantic map, with a higher F1 performance than option-B. However, due to the simple strategy of labelling these regions as background in the instance map, the IoU between the predicted and ground-truth instances also become lower. Compared to option-B, this results in lower PQ performance on the UEC and FoodSeg datasets.

In general, performing pixel-level clustering directly and globally on the contour maps yields better-quality of instance maps in terms of PQ. Nevertheless, without per-category counting distribution and semantic map as the priors, more missed and false instances are produced (i.e., lower F1) by option-B than option-C. Compared to counting prediction and semantic map generation, instance map formation which involves pixel-level clustering is more computationally expensive. Option-C has the advantage of skipping unnecessary clustering on regions where the predicted counts are insignificant. Despite its simplicity, this pruning strategy speeds up option-C by more than double by processing 38.85 images per second, compared to 16.54 and 18.32 images by options A and B, respectively. Although imperfect, the counting branch still performs reasonably well with F1 performance close to or even better than that of option-A in some datasets, which provides reliable prior for the pruning strategy.

Figure 6 shows three examples contrasting these options. In Figure 6a, the predicted counts and semantic masks are near-accurate, and all three options yield almost equally

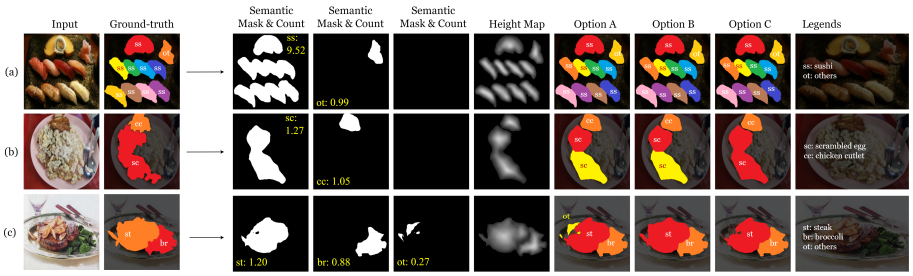

Figure 6: Three examples illustrating the intermediate results of FoodMask, including counting prediction, semantic segmentation and instance map generation.

Table 3: The effect of varying the trade-off parameters in the FoodMask loss function (Equation 11) to the instance segmentation performance (PQ (%)) on three food datasets.

Loss weights	Mixed Dishes	UEC	FoodSeg103
$\lambda_T = 10$	66.12	58.19	50.59
$\lambda_S = 10$	65.49	55.75	49.02
$\lambda_Y = 10$	66.98	61.26	52.17
standard	**66.99**	**61.35**	**52.38**

good performance. Similarly, in Figure 6b, the counts and masks are almost perfect, nevertheless, the instance labelling algorithm generates an additional instance due to the fuzziness texture of scrambled egg. By leveraging the prior, option-C indeed avoids performing pixel-level clustering and results in better performance than other implementation options. In Figure 6c, a semantic mask is falsely predicted, which negatively impacts the instance map generated by option-A. By the insignificant count predicted on the wrongly labelled semantic mask, option-C also avoids performing clustering on the masked region. Nevertheless, by labelling the pixels on the mask as the background, the produced instance map has lower PQ than option-B, which performs clustering globally on the counter map of the entire image to produce a better quality instance map.

5.1.2. Weight settings for loss function

In the experiment, the three parameters of Equation 11 are set to be equal value, i.e., $\lambda_T = \lambda_S = \lambda_Y = 1$. This setting is considered "standard" for equally weighting the importance of counting, semantic, and height map generation. To study the impact of

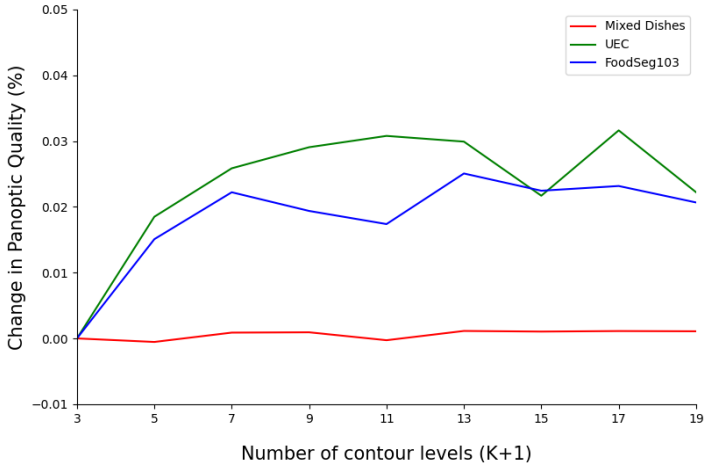

Figure 7: The change of PQ scores compared to $K + 1 = 3$ over different number of contour levels in Algorithm 1. Note that the origin is set to (3, -0.01) for the ease of visualization.

parameter weighting, we contrast this standard setting by assigning a higher weight to one of the parameters and repeating the experiment three times. The results are shown in Table 3. Basically, overemphasizing any of the three components of counting, semantic segmentation, and instance height map generation, the performance will be negatively impacted. For PQ, which assesses pixel-level segmentation quality, overweighting λ_T (counting) or λ_S (semantic segmentation) will likely degrade the performance as indicated by the results on the UEC and FoodSeg103 datasets.

5.1.3. Number of contour levels

In Algorithm 1, the number of contour levels, $K + 1$, is set to be the value of 9 to compromise the tradeoff between modelling efficiency and localization precision. Fig. 7 shows the effect of this parameter across different ranges of value. In general, the change of $K + 1$ value results in 0.03% of performance fluctuation. Practically, setting the range of values within $[5, 9]$ is a good compromise between speed and accuracy although $K + 1 = 17$ appears to be an optimal value in the experiment.

5.1.4. Network backbone

We contrast the performances of backbone models pre-trained on the general-purpose dataset (ImageNet [40]) and the food-specific datasets (Food-101 [41], Food2K

Table 4: The performances of FoodMask with ResNet50 [39] backbone pretrained on different datasets.

Option	Instance level (F1 (%))			Pixel level (PQ (%))		
	Mixed Dishes	UEC	FoodSeg103	Mixed Dishes	UEC	FoodSeg103
Scratch	76.43	41.01	38.07	57.8	32.42	30.30
ImageNet [40]	87.02	72.91	60.81	66.99	61.35	52.38
Food-101 [41]	87.27	72.85	61.37	67.01	61.45	52.43
Food2K [42]	**87.57**	**73.09**	**61.40**	**67.44**	**61.51**	**52.91**

[42]). ImageNet [40] comprises a vast collection of over 100,000 object categories, with an average of 1,000 labelled images per category. In contrast, Food-101 consists of 101 food categories, each accompanied by 1,000 images. Food2K is a large dataset with 2,000 categories and over 1 million images. Table 4 presents the performances of FoodMask using backbones pre-trained on these datasets. Note that these pre-trained models are further fine-tuned using our food datasets. In Table 4, "scratch" refers to the model trained using one of our datasets from scratch without using a pretrained model. The results show that using pre-trained models can improve significantly a model trained from scratch. Furthermore, the models pre-trained on food-specific datasets exhibit better performance than the model pre-trained with the general-purpose dataset.

5.2. Performance Comparison

We compare FoodMask (option C) to four major branches of approaches, i.e., regression counting (without localization), object detection (with bounding box), proposal-based and proposal-free instance segmentation. For counting, we compare subitising [43], single-task [20] and multi-task counting. Subitising [43] practices a divide and conquer strategy which learns to generate a grid map where each grid cell predicts a fraction of instance for every food category. A more straightforward approach is by training a regression-based CNN for the single-task of counting [20]. The CNN can be extended for multiple-task counting by having an additional branch for semantic or instance segmentation. In the experiment, we implement the single-task counting using the same architecture as the multi-label recognition model in [20]. The multi-task counting is implemented similarly based on the two-branch architecture for counting

and semantic segmentations in FoodMask. Object detection techniques (CR-Net [9], FCOS [16]) localize instances with bounding boxes. Using Faster R-CNN [15] which is a two-stage detector, CR-Net [9] models the co-occurrence of dishes on a plate with a context graph for category prediction. FCOS [16] is a one-stage object detector, aiming to perform faster by the removal of Regional Proposal Network in [15] and better by learning a centre-ness mask to suppress low-quality detections. Proposal-based instance segmentation methods (Mask R-CNN [1], CenterMask [3], Yolact++ [12], YOLOv8 [44], HTC [2], DSC [8]) extend object detection to perform instance segmentation. Mask R-CNN [1] and CenterMask [3] add segmentation sub-networks to Faster R-CNN [15] and FCOS [16] respectively. HTC [2] and DSC [8] further introduce cascade modules for progressive refinement of instance segmentation. Yolact++ [12] proposes a fast non-maximum suppression algorithm, along with the mask re-scoring and deformable convolution networks, for real-time instance segmentation. Mask2Former [27] is a universal architecture to achieve SOTA performance in major image segmentation tasks (panoptic, instance and semantic) on four popular datasets. For proposal-free approaches, the comparison is made against Terrace [6] and SibNet [7]. Nevertheless, as category prediction is not considered, we re-implement both models with a multi-tasking architecture. Specifically, a second branch for semantic segmentation is added to the original architecture. An instance map is generated by pixel-wise majority voting of semantic category for the classification of an instance mask, i.e., the option-B implementation discussed in the previous section. We name these approaches Terrace++ and SibNet++, respectively.

Table 5 lists the result comparison. Among the four branches of approaches, proposal-free approaches consistently outperform all other methods across three different datasets. Despite the superior performances reported by three other branches of approaches on the general object datasets, the result of locating food instances densely placed on a plate is generally imperfect. Their performances on food datasets are similar but with some variations depending on evaluation measures. For example, Yolact++ and DSC perform better at instance-level (F1) and pixel-level evaluation (PQ), respectively. Mask2Former [27] has the best F1 and PQ measures among other proposal-based approaches in three food datasets, and it also performs better than Deep Watershed. And

Table 5: Performance comparison across four major branches of approaches of instance-level counting and pixel-level segmentation on Mixed Dishes (MD), UEC and FoodSeg103 (FS) datasets.

Method		Instance level (F1 (%))			Pixel level (PQ (%))		
		MD	UEC	FS	MD	UEC	FS
Regression counting	Single-task [20]	83.36	63.43	54.89	NA	NA	NA
	Subitizing [43]	80.11	58.36	53.49	NA	NA	NA
	Multi-task	86.01	70.91	58.59	NA	NA	NA
Object detection	CR-Net [24]	81.97	64.24	56.99	61.93	49.95	43.28
	FCOS [16]	79.61	66.30	51.40	59.60	52.64	41.30
Proposal-based instance segmentation	Mask R-CNN [1]	80.41	65.28	53.18	60.30	52.35	43.54
	CenterMask [3]	80.78	68.42	53.83	60.87	54.22	44.67
	Yolact++ [12]	81.86	70.45	57.25	58.63	55.60	44.16
	HTC [2]	80.92	63.28	53.38	61.22	53.22	44.11
	DSC [8]	81.15	65.06	54.56	61.77	53.41	46.94
	YOLOv8 [44]	82.06	61.37	49.78	63.04	51.02	43.65
	Mask2Former [27]	83.65	70.54	60.03	64.06	59.61	47.77
	FastInst [29]	83.57	69.05	59.17	64.07	58.26	48.66
Proposal-free instance segmentation	Watershed [4]	82.80	70.35	56.26	61.97	56.66	46.35
	PolarMask [34]	71.53	59.45	40.89	51.05	40.27	30.17
	Terrace++ [6]	86.04	71.91	60.13	65.70	62.84	54.32
	SibNet++ [7]	85.91	72.06	59.38	66.60	**63.86**	**54.95**
	FoodMask	**87.02**	**72.91**	**60.81**	**66.99**	61.35	52.38

FastInst [29], a lightweight version of Mask2Former, has also achieved comparable measures. Note that multi-task counting is indeed a strong baseline although it is not capable of performing instance localization. Among the proposal-free approaches, except Deep Watershed, FoodMask performs particularly well on the Mixed Dishes dataset in both F1 and PQ measures. Compared to Terrace++ and SibNet++, FoodMask generally shows better F1 but lower PQ performance. This is mainly due to the pruning strategy that selectively performs instance labelling on some regions of an image. As discussed in the ablation studies, the option-B implementation as adopted by two other models generates higher quality instance maps but with more false and missed instances compared to FoodMask.

Fig. 8 shows examples contrasting the results of different approaches. The example

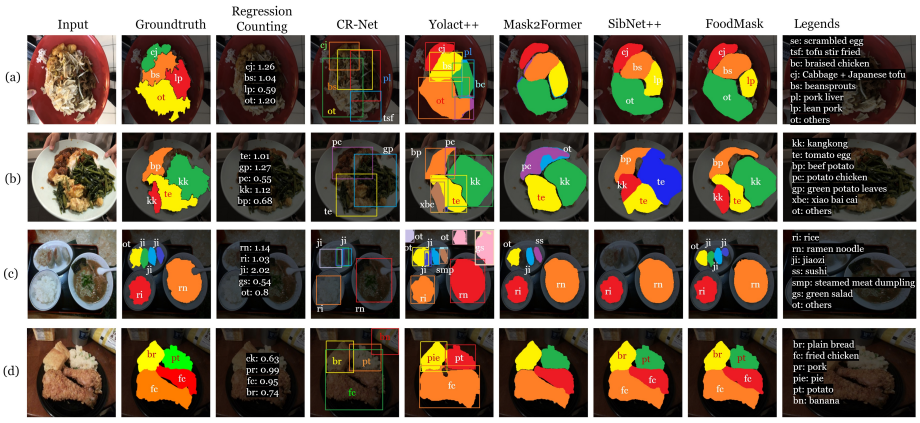

Figure 8: Instance counting and segmentation results on various datasets.

in Fig. 8a shows a typical example with multiple dishes being fuzzily put on a plate. While all the methods can segment the dishes out quite correctly, only FoodMask and SibNet++ can predict all the categories correctly. In Fig. 8b, the SibNet++ correctly spots two instances at the area of "kang kong" and "tomato egg" (marked in yeallow and red colours), but the boundary detection is not as good as FoodMask. As seen in Fig. 8c, FoodMask shows the ability of counting both small and large instances where SibNet fails to detect three smaller objects of "jiaozi". Fig. 8d shows an example where both object detection and proposal-based approaches fail to separate the two instances of "chicken leg" on a plate. By pixel-level clustering, proposal-free approaches are more resilience to the situations when the instances of a dish category occludes or are placed near to each other.

5.3. Speed Efficiency

We compare the speed of FoodMask to other approaches, as listed in Table 6. For simplicity, we group these approaches as "mask-based" and "pixel clustering". The former encloses an instance in a mask or bounding box, while the latter segments instances by clustering of pixels. In general, the mask-based approaches are computationally slower due to additional time incurred in predicting masks. Even with careful optimization, such as Yolact++ [12], the speed can only reach up to about 18 frames per second (FPS). Each additional step, such as the exploitation of cascade modules by

Table 6: The average speed performance of different approaches on food datasets.

Method		Remark	FPS
Object proposal	CR-Net [24]		6.52
	FCOS [16]		7.48
	Mask R-CNN [1]		5.05
	CenterMask [3]		5.58
	Yolact++ [12]	fast NMS	18.19
	HTC [2]	cascade	3.04
	DSC [8]	cascade	2.74
	YOLOv8 [44]		31.78
	Mask2Former [27]		7.47
	FastInst [29]		30.03
Pixel clustering	Watershed [4]		14.37
	PolarMask [34]		8.48
	Terrace [6]++ B		18.81
	SibNet [7]++ B		19.22
	FoodMask		**38.85**

HTC [2] and DSC [8] for progressive refinement, will result in even slower processing time. Without mask generation, the speed of pixel clustering approaches are at least as fast as Yolact++ [12]. FastInst [29] achieves very impressive speed at about 30 FPS. This due to the lightweight pixel decoder and instance activation-guided queries which enable it to perform comparable segmentation performance on one layer transformer decoder only. With an additional pruning mechanism, FoodMask can easily push the speed to more than 38 FPS. With a reliable food counting branch in FoodMask, almost 95% of dishes in Mixed Dishes and 80% of dishes in UEC and FoodSeg103 are rapidly skipped by FoodMask to avoid unnecessary pixel clustering.

6. Conclusion

We have presented FoodMask, a multi-task network architecture, for real-time recognition of food instances. Experimental studies on Mixed Dishes, UECFoodPixComp and FoodSeg103, verify the recognition effectiveness of FoodMask over other branches of approaches, including counting-based object detection and instance segmentation methods. FoodMask attains not only high recognition accuracy at both instance and

pixel levels but also impressive real-time processing speed. This paper, additionally, also contributes instance-level labels on Mixed Dishes, UECFoodPixComp and FoodSeg103 for future research on food recognition.

FoodMask is designed to be a proposal-free class-specific instance segmentation model functioning on a close set of food categories known during training time. One inherent issue is that counting and semantic segmentation are not class-agnostic, which potentially limits the extension of FoodMask for open-vocabulary food segmentation and recognition. Extension of FoodMask for open-vocabulary settings, such as by leveraging Large Langugage Model (LLM), will be our future direction.

7. Acknowledgment

This research/project is supported by the City University of Hong Kong (project No. 9678180), and by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Proposal ID: T2EP20222-0046). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

References

[1] K. He, G. Gkioxari, P. Dollár, R. B. Girshick, Mask R-CNN, in: IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. doi:10.1109/ICCV.2017.322.

[2] K. Chen, W. Ouyang, C. C. Loy, D. Lin, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, Hybrid task cascade for instance segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4969–4978. doi:10.1109/CVPR.2019.00511.

[3] Y. Lee, J. Park, Centermask: Real-time anchor-free instance segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. doi:10.1109/CVPR42600.2020.01392.

[4] M. Bai, R. Urtasun, Deep watershed transform for instance segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2858–2866. doi:10.1109/CVPR.2017.305.

[5] D. Neven, B. D. Brabandere, M. Proesmans, L. V. Gool, Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. doi:10.1109/CVPR.2017.471.

[6] H.-T. Nguyen, C.-W. Ngo, Terrace-based food counting and segmentation, in: The Association for the Advancement of Artificial Intelligence (AAAI), 2021, pp. 2364–2372. doi:10.1609/aaai.v35i3.16337.

[7] H.-T. Nguyen, C.-W. Ngo, W.-K. Chan, Sibnet: Food instance counting and segmentation, in: Pattern Recognition, Vol. 124, 2022, p. 108470. doi:10.1016/j.patcog.2021.108470.

[8] H. Ding, S. Qiao, A. Yuille, W. Shen, Deeply shape-guided cascade for instance segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 8274–8284. doi:10.1109/CVPR46437.2021.00818.

[9] L. Deng, J. Chen, Q. Sun, X. He, S. Tang, Z. Ming, Y. Zhang, T. S. Chua, Mixed-dish recognition with contextual relation networks, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, p. 112–120. doi:10.1145/3343031.3351147.

[10] K. Okamoto, K. Yanai, UEC-FoodPIX Complete: A large-scale food image segmentation dataset, in: Proc. of ICPR Workshop on Multimedia Assisted Dietary Management(MADiMa), 2021. doi:10.1007/978-3-030-68821-9_51.

[11] X. Wu, X. Fu, Y. Liu, E.-P. Lim, S. Hoi, Q. Sun, A large-scale benchmark for food image segmentation, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 506–515. doi:10.1145/3474085.3475201.

[12] D. Bolya, C. Zhou, F. Xiao, Y. J. Lee, YOLACT++ better real-time instance segmentation, in: IEEE Transactions on Pattern Analysis & Machine Intelligence, 2022, pp. 1108–1121. doi:10.1109/TPAMI.2020.3014297.

[13] L. Yuliang, J. Lianwen, Z. Shuaitao, Z. Sheng, Detecting curve text in the wild: New dataset and new solution, arXiv: Computer Vision and Pattern Recognition (2017).

[14] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, S. Shao, Shape robust text detection with progressive scale expansion network, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9328–9337. doi:10.1109/CVPR.2019.00956.

[15] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015. doi:10.1109/TPAMI.2016.2577031.

[16] Z. Tian, C. Shen, H. Chen, T. He, FCOS: fully convolutional one-stage object detection, in: IEEE International Conference on Computer Vision (ICCV), 2019. doi:10.1109/ICCV.2019.00972.

[17] Y. Zhu, J. Du, Textmountain: Accurate scene text detection via instance segmentation, in: Pattern Recognition, 2021, p. 107336. doi:10.1016/j.patcog.2020.107336.

[18] A. Kirillov, K. He, R. Girshick, C. Rother, P. Dollar, Panoptic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. doi:10.1109/CVPR.2019.00963.

[19] A. Ferrari, S. Lombardi, A. Signoroni, Bacterial colony counting with convolutional neural networks in digital microbiology imaging, in: Pattern Recognition, 2017, pp. 629–640. doi:10.1016/j.patcog.2016.07.016.

[20] Y. Wang, J.-J. Chen, C.-W. Ngo, T.-S. Chua, W. Zuo, Z. Ming, Mixed dish recognition through multi-label learning, in: Proceedings of the 11th Workshop on

Multimedia for Cooking and Eating Activities, 2019. doi:10.1145/3326458.3326929.

[21] J. Lei, J. Qiu, F. P.-W. Lo, B. Lo, Assessing individual dietary intake in food sharing scenarios with food and human pose detection, in: Pattern Recognition. ICPR International Workshops and Challenges, 2021, pp. 549–557. doi:10.1007/978-3-030-68821-9_45.

[22] D. Liang, W. Xu, X. Bai, An end-to-end transformer model for crowd localization, in: European Conference on Computer Vision (ECCV), Springer-Verlag, Berlin, Heidelberg, 2022. doi:10.1007/978-3-031-19769-7_3.

[23] G. Sun, Z. An, Y. Liu, C. Liu, C. Sakaridis, D.-P. Fan, L. Van Gool, Indiscernible object counting in underwater scenes, in: Proceedings of the IEEE/CVF International Conference on Computer Vision and Patern Recognition (CVPR), 2023.

[24] L. deng, J. Chen, C.-W. Ngo, Q. Sun, S. Tang, Y. Zhang, T.-S. Chua, Mixed dish recognition with contextual relation and domain alignment, IEEE Transactions on Multimedia 24 (2022) 2034–2045. doi:10.1109/TMM.2021.3075037.

[25] K. Lv, Y. Zhang, Y. Yu, H. Wang, L. Li, H. Jiang, C. Dai, Contour deformation network for instance segmentation, Pattern Recognition Letters 159 (05 2022). doi:10.1016/j.patrec.2022.05.025.

[26] R. Zhang, T. Kong, X. Wang, M. You, Mask encoding: A general instance mask representation for object segmentation, Pattern Recognition 124 (2022) 108505. doi:10.1016/j.patcog.2021.108505.

[27] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, R. Girdhar, Masked-attention mask transformer for universal image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 1280–1289. doi:10.1109/CVPR52688.2022.00135.

[28] B. Cheng, A. Schwing, A. Kirillov, Per-pixel classification is not all you need for semantic segmentation, Advances in Neural Information Processing Systems (2021) 17864–17875.

[29] J. He, P. Li, Y. Geng, X. Xie, Fastinst: A simple query-based model for real-time instance segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2023, p. 23663–23672.

[30] Y. Ou, Y. Rui, L. Ma, Y. Liu, J. Yan, S. Xu, C. Wang, X. Li, Uniinst: Unique representation for end-to-end instance segmentation, Neurocomputing 514 (09 2022). doi:10.1016/j.neucom.2022.09.112.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision (ECCV), 2014. doi:10.1007/978-3-319-10602-1_48.

[32] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016. doi:10.1109/CVPR.2016.350.

[33] N. Gao, Y. Shan, Y. Wang, X. Zhao, K. Huang, Ssap: Single-shot instance segmentation with affinity pyramid, IEEE Transactions on Circuits and Systems for Video Technology 31 (2) (2021) 661–673. doi:10.1109/TCSVT.2020.2985420.

[34] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, P. Luo, Polarmask: Single shot instance segmentation with polar representation, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 12190–12199. doi:10.1109/CVPR42600.2020.01221.

[35] E. Xie, W. Wang, M. Ding, R. Zhang, P. Luo, PolarMask++: Enhanced polar representation for single-shot instance segmentation and beyond, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (09) (2022) 5385–5400. doi:10.1109/TPAMI.2021.3080324.

[36] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: 2017 IEEE Conference on Computer Vision

and Pattern Recognition (CVPR), 2017, pp. 936–944. doi:10.1109/CVPR.2017.106.

[37] A. Kirillov, R. Girshick, K. He, P. Dollar, Panoptic feature pyramid networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. doi:10.1109/CVPR.2019.00656.

[38] A. Dutta, A. Zisserman, The VIA annotation software for images, audio and video, in: Proceedings of the 27th ACM International Conference on Multimedia, ACM, 2019. doi:10.1145/3343031.3350535.

[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. doi:10.1109/CVPR.2016.90.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, Imagenet: a large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[41] L. Bossard, M. Guillaumin, L. V. Gool, Food-101 - mining discriminative components with random forests, in: European Conference on Computer Vision, 2014. doi:10.1007/978-3-319-10599-4_29.

[42] W. Min, Z. Wang, Y. Liu, M. Luo, L. Kang, X. Wei, X. Wei, S. Jiang, Large scale visual food recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (8) (2023) 9932–9949. doi:10.1109/TPAMI.2023.3237871.

[43] P. Chattopadhyay, R. Vedantam, R. R. Selvaraju, D. Batra, D. Parikh, Counting everyday objects in everyday scenes, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi:10.1109/CVPR.2017.471.

[44] G. Jocher, A. Chaurasia, J. Qiu, YOLO by Ultralytics (Jan 2023).
URL https://github.com/ultralytics/ultralytics

Huu-Thanh Nguyen is currently working towards a PhD degree at VIREO Group, Department of Computer Science, City University of Hong Kong. He was a visiting

researcher with NExT++ research centre at National University of Singapore in 2018 and 2019. His research interests include deep learning and computer vision. His works focus on food recognition, counting, detection, segmentation, and recipe retrieval.

Yu Cao is a PhD student in the School of Computing and Information Systems at Singapore Management University. He received his master's degree at National University of Singapore in 2017. He was a software engineer in the NExT++ research centre at National University of Singapore from 2017 to 2020, focusing on building the dietary tracking system. His research interests include deep learning, computer vision and human-computer interaction.

Chong-Wah Ngo is a Professor with the School of Computing and Information Systems, Singapore Management University. His research interests include large-scale multimedia information retrieval, video computing, multimedia mining, and visualization. Prof. Ngo was an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA (2011-2014), Conference Co-Chair of MM 2019 and ICMR 2015, Program Co-Chair of MMM 2012 and ICMR 2012, Chairman of ACM (Hong Kong Chapter) from 2008 to 2009.

Wing-Kwong Chan is an Associate Professor at the City University of Hong Kong. His current main research interest is software engineering, program analysis and software infrastructure for AI-based systems. Currently, he is a Special Issues Editor of the Journal of Systems and Software, associate editor of Software Testing, Verification and Reliability and International Journal of Creative Computing, and program chair of COMPSAC 2021 and AITest 2021.