Foodness Proposal for Multiple Food Detection by Training with Single Food Images

Wataru Shimoda Keiji Yanai

Department of Informatics,
The University of Electro-Communications, Tokyo
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585 JAPAN
{shimoda-k,yanai}@mm.inf.uec.ac.jp

ABSTRACT

We propose a CNN-based "food-ness" proposal method which requires neither pixel-wise annotation nor bounding box annotation. Some proposal methods have been proposed to detect regions with high "object-ness" so far. However, many of them generated a large number of candidates to raise the recall rate. Considering the recent advent of the deeper CNN, these methods to generate a large number of proposals have difficulty in processing time for practical use. Meanwhile, a fully convolutional network (FCN) was proposed the network of which localizes target objects directly. FCN saves computational cost, although FCN is essentially equivalent to the sliding window search. This approach made large progress and achieved significant success in various tasks.

Then, in this paper we propose an intermediate approach between the traditional proposal approach and the fully convolutional approach. Especially we propose a novel proposal method which generates high "food-ness" regions by fully convolutional networks and back-propagation based approach with training food images gathered from the Web.

Keywords

foodness detection, food segmentation, convolutional neural network, deep learning, UECFOOD-100

1. INTRODUCTION

In recent years, a recording food habit for health control has become popular. Food records can provide various numerical statements such like calories and nutrition. These numerical statements have potential to solve the problems of unbalanced food habit. While food recordings are useful for us, food recording imposes us on some labor. This problems are remarkable with manual way, most of people are expected to feel that it is troublesome labor to input names of food items by text at every meal. To simplify food recordings are needed.

Food image recognition plays an important role in simplifying food recordings. If we replace manual recordings to taking a picture, we can reduce burden dramatically and may get feeling free even though the process is needed at every meal. To simplify food recording also matches the recent fashion so that there are trends uploading food images to SNS. In technical aspects, food image recognition also matches recent fashion because of recent large advance of deep neural network. Deep Convolutional Neural Network (DCNN) have been proved for large-scale object recognition at ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012. Krizhevsky et al. [21] won ILSVRC2012 with a large margin to all the other teams who employed a conventional hand-crafted feature approach. Tremendous advance with DCNN still continue now since that success. In ILSVRC Challenge 2015, Szegedy et al. [40] and Simonyan et al. [37] overcome human score. In ILSVRC Challenge 2016, He et al. [13] achieved state-of-the-arts and even human performance with over one hundred layers. Besides, DCNN also has broken the state-of-the-arts in the other various tasks of computer vision. Especially, DCNN have made great progress in object detection and semantic segmentation tasks.

In food recognition, object detection and semantic segmentation are also important tasks. Detection task provides bounding box with target object and semantic segmentation task predicts class labels of each pixel in a given image. We can estimate food position and obtain food sizes with bounding-box-level or pixel-level from the results of these tasks. In particular, we consider predicting food sizes are important in food recognition so that food sizes must be related to food amount. Precise food calorie estimation is one of the most practical problem in food recording. The knowledge on food amount related to a food calorie are widely accepted as common understanding in general. Hence, object detection and semantic segmentation will lead calorie estimation better.

However, most of the CNN based object detection and semantic segmentation methods assume that additional annotation are available such like bounding-box annotation and pixel-wise annotation, which is costly to obtain in general.

MADiMa'16, October 16 2016, Amsterdam, Netherlands

ISBN 978-1-4503-4520-0/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2986035.2986043

On the other hand, collecting images with image-level annotation is relatively easier than those pixel-level annotation, since many images attached with tags are available on handcrafted open image data sets such as ImageNet as well as on the Web. In this work, we focus on weakly-supervised semantic segmentation which requires neither pixel-wise annotation nor bounding box annotation, but only image-level annotation.

In general, object detection and semantic segmentation with bounding-box annotation or pixel-wise annotation are called as fully supervised method, while object detection and semantic segmentation with only image-level annotation are called as weakly supervised method. In the recent years, some weakly supervised object detection and semantic segmentation methods with DCNN were proposed. However, most of the previous works were tested on only Pascal VOC 2012 dataset. Though Pascal VOC 2012 dataset includes multi-label images, most of the images in the food image dataset have only a single label. The characteristics of both dataset are different. Therefore, we believe that it is meaningful to confirm the effectiveness of weakly supervised methods for food image domain. In this paper, we focus on weakly supervised detection and segmentation for food images and confirm the effectiveness of weakly supervised methods for food image domain.

In this work, we use "Distinct Class-specific Saliency Maps (DCSM)" [36] as a weakly supervised detection and segmentation method. This method showed high performance in weakly supervised segmentation task, and it can be adapted to the other targets easily. However, DCSM is not always effective for food image domain, since food image datasets contain mainly single label training images and the number of a few multiple label images is very limited. Therefore, instead of using DCSM for food segmentation directly, we use DCSM to generate high "food-ness" regions in the similar way to the traditional detection or segmentation methods such as RCNN (Region CNN) [9] and SDS (Simultaneous Detection and Segmentation) [11]. Because the number of food categories are larger, and food textures across different kinds of foods are more similar to each other compared to the case of generic objects, firstly we estimate food candidate regions by DCSM, and secondly classify each of the candidate regions into one of all the trained food categories (100 in the experiment).

We summarize our contributions as below:

- We proposed a novel DCSM-based proposal method for food images.
- We made experiments and confirmed the effectiveness of the proposed method.

2. RELATED WORKS

In the paper, we focus image recognition on food domain. Our work is also related to object detection and semantic segmentation. As related works, we mention the previous food recognition studies including food detection and segmentation, and the recent CNN-based detection and segmentation works for generic images.

2.1 Food Recognition

Food image recognition is one of the promising applications of visual object recognition, since it will help estimate food calories and analyze people's eating habits for healthcare. Many works have been published so far [2, 15, 18, 23, 4, 1, 45].

Meanwhile, recently the effectiveness of Deep Convolutional Neural Network (DCNN) have been proved for large-scale object recognition at ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012. Krizhevsky et al. [21] won ILSVRC2012 with a large margin to all the other teams who employed a conventional hand-crafted feature approach. In the DCNN approach, an input data of DCNN is a resized image, and the output is a class-label probability. That is, DCNN includes all the object recognition steps such as local feature extraction, feature coding, and learning. In general, the advantage of DCNN is that it can estimate optimal feature representations for datasets adaptively, the characteristics of which the conventional hand-crafted feature approach do not have. In the conventional approach, we extract local features such as SIFT and SURF first, and then code them into bag-of-feature or Fisher Vector representations. Regrading food image recognition, the classification accuracy on the UECFOOD-100 dataset [23] was improved from 59.6% [18] to 72.26% [17] by replacing Fisher Vector and liner SVM with DCNN.

However, most of the works assumed that one food image contained only one food item. They cannot handle an image which contains two or more food items such as a hamburger-and-french-fries image. To list up all food items in a given food photo and estimate calories of them, segmentation of foods is needed. Some works attempted food region segmentation [23, 25, 16, 14].

Matsuda et al. [23] proposed to used multiple methods to detect food regions such as Felzenszwalb's deformable part model (DPM) [7], a circle detector and the JSEG region segmentation method [6].

He et al. [14] employed Local Variation [8] to segment food regions for estimating total calories of foods in a given food photo. In some works for mobile food recognition [25, 16], they asked users to point rough locations of each food item in a food photo, and perform GrabCut [34] to extract food item segments.

In addition, there are some study for estimating calorie with computer vision technique. Kong et al. [19] reconstructed 3D food models with multi-angle pictures and estimated calories from cubic volume of 3D models. Chen et al. [5] recognized an image and computed cubic volume from depth information. Note that they obtained depth information by sensor. 3D based calorie estimation methods tend to impose some labors on users. On the other hand, Myers et al. [24] proposed calorie estimation application which called "im2calorie". They obtained each pixel depth information by prediction of deep learning and estimated calories. However, Myers et al. have not achieved practical stage.

Pouladzadeh et al. [32] estimated calories from segmentation results of an image. They defined a thumb as a base object, and estimated food volumes and calories from area ratios of a thumb area and foods. While we can always take a food picture with a thumb, this method has potential to make looking worse and taking picture with only one hand is difficult.

In contrast to previous works, we tackled food image detection in a weakly supervise manner.

2.2 CNN-based Fully-Supervised Object Detection and Semantic Segmentation

As the early works on CNN-based semantic segmentation, Girshick et al. [9] and Hariharan et al. [11] proposed object segmentation methods using region proposal and CNN-based image classification. Firstly, they generated 2000 region candidates at most by Selective Search [41], and secondly apply CNN image classification by feed-forwarding of the CNN to each of the proposals. Finally they integrated all the classification results by non-maximum suppression and generated the final object regions. Although these methods outperformed the conventional methods greatly, they had a drawback that they required long processing time for CNN-based image classification of many region proposals.

While Girshick et al. [9] and Hariharan et al. [11] took advantage of excellent ability of a CNN on image classification task for semantic image segmentation in a relatively straightforward way, He et al. [12], Long et al. [22] proposed CNN-based semantic segmentation in a hierarchical way. A CNN is much different from conventional bag-of-features framework regarding multi-layered structure consisting of multiple convolutional and pooling layers. Because CNN has several pooling layers, location information is gradually losing as the signal is transmitting from the lower layers to the upper layers. In general, the lower layers hold location information in their activations, while the upper layers hold local information weakly. He et al. [12] proposed spatial pyramid pooling which exploited lower layer information for object detection and reduced large computational cost from RCNN [9]. Long et al. [22] replaced fully connected layers to convolutional layers and learned matrix outputs of fully convolutional networks directly with pixel-wise-annotation which was often called as an end-to-end network. Later, Ren et al. [33] proposed Faster RCNN which is an end-to-end network for object detection task.

2.3 CNN-based Weakly-Supervised Semantic Segmentation

Most of the conventional non-CNN-based weakly supervised segmentation method employed Conditional Random Field (CRF) with unary potentials estimated by multiple instance learning [42], extremely randomized hashing forest [43], and GMM [46].

As a CNN-based method, Pedro et al. [31] achieved weakly-supervised segmentation by using multi-scale CNN proposed in [35]. They integrated the outputs which contained location information with log sum exponential, and limited object regions to the regions overlapped with object proposals [27].

Pathak et al. [30, 29] and Papandreou et al. [28] achieved weakly-supervised semantic segmentation by adapting CNN models for fully-supervised segmentation to weakly-supervised segmentation. In MIL-FCN [30], they trained the CNN for full-supervised segmentation proposed in Long et al. [22] with a global max-pooling loss which enabled training of the CNN model using only training data with image-level labels. Constrained Convolutional Neural Network (CCNN) [29] improved MIL-FCN by adding some constraints and using fully-connected CRF [20]. Papandreou et al. [28] trained the DeepLab model [3] proposed as a fully-supervised model with EM algorithm, which is called as "EM-adopt". Both CCNN and EM-adopt generated pseudo-pixel-level labels from image-level labels using constraints and EM algorithms to train FCN and DeepLab which were originally proposed for fully supervised segmentation, respectively. Both showed Dense CRF [20] were helpful to boost segmentation performance even in the weakly supervised setting.

Meanwhile, Simonyan et al. [37] proposed a method to generate object saliency maps by back propagation (BP) over a pre-trained DCNN, and showed it enabled semantic object segmentation by applying GrabCut [34] using saliency maps as seeds. While all the above-motioned methods on weakly supervised segmentation employed only feed-forward computation, Shimoda et al. [36] proposed an improved weakly supervised segmentation method based on back-propagation (BP) computation.

Though the above-mentioned weakly supervised method achieved remarkable progress, they only tested their performance with the Pascal VOC 2012 dataset, i.e. using multi-label training images and including only general object classes. In this paper, we propose weakly supervised method with only single-label images for food object detection which is known as one of the fine-grained domain. Our method combines traditional proposal-based approach and fully convolutional approach. We show our method is robust to changing domain from training with single-labeled Web images to testing with multi-label images.

3. PROPOSED METHOD

We propose a new method to generate food-ness regions with weakly supervised annotation. Our method is based on Distinct Class-specific Saliency Maps (DCSM) [36] which is extension of Simonyan et al. [37]. In this section, we explain the DCSM and how to adopt the DCSM to food-ness proposal.

3.1 Overall Architecture

We follow traditional detection method using proposal. First of all, we generate proposal based on DCSM. Second, we recognize each candidate region. Finally, we unify overlapped candidates by Non Maximum Suppression (NMS). In this work, we prepare two CNNs for proposal and recognition. We illustrate an overview in Figure 1. The procedure of the proposed method process is as follows:

- Recognize an image.
- Sort each food class based on soft max output.
- Back-propagate the upper rank class scores.
- Subtract each class derivative value.
- Obtain food-ness proposals.
- Recognize each of the food-ness candidates.
- Unify overlapped candidates by NMS.

3.2 DCSM

In [37], Simonyan et al. regarded the derivatives of the class score with respect to the input image as class saliency maps. However, the position of an input image is the furthermost from the class score output in the deep CNN layers, which sometime causes weakening or vanishing of gradients. Instead of the derivatives of the class score with respect to the input image, Shimoda et al. [36] uses the derivatives with respect to feature maps of the relatively upper intermediate layers which are expected to retain more high-level semantic information. In addition, Shimoda et al. [36] apply some techniques which is known as effective in semantic segmentation to a backward approach. They select the maximum absolute values of the derivatives with respect to the feature maps at each location of feature maps across all the kernels, and up-sample them with bi-linear interpolation so that their size becomes the same as an input image.

The class score derivative v_i^c of a feature map layer is the derivative of class score Sc with respect to the layer L_i at the point (activation signal) L_i^0:

$$v_i^c = \left.\frac{\partial S_c}{\partial L_i}\right|_{L_i^0} \tag{1}$$

v_i^c can be computed by back-propagation. After obtained v_i^c, Shimoda et al. [36] up-sample it to w_i^c with bi-linear interpolation so that the size of an 2-D map of v_i^c becomes the same as an input image. Next, the class saliency map $M_i^c \in \mathcal{R}^{m \times n}$ is computed as

$$M_{i,x,y}^c = \max_{k_i} |w_{i,h_i(x,y,k)}^c|, \tag{2}$$

where $h_i(x, y, k)$ is the index of the element of w_i^c. Note that each value of the saliency map is normalized with $tanh$ for visualization with $\alpha = 3$.

$$\tanh(\alpha M_{i,x,y} / \max_{x,y} M_{i,x,y}) \tag{3}$$

The saliency maps of two or more different classes tend to be similar to each other especially in the image-level. The saliency maps by [37] are likely to correspond to foreground regions rather than specific class regions. To resolve that, Shimoda et al. [36] subtract saliency maps of the other candidate classes from the saliency maps of the target class to differentiate target objects from other objects. Here, Shimoda et al. [36] assume that they select several candidate classes with a pre-defined threshold and a pre-defined minimum number.

The improved class saliency maps with respect to class c, $\tilde{M}_i^c$, are represented as:

$$\tilde{M}_{i,x,y}^c = \sum_{c' \in candidates} \max\left(M_{i,x,y}^c - M_{i,x,y}^{c'}, 0\right) [c \neq c'], \tag{4}$$

where *candidates* is a set of the selected candidate classes. Subtraction of saliency maps resolved overlapped regions among the maps of the different classes.

Shimoda et al. [36] use fully convolutional networks (FCN) which accept arbitrary-sized inputs for multi-scale generation of class saliency maps. If the larger input image than one for the original CNN is given to the fully-convolutionalized CNN, the output becomes class score maps represented as $h \times w \times C$ where C is the number of classes, and h and w are larger than 1. To obtain CNN derivatives with respect to enlarged feature maps, Shimoda et al. [36] simply back-propagate the target class score map which is defined as $S_c(:, :, c) = 1$ (in the Matlab notation) with 0 for all the other elements, where c is the target class index.

The final class saliency map $\hat{M}^c$ averaged over the layers and the scales is obtained as follows:

$$\hat{M}_{x,y}^c = \frac{1}{|S||L|} \sum_{j \in S} \sum_{i \in L} \tanh(\alpha \tilde{M}_{j,i,x,y}^c), \tag{5}$$

where L is a set of the layers for which saliency maps are extracted, S is a set of the scale ratios, and α is a constant which we set to 3 in the experiments. Note that we assume the size of $\tilde{M}_{j,i}$ for all the layers are normalized to the same size as an input image before calculation of Eq. 5.

In [39], guided back-propagation (GBP) [39] was adopted as a back-propagation method instead of normal back-propagation (BP) used in [37]. The difference between two methods is in the backward computation through ReLU. GBP can visualize saliency maps with less noise components than normal BP by back-propagating only positive values of CNN derivatives through ReLU [39].

3.3 Food-ness Proposal

In this paper, we focus on training with single food images and testing with multiple foods images. In general, changing domain between training data and testing data causes performance drop. This problem was known as one of the cross domain problems or the domain adaptation problems. With the DCSM method this problem was also observed and accuracy dropped remarkably. We illustrate our situation with this domain adaptation problem using Figure 2 in food images.

In this paper, simply, we avoid this domain adaptation problem by using region proposals. Proposal methods generate object region candidates and these candidates should include target objects. When recognizing target objects in the candidates, we will obtain better results than recognizing raw images without proposals. Because, in our situation, test images might include multiple food images, while candidate regions even in the multiple food images can be regarded as single food images. Hence, the condition within candidate regions can be considered as similar to the condition of training images.

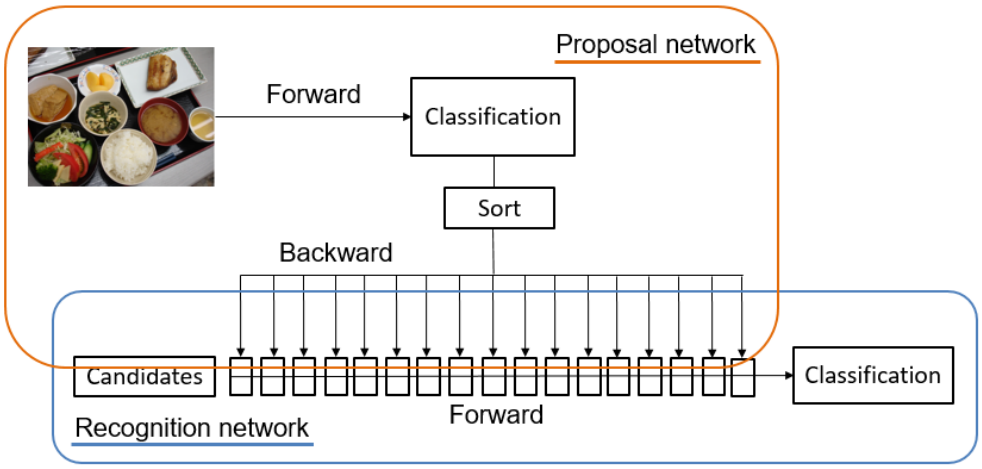

Figure 1: Processing flow of the our method.

Figure 2: Example of our cross domain situation.

RCNN [10] and SDS [11] are typical methods of detection and segmentation using proposals with DCNN. They use Selective Search [41] and MCG (Multiscale Combinatorial Grouping) [27] as proposal methods. These proposal methods are also typical and generate a lot of candidate about 2000 with local features. A large number of candidates raise recall but increase computational cost as well. We consider the candidates number about 2000 is too large and there are useless processes for food recognition. Therefore, we propose a novel proposal method for foods with DCNN.

Briefly, we adapt DCSM for food-ness proposal. The original DCSM approach is not effective for the cross domain problem we mentioned before. In fact, the estimated regions by DCSM with cross domain are not precise. However, we observed that most of these regions corresponds to any of the trained food items in an image. Interestingly, estimated regions for food classes which is not included in a given image still belong to the other existing objects, and some regions fit with food regions as shown in Figure 3. This means that CNN training with cross domain could not transfer knowledge precisely for food, but could learn rough food concepts. Therefore, we regard the regions estimated by DCSM as food region candidates which have high "foodness".

In practice, to adapt DCSM for food-ness proposal we only augment *candidate* in Eq. 4. However, we do not aggregate multi-input-scale results due to increasing computational cost. We can obtain estimated regions P with candidate numbers.

$$\mathrm{P}^c = \frac{1}{|\mathrm{L}|}\sum_{i\in \mathrm{L}} \tanh(\alpha \tilde{\mathrm{M}}_i^c), \tag{6}$$

P^c are probability map such like saliency maps. We cut P^c with a threshold so that this probability maps often respond to some foods. In other words, we divide the probability maps P^c into several regions based on the peaks of those. Incidentally we discard some isolate small regions.

$$\hat{\mathrm{P}}_k^c \in \mathrm{P}^c \tag{7}$$

where K and $k \in \{1, 2, \cdots, K\}$ represents the number of regions and a region index, respectively.

Finally we regard these regions $\hat{\mathrm{P}}_k^c$ as food-ness proposal. To sum up, we augment the candidates class in the DCSM method and obtain regions from output probability with DCSM. We can increase the number of candidate classes until the number of target classes. The maximum number is 100 in case of using UECFOOD-100 dataset [23] which we used in the experiments. We will discuss how to choose augment classes number in Section 4.2.

4. CNN TRAINING

In this paper, we use VGG-16 [38] as a base convolutional network for fine-tuning with food images. We build two kinds of the networks separately. They are a proposal network and a recognition network. We fine-tune VGG-16 as proposal network with fully convolutional technique. We also fine-tune VGG-16 for recognition network with a traditional way. In this section, we mention about detail of these

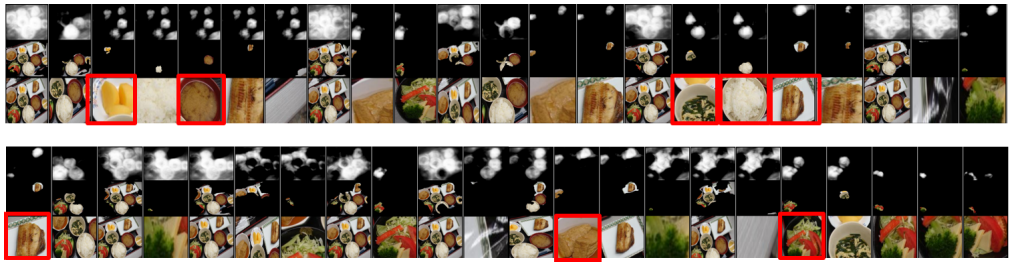

Figure 3: Proposal results. First row show saliency obtained from DCSM. Second row shows regions obtained from saliency maps. Third row shows bounding boxes we recognize. Red rectangle means it seems to be a good candidate.

two networks.

4.1 Proposal Network

As an off-the-shelf basic CNN architecture for a proposal network, we use the VGG-16 [38] pre-trained with 1000-class ILSVRC datasets. In our framework, we fine-tune a CNN with training images with only image-level annotation. Recently, fully convolutional networks (FCN) which accept arbitrary-sized inputs are used commonly in the works on CNN-based detection and segmentation such as [26] and [22], in which fully connected layers with n units were replaced with the equivalent convolutional layers having n 1×1 filters. Following them, we introduce FCN to enable multi-scale generation of class saliency maps. When training, we insert global max pooling before the final loss function layer to deal with larger input images than the images used for pre-training of the VGG-16. Global max pooling is an operation which is commonly used for training of CNN in the works on weakly supervised segmentation. The purpose of this operation is to convert last output to a vector from a matrix. Therefore we can train FCN with usual image-level-label and soft-max loss.

In the training time, we replace fully connected layers with convolution layers for VGG-16 model and add as global max pooling layer as we mentioned above.

4.2 Recognition Network

For recognition time, though we change only last layer for food category outputs, we prepare additional categories for training. The purpose of the recognition network is to classify each of the candidate regions estimated by the proposal network. The conditions of recognizing candidates is different from the condition of the training phase in terms of including non-target-category-object images and small-food-patch images. In RCNN [10] and SDS [11], they consider only non-target-category-object images as background so that they were tested on generic object detection dataset. However, food recognition is different from generic object recognition. Food recognition is similar to texture recognition rather than object recognition, since most of foods have no fixed shapes. Their shapes varies depending on the way to serve. Therefore, food tends to be recognized with their patches by DCNN. For example, in case of recognition of "a dog" with DCNN, the small part of dogs such as legs and tails would have low scores in the dog probability map, while, in case of food recognition, even small patches of rice will have high score in the rice probability map. To sum up, DCNN cannot discriminate general objects with limited parts but can discriminate foods with minimum patch information. To prevent small part of foods from being recognized as foods, we add small patches of foods to the non-food class.

Furthermore we add low-resolution food images. Because we found that a low-resolution image tends to be classified as food patch category. We guess this is why a small-food-patch image tends to be a low-resolution image. Therefore, we add low-resolution images to each food class. Our intuition is that if we remove low-resolution images from training images, low-resolution images will not be recognized as small-food-patch images.

We augment training images by cropping and resizing. Practically, we cropped three images from each training image as food path by random position with random sizes. The minimum size of cropped image is 50 and maximum size is 150. Note that original image size is 256, i.e. each cropped image size rate for original image is about 0.2 and 0.6. We also prepare three images for low-resolution image by down-sampling and rescaling. We defined down-sample sizes randomly. The minimum down-scale size is 10, and maximum size is 256 which is equal to original image size. Finally we obtain an augmented training image set the size of which is seven times larger than the original training image set.

5. EXPERIMENTS

In the experiments, we used the UECFOOD-100 dataset [23] and Web food images. The UECFOOD-100 dataset [23] consists of 100 class food categories and each category have 100 images. Note that each food item is annotated with bounding boxes.

In addition, we collected food images of the same category as the UECFOOD-100 dataset from the Web. The collected Web food image set have 100 categories and 1000 images for each category without bounding box annotation. Most of these Web food images are obtained from the twitter stream [44] and some images are obtained from Bing API.

We use 1175 multiple-food images included in UECFOOD-

100 as test dataset for object detection. All detection evaluations are based on mean average precision (MAP) which is used in the Pascal VOC detection task as an evaluation measure.

In addition to 100 food categories, we gathered non-food images as the 101-th category. We built a negative food image set by gathering images using the Web image search engines with query keywords which are expected to related to noise images such as "street stall", "kitchen", "dinner party" and "restaurant" and excluding food photos by the CNN trained with UECFOOD-100. We collected 10,000 non-food images as negative food samples.

In the experiments, we used a mixture of UECFOOD-100 and Web food images, and a dataset containing onlty Web food images.

5.1 Augmentation of Training Data for Recognition Network

First of all, we evaluated three cases of recognition network with two datasets with a fixed proposal network setting. Table 1 shows the average precision by three kinds of training data augmentation with two training data.

"Foodness 2" achived higher performance than "Foodness 1". This means that adding small patch to the non-food class is effective. On the other hand, "Foodness 3" achieved the better results than "Foodness 2". We can see that adding low-resolution images is also effective for recognition network. "Foodness 4, 5, 6" are trained with only Web food images. The average precision (AP) of "Foodness 6" is higher than AP of "Foodness 4" and "Foodness 5", which shows that adding small patch to the on-food class and using low resolution images are also effective. "Foodness 6" exibits drop of AP compared with "Foodness 3", but outperformed "Foodness 2". From these results, we can say that increasing the number of training images is effective.

5.2 Global Pooling for Proposal Network

Next, we compare two general global-pooling operation, global average pooling and global max pooling. Table 2 show comparison of final pooling operation for each two data. Though the previous papers such as [47] mentioned the effectiveness of global average pooling for object localization, in food-ness proposal, global max pooling is better than global average pooling. We guess global-max-pooling network captured smaller items than global-average-pooling and this matches food item detection task. We need to check which global-pooling operation matches to the task.

5.3 Comparison with Other Traditional Proposal Methods

Next, we compare our proposal method with other traditional proposal methods. We evaluate the methods in terms of mean AP and speed factors. We prepare two traditional proposal method as baselines. Selective Search [41] and Multiscale Combinatorial Grouping (MCG) [27] are region proposal methods. Both of them generate a large number of candidates which is around 2000. To evaluate our proposal method, we changed the number of candidate classes. Small candidate class leads more small computational cost since backward time is reduced. Table 3 shows comparison results. Note that recognition time is a theoretical value computed from the candidate number and computational cost for one image. Foodness with 30 candidate classes AP are better than [41] and [27] even though 40 times smaller for candidate numbers. In addition, even if we reduced the candidate class number, the mean AP still holds 30%. This mean that our proposal has good quality for food-ness detection.

Figure 4: Examples of results. Left images are input images. Center images are detection results. Right images are ground truth images

6. CONCLUSIONS

We proposed a CNN-based food-ness proposal method which required neither pixel-wise annotation nor bounding box annotation. We adopted an intermediate approach in traditional proposal approach and fully convolutional approach. Especially we proposed a novel proposal method which generated food-ness regions by fully convolutional networks based backward approach with training Web food images. Then, we achieved reducing computational cost while keeping quality for food detection.

In the future work, we will try weakly-supervised food segmentation in addition to detection, because our food-ness proposal can generates segmentation results as well. Although we used Web images in this work, the categories were limited to the same ones to UECFOOD-100. This can be regarded as training sample augmentation with Web images. For future work, we like to train any other kinds of foods than UECFOOD-100 from the Web automatically. If

Table 1: Mean average precision with six conditions over all the 100 categories, 53 categories (more than 10 items of which are included in the test data), and 11 categories (more than 50 items of which are included in the test data).

method	small-patch class	low-resolution images	training with only web images	100class (all)	53class (#item ≥ 10)	11class (#item ≥ 50)
Foodness 1	-	-	-	30.0	29.3	31.9
Foodness 2	✓	-	-	33.7	39.0	33.6
Foodness 3	✓	✓	-	39.5	46.0	38.9
Foodness 4	-	-	✓	33.5	35.1	33.3
Foodness 5	✓	-	✓	32.2	34.8	31.8
Foodness 6	✓	✓	✓	36.4	39.9	36.3

Table 2: Comparison of global pooling operation for food-ness.

method	training with only web images	100class (all)	53class (#item ≥ 10)	11class (#item ≥ 50)
Foodness (average pooling)	-	39.5	46.0	38.9
Foodness (average pooling)	✓	36.4	39.9	36.3
Foodness (max pooling)	-	39.9	48.3	37.6
Foodness (max pooling)	✓	38.9	42.5	38.1

we can this, we can bulit a system to recognize any kinds of foods.

7. REFERENCES

[1] M. Bosch, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp. Combining global and local features for food identification in dietary assessment. In *Proc. of IEEE International Conference on Image Processing*, 2011.

[2] L. Bossard, M. Guillaumin, and L. V. Gool. Food-101 - mining discriminative components with random forests. In *Proc. of European Conference on Computer Vision*, 2014.

[3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and Y. A. L. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *Proc. of International Conference on Learning Representations*, 2015.

[4] M. Chen, Y. Yang, C. Ho, S. Wang, S. Liu, E. Chang, C. Yeh, and M. Ouhyoung. Automatic chinese food identification and quantity estimation. In *SIGGRAPH Asia*, 2012.

[5] M. Chen, Y. Yang, C. Ho, S. Wang, S. Liu, E. Chang, C. Yeh, and M. Ouhyoung. Automatic chinese food identification and quantity estimation. In *SIGGRAPH Asia Technical Briefs*, page 29, 2012.

[6] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Image segmentation using local variation. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 98–104, 1998.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2014.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[11] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Proc. of European Conference on Computer Vision*, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Proc. of European Conference on Computer Vision*, pages 346–361, 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2016.

[14] Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp. Food image analysis: Segmentation, identification and weight estimation. In *Proc. of IEEE International Conference on Multimedia and Expo*, pages 1–6, 2013.

[15] H. Kagaya, K. Aizawa, and M. Ogawa. Food detection and recognition using convolutional neural network. In *Proc. of ACM International Conference Multimedia*,

Table 3: Comparison with other traditional proposal method.

method	100class (all)	53class (#item ≥ 10)	11class (#item ≥ 50)	proposal speed[s]	Recognition speed for candidates[s]
Selective Search [41]	38.3	39.1	35.7	7.6	35.0
Multiscale Combinatorial Grouping [27]	33.9	43.7	33.4	2.5	35.0
Foodness with 10 candidate classes	33.1	33.0	33.2	0.5	1.1
Foodness with 20 candidate classes	36.5	40.1	37.7	1.0	2.6
Foodness with 30 candidate classes	38.9	42.5	38.1	1.4	3.8

pages 1085–1088, 2014.

[16] Y. Kawano and K. Yanai. Real-time mobile food recognition system. In *Proc. of IEEE CVPR International Workshop on Mobile Vision (IWMV)*, 2013.

[17] Y. Kawano and K. Yanai. Food image recognition with deep convolutional features. In *Proc. of ACM UbiComp Workshop on Workshop on Smart Technology for Cooking and Eating Activities (CEA)*, 2014.

[18] Y. Kawano and K. Yanai. Foodcam: A real-time food recognition system on a smartphone. *Multimedia Tools and Applications*, pages 1–25, 2014.

[19] F. Kong and J. Tan. Dietcam: Automatic dietary assessment with mobile camera phones. In *Proc. of Pervasive and Mobile Computing*, pages 147–163, 2012.

[20] P. Krahenbuhl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, 2011.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[22] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2015.

[23] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proc. of IEEE International Conference on Multimedia and Expo*, pages 1554–1564, 2012.

[24] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy. Im2calories: Towards an automated mobile vision food diary. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[25] C. Morikawa, H. Sugiyama, and K. Aizawa. Food region segmentation in meal images using touch points. In *Proc. of ACM MM WS on Multimedia for Cooking and Eating Activities (CEA)*, pages 7–12, 2012.

[26] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? -weakly-supervised learning with convolutional neural networks. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2015.

[27] A. Pablo, J. T. Jordi, P., M. Ferran, and M. Jitendra. Multiscale combinatorial grouping. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2014.

[28] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. In *Proc. of IEEE International Conference on Computer Vision*, 2015.

[29] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proc. of IEEE International Conference on Computer Vision*, 2015.

[30] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. In *Proc. of International Conference on Learning Representations*, 2015.

[31] P. Pedro and C. Ronan. From image-level to pixel-level labeling with convolutional networks. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2015.

[32] P. Pouladzadeh, S. Shirmohammadi, and R. Almaghrabi. Measuring calorie and nutrition from food image. *IEEE Transactions on Instrumentation and Measurement*, pages 1947–1956, 2014.

[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[34] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

[35] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proc. of International Conference on Learning Representations*, 2014.

[36] W. Shimoda and K. Yanai. Distinct class-specific saliency maps for weakly supervised semantic segmentation. In *Proc. of European Conference on Computer Vision*, 2016.

[37] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proc. of*

International Conference on Learning Represenation Workshop Track, 2014.

[38] K. Simonyan, A. Vedaldi, and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of International Conference on Learning Representations*, 2015.

[39] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *Proc. of International Conference on Learning Representations*, 2015.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2015.

[41] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[42] A. Vezhnevets and J. M. Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2010.

[43] A. Vezhnevets and J. M. Buhmann. Weakly supervised structured output learning for semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2012.

[44] K. Yanai and Y. Kawano. Twitter food image mining and analysis for one hundred kinds of foods. In *Proc. of Pacifit-Rim Conference on Multimedia (PCM)*, 2014.

[45] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2010.

[46] L. Zhang, Y. Gao, Y. Xia, K. Lu, J. Shen, and R. Ji. Representative discovery of structure cues for weakly-supervised image segmentation. *IEEE Transaction on Multimedia*, 16(2):470–479, 2014.

[47] B. Zhou, A. Khosla, A. Lapedriza, and A. Oliva, A. Torralba. Learning deep features for discriminative localization. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2016.