Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Food det: Detecting foods in refrigerator with supervised transformer network

Yousong Zhu[a,b,1,*], Xu Zhao[a,b,1], Chaoyang Zhao[a,b], Jinqiao Wang[a,b], Hanqing Lu[a,b]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No.95, Zhongguancun East Road, Beijing 100190, China
[b] University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 4 December 2018
Revised 3 August 2019
Accepted 28 October 2019
Available online xxx

Communicated by Dr. Li Zhifeng

Keywords:
Food detection
Spatial transformer
Object detection

ABSTRACT

Most of existing methods mainly focus on the food image recognition which assumes that one food image contains only one food item. However, in this paper, we present a system to detect a diversity of foods in refrigerator where multiple food items may exist. In view of the refrigerator environment, we propose a food detection framework based on the supervised transformer network. More specifically, the supervised transformer network, dotted as RectNet, is first proposed to automatically select the irregular food regions and transform them to the frontal views. Then, based on the rectified food images, we further propose an end-to-end detection network that predicts the categories and locations of food items. The proposed detection network, called Lite Fully Convolutional Network (LiteFCN), is evolved from the advanced object detection algorithm Faster R-CNN while several significant improvements are tailored to achieve a higher accuracy and keep inference time efficiency. To validate the effectiveness of each component of our method, we build a real-world refrigerator dataset with 80 classes. Extensive experiments demonstrate that our methods achieve the state-of-the-art results, which improves the baseline by a large margin, *e.g.*, 3–5% in terms of F-measure. We also show that the proposed detection network achieve a competitive result on the public PASCAL VOC2007 dataset, which outperforms the Faster R-CNN by 2.3% with a higher speed.

1. Introduction

With the rapid development of economy and the improvement of living conditions, nowadays people are more concerned about their nutrition habits and personal health. Therefore, automatic food analysis plays an important role in recording our daily dietary intake. In recent years, due to the remarkable progress [1–4] of deep neural network, computer vision has once again become the focus of attention. Through the computer vision technology, we can constantly enrich the recognition of categories of food [5,6], record the daily dietary data and the preferences of users [7] and analyze their eating habits. According to the nutritional analysis of family diet structure, we can further provide several applications like food preference learning, food calorie estimation and personalized recipe recommendation. As a necessity for household food storage, refrigerators are likely to become the central system of intelligent food analysis and it is of great significance to the construction of the whole intelligent life. Therefore, in this paper, we first focus on the multi-class food detection in refrigerator to provide comprehensive food analysis.

Actually, the problem of food detection has been usually addressed as a binary classification problem [8–10], where the algorithm simply has to determine the presence or absence of food in a given image. The main problem of these works is that they cannot find the precise regions (bounding boxes) of foods, thus resulting in only obtaining limited information and failure to provide further follow-up services. Although a few existing approaches [11–13] have been proposed to simultaneously address the problem of food localization, they only work in simple conditions where the food items are well separated (dish detection) and the number of food items is small. Moreover, some of these methods exploit traditional, hand-crafted visual features and only use machine learning methods to classify the categories. However, in this paper, we deal with the multi-class food detection in refrigerator where the food items are always placed in a crowded and cluttered way and suffered from visual distortions as well as illumination changes due to various surroundings. Fig. 1 shows a real-world example of the inside view of our refrigerator, which includes two

[*] Corresponding author at: National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No.95, Zhongguancun East Road, Beijing 100190, China.
E-mail address: yousong.zhu@nlpr.ia.ac.cn (Y. Zhu).
[1] Equal contribution.

https://doi.org/10.1016/j.neucom.2019.10.106
0925-2312/© 2019 Elsevier B.V. All rights reserved.

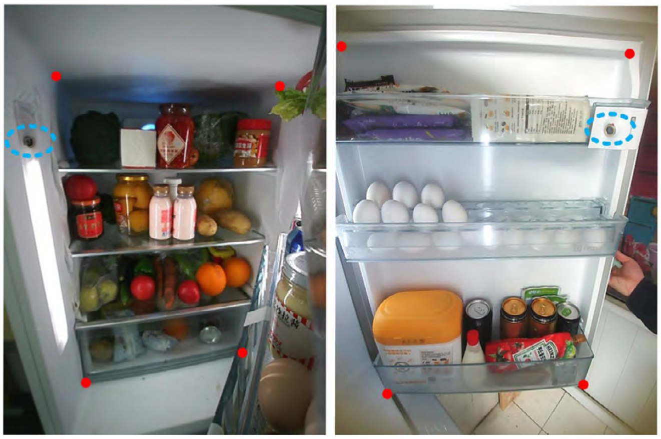

Fig. 1. Example snapshots captured by the cameras. The dash ellipses indicates the position of cameras. The left side image represents the refrigerator box captured by the camera on the door. The right side image represents the refrigerator door captured by the camera on the box. Both the snapshots include various food categories and a large number of food items. Moreover, they also suffer from varying degrees of spatial variances due to the problem of the capture angle. The red points indicate the predicted corner landmarks.

snapshots representing the box and door of refrigerator. Both the snapshots contain various food categories and a large number of food items. Therefore, it is necessary to develop a more robust and sophisticated detection algorithm.

Meanwhile, generic object detection algorithms have been proposed consecutively [14–18] and achieved significant progress due to the successful application of convolutional neural network. These object detectors typically can be categorized into proposal-free detectors and proposal-based detectors.The proposal-free detectors usually achieve a sweet-spot of fast speed and reasonable accuracy. The proposal-based detectors provide a two-step solution: by generating object proposals, and then classifying each proposal into different categories. Usually, the proposal-based detectors can achieve a very high accuracy by using a heavy prediction head, thus leading to a relatively slow inference speed. One of the most classical proposal-based detector is Faster R-CNN [15]. Therefore, inspired by the generic object detection approaches, we propose a efficient yet accurate detection network for multi-class food detection. Specifically, our method is based on the proposal-based detector Faster R-CNN, but we propose several effective designs to maximize the shared convolutional computation and lighten the prediction head, thus improving the detection accuracy without losing time efficiency. The optimized detection network is denoted as LiteFCN and strikes the best trade-off between the speed and accuracy.

In addition, as shown in Fig. 1, the snapshots captured by the cameras undergo varying degrees of spatial variances due to the uncertainty of the angle of each capture. Actually, the cameras are triggered only when the fridge door is closed to a default range of angle, which results in the snapshots captured by the cameras suffer from scaling, out-of-plane rotations and other generic warping. Therefore, it is necessary to learn a spatial transformer to not only select the regions of the captured snapshots that are most relevant, but also to transform those regions to a canonical and expected pose for better detecting foods and displaying to the users.

Referring to the idea of Spatial Transformer Network (STN) [19], we propose a new supervised transformer network, dubbed as RectNet, to adaptively rectify the images. Different from the STN, our RectNet first conducts spatial transform on the input image directly instead of the convolutional feature maps, and the rectified image is further served for the subsequent food detection and results exhibition. Second, we predict the four corner landmarks to select the food region that needs to be transformed as shown in Fig. 1 (the red points). Note that the canonical positions of the corner landmarks in the rectified image and the predicted corner landmarks in the original image jointly defines the spatial transform from the original image. Finally, the learning of the prediction network of corner landmarks are explicitly supervised by the annotated corner landmarks in each image. We note that the prediction of landmarks is more flexible and can deal with large-scale transformations, while STN regresses the transformation parameters directly and is implicitly supervised by the final recognition objective, which can only cope with some tiny rotations, translations, and scale transformations. Moreover, in order to alleviate the extra computational burden brought by transformer network, we further build an efficient Inception-like backbone for RectNet, thus enabling nearly cost-free spatial transform operations.

In summary, the main contributions of this work are as follows:

1. We propose a novel supervised transformer network, named as RectNet, which enables to learn the optimal positions of food regions to best rectify the images for food detection.
2. We propose a new food detection network, denoted as LiteFCN, which achieves a higher accuracy while keeping the speed.
3. We collect a real-word food dataset within refrigerator where we have validated the effectiveness of our proposed method. We also achieve a competitive result on PASCAL VOC2007 dataset, *i.e.*, a mAP of 81.3% with 10.5 fps, which further verify the superiority of our detection network.

2. Related work

Food recognition. Many research works focus on food recognition [5,6,8,20–22]. Niki et al. [20] introduced a slice convolution block to capture the vertical structure of food dishes for specific food classification. Shota et al. [5] proposed a personalized classifier to learn the novel classes incrementally, which attains performance comparable to that of the CNN classifier at first and becomes more accurate after sequential personalization without any heavy retraining. Marc et al. [21] explored the problem of food ingredients recognition from a multi-label perspective by using a CNN model. Several works [23–25] exploited external knowledge, such as menus, scenarios and geographic locations, to develop context-specific dish recognition. For example, Luis et al. [24] proposed a probabilistic model that connects locations, restaurants, dishes and visual features to simplify the dish recognition and improve the performance. Several food dataset are also publicly available, such as UECFOOD-100 [11], UECFOOD-256 [26], ETHZ Food-101 [27], and Food201-MultiLabel [28]. However, these methods just treat the food recognition as a problem of image classification which cannot accurately obtain the location and size of food items, thus resulting in limited food analysis and application.

In fact, few works have been proposed to simultaneously deal with food localization and recognition [11–13,29,30]. Marc et al. [29] proposed to first produce a food activation map on the input image for generating proposals and then recognize each food proposal using a convolution neural network. Wataru et al. [30] presented a CNN-based food image segmentation which requires no pixel-wise annotation. The proposed method also consists of several off-line procedures, like generating candidates, grouping bounding boxes, obtaining saliency maps and extracting segments. Joachim et al. [13] exploited edge detection to locate the dish and adopted the Seed Region Growing (SRG) [31] to segment food items. All of these methods separate food detection from multiple off-line stages, which is inconvenient for training and test. Moreover, both the methods can only copy with some simple situations, such as clean backgrounds, non-occlusions and few food items in an image. Therefore, in this paper, we first propose to detect foods in the crowded and cluttered refrigerator environment. In addition, a lightweight and end-to-end food detection network is designed to simultaneously output the location and category of each food item.

Object detection. Thanks to the tremendous development of deep neural networks [1–4,32], considerable object detection methods [14–18,33–36] based on deep learning have been proposed. Based on whether generating region proposals, these methods can be roughly divided into two categories, *i.e.*, proposal-free methods such as SSD [17], YOLO [18], RetinaNet [35], and proposal-based methods such as Fast/Faster R-CNN [14,15], R-FCN [33]. In general, proposal-free methods have advantages in the detection speed, but proposal-based methods are always superior to proposal-free methods in accuracy. In addition, the more advantages of proposal-based methods lies in: First, by exploiting a divide-and-conquer strategy, the proposal-based framework is more stable and easier to converge. Second, without the complicated data augmentation and training skills, you can still easily achieve state-of-the-art performance. Therefore, we choose the proposal-based method Faster R-CNN as our base framework for food detection. Based on Faster R-CNN, we further propose some improvements both for accuracy and speed. Our improved version is denoted as LiteFCN.

Spatial transform. Convolutional Neural networks (CNNs) are actually lack of ability to be spatially invariant to large transformations of the input data [37,38]. Therefore, early work [19] introduced the *Spatial Transformer* module, which explicitly allows the spatial manipulation of data within the network. But the spatial transformation capabilities are still limited due to the implicit supervision by the final recognition loss. On the contrary, the Adversarial Spatial Transformer Network (ASTN) was proposed to create deformations on the object features and make object recognition by the detector difficult in [39], which enhances the generalization ability of the network. In face detection, Chen et al. [40] utilized the Region Proposal Network (RPN) to predict the facial landmarks of each face proposals, and then warped each face region to a canonical pose for further verification. Similar to [40], we also regress four corner landmarks within food images. The original food images are then transformed by mapping the detected corner landmarks to their canonical positions to better detect the foods.

3. Method

In this section, we first present the system overview of the food detection framework proposed in this paper. Then we describe the design philosophy of the transformer network, including the base network, corner landmarks regression and the spatial transformer layer. Finally, we introduce the lightweight food detection network based on the transformer network.

3.1. Overview

As shown in Fig. 2, the system consists of two sequential components: the RectNet to transform the input image and the LiteFCN to detect foods in the transformed image. The role of RectNet is to predict the four associated corner landmarks in the input image, which may represent a irregular food region, and then perform spatial transformation to the food region according to the predicted landmarks, thus obtaining a canonical and detection-friendly image. In addition, since the prediction of landmarks here is based on the whole image, it is necessary to consider the model complexity for the system. Therefore, we build an efficient Inception-like network to reduce the computation cost and accelerate the inference.

Taking the transformed image as input, a Lite Fully Convolutional Network contained a shared fully convolutional structure, is proposed to better find the foods and output their categories. The main difference between the proposed LiteFCN and the original Faster R-CNN [2] is the architecture of prediction head. The original Faster R-CNN contains a deep RoI-wise subnetwork to improve the accuracy at the cost of low speed. By contrast, our proposed detector use a lightweight prediction head for inference efficiency while keeping the accuracy.

Table 1
An efficient Inception-like network architecture. KSize: kernal size of the convolution or pooling filters. S: the stride of filters. Rep: repeated times. Comp*: the complexity of the networks, that's the number of floating-point operations (FLOPs).

Type	Output size	KSize	S	Rep	Output channels
Image	224×224				
Conv	112×112	3×3	2	1	16
Down-sampling block	56×56		2	1	32
Conv	56×56	1×1	1	1	32
Stage2,Res-Inception	56×56		1	1	64
Conv	56×56	1×1	1	1	64
Pooling	28×28	2×2	2	1	64
Stage3,Res-Inception	28×28		1	2	128
Conv	28×28	1×1	1	1	128
Pooling	14×14	2×2	2	1	128
Stage4,Res-Inception	14×14		1	3	224
Conv	14×14	1×1	1	1	224
Pooling	7×7	2×2	2	1	224
Stage5,Res-Inception	7×7		1	3	320
Conv	7×7	1×1	1	1	320
GAP	1×1	7×7			
FC					1000
Comp*					286M

3.2. RectNet

3.2.1. Network architecture

As mention above, we build an efficient Inception-like network to predict the corner landmarks. As shown in Fig. 3, our network includes two building blocks: the simplified Residual Inception block and the Down-sampling block. The whole network does not contain any special layers, such as Depthwise Convolution [41] and Shuffle operations [42]. Therefore, it is easy to implement and deploy.

A complete description of the details of the network is given in Table 1. The proposed network is mainly composed of a stack of the simplified Residual Inception block grouped into four stages. All convolutional layers are followed by a batchnorm [43] and ReLU nonlinearity with the exception of the final fully connected layer which feeds into the softmax layer for image classification. At the beginning of the network, we adopt the Down-sampling block to reduce the resolution and improve representation capacity without adding too much computation cost. In the middle stage, we first append a 1×1 convolution on the concatenated feature maps and then just use a average pooling layer for down-sampling. We first train our network on ImageNet dataset [44] and achieve 61.2% Top1 accuracy.

Taking the pre-trained Inception-like network as the base network, our RectNet predicts the four corner landmarks. Here, the last fully connected layers (1000-way) has been removed from the Inception-like network and we apply a new 8-way *fc* layer initialized from a zero-mean Gaussian distribution with standard deviation 0.01 for regressing the coordinates. The input size of RectNet is set to 320×240 with RGB channels. After predicting the landmarks, we map the coordinates of the landmarks back to the original image according to the scaling factor.

3.2.2. Corner landmarks regression

Given an image I, we denote the ground-truth corner landmarks in the image I as $K = \{(x_i, y_i) | i = 1, 2, 3, 4\}$. Let $O = (x_c, y_c)$ specify

Please cite this article as: Y. Zhu, X. Zhao and C. Zhao et al., Fooddet: Detecting foods in refrigerator with supervised transformer network, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.10.106

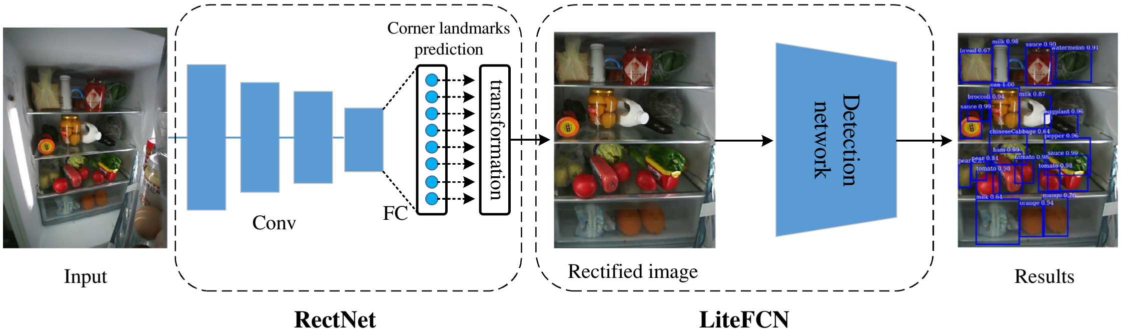

Fig. 2. The system overview of the proposed food detection framework. The system is composed of two components: the RectNet and the LiteFCN. The RectNet outputs four corner landmarks to rectify the input image. Based on the Rectified image, the LiteFCN is proposed to predict the food's categories and locations efficiently.

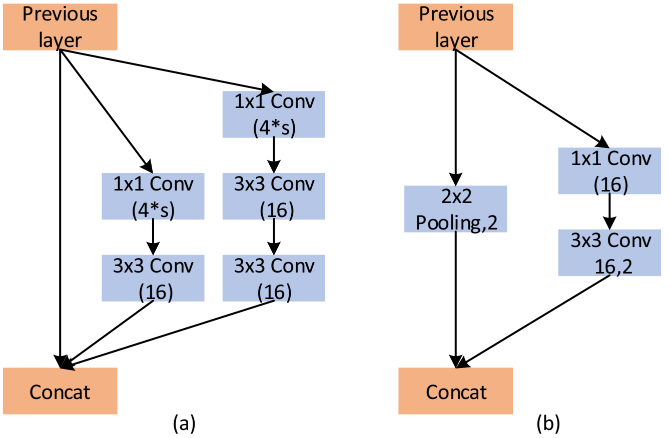

Fig. 3. Main building blocks of the proposed Inception-like network. Each convolutional layer in these building blocks follows a BatchNorm and a ReLU layer. (a): the simplified Residual Inception block (Res-Inception). s means the network stride in the current block. (b): the Down-sampling block.

the pixel coordinates of the center of image I. Instead of regressing the coordinates of landmarks directly, our goal is to learn the normalized offsets between the corner landmarks K and the coordinates of image center O, which does not require significant tuning of learning rate in order to prevent exploding gradients and is easy to converge. Therefore, the regression targets are formulated as follows:

$$t_{x,i} = (x_i - x_c)/w$$
$$t_{y,i} = (y_i - y_c)/h \tag{1}$$

where w and h indicate the width and height of image respectively. Eq. 1 specifies a scale-invariant translation of corner landmarks with respect to the center of image. After learning these translations, we can transform the predicted targets $\hat{t}$ into its corresponding ground-truth corner landmarks by applying the transformation:

$$\hat{x}_i = \hat{t}_{x,i} * w + x_c$$
$$\hat{y}_i = \hat{t}_{y,i} * h + y_c \tag{2}$$

Referring to the bounding box regression in detection task, here we also use smoothed L_1 loss to train the network, since smoothed L_1 is less sensitive to outliers than the L_2 loss. The loss function is defined as:

$$L(t, \hat{t}) = \frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{4} smooth_{L_1}(t_i, \hat{t}_i) \tag{3}$$

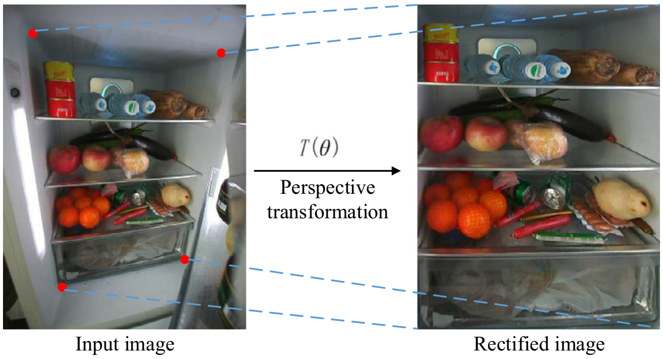

Fig. 4. An example of applying the perspective transformation to the input image I. The red points are the regressed corner landmarks. 4 points on the input image and corresponding points on the rectified image jointly determine the transformation matrix $T(\theta)$. After obtaining the $T(\theta)$, the rectified image can be produced by computing the inverse mapping and applying bilinear interpolation.

in which

$$smooth_{L_1}(\Delta t) = \begin{cases} 0.5\sigma^2\Delta t^2, & |\Delta t| < 1/\sigma^2 \\ |\Delta t| - 0.5/\sigma^2, & otherwise \end{cases} \tag{4}$$

where N is the number of images per batch. σ is the parameter that controls the point where the function will change from quadratic to linear. Here σ is set to the default value 3.

3.2.3. Spatial transformer layer

We propose to learn the corner landmarks in the input image to define the spatial transform and determine the transformed region. Specifically, in this paper, a perspective transformation with a corresponding 3×3 transformation matrix $T(\theta)$ is applied to rectify the images. This can be formulated as:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = T(\theta)\begin{pmatrix} x^s \\ y^s \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{bmatrix}\begin{pmatrix} x^s \\ y^s \\ 1 \end{pmatrix} \tag{5}$$

$$x^d = \frac{x'}{z'} = \frac{\theta_{11}x^s + \theta_{12}y^s + \theta_{13}}{\theta_{31}x^s + \theta_{32}y^s + \theta_{33}} \tag{6}$$

$$y^d = \frac{y'}{z'} = \frac{\theta_{21}x^s + \theta_{22}y^s + \theta_{23}}{\theta_{31}x^s + \theta_{32}y^s + \theta_{33}} \tag{7}$$

where (x^s, y^s) and (x^d, y^d) are the points in the source image and destination image respectively. Therefore, we can obtain the solution of parameters in transformation matrix $T(\theta)$ (θ_{33} is usually set to 1) according to 4 pairs of transformation points shown in Fig. 4, *i.e.*, $(\hat{x}_1, \hat{y}_1) \to (0, 0)$, $(\hat{x}_2, \hat{y}_2) \to (0, w')$, $(\hat{x}_3, \hat{y}_3) \to (h', 0)$

Please cite this article as: Y. Zhu, X. Zhao and C. Zhao et al., Fooddet: Detecting foods in refrigerator with supervised transformer network, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.10.106

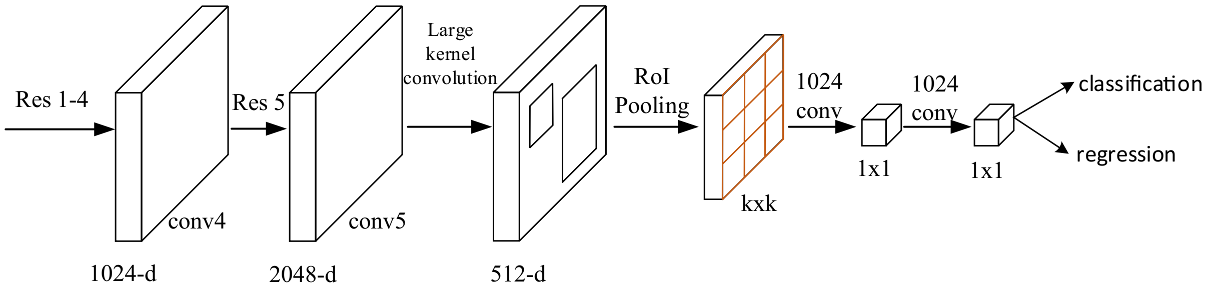

Fig. 5. The architecture of the proposed LiteFCN. The whole network is fully convolutional and it could input images of any size for training and test. Like Faster R-CNN, we also employ the Region Proposal Network (RPN) to generate food proposals. And then the food proposals are fed into the lightweight prediction head by RoI Pooling for further classification and regression.

and $(\hat{x}_4, \hat{y}_4) \rightarrow (h', w')$, w' and h' are the width and height of the rectified image.

In practice, to avoid sampling artifacts, the transformation is done in the reverse order, from destination to the source. That's, each point (x^d, y^d) in the rectified image can be mapped back to the corresponding "donor" point in original image space and copy the pixel value of that point:

$$dst(x^d, y^d) = src(f_x(x^d, y^d), f_y(x^d, y^d)) \tag{8}$$

The $\langle f_x, f_y\rangle$ represents the corresponding inverse mapping $dst \rightarrow src$. And usually $f_x(x^d, y^d)$ and $f_y(x^d, y^d)$ are floating-point numbers, which means an interpolation method (always bilinear interpolation) is used to obtain the pixel value at fractional coordinates.

The spatial transformer layer is placed between the RectNet and detection network. It automatically utilizes the predicted corner positions to adaptively adjust the transformation matrix for each image so that the rectified image can be precisely generated for subsequent food detection and exhibition.

3.3. Food detection network

We basically follow the two-stage method Faster R-CNN [15] for food detection, since Faster R-CNN and its variants [45,46] are the state-of-the-art in the field of object detection. But the good performance is largely benefit from adopting a costly RoI-wise prediction head, such as two 4096-d fully connected layers in VGG16 or the whole Conv5 stage (10 convolutional layers) in ResNet models, which definitely slows down the inference speed especially when the number of proposals is large. Therefore, in order to better balance the accuracy and speed, we introduce some modifications for the original Faster R-CNN and propose a LiteFCN for food detection.

Specifically, we utilize ResNet-101 as our backbone network. Since all the ResNet models share the same topology structure, it is convenient to switch to other ResNet models. For the detection task, we remove the last average pooling layer and the fc layer. Fig. 5 shows the architecture of our food detection network. The network is based on the Faster R-CNN detection network [2] with several improvements. (i) Inspired by the global branches of CoupleNet [47], we insert the RoI Pooling layer to the end of backbone network instead of the Conv4 stage, which means more convolutional layers share computation on the entire image and the feature maps for detection contains more semantic information. (ii) We reduce the stride of backbone from 32 pixels to 16 pixels by modifying the convolution stride in the first block of Conv5 to 1 and using dilation strategy on Conv5 to compensate for the reduced stride. (iii) Referring to the large kernel in semantic segmentation [48], we also apply a large kernel convolution layer between the last block of backbone and RoI Pooling layer to reduce the dimension and increase the receptive field of feature maps, thus saving computation for prediction head while also improving the feature representation ability. Fig. 6 illustrates the large kernel convolution layer, different from [48], the large kernel convolution does not contain category information and we set k to 7 and c_1 to 64. The number of output channels c_2 is 512. (iv) We append two 1024-d convolutional layers with kernel size $k \times k$ and 1×1 respectively (k is set to 7 by default) after RoI Pooling to mimic the RoI-wise subnetwork. Compared to the heavy head (the whole Conv5 stage) in original ResNet-based Faster R-CNN system, the proposed network shares more convolutional computation on entire image and enjoys a lightweight prediction head to evaluate individual proposal. Therefore, the whole network performs the inference efficiently, which runs faster than Faster R-CNN while achieving a slightly higher accuracy.

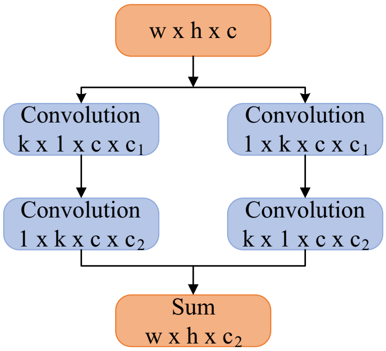
w x h x c
Convolution k x 1 x c x c₁
Convolution 1 x k x c x c₁
Convolution 1 x k x c x c₂
Convolution k x 1 x c x c₂
Sum w x h x c₂

Fig. 6. The architecture of large kernel convolution [48]. Instead of using the common $k \times k$ convolution, this structure employs two sequential $k \times 1$ and $1 \times k$ convolution to reduce the computational complexity.

4. Experiments

In this section, we first describe the food dataset used to evaluate the proposed approach, which consists of images taken from the real refrigerator scene. Then we introduce the evaluation metrics and present the experimental settings and results for the food dataset. Finally, we also evaluate the proposed food detection network on the public PASCAL VOC07 dataset [49] to further validate our method.

4.1. Food dataset

Our food dataset is composed of 49,557 images with 80 categories in total, all of which have been collected from the real refrigerator environment. We divided the images into *Ftrain* and *Ftest* for training and evaluation, which contains 46,276 images with 406,548 annotated objects and 3281 images with 45,901 annotated objects respectively. We strived to ensure that the number of each category is as balanced as possible and that the number of annotated objects is around 10% of total objects when preparing the

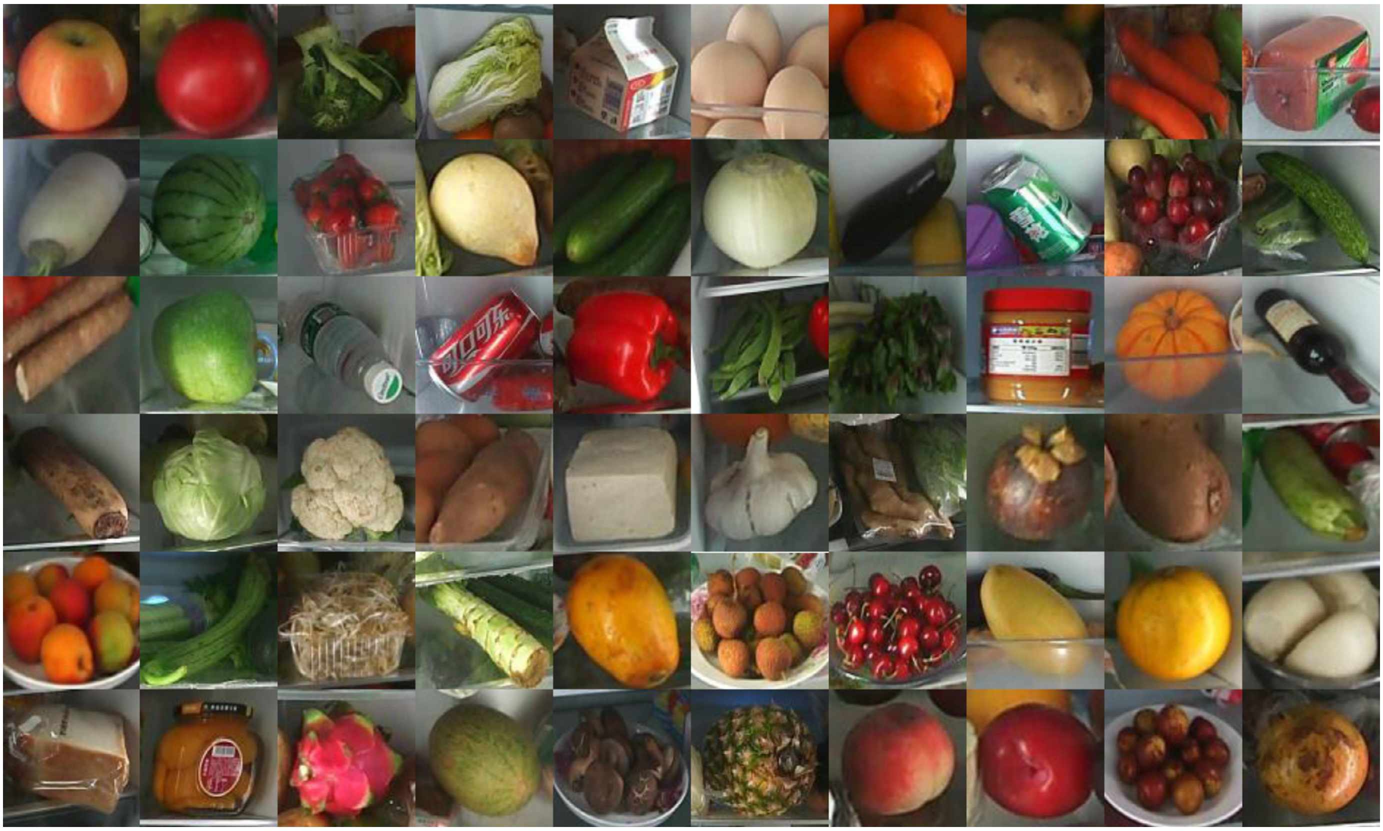

Fig. 7. Examples for 60 out of 80 categories within the proposed food dataset. The categories cover a wide range of foods, such as fruits, vegetables, meat, condiments, drinks, liquors and others. Best viewed in color.

test set. All the images are captured by the cameras located on refrigerator box and door. As a result, the images have a resolution of 1280 × 960 pixels in RGB and present visual deformations and variable backgrounds as aforementioned.

Fig. 7 shows the examples of some categories, the object categories cover a selection of common and everyday foods such as fruits, vegetables, meat, condiments, drinks, liquors and others. Usually, foods can be placed anywhere in the refrigerator, which may result in different amounts of occlusion caused by objects of the same class, objects of different classes, objects in plastic bags or clutter objects. In addition, there are also large intra-class diversities, high inter-class similarities and various scene illuminations. Therefore, food detection in refrigerator is also a challenging task and the above problems are still remaining in this research direction that should be put into concentration for the future research.

4.2. Evaluation metric

In the practical application system, it is necessary to define a score threshold S to facilitate the return of test results. That's we only return the results whose scores are greater than S (S is set to 0.6 in this paper). Therefore, we chose the standard *F-score* measure commonly used in multi-class object detection to evaluate our method:

$$F\text{-}score = (1+\beta^2)\frac{P*R}{\beta^2 P + R} \tag{9}$$

where P and R are the precision and recall respectively under the condition of current threshold S. *F-score* is the harmonic mean of precision and recall. β is the balance parameter to control the importance of recall and precision. Here we set β to 1, which means the recall and precision share the same weight and the evaluation measure becomes *F1-score*.

4.3. Experimental setup

4.3.1. RectNet

We first build and train RectNet based on caffe. Since the learning of corner landmarks is a relatively easy task in the specific refrigerator scene, we randomly selected about 4000 images from *Ftrain* as a training subset *Ftrain-s* for training RectNet in which the box and door images each account for half. Thus we only need to annotate the ground-truth corner landmarks on *Ftrain-s* which effectively reduces the cost of annotation. Using the pretrained Inception-like network as the base network, the RectNet was trained based on 8 Pascal TITAN XP GPUs using synchronized SGD with a weight decay of 0.0001 and momentum of 0.9. Each mini-batch has 32 images per GPU and the input resolution is 320 × 240 pixels. The learning rate is set to 0.001 for the first 15k iterations and 0.0001 for the later 7.5k iterations.

4.3.2. Food detection network

We implemented the proposed food detection network based on the public available code *py-R-FCN*[2]. The input resolution is set to 1280 × 960 pixels. For generating food proposals, we set six scales $\{32^2, 64^2, 128^2, 192^2, 256^2, 320^2\}$ and five aspect ratios {0.33, 0.5, 1.0, 2.0, 3.0} for RPN to cover foods of different sizes and shapes. So there are 30 anchors in total. We train the models with 8 GPUs and the effective mini-batch size thus becomes 16 (2 per GPU). We use a weight decay of 0.0005 and a momentum of 0.9. The learning rate is 0.001 for the first 120k iterations and 0.0001 for the next 24k iterations.

[2] https://github.com/bharatsingh430/py-R-FCN-multiGPU.

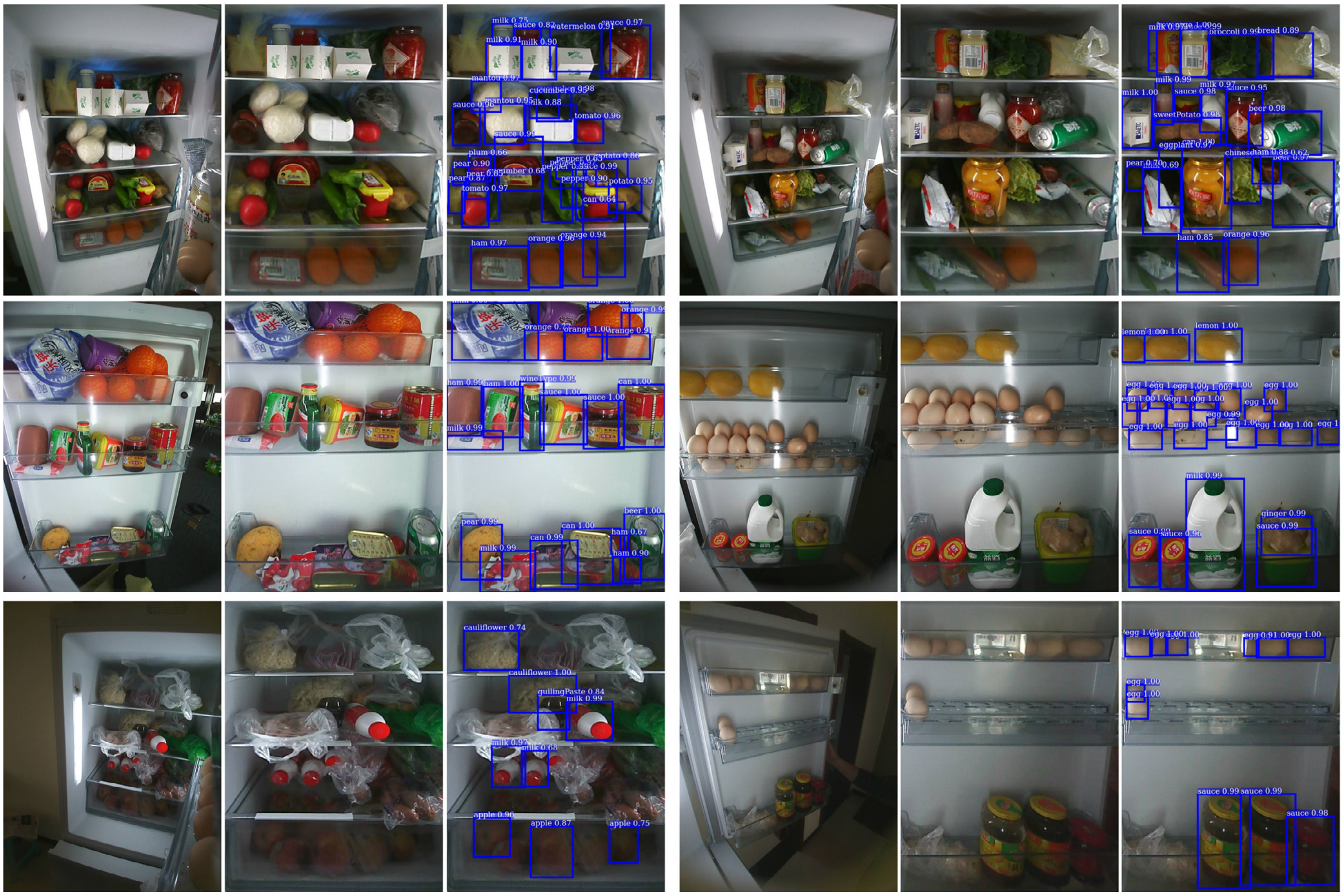

Fig. 8. Examples generated by our method RectNet+LiteFCN (ResNet-101) on *Ftest*. For each triple of images, the left side is the original image, the middle is the rectified image and the right side is the results of our food detection network. All the images are from the real-world environment with occlusions and clutter objects. The first two rows indicate some generic warping of the original images and the last row shows some rare cases with large out-of-plane rotations. A score threshold of 0.6 is used to draw the detection bounding boxes. Best viewed in color.

Table 2
Ablation results on the food test set. FRCN: the original Faster R-CNN. R, P and F1-score are the recall, precision and F1-score averaged over 80 categories. The backbone used here is based on ResNet-18. A score threshold *S* is set to 0.6.

Methods	RectNet	test proposals	R	P	F1-score
FRCN [2]		400	0.597	0.691	0.641
LiteFCN		400	**0.619**	**0.732**	**0.671**
FRCN [2]	yes	400	0.662	0.686	0.674
LiteFCN	yes	400	**0.702**	**0.747**	**0.724**

During the inference, the RectNet and the food detection network are connected by the spatial transformation layer, thus achieving an end-to-end test process.

4.4. Experimental results

We first present the performance of our model on *Ftest*. Qualitative results of our method are shown in Fig. 8. As we can see that the pose of refrigerator varies greatly in the real-world scene, but our RectNet works well even for some large out-of-plane rotations (the last row in Fig. 8) which significantly facilitates the subsequent food recognition and localization.

Some quantitative results are shown in Table 2. In order to verify the effectiveness of our method quickly, we first choose to use a small base network ResNet-18 to conduct the ablation studies. We found that the performance of the food detection models trained solely on the original images are far lower than that of models along with RectNet. Specifically, based on the rectified images, the performance of Faster R-CNN and our LiteFCN were improved by 3.3 percentage points (64.1% *vs.* 67.4%) and 5.3 percentage points (67.1% *vs.* 72.4%) respectively. The main reason is that the RectNet warps each non-frontal food images to a canonical frontal view which makes the food objects more clearer and larger, thus it is more suitable for detection. We further explore the recall of foods with and without the RectNet. As shown in Fig. 9(a), the recall of models based on RectNet are always much higher than those without RectNet when returning different number of food proposals. The same improvements can also be drawed from Fig. 9(b) which compares the recall under different IoU threshold with 400 proposals returned. Note that the recall behaves smoothly when the number of proposals rises from 400 to 600, so we just use as few as 400 proposals for the second stage of detection network. In addition, the proposed LiteFCN also achieves a higher performance compared to the original Faster R-CNN, *i.e.*, improving by 3 percentage points on the original images and 5 percentage points on the rectified images, which also validates the superiority of the proposed detection method.

Moreover, we also compare the proposed detection network with other sophisticated methods. As shown in Table 3, we obtain a *F1-score* of 76.2% and 77.7% while replacing the ResNet-18 with ResNet-50 and ResNet-101. We can observe that our results are even slightly higher than the FPN [45], which combines

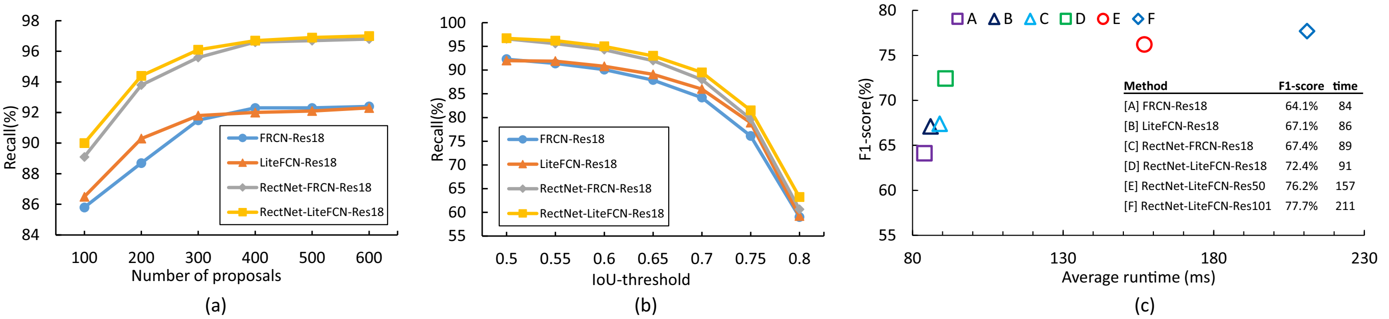

Fig. 9. (a): Recall *vs.* number of food proposals. (b): Recall *vs.* IoU-threshold with 400 proposals returned. (c): Performance *vs.* runtime for evaluated models. The RectNet is nearly cost-free (about 5ms).

Table 3
Comparison with different methods. FRCN: the original Faster R-CNN. R, P and F1-score are the recall, precision and F1-score averaged over 80 categories. 400 proposals are selected for testing.

Methods	Networks	R	P	F1-score
YOLOv2 [50]	Darknet19	0.589	**0.853**	0.697
FPN [45]	ResNet-50	0.759	0.760	0.760
	ResNet-101	0.772	0.762	0.767
FRCN [2]	ResNet-18	0.662	0.686	0.674
LiteFCN	ResNet-18	0.702	0.747	0.724
	ResNet-50	0.765	0.760	0.762
	ResNet-101	**0.780**	0.774	**0.777**

predictions from multiple layers thus enjoying heavy computation complexity. Actually, in the proposed refrigerator scene, it has been able to achieve the corresponding accuracy while just using single-level predictions. In addition, our method is also far superior to YOLOv2 [50], which improves by 8%. Fig. 9(c) shows the speed and accuracy of various evaluated models. All the models are tested under a TITAN XP GPU with CUDA 8.0 and CUDNN-v5.1. Thanks to the high efficiency of Inception-like network, our RectNet is nearly cost-free which only takes about 5ms to rectify an image. We note that our proposed LiteFCN achieves a similar speed with the original Faster R-CNN while using ResNet-18 as the backbone, that's because the RoI-wise subnetwork (the whole Conv5 stage) of ResNet-18 is already computationally efficient. Finally, the whole system achieves a speed of 4.7 fps with ResNet-101 while keeping a relative high accuracy.

4.5. Results on VOC2007

Next, we evaluate the proposed LiteFCN on the Pascal VOC 2007 dataset. The evaluation metric used for this dataset is mean average precision (mAP). Here we use the ResNet-101 as the initialization model to explore the speed and accuracy trade-off. The input resolution is no large than 600 × 1000, that's the shorter side of the input image is resized to 600 pixels and the longer side is restricted to 1000 pixels. The hyper-parameters for training and test are the same as in [2,33,47]. We train the model with 1 GPU and the effective mini-batch size is 2 images by setting the *iter_size* to 2. The whole network is trained for 80k iterations with a learning rate of 0.001 and then for 30k iterations with a learning rate of 0.0001.

Table 4 shows the detailed comparisons with some state-of-the-art methods Faster R-CNN [2], R-FCN [33] and CoupleNet [47]. The Faster R-CNN in [2] was trained by randomly sampling the positive and negative samples and tested under a K40 GPU. For a fair comparison, we first re-implemented Faster R-CNN using ResNet-101 and online hard example mining (OHEM) [51] and tested under a TITAN XP GPU, denoted as FRCN-*ReIm* in Table 4. As we can see that our method achieves a mAP of 81.3%, which outperforms the original Faster R-CNN by 4.9 points, the re-implenented Faster R-CNN by 2.3 points and the R-FCN by 1.8 points, and is only slightly lower than CoupleNet. From the perspective of speed, as shown in the last column of Table 4, our method can perform the inference efficiently, *i.e.*, 95ms for an image, which runs slightly slower than R-FCN but much more faster than Faster R-CNN and CoupleNet. We note that the sharing of more convolutional computation on entire image and the lightweight RoI-wise subnetwork after RoI pooling both reduce the model complexity. Meanwhile, extracting features on deeper convolutional layers and the application of large kernel convolution both enhance the feature representation ability. Therefore, our method achieves the best trade-off between accuracy and speed.

5. Conclusion

In this paper, we present a food detection system in refrigerator to simultaneously address the problem of food images warping and multi-class food detection. Our system is composed of two serial stages. The first stage is a supervised transformer network, named as RectNet, which predicts four corner landmarks to specify a food region and then rectifies the food region to a canonical pose. The learning of corner landmarks is turned into

Table 4
Results on Pascal VOC 2007 test set. The input resolution is about 600 × 1000. For fair comparison, all the methods use ResNet-101 as the base network. “07+12”: VOC07 trainval union with VOC12 trainval. FRCN: the original Faster R-CNN in [2]. *ReIm*: our reimplementation using online hard example mining [51].

Methods	Training data	Test proposals	mAP(%)	GPU	avg. runtime (ms/img)
FRCN [2]	07+12	300	76.4	K40	420
FRCN-*ReIm*	07+12	300	79.0	TITAN XP	130
R-FCN [33]	07+12	300	79.5	TITAN XP	**83**
R-FCN *multi-sc train* [33]	07+12	300	80.5	TITAN XP	**83**
CoupleNet [47]	07+12	300	**81.7**	TITAN XP	122
Ours	07+12	300	81.3	TITAN XP	95

regressing the normalized offsets with the image center to make the network easy to converge, and is supervised by the ground-truth offset targets using a smoothed L_1 loss. Due to the explicit supervision and the flexibility of prediction of landmarks, our RectNet can be insensitive to various spatial variances. The second stage is a optimized multi-class food detection network, named as LiteFCN, which aims to share convolutional computation on the entire image as more as possible and improve the feature representation. As a result, the optimized designs enable us to significantly improve the detection performance without compromising the computational speed. Finally, the RectNet and the LiteFCN are connected by the spatial transformer layer, thus achieving an end-to-end inference. Experimental results demonstrate the effectiveness of the proposed design decisions on both refrigerator food dataset and PASCAL VOC2007 dataset. In addition, the large intra-class diversities and the high inter-class similarities will be investigated in a future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Natural Science Foundation of China under Grants 61772527, 61806200 and 61876086.

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the NIPS, 2012, pp. 1097–1105.
[2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the CVPR, IEEE, 2016, pp. 770–778.
[3] J. Kim, H. Kim, S. Huh, J. Lee, K. Choi, Deep neural networks with weighted spikes, Neurocomputing 311 (2018) 373–386.
[4] lvaro Arcos-Garca, J.A. lvarez Garca, L.M. Soria-Morillo, Evaluation of deep neural networks for traffic sign detection systems, Neurocomputing 316 (2018) 332–344.
[5] S. Horiguchi, S. Amano, M. Ogawa, K. Aizawa, Personalized classifier for food image recognition, IEEE TMM 20 (2018) 2836–2848.
[6] G.M. Farinella, D. Allegra, F. Stanco, A benchmark dataset to study the representation of food images, in: Proceedings of the ECCV, Springer, 2014, pp. 584–599.
[7] G. Waltner, M. Schwarz, S. Ladstätter, A. Weber, P. Luley, M. Lindschinger, I. Schmid, W. Scheitz, H. Bischof, L. Paletta, Personalized dietary self-management using mobile vision-based assistance, in: Proceedings of the ICIAP, Springer, 2017, pp. 385–393.
[8] H. Kagaya, K. Aizawa, M. Ogawa, Food detection and recognition using convolutional neural network, in: Proceedings of the ACM MM, 2014, pp. 1085–1088.
[9] K. Aizawa, Y. Maruyama, H. Li, C. Morikawa, Food balance estimation by using personal dietary tendencies in a multimedia food log., IEEE TMM 15 (8) (2013) 2176–2185.
[10] H. Kagaya, K. Aizawa, Highly accurate food/non-food image classification based on a deep convolutional neural network, in: Proceedings of the ICIAP, Springer, 2015, pp. 350–357.
[11] Y. Matsuda, H. Hoashi, K. Yanai, Recognition of multiple-food images by detecting candidate regions, in: Proceedings of the ICME, IEEE, 2012, pp. 25–30.
[12] W. Shimoda, K. Yanai, Foodness proposal for multiple food detection by training of single food images, in: Proceedings of the 2nd Int. Workshop Multimedia Assisted Dietary Management, ACM, 2016, pp. 13–21.
[13] J. Dehais, M. Anthimopoulos, S. Mougiakakou, Dish detection and segmentation for dietary assessment on smartphones, in: Proceedings of the ICIAP, Springer, 2015, pp. 433–440.
[14] R. Girshick, Fast r-cnn, in: Proceedings of the ICCV, IEEE, 2015, pp. 1440–1448.
[15] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE TPAMI (6) (2017) 1137–1149.
[16] X. Chen, H. Ma, C. Zhu, X. Wang, Z. Zhao, Boundary-aware box refinement for object proposal generation, Neurocomputing 219 (2017) 323–332.
[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: Proceeding so the ECCV, Springer, 2016, pp. 21–37.
[18] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the CVPR, IEEE, 2016, pp. 779–788.
[19] M. Jaderberg, K. Simonyan, A. Zisserman, et al., Spatial transformer networks, in: Proceedings of the NIPS, 2015, pp. 2017–2025.
[20] N. Martinel, G.L. Foresti, C. Micheloni, Wide-slice residual networks for food recognition, in: Proceedings of the WACV, IEEE, 2018, pp. 567–576.
[21] M. Bolaños, A. Ferrà, P. Radeva, Food ingredients recognition through multi-label learning, in: Proceedings of the ICIAP, Springer, 2017, pp. 394–402.
[22] S. Yang, M. Chen, D. Pomerleau, R. Sukthankar, Food recognition using statistics of pairwise local features, in: Proceedings of the CVPR, IEEE, 2010, pp. 2249–2256.
[23] R. Xu, L. Herranz, S. Jiang, S. Wang, X. Song, R. Jain, Geolocalized modeling for dish recognition, IEEE TMM 17 (8) (2015) 1187–1199.
[24] L. Herranz, S. Jiang, R. Xu, Modeling restaurant context for food recognition, IEEE TMM 19 (2) (2017) 430–440.
[25] V. Bettadapura, E. Thomaz, A. Parnami, G.D. Abowd, I. Essa, Leveraging context to support automated food recognition in restaurants, in: Proceedings of the WACV, IEEE, 2015, pp. 580–587.
[26] Y. Kawano, K. Yanai, Automatic expansion of a food image dataset leveraging existing categories with domain adaptation, in: Proceedings of the ECCV, Springer, 2014, pp. 3–17.
[27] L. Bossard, M. Guillaumin, L. Van Gool, Food-101–mining discriminative components with random forests, in: Proceedings of the ECCV, Springer, 2014, pp. 446–461.
[28] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, K.P. Murphy, Im2calories: towards an automated mobile vision food diary, in: Proceedings of the ICCV, IEEE, 2015, pp. 1233–1241.
[29] M. Bolaños, P. Radeva, Simultaneous food localization and recognition, in: Proceedings of the ICPR, IEEE, 2016, pp. 3140–3145.
[30] W. Shimoda, K. Yanai, Cnn-based food image segmentation without pixel-wise annotation, in: Proceedings of the ICIAP, Springer, 2015, pp. 449–457.
[31] R. Adams, L. Bischof, Seeded region growing, IEEE TPAMI 16 (6) (1994) 641–647.
[32] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[33] Y. Li, K. He, J. Sun, et al., R-fcn: object detection via region-based fully convolutional networks, in: Proceedings of the NIPS, 2016, pp. 379–387.
[34] Y. Zhu, J. Wang, C. Zhao, H. Guo, H. Lu, Scale-adaptive deconvolutional regression network for pedestrian detection, in: Proceedings of the ACCV, Springer, 2016, pp. 416–430.
[35] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: Proceedings of the ICCV, IEEE, 2017, pp. 2999–3007.
[36] Q. Hu, P. Wang, C. Shen, A. van den Hengel, F. Porikli, Pushing the limits of deep cnns for pedestrian detection, IEEE TCSVT 28 (6) (2018) 1358–1368.
[37] K. Lenc, A. Vedaldi, Understanding image representations by measuring their equivariance and equivalence, in: Proceedings of the CVPR, IEEE, 2015, pp. 991–999.
[38] T.S. Cohen, M. Welling, Transformation properties of learned visual representations, in: Proceedings of the ICLR, 2014.
[39] X. Wang, A. Shrivastava, A. Gupta, A-fast-rcnn: hard positive generation via adversary for object detection, in: Proceedings of the CVPR, IEEE, 2017, pp. 3039–3048.
[40] D. Chen, G. Hua, F. Wen, J. Sun, Supervised transformer network for efficient face detection, in: Proceedings of the ECCV, Springer, 2016, pp. 122–138.
[41] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications, CoRR abs/1704.04861 arXiv:1704.04861 (2017).
[42] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: an extremely efficient convolutional neural network for mobile devices, in: Proceedings of the CVPR, IEEE, 2018.
[43] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the ICML, 2015, pp. 448–456.
[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of the CVPR, IEEE, 2009, pp. 248–255.
[45] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the CVPR, IEEE, 2017, pp. 2117–2125.
[46] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the ICCV, IEEE, 2017, pp. 2980–2988.
[47] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, H. Lu, Couplenet: coupling global structure with local parts for object detection, in: Proceedings of the ICCV, IEEE, 2017, pp. 4146–4154.
[48] C. Peng, X. Zhang, G. Yu, G. Luo, J. Sun, Large kernel matters improve semantic segmentation by global convolutional network, in: Proceedings of the CVPR, IEEE, 2017, pp. 1743–1751.
[49] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, IJCV 88 (2) (2010) 303–338.
[50] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the CVPR, IEEE, 2017, pp. 7263–7271.
[51] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: Proceedings of the CVPR, IEEE, 2016, pp. 761–769.

Yousong Zhu received the B.E. degree from Central South University in 2014 and the Ph.D. degree in pattern recognition and intelligence systems from the Institute of Automation, Chinese Academy of Sciences and University of Chinese Academy of Sciences in 2019. He is currently an assistant researcher in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research interests include object detection, video object detection, pattern recognition and machine learning, and intelligent video surveillance.

Xu Zhao received the B.E. degree in 2014 from Dalian University of Technology and the Ph.D. degree in pattern recognition and intelligence systems from the Institute of Automation, Chinese Academy of Sciences and University of Chinese Academy of Sciences in 2019. He is currently an assistant researcher in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include object detection, scene text detection, image and video processing, and intelligent video surveillance.

Chaoyang Zhao received the B.E. degree and the M.S. degree in 2009 and 2012 respectively from University of Electronic Science and Technology of China. He received the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, in 2016. He is currently an Assistant Professor in National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include object detection, image and video processing and intelligent video surveillance.

Jinqiao Wang received the B.E. degree in 2001 from Hebei University of Technology, China, and the M.S. degree in 2004 from Tianjin University, China. He received the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, in 2008. He is currently a Professor with Chinese Academy of Sciences. His research interests include pattern recognition and machine learning, image and video processing, mobile multimedia, and intelligent video surveillance.

Hanqing Lu received his B.E. degree in 1982 and his M.E. degree in 1985 from Harbin Institute of Technology, and Ph.D. degree in 1992 from Huazhong University of Sciences and Technology. Currently, he is a deputy director of National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include image and video analysis, medical image processing, object recognition, etc.