Food Recognition and Dietary Assessment for Healthcare System at Mobile Device End Using Mask R-CNN

Hui Ye[1] and Qiming Zou[2(✉)]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[2] Computing Center, Shanghai University, Shanghai 200444, China
kim@shu.edu.cn

Abstract. Monitoring and estimation of food intake is of great significance to health-related research, such as obesity management. Traditional dietary records are performed in manual way. These methods are of low efficiency and a waste of labor, which are highly dependent on human interaction. In recent years, some researches have made progress in the estimation of food intake by using the computer vision technology. However, the recognition results of these researches are usually for the whole food object in the image, and the accuracy is not high. In terms of this problem, we provide a method to the food smart recognition and automatic dietary assessment on the mobile device. First, the food image is processed by MASK R-CNN which is more efficient than traditional methods. And more accurate recognition, classification and segmentation results of the multiple food items are output. Second, the OpenCV is used to display the food category and the corresponding food information of unit volume on the recognition page. Finally, in order to facilitate daily use, TensorFlow Lite is used to process the model to transplant to the mobile device, which can help to monitor people's dietary intake.

Keywords: Food image processing · Dietary monitoring · Mobile terminal recognition

1 Introduction

An unhealthy diet is one of the critical reasons for health problems. Obesity, diabetes, and other chronic diseases are all caused by unhealthy eating habits. According to the WHO [1], in 2016, more than 1.9 billion adults, 18 years and older, were overweight. Of these over 650 million were obese. One of the most important causes of this phenomenon is that many people have unhealthy lifestyle and poor eating habits, such as excessive intake of high-calorie and high-fat foods. Most people are reluctant to take the time to estimate their food intake due to the tedious methods which lack of real-time feedback. Traditional dietary methods require people to record intake manually, and subjective estimates limit the accuracy. So fewer people know their daily intake.

The development of technology provides iOS and Android systems with sufficient computing power to perform real-time image recognition. More and more applications

H. Gao et al. (Eds.): TridentCom 2019, LNICST 309, pp. 18–35, 2020.
https://doi.org/10.1007/978-3-030-43215-7_2

are developed and widely used under these systems. The spread of social platforms such as Facebook, Instagram, Weibo, and WeChat have enabled people to develop the habit of taking pictures and sharing before eating the food. This paper is also based on the increasing property of mobile device, and the growing reliance of people on them. After obtaining the food image through the mobile device, the information contained in the food image can be extracted to provide a useful reference for people's diet.

Image processing is a hot research direction because of the rapid development of AI. In recent years, it has made some progress in the field of automatic bounding-box object detection, recognition, and classification of food items from images by PC. Some researchers have used CNN to recognize and classify food and made the results more and more accurate. However, the results of these research methods show that only one food item for the whole image was recognized and there's no precise segmentation of multiple food items of the image. Some studies have already developed health recognition and management applications on mobile devices, but they still require users' involvement, such as framing the location of the food items manually, which degrades the user experience. Besides, the processing of food images is a fine-grained image processing problem. The existing methods for food image recognition generally use a single CNN, which is relatively troublesome in dealing with multiple kinds of classification problems. So,its accuracy is limited.

The paper is to achieve accurate segmentation, recognition, and classification of multiple food items of one image. The integrated network can provide more accurate results while considering the demand, such as Mask R-CNN [2] which is well known for its instance segmentation. However, it has not been applied to food images in existing researches to complete the recognition and classification of food items. Based on the above research status, this paper proposes a real-time food intake monitoring system using modern mobile devices with more powerful computing power and advanced target recognition technology, which offer a feasible method to help people manage their health. It can capture food image which will be classified in real-time by mobile devices, then display the type of food items and the intake information of per unit volume on the image.

The rest of the paper is organized as follows. Section 2 introduces the related work, and Sect. 3 gives an overview of the proposed system and details the implementation method. Section 4 introduces the experiment and results, and Sect. 5 summarizes the paper.

2 Related Work

2.1 Food Recognition

For food recognition, Parrish et al. [3] may be the first to use computer vision technology for food analysis tasks back to 1977. In 2010, Yang et al. [4] proposed a method using pair-wise local feature statistics for single-category food classification. They divided the image soft pixel level into eight components with labels such as bread and beef. To understand these spatial relationships, pair-wise local features based on distance and orientation between ingredients can be used. A multidimensional histogram

can be used to represent how these features are distributed. SVM is further used as a classifier. Matsuda et al. [5] in 2012 proposed a method to recognize multiple food items in two steps by detecting candidate regions. The first step was to detect candidate regions by the circular detector and JSEG region segmentation. Then, image features such as food texture, gradient, color, etc. in the candidate area were extracted, and multi-core learning was used for recognition to train the model. Martinel et al. [6] proposed a wide-band residual network for food identification in 2018. They thought that the food has a vertical food layer regardless of the form of cooking, and they learn from residuals to achieve food recognition. For multi-food items recognition, He et al. [7] proposed a multi-core SVM method for multi-food images in 2016, which is a food recognition system based on a combination of part model and texture model. Part-based model is a common method for rigid target detection and classification. Considering the differences in food appearance and texture, the authors chose STF texture filters for integration into component-based detectors. Mask R-CNN is an integrated network that can recognize and classify multiple targets in an image. In this paper, Mask R-CNN is used to achieve accurate segmentation through the branch network which can generate mask of multiple food items.

2.2 Food Intake Monitoring

Dietary assessments or diet logs provide valuable reference for preventing many diseases. Most traditional methods rely on questionnaires or self-reports. These methods may cause inaccurate results due to subjective judgments, such as underreporting or miscalculating food consumption. With the development of computer technology, more and more methods are adopted based on these more efficient diet health management methods.

The first method to estimate food intake through digital photography was proposed by Williamson et al. [8] in 2003. They used a digital camera in a cafeteria to record food choices and post-meal residues to measure food intake. The registered dietitian analyzes the image based on the USDA data and enters the estimating size of the food into the computer application that has been designed to calculate the food's metrics. In 2011, Noronha et al. [9] launched a diet system platform for analyzing nutrients in food images, but it is still a semi-automatic method that requires the participation of registered dietitians.

To automate the monitoring, Zhu et al. [10] proposed a technology-assisted diet estimation system in 2008 to evaluate the type and consumption of food by obtaining food images before and after eating. There are also some similar approaches which estimate the amount of food for scene reconstruction and multi-view reconstruction. In recent years, more and more diet-related researches have provided nutrition-related information. Some foods are also labeled with relevant nutrients to help people get a healthier eating habits. The method proposed in this paper is to take the food image taken by the user before meal. The analysis output is carried out by the backend server, the food items in the image are automatically segmented, recognized and classified, and the nutritional information of per unit volume is displayed.

2.3 Food Recognition on Mobile Devices

Due to the increase in diet-related diseases such as obesity, many simple healthy diet applications have been developed, from manual monitoring of dietary activity analysis to the transmission of monitored information to web applications for immediate analysis.

To make it easier to monitor what people eat, Joutou et al. [11], in 2009, were the first to propose an algorithm for monitoring diet on mobile devices. They introduced multi-core learning into food recognition and then classified food images according to feature weights. In the same year, Puri et al. [10] proposed a system for food recognition and volume estimation, realizing food recognition by acquiring image data and voice data through mobile devices. However, this system only collects image and voice information by mobile devices, and the operation of image processing is still carried out by the server. Different from the above method, in 2012, Kong et al. [12] only implemented two functions according to the image. Users capture three pictures from different angles or surrounding videos of food through the mobile device, and then the obtained image information will be sent in XML format to the server for the remaining processing operations.

In 2013, Kawano and Yanai et al. [13] developed a lightweight food recognition mobile application for the first time. After the user pointed the camera at the food, manually framed the detection area, segmented the image using GrubCut, extracted the color histogram, and identified it based on the SURF feature pack. They continued to optimize this project in 2015 [14]. Pouladzadeh et al. [15] proposed a method similar to Kong's. Users need to record two photos of food from the top and side, extract each segment of food image by k-means clustering and texture segmentation, extract color features by edge detection and k-means, and extract texture features by Gabor filtering.

3 System Structure and Method

The ultimate goal of the system is to support users to capture types and nutrition information of food in real-time through general mobile devices before the meal which can help users to monitor their diet and manage their health.

The overall structure of the system can be seen in Fig. 1. The food image, which is taken by the mobile camera or existing already, is input into the mobile device. After the food image is pre-processed to a uniform size by the server of the system, the feature map of the whole image is extracted by five shared convolution layers. The network outputs five feature map for the entire image and FPN merges the feature maps to generate five fusion feature layers for subsequent model training and region generation. The RPN layer will operate on the bottom feature map to generate various anchors for each pixel after feature extraction of the input image. And also, the remaining feature layers are input into the sibling convolutional network for rough classification and rough bounding-box object detection to output a certain amount of optimized RoIs, namely the target item to be detected in the food image. The RoI Align operation is performed on the screened candidate boxes so as to match the feature maps of each food target with the original images and RoIs more accurately. The last three

branches of the network implement multi-category classification of these RoIs, and perform border regression again for fine bounding-box object detection and mask generation. In terms of displaying food information, types of food items and corresponding nutritional information such as calorie, which is obtained from the BooheeNet [16], are stored in a food information database in advance. The food types obtained from the classification process are used to search in the food information database and get the corresponding food information. Finally, the classification results of food items and the obtained food information are sent to the client for display.

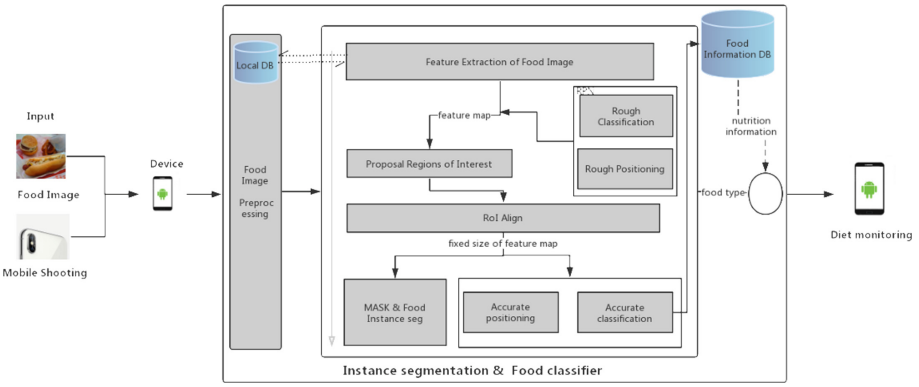


Fig. 1. System architecture diagram.

The network structure (see in Fig. 2) is used to train the model of food image processing. Corresponding food information is obtained from the food database through the classification results, which will be displayed to the users by OpenCV and PIL method at the same time. For the convenience of daily use, TensorFlow Lite is used to transplant the pc-trained model to the mobile terminal. Next section will detail the main approaches to implementing this process.

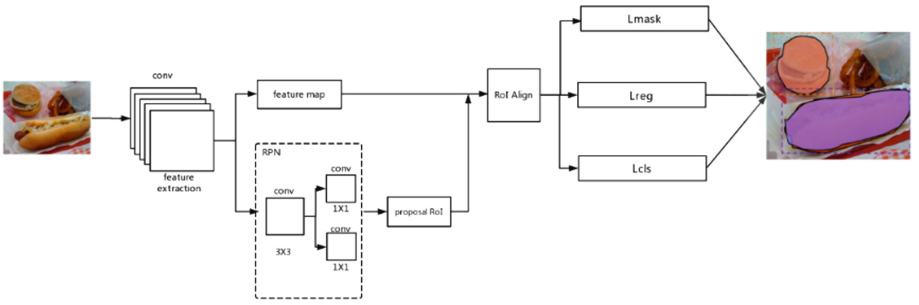


Fig. 2. Network model diagram.

3.1 Feature Extraction and Target Detection

In this paper, we adopt ResNet50+FPN as the backbone network to improve recognition accuracy. The food image is pre-processed into the size of 1024 * 1024 and extracted feature maps of the whole picture through five shared convolution layers. Here we divide ResNet50 into five stages to output five graphs that would be used next in the FPN network. FPN network is used to make better fusion feature map. Normal networks are directly using feature map output by the lowest layer because of its strong semantics. However, it is not easy to detect small items because of low location and resolution. So, five-layer fusion feature maps are carried out after that FPN deals with the five-layer feature maps output by the ResNet through reverse bounding-box object detection procession.

The lowest feature layer is used to generate nine different sized anchor boxes for each pixel in the image, and use a small network as a sliding window to detect item on the feature map generated before. For each point, the network generates nine anchors of different three areas and three aspect ratios, and get the coordinate values. The specific coordinates of each anchor box can be calculated as long as knowing the coordinates of the sliding window. In order to generate anchors for a pixel point, a base anchor is used to determine constant area at first and make the length-width ratios as 1:1, 1:2 and 2:1 to obtain three anchors. Figure 3 shows that nine anchors can be generated for each pixel with three areas and three ratios.

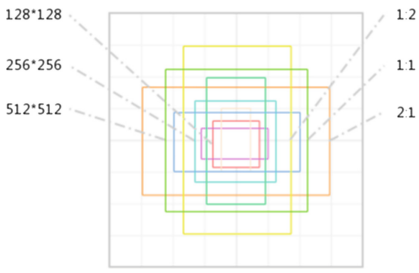


Fig. 3. Nine anchor boxes with different three aspect ratios and three areas.

In the part of target detection, the remaining feature maps output from top to bottom is used to train RPN. The trained RPN realizes the simple bounding-box object detection and classification of target items. The extracted food image feature maps are input into the 512d convolution layer for processing firstly and then input to two 1 * 1 sibling convolution layers which one is for classification and another is for bounding-box object detection.

To achieve simple classification, the 256-dimensional features obtained before are input into the 1 * 1 cls layer. The softmax layer of cls layer is used for each anchor box to assign a binary degree to distinguish the foreground and background. There're two types of anchor boxes, they are assigned positive labels as the foreground. One is the anchor box whose IoU overlaps with any ground truth (GT) box more than 0.7. And the another is the anchor box with the highest IoU overlaps with a GT box (maybe less

than 0.7). Therefore, a GT box can correspond to multiple anchors of positive labels as is shown in Fig. 4. Besides, negative labels are assigned to the anchors whose IoU with GT boxes are less than 0.3 as the background. In order to reduce redundancy, the anchor boxes with positive and negative labels are output and the remaining anchors are removed. At this time, the output is 18 (2 * 9) dimensions in total. After obtaining nine anchors for each pixel point, the cls layer outputs 2k scores for each region to estimate the probability if it's the target item. The scores will be used next.

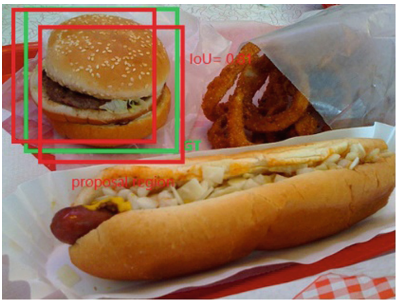


Fig. 4. Comparison example of anchor box and GT.

As for achieving simple bounding-box object detection, the anchors of target items are input to the 1 * 1 reg layer for fine-tuning in the image. Each anchor box has $[x, y, w, h]$ 4 values corresponding to 4 offsets related to bounding-box object detection and size, where (x, y) is the coordinate of the center point and (w, h) is the width and height. As mentioned above, the *GT* boxes are set to help detect the target items in the food image. However, the foreground anchors may have a great deviation from the *GT*. So, it is necessary to fine-tune the generated foreground anchor boxes so that they're more closely with the GT boxes. For each anchor, smooth loss is adopted for regression correction. The foreground anchor boxes are carried out before, and then translation parameter (d_x, d_y) and scaling parameter (d_w, d_h) are regression corrected to make the original anchor box A adjust to be GT' which is closer to GT. Namely for anchor box $A[A_x, A_y, A_w, A_h]$, there is a mapping f, make $f(A_x, A_y, A_w, A_h) = (GT'_x, GT'_y, GT'_w, GT'_h)$, where $(GT'_x, GT'_y, GT'_w, GT'_h) \approx (GT_x, GT_y, GT_w, GT_h)$. The output value is not the absolute coordinate of the anchor, but the offset correction relative to the GT.

Of course, for the output anchors to be detected, it is also necessary to determine whether exceeds the, and remove which is exceeded seriously. The scores obtained by the softmax loss before are sorted from large to small, and the top 2000 are extracted. Then, the NMS operation is performed on the top 2000 and redundancy is removed to obtain the anchors which are the local maximum values. At last, 300 anchors to be detected are output after being sorted again, and the area to be tested is preliminarily determined. Since workload for training all the anchors is relatively large, the program selects 128 positive label anchors and 128 negative label anchors randomly for training among the appropriate anchors during the training process.

3.2 Food Identification and Classification

Food recognition is a challenging task because that even if it is the same type of food, their shapes maybe vary depending on the way they are cooked, the eating habits of different countries or regions and the way they are dished up. The factors of different lighting conditions and shooting angles when taking food pictures can also lead to the challenge of food recognition. Besides, the integrated network such as the Mask R - CNN has not yet to be directly used in existing research of food processing. On the one hand, there're external factors above-mentioned may cause some recognition difficulties for food image. On the other hand, although the integrated network has advantages of good effect and high precision, their structures are more complicated than the simple classification and identification network. Under the calculation condition of mobile device support a few years ago, the performance of application on food recognition cannot keep up.

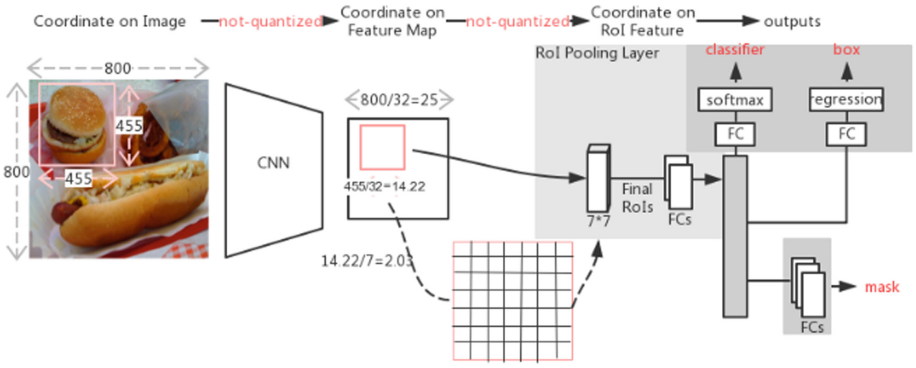


Fig. 5. Food identification, recognition and segmentation process.

In this paper, after filtering out some useless candidate RoIs, the process of image processing can be seen in Fig. 5. In order to consistently maintain a clear single-pixel spatial relationship, RoI Align operation is performed on the remaining candidate RoIs. In order to match the original image with the feature map obtained earlier and to match the feature map with the fixed size RoI, the RoI size in the feature space can be corrected by two quantization operations, and the floating-point pixel value appearing during the conversion process can be quantized (rounded). However, due to the high probability of rounding errors, the two quantized errors of images and features may have a large impact on the final matching. To avoid the errors caused by quantization, the "bilinear interpolation" method is used to solve the floating-point pixel values in the transform and make the RoI of the feature space corresponding to the original image more precisely.

The so-called "bilinear interpolation" method uses the four existing real pixel values around the virtual point to jointly determine a pixel value in the target image (see in Fig. 6). After adopting this method, the deviation caused by rounding can be well avoided. It is not necessary to quantize the RoI for the first time, but map to the feature map to divide the bin of 7 * 7 directly and accurately. The bilinear interpolation is performed on each bin to obtain four points, and then perform max pooling after

inserting the value to get the final 7 * 7 RoI. That means the process of RoI Align has completed. The RoIs in the feature space are processed by RoI Align to output RoIs matching the original image, the full feature map, and the RoI feature.

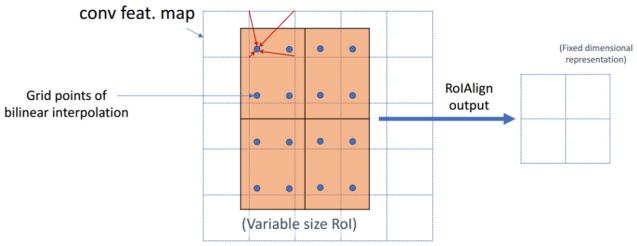


Fig. 6. RoI Align implementation.

Due to the addition of a mask branch, the loss function for each RoI is (1).

$$L = L_{cls} + L_{reg} + L_{mask} \tag{1}$$

After the final RoIs are obtained, the processing outputs of the full connection layer are input into the full connection layers for classification, bounding-box object detection and the generation of masks. Because that classification process is earlier than segmentation, it is only necessary to split it semantically if each RoI corresponds to only one category, which is equivalent to instance segmentation. Although for each RoI, the mask branch can complete the k-category classification, we hope that the segmentation and classification are implemented separately, so only take the foreground and the background as a two-category semantic segmentation. The other two branches are the final exact classification and bounding-box object detection.

3.3 Food Index Coefficient

The final purpose of this paper is to transplant the research contents above to mobile devices and add food-related information to monitor the users' diet. The local food database, on the one hand, can improve the recognition efficiency and reduce the pressure on the server. On the other hand, the food which is not stored locally will be saved in the same mode to the database, which can expand the variety of food and provide more data foundation for future identification.

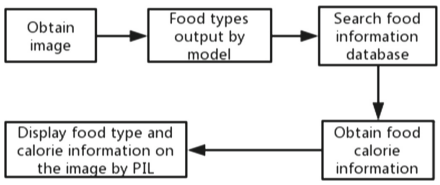


Fig. 7. Process of displaying the final output results.

After the PC-side model is trained, the recognition result and the corresponding food information need to be present at the same time to improve the users' use. After getting an image of an approximate video stream of food, then use the OpenCV library, which is provided in Python, to capture real-time food images and process on them. Figure 7 shows the process of displaying the final output results. The results of the model recognition are used to search in the database to obtain the corresponding food information, and call the PIL library to display the obtained food information and the recognition result together in the food image.

4 Food Recognition

4.1 Model Training of PC End

All ten categories of food in the COCO2017 [17] data set are prepared for the experiments in this paper, each category with a different quantity. Some categories do not have independent images but are labeled in other images. In the experiment, 8k images are selected for training and 1k are for verification. Through training, we hope to achieve bounding-box object detection, segmentation for one or more types of foods present in the image and classify them accurately.

In the experiments, ResNet50+FPN is selected as the basic network for the sake of accuracy. Of course, considering the image processing work on the mobile terminal, the lighter network YOLO will be used as the backbone network in the future work, which can improve the recognition efficiency and provide relatively more flexible model.

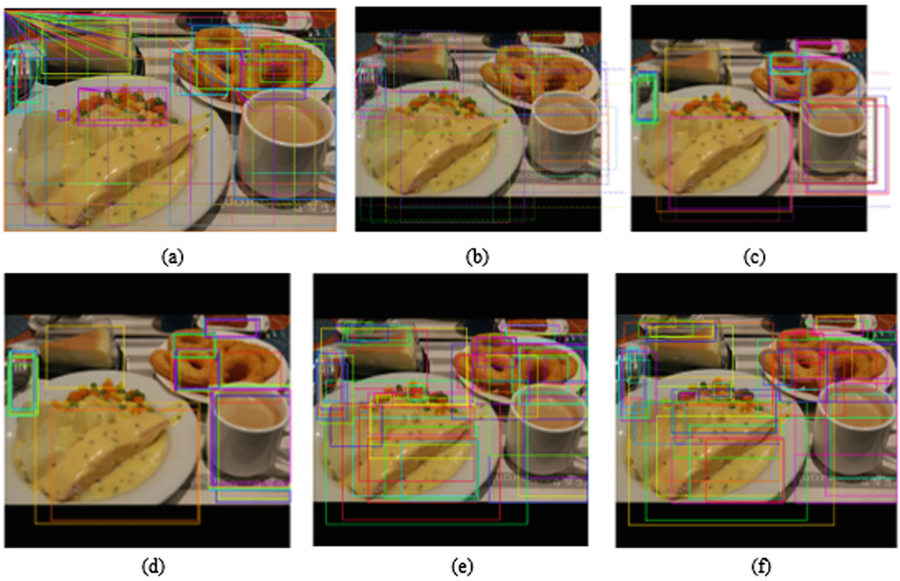
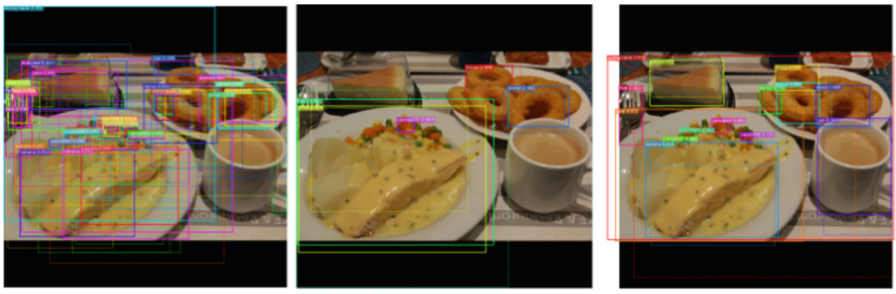


Fig. 8. The output of bounding-box object detection process. (a) Anchors for each pixel. (b) Anchors after filtering according to IoU. (c) (d) Refine some anchors out of bounds. (e) Anchors after NMS operation. (f) Final output result.

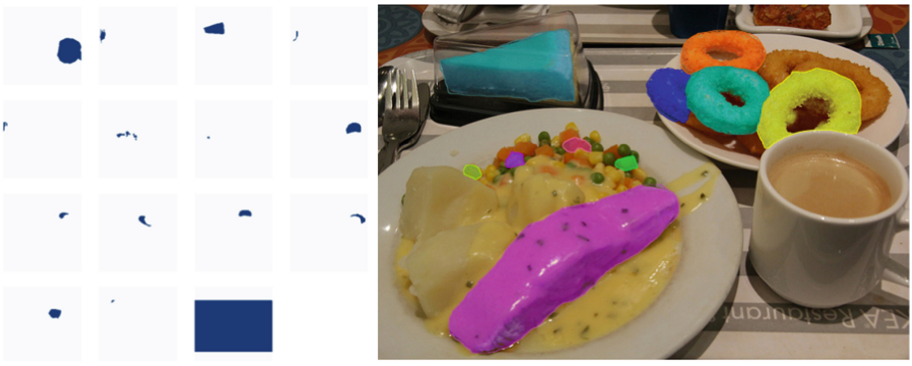
After extracting the feature map of entire image, the bounding-box object detection process is output at first. Candidate regions are generated for each pixel point by the RPN. Figure 8(a) shows the anchors of different proportions produced by the RPN. Each point generates nine candidate anchors of different areas and aspect ratios. Then the IoU of the generated candidate region with the GT is calculated to perform further screening to obtain more accurate anchors, which aim to achieve coarse bounding-box object detection of the food items in the image. Figure 8 shows the process and final results of bounding-box object detection by generating bounding boxes.

It is known from the introduction of experiment methods in Sect. 3 that the bounding-box object detection and classification are carried out through two parallel networks in the RPN. When classifying the food items in the image, the initial results are very confusing. Therefore, as shown in Fig. 9, the NMS method is also used to correct the outputs after removing all which are out-of-boundary or non-positive and negative labels to eliminate the redundancy. Finally, more accurate classification results are output.



(a) (b) (c)

Fig. 9. The output of the classification process. (a) The result of classification before refinement. (b) The result of classification after refinement. (c) The final result of classification after NMS.



(a) (b)

Fig. 10. The output of the segmentation process. (a) Mask target of different food items. (b) Mask result of each food item.

Except bounding-box object detection and classification, it's also necessary to output a binary mask corresponding to each final RoI, which can segment multiple food items of the image. The mask targets and results corresponding to RoI are shown in Fig. 10.

After bounding-box object detection, segmentation, and classification are completed, it's necessary to display categories and corresponding food nutrition information at the same time on the image. The nutrition information of 10 kinds of food had stored in advance in the food information database, and adopt the methods mentioned in Sect. 3.3 for the experiment. Figure 11 shows the final training output on the PC that food-related information was presented with the categories on the food image at the same time.

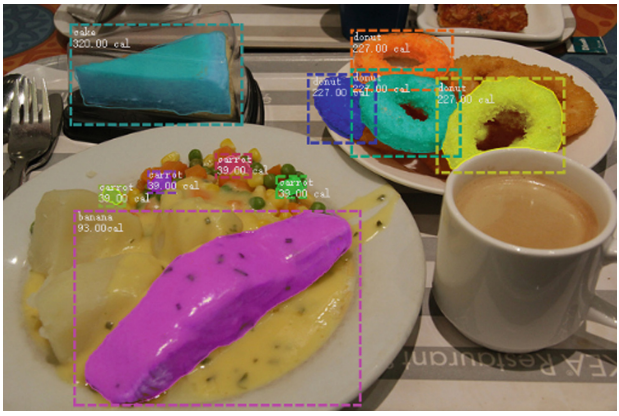


Fig. 11. The final result of processing food image on the PC-end.

4.2 Food Recognition on Mobile End

AI applications for mobile devices typically have features such as no network latency, more timely response, and higher data privacy. After training the model on the PC side, it's time to consider how to transplant it to the mobile terminal and implement the same functions on the mobile side as the PC. TensorFlow Lite is a cross-platform lightweight solution released by Google that runs machine learning on mobile devices. It can help transplant the PC models trained by TensorFlow to mobile and embedded devices, which support multiple platforms including Android, iOS, etc. Because there is no cellular network delay with minimal runtime library, the application can run more smoothly, and its high portability supports not only PC-side but also mobile devices operation. It's useful for implementing the same AI-related functions on the PC side and the mobile terminal.

The model trained by TensorFlow on the PC side cannot be directly used by the mobile terminal because of incompatible formats with each other. In this paper, after the process of training model on the PC side is completed, the generated .pb model file of saving constant and the .ckpt model file of saving variables are format converted. At

first, the two model files are frozen into a .pb graph file, and then quantized the frozen file by the toco tool mentioned above to generate a callable lite model file. the model file, which is generated before, can be directly called when developing system through the Java interface provided in Android studio. The TensorFlow Lite provides both C++ and JAVA APIs, but the transplant process is the same regardless of the API's type for that the task to be done is to load the model and then run the model. In the experiments, the Interpreter class of JAVA API is used to complete the process of loading and running. The following (see in Fig. 12) is the recognition results carried out by using the virtual device in Android Studio. It's known that the system can process and output all food item of the image. Not only display the results on the image, but also show in the text output box below the image. If there are food items of the same type in one image, each processing result will be output on the image while the bottom text will only output once.

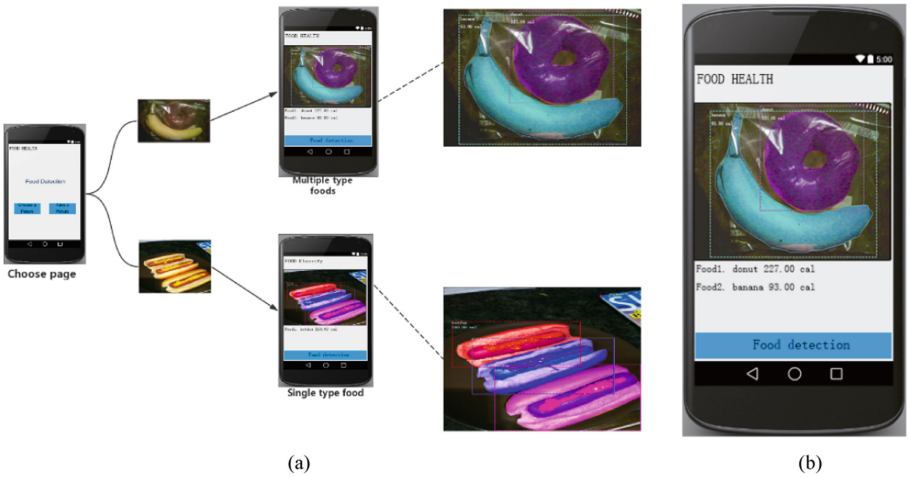


Fig. 12. (a) Food detection results at the mobile terminal simulator. (b) Main process result screen of the proposed system.

5 Experiment

In this section, we describe experimental results regarding recognition accuracy and spending time. In the experiments, we have selected ten-category food dataset randomly from the training dataset and validation dataset of the COCO2017, which has more than 100 testing images per category that have been pre-processed into standard size already. The total number of food images in the dataset is 2000. Appropriate number of food image dataset have been used to calculate the classification error rate for each food item separately and the recognition accuracy of four plate situations.

5.1 Evaluation on Recognition Accuracy

The food classifier's accuracy is affected by many factors, such as fault segmentations, failing classification to a different shaped food type, misinterpretations between similar shaped food types, and food information missing in database. Therefore, the proposed method was tested with different types of test cases to cover as many situations as possible. Then, the overall accuracy was estimated.

Firstly, we evaluate recognition accuracy by ten types of food items as ten test cases. For a certain food, 100 image data are input and get the classification results at first, and then the classification error rate is calculated by function (2). TN is the number of positive classes classified as positive classes while FN is the number of positive classes classified as negative classes. As shown in Table 1, the rates of food items like sandwich, which is closer to a specific shape, is much higher than the rates while items with a variable shape like broccoli. The system has achieved a reasonable error of about 3.82% on average.

$$P_{error} = 1 - (TN/(TN + FN)) \tag{2}$$

Table 1. Classification error rates of ten types of food items.

No.	Food items	Error rate (%)	No.	Food items	Error rate (%)
1	Apple	2.37	6	Carrot	0.63
2	Banana	2.12	7	Hot dog	2.33
3	Sandwich	0.23	8	Pizza	7.91
4	Orange	4.45	9	Donut	1.57
5	Broccoli	9.86	10	Cake	6.78

After classification error rate of different types of food items was tested, the food image dataset is divided into four test situations of fast food, fruits, salad and dessert, which are include 400 for each plate situation removing mix-types images and redundance. Each situation is divided into 5 categories according to the amount of food in an image, and each category has 80 food images. Figure 13 shows that the average lowest classification accuracy rate of each plate situation which is with the same number of food items. It is clear that, when the number of food items increases of plate, the accuracy rate will drop. Besides, we just average the lowest classification rates without considering the number of food items, and obtain the average lowest classification rates of each plate situation. As shown in Fig. 13 and Table 2, another fact is that the method performs better on shaped food items like fast food, fruits and dessert when the number of food items is the same. The salad is hard to be classified for it usually does not have plated with a standard pattern. In general, the proposed method can achieve the lowest recognition accuracy of 90% for standard shape food items and 79% for irregular shape food items, which improves the accuracy of food recognition to 88%.

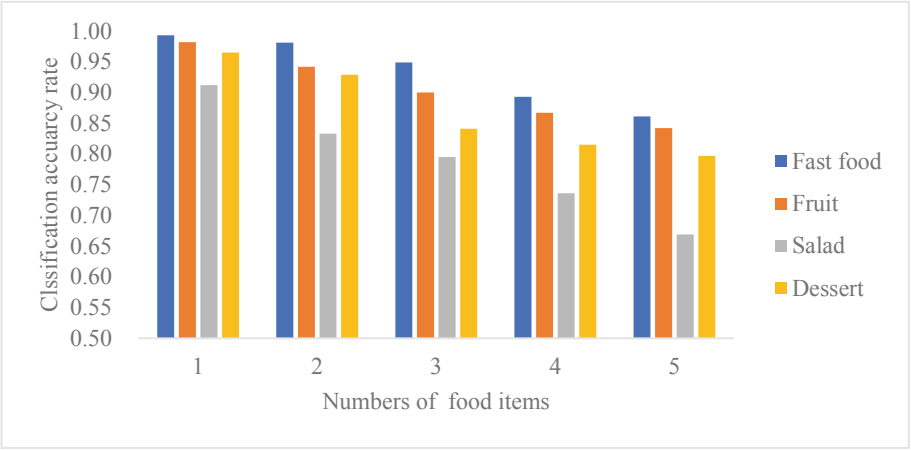


Fig. 13. The classification accuracy rate of different plate situations with the number of food items.

Table 2. The classification accuracy rate regard to different plate situations.

Plate situation	Fast food	Fruit	Salad	Dessert	Average
Accuracy rate (<1.000)	0.935	0.907	0.789	0.869	0.875

In this experiments, we compare the proposed system with server-side recognition system by Martinel et al. [6] and mobile-end recognition system by He et al. [7]. We randomly selected 250 food images from the test dataset to compare the three methods. Among these food images are the mixture of four kinds of plate situations mentioned above.

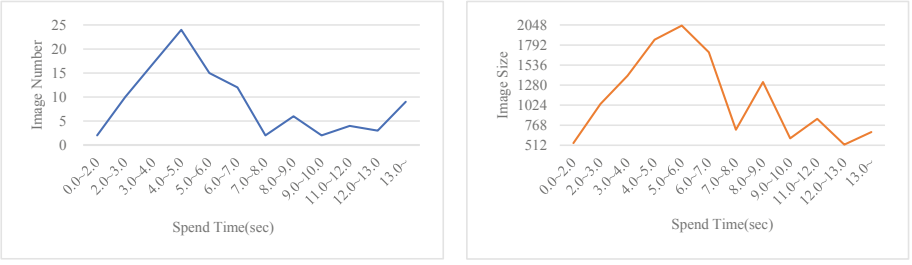
An comparison with existing methods based on the COCO2017 dataset is shown in Table 3. Top1 accuracy refers to the probability that the maximum probability of the prediction result is the correct result, while top5 accuracy means the probability that there is correct result in the top five prediction results. Results demonstrates that the proposed method performs better than the newest server-side recognition method by achieving a Top1 accuracy of more than 92%. Besides, our method on mobile-end is also showing a better recognition accuracy than Multi-SVM proposed in latest mobile-end food recognition system and the Top 1 accuracy can be more than 90%.

Table 3. Top1 and Top5 performance obtained by server-end and mobile-end approaches separately on test dataset.

Method	Top1	Top5
WISeR [6]	90.27	97.84
ResNet+FPN (proposed on pc-end)	92.35	99.87
Multi-SVM [7]	88.86	95.72
ResNet+FPN (proposed on mobile-end)	90.21	99.16

5.2 Evaluation of Processing Time

The system is implemented as an Android application required 36 MB memory (31 MB is mainly for except image processing and 5 MB is mainly for image processing) for the Android virtual mobile device which has a quad-core CPU and using Android 5.0 operation system. Click the button "Choose a food image" at the choose page and input a food image prepared in advance to the system, which will output the results among 2 s and 15 s depending on the size of image and the number of images. As shown as in Fig. 14, the number and size of the input image are the factors that affect the spending time. Firstly, we selected 25 food images of the same size for the affection of images' number. Figure 14(a) shows that the more images we input, the longer the time spend. Input image number which is close to the maximum, and the normal processing spend time is close to 5 s. In addition, although the number of images we output is small, it takes a long time because the food items which need to be recognized are difficult to process. Secondly, we selected 10 food images of different sizes to test the influence of image size. As shown as in Fig. 14(b), the larger the input image size is, the longer the time spend, and the longest time is close to 6 s. Except that spending time increases linearly with size, there are also exceptions due to the difficulty of processing food items. Overall, the system can help a user obtain food information faster than traditional methods.



(a) (b)

Fig. 14. Spend time of different image numbers and different sizes.

From the aspect of memory feasibility, if the system proposed in this paper only needs about 40M memory as an independent application, the main memory consumption is used to call the camera interface, request the server, etc., and the memory consumption in image processing is small, only about 1/4 of which is consumed. Therefore, if the existing image is used for processing, the memory cost is very feasible. In terms of time feasibility, it can be seen from the above that when the image size is not more than 2M, the processing time will not exceed 5 s, and the fastest time can be less than 1 s. Therefore, when it is used in daily life to process food images, it is within the acceptable range in terms of time, because it is also feasible in terms of time cost.

6 Summary and Future Work

This paper has realized to output the types of food items and the corresponding nutrition information on the food image, which is taken real-timely by the mobile terminal, to help people monitor diet and eat healthier. Although the experiment results are not very mature, and there is still a long way to go. Same as many seniors who have brought many methods and ideas of this field, we hope that our efforts can not only provide a theoretical foundation for managing people's health and life, but also help people who interested in this direction continue related study base on it.

To improve the flexibility of the model and the lightness and efficiency of the mobile application, the new real-time target detection network YOLO will be considered as the backbone network in the next work to make the system more applicable. And the way of 3D modeling will be used to optimize the system and make a more detailed estimate of the volume of food to get more accurate food information.

References

1. WHO Homepage. https://www.who.int/zh/news-room/fact-sheets/detail/obesity-and-overweight. Accessed 20 Oct 2019
2. He, K., Gkioxari, G., Dollar, P., et al.: Mask R-CNN. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), **1** (2017)
3. Parrish, E., Goksel, A.K.: Pictorial pattern recognition applied to fruit harvesting. Trans. ASAE **20**, 822–827 (1977)
4. Yang, S., Chen, M., Pomerleau, D., et al.: Food recognition using statistics of pairwise local features. In: The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13–18 June 2010. IEEE (2010)
5. Matsuda, Y., Hoashi, H., Yanai, K.: Recognition of multiple-food images by detecting candidate regions. In: 2012 IEEE International Conference on Multimedia and Expo (ICME). IEEE (2012)
6. Martinel, N., Foresti, G.L., Micheloni, C.: Wide-slice residual networks for food recognition. In: Proceedings of 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, vol. 2018-January, pp. 567–576, December 2016
7. He, H., Kong, F., Tan, J.: DietCam: multiview food recognition using a multikernel SVM. IEEE J. Biomed. Health Inf. **20**(3), 848–855 (2017)
8. Williamson, D.A., Allen, H.R.: Digital photography: a new method for estimating food intake in cafeteria settings. Eat. Weight Disord. – Stud. Anorexia Bulimia Obes. **9**(1), 24–28 (2004)
9. Noronha, J., Hysen, E., Zhang, H., Gajos, K.Z.: Platemate. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology - UIST 2011, p. 1 (2011)
10. Puri, M., Zhu, Z., Yu, Q., et al.: Recognition and volume estimation of food intake using a mobile device. In: IEEE Workshop on Applications of Computer Vision (WACV 2009), Snowbird, UT, USA, 7–8 December 2009. IEEE (2009)
11. Joutou, T., Yanai, K.: A food image recognition system with multiple kernel learning. In: IEEE International Conference on Image Processing. IEEE Press (2009)
12. Kong, F., Tan, J.: DietCam: automatic dietary assessment with mobile camera phones. Pervasive Mob. Comput. **8**(1), 147–163 (2012)

13. Kawano, Y., Yanai, K.: Real-time mobile food recognition system. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops. IEEE (2013)
14. Kawano, Y., Yanai, K.: FoodCam: a real-time food recognition system on a smartphone. Multimed. Tools Appl. **74**(14), 5263–5287 (2015)
15. Pouladzadeh, P., Shirmohammadi, S., Arici, T.: Intelligent SVM based food intake measurement system. In: IEEE International Conference on Computational Intelligence & Virtual Environments for Measurement Systems & Applications. IEEE (2013)
16. Boohee Homepage. http://www.boohee.com/food/. Accessed 20 Oct 2019
17. COCO dataset Homepage. http://cocodataset.org/. Accessed 20 Oct 2019