International Journal of Computers and Applications

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjca20

CEP: calories estimation from food photos

Hong Liang, Yuan Gao, Yunlei Sun & Xiao Sun

To cite this article: Hong Liang, Yuan Gao, Yunlei Sun & Xiao Sun (2020) CEP: calories estimation from food photos, International Journal of Computers and Applications, 42:6, 569-577, DOI: 10.1080/1206212X.2018.1486558

To link to this article: https://doi.org/10.1080/1206212X.2018.1486558

Published online: 25 Jun 2018.

INTERNATIONAL JOURNAL OF COMPUTERS AND APPLICATIONS
2020, VOL. 42, NO. 6, 569–577
https://doi.org/10.1080/1206212X.2018.1486558

Taylor & Francis
Taylor & Francis Group

Check for updates

CEP: calories estimation from food photos

Hong Liang, Yuan Gao, Yunlei Sun and Xiao Sun

College of Computer & Communication Engineering, China University of Petroleum (East China), Qingdao, People's Republic of China

ABSTRACT

Identifying food and estimating its calorie is a novel research area on deep learning. This application comes from the actual needs of diabetics to control their blood glucose level at home. Food calorie estimation can start with estimating the volume of food by three-dimensional reconstructions through depth camera or planar image sequences, and then calculating weight and calories, or considering the food as a whole, through deep learning techniques to first identify food category and then estimates its calorie. In this paper, we propose a deep learning-based food calorie estimating approach. The user takes a photograph of the food, and the object detection model identifies the location and category of the food in the picture. The weight prediction model predicts the weight of the food and finally calculates the calorie according to the category of the food. The results of calorimetry R^2 and RMSE is about 0.95 and 43, respectively, MSE is about 32, and the draw error rate is about 9%, which show that this method has some practical value.

ARTICLE HISTORY
Received 12 March 2018
Accepted 5 June 2018

KEYWORDS
Food recognition; deep learning; calorie estimating; diabetes

1. Introduction

Deep learning is a widely used object detection technique today, but identifying food and estimating calories is a novel research region. This application comes from the actual needs of diabetics to control their blood glucose level at home. Diabetes is one of the major chronic non-communicable diseases in the world. Diabetes is a lifelong disease, which is accompanied by a variety of complications, and there is no complete cure for this disease. A healthy and reasonable lifestyle can significantly reduce the severity of diabetes and reduce the risk of developing complications of diabetes [1]. Snel et al. [2] pointed out that low-calorie diet and exercise can improve the quality of life of diabetics with type 2 diabetes mellitus, and effectively improve their physical condition. However, this process can be very tedious as it requires the patients to continually record the food and make messy calculations in order to track the calories of each food.

Some software can help people with the need for food calorie management, which usually requires the user to input food information manually and then compares the information with food database in order to calculate the calories the user consumes. In general, these databases are mainly about prepared food and have certain standards. For example, foods eaten at restaurants, meals bought at supermarkets, each with a fixed weight and calories, users may not be able to cut the exact amount needed for their individual requirement.

With the popularization of mobile devices and improvement of the computing capabilities, smartphones and tablets can achieve more complex computing tasks, while deep learning image recognition technology makes it easy to distinguish the objects in the image accurately and efficiently. With the improvement of people's health awareness, more and more attention has been paid to automatic food calorie estimation task. Food calorie estimating may start with estimating the volume of food by three-dimensional reconstructions through depth camera [6] or planar image sequences [7,8], and then derives weight and calories. It can also take the food as a whole [14,15], through deep learning techniques to first identify and then predict calories. Combining the computational capabilities of mobile devices and image recognition algorithms, on the principle of facilitating the use of diabetics, we propose a novel approach based on deep learning technique, called Calories Estimate from food Photos (CEP). First, the food detector identifies the location and type of ingredients in a food photo, and then the weight predictor estimates the food weight and finally calculates the food calorie information.

Our method is based on deep learning technology and can run on a mobile device. It only needs a piece of the ingredients picture to estimate food calories, which is

CONTACT Yuan Gao 744275444@qq.com College of Computer & Communication Engineering, China University of Petroleum (East China), Huangdao District, Qingdao, People's Republic of China

convenient and efficient. The user takes a picture of the ingredients and CEP can identify the types of ingredients in the picture and estimate the calories of the ingredients. The fitting result of our approach with R^2 0.95, RMSE 43, MSE 32, and the mean error rate 9%, which show that this approach has some practical value. There are two main contributions to the research field of food calorie estimation: (1) a method for estimating food calories based on deep learning using a white plate as a reference is proposed; (2) a food dataset for the above training was constructed and will be published in the future to support other researches.

The rest of this paper is organized as follows: Section 2 introduces food calorie design ideas. Section 3 presents experiment design and result evaluation. Section 4 describes related work. In Section 5, we conclude this paper.

2. Food calorie estimation design

2.1. Main idea

The goal of our work is to automatically estimate calorie in food and taking a photo of the food through the user's smartphone is a rather convenient way. To do this, first, we need to identify the type and location of the food in the photo, then estimate the weight of the food, and finally calculate the total calorie content of the food. It is well-known that the weight of a certain food has a positive correlation with its volume, and the weight of a food also has a positive correlation with its visual area in the planar image with the same perspective. We use the polynomial linear regression method to fit the correlativity between the weight of a food and its visual area, and through this correlativity to estimate the weight of the food in the picture.

The system architecture of CEP is shown in Figure 1. Firstly, the user takes a photo of the food, and the photo is sent to a food recognition model to identify the position and category information of food in the photo. The food recognition model is a deep learning neural network that needs us to conduct training. Then, the weight prediction model predicts the weight of the food based on the recognition result. The weight prediction model is a polynomial linear regression model and also needs to be trained. Finally, query the caloric density of the food to calculate the amount of calories.

2.2. Deep learning food recognition

Food identification is usually considered as a binary classification problem. The algorithm only needs to distinguish what kind of food the given image represents [4,6,7]. In fact, food identification should be an object recognition process. You have to identify not only how many foods are in the picture but also their categories. At present, deep learning food identification researches are usually based on the Food-101, UESFood256, UEC-Food100, and other classic food datasets; they can only identify prepared restaurant food (burgers, French fries, salads, etc.). Our goal, however, is to identify the ingredients, which requires a customized ingredient dataset to train our food recognition model. At the same time, our approach should be deployed on mobile devices, so the trained model should be efficient and lightweight.

The first step is to find an object recognition model with high accuracy and lightweight. Google proposed a lightweight deep neural network for mobile devices such as embedded devices, named MobileNets [3]. MobileNets uses the idea of depthwise separable convolutions [4]. Instead of fusing channels with convolution of 3×3 (or larger), MobileNets uses the dextral (or channelwise) and 1×1 pointwise methods to deconvolute the convolution. It decomposes the standard convolution into a depth convolution and a dot convolution (1×1 convolution kernel). Deep convolution applies each convolution kernel to each channel, while 1×1 convolutions combine the channel convolution output. This decomposition can effectively reduce the computational cost and reduce the size of the model. The MobileNets structure is built on the deep-resolution deconvolution (only the first layer is a standard convolution) mentioned above. The network allows us to explore the network topology and find a suitable good network. Except for the final fully connected layer, all layers followed by BatchNorm and ReLU finally output to the softmax layer for classification.

The second step is to retrain the model using our customized food dataset to improve the accuracy of

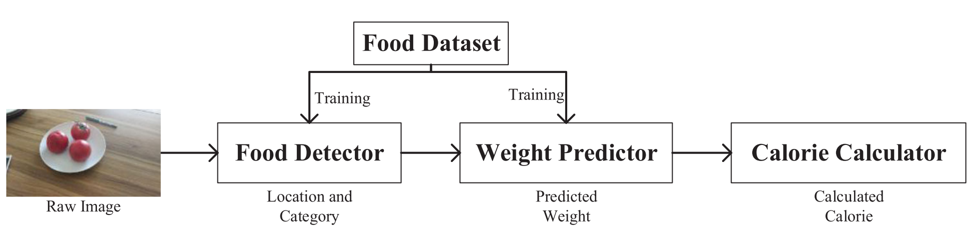


Figure 1. CEP system architecture.

ingredient recognition. The modern object recognition model has millions of parameters and can take weeks to complete a full training. Migration learning [5] is a technique that drastically reduces the cost of training process, which simplifies a lot of work by using well-trained models such as ImageNet tuned for new classes. Although this process is not as good as complete training, for many scenarios this is very effective. This paper applies this technique. The main purpose of training is to not only improve the accuracy of ingredient recognition but also pay attention to the identification of non-food (plate), because it is an important reference in this approach.

After identifying the type of food, the next step is weight estimation.

2.3. Weight estimation

The weight of an object is equal to the product of its density and volume. The density of an object is constant when the environment is constant; the volume is a reflection of the space occupied by the object in a three-dimensional world. How to add missing stereoscopic information is the key issue. One way is to use a depth camera [6]; specialized devices, such as the Intel RealSense F200 depth sensor, Kinect, and so on, can shoot an image containing depth information and utilize depth information to perform 3D reconstruction to calculate the volume. In addition, we can also consider the planar images sequence-based three-dimensional reconstruction model, which requires the user to shoot a series of photos around the object. The shots must be partly overlapped and then uploaded to professional software which can generate a three-dimensional model of the object and then calculates the volume. There are some common desktop software, such as Autodesk Recap and 123D Catch, and also some mobile solutions work such as [7,8]. The depth camera is a specialized equipment which is so expensive and purpose-specific that ordinary families purchasing it only for food shooting is like using a sledgehammer to crack a nut. The disadvantage of the planar image sequence 3D model reconstruction is the complexity of preparatory work that makes the user's burden heavier, and the reconstruction of the software requires powerful computational resources' support. The above method is specialized, complicated, and costly, which is not suitable for home operation.

It is well known that the weight of a certain food has a positive correlation with its volume, and the weight of a food also has a positive correlation with its visual area the same correlation in the planar image with the same perspective. Visual area of food changes with different shooting angles, which requires a fixed size object in the image as a reference object; the reference object and food ratio in the picture will not change too much. Household white plate is a good choice, commonly used size is 9 inches (diameter 22.5 cm). Dividing the white plate area S_{vege} in the image by the food area S_{vege} to obtain the ratio $R_{plate/vege}$, the linear relationship between $R_{plate/vege}$ and W_{vege} was fitted by polynomial linear regression to get the linear relationship between the two, in order to predict the food weight in the image after object recognition.

As shown in Figure 2, the position of the food in the image is determined by the rectangles consisting of four values $x1$, $y1$, $x2$, and $y2$, i.e. two points (upper left

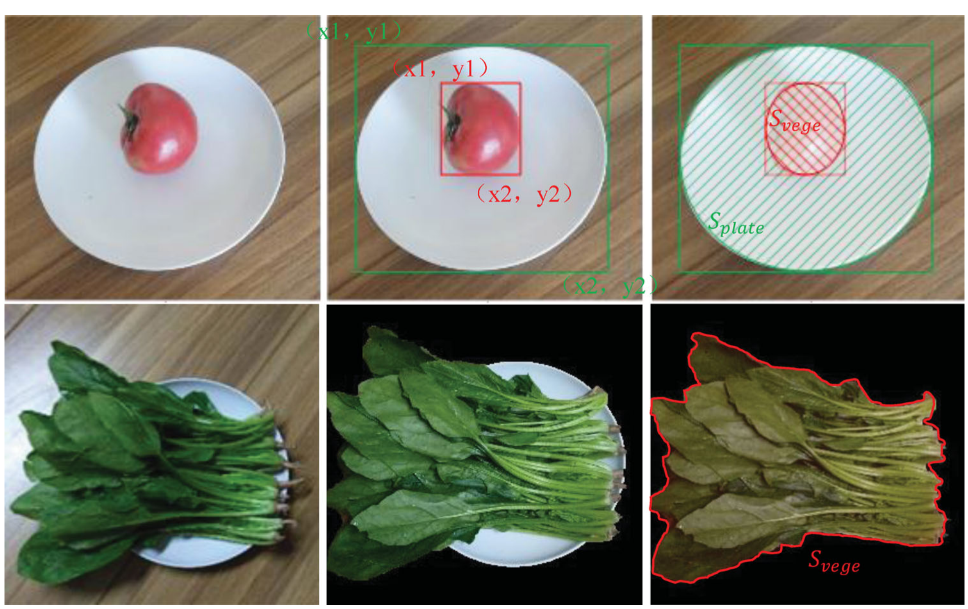


Figure 2. Area calculation.

and lower right corners). The visible area S_{vege} can be approximately calculated by the inscribed ellipse area,

$$S_{\mathrm{vege}} = \pi \times \frac{|x2 - x1|}{2} \times \frac{|y2 - y1|}{2}. \tag{1}$$

It should be noted that this method is effective only for food with regular shapes, such as potatoes and tomatoes, and the error for irregular foods is relatively large. The visual area can be obtained by calculating the number of pixels of food. Similarly, the reference area S_{plate} of the reference object is also determined by its position information. The ratio $R_{\mathrm{plate/vege}}$ of the two is,

$$R_{\mathrm{plate/vege}} = \frac{S_{\mathrm{plate}}}{S_{\mathrm{vege}}}. \tag{2}$$

The positive correlation between the visible area of food and its volume can be expressed by the following correlation:

$$V_{\mathrm{vege}} = f(R_{\mathrm{plate/vege}}). \tag{3}$$

Then the weight of food in the image can be calculated by the following formula:

$$\begin{aligned} W_{\mathrm{vege}} &= V_{\mathrm{vege}} \times D_{\mathrm{vege}}, \\ &= f\left(\frac{S_{\mathrm{plate}}}{S_{\mathrm{vege}}}\right) \times D_{\mathrm{vege}}, \\ &= f(R_{\mathrm{plate/vege}}) \times D_{\mathrm{vege}}, \\ &= g(R_{\mathrm{plate/vege}}). \end{aligned} \tag{4}$$

W_{vege}, S_{plate}, S_{vege}, and D_{vege} can be obtained by measurement or calculation. W_{vege} is considered as the dependent variable, and S_{plate} and S_{vege} are used as independent variables to perform fitting using polynomial linear regression. D_{vege} is a fixed value and is included as a constant in the formula. The food area, the mapping between the area of the white plate, and the weight can be obtained to predict the weight of the food in the image after the object is recognized.

Assuming that the food recognition model has successfully identified the position of each ingredient and plate in the image and marked with surrounding box, the area of the object can then be calculated and if you have multiple objects of the same type, pay attention to accumulate them. In this way, S_{vege} and S_{plate} are respectively obtained, and the two are divided to obtain the ratio $R_{\mathrm{plate/vege}}$. Substituting $R_{\mathrm{plate/vege}}$ into $g(R)$ yields W_{vege}, multiplied by the food for every 100 g of calories to get the predicted value of C.

$$C = c_{\mathrm{vege}} \times g(R). \tag{5}$$

Note that the c_{vege} is a fixed value, which needs to be inquired according to the recognition result of the food identification model. The next step is the calculation of food calories.

2.4. Calorie calculation

The key point during a diet is to control food calorie intake. A reasonable diet can reduce the load of pancreatic β-cells which is conducive to the control of blood glucose levels, delay the occurrence of complications, and improve quality of patient's life. Steven et al. [9] through experiments confirmed that very low-calorie diet can significantly reduce fasting blood glucose and HbA1c in patients with type 2 diabetes mellitus.

There are various cooking methods to process food and different cooking methods will make the final food calories slightly different; accurate measurement of prepared food calories generally utilizes the oxygen bomb method. The oxygen bomb method uses a certain amount of test objects into airtight metal containers filled with oxygen (oxygen bomb) to fully combust; the heat released by the combustion is transferred to the surrounding water. The calorimeter calculates the energy value of the sample according to the degree of water temperature rise, the calorific value, that is the caloric content of the food. This type of professional measuring equipment is too costly for an average household and is complicated to operate. We adopt a workaround to measure the calories of raw ingredients such as vegetables, meat, and so on, to approximate the calorie content of the prepared food meanwhile ignoring the influence of ingredients and cooking methods.

From weight to calories, you need to know the caloric density of each food, which is the amount of calories per 100 g of food. The standard source is the U.S. Department of Agriculture's National Nutrition Database (NNDB). The latest version (May 2016) is Standard Edition 28 [10], which lists the nutritional content of 8789 basic foods. The focus of NNDB is 'raw' food, not cooked food, which is the focus of our work.

3. Experiment and evaluation

3.1. Dataset

Microsoft COCO (Common Objects in Context) and Pascal VOC (Visual Object Classes) are classic datasets in the field of image recognition, but the information contained in them does not meet our requirements, mainly because they do not contain food weight information. At the heart of this work is the caloric identification of ingredients, not the prepared food, so we need to build our own food dataset that can contain weight and surrounding box information.

We bought vegetables (common spices), 9-inch (22.5 cm diameter) white plates, and electronic scales (to the nearest 0.1 g). We took pictures with cellphones since the user's environment is cellphones too. First, weigh the vegetables on an electronic scale and place them on a white plate. White plate works as a reference, so that you can easily estimate the size of the object through the ratio of the disk. Model training requires a variety of images, i.e. training samples. Put one, two, and more vegetables into the plate, then mix different spices randomly. Shooting angle should be changed to ensure the diversity of training samples (Figure 3).

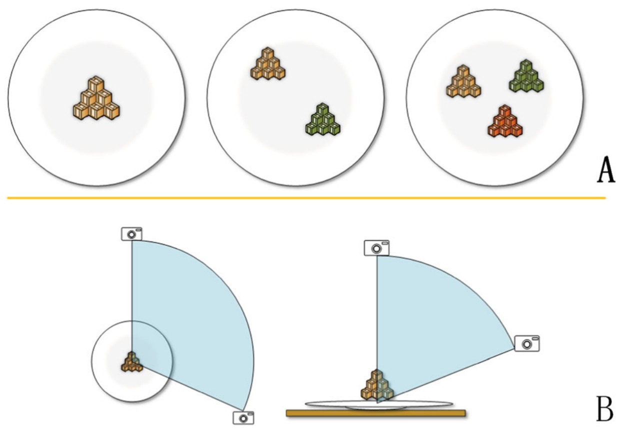


Figure 3. Image acquisition plan: (A) Various food ingredients; (B) Various shooting angles.

The collection process was performed by four members together, each using their own cellphone for shooting; mobile phone camera parameters are different so shooting effects are also different. The result is a total of 633 original images in 11 categories with a resolution of at least 3968 × 2240 pixels. In order to improve the efficiency during the following training, the width and height are respectively halved, that is, the original image is scaled to a quarter, and the image still has a high resolution and will not affect the training effect. The images should be marked manually with the LabelImg software, and finally the images will be gray scale and salt and pepper noise added separately to get additional training samples (Figure 4).

3.2. Experimental configuration and process

The experimental model is based on the TensorFlow Object Detection API [12] provided by TensorFlow [11]. TensorFlow is an open source software library for numerical computation using data flow graphs. The flexible architecture allows the developer to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a

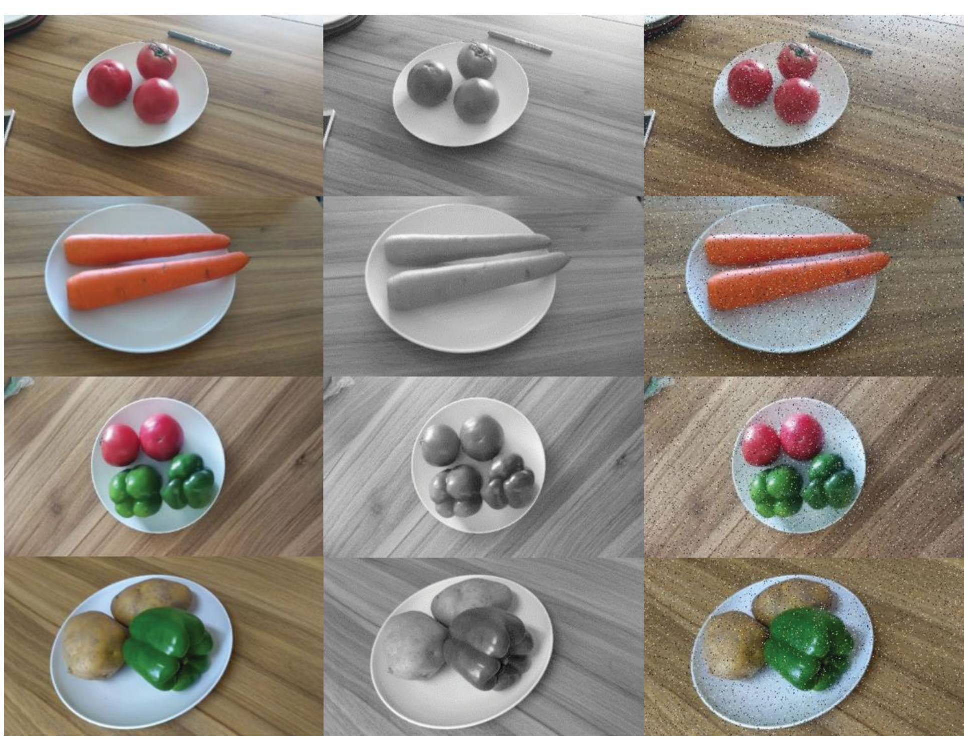


Figure 4. Image data samples from left to right are the original image gray scale image, and adding salt and pepper noise image.

wide variety of other domains as well. TensorFlow Object Detection API is an open source framework built on TensorFlow which makes it easy to build, train, and deploy object detection models.

Object detection model uses SSD (Single Shot MultiBox Detector) [13] as an object detector, built on MobileNet model. SSDis a method of detecting objects in an image using a single deep neural network. The SSD discretizes the bounding box's output space into a set of default boxes that have different widths at each of the feature maps height ratio and size. During the forecast, the network generates a score for each existing object category in each default box and adjusts the box to better match the shape of the object. In addition, the network combines the prediction comes from multiple feature maps with different resolutions to accommodate objects of various sizes. MobileNet [3] is a mobile-first computer vision model introduced by TensorFlow in order to effectively improve the accuracy. With limited resources for both device and embedded applications, MobileNet is a parametric, small, low-latency, low-power model that meets the resource constraints of various use cases. They can be based on classification, detection, embedding, and segmentation, similar to the use of other popular large models (such as Kai). Since the user's environment is mostly a smartphone, taking into account the operating speed, computing power, accuracy, and other factors, we have chosen SSD_Mobilenet object detection model (Table 1).

The SSD_Mobilenet model provided by the TensorFlow Object Detection API has been trained on the Microsoft COCO dataset and has been able to identify most common items accurately and quickly, but it does not satisfy our need to accurately identify the type of food, so we need to retrain it with our ingredients. Fortunately, TensorFlow also provides related APIs to facilitate our training.

Table 1. Performance comparison of different models.

Model name	Speed (ms)[1]	COCO mAP[2]
ssd_mobilenet_v1_coco	30	21
ssd_inception_v2_coco	42	24
faster_rcnn_inception_v2_coco	58	28
faster_rcnn_resnet50_coco	89	30
faster_rcnn_resnet50_lowproposals_coco	64	
rfcn_resnet101_coco	92	30
faster_rcnn_resnet101_coco	106	32
faster_rcnn_resnet101_lowproposals_coco	82	
faster_rcnn_inception_resnet_v2_atrous_coco	620	37
faster_rcnn_inception_resnet_v2_atrous_lowproposals_coco	241	
faster_rcnn_nas	1833	43
faster_rcnn_nas_lowproposals_coco	540	

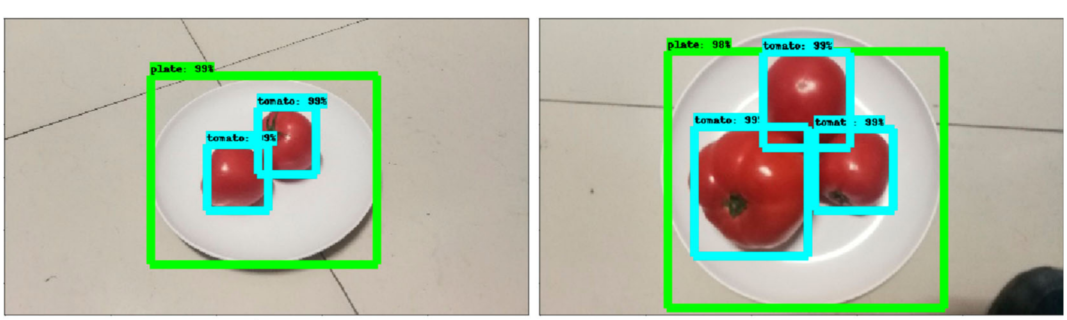
[a]Running time in ms per 600×600 image (including all pre and post-processing), but please be aware that these timings depend highly on one's specific hardware configuration (these timings were performed using an Nvidia GeForce GTX TITAN X card) and should be treated more as relative timings in many cases.

[b]Detector performance on subset of the COCO validation set or Open Images test split as measured by the dataset-specific mAP measure. Here, higher is better, and we only report bounding box mAP rounded to the nearest integer.

3.3. Results and evaluation

To verify the effectiveness of the food recognition model and the weight prediction model, we conducted validation experiments on the ingredient dataset. The dataset contains a total of 633 original images in 11 categories, averaging nearly 60 images in each category. The experiment includes the training and verification of food recognition model and the training and verification of weight prediction model. The training process was performed on a PC with 40% of the dataset for training, 40% for validation, and 20% for testing.

Figure 5 demonstrates the results of the food recognition model. Taking the identification results of tomatoes as an example, we can see that our model can accurately identify the position and classification of objects in the photo with high recognition accuracy, mAP @ 0.5IOU reached 0.99. This is because the SSD_Mobilenet model we used has been fully trained on the Microsoft COCO dataset, which already has a high



A B

Figure 5. Food identification result: different categories of objects were marked with different colors.

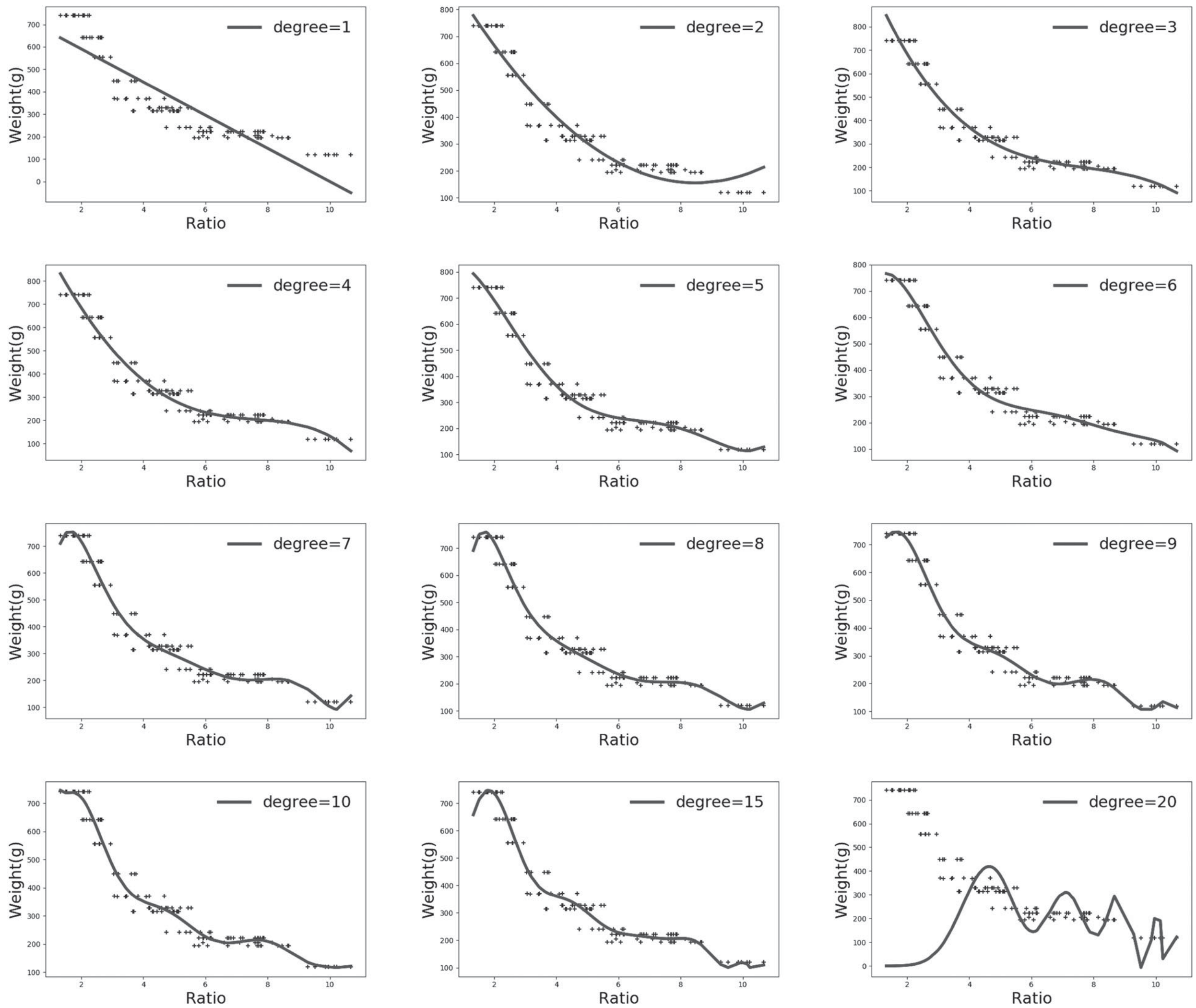


Figure 6. Fitting result curves: dots are real data and lines are the predicted values.

accuracy. This experiment uses the migration learning method to retrain its classifier, so it will get very good results.

$R_{\text{plate/vege}}$ and W_{vege} n-degree polynomial linear regression fitting results are shown in Figure 6 and Table 2. Take $n = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20\}$. When $n = 3$ or 4, the fitting result is ideal. R^2 is 0.95, RMSE is about 43, mean absolute error (MAE) is about 32, and the average error rate is about 9%. Although the scores from 5 to 10 have an increasing trend in value, it can be seen from the fitting curve that the trend of the curve changes anomalously around the maximum and the minimum of W_{vege}. As n increases, the fitting result becomes more unpredictable. So the final choice is when $n = 3$ or 4, that is three- or four-degree polynomial linear regression fitting result as $R_{\text{plate/vege}}$ and W_{vege} regression function (Table 3).

Table 2. Fitting result score.

Degree	R^2	RMSE	MAE	Percentage
1	0.796371	87.125652	74.987232	25.678905
2	0.927827	51.869787	41.865874	14.754003
3	0.950511	42.951731	32.466887	9.503138
4	0.951282	42.616010	32.027347	9.468686
5	0.954998	40.958356	31.450706	9.074906
6	0.956695	40.178629	31.784793	9.779162
7	0.962786	37.246223	28.687141	8.888155
8	0.963484	36.895138	27.903825	8.183153
9	0.965990	35.606585	26.183026	7.774644
10	0.966907	35.123040	25.246171	7.299068
15	0.964023	36.621519	26.837035	7.643487
20	−2.241336	347.607194	229.903128	47.550964

Notes: R^2 is comparison between predicted value and mean value only, and the interval is usually between (0, 1). 0 means that you are not predicting anything at all, and 1 means that all the predictions match perfectly with true result. Root Mean Squared Error (RMSE) is the mean square root of the error between predicted and true values. Mean Absolute Error (MAE) is the average of the absolute error, and the average absolute error can better reflect the actual situation of the prediction error. Percentage is the average error rate.

Table 3. Weight and calorie prediction results.

Image	Actual	Predicted
A	241 g, 39 kcal	268 g, 43 kcal
B	642 g, 103 kcal	600 g, 96 kcal

The results of calorie prediction are shown in Table 3; the prediction error rate of photo A in Figure 5 is 10.25% and the absolute error is 4 kcal, and the prediction error rate of photo B in Figure 5 is 6.80% and the absolute error is 7 kcal. The error is kept within a relatively small range; it does not have much impact on the formulation of diet plans in practice and shows that this approach has some practical value.

4. Related work

Calorie estimating method can be roughly divided into these two categories, either directly estimate the total calories of prepared food, or also estimate the calorific value of ingredients and then calculate the total calorie of the food. For prepared foods, as proposed in [6,14,15], the food information (weight and calories) provided by the restaurant is taken as key features and the food is taken as a whole to estimate calorie of the food directly. As for home cooking, because cooking methods and food ingredients do not have the same set of standards as restaurant-prepared foods, a better way is to start with the estimation of the calories of the ingredients and then calculate the total calories of the food. This paper belongs to the second category of work.

Im2Calories [6] is an automatic food calorie recognition system using deep learning technology. The food images are sent to a CNN model for semantic image segmentation to automatically identify all the foods contained within, and then through another CNN model for depth prediction. Calculate the volume by voxel grid estimation. According to depth prediction, finally query USDA NNDB or FNDDS for food calorie calculation. The advantage of Im2Calories lies in the stage of model training of depth prediction CNN; Myers et al. used a depth camera to take the actual food image with depth information as the training sample, and the trained model has a better prediction performance. But at the same time, a depth camera is a specialized equipment which is expensive and purpose-specific, ordinary families purchasing it only for food shooting is like using a sledgehammer to crack a nut. The work of Myers et al., which is a prediction of the calorie content of prepared foods in restaurants, may not be suitable for users who like to cook their own food.

Chokr et al. [14] have done some work on food image calorie estimation. They extracted features from food images by Information Gain and PCA, then fed into SMO for food classification and Random Forests for food volume prediction, and Multilayer Perceptron for calorie prediction. With SMO Accuracy and F-measure reaching 0.991 and 0.991, respectively, Random Forests and Multilayer Perceptron's MAE reached 2.750 and 0.0933, respectively. The final prediction result is very close to the true value. The disadvantage of their current work is that only one type of restaurant-prepared food can be predicted in every picture, and there are certain restrictions on usage scenarios (white background).

The approach proposed by Akpa et al. [15] is to use a mobile phone camera to take a food photo and take two chopsticks next to the food container as a reference object to estimate the food container based on the EXIF metadata (camera focal length and sensor size) of the food image in order to estimate the size and the volume of food, according to food density and nutrient information obtained by the type of food to estimate calories. For weight estimation, the average relative error rate was 6.65% and the calorie estimated relative error rate was 6.70%. Akpa does not implement automatic food recognition, so users still need to select a recognition area. A phone is only used as a shooting device while computing job needs to be uploaded to the backend server, which may affect the user experience in a poor network environment. Likewise, their job is also focusing on calorie estimation of the food prepared in a restaurant.

Restaurant food is not a good option for people with diabetes, because the amount of a food meal is basically fixed, and even knowing the total calorie information still will not be able to accurately segment the portion that suits your dietary needs. Home cooking is a relatively good option for quantitative control of food quantity and calories, but there is no work on calorie estimation for home cooking. We hope that some attempts will be made in this regard.

5. Conclusion

To help diabetics better manage food calorie intake, our paper proposed a novel approach to automatically estimate food calories from ingredient images based on deep learning neural networks. The approach first uses an ingredient image dataset to train an object recognition model, which can identify all the food ingredients in the image and the white plate works as a reference object, and then, through polynomial linear regression, fits the relationship between the weight of the food ingredient and the area of the food ingredients in the image, and finally calculates the calories according to the calorific value of the ingredients. Other methods can only estimate the calorie content of restaurant foods, while each serve of restaurant food has a fixed amount. Compared to other methods, our method is suitable for home cooking, and users can flexibly control the amount of food, which is conducive to heat control. The fitting results showed that R^2 was 0.95, RMSE was about 43, MAE was about 32, and the average error rate was about 9%. For vegetables with low calorific value, the 9% error rate is within the acceptable range and will not have a serious impact on the diet plan. This method still has some practical value.

The method proposed in this paper still needs to be improved. The use of the white plate will make the method limited. However, the application scenario is mainly in a home kitchen, which is not a big problem. In future work, we plan to continue to improve our approach to reduce restrictions and enhance its usage scenarios, expand the database, and enrich the types of ingredients; besides, cooking methods are also included in the calculation. Update food ingredients nutrient information to meet the needs of different users. At the same time, using a more accurate model to fit the relationship between food weight and area in the image reduces the error. Gather larger training sets and improve model structure to improve accuracy and speed.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Hong Liang is a Professor in the College of Computer and Communication Engineering, China University of Petroleum (East China). His research focuses on high performance computing and Big data technology.

Yuan Gao and *Xiao Sun* are graduate students in the College of Computer and Communication Engineering, China University of Petroleum (East China). Their research focuses on Big data technology.

Yunlei Sun is a lecturer in the College of Computer and Communication Engineering, China University of Petroleum (East China). His research focuses on high performance computing and Big data technology.

References

[1] Diabetes Control and Complications Trial Research Group. The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus. N Engl J Med. 1993;329(14):977–986.
[2] Snel M, Sleddering MA, vd Peijl ID, et al. Quality of life in type 2 diabetes mellitus after a very low calorie diet and exercise. Eur J Intern Med. 2012;23(2):143–149. doi:10.1016/j.ejim.2011.07.004
[3] Howard AG, Zhu M, Chen B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. 2017.
[4] Sifre L, Mallat S. Rigid-motion scattering for texture classification. Comp Sci. 2014;3559:501–515.
[5] Donahue J, Jia Y, Vinyals O, et al. DeCAF: a deep convolutional activation feature for generic visual recognition. International Conference on International Conference on Machine Learning JMLR.org; 2014. I-647.
[6] Meyers A, Johnston N, Rathod V, et al. Im2Calories: towards an automated mobile vision food diary. IEEE International Conference on Computer Vision; IEEE; 2015. p. 1233–1241.
[7] Kong F, Tan J. Dietcam: automatic dietary assessment with mobile camera phones. Pervasive Mob Comput. 2012;8(1):147–163.
[8] Pouladzadeh P, Shirmohammadi S, Al-Maghrabi R. Measuring calorie and nutrition from food image. IEEE Trans Instrum Meas. 2014;63(8):1947–1956.
[9] Steven S, Hollingsworth KG, Al-Mrabeh A, et al. Very low-calorie diet and 6 months of weight stability in type 2 diabetes: pathophysiological changes in responders and nonresponders. Diabetes Care. 2016;39(5):808–815.
[10] US Department of Agriculture. Agricultural research service, nutrient data laboratory. USDA national nutrient database for standard reference, release 28 (Slightly revised). [Version Current: May 2016]. Internet. Available from: http://www.ars.usda.gov/ba/bhnrc/ndl
[11] TensorFlow. https://www.tensorflow.org/
[12] Huang J, Rathod V, Sun C, et al. Speed/accuracy trade-offs for modern convolutional object detectors. 2016.
[13] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector. European conference on computer vision. Cham: Springer; 2016. p. 21–37.
[14] Chokr M, Elbassuoni S. Calories Prediction from Food Images. Innovative Applications of Artificial Intelligence; North America; 2017. p. 4664–4669.
[15] Akpa EAH, Suwa H, Arakawa Y, et al. Smartphone-based food weight and calorie estimation method for effective food journaling. SICE JCMSI. 2017;10(5):360–369.