S & M 3304

Food Calorie Estimation System Based on Semantic Segmentation Network

Xiang-Yong Kong,[1] Xiao-Han Sun,[1] Yu-Ze Wang,[1] Rui-Yang Peng,[1]
Xin-Yue Li,[1] Yi-Heng Yang,[1] Ying-Rui Lv,[2] and Shih-Pang Tseng[3,4*]

[1]School of Health Science and Engineering, University of Shanghai for Science and Technology,
No. 516 Jungong Road, Shanghai 200093, China
[2]Business School, University of Shanghai for Science and Technology,
No. 516 Jungong Road, Shanghai 200093, China
[3]School of Software and Big Data, Changzhou College of Information Technology,
No. 22, Mingxin Middle Road, Changzhou 213164, China
[4]School of Information Science and Technology, Sanda University,
No. 2727 Jinhai Road, Shanghai 201209, China

(Received July 31, 2022; accepted May 19, 2023)

Keywords: food calorie estimation, semantic segmentation, pattern recognition, deep learning

The food calorie estimation system (FCES) is designed to record dietary information for diabetic patients to monitor their dietary intake to estimate the number of calories they are consuming. Deep learning technologies have recently been used for FCESs. In this work, we use the neural network for the pattern recognition of food images to calculate the number of calories. In contrast to the traditional convolutional neural network, we build a semantic segmentation network model based on SegNet + MobileNet to segment the food images and extract the area feature of food images. By determining the corresponding relationship between the area feature of the food image and the food calorie value, the number of calories in the food can be estimated and realized. The experimental results show that the accuracy of food recognition reached 97.82% and that of calorie estimation was above 84.95%.

1. Introduction

In recent years, with the aging of the population and changes in lifestyles, diabetes has gradually developed into a familiar and frequently occurring disease in several modern societies. In particular, type 2 diabetes is the most common type of diabetes in the 21st century.(1) Diabetes is a chronic metabolic disease characterized by increased blood sugar levels. Typical symptoms include polyuria, polydipsia, polyphagia, and weight loss.(2) After discovering elevated blood sugar and diagnosing diabetes, most patients tend to underestimate the dangers of diabetes because the symptoms are not obvious, and they do not pay attention to the treatment of diabetes. If there is poor health management in the early stage, diabetes can cause various serious complications later.

[*] Corresponding author: e-mail: tsengshihpang@czcit.edu.cn

https://doi.org/10.18494/SAM4061

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

The application of artificial intelligence (AI) in the medical field has changed the mode of medical service and the concept of health management to a certain extent. This development is conducive to strengthening disease prevention, enhancing patient compliance, and making the management system more intelligent to improve the management efficiency of chronic diabetes. Moreover, it also inspires people's concept of a healthy life and fundamentally reduces the medical cost of the whole society.

In this paper, we propose an effective food calorie estimation approach based on deep learning image semantic segmentation. Three main tasks are performed in this approach: food recognition, volume calculation, and calorie conversion.(3) The deep learning neural network can perform food recognition and volume calculation well. According to a standard food nutrition table, we can quickly establish a conversion formula from the food category and volume to calories. In addition, an easy-to-use smartphone application (APP) prototype is designed and implemented for diabetic patients. The trained deep learning model has been deployed in the APP to perform the calorie estimation. Moreover, the APP can also provide some useful functions, such as sports management and blood sugar monitoring, to diabetic patients.

The rest of this paper is organized as follows. Related works are briefly described in Sect. 2. In Sect. 3, we describe the proposed approaches in detail, including the backbone network and three different network models. In Sect. 4, we show the experimental processes and results, including the image acquisition, preprocessing, setting, and performance. The APP prototype is demonstrated in Sect. 5. Finally, the conclusions are given in Sect. 6.

2. Related Works

In recent years, AI has also penetrated into diabetes-related fields, and many new advances have been made in disease prediction, diagnosis, blood glucose monitoring, and complication screening. In diabetes management, AI technology plays an important role.(4,5) Studies have used AI to mine data in sequential patterns, determine the order of use between drugs, and accurately predict the next drug that the doctor may specify for the patient. In addition, the artificial pancreas with intensive learning can better control the blood sugar of diabetic patients and reduce the risk of hypoglycemia.(6)

Norouzi *et al.*(7) proposed a mobile APP for managing food nutrition for diabetic patients. It can provide the food plan according to the health status of the user. However, the user still inputs all health data by hand. An image-based automatic food energy estimation technique, which uses the generative adversarial network (GAN), has been proposed by Fang *et al.*(8) However, this approach has a high response delay in physical applications.

In the current field of AI in the management of diabetic patients, the diet monitoring of diabetic patients is important and necessary. Studies have shown that the manual reporting of food intake is inaccurate and usually impractical,(9) so an automated solution for diet monitoring needs to be sought. A diet monitoring system can be designed and implemented by image analysis. It requires users to use smartphones to take pictures of food and send the pictures to the server. The server analyzes the images, estimates the nutritional characteristics of the food, reports it to the user, and send it to the health professional.(10) It can be seen that the food image analysis system needs to solve image segmentation, food recognition and classification, food volume estimation, and calorie conversion. These tasks need to be linked and studied for the entire system to build a complete and accurate food image analysis system. In summary, the application of AI technology in the field of diabetes is not perfect, and there is still much room for performance improvement. On the basis of these previous studies, we proposed and implemented a novel food recognition and calorie conversion method in this work.

3. Construction of Food Image Semantic Segmentation Model

3.1 Backbone network

Considering that the client system to be constructed in the next step of this experiment is based on the WeChat applet, in this study, we use the MobileNet network structure to form the backbone network for image feature recognition.(11) MobileNet is a lightweight convolutional neural network model that can be applied to mobile terminals. It is a compromise between accuracy and response time.

The core technology of MobileNet is depthwise separable convolution. The overall convolution effect is similar to a standard convolution result. However, regarding the amount of calculation, the depthwise separable convolution can considerably reduce the model's parameters, reduce memory saturation, and improve training speed.(11) Unlike ordinary convolution, which considers channels and regions simultaneously, the idea of depthwise separable convolution is first to examine the region and then merge multiple channels to separate regions and channels. As a result, MobileNet uses deep separable convolution for the first time to significantly reduce the amount of calculation and is suitable for building lightweight networks for mobile deployment. When building the model, we use the DepthwiseConv2D layer in Keras to achieve deep separable convolution.

3.2 Semantic segmentation model based on SegNet + MobileNet

SegNet is a classic encoder-decoder structure in which the encoder draws on the convolutional layer structure of VGG-16 and removes the fully connected layer of CNN like FCN. The decoder uses the max-pooling index similar to the leading pooling backpropagation technology to record the corresponding up-sampling output value position, which improves the recognition effect of boundary features and reduces the amount of calculation.(12)

The main structure of the semantic segmentation model based on SegNet + MobileNet constructed in this paper is shown in Fig. 1. It is mainly composed of deep separable convolutional blocks. First, in the encoder, the input image undergoes multiple separable convolutions in the backbone model to extract a layer with specific characteristics and then uses the UpSampling2D function in the decoder to perform threefold sampling. Finally, a layer of a certain width and height with the number of channels equal to that of categories, $n_{classes}$, is obtained, which is the result of semantic segmentation. $n_{classes}$ is termed as the number of channels because it represents the number of categories each pixel belongs to. Here are some

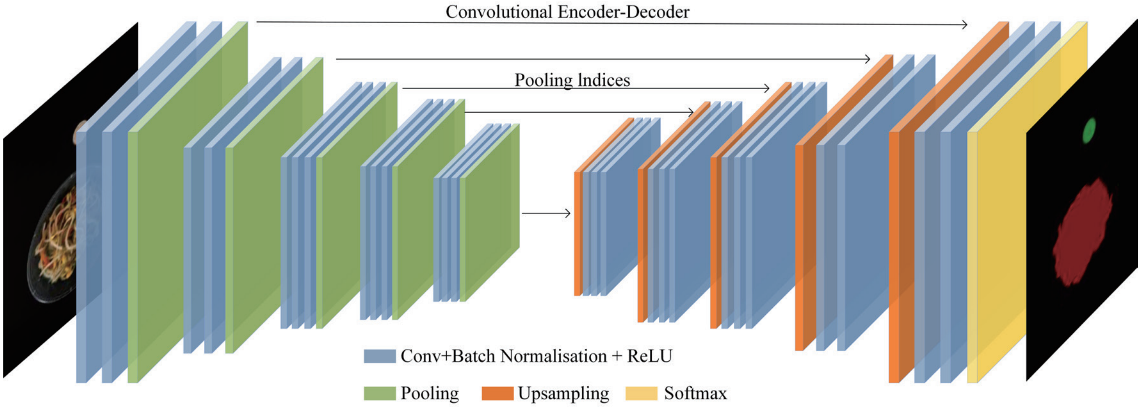

Convolutional Encoder-Decoder
Pooling Indices
Conv+Batch Normalisation + ReLU
Pooling
Upsampling
Softmax

Fig. 1. (Color online) Network architecture of SegNet.

points to note: the input image is first zero-padding, and sometimes the edge of the input matrix is filled with zero values so we can filter the edge of the input image matrix. The significant advantage of zero padding is that it allows us to control the size of the feature map. BatchNorm keeps the input of each layer of the neural network in the same distribution during the deep neural network training process so as to avoid the problem of gradient disappearance. The activation function uses the ReLU function.

The network parameters are shown in Table 1. We resize the input image to 416 × 416 × 3, and after a series of depth separable convolutions, it becomes a feature map of 26 × 26 × 512. It becomes 28 × 28 × 512 after the zeropad layer, then becomes 208 × 208 × 128 after three up-samplings, becomes 210 × 210 × 128 after the zeropad layer, and becomes 208 after two convolutions and a batch Norm layer × 208 × 22. After the reshape and softmax layers, it becomes 43264 × 22. Twenty-two is the predicted object type (20 types of food, background, and coins).

Regarding the composition of the loss function, we need to know the predicted and true values. The first is the prediction result. In the conv2d_4(Conv2D) layer, the output results are $height_i$, $width_i$, and $n_{classes}$, where $height_i$ represents the height of the input image and $width_i$ represents the width of the input image. Finally, the softmax function is used to calculate the probability of each category as a result of prediction. First, we resize the input image to the same size array as the predicted result, and then assign each pixel to its category in turn and store it in the array to obtain the real result. Finally, the cross entropy of the predicted and true results is calculated as the value of the loss function.

3.3 Semantic segmentation model based on UNet + MobileNet

UNet was firstly used in the field of medical image segmentation. It can be applied to data sets with a small amount of data and can achieve good segmentation results, so it is very popular. UNet mainly provides a set of data enhancement methods to maximize the use of data. The main structure of UNet is also similar to the convolutional encoder-decoder structure. The two parts of the encoder and decoder form a U-shaped network structure, as shown in Fig. 3.

Table 1
Network parameters of SegNet + MobileNet.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 416, 416, 3)	0
	...	
up_sampling2d_2 (UpSampling2D)	(None, 208, 208, 128)	0
zero_padding2d_3 (ZeroPadding)	(None, 210, 210, 128)	0
conv2d_3 (Conv2D)	(None, 208, 208, 64)	73792
batch_normalization_3 (BatchNorm)	(None, 208, 208, 64)	256
conv2d_4 (Conv2D)	(None, 208, 208, 22)	12694
reshape (Reshape)	(None, 43264, 22)	0
softmax (Softmax)	(None, 43264, 22)	0
	Total params: 5552918	

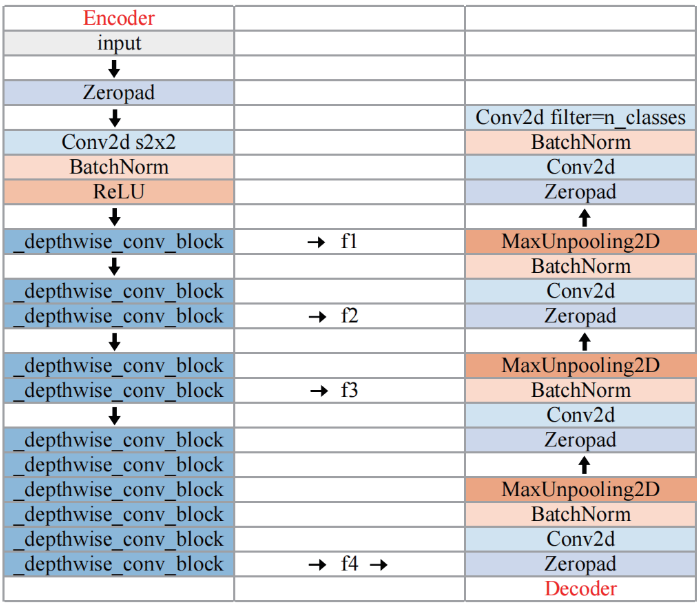


Fig. 2. (Color online) Semantic segmentation model of SegNet + MobileNet.

The encoder part of UNet is basically the same as the ordinary CNN network structure, using convolution and pooling and performing the conventional operation of extracting information between pixels. It is worth mentioning that the second half, the decoder part, is basically symmetrical in the form of the first half. Convolution is also used, but an up-sampling operation is introduced to output the segmentation result images of equal size. The most important thing is that the feature maps of each layer of the encoder are transmitted to those output by the decoder after being sampled by copying and appropriate cutting. They are connected to obtain more accurate context information and a different feature fusion method is used to improve the accuracy of segmentation.[13]

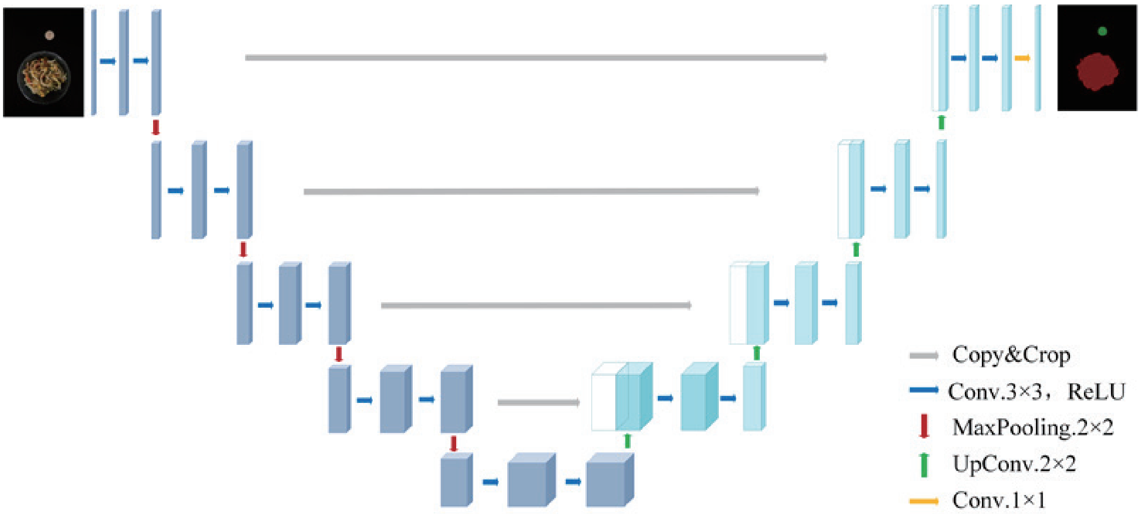
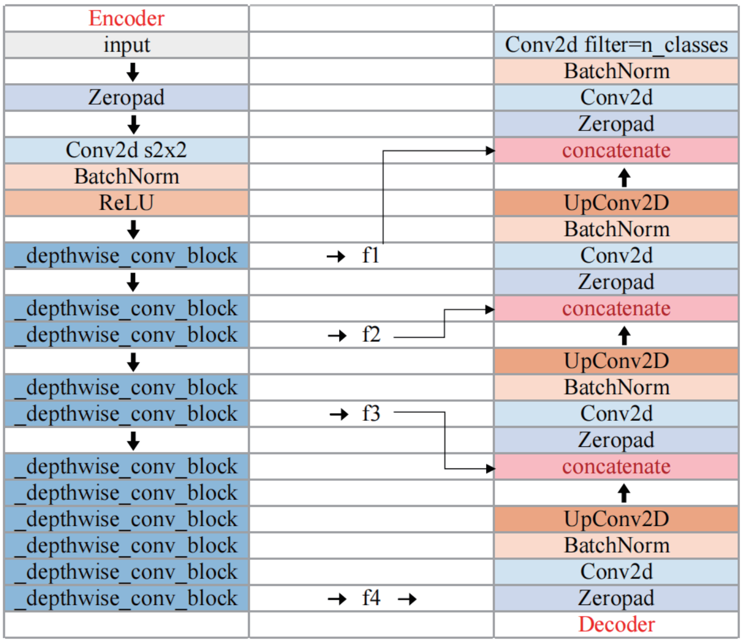


Fig. 3. (Color online) Network architecture of UNet.



Encoder		
input		Conv2d filter=n_classes
↓		BatchNorm
Zeropad		Conv2d
↓		Zeropad
Conv2d s2x2		concatenate
BatchNorm		↑
ReLU		UpConv2D
↓		BatchNorm
_depthwise_conv_block	→ f1	Conv2d
↓		Zeropad
_depthwise_conv_block		concatenate
_depthwise_conv_block	→ f2	↑
↓		UpConv2D
_depthwise_conv_block		BatchNorm
_depthwise_conv_block	→ f3	Conv2d
↓		Zeropad
_depthwise_conv_block		concatenate
_depthwise_conv_block		↑
_depthwise_conv_block		UpConv2D
_depthwise_conv_block		BatchNorm
_depthwise_conv_block		Conv2d
_depthwise_conv_block	→ f4 →	Zeropad
		Decoder

Fig. 4. (Color online) Semantic segmentation model of UNet + MobileNet.

The semantic segmentation network structure based on UNet + MobileNet constructed in this study differs from the network structure based on SegNet + MobileNet. It not only uses the feature layer compressed four times in the encoder but also uses those compressed twice and three times. The specific network structure is shown in Fig. 3. In the encoder, the backbone network is also MobileNet, and the input image is compressed four times after multiple deep separable convolutions, and the layers obtained after each compression are denoted as f1, f2, f3, and f4. f4 is up-sampled once and then concatenated with f3, then up-sampled again, then concatenated with f2, then up-sampled again, and then concatenated with f1. At last, the number of channels is the number of categories, and the semantically segmented result of the image is output through convolution operation.

The network parameters are shown in Table 2. Similarly, the input image is resized to 416 × 416 × 3. After a series of depth separable convolutions and one up-sampling, it becomes a 52 × 52 × 512 feature map, which is concatenated with f3 to obtain a 52 × 52 × 768 feature map. After Zeropad and BatchNorm layers and up-sampling, a 104 × 104 × 256 feature map is obtained, which is connected in series with f2 to obtain a 104 × 104 × 384 feature map. After up-sampling, a 208 × 208 × 128 feature map is obtained, and f1 is connected in series to obtain a feature map of 208 × 208 × 192. After the zeropad layer, it becomes 210 × 210 × 192. Next, after two convolutions and a batch Norm layer, it becomes 208 × 208 × 22. Finally, after the reshape and softmax layer, it becomes 43264 × 22. The loss function is consistent with the model loss function based on SegNet + MobileNet.

3.4 Semantic segmentation model based on PspNet + MobileNet

In image semantic segmentation, the global information obtained by the convolution operation of different receptive fields and the context semantic relationship strongly correlate

Table 2
Network parameters of UNet + MobileNet.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 416, 416, 3)	0	
	…		
up_sampling2d (UpSampling2D)	(None, 52, 52, 512)	0	batch_normalization[0][0]
concatenate (Concatenate)	(None, 52, 52, 768)	0	up_sampling2d[0][0] conv_pw_5_relu[0][0]
	…		
up_sampling2d_1 (UpSampling2D)	(None, 104, 104, 256)	0	batch_normalization_1[0][0]
concatenate_1 (Concatenate)	(None, 104, 104, 384)	0	up_sampling2d_1[0][0] conv_pw_3_relu[0][0]
	…		
up_sampling2d_2 (UpSampling2D)	(None, 208, 208, 128)	0	batch_normalization_2[0][0]
concatenate_2 (Concatenate)	(None, 208, 208, 192)	0	up_sampling2d_2[0][0] conv_pw_1_relu[0][0]
zero_padding2d_3 (ZeroPadding2D)	(None, 210, 210, 192)	0	concatenate_2[0][0]
conv2d_3 (Conv2D)	(None, 208, 208, 64)	110656	zero_padding2d_3[0][0]
batch_normalization_3 (BatchNorm)	(None, 208, 208, 64)	256	conv2d_3[0][0]
conv2d_4 (Conv2D)	(None, 208, 208, 22)	12694	batch_normalization_3[0][0]
reshape (Reshape)	(None, 43264, 22)	0	conv2d_4[0][0]
softmax (Softmax)	(None, 43264, 22)	0	reshape[0][0]
	Total params: 6327062		

with the generated error segmentation results. Therefore, the deep network appropriately pays attention to the scene features in the global scope, which helps to improve the accuracy of semantic segmentation significantly. PspNet belongs to such a type of network model. It is based on ResNet and FCN, uses multiscale feature fusion technology and pyramid pooling, and finally performs pixel-level segmentation prediction through convolution.

The pyramid pooling module extracts four characteristic regions of different scales.(14) The red block is a single bitmap generated after global average pooling, representing the roughest pyramid level. The other three pyramid levels actively divide the feature area into 2 × 2, 3 × 3, and 6 × 6 subregions. The finer the division, the more refined scene features can be mined. After the pooling operation is performed on each layer, features of different depths are obtained. Then, through 1 × 1 convolution, the feature dimensions are reduced and directly up-sampled to the same size as the shallow features. Next, the fusion's deep global features (context information) are combined with the shallow detailed features to obtain the final feature map. Finally, the final semantic segmentation prediction map is generated through a layer of the convolution operation.

As shown in Fig. 5, the encoder part of the semantic segmentation network based on PspNet + MobileNet constructed in this paper still uses the MobileNet network structure. The feature map is compressed five times through deep separable convolution, denoted as f5. In the decoder part, f5 is passed through four different average pooling layers of different lengths and sizes, and the result of pooling is adjusted by linear interpolation. Finally, the four resized feature maps are connected in series with f5. The product operation outputs an image whose number of channels is the number of categories, which is the result of semantic segmentation.

The network information is shown in Table 3. We resize the input image to 576 × 576 × 3, go through a series of depth separable convolutions, and compress it five times to obtain the feature map f5. The size is 18 × 18 × 1024. After f5 undergoes four different maximum pooling operations, as well as 1 × 1 convolution, BatchNorm, ReLU activation, and linear interpolation, we resize the operations, and four feature maps with a size of 18 × 18 × 512 are obtained and concatenated with f5. Then, after 1 × 1 convolution and 3 × 3 × n classes convolution, the resize

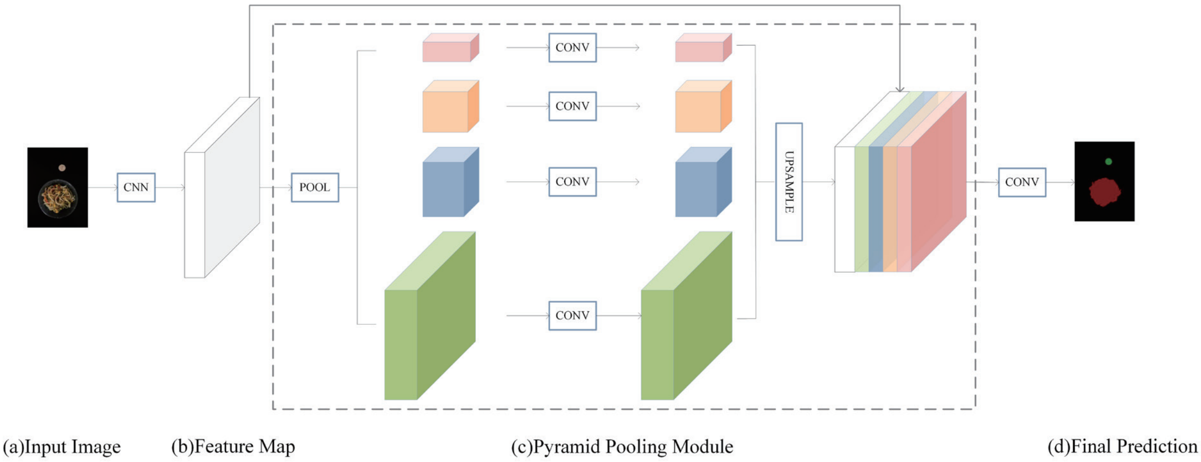


Fig. 5. (Color online) Network architecture of PspNet.

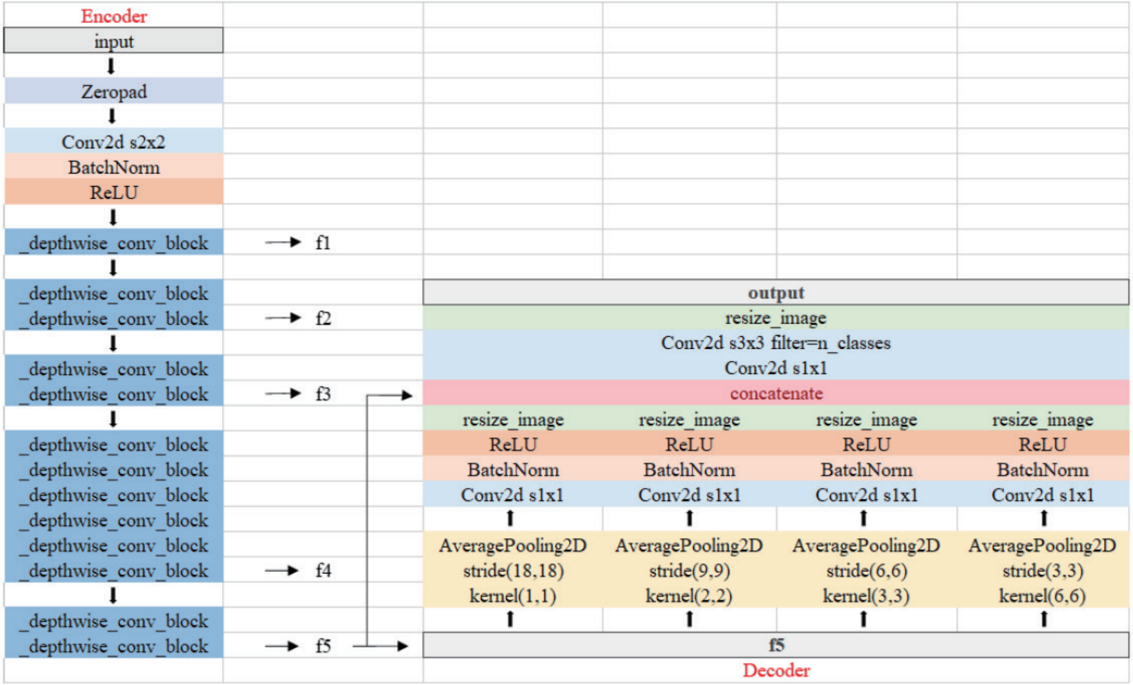


Fig. 6. (Color online) Semantic segmentation model of PspNet + MobileNet.

Table 3
Network parameters of PspNet + MobileNet.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 576, 576, 3)	0	
	…		
conv_pw_13_relu (Activation)	(None, 18, 18, 1024)	0	conv_pw_13_bn[0][0]
average_pooling2d (AveragePooling)	(None, 1, 1, 1024)	0	conv_pw_13_relu[0][0]
average_pooling2d_1 (AveragePooling)	(None, 2, 2, 1024)	0	conv_pw_13_relu[0][0]
average_pooling2d_2 (AveragePooling)	(None, 3, 3, 1024)	0	conv_pw_13_relu[0][0]
average_pooling2d_3 (AveragePooling)	(None, 6, 6, 1024)	0	conv_pw_13_relu[0][0]
	…		
lambda (Lambda)	(None, 18, 18, 512)	0	activation[0][0]
lambda_1 (Lambda)	(None, 18, 18, 512)	0	activation_1[0][0]
lambda_2 (Lambda)	(None, 18, 18, 512)	0	activation_2[0][0]
lambda_3 (Lambda)	(None, 18, 18, 512)	0	activation_3[0][0]
concatenate (Concatenate)	(None, 18, 18, 3072)	0	conv_pw_13_relu[0][0] lambda[0][0] lambda_1[0][0] lambda_2[0][0] lambda_3[0][0]
conv2d_4 (Conv2D)	(None, 18, 18, 512)	1572864	concatenate[0][0]
batch_normalization_4 (BatchNorm)	(None, 18, 18, 512)	2048	conv2d_4[0][0]
conv2d_5 (Conv2D)	(None, 18, 18, 22)	101398	activation_4[0][0]
lambda_4 (Lambda)	(None, 144, 144, 22)	0	conv2d_5[0][0]
reshape (Reshape)	(None, 20736, 22)	0	lambda_4[0][0]
softmax (Softmax)	(None, 20736, 22)	0	reshape[0][0]
	Total params:6327062		

operation obtains a feature map with a size of 144 × 144 × 22, and then the image passes through the reshape and softmax layers to become 20736 × 22. The loss function is inconsistent with those of the previous two models. When calculating the prediction result, we need to first change the number of feature map channels after serial output to the number of categories and use softmax to calculate the category probability after resizing.

3.5 Method of food calorie conversion

When constructing the data set, we also collect the food weight of the *i*-th training sample and obtain the caloric value, denoted by K_i, of the sample food according to the Chinese Food Nutrition Table.[15] In addition, the area ratio of the food to the coin of each sample food image can be obtained according to the label image, denoted by S_i. Therefore, the corresponding relationship between the area feature information of the food image and the caloric value of the food can be established, denoted by K_S. For n samples of the same class, the formula for calculating K_S of this type of food is

$$K_S = \frac{\sum_{i=1}^{n} K_i}{\sum_{i=1}^{n} S_i}. \tag{1}$$

Essentially, K_S represents the caloric value contained in a unit of food, which conforms to the law of nutrition. The area ratio of the realistic food to the coin is termed as S_r. According to the predicted food category and pixel area output by the model, the real caloric value, denoted by K_r, of the food in the image can be calculated as

$$K_r = S_r \times K_s. \tag{2}$$

4. Experiments and Implementation

4.1 Image acquisition

Since the distance and angle between the camera and the food cannot be guaranteed in the actual capturing process, it is necessary to borrow a standard reference object to construct the corresponding relationship with the food object to compensate for the loss of information caused by the shooting method. In this study, we seek the corresponding relationship between the information of the food area in the image and the calories so as to convert the calories on the basis of the area information. Since the area of the standard reference object is fixed, we plan to use image processing to calculate the ratio between the food area and the reference object area to calculate the food area. The food object and camera position can be fixed in actual application scenarios to avoid shooting technique problems.

In this study, we choose a common one yuan (CNY) coin with a uniform specification as the standard reference object. The coin is placed near the food and separated from the food area, and an image is taken of the coin and food together. After the capturing, the dish sample is weighed with an electronic scale and the weight of the sample is recorded. The experimental equipment needed in the data collection includes a 300 ml disposable lunch box for holding dish samples, standard reference objects (coins), black shading plates for shading, and an electronic scale for weighing dish samples, as shown in Fig. 7.

When shooting images, we use different models of mobile terminals to shoot and there are no strict restrictions on the constraints, and it can be carried out under different lighting conditions, angles, and coin placement environments. However, it is necessary to ensure that the food and coin are exposed to the lens simultaneously. It can be ignored if the area covered by the disposable lunch box does not exceed 10%. If the area covered by the coin and food exceeds 10% of its own, it will be considered invalid data and will not be included in the data set. We chose 20 familiar dishes for data collection.

An example of image data is shown in Fig. 8. The dishes taken are familiar dishes, including vegetarian and meat dishes, single foods, and combined dishes. Food images visually present the following features: shape features include granular, block, filament, flake, and combination; color features are mainly white, yellow, black, brown, green, red, and other color systems.

To calculate the actual caloric value of food, it is necessary to know the actual weight of each dish. Therefore, after each food image is taken according to the specifications, we use an

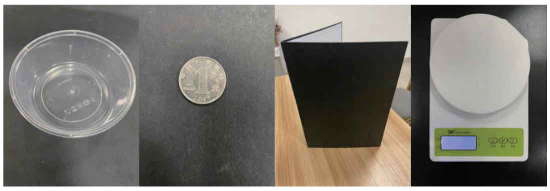


Fig. 7. (Color online) Tools for image acquisition.

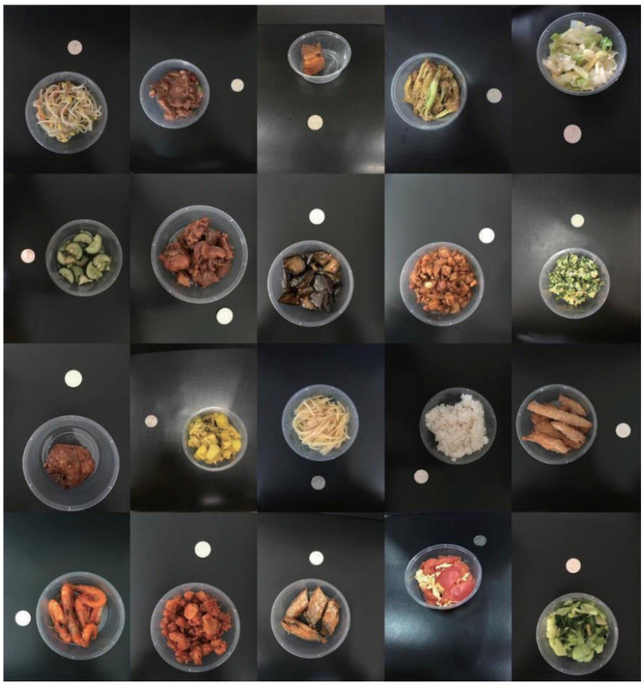


Fig. 8. (Color online) Twenty food samples.

electronic scale to weigh the dish, and the net weight of the dish (minus the weight of the standard disposable lunch box) is recorded. For the image generated by data enhancement, since the data enhancement does not change the area information of the food and coin in the image, the recorded weight is equal to the weight value of the original image.

4.2 Image preprocessing

The data reinforcement is used to obtain a certain amount of data for training. There are four image reinforcement techniques, namely, brightness adjustment, horizontal flip, vertical flip, and rotation. Figure 9 shows the physical effects of image reinforcement techniques.

After rigorous screening and data enhancement, 10000 image data were obtained. The Labelme labeling software is used to label all images, generate corresponding JSON labeling files, and generate corresponding labeled images through batch decompilation. After performing the corresponding classification, it is stored in the deep learning server. Figure 10 shows the original and labeled images.

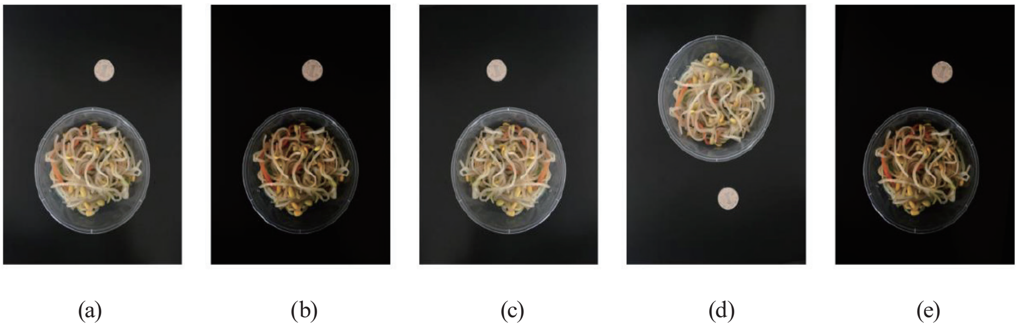


Fig. 9. (Color online) Image reinforcement. (a) Original image, (b) brightness adjustment, (c) horizontal flip, (d) vertical flip, and (e) rotation.

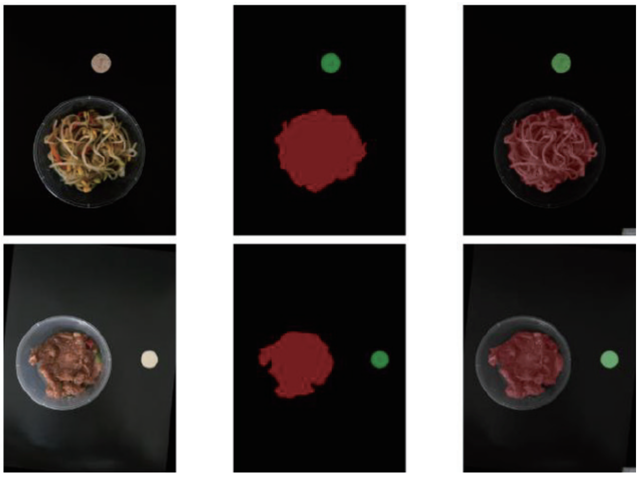


Fig. 10. (Color online) Original and labeled images.

4.3 Experimental setting

According to the ratio of 3:1:1, the data set is randomly divided into the training, validation, and test sets. Table 4 shows the distribution of the data set. During the experiment, we need to use the training set to train the food segmentation network model. To reduce the generalization error, we need to continuously train our model through the training set so that the model can learn more features and be closer to the real results. To prevent the over-fitting effect, the verification set is used in the training process. Therefore, the test set that does not participate in the training process is used to evaluate the final generalization ability of the model.

In the food calorie calculation task, calculating the area ratio of the food to the coin in the image is needed. The actual area ratio is denoted by Y_t and the predicted area ratio is denoted by Y_r. In this task, the error and accuracy are calculated as

$$Error = \frac{\left|Y_r - Y_t\right|}{Y_t}, \tag{3}$$

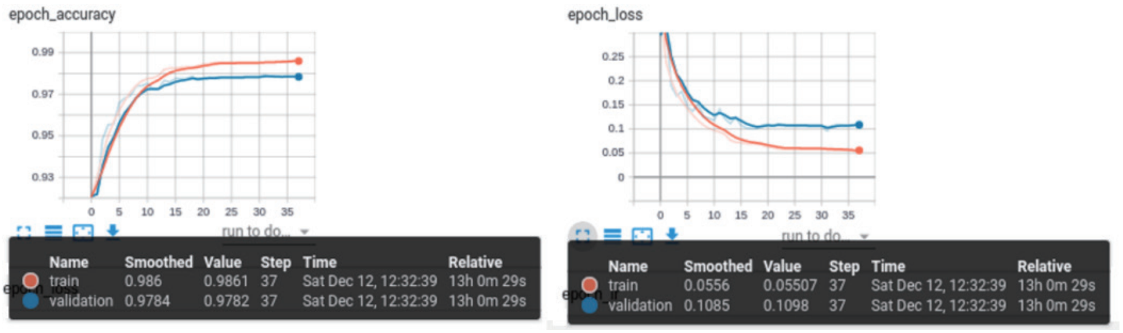
$$Accuracy = 1 - Error. \tag{4}$$

4.4 Performance

The training curves of the three models are shown in Figs. 11–13. The figures on the left are the accuracy curves of the training and validation sets during the training process, and the figures on the right are the loss curves. Because the early stopping mechanism is set, the PspNet model training stops at 16 steps, which does not affect the rationality of the training results.

Table 4
Distribution of data set.

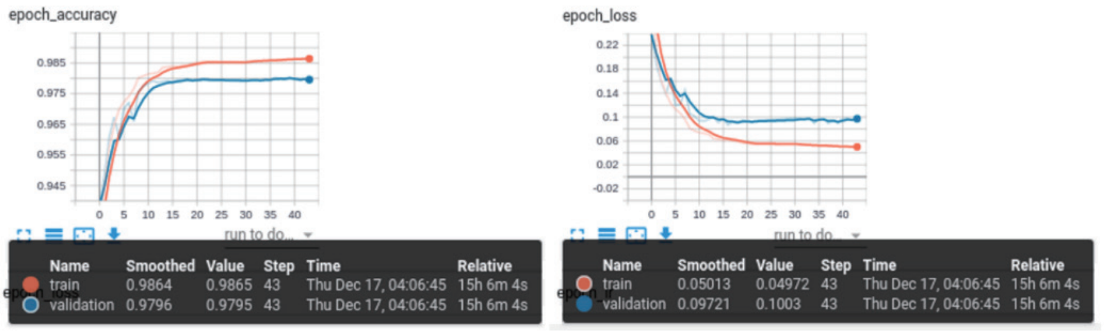
Data set	Images
Training set	6000
Validation set	2000
Test set	2000



Name	Smoothed	Value	Step	Time	Relative
● train	0.986	0.9861	37	Sat Dec 12, 12:32:39	13h 0m 29s
● validation	0.9784	0.9782	37	Sat Dec 12, 12:32:39	13h 0m 29s

Name	Smoothed	Value	Step	Time	Relative
● train	0.0556	0.05507	37	Sat Dec 12, 12:32:39	13h 0m 29s
● validation	0.1085	0.1098	37	Sat Dec 12, 12:32:39	13h 0m 29s

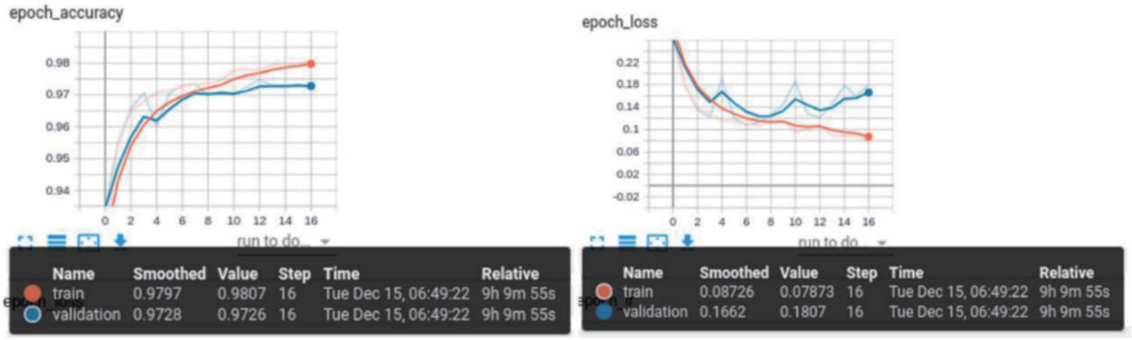
Fig. 11. (Color online) Training curve of SegNet + MobileNet.



Name	Smoothed	Value	Step	Time	Relative
train	0.9864	0.9865	43	Thu Dec 17, 04:06:45	15h 6m 4s
validation	0.9796	0.9795	43	Thu Dec 17, 04:06:45	15h 6m 4s

Name	Smoothed	Value	Step	Time	Relative
train	0.05013	0.04972	43	Thu Dec 17, 04:06:45	15h 6m 4s
validation	0.09721	0.1003	43	Thu Dec 17, 04:06:45	15h 6m 4s

Fig. 12. (Color online) Training curve of UNet + MobileNet.



Name	Smoothed	Value	Step	Time	Relative
train	0.9797	0.9807	16	Tue Dec 15, 06:49:22	9h 9m 55s
validation	0.9728	0.9726	16	Tue Dec 15, 06:49:22	9h 9m 55s

Name	Smoothed	Value	Step	Time	Relative
train	0.08726	0.07873	16	Tue Dec 15, 06:49:22	9h 9m 55s
validation	0.1662	0.1807	16	Tue Dec 15, 06:49:22	9h 9m 55s

Fig. 13. (Color online) Training curve of PspNet + MobileNet.

Through the comparison of the training processes of the three models, we found that the models SegNet + MobileNet and UNet + MobileNet models are similar in convergence and far faster than the PspNet + MobileNet model. This means that these two models are more efficient in finding a more accurate solution.

Table 5 shows the accuracies of the three models in food recognition. In the food recognition, the UNet + MobileNet model has the highest accuracy in the test set, reaching 0.9795. The gap between the SegNet + MobileNet and UNet + MobileNet models is small (only 0.0013). The PspNet + MobileNet model has the lowest accuracy of 0.9726. However, the accuracy differences of the three models in the test set are not significant and all reach more than 0.97.

Figures 14–16 show the semantic segmentation results obtained using the three models. After the segmentation, we mark the divided coin and dish areas with different colors in Figs. 14–16. For each image, the contour extraction and area calculation are used to obtain the coin and food areas. Therefore, the food-to-coin area ratio can be obtained. The actual size of the coin is known and that of the dish can be estimated. Moreover, the predicted food-to-coin area ratios are shown in Table 6.

In calculating the area ratio of the food to the reference object in the image, it can be found that the accuracy of the SegNet + MobileNet model is the highest and much higher than those of the other two models. Combined with the model performance results in semantic segmentation,

Table 5
Food recognition accuracies of the three models.

Model	Train_accuray	Val_accuray
SegNet + MobileNet	0.9861	0.9782
UNet + MobileNet	0.9865	0.9795
PspNet + MobileNet	0.9807	0.9726

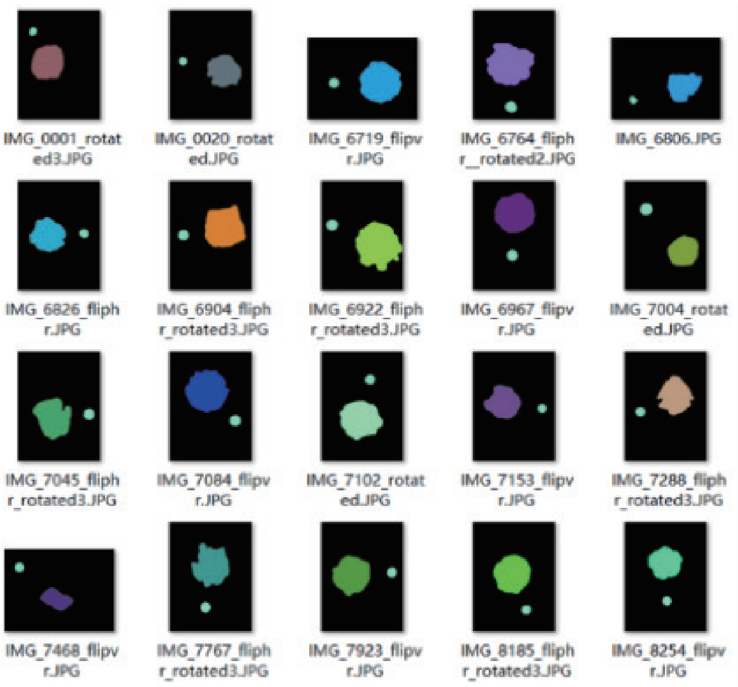


Fig. 14. (Color online) Segmentation result of SegNet + MobileNet.

we considered using the SegNet + MobileNet model for food recognition and regional area extraction. Although the performance of the UNet + MobileNet model is the best, that of the SegNet + MobileNet model is very close to it. The label map effect segmented by the SegNet + MobileNet model also does not show glitches or loss of feature information. The most important thing is the area ratio calculation. The accuracy rate is the highest in Table 7.

4.5 Implementation of APP

For the health management of diabetic patients, the APP for food calorie estimation is designed and implemented. At this stage, only Android smartphones are supported. According to the experimental result, the trained SegNet + MobileNet model is selected to be deployed in the APP. Figure 17 shows how to perform food calorie estimation. The test results of 24 dishes are shown in Fig. 18.

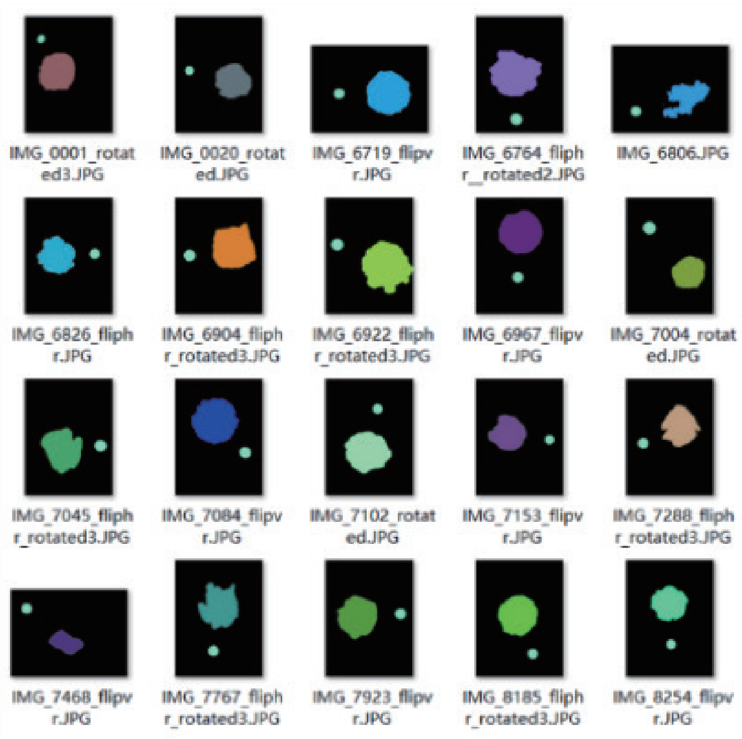


Fig. 15. (Color online) Segmentation result of UNet + MobileNet.

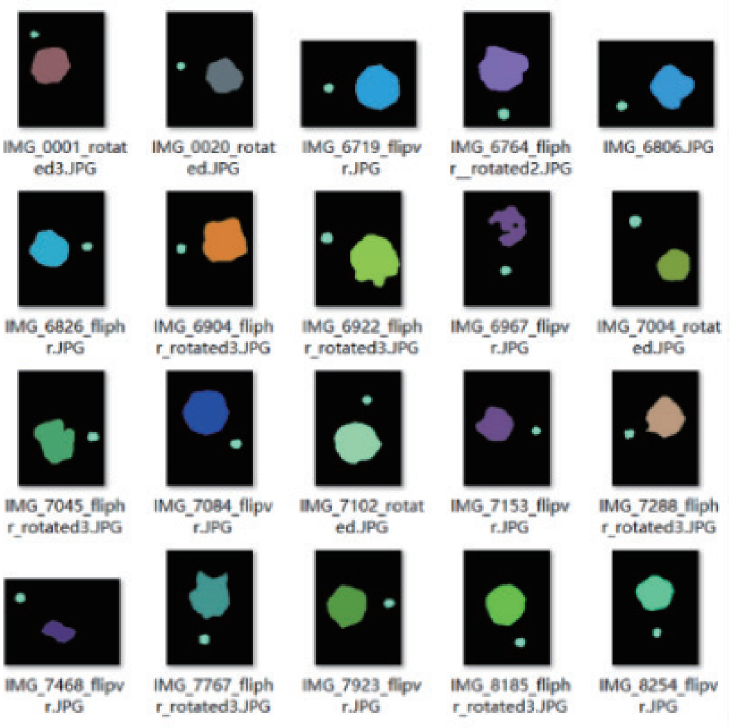


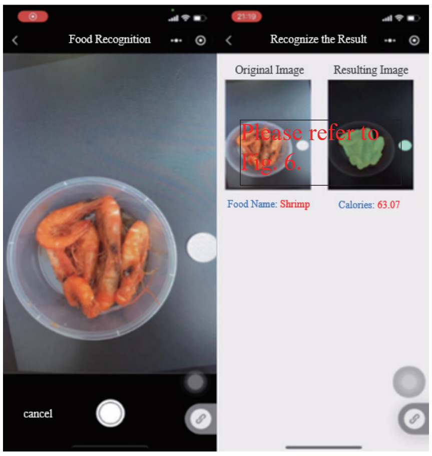
Fig. 16. (Color online) Segmentation result of PspNet + MobileNet.

Table 6
Y_r predicted area ratios of the labeled food to the coin.

Image	SegNet + MobileNet	UNet + MobileNet	PspNet + MobileNet	Y_t
IMG_0001_rotated3	18.8338	22.1897	22.2000	24.1344
IMG_0020_rotated	16.6019	14.0230	13.6722	17.1314
IMG_6719_flipvr	12.7678	12.9190	15.8785	19.4562
IMG_6764_fliphr_rotated2	13.5251	13.7769	13.1018	19.3455
IMG_6806	16.7430	10.8600	13.3545	16.9573
IMG_6826_fliphr	10.6119	9.3626	12.9407	11.6529
IMG_6904_fliphr_rotated3	13.8999	13.2196	15.1380	16.1179
IMG_6922_fliphr_rotated3	15.5187	16.1185	15.7356	18.7903
IMG_6967_flipvr	13.4660	15.7838	7.2804	18.2252
IMG_7004_rotated	6.6701	6.7683	6.4021	7.2573
IMG_7045_fliphr_rotated3	11.5446	11.6430	12.6867	13.5620
IMG_7084_flipvr	13.7491	14.3463	14.8088	17.3780
IMG_7102_rotated	15.8843	17.6040	11.2742	17.9554
IMG_7153_flipvr	9.6382	9.3930	10.7954	13.9957
IMG_7288_fliphr_rotated3	12.9061	11.7559	13.5358	13.4654
IMG_7468_flipvr	4.9167	4.7281	4.9601	6.1282
IMG_7767_fliphr_rotated3	13.5097	13.4403	13.5916	14.8487
IMG_7923_flipvr	10.6331	10.1943	16.3856	13.5984
IMG_8185_fliphr_rotated3	16.4263	15.3953	13.6127	16.5103
IMG_8254_flipvr	12.1299	12.0678	12.9722	14.2298

Table 7
Average errors and accuracies of the three models.

Model	Error	Accuracy
SegNet + MobileNet	0.1505	0.8495
UNet + MobileNet	0.1712	0.8288
PspNet + MobileNet	0.1748	0.8252

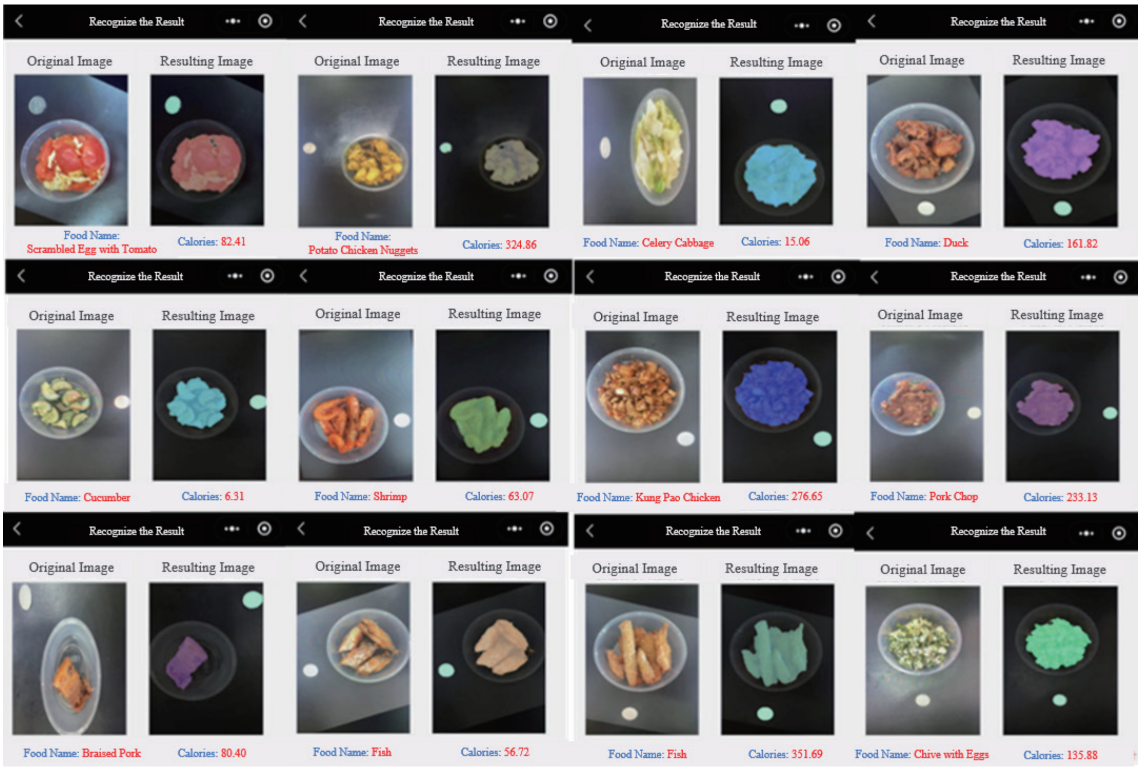


Please refer to Fig. 6.

Food Name: Shrimp

Calories: 63.07

Fig. 17. (Color online) Food calorie estimation APP for diabetic patients.



Food Name: Scrambled Egg with Tomato Calories: 82.41

Food Name: Potato Chicken Nuggets Calories: 324.86

Food Name: Celery Cabbage Calories: 15.06

Food Name: Duck Calories: 161.82

Food Name: Cucumber Calories: 6.31

Food Name: Shrimp Calories: 63.07

Food Name: Kung Pao Chicken Calories: 276.65

Food Name: Pork Chop Calories: 233.13

Food Name: Braised Pork Calories: 80.40

Food Name: Fish Calories: 56.72

Food Name: Fish Calories: 351.69

Food Name: Chive with Eggs Calories: 135.88

Fig. 18. (Color online) Test results of 24 dishes on a smartphone.

5. Conclusions

In this paper, we proposed a novel approach based on food image recognition and calorie conversion. It mainly elaborates on the establishment of a food image data set and the specific methods of image recognition and calculation of food calories based on food images. We constructed a model based on deep learning to predict the classification and segmentation of food images. Therefore, we calculated the food area via the image segmentation. In addition, we determined the corresponding relationship between the unit area of the food image and the food calorie value using various dishes. The experimental results showed that the semantic segmentation performance of SegNet + MobileNet is 0.9782 and that the accuracy of calculating the area ratio of the food to the reference object in the food calorie recognition task can reach 0.8495.

On this basis, we have also designed and implemented an APP for diabetic patients. This APP provides a function to calculate the food calories via a smartphone. In the future, more types of food information will be collected in more scenarios, and we will develop a method of calculating food volume without a reference object.

Acknowledgments

This work was partially supported by Changzhou College of Information Technology under Grant No. 2019KYQD03.

References

1 E. Standl, K. Khunti, T. B. Hansen, and O. Schnell: Eur. J. Prev. Cardiol. **26** (2019) 7. https://doi.org/10.1177/2047487319881021

2 V. Basevi: Diabetes Care **34** (2011) S62. https://doi.org/10.2337%2Fdc11-S062

3 P. McAllister, H. Zheng, R. Bond, and A. Moorhead: Comput. Biol. Med. **95** (2018) 217. https://doi.org/10.1016/j.compbiomed.2018.02.008

4 I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda: Comput. Struct. Biotechnol. J. **15** (2017) 104. https://doi.org/10.1016/j.csbj.2016.12.005

5 I. Contreras and J. Vehí: J. Med. Internet Res. **20** (2018). https://doi.org/10.2196/10775

6 E. Daskalaki, P. Diem, and S. Mougiakakou: PloS One **11** (2016) 7. https://doi.org/10.1371/journal.pone.0158722

7 S. Norouzi, A. K. Ghalibaf, S. Sistani, V. Banazadeh, F. Keykhaei, P. Zareishargh, F. Amiri, M. Nematy, and K. Etminani. Arch. Iran Med. **2110** (2018) 466. http://journalaim.com/Article/aim-4210

8 S. Fang, Z. Shao, D. A. Kerr, C. J. Boushey, and F. Zhu: Nutrients **11** (2019) https://doi.org/10.3390/nu11040877

9 D. A. Schoeller: Metabolism **44** (1995) 18. https://doi.org/10.1016/0026-0495(95)90204-X

10 F. Zhu, M. Bosch, C. J. Boushey, and E. J. Delp: IEEE Int. Conf. Image Processing (IEEE, 2010) 1853. https://doi.org/10.1109/ICIP.2010.5650848

11 A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam: arXiv (2017) https://doi.org/10.48550/arXiv.1704.04861

12 V. Badrinarayanan, A. Kendall, and R. Cipolla: IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2017) 2481. https://doi.org/10.1109/tpami.2016.2644615

13 O. Ronneberger, P. Fischer, and T. Brox: Int. Conf. Medical Image Computing and Computer-Assited Intervention-MICCAI 2015 **9351** (2015) 234. https://doi.org/10.1007/978-3-319-24574-4_28

14 H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia: IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2017) 6230. https://doi.org/10.1109/CVPR.2017.660

15 Chinese Center for Disease Control and Prevention: Chinese Food Nutrition Table (Peking University Medical Press, 2018).

About the Authors

Xiang-Yong Kong received his master's degree in biomedical engineering from Zhejiang University, China. He is currently a lecturer with the School of Health Science and Engineering, University of Shanghai for Science and Technology, where he is leading the Info Science Laboratory. His research interests include machine learning, artificial intelligence in medicine, intelligent healthcare, and their application in the service and manufacturing industries. He has cooperated with many hospitals and governments in China, presided over many research projects, and launched more than ten products. His research papers have been published in Pattern Recognition Letters and other professional journals. (kxy@usst.edu.cn)

Xiao-Han Sun received her bachelor's degree in biomedical engineering in 2019. She is currently studying for a master's degree in electronic information at the School of Health Medicine and Engineering, University of Shanghai for Science and Technology. Her current research interests include deep learning, medical artificial intelligence, medical image processing, and medical information engineering. Currently, as a member of the Medical Information Research Institute, she is engaged in the research on medical information engineering and pathological and ultrasonic image processing. She has won individual and team awards and scholarships many times with excellent performance. (sxh916651551@163.com)

Yu-Ze Wang is an undergraduate student majoring in medical information engineering at the University of Shanghai for Science and Technology. As a member of the University of Shanghai for Science and Technology, he studies and engages in smart medical software development and project management. Recently, his team has organized and developed a children's health management system, which won first prize in the Chinese college students' computer design competition. His research interests include medical image processing, software engineering, and their applications in medical services and hospital information sectors. (wang_yz0613@163.com)

Rui-Yang Peng is a graduate student majoring in electronic information (biomedical engineering) at the University of Shanghai for Science and Technology. Her research direction is intelligent medical care and big clinical data. Currently, the Institute of Medical Informatics is engaged in big data analysis, data mining, and software development. Her research interests include intelligent medical care, big data technology, machine learning, and other related fields. At the same time, she has participated in and won many competitions during her school days, including winning the national second prize in the China Postgraduate Mathematical Modeling Competition and also the school second prize scholarship. (peng_ruiyang@163.com)

Xin-Yue Li is an undergraduate student majoring in medical information engineering at the University of Shanghai for Science and Technology. She studies and participates in intelligent medical software development and project management and has won individual and team awards. Her research interests are mainly in medical information engineering, software engineering, and artificial intelligence. (2593045307@qq.com)

Ying-Rui Lv is an undergraduate student majoring in business administration at the University of Shanghai for Science and Technology. She studies and participates in intelligent medical project management. Her research interests are mainly in corporate governance, programming and artificial intelligence. (771963328@qq.com)

Yi-Heng Yang is an undergraduate student majoring in rehabilitation engineering at the University of Shanghai for Science and Technology. He studies and participates in rehabilitation system development and project management and has won team awards. His research interests are mainly in neural engineering, mechanical engineering, and artificial intelligence. (yyh20030516@163.com)

Shih-Pang Tseng received his B.S. and M.S. degrees from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, and his Ph.D. degree from the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. He has been a professor in the Department of Computer Science and Engineering, Tajen University, Pingtung, Taiwan. At present, he is a professor in the School of Software and Big Data, Changzhou College of Information Technology, Changzhou, China, and the School of Information Science and Technology, Sanda University, Shanghai, China. His current research interests include artificial intelligence, learning technology, Internet of Things, and robotics.