Image-Based Estimation of Real Food Size for Accurate Food Calorie Estimation

Takumi Ege, Yoshikazu Ando, Ryosuke Tanno, Wataru Shimoda and Keiji Yanai
Department of Informatics, The University of Electro-Communications, Tokyo
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585 JAPAN
{ege-t,ando-y,tanno-r,shimoda-k,yanai}@mm.inf.uec.ac.jp

Abstract

In this paper, we review our works on image-based estimation of real size of foods for accurate food calorie estimation which including three existing works and two new works: (1) "CalorieCam" which is a system to estimate real food size based on a reference object, (2) Region segmentation based food calorie estimation, (3) "AR DeepCalorieCam V2" which is based on visual inertial odometry built in the iOS ARKit library, (4) "DepthCalorieCam" which employs stereo cameras on iPhone X/XS, and (5) "RiceCalorieCam" which exploits rice grains as reference objects. Especially, the last two new methods achieved 10% or less estimation error, which was enough for robust food calorie estimation.

1 Introduction

In food image recognition, CNN-based methods have achieved great improvement in recent years and some smartphone applications employ them. However, in most of the calorie estimation, the estimated calories are just associated with the estimated food categories and these applications often require users to enter information such as size or volume, there are problems that it is a troublesome and subjective evaluation. Currently, no applications which can estimate food calories automatically exist.

Although most of the image recognition tasks including food category recognition have been almost solved due to great progress of CNN-based image recognition methods, fully-automatic food calorie estimation from a food photo has still remained an unsolved problem. We think that food calorie estimation not only helps people's health a lots, but also is promising as a new problem of image recognition studies.

In this paper, we review three of our works related to food size estimation, and propose two new methods for food size estimation.

(1) "CalorieCam" [1] which is a reference-object-based food calorie estimation system.
(2) Weakly-supervised segmentation based food calorie estimation [2].
(3) "AR DeepCalorieCam V2" [3] which is a real food size and calorie estimation system based on visual inertial odometory implemented in iOS ARKit.
(4) "DepthCalorieCam" (new system) which is a food calorie estimation system exploiting iPhone stereo cameras.
(5) Rice grain based size estimation (new method) which uses rice grained the size of which are usually almost the same as a reference object.

2 CalorieCam

"CalorieCam" [1] is an image-based calorie estimation system which estimate food calories automatically by simply taking a food photo from the top with a pre-registered reference object. Taking into account usability and mobility of a system, we implemented the system as a stand-alone mobile application running on a normal Android smartphone.

To estimate food calorie from a single image, a user needs to register a size-known reference object in advance and to take a food photo with the registered reference object. As a reference object, we assume a personal belonging which we are always carrying such as a wallet and a credit-card-size card in a wallet (except a smartphone for taking a meal photo). After taking a meal photo with a reference object, the system carries out segmentation of food items and the pre-registered reference object. Because the real size of the reference object is known (e.g. In case of a credit-card-size object, the size is 85.6mm x 54mm.), the system can estimate the real size of each detected food items by comparing the number of pixels of the reference objects and the detected food items. By using the estimated real size and the equations to calculate food calorie from their size, the system finally estimates the calorie of the food items in the real photo. Figure 1 shows an example of usage and a screen-shot of the proposed system.

978-1-7281-1198-8/19/$31.00 ©2019 IEEE
DOI 10.1109/MIPR.2019.00056

Figure 1. (Upper) A user taking a food photo with the reference object. (Lower) A screen-shot.

To extract regions of food items and a reference object, firstly we estimate rough position of dishes based on edge detection results, and secondly we apply color-pixel-based k-means clustering for estimating bounding box of food regions. Finally we apply GrabCut [4] with the detected bounding box as a segmentation seed. For food classification, we use a Convolutional Neural Network (CNN) based recognition engine which run on a consumer smartphone [5] with high accuracy. It takes only 0.2 seconds on a consumer Android phone.

3 Weakly-supervised Segmentation Based Calorie Estimation

In the previous work, we assumed that one meal photo contains only one dish. This means we have to recognize dishes one by one for the case of multiple dishes. Then, in this work [2], we introduced CNN-based region proposal to cope with multiple-dish meals. We estimate calories from segmentation results by taking account of area ratios of multiple foods. With our method, we can estimate food calories for a photo without multiple-view photos and specific reference objects such as wallets and cards.

The proposed method on CNN-based region detection consists of the following steps as shown in Fig.2:

- Apply selective search [6] and obtain 2000 bounding box proposals at most.
- Group them and select bounding boxes.
- Perform back propagation over the pre-trained CNN regarding all the selected bounding boxes.
- Obtain saliency maps by averaging BP outputs within each group.

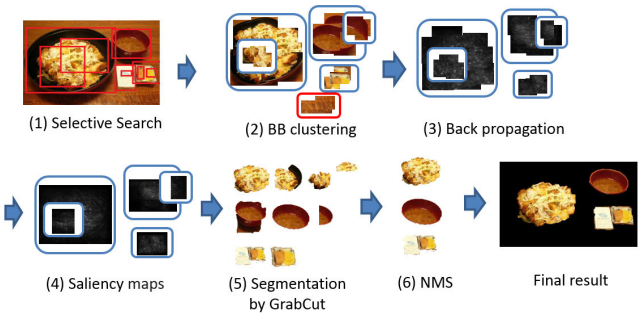


Figure 2. The processing flow of the proposed method.

- Extract segments based on the saliency maps with GrabCut [4].
- Apply non-maximum suppression (NMS) to obtain final region results.

After region segmentation, we choose a base food region the real size of which is expected to the same as the standard dish in the database from the segmentation results of multiple foods. We do not fix the category of a base food so that there are a lot of pattern in food combination. Therefore, we define the priorities of food items for choosing a base food class. We decide the priorities based on a tendency of unchanging food volumes. Some food volumes change frequently, while some foods volume rarely change. For example, in "Teishoku" which is Japanese traditional food combo menu, we can often change the volume of "rice" as options, while we cannot change "miso-soup" volume in general. In this case, "miso-soup" is more appropriate for a base food. In this way, there are difference in tendency of unchanging food volumes and we defined food ranking the volume of which are relatively stable as a high priority class for choosing a base food class. After selecting a base food the actual size and calorie is assumed to be known, we calculate calories of other dishes than the dish of the base food by comparing the number of pixels between each dishes and the dishes of the base food taking account of calorie densities which depends on food categories.

Fig.3 shows some examples of calorie estimation results.

4 AR DeepCalorieCam

The previous two method assumed to use only a single photo for estimating food calories. Being different from them, "AR DeepCalorieCam V2" [3] uses an inertial sensor built in a standard smartphone in addition to photos.

The iOS ARKit library contains a function of visual inertial odometory which can estimate a real size of objects

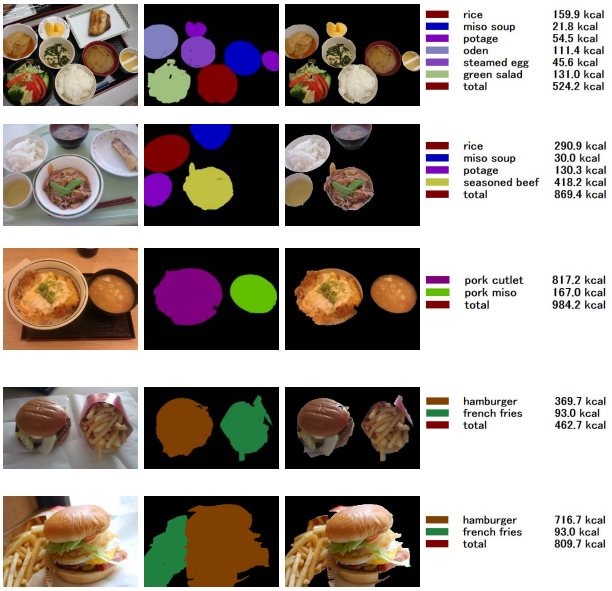


Figure 3. Examples for results of calorie estimation on UEC-FOOD100.

in the scene. By taking advantage of it, we can directly get to know the real size of foods without size-known reference objects.

The system outline is as shown in Figure 4. The processing of the proposed system consists of the following two steps:

- Recognize a category of each food item.
- Directly calculate the size of the region of the food item of food items using AR, and then calculate food calories based on their actual size and the calorie estimation curves defined depending on food categories.

We assume that the height of food portion correlates with the size of foods and food categories, and we estimate calories of food items directly from the food sizes estimated from the top-view image. To do that, we use not simple linear estimation but quadratic curve estimation from the 2D size of foods to their calories. The quadratic curve of each food category is estimated based on the training data annotated with real food calories independently. We use a quadratic curve estimation from the 2D size of foods to their calories. In the case of the proposed method, we can calculate the real size of foods area directly using AR technology on mobile devices such as iPhone. Therefore, the reference object, which was conventionally necessary to obtain the actual meal area, became unnecessary. By using Apple ARKit framework, we can measure the actual size of the meal area

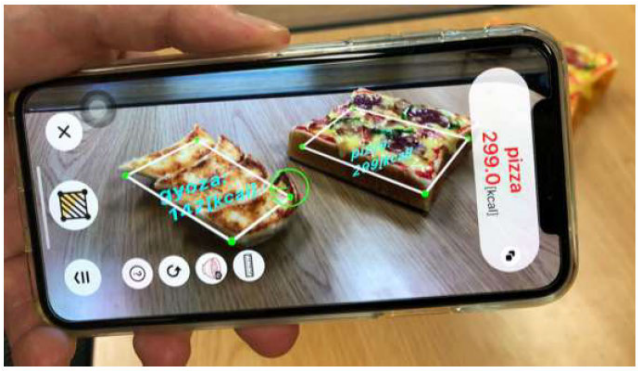


Figure 4. AR DeepCalorieCam V2.

by acquiring the coordinates on the real world as a three-dimensional vector and then calculate food calories based on their actual size and food categories.

5 DepthCalorieCam

The recent iPhones including iPhone 7s Plus/8s Plus/X/XS/XS Max have two cameras on the backside, which can be used as stereo cameras. The latest iOS has a function on real-time depth image acquisition by using two backside cameras as stereo cameras. Since the baseline distance of two backside cameras are known, based on triangulation we can get to know actual sizes of objects in the depth image.

We implemented a new application running on iPhone having two backside cameras for food calorie estimation, "DepthCalorieCam". Figure 5 shows acquired RGB and depth images and the results of calories estimation by DepthCalorieCam.

The processing flow is as follows:

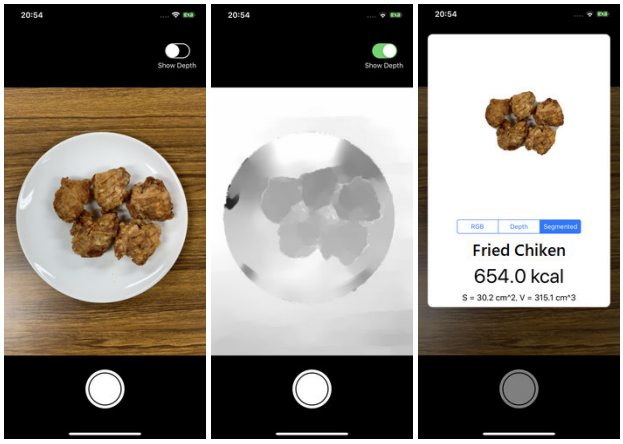
1. Food regions are extracted by U-NET [7] trained with UECFood-100 [8] and segmentation mask sets.

2. Estimate the depth of each of the pixels corresponding to food regions and their actual volumes.

3. Calculate food calories from the actual volume and the regression curves associating with volumes and calories.

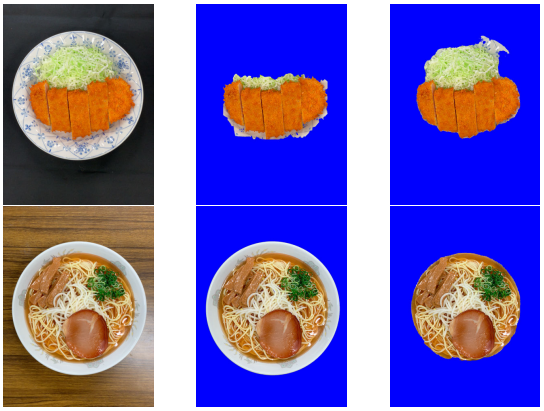
5.1 CNN-based food region segmentation

We used U-Net [7] for food region segmentation. For training of U-Net, we used the newly-created 5,301 segmentation masks on the images of UECFOOD-100 [8]. In our previous work, we adopted weakly-supervised food region segmentation, because we have no segmentation mask



An obtained RGB image. An obtained depth image. Estimated result with calorie intake and segmented food region.

Figure 5. DepthCalorieCam.



(a) (b) (c)

Figure 6. Segmentation results. (a) input images. (b) results by GrabCut (c) results by U-Net.

Pork with sweet and sour source. Fried chicken. Croquette.

Figure 7. The foods used for the experiments.

dataset for food image. This caused low performance of segmentation-based methods. Then, we annotated segmentation masks for about half of the images of UECFOOD-100 [8] for training of food segmentation CNN. As a results, the performance of food region segmentation has been much improved. In case of using 4771 for training and 530 for testing, the mIoU (mean Intersecting over Union) of test samples was 0.800. The examples of segmentation results by GrabCut [4] used in CalorieCam and U-Net used in DepthCalorieCam are shown in Figure 6.

5.2 Comparative Experiments with CalorieCam and AR DeepCalorieCam V2

In this experiments, we made use studies for comparing the proposed DepthCalorieCam with two previous works: CalorieCam [1] and AR DeepCalorieCam V2 [3]. In the experiments, we used three kinds of foods the images of which are shown in Figure 7 and the calories of which are shown in Table 1.

Three subjectives estimated calorie values of all the three foods with three systems, CalorieCam, AR DeepCalorieCam V2, and DepthCalorieCam five times with each of the conditions. Table 2 shows the average error and standard deviation. From these results, apparently DepthCalorieCam has improved accuracy of food calorie estimation greatly over two previous systems.

6 Rice grain based size estimation

In this work, we propose CNN-based real size estimation from boiled rice grains images. We use grains of boiled rice as a reference object and construct a CNN-based method that input a patch image of boiled rice grains images and output real size of a side of an input patch image. Figure 8 shows examples of real size annotated boiled rice photos, and Figure 9 shows the architecture of our network.

The architecture of our network is based on VGG16 [9]. As shown in the Figure 9, it has an output layer composed of a single unit that outputs a real size. An input image is a patch image cropped from boiled rice grains image and the output is a real size of a side of the input patch image. In this work, since the input patch image is 244×244, the output the real size for 224 pixels.

The real size estimation is treated as a regression problem. Generally, in the regression problem, a mean square error is used as the loss function. In this work, we use the mean square error shown in the Equation (1).

$$L = \frac{1}{n}\sum_{k=1}^{n}(x_k - y_k)^2 \quad (1)$$

6.1 Dataset

In this work, we construct a real size annotated boiled rice photo dataset for training and evaluation of our

Table 1. The real calorie values of three foods used for the experiments.

category	calorie [kcal]
pork with source	500
fried chicken	655
croquette	246

Table 2. Comparison on calorie estimation error (Avg.±SD [kcal]) among CalorieCam [1], AR CalorieCam V2 [3] and DepthCalorieCam.

category	CalorieCam	AR CalorieCam	DepthCalorieCam
pork with source	364±552	-112±163	2±52
fried chicken	-123±171	343±51	-5±64
croquette	-48±16	-104±12	-35±22

Figure 8. Examples of real size annotated rice photos.

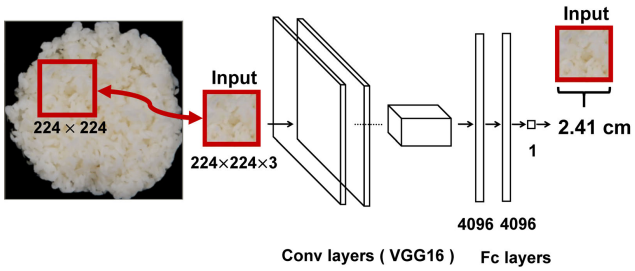


Figure 9. Our network that estimate a real size from boiled rice grains images.

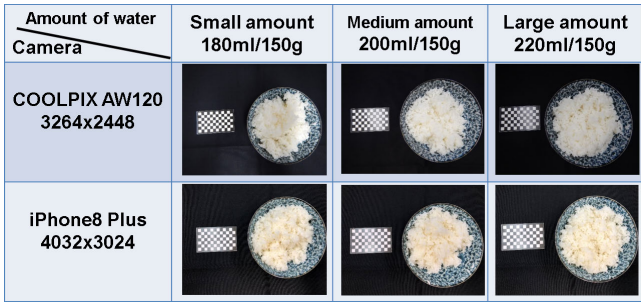


Figure 10. Examples of boiled rice images with an segmentation mask.

network. We prepare 2 types of camera (COOLPIX AW120, iPhone8 Plus) and 3 amount of water for boiling rice (180ml/150g, 200ml/150g, 220ml/150g). In addition, a distance between camera and rice is changed for each image and shape of rice is also changed for every five images. Since 60 images are taken for each combination of 2 cameras and 3 amount of water, a total of 360 images are collected. Figure 10 shows the collected boiled rice images. All images are annotated a real size per pixel based on a diameter of a dish and a segmentation mask of boiled rice for delete background. Figure 8 shows boiled rice images with an annotated segmentation mask.

6.2 Experiments

In this experiment, we estimate a real size from boiled rice grains image. We split the built dataset into 6 for the combination of 2 cameras and 3 amount of water, and one is used for evaluation and the rest is used for training, so we conduct experiments for 6 combinations of train data and evaluation. For each experiment, 300 images are used for training and 60 images for evaluation in the built dataset.

For training of our network, an input patch image are cropped from a random position of a boiled rice grains image and the output is a real size of a side of the input patch image. On the other hand, for evaluation of our network, 16 patch images are cropped from a boiled rice grains image by 4×4 grid sampling and output is the average value of the output real size of these patch images. For both training and evaluation, input patch images where the background region occupies more than 50% are eliminated. Figure 11 and Figure 12 shows the input patch images in training and evaluation respectively.

The architecture of our network is based on VGG16 [9] and initially, each layer contained in the original VGG16 is pre-trained by ImageNet dataset. For optimization of our network, we used Momentum SGD with the learning rate 10^{-5} for about 1,900 iterations, and the size of mini-batch was 16.

In the evaluation, we show the average of the relative error representing the ratio between the estimated values and ground-truth, and the absolute error representing the differ-

Table 3. The results of the real size estimation from boiled rice grains images.

Evaluation data.	abs. err.(cm/224pixels)	rel. err.(%)	corr.	$\leq$ 5% err.(%)	$\leq$ 10% err.(%)	$\leq$ 20% err.(%)
Camera:COOLPIX, Small amount of water	0.212	7.182	0.958	41.667	75.000	91.667
Camera:COOLPIX, Medium amount of water	0.178	6.550	0.973	43.333	76.667	93.333
Camera:COOLPIX, Large amount of water	0.197	6.668	0.962	48.333	78.333	90.000
Camera:iPhone8 Plus, Small amount of water	0.127	5.652	0.945	50.000	75.000	98.333
Camera:iPhone8 Plus, Medium amount of water	0.170	7.512	0.903	43.333	68.333	88.333
Camera:iPnone8 Plus, Large amount of water	0.105	4.800	0.967	58.333	88.333	98.333

Figure 11. Input patch images of a batch in training. 16 patch images are cropped from 16 boiled rice grains images respectively. Background patch images are eliminated.

Figure 12. Input patch images of a batch in evaluation. 16 patch images are cropped from one boiled rice grains images by 4×4 grid sampling. Background patch images are eliminated.

ences between both. In addition, we show the correlation coefficient between the estimated value and ground-truth and the ratio of the estimated value within the relative error of 5%, 10%, and 20%. Note that the evaluations are calculated for the estimated real size of 224 pixels.

Table 3 shows the results of the real size estimation. The average value of absolute error and relative error for the estimated real size for 224 pixels are 0.165cm and 6.394% respectively, and the average correlation coefficient is 0.951. Table 3 shows the relative errors of all evaluation data are less than 10% and correlation coefficients are more than 0.9, and most of the relative error is less than 20%.

In the work on rice grained size estimation, we construct CNN-based real size estimation from boiled rice grains images based on the size of grains of boiled rice and as a result, the correlation coefficient is more than 0.9. As future work, we plan to estimate food calories from food images considering 2D size of foods by combining our real size estimation and food segmentation.

7 Conclusion

In this paper, we reviewed our three previous works and proposed two new works. Currently, DepthCalorieCam is the most promising approach. However, large-scale calorie-annotated 3D food volume data is needed to extend the system into large-scale categories, which is very costly and time-consuming. In addition, the rice grain based method is also promising for meals containing white steamed rice. Especially it is appropriate for Japanese foods.

References

[1] K. Okamoto and K. Yanai. An automatic calorie estimation system of food images on a smartphone. In *Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management*, 2016.

[2] W. Shimoda and K. Yanai. CNN-based food image segmentation without pixel-wise annotation. In *Proc. of IAPR International Conference on Image Analysis and Processing*, 2015.

[3] R. Tanno, T. Ege, and K. Yanai. AR DeepCalorieCam V2: Food calorie estimation with cnn and ar-based actual size estimation. In *Proc. of ACM Symposium on Virtual Reality Software and Technology (VRST)*, 2018.

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

[5] R. Tanno, K. Okamoto, and K. Yanai. DeepFoodCam: A DCNN-based real-time mobile food recognition system. In *Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management*, 2016.

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

[7] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Springer*, pages 234–241, 2015.

[8] Y. Matsuda, H. Hajime, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proc. of IEEE International Conference on Multimedia and Expo*, pages 25–30, 2012.

[9] K. Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556*, 2014.