DepthCalorieCam: A Mobile Application for Volume-Based Food Calorie Estimation using Depth Cameras

Yoshikazu Ando Takumi Ege Jaehyeong Cho Keiji Yanai
The University of Electro-Communications, Tokyo
{ando-y,ege-t,cho,yanai}@mm.inf.uec.ac.jp

ABSTRACT

Some recent smartphones such as iPhone Xs have pair of cameras which can be used as stereo cameras on their backside. Regarding iPhone with iOS11 or more, the official API provides the function to estimate depth information from two backside cameras in the real-time way. By taking advantage of this function, we have developed an iOS app, "*DepthCalorieCam*", which estimates the amount of food calories based on food volumes. In the proposed app, it takes a RGB-D image of a dish, estimate categories and volumes of foods on the dish, and calculate the amount of their calories using the pre-registered calorie density of each food category. We have achieved very accurate calorie estimation by using depth information. The error of estimated calories was reduced greatly compared with the existing size-based systems.

CCS CONCEPTS

• **Computing methodologies** → *Scene understanding*; *Object recognition*.

KEYWORDS

food calorie estimation, depth camera, RGB-D, mobile app, iPhone app

ACM Reference Format:
Yoshikazu Ando Takumi Ege Jaehyeong Cho Keiji Yanai. 2019. DepthCalorieCam: A Mobile Application for Volume-Based Food Calorie Estimation using Depth Cameras. In *5th International Workshop on Multimedia Assisted Dietary Management (MADiMa '19), October 21, 2019, Nice, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3347448.3357172

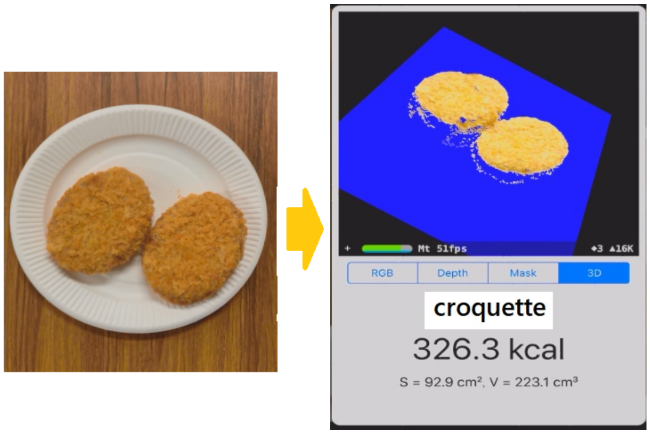


Figure 1: The proposed system, *DepthCalorieCam*.

1 INTRODUCTION

In recent years, the demand for health management using smartphones is increasing because everyone is carrying a smartphone everytime. It is desirable to record food contents and the amount of their calories by a smartphone app easily. Therefore, various apps for recording of food habits have been released so far. However, although in principle the calorie amount of foods is proportional to its volume, most of the apps do not care about the volumes of foods or ask users to provide the size of dishes even if they have the function of food image recognition.

On the other hand, the ability of smartphone to recognize the real-world information is improving year by year. Some recent smartphones such as iPhone X and Xs have a pair of cameras which can be used as stereo cameras on their backside. Regarding iPhone with iOS11 or more, the official API provides the function to estimate depth information from two backside cameras in the real-time way. By taking advantage of this function, we have developed an iOS app, "*DepthCalorieCam*", which estimates the amount of food calories based on food volumes (Figure 1). In the proposed app, it takes a RGB-D image of a dish, estimate categories and volumes of foods on the dish, and calculate the amount of their calories using the pre-registered calorie density of each food category. We achieved very accurate calorie estimation by using depth information. The relative error of estimated calories was reduced greatly compared with the existing 2D size-based food calorie estimation systems.

To summarize it, our contributions of this paper are as follows: (1) We developed an iOS app which estimates real volumes and calories of foods using RGB-D images without any reference objects. (2) We confirmed the effectiveness of RGB-D images taken by an iPhone for food calorie estimation.

2 RELATED WORKS

2.1 Food Calorie Estimation

Some works for image-based food calorie estimation have been proposed so far. Ege *et al.* estimated the calorie amount directly from a single food image without depth information using a multi-task CNN simultaneously trained with recipe information (food categories, ingredients and cooking directions) and calorie amount [6, 7].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MADiMa '19, October 21, 2019, Nice, France
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6916-9/19/10…$15.00
https://doi.org/10.1145/3347448.3357172

However, this method did not consider the size or volume of foods, and they assumed that a dish in a given image was a portion for one person. Being different from this work, in our system, we estimate the amount of food calories based on the estimated 3D volume of the foods.

Because multiple view stereo was used to be one of the main topics of computer vision research 10 years ago, some old works tried to introduce multi-view 3D reconstruction into food calorie estimation. As one of such the existing works, Puri *et al.* proposed a volume-based food calorie estimation system employing multiple-view 3D reconstruction [14]. However, to reconstruct 3D shapes of dishes accurately, many images and heavy computation were needed. Dehais et al. [5] also proposed a food volume estimation by two-view 3D reconstruction.

On the other hand, Myers *et al.* used CNN-based depth estimation from a single image for food calorie estimation [12]. They estimated the volume of food by voxelization with depth information estimated by a depth estimation CNN. They employed CNN-based segmentation [3] and CNN-based 3D volume estimation from 2D single images [8] in addition to CNN-based food category recognition. Although they achieved relatively high accuracy for volume estimation, their method needed a large amount of RGB-D food images and pixel-wise annotated segmentation masks of food images which were costly to obtain for training. Although they announced to make an Android application as well as the dataset to the public in the paper, they have not been released them so far. Allegra *et al.* [1] also tried depth and volume estimation of a food image from a single image employing a CNN. They created a new food RGB-D image dataet, Madima17, for training of a CNN. Lu *et al.* [10] proposed a multi-task CNN architecture to perform food segmentation and food volume estimation simultaneously. They extended Mask R-CNN [9] by adding depth and volume estimation networks. They used Madima17 for training of the proposed network.

Since in our system we use built-in stereo cameras in a smartphone, we do not need to take multiple-view images and to gather training samples of RGB-D food images for training of depth estimation CNN. In addition, the real volume values can be estimated by stereo vision, because all the camera parameters of the iPhone are known in advance. This is why we do not need any reference objects to estimate real volumes.

2.2 CalorieCam

Okamoto *et al.* developed "*CalorieCam*", a system that estimates the calorie amount of food simply by shooting with the camera of the smartphone [13]. *CalorieCam* can estimate food regions automatically from a single image taken by a smartphone built-in camera, and estimate the amount of their calories based on the real size of food regions. To estimate real size of foods, a user need to take a food photo with a reference object the real size of which is known in advance at the time of photographing. The system extracts both regions of foods and a reference object from a photo using GrabCut [16] as shown in the Figure 2, and estimate their size by the following equation:

$$\begin{aligned} r_S &= \frac{\text{(the number of food pixels)}}{\text{(the number of the pixels of the reference object)}} \\ S_{food} &= r_S \times \text{(the real size of the reference object)} \qquad (1) \end{aligned}$$

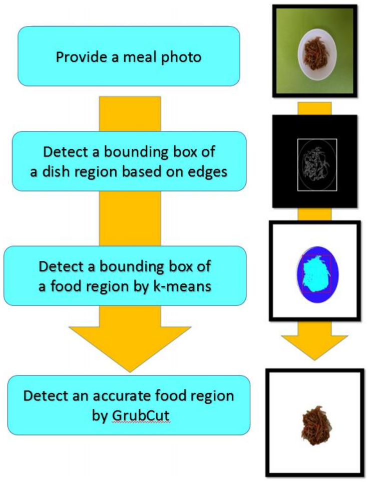

Provide a meal photo
Detect a bounding box of a dish region based on edges
Detect a bounding box of a food region by k-means
Detect an accurate food region by GrubCut

Figure 2: The procedure of food region segmentation in *CalorieCam* [13].

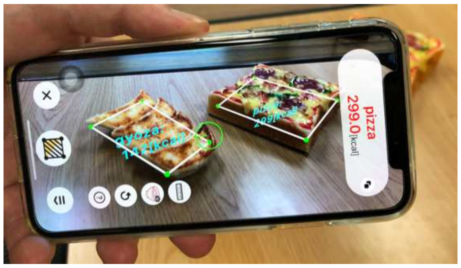


Figure 3: *AR DeepCalorieCam V2* [17].

While *CalorieCam* need to take a food photo with a size-known reference object, our system, *DepthCalorieCam*, does not need to prepare any reference objest. In addition, since we empoly CNN-based image segmentation, the accuracy of food segmentation is much higher that *CalorieCam*.

2.3 AR DeepCalorieCam V2

Tanno *et al.* have developed "*AR DeepCalorieCam V2*" which does not need any reference object to estimate the calorie of the foods by using AR [17]. By using visual inertial odometry which is a fundamental technique of AR, we can estimate the real size of things by a smartphone having only a single camera. In *AR DeepCalorieCam V2*, a user moves the smartphones and draw lines on the AR space which correspond to rough boundaries of food items to be measured as shown in Figure 3. Based on the move of a smartphone camera measured by an inertial sensor and multiple-view images, the iPhone AR library (ARKit) estimates the real size of food regions. By multiplying the food-category-dependent calorie density with the size, we can get to know the amount of food calories.

As a drawback of this system it is very difficult to draw lines of the actual food boundaries in the AR space, and the accuracy of their position affects calorie values largely. In addition, the accuracy of *AR DeepCalorieCam V2* is lower than that of volume-based estimation because calories are estimated based on the 2D area size, not the volume of the food. In our system, it is not necessary to move a smartphone, since we employs built-in stereo cameras. What a user of our system needs to do is only pushing a shutter button on the app UI.

3 PROPOSED METHOD

The process to estimate the amount of food calories in the proposed system consists of the following steps:

(1) Perform food region segmentation on an input image.
(2) Calculate the volume of the food region.
(3) Recognize the category of the food in the input image.
(4) Estimate the amount of food calories based on the estimated food volume and food category.

Note that an input image is represented as a RGB-D image which is taken by backside stereo cameras of an iPhone. In addition, we assume that the RGB-D food image is taken from right above. We explain the detail of each step in this section.

3.1 Food Segmentation

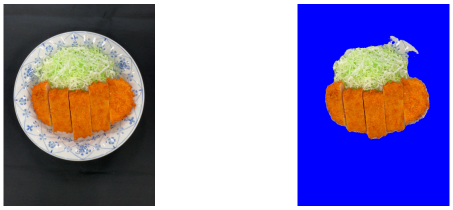
To estimate the amount of calories from a food image, segmentation of food regions is essential. In our system, we use a CNN-based segmentation method. As a model, we use a U-Net [15], which is a standard CNN-based segmentation model. As a dataset to train a segmentation network, we use 5301 images selected from UECFOOD-100 [11] after annotating them with pixel-wise segmentation masks of food regions by ourselves. At the time of training, we use 4771 images as training samples and 530 images as testing samples. The trained U-Net achieved 0.800 mean IoU (Intersection over Union) regarding testing samples. We show an example of a segmentation result in Figure 4. In this example, the complicated shape of foods was successfully separated from the backgroud.

3.2 Food Volume Estimation from a RGB-D Image

To estimate the volume of foods, we divide 3D food objects into many small pieces of elongated rectangular parallelepiped, calculate the volume of each piece by multiplying their depth and area, and finally sum up them like a surface integral.

As a preliminary step of estimation, we estimate the distance between a camera and a reference plane on which foods are put, typically a dish. To make estimation of the food volume simpler, we assume the following conditions:

- A reference plane is flat.
- A reference plane and a smartphone taking a food photo are in parallel.
- A reference plane is located in the farthest in a camera view. This means no other objects are located further than the plane in the camera view.



(a) Original image

(b) Segmented result

Figure 4: An example of food segmentation with U-Net.

In general, when foods are located on the center of the camera view, the distance CR between a camera C and a reference plane R (that is, depth of R) cannot be obtained directory in the RGB-D image as shown in Figure 5. This is a common situation. Therefore, in our system, we estimate the distance CR by averaging the values of multiplying the depth from the depths of the middle point of the upper, lower, left, and right sides of the view (R_N, R_S, R_E, R_W) and the cosine values of their camera angles ($\angle R_N C R_S/2$, $\angle R_E C R_W/2$ in Figure 5), which is represented in Eq.(2), and use it as a distance from the camera to the reference plane.

$$CR = (1/4) \sum_{X \in \{N, E, S, W\}} CR_X \cos \angle RCR_X \tag{2}$$

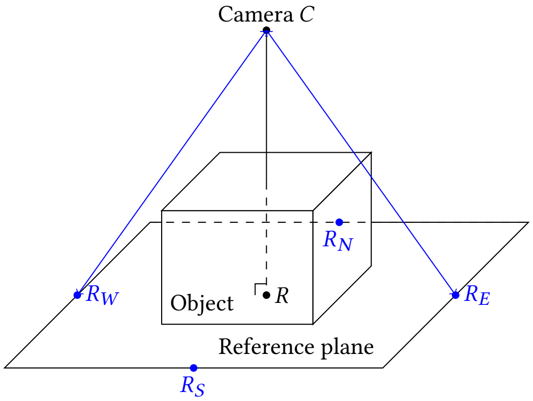


Figure 5: The spatial relation between a camera and a reference plane.

Next, we estimate the actual size of the view area on the reference plane ($R_N R_S$ and $R_E R_W$ in Figure 5) and the actual area on the reference plane per pixel in the image S_{pixel} shown in Figure 6. These could be calculated from the camera angle, the distance CR between the camera and the reference plane which was calculated previously, and the total number of pixels N_{pixel} in Eq.(3) ~(5).

$$R_N R_S = 2CR \tan \frac{\angle R_N C R_S}{2} \tag{3}$$

$$R_E R_W = 2CR \tan \frac{\angle R_E C R_W}{2} \tag{4}$$

$$S_{pixel} = \frac{R_N R_S \times R_E R_W}{N_{pixel}} \tag{5}$$

Finally, we compute the volume of each piece of the elongated rectangular parallelepiped which corresponds to each of the pixels in the RGB-D image and sum up them within the food region P. We define the distance between the camera and the surface of the foods (the depth of the food surface) as z_p and the distance between the camera and the reference plane (the depth of the reference plane) as $z_{ref} = CR$ as shown in Figure 6. We calculate an expansion rate r_p which represents how large the unit surface corresponding to one pixel is compared to the unit area in the reference plane (Figure 6) by the following equation:

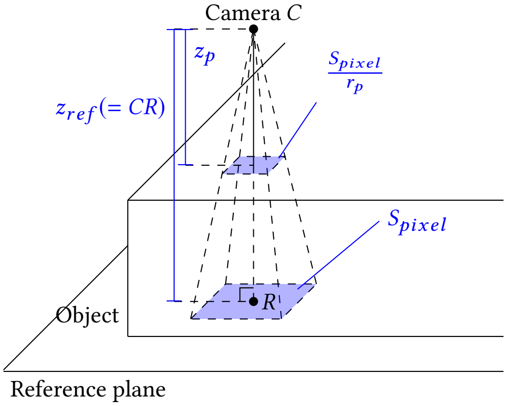


Figure 6: Volume estimation (R is the center of pixel)

$$r_p = \left(\frac{z_{ref}}{z_p}\right)^2 (p \in P) \tag{6}$$

We compute the volume of each piece and aggregate them to obtain the volume of the food as V.

$$V = \sum_{p \in P} \frac{S_{pixel}}{r_p}(z_{ref} - z_p) \tag{7}$$

Note that the volume estimated by the above-mentioned method is valid only if the following conditions hold:

- The reference plane is viewed from the right top. (CR_E, CR_S, CR_W, CR_N are equal.)
- The food has the same shape as the top surface in any height until the reference plane as shown in Figure 7.

Therefore, the volume estimated by the proposed method might include an empty space that cannot be seen from the right above as shown in Figure 7. If a target object has a complex shape, it is expected that the difference between an estimated volume and a true volume occurs. To compensate it, we train the parameters for estimation of the calorie amount from the estimated volume and the amount of actual food calories even if the case like Figure 7 happens.

3.3 Food Category Estimation

In our system, we need to know the food category of a target food item, because calorie density depends on a food category. To classify a food category of the target, we use Xcepition [4] which is considered to have relatively higher performance than that used for food category recognition in the existing works. We fine-tune an ImageNet-pre-trained model with Food-101 [2], and fine-tune it again with a part of UECFOOD-100 [11] which contains only target food categories (described later).

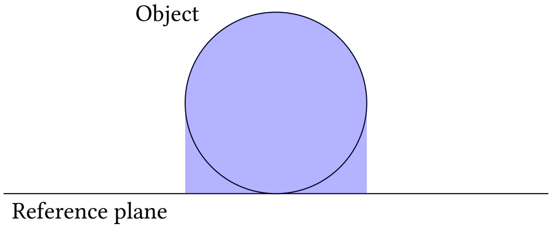


Figure 7: The volume the proposed method can estimate (marked with violet regions). Even if the object is a sphere, we assume the invisible parts from the top are elongated until the reference plane.

3.4 Regression of the Amount of Food Calories

The amount of food calories is calculated by applying the estimated food volume to a regression equation of the calorie amount the parameters of which are trained in advance for each food category. The equation is represented by a simple linear transform:

$$Cal = w_{i,1} * V + w_{i,0} \tag{8}$$

where Cal represents the estimated calories, $w_{i,1}$ and $w_{i,0}$ are parameters to be trained from actual food calories and the estimated volume, and i represents the index of food categories.

For the experiments, we trained the parameters for estimating the calorie amount from the volume on three categories consisting of "sweet and sour pork", "fried chicken", and "croquette". As training samples, we gathered three sizes (small, medium, and large) of calorie-known food samples, and took RGB-D images from the distance of the range from 20 to 70cm.

4 EXPERIMENTS

We implemented the proposed method as an iOS app, *DepthCalorieCam*. We used this app for performance evaluation. To evaluate the propose system, we compared it with the 2D-area-based food calorie estimation systems, *CalorieCam* and *AR DeepCalorieCam V2*, both of which are mobile apps. Note that although the original *CalorieCam* was implemented as an Android app three years ago, which cannot run on the current version of Android OS, we re-implemented the *CalorieCam* as an iOS app for the evaluation.

In the experiment, we used the above-mentioned three categories of foods with the instances which were different from ones used in the training time. The calorie amount of all of the test samples are known for evaluation of estimation accuracy. We shows example photos of three categories of the foods and their actual calories in Figure 8 and Table 1.

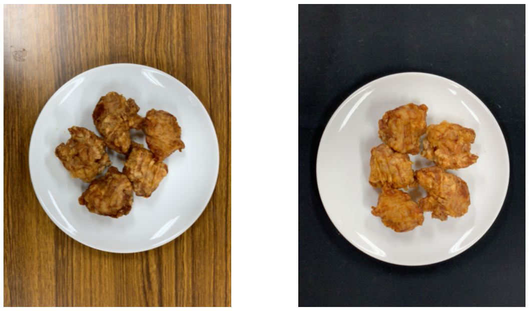
Table 1: The amount of calories and the weights of Eq.8 for each foods.

Category	Calorie [kcal]	w_1	w_0
Croquette	246	1.43	7.30
Fried chicken	655	1.91	53.4
Sweet and sour pork	500	1.50	33.1

(a) Croquette **(b) Fried chicken** **(c) Sweet and sour pork**

Figure 8: The foods used for the experiments.

The experiments were conducted on six different settings in which the three categories of the foods were provided in a dish and two kinds of a reference plane as shown in Figure 9. For each pattern, three subjects who do not have special knowledge about the amount of calories measured five times using each of the three mobile apps, making a total of 270 measurements.



(a) Textured **(b) Non-textured**

Figure 9: Examples of textured/non-textured reference planes.

The mean and standard deviation of the errors in the calorie estimates for each food category system are shown in Table 2. This table shows that for the three categories of food used in this experiment, it has become possible to estimate the calorie amount with accuracy higher than that of the existing method. The proposed system achieved a high accuracy especially in "fried chicken" and "sweet and sour port". The average absolute errors were reduced to 1/125 and 1/70 for "fried chicken", and 1/56 and 1/180 for "sweet and sour port" compared with CalorieCam [13] and AR DeepCalorieCam V2 [17], respectively. This is because the existing methods employ 2-D size based calorie estimation and the calories are regressed using the second-order regression equation, which means that it is easily susceptible to the error of the estimated surface area. On the other hand, the proposed system employs linear regression based on the estimated volume. Therefore, the amount of calories estimated by the proposed 3D-based method is less susceptible to volume errors than 2D area based methods.

Table 3 shows the results of each of the cases with textured/non-textured reference planes. The results were hard to be affected by the differences on the types of the reference planes and the user's photographing methods, and the stability of the calorie estimation can be greatly improved compared with the existing methods.

Table 2: The mean±standard deviation of the errors in estimated calories.

Category	CalorieCam	AR CalorieCam V2	DepthCalorieCam
Sweet and sour pork	364±552	-112±163	2±52
Fried chicken	-123±171	343±51	-5±64
Croquette	-48±16	-104±12	-35±22

Table 3: The mean±standard deviation of the errors in estimated calories with a textured/non-textured plane (unit: kcal)

Category	Texture	CalorieCam	AR CalorieCam V2	DepthCalorieCam
Sweet and sour pork	Textured	571±713	-130±224	1±28
	Non-textured	156±177	-93±65	2±69
Fried chicken	Textured	15±129	-364±64	17±68
	Non-textured	-260±58	-321±19	-27±52
Croqutte	Textured	-46±21	-111±10	-28±19
	Non-textured	-50±11	-96±9	-43±24

5 CONCLUSIONS AND FUTURE WORKS

We have developed an iOS app, "*DepthCalorieCam*", which estimates the amount of food calories based on food volumes. In the proposed app, it takes a RGB-D image of a dish, estimate categories and volumes of foods on the dish, and calculate the amount of their calories using the pre-registered calorie density of each food category. We have confirmed that the proposed app can perform very accurate calorie estimation by using depth information. The error of estimated calories was reduced greatly compared with the existing size-based systems.

For future work, we plan to increase the number of food categories the system can handle with. The current system supports only three kinds of food categories, which is very smaller than the existing systems. Due to the structure of the proposed system, to add new food categories, we need to take RGB-D images of real foods the calorie amount of which are known by using iPhones having two backside cameras. This is very costly. However, since calorie amounts can be estimated much more accurately than the existing systems, it can be considered that the accuracy of depth and volume estimation have reached practical level. So we think it is worth paying cost to increase the number of food categories the proposed app can treat with.

In addition, to make the system more practical, we like to extend the system so as to estimate calories in the cases of both multiple dishes and multiple foods on a single dish, since the current system assumes that one dish has only one kinds of foods, which seems strong limitation for practical use.

Note that "*DepthCalorieCam*" can be downloaded from the iOS app store.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 15H05915, 17H01745, 17H06100 and 19H04929.

REFERENCES

[1] D. Allegra, M. Anthimopoulos, J. Dehais, Y. Lu, F. Stanco, G. M. Farinella, and S. Mougiakakou. 2017. A Multimedia Database for Automatic Meal Assessment Systems. In *Proc. of International Workshop on Multimedia Assisted Dietary Management (MADiMa)*.

[2] L. Bossard, M. Guillaumin, and L. Van Gool. 2014. Food-101 – Mining Discriminative Components with Random Forests. In *Proc. of European Conference on Computer Vision (ECCV)*. 446–461.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. 2014. Semantic image segmentation with deep convolutional nets and fully connected CRFs.

[4] F. Chollet. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 1800–1807.

[5] J. Dehais, M. Anthimopoulos, S. Shevchik, and S. Mougiakakou. 2017. Two-View 3D Reconstruction for Food Volume Estimation. *IEEE Transactions on Multimedia* 19, 5 (2017), 1090–1099.

[6] T. Ege and K. Yanai. 2017. Image-Based Food Calorie Estimation Using Knowledge on Food Categories, Ingredients and Cooking Directions. In *Proc. of ACM Multimedia Thematic Workshops on Understanding*.

[7] T. Ege and K. Yanai. 2018. Image-Based Food Calorie Estimation Using Recipe Information. *IEICE Transactions on Information and Systems* E101-D, 5 (2018), 1333–1341.

[8] D. Eigen and R. Fergus. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolution al architecture. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2650–2658.

[9] K. He, G. Gkioxari, P. Dollar, and R. Girshick. 2016. Mask R-CNN. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*.

[10] Y. Lu, D. Allegra, M. Anthimopoulos, F. Stanco, G. M. Farinella, and S. G. Mougiakakou. 2018. A Multi-Task Learning Approach for Meal Assessment. In *Proc. of International Workshop on Multimedia Assisted Dietary Management (MADiMa)*.

[11] Y. Matsuda, H. Hoashi, and K. Yanai. 2012. Recognition of Multiple-Food Images by Detecting Candidate Regions. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*.

[12] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy. 2015. Im2Calories: towards an automated mobile vision food diary. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 1233–1241.

[13] K. Okamoto and K. Yanai. 2016. An Automatic Calorie Estimation System of Food Images on a Smartphone. In *Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management (MADiMa)*.

[14] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney. 2009. Recognition and Volume Estimation of Food Intake Using a Mobile Device. In *Proc. of Workshop on Applications of Computer Vision (WACV)*.

[15] O Ronneberger, P Fischer, and T Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 234–241.

[16] C. Rother, V. Kolmogorov, and A. Blake. 2004. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, Vol. 23. 309–314.

[17] R. Tanno, T. Ege, and K. Yanai. 2018. 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)*.